Annals of Information Systems 17

Mahmoud Abou-Nasr
Stefan Lessmann
Robert Stahlbock
Gary M. Weiss *Editors*

Real World Data Mining Applications

Springer

Annals of Information Systems

Volume 17

Annals of Information Systems comprises serialized volumes that address a specialized topic or a theme. AoIS publishes peer reviewed works in the analytical, technical as well as the organizational side of information systems. The numbered volumes are guest-edited by experts in a specific domain. Some volumes may be based upon refereed papers from selected conferences. AoIS volumes are available as individual books as well as a serialized collection. *Annals of Information Systems* is allied with the 'Integrated Series in Information Systems' (IS^2).

Proposals are invited for contributions to be published in the *Annals of Information Systems*. The Annals focus on high quality scholarly publications, and the editors benefit from Springer's international network for promotion of your edited volume as a serialized publication and also a book. For more information, visit the Springer website at http://www.springer.com/west/home/authors

Or contact the series editors by email

Ramesh Sharda: sharda@okstate.edu or Stefan Voß: stefan.voss@uni-hamburg.de

More information about this series at http://www.springer.com/series/7573

Mahmoud Abou-Nasr • Stefan Lessmann • Robert Stahlbock • Gary M. Weiss
Editors

Real World Data Mining Applications

Springer

Editors
Mahmoud Abou-Nasr
Research & Advanced Engineering
Ford Motor Company
Dearborn
Michigan
USA

Robert Stahlbock
Universität Hamburg Inst.
Wirtschaftsinformatik
Hamburg
Germany

Stefan Lessmann
Universität Hamburg Inst.
Wirtschaftsinformatik
Hamburg
Germany

Gary M. Weiss
Deptartment of Computer & Information Science
Fordham University
Bronx
New York
USA

ISSN 1934-3221
ISBN 978-3-319-07811-3
DOI 10.1007/978-3-319-07812-0
Springer Cham Heidelberg New York Dordrecht London

ISSN 1934-3213 (electronic)
ISBN 978-3-319-07812-0 (eBook)

Library of Congress Control Number: 2014953600

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Acknowledgments

We would like to thank all authors who submitted their work for consideration to this focused issue. Their contributions made this special issue possible. We would also like to thank the referees for their time and thoughtful reviews. Finally, we are grateful to Ramesh Sharda and Stefan Voß, the two series editors, for their valuable advice and encouragement, and the editorial staff at Springer for their support in the production of this special issue.

Dearborn, Hamburg, New York
June 2013

Mahmoud Abou-Nasr
Stefan Lessmann
Robert Stahlbock
Gary M. Weiss

Contents

Contributors

Mahmoud Abou-Nasr Research & Advanced Engineering, Research & Innovation Center, Ford Motor Company, Dearborn, MI, USA

Eya Ben Ahmed Higher Institute of Management of Tunis, University of Tunis, Tunis, Tunisia

Peter Azo Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Julien Beasley Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Mahua Bhattacharya ABV – Indian Institute of Information Technology and Management, Gwalior, MP, India

Eric Bjorklund Computer Sciences Corporation, Falls Church, VA, USA

Aishwarya Bose School of Electrical Engineering and Computer Science, Science and Engineering Technology, Queensland University of Technology, Brisbane, Australia

Wesley Brandi Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Laurent Brisson UMR CNRS 6285 Lab-STICC, Institut Telecom, Telecom Bretagne, Brest Cedex 3, France

Kathryn Burn-Thornton OUDCE, University of Oxford, Oxford UK

Tim Burman School of Computing and Engineering Science, University of Durham, Durham, UK

Ning Chen GECAD, Instituto Superior de Engenharia do Porto, Porto, Portugal

Catherine Cheung National Research Council Canada, Ottawa, ON, Canada

Fabrice Clérot Orange Labs, Lannion, France

Nicolas Creff Orange Labs, Lannion, France

Peter Eisert Fraunhofer Heinrich Hertz Institute, Berlin, Germany

John Ettedgui Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Ya Ju Fan Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA

Dimitar Filev Research and Advanced Engineering, Research & Innovation Center, Ford Motor Company, Dearborn, MI, USA

Roy B. Friedenthal Central Orthopedics, Hammonton, NJ, USA

Faïez Gargouri Higher Institute of Computer Science and Multimedia of Sfax, Sfax University, Sfax, Tunisia

Feng Gao Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Émilien Gauthier Statlife company, Institut Gustave Roussy, Villejuif Cedex, France

UMR CNRS 6285 Lab-STICC, Institut Telecom, Telecom Bretagne, Brest Cedex 3, France

Steve Gorrell Brigham Young University, Provo UT, USA

Michael Gosnell 21st Century Systems, Inc., Omaha NE, USA

Craig Hartrick Anesthesiology Research, School of Medicine, Oakland University, Rochester, MI, USA

Kuo-Wei Hsu Department of Computer Science, National Chengchi University, Taipei, Taiwan (ROC)

Yinghao Huang Computer and Information Science, University of Michigan—Dearborn, Dearborn, MI, USA

Chandrika Kamath Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA

Naoki Kaisaku Graduate School of Engineering, Osaka Prefecture University, Osaka, Japan

Brendan Kitts Microsoft Corporation, One Microsoft Way, Redmond,WA, USA

Aaron Lai Market Analytics, Blue Shield of California, San Francisco, CA, USA

Vincent Lemaire Orange Labs, Lannion, France

Philippe Lenca UMR CNRS 6285 Lab-STICC, Institut Telecom, Telecom Bretagne, Brest Cedex 3, France

Stefan Lessmann Institute of Information Systems, University of Hamburg, Hamburg, Germany

Matthew Li National Research Council Canada, Ottawa, ON, Canada

Raj Mahato Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Jean Martinet Lille 1 University, Villeneuve d'ascq, Lille, France

Kieran Morrill Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

John Michelini Research and Advanced Engineering, Research & Innovation Center, Ford Motor Company, Dearborn, MI, USA

Clifton Mortensen Brigham Young University, Provo UT, USA

Yi Lu Murphey Electrical and Computer Engineering, University of Michigan—Dearborn, Dearborn, MI, USA

Ahlem Nabli Faculty of Sciences of Sfax, Sfax University, Sfax, Tunisia

Richi Nayak School of Electrical Engineering and Computer Science, Science and Engineering Technology, Queensland University of Technology, Brisbane, Australia

Nishith Pathak Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Guangzhi Qu Computer Science and Engineering Department, Oakland University, Rochester, MI, USA

Dharmveer Singh Rajput ABV – Indian Institute of Information Technology and Management, Gwalior, MP, India

Stéphane Ragusa Statlife company, Institut Gustave Roussy, Villejuif Cedex, France

Bernardete Ribeiro CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

Ryosuke Saga Graduate School of Engineering, Osaka Prefecture University, Osaka, Japan

Ismail El Sayad Fraunhofer Heinrich Hertz Institute, Berlin, Germany

Ishwar Sethi Computer Science and Engineering Department, Oakland University, Rochester, MI, USA

Naeem Seliya Computer and Information Science, University of Michigan—Dearborn, Dearborn, MI, USA

Sid Siddhartha Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Pramod Kumar Singh ABV – Indian Institute of Information Technology and Management, Gwalior, MP, India

Jaideep Srivastava Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Robert Stahlbock University of Hamburg, Institute of Information Systems, Hamburg, Germany

FOM University of Applied Sciences Essen/Hamburg, Germany

Hiroshi Tsuji Graduate School of Engineering, Osaka Prefecture University, Osaka, Japan

Greg Tschida Department of Revenue, State of Minnesota, St. Paul, MN, USA

Julio J. Valdés National Research Council Canada, Ottawa, ON, Canada

Gary M. Weiss Department of Computer & Information Science, Fordham University, Bronx, NY, USA

Gang Wu Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Robert Woodley 21st Century Systems, Inc., Omaha NE, USA

Hong Yuan Microsoft Corporation, One Microsoft Way, Redmond, WA, USA

Hui Zhang State Key Laboratory of Software Development Environment, School of Computer Science, Beihang University, Beijing, China

Jing Ying Zhang Microsoft Corporation, One Microsoft Way, Redmond,WA, USA

Zhongfei (Mark) Zhang Computer Science Department, SUNY at Binghamton, NY, USA

Editors' Biographies

Dr. Mahmoud Abou-Nasr is a Senior Member of the IEEE and Vice Chair of the Computational Intelligence & Systems Man and Cybernetics, Southeast Michigan Chapter. He has received the B.Sc. degree in Electrical Engineering in 1977 from the University of Alexandria, Egypt, the M.S. and the Ph.D. degrees in 1984 and 1994 respectively from the University of Windsor, Ontario, Canada, both in Electrical Engineering. Currently he is a Technical Expert with Ford Motor Company, Research and Advanced Engineering, Modern Control Methods and Computational Intelligence Group, where he leads research & development of neural network and advanced computational intelligence techniques for automotive applications. His research interests are in the areas of neural networks, data mining, machine learning, pattern recognition, forecasting, optimization and control. He is an adjunct faculty member of the computer science department, Wayne State University, Detroit, Michigan and was an adjunct faculty member of the operations research department, University of Michigan Dearborn. Prior to joining Ford, he held electronics and software engineering positions with the aerospace and robotics industries in the areas of real-time control and embedded communications protocols. He is an associate editor of the DMIN'09-DMIN'14 proceedings and a member of the program and technical committees of IJCNN, DMIN, WCCI, ISVC, CYBCONF and ECAI. He is also a reviewer for IJCNN, MSC, CDC, Neural Networks, Control & Engineering Practice and IEEE Transactions on Neural Networks & Learning Systems. Dr. Abou-Nasr has organized and chaired special sessions in DMIN and IJCNN conferences, as well as international classification competitions in WCCI 2008 in Hong Kong and IJCNN2011 in San Jose CA.

Dr. Stefan Lessmann received a M.Sc. and a Ph.D. in Business Administration from the University of Hamburg (Germany) in 2001 and 2007, respectively. He is currently employed as a lecturer in Information Systems at the University of Hamburg. Stefan is also a member of the Centre for Risk Research at the University of Southampton, where he teaches courses in Management Science and Information Systems. His research concentrates on managerial decision support and advanced analytics in particular. He is especially interested in predictive modeling to solve

planning problems in marketing, finance, and operations management. He has published several papers in leading scholarly outlets including the European Journal of Operational Research, the ICIS Proceedings or the International Journal of Forecasting. He is also involved with consultancy in the aforementioned domains and has completed several technology-transfer projects in the publishing, the automotive and the logistics industry.

Dr. Robert Stahlbock holds a diploma in Business Administration and a PhD from the University of Hamburg (Germany). He is currently employed as a lecturer and researcher at the Institute of Information Systems at the University of Hamburg. He is also lecturer at FOM University of Applied Sciences (Germany) since 2003. His research interests are focused on managerial decision support and issues related to maritime logistics and other industries as well as operations research, information systems and business intelligence. He is author of research studies published in international prestigious journals as well as conference proceedings and book chapters and serves as reviewer for international leading journals as well as a member of conference program committees. He is General Chair of the International Conference on Data Mining (DMIN) since 2006.

Dr. Gary Weiss is an Associate Professor in the Computer and Information Science Department at Fordham University in New York City. His current research involves the mining of sensor data from smartphones and other mobile devices in support of activity recognition and related applications. His Wireless Sensor Data Mining (WISDM) Lab recently released the actitracker activity tracking app (actitracker.com). Prior to coming to Fordham, Dr. Weiss worked at AT&T Labs as a software engineer, expert system developer, and as a data scientist. He received a B.S. degree in Computer Science from Cornell University, an M.S. degree in Computer Science from Stanford University, and a Ph.D. degree in Computer Science from Rutgers University. He has published over 50 papers in machine learning and data mining and his research is supported by funding from the National Science Foundation, Google, and Citigroup.

Introduction

Mahmoud Abou-Nasr, Stefan Lessmann, Robert Stahlbock and Gary M. Weiss

Abstract Data Mining involves the identification of novel, relevant, and reliable patterns in large, heterogeneous data stores. Today, data is omnipresent, and the amount of new data being generated and stored every day continues to grow exponentially. It is thus not surprising that data mining and, more generally, data-driven paradigms have successfully been applied in a variety of different fields. In fact, the specific data-oriented problems that arise in such different fields and the way in which they can be overcome using analytic procedures have always played a key role in data mining. Therefore, this special issue is devoted to real-world applications of data mining. It consists of eighteen scholarly papers that consolidate the state-of-the-art in data mining and present novel, creative solutions to a variety of challenging problems in several different domains.

This introductory statement might appear rather strange at first glance. After all, this is a special issue on data mining. So how could it be dead, and why? And isn't data mining more relevant and present than ever before? Yes it is. But under which label? We all observe new, more glorious and promising concepts (labels) emerging and slowly but steadily displacing data mining from the agenda of CTO's. This is no longer the time of data mining. It is the time of big data, X-analytics (with X

R. Stahlbock (✉)
University of Hamburg, Institute of Information Systems, Von-Melle-Park 5,
20146 Hamburg, Germany
e-mail: robert.stahlbock@uni-hamburg.de

FOM University of Applied Sciences
Essen/Hamburg, Germany

M. Abou-Nasr
Research & Advanced Engineering, Research & Innovation Center,
Ford Motor Company, Dearborn, MI, USA

S. Lessmann
Institute of Information Systems, University of Hamburg, Von-Melle-Park 5,
20146 Hamburg, Germany

G. M. Weiss
Department of Computer & Information Science,
Fordham University, 441 East Fordham Road, Bronx, NY, USA
e-mail: gaweiss@fordham.edu

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_1

$\in$ {advanced, business, customer, data, descriptive, healthcare, learning, marketing, predictive, risk, ... }), and data science, to name only a few such new and glorious concepts that dominate websites, trade journals, and the general press. Probably many of us witness these developments with a knowing smile on their faces. Without disregarding the—sometimes subtle—differences between the concepts mentioned above, don't they all carry at their heart the goal to leverage data for a better understanding of and insight into real-world phenomena? And don't they all pursue this objective using some formal, often algorithmic, procedure? They do; at least to some extent. And isn't that then exactly what we have been doing in data mining for decades? So yes, data mining, more specifically the *label* data mining, has lost much of its momentum and made room for more recent competitors. In that sense, data mining is dead; or dying to say the least. However, the very idea of it, the idea to think of massive, omnipresent amounts of data as strategic assets, and the aim to capitalize on these assets by means of analytic procedures is, indeed, more relevant and topical than ever before. It is also more accepted than ever before. This is good news and actually a little funny. Funny because we, as data miners, now find ourselves in the position statisticians have been ever since the advent of data mining. New players in a market that we feel belongs to us: the data analysis market. It may be that the relationship between data mining and statistics, which has not always been perfectly harmonic, benefits from these new players. That would just be another positive outcome. However, the main positive point to make here is that we have less urge to defend our belief that data can tell you a lot of useful things in its own right, with and also without a formal theory how the data was generated. This belief is very much embodied in the shining light of 'big data' and its various cousins. In that sense, we may all rejoice: long live data mining.

After this casual and certainly highly subjective discussion which role data mining plays in todays IT landscape and how it relates to neighboring concepts, it is time to have a closer look at this special issue. While various new terms may arise to replace 'data mining', ultimately the field is defined by the *problems* that it addresses. Problems are in fact one of the defining characteristics of data mining and why the data mining community formed from the machine learning community (and to a much lesser extent from the statistics community). Machine learning methods for analyzing data have generally eschewed other methods, such as approaches that were mainly considered to be statistical (e.g., linear and logistic regression although they now are sometimes covered in machine learning textbooks). Furthermore, much of the work in machine learning tended to focus on small data sets and ignore the complexities that arise when handling large, complex, data sets. To some degree, data mining came into being to handle these complexities, and thus has always been defined by real-world problems, rather than a specific type of method. But even though this is true, it is still often difficult to find comprehensive descriptions of real-world data mining applications. We attempt to address this deficiency in this special issue by focusing it on real-world applications and methods that specifically address characteristics of real-world problems.

The special issue strives to consolidate recent advances in data mining and to provide a comprehensive overview of the state-of-the-art in the field. It includes 18

articles, some of which were initially presented at the International Conference on Data Mining (DMIN) in 2011 and 2012. All articles had to pass a rigorous peer-review process. Especially the DMIN conference papers had to be revised and extended by adding much new material prior to submission to the special issue. The best articles coming out of this process have been selected for inclusion into the special issue. Every article among the final set of accepted submissions is a remarkable proof of the authors' creativity, diligence, and hard work. Their countless efforts to turn a good paper into an excellent one make this special issue a *special* issue.

The articles in the special issue are concerned with real-world data mining applications and the methodology to solve problems that arise in these applications. Accordingly, we group the articles in this special issue into different categories, depending on the application domain they consider. The five articles in Part I consider classic data mining tasks such as supervised classification or clustering and propose methodological advancements to address important modeling challenges. For example, the contributions of these articles could be associated with novel algorithms, modifications of existing algorithms, or a goal-oriented combination of available techniques, to enhance the efficiency and/or effectiveness with which the data mining task in question can be approached. Although such advancements are typically evaluated in a case-study, the emphasize on well-established data mining tasks suggests that the implications of these articles and the applicability of the proposed approaches in particular may reach well beyond the case-study context. The articles in the following parts of this book focus even more on the application context. Looking into modeling tasks in management (Part II), fraud detection (Part III), medical diagnosis and healthcare (Part IV), and, last but not least, engineering (Part V), these articles elaborate in much detail the relevance of the focal application, what challenges arise in this application, and how these can be addressed using data mining techniques. The specific requirements and characteristics of modeling a problem will often necessitate some algorithmic modification, which is then assessed in the context of the specific application. As such, the articles in this group provide valuable advice advice how to tackle challenging modeling problems on the basis of available technology.

We hope that the academic community and practitioners in the industry will find the eighteen articles in this volume interesting, informative, and useful. To help the readers navigate through the special issue, we provide a brief summary of each contribution in the following sections.

1 Articles Focusing on Established Data Mining Tasks

To some extent, it is a matter of debate what modeling tasks to consider 'established' in data mining. Although any textbook on data mining includes a discussion on such 'standard data mining tasks' in one of the introductory chapters, we typically observe some variation which specific tasks are mentioned under this headline. However, the most established data mining task, actually the common denominator among all

more specialized tasks, is to learn from data. In that sense, the article of Lai (this volume) serves just as a perfect introduction to the special issue. Discussing 'What Data Scientists can Learn from History', the article very much sticks out from what we normally find in the academic literature. Lai reviews different historic events and reasons the potential of data analytics in these settings, had it been available at the time. The examples are ancient but their implications are not. Referring to his cases, Lai discusses the do's and don'ts of data analytics and elaborates different ways in which it can truly add value. The exposition is somewhat philosophical, offers a number of great ideas to think about and sets the scene for applied work in data mining.

Looking more closely on common data mining tasks, one comes across association rule mining. Association rule mining represents the main analytical omnibus to perform market basket analysis. Various real-world applications demonstrate its suitability to, e.g., improve the shop layout of retail stores or cross-sell products on the Internet. Ahmed et al. (this volume) concentrate on 'On Line Mining of Cyclic Association Rules From Parallel Dimension Hierarchies,' in multi-dimensional data warehouses and OLAP cubes in particular. Data warehouses are vital components of any business intelligence strategy and OLAP is arguably the most popular technology to support managerial decision making. For example, the multi-dimensional structure of an OLAP cube allows analysts to explore numerical data, say sales figures, from multiple different angles (geographic dimension, time dimension, product/product category dimension, etc.) to gain a comprehensive understanding of the data and discover hidden patterns. However, a potential problem with this approach is that the multi-dimensional structure of the cube and parallel hierarchies in particular also conceal certain patterns that might be of relevance to the business. This is where the approach of Ahmed et al. offers a solution. They develop a theoretical framework and a formal algorithm for mining multi-level hybrid cyclic patterns from parallel dimensional hierarchies.

Clustering is another very classic data mining task. It has been successfully applied in gene expression analysis, metabolic screening, customer recommendation systems, text analytics, and environmental studies, to name only a few. Although a variety of different clustering techniques have been developed, segmenting high-dimensional data remains a challenging endeavor. First, the observations to be clustered become equidistant in high-dimensional spaces, so that common distance metrics fail to signal whether objects are similar or dissimilar. Second, several—equally valid—cluster solutions may be embedded in different sub sets of the high dimensional space. The article 'PROFIT: A Projected Clustering Technique,' by Rajput et al. (this volume) addresses these problems. Rajput et al. propose a hybrid subspace clustering method that works in four stages. First, a representative sample of the high dimensional dataset is drawn making use of principal component analysis. Second, suitable initial clusters are identified using the concept of trimmed means. Third, all dimensions are assessed in terms of the Fisher criterion and less informative dimensions are discarded. Finally, the projected cluster solutions are obtained using an iterative refinement algorithm. Empirical experiments on well-established

test cases demonstrate that the proposed approach outperforms several challenging benchmarks under different experimental conditions.

Turning attention to the field of supervised data mining, classification analysis is clearly a task that attracted much attention from both industry and academia. More recently, we observe increasing interest in the field of multi-label classification. Again, many approaches have already been proposed, but the critical issue of how to combine single labels to form a multi-label remains a challenge. Qu et al. (this volume) tackle this problem and propose 'Multi-Label Classification with a Constrained Minimum Cut Model'. This approach uses a weighted label graph to represent the labels and their correlations. The multi-label classification problem is then transformed into finding a constrained minimum cut of the weighted graph. Compared with existing approaches, this approach starts from a global optimization perspective in choosing multi-labels. They show the effectiveness of their approach with experimental results.

A well-known yet unsolved issue in classification analysis, and more generally data mining, involves identifying informative features among a set of many, possibly highly correlated, attributes. The article 'On the Selection of Dimension Reduction Techniques for Scientific Applications,' by Fan et al. (this volume) investigates the performance of different variable selection approaches ranging from feature subset selection to methods that transform the features into a lower dimensional space. Their investigation is done through a series of carefully designed experiments on real-world datasets. They also discuss methods that calculate the intrinsic dimensionality of a dataset in order to understand the reduced dimension. Using several evaluation strategies, they show how these different methods can provide useful insights into the data. The article provides guidance to users on the selection of a dimensionality reduction technique for their dataset.

Finally, an interesting field in supervised data mining concerns analyzing and forecasting time series data. An important problem in time series data mining is related with the detection of structural breaks in the time series. Intuitively, a substantial structural break in a time series renders forecasting models that extrapolate past movements of the time series invalid. Therefore, it is important to update or rebuild the forecasting model subsequent to structural breaks. Surprisingly little research has been devoted to the question how exactly this updating should be organized and, more specifically, which data should be employed for this purpose (e.g., old data is available but invalid, whereas new, representative data is scarce). Saga et al. (this volume) address this issue in their article 'Relearning Process for SPRT in Structural Change Detection of Time-Series Data'. They propose a relearning method which updates forecasting models on the basis of the sequential probability ratio test (i.e., a common test for detecting structural change points). Within their approach, Saga et al. make use of classic regression modeling to determine the amount of data that is used for relearning after detecting the structural change point in the time series. Empirical experiments on synthetic and real-world data evidence that model updating with the proposed relearning algorithm increases forecasting accuracy compared to (i) not updating forecasting models at all, and (ii) updating forecasting models with previous approaches.

2 Articles Focusing on Business and Management Tasks

Extracting managerial insight from large data stores and thus improving corporate decision making is an area where data mining has had several success. We have seen special issues on data mining in leading management and Operations Research journals and much of the current excitement about big data, analytics, etc. comes from the business world and the potential data-driven technologies offer in this environment. Two articles in the special issue illustrate this potential.

The article 'K-means Clustering on a Classifier-Induced Representation Space: Application to Customer Contact Personalization' considers a customer relationship management (CRM) setting. In particular, Lemaire et al. (this volume) discuss the problem of customer contact personalization, which is concerned with the appetency of a customer to buy a new product. Based on their model-based evaluations, customers are sorted according to the value of their appetency score, and only the most appetent customers, i.e. those having the highest probability to buy the product, are contacted. In conjunction, market segmentation is conducted and marketing campaigns are proposed, tailored to the characteristics of each market segment. In practice due to constraints, such as time, subsequent segment analysis amounts to the analysis of the representative customer in the segment, generally the center of the cluster. This may not be helpful from an appetency point of view, since the appetency scores and the market segmentation efforts are not necessarily linked. Another problem that marketing campaigns face, is the instability of the market segments over time, when the campaign is redeployed over several months on the same campaign perimeter. To resolve the aforementioned problems this article proposes the construction of a typology by means of a partitioning method that is linked to the customers appetency scores. In essence, the authors elaborate a clustering method which preserves the nearness of customers having the same appetency scores. They have demonstrated the viability of their technique on real-world databases of 200,000 customers with about 1000 variables, from March, May and August of 2009 on a churn problem of an Orange product. In their demonstration, they have also evaluated the stability of their clusters over time and show that their clusters address the stability problem advantageously over other techniques.

The article 'Dimensionality Reduction using Graph Weighted Subspace Learning for Bankruptcy Prediction' by Ribeiro et al. (this volume) considers business-to-business relationships in the credit industry and, more specifically, the prediction of corporate financial distress. The importance of managing financial risk rigorously and reliably is well-known, not only but especially because of the financial crisis in 2008/2009, whose consequences still affect our daily life 5 years later. The objective of financial distress prediction is to estimate the probability that a company will become insolvent in the near future. Such forecasts play an important role in banks' risk management endeavors. For example, an insolvency prediction model helps bankers to decide on pending credit application. Moreover, estimating the likelihood that companies run into insolvency is a crucial task in managing the compound risk of credit portfolios. In this scope, Ribeiro et al. address an important modeling

challenge, the problem of high-dimensionality. Financial distress prediction data sets usually include a large number of variables related with various financial ratios and balance sheet information. To simplify the development of prediction models on such data sets and to enhance the accuracy of such models, Ribeiro et al. develop novel ways for dimensionality reduction using a graph embedding framework. Their approach shares some similarities with the well-known principal component analysis. However, it operates in a nonlinear manner and is able to take prior knowledge into account. This feature is a key advantage of the new approach because such knowledge is easily available in financial distress prediction. For example, the rules of business imply that some balance sheet figures must maintain a certain relationship with each other. A trivial example would be an enduring imbalance between assets and liabilities, which would, in the long run, threaten any company's financial health. Furthermore, the organizational acceptance of a data mining model depends critically on it being well-aligned with established business rules and it behaving in a way consistent with the analyst's expectations. The approach of Ribeiro et al. facilitate building data mining models that comply with these requirements and, in addition, enables an intuitive visualization of complex, high-dimensional data. Ribeiro et al. demonstrate these feature within an empirical case-study using data related with French companies.

3 Articles Focusing on Fraud Detection

Fraud detection has become a popular application domain for data mining. Insurance and credit card companies, telco providers, and network operators process an enormous amount of transactions and critically depend on intelligent tools to automatically screen such transactions for fraudulent behavior. Similar requirements arise in online setting and online advertisement in particular. This is the context of the article 'Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft' by Kitts et al. (this volume). Online advertisements are commonly purchased on the basis of a cost-per-click schema. Click-fraud is then a form of fraud where an attacker uses a bot network to generate artificial ad traffic. That is, a fraudster, either for his own financial advantage or to harm an advertiser/a competitor, uses the bots under his control to simulate surfers clicking on advertisements, which, unless detected, create costs on the advertiser's side. Kitts et al. provide an insightful discussion associated with the magnitude of click fraud, its severity and business implications, and the data mining challenges that arise in click-fraud detection. In addition, the article elaborates in much detail how Microsoft adCenter, the third largest provider of search advertising, has set up a sophisticated click-fraud detection system. Kitts et al. describe the specific components of the system, and how these components work together. The article is thus an invaluable resource to learn about state-of-the-art click-fraud detection technology and the data mining challenges that remain in the field.

Clearly, fraudulent behavior does not occur in the business world only. In their article "A Novel Approach for Analysis of 'Real World' Data: A Data Mining Engine for Identification of Multi-author Student Document Submission," Burn-Thornton et al. (this volume) investigate the potential of data mining to detect plagiarism in student submissions. Online courses, blended learning, and related developments have gained much popularity in recent years and have left their mark in higher education. Larger class sizes and, more generally, a less close student-tutor relationship are part of this development and have further increased the need for software tools that assist lecturers to mark exam papers from students who they may have never met in person. Many such tools are available. However, they are far from perfect, so that further research into automatic plagiarism detection is needed. Burn-Thornton et al. present an interesting approach based on student signatures. Such signatures are basically a summary of a student's specific style of writing. Through data mining student signatures from a database of exams, Burn-Thornton et al. are able to detect whether a document contains test passages that have been written by an author other than the submitting student. Concentrating on writing styles (i.e., signatures) allows Burn-Thornton et al. to move beyond standard text matching approaches toward detecting plagiarism. Consider for example a student who copies and rephrases text from some external source. Depending on the degree of rewriting, a conventional approach might fail to discover the rephrased text, whereas the signature of the rephrased text will in many cases still be different from the student's own signature. Empirical simulations indicate the viability of the proposed approach and suggest that it has much potential to complement conventional plagiarism detection tools.

A third article in the fraud-category is the article of Hsu et al. (this volume) on 'Data Mining Based Tax Audit Selection: A Case Study of a Pilot Project at the Minnesota Department of Revenue'. In their work they describe a data mining application that combines these two areas. They point out that the 'tax gap'—the gap between what people or organizations owe and what they pay—is significant and typically ranges between 16 and 20 % of the tax liability. The single largest factor for the tax gap is underreporting of tax. Audits are the primary mechanism for reducing the tax gap. In their article, the authors demonstrate that data mining can be an effective and efficient method for identifying accounts that should be audited. The data mining approach, which applies supervised learning to training data from actual field audits, is shown to have a higher return on investment than the traditional, labor intensive, expert-driven approach. In their pilot study, the authors show that the data mining approach leads to a 63.1 % improvement in audit efficiency. Thus, this article shows that data mining can lead to improved decision making strategies and can help reduce the tax gap while keeping audit costs low.

4 Articles Focusing on Data Mining in Medical Applications

Perhaps one of the most important application areas for data mining is the area of medical sciences. Clinical research in gene expression and other areas routinely involves working with large and very high-dimensional data sets. Hence, there is a dire need

for powerful data analysis tools. Similarly, there is a great need to find novel ways to offer and finance high-quality health services to an continuously aging population. This has led private and public health insurers to investigate the potential of data mining to improve services and cut costs (consider, for example, the recently finished Heritage Health Prize competition hosted by kaggle). These are just two examples that hint at the vast social importance of medical/healthcare data mining. Accordingly, the special issue considers two articles that deal with problems in this domain.

First, Gauthier et al. (this volume) report on 'A Nearest Neighbor Approach to Build a Readable Risk Score for Breast Cancer'. In many data mining applications, the primary goal is to maximize the ability to predict some outcome. But in some situations it is just as important to build a comprehensible model as it is to build an accurate one. This is the goal of Gauthier et al. (this volume), who build an assessment tool for breast cancer risk. Statistical models have shown good performance but have not been adopted because the models are not easily incorporated into the medical consultation. However, discussing similar cases can improve communication with the patient and thus the authors approach is to use a nearest neighbor algorithm to compute the risk scores for a variety of user profiles. In order to improve the usefulness of the models for patient discussion, domain experts were involved in the model construction process and in selecting the attributes for the model. All computation was done offline so that the risk score values for different profiles could be displayed instantly. This was done via a graphical user interface which showed the risk level as different traits were varied. The result was an easy to interpret risk score model for breast cancer prevention that performs competitively with existing logistical models.

The article 'Machine Learning for Medical Examination Report Processing,' is a second study on data mining for medical applications. Huang et al. (this volume) propose a novel system for name entity detection and classification of medical reports. Textual medical reports are available in great numbers and contain rich information concerning, e.g., the prevalence of diseases in geographical areas, the prescribed treatments, and their effectiveness. Such data could be useful in a variety of circumstances. Yet there are important ethical concerns that need to be addressed when employing sensitive medical information in a data mining context. With respect to the latter issue, Huang et al. develop machine learning algorithms for training an autonomous system that detects name entities in medical reports and encrypts them prior to any further processing of the documents. Furthermore, they develop a text mining solution to categorize medical documents into predefined groups. This helps physicians and other actors in the medical system to find relevant information for a case at hand in an easy and time-efficient manner. The name entity detection model consists of an automatic document segmentation process and a statistical reasoning process to accurately identify and classify name entities. The report classification module consists of a self-organizing-map-based machine learning system that produces group membership predictions for vector-space encoded medical documents. Huang et al. undertake a number of experiments to show that their approach achieves

higher precision and higher recall in name entity detection tasks compared to an state-of-the-art benchmark, and that it outperforms several alternative text categorization methods.

5 Articles Focusing on Data Mining in Engineering

From a general point of view, a common denominator among the above categories is that they all have a relatively long tradition in the data mining literature. Arguably, this is less true for applications in engineering, which have only recently received more attention in the field. Therefore, the special issue features five articles that illustrate the variety of opportunities to solve engineering problems using data mining.

In their contribution, 'Data Mining Vortex Cores Concurrent with Computational Fluid Dynamics Simulations', Mortensen et al. (this volume) elaborate the use of data mining in computational fluid dynamics (CFD) simulations. This is a fascinating new application area, well beyond what is typically encountered in the data mining literature. CFD simulations numerically solve the governing equations of fluid motion, such as ocean currents, ship hydrodynamics, gas turbines, or atmospheric turbulence. The amount of data processed and generated in CFD simulations is massive; even for data mining standards. Mortensen et al. discuss several possibilities how data mining methods can aid CFD simulation tasks, for example, when it comes to summarizing and interpreting the results of corresponding experiments. Next, they focus on one particular issue, the run of typical CFD simulation experiments and elaborate how they use data mining techniques to anticipate the key information resulting from complex CFP simulation long before the experiment is completed. To that end, they use simulation data produced in the early stages of an experiment and predict its final outcome using a combination of tailor-made feature extraction and standard data mining techniques. The potential of the approach is then demonstrated in a case study concerned with detecting vortex cores in well-established test cases.

Nayak et al. (this volume) consider the use of data mining within the scope of software engineering. The article 'A Data Mining Based Method for Discovery of Web Services and their Compositions' develops an approach for identifying and integrating a set of web services to fulfill the requirements of a specific user request. Web services are interoperable software components that play an important role in application integration and component-based software development. Albeit much progress in recent years, the identification of a web service that matches specific user requirements is an unsolved problem, especially if the web service consumer and supplier use different ontologies to describe the semantics of their request and offer, respectively. Therefore, Nayak et al. develop a data-mining-based approach to exploit semantic relationships among web services so as to enhance the precision of web service discovery. An important feature of their solution is the ability to link a set of interrelated web services. A common scenario in software development is that some required functionality cannot be supplied by a single web service. In such a case, the approach of Nayak et al. allows for aggregating a set of single

web services into a composite service, which provides the specified functionality. The proposed approach consists of three main components: (i) a semantic kernel to identify semantically similar web services for a service consumer, (ii) a composition algorithm that first models semantically similar web services as nodes of a graph and then selects the best option for invoking multiple services according to an all-pair shortest-path algorithm, and (iii) a fusion algorithm that creates a composite service through merging the results of the other two modules. Empirical experiments on real-world data evidence the effectiveness of the proposed methodology and demonstrate that the proposed system is well-prepared to recommend multiple inter-related web services that match the consumer's requirements if a single services fails to do so.

In their article 'Exploiting Terrain Information for Enhancing Fuel Economy of Cruising Vehicles by Supervised Training of Recurrent Neural Optimizers,' Abou-Nasr et al. (this volume) show how a data-driven approach can be used to solve an engineering optimization problem. Their goal is to build a smart cruise control, which modifies the automobile's speed in order to maximize fuel economy, while generally averaging the cruise control speed set by the driver. They describe how supervised training of recurrent neural networks can approximate the solution of a deterministic, discrete, dynamic programming problem, to determine a good policy of control decisions. The learned policy considers the current vehicle speed and road grade, as well as past history of vehicle speeds and road grades. Simulation results demonstrated that over three road segments the learned policy yielded an increase in fuel economy of about 9 % when compared to the strategy of maintaining the fixed speed.

Cheung et al. (this volume) develop a holistic approach to enhance aircraft safety management. More specifically, their manuscript 'Exploration of Flight State and Control System Parameters for Prediction of Helicopter Loads via Gamma Test and Machine Learning Techniques' concentrates on helicopters and estimate the load of critical components during flight operations, which, in turn, helps to determine whether such components remain fully-functional or require overhaul/replacement. The article combines an exciting novel application field for data mining techniques with classic requirements in predictive modeling. One the one hand, an accurate solution for the forecasting problem at hand (i.e., component load estimation) is needed. On the other hand, to meet the requirements of safety engineers and other stakeholders, the prediction model is also required to provide detailed insight as to which input features (e.g., sensor data dynamically collected during flight operations, control system parameters, etc.) are most correlated with component load. The identification of such causal drivers is indeed pivotal to better understand which flight state parameters are most relevant for specific loads in airframe and dynamic components of a helicopter. Cheung et al. address the two dimensions of their problem (prediction and structural process understanding) through integrating several analytic techniques such as principal component analysis, multi-objective optimization, and artificial neural networks into a fully-functional framework for estimating component load and retirement, respectively.

Finally, in their work on 'Multilayer Semantic Analysis In Image Databases', Sayad et al. (this volume) propose a higher-level image representation, semantically

significant visual glossary, in order to retrieve and classify images beyond their visual appearances. They first introduce a new multilayer semantic significance model in order to select semantically significant visual words (SSVWs) from the classical visual words according to their probability distributions relating to the relevant visual latent topics in order to overcome the rudeness of the feature quantization process. Then they exploit the spatial co-occurrence information of SSVWs and their semantic coherency in order to generate a more distinctive visual configuration, i.e., semantically significant visual phrases. Finally, they combine the two representation methods to form SSIVG representation. Through experimental studies, they demonstrate the good performance of their approach compared with several approaches in retrieval, classification, and object recognition.

Part I
Established Data Mining Tasks

What Data Scientists Can Learn from History

Aaron Lai

Abstract We argue that technological advances and globalization are driving a paradigm shift in data analysis. Data scientists add value by properly formulating a problem. A deep understanding of the context of a problem is necessary because our incomplete answer will be worse than incorrect—it is misleading. Therefore, we propose three innovative analytical tools that define the problem in a solvable way: institution, data, and strategy. Afterward, we use three historical examples to illustrate this point and ask "What would a 'typical' data scientist do?" Finally, we present the actual solutions and their business implications, as well as data mining techniques we could have used to tackle those problems.

1 Introduction

Benjamin Disraeli said "What we anticipate seldom occurs; what we least expected generally happens." As the volume of data grows exponentially, quantitative analysis, statistical modeling, and data mining are becoming more important. Predictive modeling is the use of statistical or mathematical techniques to predict the future behavior of a target group. It is different from forecasting in that forecasting uses time-series data to forecast the future. Predictive models are independent of time[1] so it will only be affected by random factors. Predictive modeling assumes that people, as a group, will behave in the same way given the same situation. The variations or errors are caused by an individual's unobserved characteristics.

Part of the material of this article is based on my presentation titled "Predictive Innovation or Innovative Prediction?" for the Predictive Analytics Summit held in San Francisco in November 2010. Only the Powerpoint version was distributed to the participants. This paper has not been submitted to any other places. All opinions are my own personal views only and do not necessarily reflect those of my employer or my affiliation.

[1] In technical terms, they are called stationary.

A. Lai (✉)
Market Analytics, Blue Shield of California,
San Francisco, CA, USA
e-mail: aaron.lai@st-hughs.oxon.org

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_2

Of course, prediction is not the only thing a data scientist will do. Data science, a new yet undefined term, is to make sense out of data. It could be statistical analysis, algorithmic modeling, or data visualization. High volumes of data, which is commonly known as Big Data, require a new approach in problem solving. To succeed, we need an innovative approach to data analysis.

In this article, we argue that model building processes will be changed due to technological advances and globalization of talents. We analysts add value by a creative adaptation of modeling and an innovative use of modeling. It is the survival of the fittest and not the survival of the strongest!

As Louis Pasteur said centuries ago, "Chance favors prepared minds." Predictive methods, when used properly and innovatively, could result in sparkling outcomes. Competitive pressure will make it just too important to leave it to non-professionals. It is very common for a half-knowing analyst to jump into the labyrinth of modern tools without thinking. It is thus essential to be innovative.

We look at the model building process from three angles: Institution, Data, and Strategy. We will use three historical examples to illustrate this point by asking, "What would a 'typical' data scientist do?" It is not uncommon for an inexperienced analyst to blindly apply what he or she learned from the textbooks irrespective of the root cause. We will contrast our "default" answers to the ingenious historical solutions. In describing the aftermath of the Long-term Capital Management fiasco, Niall Ferguson wrote "To put it bluntly, the Nobel prize winners had known plenty of mathematics, but not enough history. They had understood the beautiful theory of Planet Finance, but overlooked the messy past of Planet Earth. And that, put very simply, was why Long-Term Capital Management ended up being Short-Term Capital Mismanagement".[4, p. 329] Andrew Lo of MIT used another "P envy" [16] that echoed Ferguson's comment as it was titled "WARNING: Physics Envy May Be Hazardous To Your Wealth!"

The model development cycle is being compressed at an unprecedented speed. This is due to three factors: technological advance, outsourcing, and innovation diffusion. The latest statistical or data-mining software can easily replace a team of analysts. For example, SAS has a fully automated forecasting system that can create and fit a series of ARIMA models; Tableau analyzes data and suggests what type of chart would be most appropriate. Since we live in a global village, if I can formulate the problem in an equation or write down a specification, I can recruit an expert across the world to solve it. There are many sites or companies that allow people to pose questions and source answers. Crowdsourcing makes geographical limitation irrelevant. The last factor is an escalating pace of innovation as news travel fast—the latest techniques could be instantly imitated.

2 Case I: The War Chest

This was 1694 England. The Crown was under severe financial pressure and no easy answer was in sight.

2.1 Background

The main source of income of William the Conqueror since 1066 was the possession of royal properties (Royal Demesne) and the feudal system of land tenure (Feudal Aids). Feudal Aids was the right for the King to levy a tax for his ransom should he be taken prisoner by an enemy (thus we have the term the King's Ransom). This land tax system had been abused by the Crown so much that the nobles needed to create the Magna Carta to protect the lender's right. Customs were invented in 1643 after adopting the Holland system of excise taxes. The first record of currency debasement in England (decreasing the amount of precious metals and thus lowering the value of the coins) is from the reign of Edward I in 1300. There were many subsequent debasements. The metal content of the same coin dropped to only one-seventh from the beginning to the end of the reign of Henry VIII!

Henry III had the first recorded debt. Since interest payment was forbidden (usury), the Crown only needed to pay back the principal in those early days. During the Hundred Years War (1337–1453), Henry V had incurred so much debt that he would need to secure his debts by securities such as tax and jewels in 1421. In the twentieth century, those securities were called revenue bonds and asset-backed securities. Henry VIII defaulted on his loans several times by releasing himself from repaying those borrowed monies while Elizabeth I had excellent credit (could borrow at 10 % interest from Antwerp) and she finally paid all her loans.[5, p. 61, 67, 70, 72–74]

The financial situation was indeed very challenging in 1690s. William of Orange arrived in England in 1688 and England was at war with France in 1689 for the Nine Years War (1689–1697). The credit of the Crown remained weak until the Glorious Revolution of 1688 institutionalized the financial supremacy of the Parliament. The Parliament controlled new taxes and limited the power of the King. The whole system changed from the King to the King in Parliament and thus it established the financial superiority of the Parliament. One of the financial revolutions was to make notes transferable [19]. The governmental expenditure increased from £ 0.5 million in 1618 to £ 6.2 million in 1695 while debt increased from £ 0.8 million in 1618 to £ 8.4 million in 1695 [19]!

2.2 Problem Statement

Governmental debt was increasing at an astonishingly high rate. Even after some costly wars in continental Europe, there were no signs that any kind of peace would come soon. The King and Country needed a lot of money to finance military build-up and prepare for the next war.[2] The Crown had recovered his credit standing and thus was able to borrow more. In 1693, there was a large long-term loan (£ 1 million) secured by new taxes but it was almost immediately exhausted by 1694 [19].

[2] In fact the War of the Spanish Succession (1702–1713) was just around the corner.

The creditors were growing uneasy about the debt level and they demanded interest rate as high as 14 % in 1693 and 1694 [19]. Since those debts were "asset-backed securities"[3], the HM Treasury officials had already used up high quality assets to do credit-enhancement.

2.3 What if We Were There?

Government revenue comes from two sources: tax and borrowing. Following a standard modeling approach, we could create an econometric model to investigate the elasticity of taxation. We could also use a segmentation model to put citizens/institutions into buckets, since they all had different coefficient of elasticity. A tax maximization policy would tax the most tax inelastic groups, subject to their ability to pay. It would be a typical constrained optimization exercise.

On the borrowing side, we would have to estimate the borrowing capability for our sovereign debts. We might run some macroeconomic models to assess our financial strength so as to present a credible plan to convince the market of our credit worthiness. There are only three ways a country can handle her debt: grow out of it, inflate over it, or default on it. Of course the investors hate the last two options. Thus it is the job of the Chancellor of Exchequer to make a convincing case.[4] This is also why the central banks need to be considered as independent so that their will to fight inflation is strong.

2.4 The Endgame

Two important innovations helped drive down the borrowing cost and increase the borrowing capability. The first was the invention of fractional reserve by goldsmith-bankers and the second one was the incorporation of the Bank of England. During the medieval time, people stored gold and other valuables in the vault protected by the goldsmiths. The depositor received a certificate that could be redeemed on demand. Since only the goldsmiths knew the exact amount in a vault, they found that they could lend money (by issuing certificates, just like the Certificates of Deposit we have now) without doing anything [1]. The goldsmiths could then lend a substantial amount of money to both the Crown and the public. They also used reserve ratio and loan diversification to manage risk; operation risk for the former one and credit risk for the latter one. In the case of Sir Francis Child, he maintained 50–60 % reserve-to-asset ratio and diversified his lending to the general public and various Crown debts backed by different revenue stream such as Customs, Excise, East India Goods, Wine

[3] They were backed by additional excise and duties on imports respectively.
[4] For an explanation on the history of the Bank of England could help understanding the eurozone crisis, refer to [13].

and Vinegar etc. The increasing use of discounting (delay payments in exchange of a fee) by bankers like Sir Francis facilitated the circulation and liquidity of long-term debts. Discounting also allowed them to shorten the term structure of their liabilities [21].

Given the insights of using fractional reserve to increase the loan (i.e. money) supply and using high quality assets to enhance investment attractiveness, we could reformulate this problem into a portfolio optimization exercise. Following the standard mean-variance approach pioneered by Markowitz, we could create efficient portfolio of assets based on risk and return, as well as the inter-asset covariance. Many optimization algorithms could help solve this problem and a classical solution is quadratic programming. An alternative approach to optimization is econometrics modeling. We could use discrete choice analysis to find out who is going to buy what type of asset. In addition, Monte Carlo simulation and Agent-based Modeling (ABM) could also be employed. This kind of approach would allow us to model the dynamic interactions and inter-agent interactions in various consumption and preference trade-offs.

In modeling a solution, we need to be aware of the principal-agency problem as perceived by the investors. The HM Treasury served at the pleasure of the King and it was not there to serve the investing public. Therefore, any solution needed to be a credible solution from the point-of-view of the investors; they needed to be reassured that the government was determined to repay her debt. People said, "It is not about the return of money; it is about the return of my money."

The subscribers of government debts were invited to incorporate as the Bank of England in 1694. The Bank was responsible for handling the loans and the promised distributions. One of the most important characteristics was that the Bank could not lend the Crown money or purchase any Crown lands without the explicit consent of the Parliament [19]. To further lower the risk of the lenders, the government created a separate fund to make up deficiencies in the event that the revenue earmarked for specific loans was insufficient to cover the required distribution [19].

Government needs money and wars need a lot of money. The ability to borrow a large amount of long-term money cheaply was the reason that Britain beat France and emerged as a major power of the world [19]. Finance was so important that the Prime Minister was also the Chancellor of the Exchequer until the eighteenth century. The modern Chancellor of the Exchequer is always the Second Lord of the Treasury (No. 11 Downing Street) while the Prime Minister is still the First Lord of the Treasury. The official sign is still nailed to the front door of No. 10 Downing Street. These two innovations fundamentally changed the financing ability of Britain and that led to centuries of British Empire, especially for the funding of an expensive Royal Navy. The Bank of England became so prominent that it even had a nickname "The Old Lady" since 1797. Institution arrangement is very important to economic development, and Douglass North received his Nobel Prize because of his contribution to this area [25, p. 21] (Fig. 1).

Given the incomplete nature of old data, it would be difficult for us to assess the situation via quantitative analysis. However, researchers have [25] built a VAR (Vector Autoregressive) model to study the dynamics of the determination of interest rate on government debt from 1690 to 1790. They found that industrial revolution,

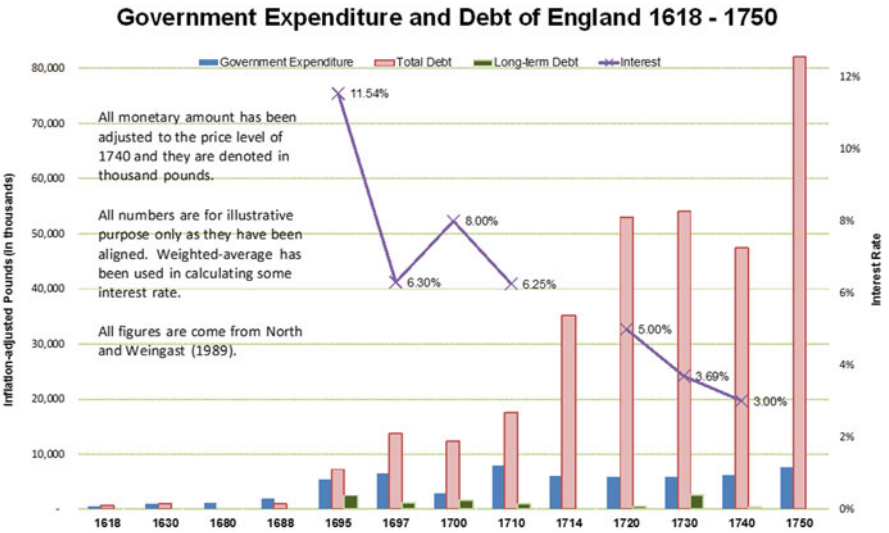


Fig. 1 Debt of England based on the figures listed in [19, pp. 820, 822, and 824]

military victories, and institutional reforms contributed a lot, especially the flight of capital from Napoleon's reign.

2.5 Business Implication

GMAC was an example of an institutional innovation. It was originally created as a wholly owned subsidiary of General Motors to provide financing support to GM dealers. With this new institution, GM could offer incentive car loans to customers or dealers with very low interest rates. The increased sales further lowered the production cost (average fixed cost from the economies of scale) of a car. This kind of institutional arrangement has become a standard practice in the automotive industry. Now all major car manufacturers have subsidiaries to do automobile financing. The same idea has been extended to private label credit cards and other manufacturer financing. Data mining could help determine the optimal asset allocations for both the parent and the spin-off. Financial engineering can also decide the best capital structure and borrowing level.

3 Case II: London Outbreak

This was an ordinary August day (24th) in 1854. Mrs. Lewis of 400 Broad Street was washing her baby's diaper in water, and she subsequently emptied the water into

a cesspool in front of the house. Little did she know that this simple action would cause 700 deaths within a 250-yard radius of a nearby water pump since her baby was infested with cholera [18].

3.1 Background

England was in a state of panic as there were over 20,000 deaths in England and Wales in 1853–1854. Asiatic cholera reached Great Britain in October 1831 and the first death occurring in that month was at Sunderland [7]. Cholera was first found in 1817. It caused 10,000 deaths out of a population of 440,000 in St. Petersburg in August 1831.[5] Even though it had been researched extensively in a previous India outbreak[6], no one really knew much about the disease and the Russians had even offered a prize for the best essay on *cholera morbus*. Miasma (spread via air) was the prevailing theory of transmission for the greater part of the nineteenth century. The irony was that even though sanitarians' casual theory was incorrect, they were able to demonstrate how and where to conduct the search for causes in terms of the clustering of morbidity and mortality. Jakob Henle argued in 1840 that cholera was caused by minute organism, and John Snow's works in 1849 to 1854 were consistent with this theory. Unfortunately, nothing until Louis Pasteur's experiment in 1865 could the establishment accept infectious disease epidemiology [24]. Snow questioned the quality of water, and after performing some microscopic works, he was not able to find the cholera micro-organisms [9, p. 99].

3.2 What if We Were There?

Snow was a very analytical person and is one of the pioneers of analytical epidemiology. William Farr, an established epidemiologist at that time, realized that the "Bills of Mortality" would be much more amenable to analysis when they contained variables in addition to names and parishes. His reports published in mid-1840s counted deaths not only by 27 different types of disease, but also by parish, age, and occupation. Snow used Farr's data to investigate the correlations among them.

If we were there, we could develop some logistic models with all variables to see if we could support or refute the prevalent theories[7]. However, we would have difficulties in developing a comprehensive model because we could not directly test both the contagion and the miasmatic hypotheses. And according to sanitarians, organic matters were not the direct causes of disease themselves, but as raw materials

[5] p. 1, 16 [8].
[6] Just the Madras volume ran to over 700 pages, p. 30 and 31 [8].
[7] In fact, a paper used a logistic model on the Farr data and it rejected the Farr theory that cholera was caused by elevation [2].

District	Deaths from cholera in 1849 per 10,000 inhabitants	Elevation above high water (feet)	Annual deaths from all causes 1838–1844 per 10,000 inhabitants	Persons per acre	Persons per inhabited house	Average annual value of house (£)	Annual value of house per person (£)	Poor rate precept per pound of house value	Water supply[a]
Newington	144	−2	232	101	5.8	22	3.788	0.075	1
Rotherhithe	205	0	277	19	5.8	23	4.238	0.143	1
Bermondsey	161	0	264	66	6.2	18	3.077	0.134	1
St George	164	0	267	181	7.0	22	3.318	0.089	1

Fig. 2 Eight possible explanatory variables [2, p. 389]

Explanatory variable	Low 95% CL	Odds ratio	High 95% CL	*P*
Constant	6.006×10^{-4}	0.002626	0.01149	-
Water from Thames between Battersea Bridge and Waterloo Bridge[a]		1.00		<0.001
Water from New River and Rivers Lea and Ravensbourne	0.44	0.59	0.79	
Water from Thames between Kew and Hammersmith	0.22	0.40	0.72	
Increase in elevation above high water (10 feet)	0.85	0.91	0.98	<0.01
Decrease in poor rate (£/100)	0.87	0.91	0.96	<0.001
Average annual death rate 1838–1844	1.00	1.00	1.01	0.48
Persons per inhabited house	0.89	1.03	1.19	0.71
Persons per acre	1.00	1.00	1.00	0.67
Average house value per person (£)	1.00	1.00	1.00	0.35
Average house value within district (£)	1.00	1.00	1.00	0.79

[a] Baseline.

Fig. 3 Logistic regression results [2, p. 392]

to be operated upon by disease "ferments" presented in the atmosphere during epidemics [20]. The significance results from miasmatic research at that time could be caused by the spurious correlation problem. Spurious correlation is the appearance of correlation caused by unseen factors.

Figures 2, 3 and 4 provide some tables and results from [2]. This model shows that poverty is the most significant factor!

3.3 The Endgame

Dr. Snow marked each death on the map as an individual event[8] rather than a location of death. He did find that all deaths were within a short walking distance from the pump. Secondly, he made another map to show that those deaths were indeed closer to the Broad Street pump than the others [10]. Thirdly, he obtained water samples from several pumps in the area but the Broad Street water looked cleanest. Furthermore, he had two "negative data" points that supported his case: no deaths in the Lion Brewery (workers drank the beer) and the workhouse (which had its own well) [18].

[8] Many people, including Edward Tufte and the CDC, took E.W. Gilbert's version of map (with dots instead of bars) as John Snow's original maps.

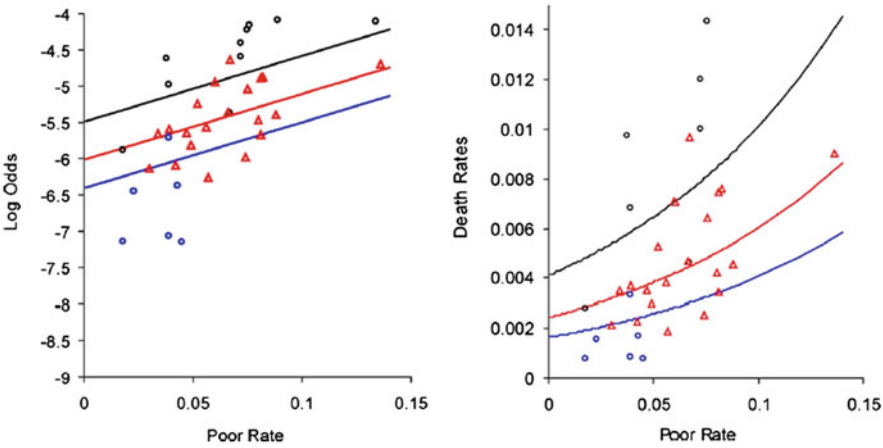


Fig. 4 Odds diagrams [2, p. 392]

The success of Snow's hypothesis rested on its narrow focus, while the Board of Health had a general hypothesis only. Snow's predictions were so specific that only a few observations were contradictory. Snow personally investigated on-site for those contradictory observations (e.g. brewery and workhouse) until he was satisfied with them. His hypothesis was also consistent with clinical observation. Snow insisted that the disease was gastrointestinal and all symptoms could be explained by fluid loss from the gastrointestinal tract. This led him to conclude that the infecting agent was oral and not respiratory [18, 20].

This is an important point for data scientists because our results or conclusions have to be consistent with all other information, both within and outside our model. The results need to be not only statistically and logically sound, but also must be consistent with observation. If you find something that contradicts to common sense, it is more likely for you to have made a mistake than to have discovered a new world.

Henry Whitehead did a survey to try to refute the conclusion of Snow. However, his results were in fact confirmed Snow's analysis. For those who drank water from the Broad Street pump, 58 % developed cholera compared with only 7 % of those who did not. Snow found that the mortality was related to the number of people who drunk from the pump during the infested period (from the date of washing the infested diaper to the removal of the pump handle). Another engineering survey[9] concluded that there had been a consistent leak from the cesspool to the pump shaft [18]. For a more detailed discussion on the contribution of John Snow to analytical epidemiology, see [14].

[9] The Board opened up the brick shaft but it seemed perfectly in order.

3.4 Business Implications

Google Maps has opened many possibilities of marrying data and geographical information. People create map-based websites ranging from restaurant guides to Haiti disaster relief.[10] It is impossible to underestimate the impact of seeing information displayed on a map! This is the power of Data Innovation—collecting, using, and displaying data in innovative ways.

A recent BBC report[11] showed that Google, Microsoft, and Apple were all eyeing the rapidly growing spatial information market. We predict that spatial analysis and data visualization will gain lots of momentum when our infrastructure could support collecting, storing, and analyzing vast amount of data everywhere anytime.

Nevertheless, data visualization provides hints to the solution but cannot be the solution itself. Given almost identical information (even similar maps), the Board and Snow arrived at completely different conclusions. Why? It was because the Board analyzed the situation through a conventional len. They were all distinguished scholars or practitioners, and they fitted the facts into the model rather than retrofitting the model for the facts. The success of Snow rested on his particular attention to anomalous cases [10]. It is very common for us to downplay the importance of outliers rather than drilling down to the root cause of those "unfitted" observations. We tend to blame the customers for not behaving as our model predicted and not acknowledging it as a limitation of the model. The same rationale can be extended to financial model development as well [15].

When we perform spatial analysis and data visualization, we need to be careful that we are convincing rather confusing our audience. As explained in a New York Times article, an Army platoon leader in the Iraq war could spend most of his time making PowerPoint slides [3].

4 Case III: A Tale of Two Navies

This was 1904. Russia under the Tsar was an established European power with high self-image while Japan was a rising industrial power in Asia after victory in the Sino-Japanese War (1894–1895).

4.1 Background

Russians did not think highly of the Japanese navy because 50 years earlier Japan had no fleet at all. The Russian Foreign Minister, when asked about the possibility of

[10] Dan Mascai wrote in January 14, 2010 for Fast Company titled, Haiti Earthquake Disaster: Google Earth, Online-Map Makers, Texts "Absolutely Crucial" http://www.fastcompany.com/blog/dan-macsai/popwise/haiti-earthquake-google-maps-web-tech.

[11] "Tech giants compete over mapping" from BBC Click, August 10, 2012.

war with the Japanese, replied "One flag and one sentry: Russian prestige will do the rest". Japan had a close business relationship with Britain: Vice Admiral Togo was trained in Britain with the Royal Navy, many battleships were built by the British, and Japan was also largely dependent on Britain for guns, ammunition, and coal. Togo was an accomplished student of Admiral Mahan of the United States Navy and Admiral Markarov of the Imperial Russian Navy. All of Togo's battleships were less than 10 years old and had similar speeds, turning circles, and optimum gun ranges [11, p. 123]. These factors played a strong role in their innovative strategies.

Russia had three fleets: the Baltic, the Black Sea, and the Pacific Fleets. Russia was poorly situated in fighting a war in the Pacific given the geographic distance (15,000 miles away) between the Baltic and the Pacific Fleets and also the immobility of the Black Sea Fleets [23] to enter the Russo-Japanese War.[12] The Japanese realized that they needed to attack Port Arthur (in the Yellow Sea) because Russia would have half-dozen new battleships within one year.

4.2 Problem Statement

The problem of Japanese navy was that they had to win a quick and decisive battle in Port Arthur because they could not afford a resource-intensive long war. Pacific Fleets outnumbered Japanese fleets and the Russians had more supplies despite long distance.

4.3 What if We Were There?

If we were navy planners, we could construct some game-theoretic models to analyze the movement of battleships and determine the optimal interactions.[13] Supply-chain optimization programs could also be used to plan for the logistics. Forecasting models might be built to predict the scenarios of Japanese attack. Large scale simulation could also be used to incorporate information as diverse as morale and weather forecast.

4.4 The Endgame

The Russian Navy was ill-prepared for a naval warfare in the Pacific even though they had the newest and best ships and were larger than the entire combined Japanese Fleet. Russia was destined to lose the war due to her institutional nature—the Tsar had

[12] It was due to the treaty with Turkey; [11, p. 123] and see also p. 29, 122, 123, and 127.
[13] In fact, the submarine search problem was one of the first applications of game theory [22].

absolute power and few talents but fancied himself an expert in Asian affairs [23]. A distinctive aspect of the newly built Japanese battle group was a balanced approach instead of maximizing individual firepower. The Japanese Navy had adopted the following innovative strategies which were counter to conventional wisdom [6]:

- They used thinner, impact-detonating shells instead of thick, armor-piercing shells to damage vital above-deck components for maximum damage.
- Japanese tacticians took the T tactic one-step further and add the L tactic because their ships were faster and more maneuverable. This tactic allowed the Japanese ships to encircle the enemy and prevent them from escaping.

In addition to those innovative strategies, relentless training and exercise imposed by Togo were also critical to their success. Prior to the war with Russia, Togo took the Fleet to the area where he predicted battle would occur and rigorously trained all components [12]. The Japanese Navy was able to work as a team and their guns were far more accurate than their Russian counterpart [6].

When the battle was concluded, the Russian fleet had been almost completely destroyed. Togo captured or destroyed 31 of the 38 Russian ships while losing none of his own; Japan lost 117 men while they capturing 6000 and killing 5000 Russians [6]. The results of this battle rippled throughout the whole twentieth century: it caused a severe blow to the Romanov dynasty that led to October Revolution; it boosted the confidence of Japanese military that led her to the Second World War.

4.5 Business Implication

The overall design of Apple's iPod was enchanting even though it did not have any technological breakthrough; every piece of technology of the original iPod was proven and well established. However, Apple did a great job in integrating and executing their integrated strategy. It is a spectacular case of Strategy Innovation; Apple had done nothing path-breaking but they were able to capture a key strategic insight—simplicity and convenience. As stated in a CNET review, Apple was known for "an innovative and free-thinking approach to product design."[14] Another innovation of Apple was the changes in legal music download: it made downloading songs cheap and easy. It changed how music was delivered forever. As of 2008, three out of four digital music players sold in the U.S. were iPod or its variations [17].

Strategy Innovation is not simply building a better mousetrap—it is using a new way to build a mousetrap or even find something to replace the need of a mousetrap.

On the other hand, Google aggressively tests their products to find out what element would work under what circumstances. Statistical techniques such as conjoint analysis or experimental design could help. In the data mining arena, genetic algorithms could be used to morph winners into a winner.

[14] CNET review on Apple Computer iPod dated 10/24/01.

Market research provides valuable insights into the mind of the consumers. Nevertheless, it is the analysts and the business executives who decide when to listen, what to listen to, and when to ignore the consumer insights. If Sony had followed market research, they would not have developed the Walkman.

The latest crowdsourcing projects such as the Netflix recommender system competition have pushed the power of human and machine to the extreme. The winning teams all used ensemble techniques to blend various algorithms together to get the best possible outcome.

5 Lesson Learned

Just like a madman shouting "model building is dead", we argue that model building will gain prominent importance due to technological advance (higher computational power and smarter algorithms), outsourcing (people well versed in unbelievably strong technical skills), and innovation diffusion (good techniques will be imitated).[15] The law of diminishing marginal return will push down the marginal value of old models. We will rely more on quantitative modeling when we have bigger and bigger data. We data scientists add values by offering insights on what a model should look like, defining the functionalities of a model, and developing innovative uses for a model.

Knowing the technique is not enough; knowing how to use it is more important. There are three types of innovative analytical approaches that pave the road to success: Institution, Data, and Strategy. Competitive pressure will constantly push us to the frontier of analytics. Innovations come from challenging the conventional wisdoms: the Bank of England limited the power of the Crown which in turn provided assurance to the borrowers; Dr. John Snow questioned the mainstream epidemiology theory and pushed himself to the opposite side of the whole establishment; Togo turned a "T" into a "L" and backed his strategies up with intensive training. Of course they could all benefit from modeling if it were at their disposal. But it is their insights, hardworking, and endurance that are the currencies of success.

We believed that data scientists could add tremendous value in providing insights before building a model and offering innovative use once a model is built. The analytics business is constantly under competitive pressure from technological advance, outsourcing, and innovation diffusion. To avoid being an irrelevant artifact, we could leverage three types of innovation: institution, data, and strategy. Our idea can be illustrated in the Figure 5.

[15] As Oscar Wilde said, "Imitation is the sincerest form of flattery."

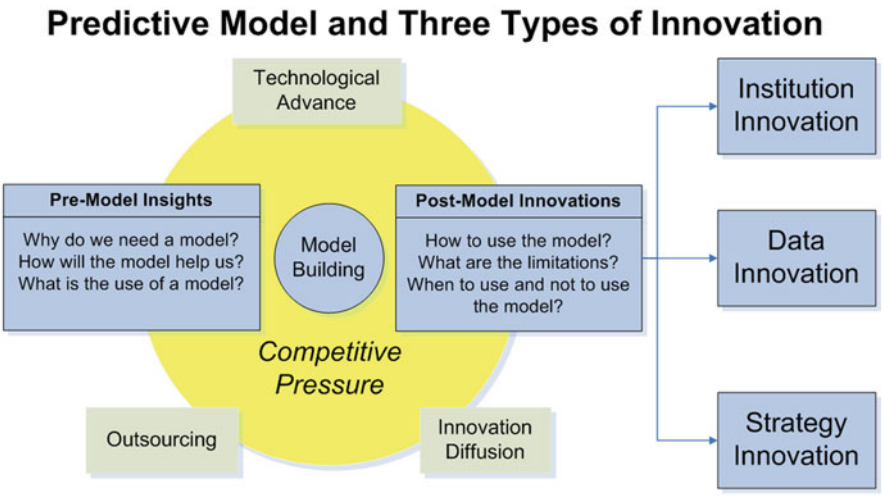


Fig. 5 Predictive model and innovation

5.1 Epilogue

We illustrated the use and misuse of quantitative method with three historical examples. The case of the Bank of England is an Institution Innovation. The invention of fractional reserve by goldsmith-bankers and the incorporation of the Bank of England enabled the Crown to borrow a lot more money with a much lower cost of capital. The case of "Broad Street Pump Maps" by Dr. John Snow on London Cholera brilliantly showed the power of Data Innovation. The juxtaposition of geographical information (data visualization) and a meticulous approach to hypothesis testing saved hundreds of lives and succeeded where several official investigations failed. In the Battle of Tsushima, the Japanese Navy won a decisive battle with Strategy Innovation. The improvement of gunnery accuracy and the "L" strategy played a critical role. We argued that dogmatically applying quantitative models only results in mediocre outcomes. We often take false comfort by the "scientific" nature of modeling when a game-changing approach may be just around the corner. Quantitative analysis is very important. But the most challenging part is to formulate the problem in a solvable way. The actual mechanics is much simpler.

If we look back at the "modern" history (defined as after the fall of the Roman Empire), we had "physical philosophy" when Issac Newton did his great work; we had "moral philosophy", and "natural philosophy" when Adam Smith wrote *The Wealth of Nations*. Afterward, we had "social philosophy" during the Enlightenment period by Voltaire and others. Industrialization brought us "political economy" from people such as David Ricardo. Unfortunately, "economic science" has become the only path and we have forgotten the moral, social, and political aspect of economics and analysis.

References

1. Bernstein, W.: Perspectives: Of laws, lending, and limbic systems. Financ. Analysts J. **66**(1):17–22 (2010)
2. Bingham, P., Verlander, N., Cheal, M.: John Snow, William Farr and the 1849 outbreak of cholera that affected london: A reworking of the data highlights the importance of the water supply. Public Health **118**(6):387–394 (2004)
3. Bumiller, E.: We have met the enemy and he is powerpoint. New York Times **26** (2010)
4. Ferguson, N.: The Ascent of Money: A Financial History of the World. Penguin Press HC, New York (2008)
5. Fisk, H.: English Public Finance from the Revolution of 1688. Bankers Trust Company, New York (1920)
6. Fleet, A.: Japanese naval transformation and the battle of Tsushima. Military Review 84, **73**(6) (2004)
7. Gilbert, E.: Pioneer maps of health and disease in England. Geogr. J. 124(2):172–183 (1958)
8. Hempel, S.: The Strange Case of the Broad Street Pump: John Snow and the Mystery of Cholera. University of California Press, Oakland (2007)
9. Johnson, S.: The Ghost Map: The Story of London's Most Terrifying Epidemic—and How it Changed Science, Cities, and the Modern World. Riverhead Books (Hardcover), Penguin Books Ltd. London, UK (2006)
10. Koch, T.: The map as intent: Variations on the theme of john snow. Cartogr. Int. J. Geogr. Inf. Geovis. **39**(4):1–14 (2004)
11. Koenig, W., Mayer, S.: Epic Sea Battles. Chartwell Books Edison, NJ, USA (1976)
12. Kornatz, S.: The operational leadership of admiral togo. Technical Report, DTIC document (1995)
13. Lai, A.: Does the eurozone need an "old lady"? CFA Mag. **21**(4):16–17 (2010)
14. Lai, A.: London cholera and the blind-spot of an epidemiology theory. Significance **8**(2):82–85 (2011)
15. Lai, A.: Paradigm lost. CFA Mag. **22**(4):11–12 (2011)
16. Lo, A., Mueller, M.: Warning: Physics envy may be hazardous to your wealth! available at SSRN 1563882 (2010)
17. Long, T.: Now hear this ... the ipod arrives. Wired Mag. (2008 Oct 23)
18. Newsom, S.: Pioneers in infection control: John snow, henry whitehead, the broad street pump, and the beginnings of geographical epidemiology. J. Hosp. Infect. **64**(3):210–216 (2006)
19. North, D., Weingast, B.: Constitutions and commitment: The evolution of institutions governing public choice in seventeenth-century england. J. Econ. Hist. **49**(04):803–832 (1989)
20. Paneth, N., Vinten-Johansen, P., Brody, H., Rip, M.: A rivalry of foulness: Official and unofficial investigations of the london cholera epidemic of 1854. Am. J. Public Health **88**(10):1545–1553 (1998)
21. Quinn, S.: Tallies or reserves? sir francis child's balance between capital reserves and extending credit to the crown, 1685–1695. Bus. Econ. Hist. **23**(1):39–51 (1994)
22. Shubik, M.: Economics, management science, and operations research. Rev. Econ. Stat. **40**(3):214–220 (1958)
23. Sprance, W.: The Russo-Japanese war: The emergence of japanese imperial power. J. Mil. Strateg Stud. **6**(3):1–24 (2004)
24. Susser, M., Susser, E.: Choosing a future for epidemiology: I. eras and paradigms. Am. J. Public Health **86**(5):668–673 (1996)
25. Sussman, N., Yafeh, Y.: Institutional reforms, financial development and sovereign debt: Britain 1690–1790. J. Econ. Hist. **66**(4):906 (2006)

On Line Mining of Cyclic Association Rules From Parallel Dimension Hierarchies

Eya Ben Ahmed, Ahlem Nabli and Faïez Gargouri

Abstract The decision-making process can be supported by many pioneering technologies such as Data Warehouse (DW), On-Line Analytical Processing (OLAP), and Data Mining (DM). Much research found in literature is aimed at integrating these popular research topics. In this chapter, we focus on discovering cyclic patterns from advanced multi-dimensional context, specially parallel hierarchies where more than one hierarchy is associated to given dimension in respect to several analytical purposes. Thus, we introduce a new framework for cyclic association rules mining from multiple hierarchies. To exemplify our proposal, an illustrative example is provided throughout the article. Finally, we perform intensive experiments on synthetic and real data to emphasize the interest of our approach.

1 Introduction and Motivations

Data warehouses are broadly spread over companies. They incorporate relevant information that can easily be analyzed and visualized using OLAP tools. Still, it is challenging to offer the tools for analysts to automatically mine suitable knowledge from such DW repositories.

At the junction of the OLAP technology and the association rules, multidimensional correlations are extracted from data cubes. The intelligent derived patterns take advantage of the multidimensional modeling, *i.e.,* investigated measures, analyzed dimensions, and scrutinized concept hierarchies. Indeed, the data warehouse

E. B. Ahmed (✉)
Higher Institute of Management of Tunis, University of Tunis, Tunis, Tunisia
e-mail: eya.benahmed@gmail.com

A. Nabli
Faculty of Sciences of Sfax, Sfax University, Sfax, Tunisia
e-mail: ahlem.nabli@fsegs.rnu.tn

F. Gargouri
Higher Institute of Computer Science and Multimedia of Sfax,
Sfax University, Sfax, Tunisia
e-mail: faiez.gargouri@isimsf.rnu.tn

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_3

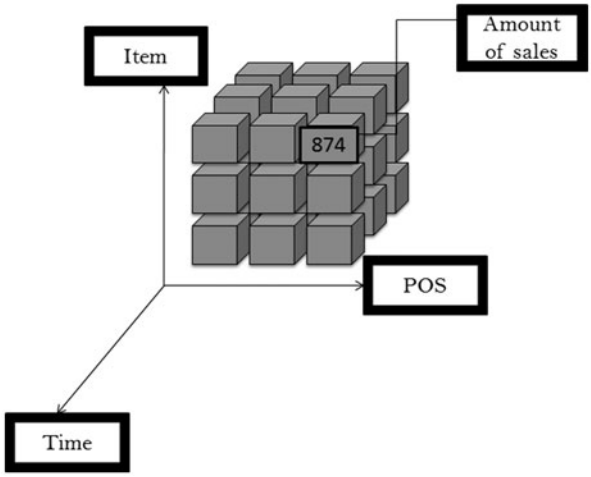


Fig. 1 Sales data cube

contains aggregated data, described by means of several dimensions that are organized through hierarchies. Thus, an excessive patterns number is generated due the highly spare data particularly, at the low hierarchy level. Such generated rules are usually irrelevant and highly diverge from the decision maker expectations. Hence, it is fundamental to mine data at different levels of granularities. Such extracted rules using concept hierarchies are known as multi-level association rules (MLAR). The latter uses only simple hierarchy to extract patterns from data cubes.

Nonetheless, in real cases, the same dimension has more than one analytical purpose. Thus, two or more hierarchies are related to such a dimension. The consideration of this particularity in the mining process may effectively investigate the various levels of granularities within divergent goals of analysis to better fit the analyst expectations.

As data are historized, we argue that cyclic patterns which basically discover rules that occur in user-defined intervals at regular periods, are well-suited to this task. In this chapter, we try to extend the use of concept hierarchies, for dimensions, particularly parallel ones, during the mining process. The core idea behind our approach is to generate multi-level hybrid cyclic patterns combining the multiple-levels of the dimensional concept hierarchies and the parallel concept hierarchies which are employed to express several granularities of given dimension depending on the analysis context.

For instance, we consider the three-dimensional cube illustrated by Fig. 1 showing the sales of articles in pharmaceutical company. Such an OLAP data cube organizes data with dimensions (*i.e.* categorical attributes) and measures (*i.e.* summary statistics). The related dimensions are the `Time` T of transactions, the `Item` I which was bought, and the `Point Of Sale` *POS* where the item is bought. The involved measure is the `amount of sales`.

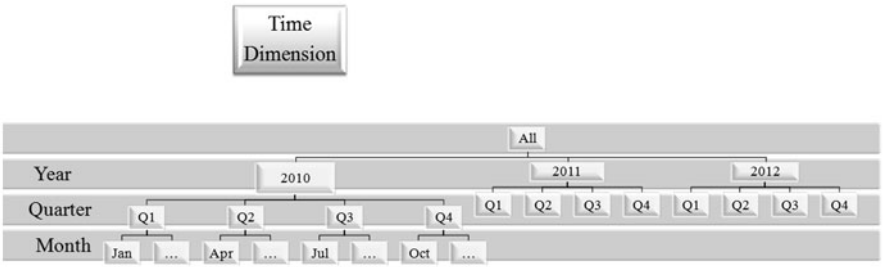


Fig. 2 Concept hierarchy for the `time` dimension

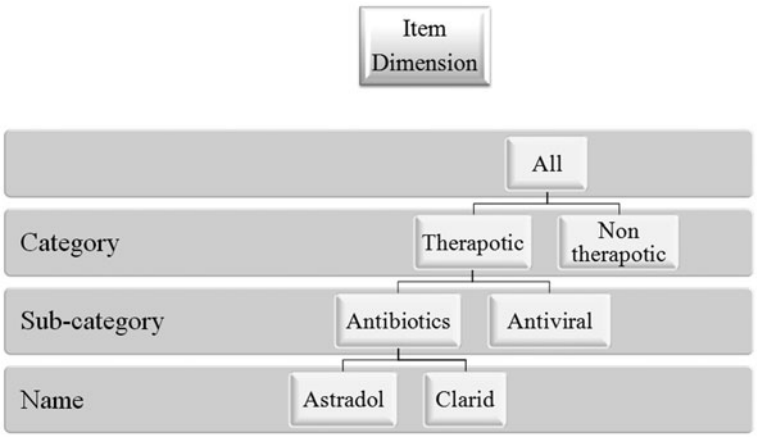


Fig. 3 Concept hierarchy for the `item` dimension

As data are aggregated in data cubes, the dimension is structured into a containment-like hierarchy composed of a number of levels, each of which corresponds to a level of detail that is of interest to performed analysis. In the following, we detail our dimensional concept hierarchies.

Figure 2 depicts the concept hierarchy for the `Time` dimension, represented using only one tree and composed of `Time` → `Month` → `Quarter` → `Year` → `All`. However, the concept hierarchy for the `Item` dimension is shown by Fig. 3 and built of `Item` → `Name` → `Sub-Category` → `Category` → `All`. Figure 4 shows the concept hierarchies `Point of sale` used as background knowledge to generalize the location of the bought item. Differently from the other dimensions, the `Point of sale` dimension is described using two hierarchies: the first hierarchy is composed of `POS` → `City` → `Country` → `All`, and the second is represented by `POS` → `Sales Group Division` → `Sales Group Region` → `All`.

In this case, the hierarchies are considered as parallel because the `Point of sale` dimension has two hierarchies with diverse analytical goals, for example, the

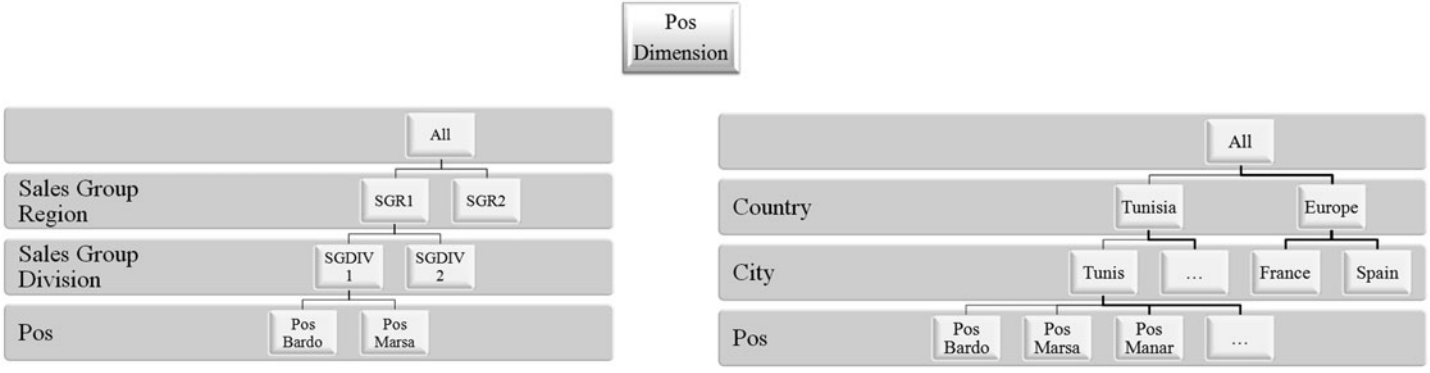


Fig. 4 Parallel concept hierarchies associated to the `point of sale` dimension

Table 1 Table $\mathcal{T}$

Time T	Item I	Point Of sale POS
Jan 2010	Astradol	PosBardo
Feb 2010	Astradol	PosBardo
Mar 2010	Astradol	PosBardo
Apr 2010	Astradol	PosBardo
May 2010	Clarid	PosMarsa
Jun 2010	Clarid	PosMarsa

member values of `Point of sale` can be analyzed using the geographic location or the organization structure criteria. Apparently, such hierarchies are mutually non-exclusive, *i.e.,* it is possible to compute the aggregates grouped by both geographic location and/or organization structure.

In such a context, it is interesting to discover the cyclic associations between the historical data. For example, the decision marker tends to find out the cyclic correlation existing between the `item` such as *Astradol* and its `point of sale` provided through its sales group division such as *SGDIV1* and its geographic position such as *Tunis*. Such correlation is cyclic and it is repeated every month in the sales data cube as shown by Table 1 where only dimension members are illustrated.

To better assist the knowledge worker in his decision process, we aim at discovering patterns that take cyclicity into account and that involve several dimensions within parallel hierarchies.

This chapter extends our research presented in [4]. Its major contributions are: (i) a theoretical framework for mining multi-level hybrid cyclic patterns from parallel dimensional hierarchies, (ii) proposal of algorithm called MIHYCAR to generated such patterns, (iii) demonstration of the efficiency of our method.

The rest of the chapter is organized as follows. First, we address a survey of related work in Sect. 2 and we explain why existing works are not suitable for mining multi-level association rules from dimensional parallel hierarchies. We recall the formal background in Sect. 3. Section 4 describes the core of our approach through introducing our new definitions and detailing our MIHYCAR algorithm. Carried out experiments on synthetic and real data, are reported in Sect. 5. Finally, we conclude and propose future work in Sect. 6.

2 Related Works

In this section, we present existing works from the literature on cyclic patterns and multidimensional association rules.

2.1 Cyclic Patterns

The association rules are the most useful data mining technique for conducting market basket analysis. Since technique was first introduced in [1], it was extended to diverse classes of association rules, namely generalized rules and multi-level rules [8], constraint-based rules [16], cyclic association rules [14].

The topic of cyclic association rules mining has been extensively studied in the last years. Cyclic association rules are known as rules that occur in expert-defined intervals at fixed periods throughout a dataset and may be used in period predictions [9]. For example, *"at weekends, customers who purchase coffee also purchase doughnuts"*.

The input data is a set of transactions, each of which consists of a set of items. Besides, each transaction is tagged with a runtime. The aim is to discover association rules that replicate themselves throughout the input data. An association rule reveals a cyclic behavior. It has a cycle (l,o) if the association holds in every lth time unit starting at time unit o. For instance, if the unit of time is an hour and *"coffee → doughnuts"* holds during the interval 7 – 8 a.m.every day (*i.e.,* every 24 h), the rule *"coffee → doughnuts"* has a cycle (24,7).

Several methods were proposed to discover cyclic patterns. The SEQUENTIAL method is the first traditional method to find such rules [14]. It applies an algorithm similar to APRIORI [1], and after generating the set of classical rules, it detects the cycles behind the rules. Inspired from the perfect periodicity of cyclic association rules, if we previously discern that a rule does not hold at a particular time instant, then the rule will not hold at a specific time instant then the rule will not hold in any cycle which involves this time moment. Based on this idea, a more efficient method to derive cyclic rules is introduced under INTERLEAVED algorithm and it consists on inverting the SEQUENTIAL's process: first learn the cyclic large itemsets and then generate the rules [14].

More flexible, Thuan studied the discovery of association rules in term of time schemas instead of intervals [17]. He developed MLP algorithm which uses time schema (day, month,year) to generate all rules that occur daily, monthly, and yearly.

Chiang et *al.* couple the mining of cyclic patterns and sequential association rules [7]. In market basket analysis, the expert may easily identify whether the selling is cyclic and how long the period between the two successive items in the sequential pattern is, thanks to such derived knowledge.

To palliate the drawbacks of the already presented approaches, the PCAR method was presented [2]. It is a three-phase based algorithm. Its basic idea is: (i) segmentation of the database in a number of partitions fixed by the user; (ii) scan of the database sequentially done partition by partition to generate the frequent cyclic itemsets; (iii) derivation of cyclic association rules from frequent cyclic itemsets.

2.2 Multi-dimensional Association Rules Mining

Academic studies were, essentially, focused on generating association rules from several dimensions. Some of them took benefit from the aggregated data stored in datacubes and derived rules at different level of abstraction.

In this subsection, we classify and discuss the multidimensional association rules according to hierarchy-based criterion. Indeed, the proposed methods are usually classified into two categories: *(i) single-level association rule*s when users are interested in deriving rules among items only at the same level, and *(ii) multi-level association rules* when mining is performed at multiple-levels of abstraction.

2.2.1 Single-level Association Rules

Kamber et *al.* investigated the data cube structure to generate multidimensional association rules [12]. Ben Messaoud et *al.* [6] generate inter-dimensional association rules from data cubes according to sum-based aggregate measure instead of frequencies offered by the COUNT function. Ben Ahmed et *al.* proposed the cyclic association rules extraction from diverse data cube dimensions [5]. Moreover, the authors extended this model to include datacube measures in the mining process [3].

2.2.2 Multi-level Association Rules

In [18], the authors provide an original framework for mining association rules in data warehouses through the measurement of aggregate data. Two algorithms, namely HAvg and VAvg, are introduced to create an initial table used as input to generated association rules. Plantevit et *al.* [15] present a novel method for sequential association rules extraction from data warehouses taking advantage of the diverse dimensions and their levels of granularity. Extended definitions and appropriate algorithms are advanced in such multidimensional context.

Figure 5 summarizes the surveyed approaches for cyclic and multidimensional association rules compared to some criteria, namely the temporality, the number of used dimensions, the number of levels in dimensional hierarchy and constraint involving. First, based on temporality, both of [7] and [15] methods generate sequential patterns. Commonly, cyclic patterns are extracted from one dimension except the algorithm introduced in [3]. [5, 6, 12] neglect the constraint inclusion on association rules mining. Yet, all the rest of approaches are backboned on constraints to reduce the search space.

It should be noted that although this panoply of approaches, only one concept hierarchy is related to dimension from which multi-level patterns are extracted. Nonetheless, some dimensions may be analyzed in several contexts according to different analysis criteria. Indeed, more than one concept hierarchy is related to each dimension. Such hierarchies are common and known as *parallel hierarchies*.

Method	Temporality			Dimension			Hierarchy		Constraint	
	Non-temporal	Sequential	Cyclic	Intra-dimensional	Inter -dimensional	Hybrid	Single-level	Multi-level	constraint-based	Without constraints
(Kamber et al.,1997)	x				x		x			x
(Ozden et al.,1998)			x	x			x		x	
(Thuan,2010)			x	x			x		x	
(Tjioe and Taniar,2005)	x				x			x	x	
(Ben Messaoud et al.,2006)	x				x		x			x
(Chiang et al.,2009)		x	x	x			x		x	
(Plantevit et al.,2010)		x			x			x	x	
(Ben Ahmed and Gouider,2010)			x	x			x			x
(Ben Ahmed et al.,2010,2011)			x		x		x		x	
(Our approach,2011)			x			x		x	x	

Fig. 5 Comparison of cyclic and multidimensional association rules mining approaches

To best of our knowledge, no work handles the generation of association rules from several dimensions with parallel hierarchies.

To overcome this shortcoming, we propose a new method for mining cyclic patterns from data warehouses, taking benefit of both of parallel hierarchies associated to some dimensions and levels of granularity. The required definitions and algorithm are extended from regular cyclic patterns to this adavanced context.

3 Formal Background

In this section, we recall the formal background that will be of use in the remainder.

3.1 Dimensions and Hierarchies

Definition 1 (***Concept Hierarchy for Dimension***)
A *concept hierarchy* for dimension is a tree whose nodes are elements belonging to the domain of this dimension [11]. It is a set of binary relationships between dimension levels. A dimension level participating in a hierarchy is called *hierarchical level* or in short *level*. The sequence of these levels is called a *hierarchical path* or in short *path*. The number of levels forming a path is called the *path length*. The first level of a hierarchical path is called *leaf* and the last is called *root* generally denoted by *ALL*. The *root* represents the most generalized view of data. The *edges* are considered as *is-a* relationships between members. Given two consecutive levels of a hierarchy, the higher level is called *parent* and the lower level is called *child*. Every instance of a level is called *member*.

Example 1 The concept hierarchy of the Time dimension is depicted by the Fig. 2. The ALL *attribute is the* ***root****, the* Month *is the* ***child*** *and* 2011 *is the* ***member****.*

Advanced hierarchies are so widespread. Several categorizations of concept hierarchies for dimension exist. In our context, we focus on the parallel concept hierarchies.

Definition 2 (***Parallel Concept Hierarchies for Dimension***)
Parallel hierarchies arise when a dimension has associated several hierarchies accounting for different analysis criteria [11]. Such hierarchies can be independent or dependent. In a parallel independent hierarchies, the different hierarchies do not share levels, *i.e.*, they represent non-overlapping sets of hierarchies.

Example 2 An example of parallel concept hierarchies is depicted by Fig. 4. In the first concept hierarchy of point of sale, each POS *is mapped into corresponding* city*, which is finally mapped into a corresponding* country*. And the second concept hierarchy, each* POS *is mapped into* sales group division*, which is mapped into* sales group region*.*

3.2 Dimensions Partition

We consider that all is set in a multidimensional context. The three necessary data for cyclic mining drawn from classic context (Customer, Product, Date) become, in multidimensional context, sets.

We consider that the table T, related to the sales data issued by customers, defined on a set $\mathcal{D}$ of n dimensions is partitioned into two sets [5]:

- *Context dimensions* $\mathcal{D}_C$ which concern the investigated dimensions;
- *Out of context dimensions* $\mathcal{D}_{\overline{C}}$ related to the rest of uninvestigated dimensions or the complementary dimensions.

The context dimensions can be divided into three subcategories [15]: (i) *Temporal dimension* $\mathcal{D}_T$: introducing a relation of temporal order (date in classical context), (ii) *Reference dimensions* $\mathcal{D}_R$: the table is segmented according to the reference dimensions values (customer in classical context), and (iii) *Analysis dimensions*: $\mathcal{D}_A = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ with $\mathcal{D}_i \subset \mathrm{Dom}(\mathcal{D}_i)$ corresponding to products in the classic context and relative to dimensions from which will be extracted the cyclic correlations.

Example 3 In our running example shown by Table 1 *, we consider the whole table as our context composed of: (i) context dimensions* $\mathcal{D}_C = \{T, I, POS\}$ *with the temporal dimension* $\mathcal{D}_T = \{T\}$*, the reference dimension* $\mathcal{D}_R = \emptyset$ *and the analysis dimensions* $\mathcal{D}_A = \{I, POS\}$*.*

4 MIHYCAR: A Novel Approach for Multi-level Hybrid Cyclic Association Rules Extraction

4.1 Innovative Concepts

4.1.1 Concept Hierarchies Partition

The analysis dimensions may be organized using one or more concept hierarchies. The latter can be partitioned into two sets:

- *Context concept hierarchies* $\mathcal{H}_C$ concern the set of involved concept hierarchies related to the analysis dimensions $\mathcal{D}_A$;
- *Out of context concept hierarchies* $\mathcal{H}_{\overline{C}}$ which report the set of unexplored concept hierarchies related to the analysis dimensions $\mathcal{D}_A$.

Let $\mathcal{T}_{1DA} = \{\mathcal{T}_{DA1}, \ldots, \mathcal{T}_{nDAm}\}$ the set of the n concept hierarchies associated to the m analysis dimensions. The elements of the analysis dimension $\mathcal{D}_{A1}$ are summarized using k concept hierarchies organizing the hierarchical relationships between the elements of this dimension:
$\mathcal{T}_{DA1} = \{\mathcal{T}_{1DA1}, \ldots, \mathcal{T}_{kDA1}\}$.

We assume that the k concept hierarchy of the i analysis dimensions $\mathcal{T}_{kDAi}$ is an oriented tree; $\forall$ node $n_i \in \mathcal{T}_{kDAi}$, label$(n_i) \in$ Dom(D_{Ai}).

Example 4 In our running example in the respect of the concept hierarchies shown by Figs. 3 and 4, we consider $\mathcal{T}_{DA} = \{\mathcal{T}_I, \mathcal{T}_{1POS}, \mathcal{T}_{2POS}\}$, *with the* $\mathcal{T}_I$ *is illustrated by Fig. 3,* $\mathcal{T}_{1POS}$ *is depicted in the left side of Fig. 4 and* $\mathcal{T}_{2POS}$ *is shown by the right side of Fig. 4.*

4.1.2 Generalization/Specialization in the Concept Hierarchies

We denote by $\blacktriangle x$ (respectively $\blacktriangledown x$) the set containing x along with all generalizations (respectively specializations) of x with respect to $\mathcal{T}_{DA1}$ that belong to Dom$(\mathcal{D}_{A1})$. Each analysis dimension $\mathcal{D}_{Ai}$ is instantiated using only one value d_{Ai} considered as node having the leaf label in the k concept hierarchy associated to the dimension $\mathcal{D}_{kAi}$.

Example 5 In our running example shown by Fig. 4, we consider $x =$ Tunis $\in \mathcal{T}_{2POS}$; *the specialization of* x *is* i.e., $\blacktriangledown x =$ ▼Tunis $=$ PosBardo *and the generalization of* x *is* i.e., $\blacktriangledown x =$ ▼Tunis $=$ Tunisia.

4.1.3 Multi-level Dimensional Cyclic Item and Multi-level Hybrid Cyclic Itemset

Definition 3 (***Multi-level Dimensional Cyclic Item***)
Let the analysis dimensions $\mathcal{D}_A = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ and a cycle length l. A multi-level dimensional cyclic item α is an item belonging to one of the analysis dimensions,

namely $\mathcal{D}_k$ and having a value of d_k for the date t and the date $t + l$ with $d_k \in \{\mathcal{T}_{Dk}\}$ and such that $\forall\, k \in [1,m]$, $d_k \in \mathrm{Dom}(\mathcal{D}_k)$.

Unlike the transactional databases, a multi-level dimensional cyclic item can be generalized using any value node associated to d_i in the k concept hierarchy without necessarily being a leaf.

Example 6 Typical example of multi-level dimensional cyclic item, considered in the multidimensional context, shown by the Table 1 and the delimitation of the context considered previously, is $\alpha =$ (PosBardo) *because it belongs to the* POS *dimension, being a part of analysis dimension and its value* PosBardo *belongs to the POS domain and is repeated each month of the first quarter of 2010.*

Definition 4 (***Multi-level Hybrid Cyclic Itemset***)
A multi-level hybrid cyclic itemset F defined on $\mathcal{D}_A = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ is a nonempty set of multi-level dimensional cyclic items $F = \{\alpha_1, \ldots, \alpha_m\}$ with $\forall\, \mathrm{j} \in [1, m]$, α_j is a multi-level dimensional cyclic item defined on $\mathcal{D}_j$ at the date t and it is repeated at each date $t + l$ with $\forall\, j,k \in [1, m]$, $\alpha_j \neq \alpha_k$.

Example 7 An example of multi-level hybrid cyclic itemset is $F =$ [Astradol, PosBardo] *because it is composed of two multi-level hybrid cyclic items* i.e., $\alpha_1 =$ (Astradol), $\alpha_2 =$ (PosBardo). *It is repeated monthly during the first quarter of 2010.*

4.1.4 Connectivity of Multi-level Hybrid Cyclic Itemsets

We study the connectivity by scrutinizing the different relationships that may exist between the multi-level hybrid cyclic itemsets. Let two multi-level hybrid cyclic itemsets $F = (d_1, \ldots, d_m)$ and $G = (d'_1, \ldots, d'_m)$, two types of connectivity between those itemsets are considered:

1. Connected multi-level hybrid cyclic itemsets
2. Disconnected multi-level hybrid cyclic itemsets

Definition 5 (***Disconnected Multi-level Hybrid Cyclic Itemsets***)
F and G are disconnected iff they do not belong to the same concept hierarchies.

Example 8 $F =$ SGDIV1 *and* $G =$ Tunisia *are disconnected because they do not belong to the same concept hierarchies,* $F =$ SGDIV1 $\in \mathcal{T}_{1POS}$ *and* $G =$ Tunisia $\in \mathcal{T}_{2POS}$.

Definition 6 (***Connected Multi-level Hybrid Cyclic Itemsets***)
F and G are connected iff they belong to the same concept hierarchies.

Example 9 $F =$ PosBardo *and* $G =$ Tunisia *are connected because they belong to the same concept hierarchy.*

If the multi-level hybrid cyclic itemsets are connected, two possible relationships may be outlined:

1. Covered multi-level hybrid cyclic itemsets
2. Uncovered multi-level hybrid cyclic itemsets.

Definition 7 (***Covered Multi-level Hybrid Cyclic Itemsets***)
F is covered by G iff $\forall\, d_i, d_i = \blacktriangle\, d_i'$ or $d_i = d_i'$.

Example 10 $F =$ [Astradol, PosBardo] *is covered by* $G =$ [Antibiotic, Tunis] *because* Tunis $= \blacktriangle$ PosBardo *and* Antibiotic $= \blacktriangle$ Astradol.

Definition 8 (***Uncovered Multi-level Hybrid Cyclic Itemsets***)
F is uncovered by G iff $\forall\, d_i, \blacktriangle\, d_i \neq d_i'$.

Example 11 $F =$ [Astradol, PosBardo] *is uncovered by* $G =$ [Antiviral, Tunis] *because* $\blacktriangle$ Astradol $\neq$ Antiviral.

If two multi-level hybrid cyclic itemsets are covered, two eventual relationships may be highlighted:

1. Adjacent multi-level hybrid cyclic itemsets
2. Non adjacent multi-level hybrid cyclic itemsets

Definition 9 (***Adjacent Multi-level Hybrid Cyclic Itemsets***)
F belongs to the n hierarchical level, G is considered as its adjacent multi-level hybrid cyclic itemset iff G belongs to $n-1$ level or $n+1$ level of the same concept hierarchy.

Example 12 $F =$ [Astradol, PosBardo] *belonging to the 1-level is adjacent to* $G =$ [Antibiotic, Tunis] *because G belongs to the 2-level in the concept hierarchies depicted by both Figs. 3 and 4.*

Definition 10 (***Non Adjacent Multi-level Hybrid Cyclic Itemsets***)
F belongs to the n hierarchical level, G is considered as a non adjacent multi-level hybrid cyclic itemset of F if G does not belong to $n-1$ level or $n+1$ level of the same concept hierarchy.

Example 13 $F =$ [Astradol, PosBardo] *belongs to the 1-level and is not adjacent to* $G =$ [Africa, Therapeutic] *because G belongs to the 3-level in the concept hierarchies which is not the 2-level in the concept hierarchies.*

4.1.5 Support of Multi-level Hybrid Cyclic Itemset

Definition 11 (***Support of Multi-level Hybrid Cyclic Itemset***)
The support of multi-level hybrid cyclic itemset, denoted $Supp(F)$ is the number of tuples that contain the itemset; $Supp(F) = COUNT(F)$.

Example 14 Consider the context shown by Table 1 *and the delimitation already presented. The multi-level hybrid cyclic itemset* $F =$ (Antibiotics, PosBardo, SGDIV1) *has an absolute support related to the sales of the products considered as Antibiotics and which are sold in the first sales group division* SGDIV1 *in* PosBardo*:*

$$\boldsymbol{Supp}(\textit{Antibiotics, PosBardo, SGDIV1}) =$$

$$\boldsymbol{COUNT}(\boldsymbol{I} = \textit{Antibiotics}, \textit{POS} = \textit{PosBardo} \wedge \textit{SGDIV1}) = 4$$

4.1.6 Support and Confidence Computing of Multi-level Hybrid Cyclic Rule

Definition 12 (***Support of Multi-level Hybrid Cyclic Rule***)
The rule support $R : F \Rightarrow G$, denoted $Supp(R)$, is equal to the ratio of the number of tuples that contain F and G to the total number of tuples in the sub-cube.

$$Supp(R) = \frac{COUNT(F \cup G)}{COUNT(ALL, ALL)};$$

The support of de R, $Supp(R) \in [0, 1]$.

Definition 13 (***Confidence of Multi-level Hybrid Cyclic Rule***)
The rule confidence $R : F \Rightarrow G$, denoted $conf(R)$, is equal to the ratio of the number of tuples that contain F and G to the number of tuples that contain F in the sub-cube.

$$conf(R) = \frac{Supp(R)}{Supp(F)};$$

The confidence of R, $conf(R) \in [0, 1]$.

Example 15 In our running example, the rule R: Antibiotics, PosBardo ⇒ SGDIV1 has:

$$Supp(R) = \boldsymbol{COUNT}(\boldsymbol{I} = \textit{Antibiotics}, \boldsymbol{POS} = \textit{PosBardo} \bigwedge \textit{SGDIV1}) = 4$$

$$conf(R) = \frac{\boldsymbol{COUNT}(\boldsymbol{I} = \textit{Astradol}, \boldsymbol{POS} = \textit{PosBardo} \bigwedge \textit{SGDIV1})}{\boldsymbol{COUNT}(\boldsymbol{I} = \textit{Astradol}, \boldsymbol{POS} = \textit{PosBardo})} = \frac{4}{4} = 1$$

4.2 MIHYCAR Method

In spite of the panoply of proposed approaches dedicated to multidimensional mining of association rules, no proposal focuses on multidimensional context where parallel hierarchies arise to account for various analysis purposes. Thus, we introduce our innovative method termed MIHYCAR to handle such advanced context. In this subsection, we present the notations used with MIHYCAR. After that, we detail the process of our algorithm.

Table 2 Encoded data cube T

Item	Encoded item	POS	Encoded POS
Astradol	[2-1-1-1]	*PosBardo*	[3-*-2-1]
Clarid	[2-1-1-1]	*PosMarsa*	[3-*-2-1]

4.2.1 Notations

The following notations will be used in the remainder:

- SC: Sub-cube;
- $\mathcal{D}_t$: Date t;
- lc: Length of cycle;
- $\mathcal{M}insupp$: Minimum support threshold;
- nd: Number of dimensions;
- d: Current dimension;
- h: Current hierarchy of dimension;
- $depth$: Depth of the current concept hierarchy;
- l: Current level;
- $\mathcal{C}[d,h,l,k]$: Set of candidates from the dimension d belonging to the hierarchy h and the level l having k itemsets;
- $\mathcal{F}[d,h,l,k]$: Set of frequent itemsets extrated from the dimension d belonging to the hierarchy h and the level l having k items;
- $Supp(\mathcal{C})$: Support of the multi-level hybrid cyclic itemset $\mathcal{C}$;
- s: Nonempty subset s of $\mathcal{F}_i$.

4.2.2 MIHYCAR Algorithm

First, we describe the key input of our algorithm which is the hierarchy-information encoding our multi-dimensional data cube. Indeed, this encoding operation greatly facilitates the extraction of frequent items. Its basic idea consists in representing each item using an encoded predicate string as follows '[d-h-l-k]' with d is the dimension, h is the concept hierarchy of the d dimension, h indicated the abstraction level in the concept hierarchy, and k represents the number of itemsets. For instance, *Tunisia* is encoded as follows [3-2-3-1] with 3 represents the Point of Sales dimension, 2 represents the second hierarchy concept and 3 describes the level of abstraction in the concept hierarchy, and 1 represents the number of itemsets which is 1-item in our case.

To better demonstrate this encoding phase, we present in Table 2 a sample of encoded items from our illustrative example.

The processing steps of our algorithm MIHYCAR can be described as follows:

1. For each dimension, the scan of all related concept hierarchies is performed and for each level l, the frequent multi-level dimensional cyclic items $\mathcal{F}[1,h,l,1]$ are extracted. After scanning, we filter out all multi-level dimensional cyclic items

whose the support is smaller than the minimum support threshold as shown by the procedure $ComputingSupport$.

2. The frequent k (for $k > 1$) itemsets for each level l are derived in two steps:
   (a) Compute the candidate set from $k-1$ frequent multi-level dimensional cyclic itemsets, as done in the APRIORI candidate generation method,
   (b) Compute the support of generated candidates and prune the infrequent ones.
3. After finding the frequent itemsets, the set of association rules for each level l can be derived from the frequent itemsets based on the minimum confidence. This is performed as follows. For every large itemset r, if a is a nonempty subset of r, the rule $r - a \Rightarrow a \rightarrow u''$ is generated when the confidence of the rule is greater than the minimum confidence. An example of generated rule is r : $Antibiotics, PosBardo \rightarrow SGDIV1$.

Algorithm 1: MIHYCAR: MultI-level Hybrid Cyclic Association Rules

```
Data: SC, 𝒰insupp
Result: Multiple-levels frequent itemsets.
begin
    // initialisation
    d=1; h=1;l=1;
    ℱ[d,h,l,1]= Find 1-frequent cyclic itemsets(SC, l,𝒟_t, 𝒰inSupp) ;
    for (d=1; d <= nd ; d++ ) do
        //scan of dimensions
        for (h=1; h < depth; h++) do
            //scan of concept hierarchies of each dimension
            for (l=1; ℱ[d,h,l,1] ≠ ∅; l++ ) do
                //scan of concept hierarchies levels of each dimension
                for (k=2; ℱ[d,h,l,k − 1] ≠ ∅; k++ ) do
                    𝒞[d,h,l,k] = CandidatGeneration (ℱ[d,h,l,k-1]);
                    if 𝒞[d,h,l,k] is a hybrid cyclic itemset then
                        foreach transaction 𝒯 ∈ SC at date 𝒟_t do
                        𝒞[d,h,l,t]=subset(𝒞[d,h,l,k], 𝒯)
                        foreach candidat 𝒞 ∈ 𝒞𝒞[d,h,l,t] do
                        ;
                        𝒞.support = SupportComputing(SC, l,𝒟_t, 𝒞);
                        ℱ[d,h,l,k] = { 𝒞 ∈ 𝒞[d,h,l,k], 𝒞.support > 𝒰insupp
                        } ;
                Return ℱ[d,h,l,k] = ∪_k ℱ[d,h,l,k]
```

```
Function Find 1-frequent cyclic itemsets (SC, l,𝒟_t, 𝓜inSupp)
Result: ℱ_1
begin
    while (!End of tuples in SC) do
        foreach transaction 𝒯 ∈ SC do
        ;
        foreach item α ∈ 𝒯 do
        ;
        foreach transaction 𝒯' ∈ SC at date 𝒟_{t+l} do
        ;
        Supp(α)=COUNT(α);
        if (Supp(α) > 𝓜inSupp ) then
            ℱ[d,h,l,1] = ℱ[d,h,l,1] ∪ α;
    Return ℱ[d,h,l,1] ;

Function SupportComputing (SC, l,𝒟_t, 𝒞)
Result: Supp(𝒞)
begin
    NoMoreCyclic: Boolean;
    NoMoreCyclic = false;
    while ((!End of tuples in SC) and (!NoMoreCyclic)) do
        𝒞[d,h,l,k] = CandidatGeneration (𝒞[d,h,l,k − 1]);
        foreach transaction 𝒯 ∈ SC at date 𝒟_{t+l} do
        ;
        if 𝒞 exists in 𝒯 then
            Supp(𝒞)= Supp(𝒞)+1;
        NoMoreCyclic = true;
    Return Supp(𝒞) ;
```

5 Experimental Study

In this section, we report experiments performed on synthetic data and real data. All experiments were carried out a PC equipped with 1.73 GHz and 1 GB of main memory.

5.1 Real Data Cube

In the following, we report experiments performed on a real sales data warehouse [1], which contains three dimensions (e.g., `Time` dimension, `Item` dimension, `point of sale` dimension) and one sales fact table. The data warehouse is built using relational OLAP (ROLAP) and is modeled in a star schema, which contains dimension tables for the hierarchies and a fact table for the dimensional attributes and measures. Our objective is to show, through our extensive experimental study: (i) the performance of our algorithm according to the length of cycle and the number of analysis dimensions; (ii) the assessment of the hierarchical aspect in respect of the number of involved concept hierarchies and the average depth of those hierarchies.

[1] The data warehouse is related to pharmaceutical listed company. It is built using the available information at http://www.bvmt.com.tn/companies/?view=listed.

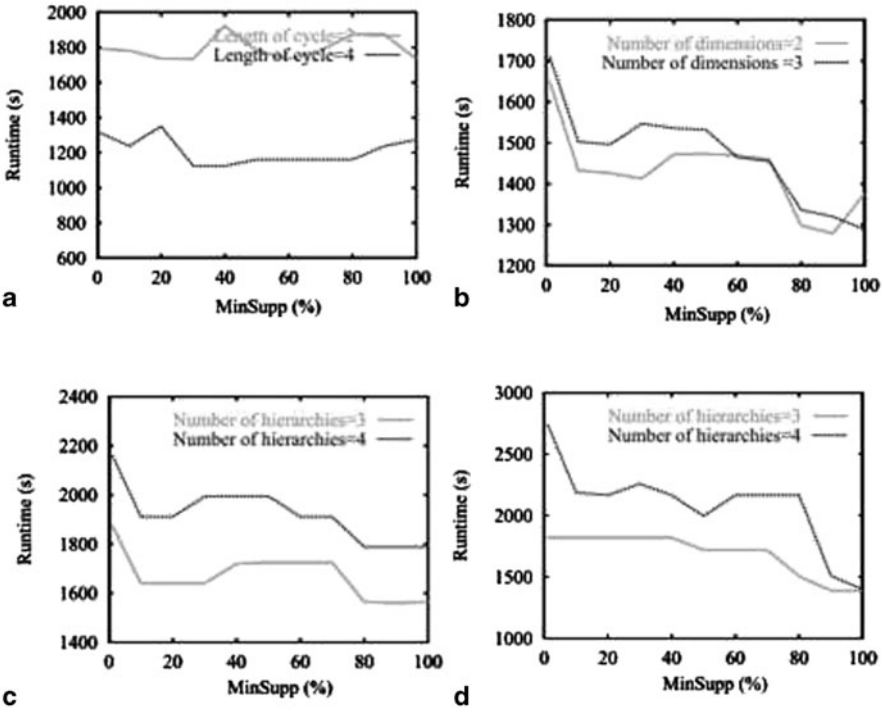


Fig. 6 Experiments carried out on real data

Figure 6a plots the runtime needed to generate multi-level hybrid cyclic association rules with respect of the length of cycle. Clearly, in efficiency terms, it can be seen from this figure that the running time decreases proportionally to the length of cycle. Figure 6b describes the behavior of our approach in terms of runtime according to the number of analysis dimensions. Obviously, we observe that the slopes of the three plots are increasing when the number of analysis dimensions increases. In fact, having more analysis dimensions, more concept hierarchies will be included. So that, the number of generated patterns will highly increase. Through the last experiments, we compare the runtime needed to generate multi-level hybrid cyclic rules over the number of involved concept hierarchies and the average depth of the concept hierarchies, also called the specialization level. Moreover, as shown in Fig. 6c, the number of concept hierarchies related to the analysis dimensions radically influences the performance of our algorithm. Taking more hierarchies into account through parallel hierarchies involving, the runtime of our algorithm significantly increases. Figure 6d shows the number of generated multi-level hybrid cyclic association rules over the depth of the concept hierarchies. In fact, increasing the size of the concept hierarchies brings additional specialization level. Accordingly, our algorithm mines less frequent patterns until it cannot mine any more knowledge.

Table 3 Characteristics of the used datasets

Benchmark	No. of items	No. of objects	Average size of objects	Max size of objects
CHESS	75	3.196	37.00	37
T10I4D100K	870	100.000	10.10	30

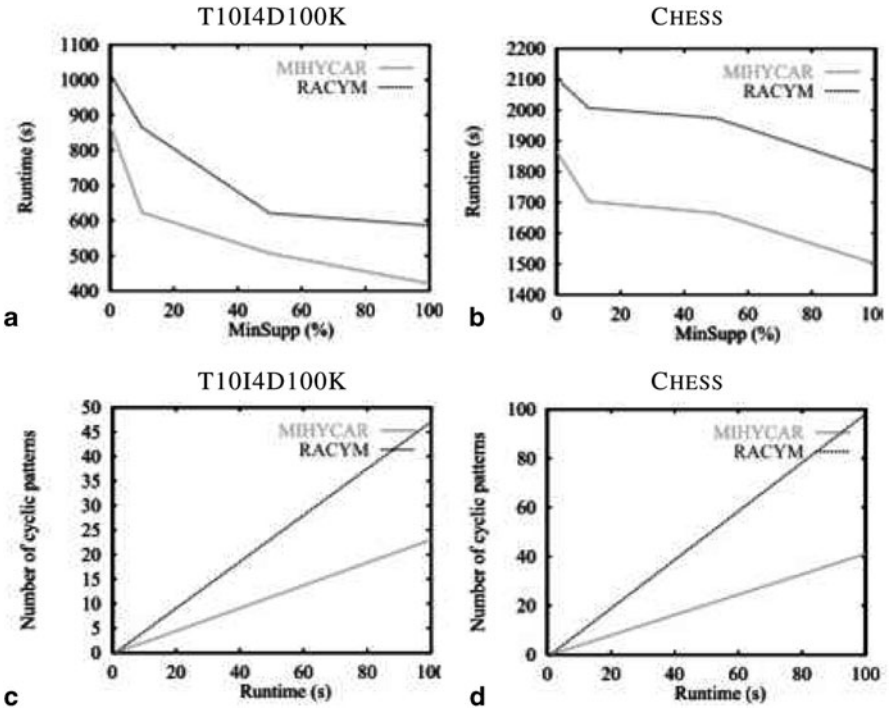


Fig. 7 Experiments carried out on synthetic data

5.2 *Synthetic Data*

We report experimental studies performed on synthetic data. We used CHESS as dense benchmark dataset, and T10I4D100K as a sparse dataset.

Table 3 summarizes dataset characteristics used during our experiments. Through these experiments, we aim to compare the runtime of our method vs. that of RACYM algorithm in multidimensional cyclic association rules extraction.

Figure 7a, b plots the runtime required to mine multidimensional cyclic association rules for considered datasets, using both MIHYCAR and RACYM. Clearly, the MIHYCAR largely outperforms the RACYM strategy especially for dense datasets. Indeed, the gap between both curves tends to become wider as far as the MinSupp decreases. Moreover, the MIHYCAR is more efficient on sparse dataset for all MinSupp values. Figure 7c, d reports the behaviors of MIHYCAR and RACYM according to

the number of mined patterns. The latter increases with the runtime. Therefore, we conclude that MIHYCAR is scalable according to the number of generated patterns. The difference between the performance of MIHYCAR and RACYM reaches its maximal for the CHESS dataset.

6 Conclusions and Future Works

In this chapter, we introduced a new approach called MIHYCAR for cyclic association rules derivation from multi-level hierarchies within parallel concept hierarchies that my exist in advanced multi-dimensional context. We tested our new algorithm on both of synthetic and real data. Experimental results indicate that MIHYCAR is a promising method for cyclic association rules mining from multi-level and parallel hierarchies.

In the future, we plan to extend our current contribution to address several further research directions. Specially, we intend to consider contextual hierarchies for cyclic patterns mining [10]. And also, we would like to tackle the problem of mining cyclic multidimensional patterns in streaming data. Finally, we concentrate on studying semantic cyclic patterns from semantic OLAP framework [13].

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of data, pp. 207–216. ACM, New York (1993)
2. BenAhmed, E., Gouider, M.S.: Towards a new mechanism of extracting cyclic association rules based on partition aspect. In: Proceedings of the International Conference on Research Challenges in Information Science, pp. 69–78. IEEE, Nice (2010)
3. BenAhmed, E., Nabli, A., Gargouri, F.: Cyclic association rules: Coupling between dimensions with measures. In: Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering, pp. 379–384. ACM, New York (2011)
4. BenAhmed, E., Nabli, A., Gargouri, F.: Cyclic association rules: Coupling multiple levels and parallel dimension hierarchies. In: Proceedings of the International Conference on Information and Knowledge Engineering, pp. 192–198. IEEE, USA (2011)
5. BenAhmed, E., Nabli, A., Gargouri, F.: Mining cyclic association rules from multidimensional knowledge. In: Proceedings of the 6th International Conference on Digital Information Management, pp. 12–17. IEEE, Melbourne (2011)
6. BenMessaoud, R., Boussaid, O., Rabaséda, S., Missaoui, R.: Enhanced mining of association rules from data cubes. In: Proceedings of the 9th International Workshop on Data Warehousing and OLAP, pp. 11–18. ACM, Nice (2006)
7. Chiang, D., Wang, C., Chen, S., Chen, C.: The cyclic model analysis on sequential patterns. IEEE Trans. Knowl. Data Eng. **21**(11), 1617–1628 (2009)
8. Han, J., Fu, Y.: Discovery of multiple-level association rules from large databases. In: Proceedings of the International Conference on Very Large Data Bases, pp. 420–431. ACM, Switzerland (1995)

9. Han, J., Gong, W., Yin, Y.: Mining segment-wise periodic patterns in time-related databases. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 214–218. ACM, New York (1998)
10. Ienco, D., Pitarch, Y., Poncelet, P., Teisseire, M.: Towards an automatic construction of contextual attribute-value taxonomies. In: Proceedings of the the 27th Symposium On Applied Computing, pp. 113–118. ACM, New York (2012)
11. Jensen, C., Pedersen, T., Thomsen, C.: Multidimensional databases and data warehousing. Synthesis Lectures on Data Management, Morgan and Claypool Publishers (2010)
12. Kamber, M., Han, J., Chiang, J.: Metarule-guided mining of multi-dimensional association rules using data cubes. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 207–210. ACM, California (1997)
13. Kamber, M., Han, J., Chiang, J.: Qc-trees: an efficient summary structure for semantic olap. In: Proceedings of the International Conference on Management of Data, pp. 64–75. ACM, California (2003)
14. Ozden, B., Ramaswamy, S., Silberschatz, A.: Cyclic association rules. In: Proceedings of the Fourteenth International Conference on Data Engineering, pp. 412–421. IEEE, Florida (1998)
15. Plantevit, M., Laurent, A., Laurent, D., Teisseire, M., Choong, Y.: Mining multidimensional and multilevel sequential patterns. In: Transactions on Knowledge Discovery from Data. pp. 155–174. ACM, Florida (2010)
16. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: International Confernce on Knowledge Discovery in Databases and Data Mining. pp. 67–73. ACM, California (1997)
17. Thuan, N.: Mining time pattern association rules in temporal database. In: Innovations and Advances in Computer Sciences and Engineering, Springer Netherlands, pp. 7–11 (2010)
18. Tjioe, H., Taniar, D.: Mining association rules in data warehouses. Int. J. Data Warehousing Mining (IJDWM) **1**(3), 28–62 (2005)

PROFIT: A Projected Clustering Technique

Dharmveer Singh Rajput, Pramod Kumar Singh and Mahua Bhattacharya

Abstract Clustering high dimensional dataset is one of the major areas of research because of its widespread applications in many domains. However, a meaningful clustering in high dimensional dataset is a challenging issue due to (i) it usually contains many irrelevant dimensions which hide the clusters, (ii) the distance, which is the most common similarity measure in most of the methods, loses its meaning in high dimensions, and (iii) different clusters may exist in different subsets of dimensions in high dimensional dataset. Feature selection based clustering methods prominently solve the problem of clustering high dimensional data. However, finding all the clusters in one subset of few selected relevant dimensions is not justified as different clusters may exist in different subsets of dimensions. In this article, we propose an algorithm PROFIT (PROjective clustering algorithm based on FIsher score and Trimmed mean) which extends the idea of feature selection based clustering to projective clustering and works well with the high dimensional dataset consisting of attributes in continuous variable domain. It works in four phases: sampling phase, initialization phase, dimension selection phase and refinement phase. We consider five real datasets for experiments with different input parameters and consider three other well-known top-down subspace clustering methods PROCLUS, ORCLUS and PCKA along with our feature selection based non-subspace clustering method FAMCA for comparison. The obtained results are subjected to two well-known subspace clustering quality measures (Jagota index and sum of squared error) and Student's *t*-test to determine the significant difference between clustering results. The obtained results and quality measures show effectiveness and superiority of the proposed method PROFIT to its competitors.

D. S. Rajput (✉) · P. K. Singh · M. Bhattacharya
ABV – Indian Institute of Information Technology and Management, Gwalior, MP, India
e-mail: dharmveer@iiitm.ac.in

P. K. Singh
e-mail: pksingh@iiitm.ac.in

M. Bhattacharya
e-mail: mb@iiitm.ac.in

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_4

1 Introduction

Clustering is a process of grouping the data objects in such a way that similar objects belong to one group whereas dissimilar objects belong to different groups. These groups of objects are referred as clusters. The general criterion of a good clustering is that objects in the same cluster are 'close' or related to each other whereas objects of different clusters are 'far apart' or very different. Gene expression analysis, metabolic screening, customer recommendation systems, text documents, insurance, city planning and earthquake studies are some major applications of cluster analysis in science and engineering [17, 21].

The existing clustering methods such as Partitioning methods, Hierarchical methods, Density based methods, Grid based methods and Model based methods are able to identify the clusters in low dimensional datasets only as they consider all the dimensions to learn as much as possible about the objects and use distance as the similarity measure. However, the distance becomes meaningless in high dimensional datasets because in sufficiently large dimensional datasets all objects are equidistant to each other. Moreover, a large number of dimensions are irrelevant [8, 20, 29].

The common approach to cluster a high dimensional dataset is a two step process. Initially, some feature transformation technique, e.g., principle component analysis [13], singular value decomposition [4], is applied in an attempt to summarize the dataset in fewer dimensions by creating linear combinations of the original dimensions. These techniques are very successful in uncovering the latent structure of the datasets but they generally preserve the original, relative distances between objects. Moreover, interpretation of the transformed data is also very difficult. Further, an existing clustering method is applied on this reduced dataset to obtain the clusters.

Rajput et al. [33] proposed an efficient method FAMCA (Fisher-score And trimmed Mean based high dimensional data Clustering Algorithm) to obtain effective clusters in the high dimensional dataset. Initially, it selects the relevant dimensions by using fisher score to reduce the ill effects of high dimensionality. Further, it computes initial clusters centers using trimmed mean then apply K-means to obtain clusters. Generally, the feature selection based clustering methods first select the relevant dimensions from high dimensional dataset then apply clustering algorithm on selected relevant dimensions to find clusters. However, it may not be applicable or appropriate as different dimensions may be relevant for different cluster.

The fact that different clusters may be embedded in different subspaces in high-dimensional dataset is shown by an example due to Parsons et al. [29]. Consider a dataset consisting of 400 objects in three dimensions Fig. 1. This dataset contains four clusters of hundred objects each; two clusters exist in dimensions a and b whereas other two clusters exist in dimensions b and c. Application of conventional clustering algorithms on the full dimensional dataset do not revel all the four separate clusters due to one irrelevant dimension. Feature extraction techniques, e.g., PCA, SVD, also do not help as they perform linear transformation on full dimensional dataset to transform it into low dimensional dataset but they preserve the relative distance and effect of irrelevant dimensions. The application of feature selection techniques

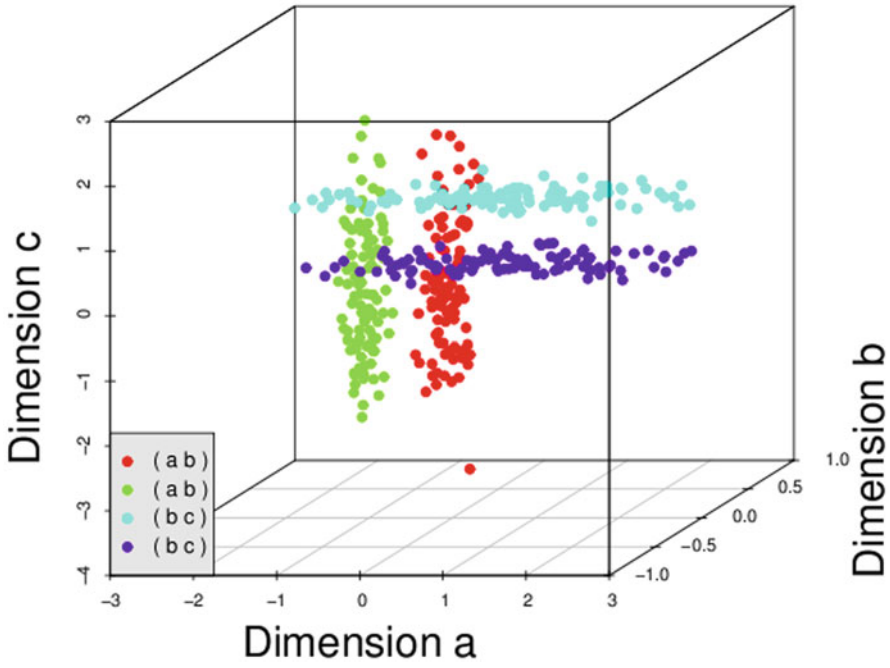


Fig. 1 Sample dataset [29]

to project the dataset onto any of a single dimension also do not help as none of the projection clearly indicates all the four clusters separately Fig. 2. A projection of the dataset onto two dimensions *a* and *b* Fig. 3a revel two clusters (red and green) but other two clusters (blue and purple) are not clearly visible. A projection onto the dimensions *b* and *c* Fig. 3b revel two clusters (blue and purple) but the other two clusters (red and green) are now not distinguishable. A projection onto the dimensions *a* and *c* Fig. 3c too does not separate the clusters fully. Thus, we observe that the reduction of any number of dimension(s) does not fully describe all the four clusters on the dataset as the clusters belong to different subspaces, i.e., different subsets of dimensions.

Subspace clustering is an extension of feature selection that attempts to find relevant clusters in different subsets of dimensions. These methods are broadly classified as bottom-up subspace clustering methods and top-down (projected) subspace clustering methods [29]. The former methods such as CLIQUE [3], ENCLUS [11] use the down word closure property to identify the dense region of every dimension in the dataset and then combine the dense regions to form clusters using disjunctive normal form. These methods produce overlapping clusters and quality of the obtained clusters depends largely on the proper tuning of grid size and density threshold parameter. The latter methods such as PROCLUS [2], ORCLUS [1] initially obtain the clusters on full dimensions and then use feature selection technique to identify the relevant

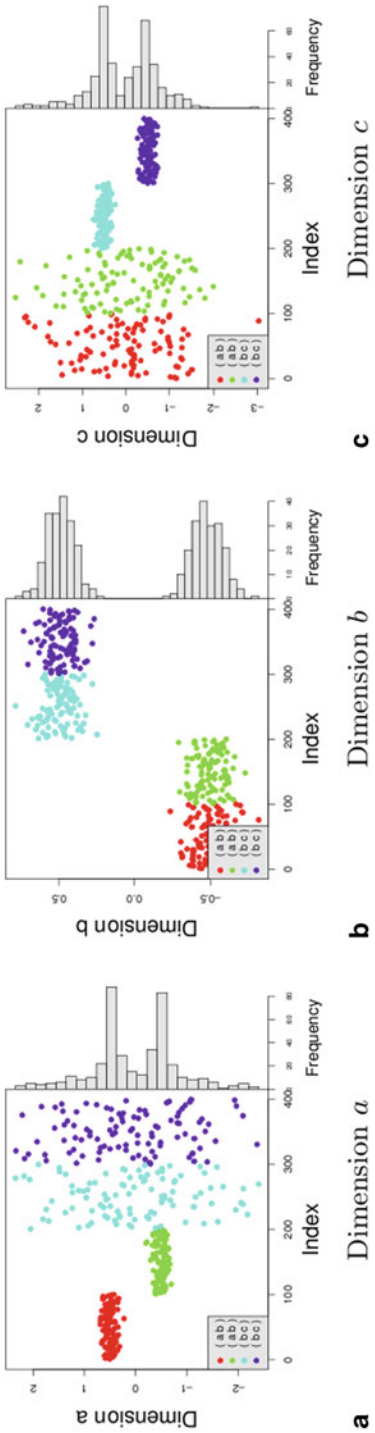


Fig. 2 Sample data plotted in one dimension with histogram [29]

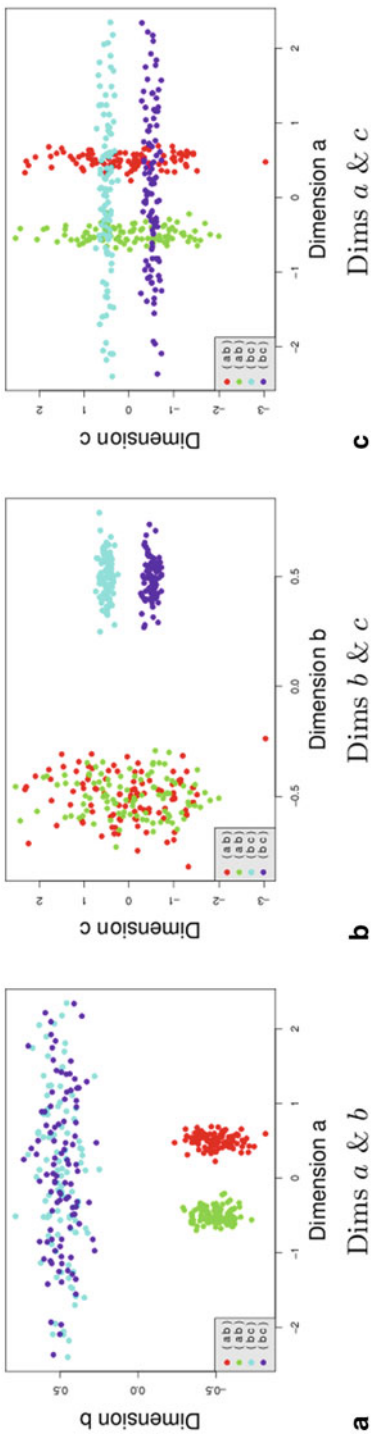


Fig. 3 Sample data plotted in each set of two dimensions [29]

dimensions of each cluster. Though these methods are fast because of sampling, they obtain hyper-spherical clusters of fixed size subspace and are sensitive to parameters like number of clusters, size of subspaces and sample size.

In this article, we propose a top-down subspace clustering method PROFIT (PROjective clustering algorithm based on FIsher score and Trimmed mean), which is an extension of the FAMCA [33], to obtain projected clusters in high dimensional dataset consisting of attributes in continuous variable domain. It consists of four phases: sampling phase, initialization phase, dimension selection phase and refinement phase. Sampling phase determines the representative sample of high dimensional dataset using principal component analysis (PCA). The initialization phase identifies the effective initial clusters centers using the concept of trimmed mean. The dimension selection phase selects the relevant dimensions of each cluster using Fisher score criteria and finally the refinement phase produces the projected cluster of full dataset using iterative process. We experiment with five real datasets (Mammals Milk, Wisconsin Prognostic Breast Cancer, Heart Disease, Image Segmentation and Reuters-21578 Corpus Text database) and find that PROFIT obtains better clusters in comparison to the FAMCA and three other well-known top-down subspace clustering methods PROCLUS, ORCLUS and PCKA based on the well-known subspace clustering quality measures Jagota index and sum of squared error. Moreover, we use Student's t-test to determine the significant difference between the clustering results obtained by the proposed method PROFIT and its competitors.

The rest of the article is organized as follows. A brief summary of the existing high dimensional data clustering methods is presented in Sect. 2. The proposed top-down subspace clustering method PROFIT is explained in Sect. 3. In Sect. 4, we present results and comparisons of our experiments of the four methods on the five real datasets. Finally, Sect. 5 summaries the conclusion and future work.

2 Literature Survey

The literature consists of various methods for initialization of clusters centers, dimension reduction and subspace clustering of high dimensional data clustering. We present briefly some well-known and relevant methods below.

2.1 Initialization of Clusters Centers

Khan and Ahmad [23] present cluster center initialization algorithm (CCIA). Initially, it selects more than k objects (here, k is the number of clusters) from dataset as clusters center and run K-means algorithm, further, it merges two nearest clusters centers to get k initial centers for K-means algorithm. However, it does not address the problem and is inefficient when three or more clusters centers are at equal distance. Arai and Barakbah [6] run K-means algorithm n times on a dataset with different initial clusters

centers and then apply hierarchical algorithm on the set of obtained clusters centers to produce k clusters. Now, the centers of these k clusters serve as effective initial clusters centers for the K-means algorithm. However, it is inefficient because of its computational complexity. Barakbah and Kiyoki [7] propose a pillar designation approach for initialization of clusters centers. It first computes the grand mean of data objects, then selects the farthest object from grand mean having density above a threshold as first initial cluster center. Further, the objects which have density above the threshold and are farthest from all previously selected initial clusters centers are chosen as initial clusters centers. A major drawback of this method is that it may select an outliers as an initial cluster center if the threshold value is not defined properly. Celebi [10] presents various cluster center initialization techniques for the K-means algorithm to improve color quantization of images.

2.1.1 Dimension Reduction

The curse of dimensionality is a major issue in clustering. A common approach to reduce its ill effects is dimension reduction. It helps in understanding and visualization of natural structure of high dimensional dataset. The various methods available for dimension reduction are categorized as Feature extraction methods and Feature selection methods. Feature extraction methods such as principal component analysis [13], singular value decomposition [4] and multi dimensional scaling [25] perform linear transformation on high dimensional dataset to transform it into low dimensional dataset. However, they preserve the relative distances between objects, hence are less effective when large numbers of irrelevant dimensions are present in the dataset. Moreover, the transformed low dimensional dataset is not easily interpretable. The feature selection methods, further categorized as filter approaches and wrapper approaches, select the relevant dimension of dataset [34, 36].

Apolloni et al. [5] propose a method called Boolean Independent Component Analysis (BICA) to select the bits of dimensions using minimal entropy and consistence assignments. It produces a vector of unique assignment to Boolean variable. This assignment indicates the relevance of feature in the dataset. However, it does not produce guaranteed results. Liu et al. [26] create a set of data blocks using clustering algorithm and then determine the intra-cluster and inter-cluster graphs to map the high dimensional dataset into low dimensional dataset. Gheyas and Smith [14] assign ranks to every feature subset of the candidate set; a higher rank indicates a fitter solution. If two subsets are assigned same rank, the subset with smaller number of features wins. However, it fails to select a proper feature subset when two or more subsets are of same rank and size. Hu et al. [18] select heterogeneous features based on the concept of neighborhood rough set. It hybridizes the technique of classification loss and neighborhood margin to evaluate the classification accuracy of feature selection. However, it does not identify the relevant features of nonlinear dataset. Sugiyama et al. [35] propose a method for reducing the dimensions and

improving the density ratio estimation accuracy of high dimensional dataset which combine the concepts of local fisher discriminate analysis and unconstrained least square importance fitting. However, it is very hard to define threshold value. Kabir et al. [22] propose a hybrid method consisting of wrapper approach and sequential search strategy to identify relevant features. However, it suffers from nesting effect because selected feature cannot be discarded.

2.1.2 Subspace Clustering

It is an approach to identify clusters in different subsets of dimensions. In high dimensional datasets, usually, different clusters exist in different subsets of dimensions as a lot of dimensions are irrelevant. These methods have commonly been categorized in literature as bottom-up subspace clustering methods and top-down subspace clustering methods [24, 27, 29, 30].

2.1.3 Bottom-Up Subspace Clustering

CLIQUE [3] is regarded as the first bottom-up subspace clustering method. It uses grid based apriori approach to determine dense subspaces, then subspace clusters are found using disjunctive normal form (DNF) expression. It identifies the clusters of varying shapes in different dimensions; however, sometime it misses small but important clusters. Moreover, it is very sensitive to the two input parameters, grid size and density threshold, which largely effect quality of the clusters. ENCLUS [11] is very similar to CLIQUE except that it uses entropy measure to find the subspaces instead of density and coverage. MAFIA [15] is another extension of CLIQUE that uses adaptive size of grid; it improves its efficiency and the quality of clusters. Comparatively, MAFIA is efficient but it suffers with a drawback similar to CLIQUE; it also sometime misses small but important clusters at pruning stage. Chu et al. [12] propose DENCOS for determining the subspace clusters which uses DFP-tree to produce different threshold value of density in every stage of the algorithm to improve the quality of subspace clustering.

2.1.4 Top-Down Subspace Clustering

PROCLUS [2] is regarded as the first top-down subspace clustering method which is heavily based on the CLARANS [28]. It first consider a sample of the dataset, selects randomly a set of k medoids from the sample and process iteratively using average Manhattan segmental distance to find initial clusters. Then, it selects the relevant dimensions for every cluster based on the statistical expectation of small average distance. It is fast due to sampling but it creates only spherical clusters of fix sized subspace. ORCLUS [1] is an extension of PROCLUS; first, it assigns every

object to randomly selected clusters centers and selects the subspace for each cluster based on the smallest Eigen value of covariance matrix. Then, the nearest clusters having similar direction are merged together to form large clusters. However, sometimes it misses some small but meaningful clusters. PCKA [9], initially, determines dense regions of each dimension, where every point in the cluster are sufficiently closed to each other, next it identify and remove the outliers. Then, it obtains the projected clusters and their subspaces. Wang et al. [38] presents K-subspace clustering model that uses distance minimization function to produce clusters of different shapes like line, plane and ball based on the significant Eigen values. Kumar and Puri [32] present an extended version of Gustafson Kessel algorithm which modifies the objective function of projective clustering so that it automatically identify the relevant dimensions of every cluster. Gunnemann et al. [16] propose ASCLU that uses subspace clusters obtained by some subspace clustering method as input to produce different subspace clusters by realizing different views on the data by using different attributes. That is why, it is known as Alternative Subspace Clustering. However, better alternate subspace clusters in comparison to input clusters are not sure.

3 Proposed Method

Rajput et al. [33] propose a three phase method FAMCA to obtain clusters in high dimensional dataset. In first phase, referred as feature selection phase, it uses Fisher's criterion score to find relevant dimensions in the high dimensional dataset. In second phase, referred as centers initialization phase, it uses trimmed mean to select efficient initial clusters centers. Finally, it uses K-means to obtain final clusters in phase three, which is referred as refinement phase. The method follows a traditional approach of finding clusters in high dimensional dataset as all the clusters are obtained in the same subspace. However, finding all the clusters in one subset of few selected relevant dimensions is not justified as different clusters may exist in different subsets of dimensions, i.e., different dimensions may be relevant in different clusters. Here, we present a top-down subspace clustering method PROFIT (PROjective clustering algorithm based on FIsher score and Trimmed mean) which is an extension of FAMCA for the projected (subspace) clustering and works well with high dimensional dataset consisting of attributes in continuous variable domain. It works in four phases: sampling phase, initialization phase, dimension selection phase and refinement phase. The sampling phase uses data transformation method PCA and systematic sampling to obtain a good representative sample of the dataset. The initialization phase uses trimmed mean to find effective initial clusters centers. The dimension selection phase uses Fisher criterion score to select relevant dimensions of each cluster. Finally, the refinement phase uses K-means to produce subspace clusters in full dataset. The sequential steps of our proposed algorithm PROFIT are shown in Algorithm 2.

Algorithm 2: : PROFIT Algorithm

Input: N x D dataset; N is the number of objects, D is the number of dimensions in the dataset; S is the size of the sample, l is the percentage of trimming, d is the number of relevant dimensions in the clusters, K is the number of clusters.

Output: K projected clusters of dimension d, where d < D;

Sampling Phase

1. Transform the dataset into one dimensional data using PCA.
2. Sort the transformed one dimensional data in ascending order and select every ||N/S|| object from the dataset.
3. The set of objects selected in step 2, in the full dimensional dataset, form sample dataset.

Initialization Phase

4. For each dimension
   a. Sort sample data in ascending order and partition it in K equal parts.
   b. Compute mean and standard deviation of each part.

$$\mu_x = \frac{\sum_{i=1}^{n} X_i}{n}, \sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{(n-1)}}$$

   c. Compute Fisher Score.

$$F_t = \frac{(\mu_1 - \sum_{i=2}^{k} \mu_i)^2 + \ldots. + (\mu_j - \sum_{i=1 \cdot i \neq j}^{k} \mu_i)^2 \ldots. + (\mu_k - \sum_{i=1}^{k-1} \mu_i)^2}{\sum_{i=1}^{k1} \sigma_i^2}$$

5. Sort the sample dataset in ascending order indexing on the highest Fisher criterion Score dimension.
6. Partition the sorted sample dataset into K equal partitions and compute l percent Trimmed mean of each dimension in each partition. The set of obtained trimmed means in each partition makes initial clusters centers.

Dimension Selection Phase

7. Assign each object of the sample dataset to the nearest initial cluster center.
8. Compute the Fisher criterion score of each dimension in each cluster.
9. Select the d dimensions having highest fisher criterion score in each cluster.

Refinement Phase

10. Redefine initial cluster centers with respect to the selected (d) relevant dimensions in each cluster.
11. Assign each object of full dataset to the nearest cluster center.
12. Recompute center of every cluster.
13. Repeat steps 11 - 12 until cluster centers stabilizes.

The sampling phase, i.e., the first three steps, uses PCA to transform full dimensional dataset into one dimensional dataset and apply systematic sampling on sorted transformed dataset to obtain a good representative sample of the original dataset. The sampling reduces time and space complexity of the method. Here, we use PCA for data transformation as PCA is very simple and effective data transformation method and use systematic sampling method to minimize the bias in the sample. The initialization phase, i.e., steps 4–6, identify efficient initial clusters centers using trimmed mean. The trimmed mean is robust to outliers as it ignores a small percentage of the highest and lowest values of the data. Though, some other methods such

as arithmetic average, geometric mean, harmonic mean etc. are also available in the literature to determine the central tendency of the dataset, they are not as effective and robust as the trimmed means. Average is a simple and popular measure to identify the central location of dataset but it applies only on the normally distributed dataset and is very sensitive to the outliers as one bad data can move the average value away from the center of the rest of the data by an arbitrarily large distance. The geometric mean and harmonic mean are suitable for log normally distributed dataset and are also sensitive to the outliers. The trimmed mean is the mean of middle portion of dataset after trim; in our case we trim 10 % of the dimension of dataset, which changes only slightly if data has large perturbation to any value, hence it is more robust to outliers. The dimension selection phase, i.e., steps 7–9, find the relevant dimensions to every cluster based on the Fisher score. The Fisher score is used to determine a relevant subset of dimensions so that distance between the data objects in different clusters is as large as possible while distance between the data objects in the same clusters is as small as possible. Though, some other methods such as range, standard deviation, variance, mean absolute deviation etc. are also available in the literature to determine the dispersion in the dimensions of the dataset. Range is very easy measure of dispersion but it is very sensitive to the outliers because it computes the difference between the maximum and minimum value of each dimension. The standard deviation and variance are also sensitive to the outliers in the presence of bad data. The mean absolute deviation is also sensitive to outliers, however it does not move quite as much as the standard deviation or variance in response to bad data. On the other hand, Fisher score is a combination of mean and standard deviation which effectively computes the dispersion of the dimensions in the dataset. Fisher score is more robust to outliers in comparison to range, standard deviation, variance and mean absolute deviation. Finally, the refinement phase, i.e., steps 10–13, uses K-means to obtain final projected clusters.

4 Experimental Results

In this section, we show the clustering results obtained by our proposed method PROFIT on five real datasets and compare them with the results of other well - known top-down subspace clustering methods (PROCLUS, ORCLUS and PCKA) and our feature selection based non-subspace clustering method FAMCA [33]. The obtained results are verified by two well-known subspace clustering quality measures Jagota index and sum of square error, and Student's t-test to determine significant difference between clustering results. The real datasets and experiments with different input parameters to the clustering methods are described in the next subsections.

4.1 A. Description of Datasets

In this experiment, we use five real datasets Mammal's Milk Data (MMD),[1] Wisconsin Prognostic Breast Cancer (WPBC),[2] Heart Disease Data (HDD),[3] Image Segmentation Data (ISD)[4] and Reuters-21578 Corpus Text Dataset.[5] Mammal's milk data is a numerical dataset which contains 25 Mammal's with five ingredients water, protein, fat, lactose and ash in their milk. Here, mammals represent the objects and ingredients of their milk represent the dimensions in the dataset. Wisconsin Prognostic Breast Cancer (WPBC) dataset is also a numerical data, which consists of 198 breast cancer records and 34 different symptoms of breast cancer where each record represents a breast cancer case. The symptoms (dimensions) of this dataset have been computed from a digitized image of a fine needle aspirate (FNA) of a breast mass, which describe the characteristics of the cell nuclei present in the image. The Heart disease dataset contains 303 objects with 14 dimensions. Here, the dimensions represent different symptoms in heart disease as resting blood pressure, serum cholesterol, fasting blood sugar etc. The dataset classify the patients into five classes for the absence and presence of heart disease with varying severity. Image segmentation dataset is also a numerical dataset which contains 2310 objects with 19 dimensions, where objects were drawn randomly from a database of seven outdoor images such as brick face, sky, foliage, cement, window, path and grass.

The fifth dataset, Reuters-21578 corpus text dataset (reut-2-000.sgm), contains 925 documents in SGML (Standard Generalized Mark-up Language) format. Usually a large portion of the documents contain uninformative terms. These terms unnecessary increase the number of dimensions in the representational model and reduce the accuracy. Therefore, it is required to pre-process the documents in order to remove these non-descriptive terms. The pre-processing essentially includes stop words removal and stemming. Stop words[6] are common words, e.g., a, an, the, who, be, which are necessarily required to be removed as they carry no weightage for clustering but make a substantial part of document. Stemming[7] converts the morphological/derivationally related words into single root form. Next, the documents are tokenized using the punctuation and white spaces to identify the token boundaries. For meaningful clustering of the documents we select only those terms which are present in more than four documents and less than or equal to 200 documents. The next and most important step is term weighting, which represents significance of the terms with respect to documents. Though numerous terms weighting schemes have been proposed over the years, TF-IDF and its variant are most widely used.

[1] http://www.uni-koeln.de/themen/statistik/data/cluster/milk.dat
[2] http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wpbc.dat
[3] http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.dat
[4] http://archive.ics.uci.edu/ml/machine-learning-databases/image/segmentation.dat
[5] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
[6] http://www.fromzerotoseo.com/stopwords-remove/
[7] http://tartarus.org/ martin/PorterStemmer/

Table 1 Description of datasets

Datasets	No. of objects	No. of dimensions
Mammal's milk data (MMD)	25	5
Wisconsin prognostic breast cancer (WPBC)	198	34
Heart disease data (HDD)	303	14
Image segmentation data (ISD)	2310	19
Reuters-21578 corpus text dataset	925	1958

Here, the vectors representing the documents are constructed based on the frequency of occurrence of the terms with respect to documents. The term weighting used in this article is summarized below in Eq. (1).

$$TF - IDF(i,d) = (\sqrt{[wf_{id}]})\, ln\left(\frac{N}{\mathrm{d}f_i}\right) \begin{cases} \text{if } wf_{id} \geq 1 \\ 0 \quad \text{otherwise} \end{cases} \tag{1}$$

Here, TF-IDF(i, d) is the IDF of ith word in the jth document, f_{id} is the word frequency (i.e., frequency of the ith word in the jth document), N is the total number of documents in the corpus and df_i is the document frequency of ith word (i.e., the number of documents in the corpus that include the ith word). The $\sqrt{[wf_{id}]}$ is introduced in place of the log transform term of word frequencies (1+ ln (wf_{id})) in the usual TF-IDF expression, to reduce the *dampening* effect of the log function on the word frequencies. In our case, the final transformed dataset contains 925 objects with 1958 dimensions. The description of the datasets such number of objects and number of dimensions is presented in Table 1.

4.2 B. Subspace Cluster Quality Measures

Here, we describe some well-known quality measures for quantitative/qualitative comparison of the results.

Jagota Index (Q) [19]: Jagota index is a very popular and well-known quality measure of clusters. It measures homogeneity, tightness and compactness of objects in the cluster. It is defined in Eq. (2), where $|C_i|$ defines the number of data objects in the cluster i, k denotes the number of clusters in the data, μ_i represents the center of ith cluster, x indicates a object in the cluster and $d(x, \mu_i, D)$ defines the distance between object x and the cluster center μ within the subspace of cluster i. A small value of the index indicates a better quality cluster.

$$Q = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{x \in C_i} d(x, \mu_i, D). \tag{2}$$

Sum of Squared Error (SSE) [37]: Sum of squared error is another popular criterion for measuring the homogeneity of the objects in the cluster. It is defined in Eq. (3),

where N, k, s and m represents the number of data objects, number of clusters, cluster's centre and the data object respectively. Here also, a smaller value of the measure is indicative to a better quality cluster.

$$\text{SSE} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n(s_i)} \parallel m_{i,j} - \overline{s_i} \parallel^2}{N}. \tag{3}$$

Student's t-test [31]: The t-test is a powerful statistic that enables to determine whether the differences obtained between two groups is statistically significant or not. It is given in Eq. (4), where $\overline{X_1}$ and $\overline{X_2}$ represent the average value of objects in Group_1 and Group_2 respectively, var_1 and var_2 denotes the variance of Group_1 and Group_2 respectively, and n_1 and n_2 indicate the number of objects in Group_1 and Group_2 respectively. This t-value is negative if the mean of Group_1 is less than the mean of Group_2 and vice versa.

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{var_1}{n_1 - 1} + \frac{var_2}{n_2 - 1}}}. \tag{4}$$

After computing the t-value of two groups, we look at the t-value in *The Table of Critical Values for Student's t-test* which indicates whether the two groups are significantly different or not. This value is identified based on the two parameters such as Alpha Level and Degrees of Freedom. Generally, in many social and educational researches, alpha level is set at 0.05 according to the "rule of thumb". This means that 5 % of the time we find the statistically significant difference between the means even if there is none. The degree of freedom is determined by the sum of elements in both groups minus two, i.e., $(n_1 + n_2 - 2)$. If the difference between computed t-value and critical value in the table is large enough then we conclude that the difference between two groups is significant.

4.3 C. Subspace Clustering with Two Relevant Dimensions

In this experiment, we consider two relevant dimensions in each dataset for every subspace cluster, 10 % sample size for every dataset and assume that Mammal's Milk, Wisconsin Prognostic Breast Cancer, Heart Disease, Image Segmentation and Reuters-21578 Corpus Text Datasets contain 4, 10, 5, 7 and 6 clusters respectively. The computed quality measure values for the obtained clusters are shown in Table 2. We observe that the proposed method PROFIT obtains better clusters in comparison to FAMCA and other well-known top-down subspace clustering methods PROCLUS, ORCLUS and PCKA. As all the values for PROFIT are comparatively small, it is clear winner in this experiment.

Table 2 Qualitative results of subspace clustering with two relevant dimensions

	MMD		WPBC		HDD		ISD		Reuters-21578	
Method	Q	SSE	Q	SSE	Q	SSE	Q	SSE	Q	SSE
FAMCA	8.96	196.08	911.65	1626850	143.87	28712.7	62.96	249113	1273.28	634428
PROCLUS	8.79	185.43	538.41	15629.79	128.48	27475.8	124.62	236385	492.12	825576
ORCLUS	10.45	172.48	667.98	12080.85	104.13	25847.2	238.77	3137610	572.83	1987923
PCKA	9.9	143.04	419.34	24534.84	234.19	81142.86	127.07	8879403	533.56	326755
PROFIT	**8.37**	**137.26**	**333.93**	**4017.07**	**84.18**	**25005.6**	**34.96**	**195509**	**428.42**	**231873**

Table 3 Qualitative results of subspace clustering with five relevant dimensions

	MMD		WPBC		HDD		ISD		Reuters-21578	
Method	Q	SSE	Q	SSE	Q	SSE	Q	SSE	Q	SSE
FAMCA	8.14	975	970	421080	655	75787	70.6	971324	438.7	489857
PROCLUS	9.05	**546**	**457**	910507	678	74381	31.18	823538	795.2	445643
ORCLUS	12.7	964	800	709202	849	**17812**	46.32	694871	765.5	646392
PCKA	9.13	957	485	950945	934	65505	**24.98**	3171712	**186.9**	709455
PROFIT	**6.32**	785	527	**141009**	**357**	39262	27.69	**344950**	381.6	**75472**

Table 4 Computed t-values ($d = 5$)

Methods	t-value
PROFIT-FAMCA	−1.1701
PROFIT-PROCLUS	−1.2995
PROFIT-ORCLUS	−1.2695
PROFIT-PCKA	−1.2765

4.4 D. Subspace Clustering with Five Relevant Dimensions

In this experiment, we assume five relevant dimensions in each dataset for every subspace cluster, sample size is 10 % for each dataset and assume that Mammal's Milk, Wisconsin Prognostic Breast Cancer, Heart Disease, Image Segmentation and Reuters-21578 Corpus Text datasets contain number of clusters 2, 4, 3, 5 and 4 respectively. The computed quality measure values for the obtained clusters are shown in Table 3. We observe that the proposed method PROFIT obtains better subspace clusters in most but not in all the cases. PROCLUS performs comparatively better in Mammal's Milk Data and Wisconsin Prognostic Breast Cancer data based on the Q and SSE index respectively. ORCLUS performs better in Heart Disease Data based on the SSE index whereas PCKA performs better in Image Segmentation Data and Reuters-21578 Corpus Text data based on the Q index.

Therefore, we apply Student's t-test for determining the significant difference between the clustering results of quality measure obtained by FAMCA, PROCLUS, ORCLUS, PCKA and our proposed method PROFIT. The computed t-values of PROFIT-FAMCA, PROFIT-PROCLUS, PROFIT-ORCLUS and PROFIT-PCKA are shown in Table 4. In this case, the degree of freedom is 18 as the sum of elements in two groups, i.e., $n_1 + n_2$, is 20. The two tailed alpha level is set to 0.05 as a "rule of thumb". The critical value in the student's t-test table based on these parameters is 2.101. The computed t-values are very small in comparison to the t-value in critical table and negative. Therefore, we claim that our proposed technique PROFIT produces better subspace clustering in comparison to other competitive methods.

Table 5 Qualitative results of subspace clustering with ten relevant dimensions

	WPBC		HDD		ISD		Reuters-21578	
Method	Q	SSE	Q	SSE	Q	SSE	Q	SSE
FAMCA	276	496854	751	95930	84.07	350890	351.7	842858
PROCLUS	679	954927	**255**	54742	25.43	196236	830.8	757342
ORCLUS	655	384704	506	**13826**	81.43	245411	585.3	782537
PCKA	162	548563	699	14953	**24.35**	613460	549.7	**380483**
PROFIT	**119**	**292368**	489	25735	32.53	**47383**	**172.9**	403278

Table 6 Computed t-values ($d = 10$)

Methods	t-value
PROFIT-FAMCA	−1.0319
PROFIT-PROCLUS	−1.1460
PROFIT-ORCLUS	−1.1196
PROFIT-PCKA	−1.1258

4.5 E. Subspace Clustering with Ten Relevant Dimensions

In this experiment, we assume ten relevant dimensions in each dataset for every subspace clusters, sample size is 20 % for each datasets and assume that the Wisconsin Prognostic Breast Cancer, Heart Disease, Image Segmentation and Reuters-21578 Corpus Text datasets contain number of clusters 6, 7, 9 and 8 respectively. The computed quality measure values for the obtained clusters are shown in Table 5. We observe that the performance of our proposed method PROFIT is better in most quality measures but not in all; PROCLUS and ORCLUS produce better results in Heart Disease Data based on Q and SSE index respectively whereas PCKA produces better results in Image Segmentation Data and Reuters-21578 Corpus Text data based on the Q and SSE index respectively.

Therefore, we apply Student's t-test for determining the significant difference between the clustering results of quality measure obtained by FAMCA, PROCLUS, ORCLUS, PCKA and our proposed method PROFIT. The computed t-values of PROFIT-FAMCA, PROFIT-PROCLUS, PROFIT-ORCLUS and PROFIT-PCKA are shown in Table 6. In this case, the degree of freedom is 14 as the sum of elements in two groups, i.e., $n_1 + n_2$, is 16. The two tailed alpha level is set to 0.05 as a "rule of thumb". The critical value in the student's t-test table based on these parameters is 2.145. The computed t-values are very small in comparison to the t-value in critical table and negative. Therefore, we claim that our proposed method PROFIT produces better subspace clustering in comparison to other competitive methods.

5 Conclusion

A meaningful clustering in the high dimensional dataset is a challenging issue as the curse of dimensionality plays an important role. Traditionally, the researchers have applied dimension reduction/selection methods to project high dimensional datasets onto lower dimensions to combat the ill effects of the curse of dimensionality. However, it may not be appropriate as different clusters may exist in different subset of dimensions; it is referred as subspace clustering. The subspace clustering methods which find relevant clusters in different subsets of dimensions are broadly classified as top-down and bottom-up subspace clustering methods. In this article, we propose an algorithm PROFIT (PROjective clustering algorithm based on FIsher score and Trimmed mean) which belongs to the class of top-down subspace clustering methods and works well with the high dimensional dataset consisting of attributes in continuous variable domain. The PROFIT works in four phases: sampling phase, initialization phase, dimension selection phase and refinement phase. We apply PROFIT on five real datasets with different input parameters and compare the results with three prominent and well-known top-down subspace clustering methods (PROCLUS, ORCLUS and PCKA) and one (non-subspace) feature selection based clustering method FAMCA. The results are obtained using different input parameters and are subject to two well-known subspace clustering quality measures (Jagota index and sum of squared error) and Student's *t*-test to show the robustness and effectiveness of our proposed method PROFIT to the competitive methods.

However, we realize that the proposed method inherits the common pitfalls of top-down subspace clustering methods, i.e., sensitive to the input parameters like number of clusters, size of the subspace and sample size. Moreover, the obtained clusters are hyper-spherical in fixed size subspace. On the positive side, these methods are fast, because of sampling, in comparison to bottom-up subspace clustering methods which also inherit common pitfalls as grid size and density threshold parameters. As a future work, the authors aim to develop a subspace clustering method which determines number of clusters on its own.

References

1. Aggarwal, C., Yu, P.: Finding generalized projected clusters in high dimensional spaces. In: ACM SIGMOD International Conference on Management of Data, pp. 70–81. ACM (2000)
2. Aggarwal, C., Wolf, J., Yu, P., Procopiuc, C., Park, J.: Fast algorithms for projected clustering. ACM SIGMOD Record **28**(2), 61–72 (1999)
3. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: ACM SIGMOD International Conference on Management of Data, pp. 94–105. ACM Press (1998)
4. Andrews, H., Patterson, C.: Singular value decompositions and digital image processing. IEEE Trans. Acoust. Speech Signal Process **24**(1), 26–53 (1976)
5. Apolloni, B., Bassis, S., Brega, A.: Feature selection via boolean independent component analysis. Inf. Sci. **179**(22), 3815–3831 (2009)

6. Arai, K., Barakbah, A.: Hierarchical *k*-means: An algorithm for centroids initialization for *k*-means. Rep. Fac. Sci. Eng. **36**(1), 25–31 (2007)
7. Barakbah, A., Kiyoki, Y.: A pillar algorithm for *k*-means optimization by distance maximization for initial centroid designation. In: Computational Intelligence and Data Mining, 2009. IEEE Symposium on CIDM'09, pp. 61–68. IEEE (2009)
8. Berkhin, P.: A survey of clustering data mining techniques. Technical Report (2002)
9. Bouguessa, M., Wang, S.: Mining projected clusters in high-dimensional spaces. IEEE Trans. Knowl. Data Eng. **21**(4), 507–522 (2009)
10. Celebi, M.: Effective initialization of *k*-means for color quantization. In: 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 1649–1652. IEEE (2009)
11. Cheng, C., Fu, A., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 84–93. ACM (1999)
12. Chu, Y., Huang, J., Chuang, K., Yang, D., Chen, M.: Density conscious subspace clustering for high-dimensional data.. IEEE Trans. Knowl. Data Eng. **22**(1), 16–30 (2010)
13. Ding, C., He, X.: *K*-means clustering via principal component analysis. In: Proceedings of the twenty-first International Conference on Machine Learning, pp. 225–232. ACM (2004)
14. Gheyas, I., Smith, L.: Feature subset selection in large dimensionality domains. Pattern Recognit. **43**(1), 5–13 (2010)
15. Goil, S., Nagesh, H., Choudhary, A.: Mafia: Efficient and scalable subspace clustering for very large data sets. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 443–452 (1999)
16. Günnemann, S., Färber, I., Müller, E., Seidl, T.: Asclu: Alternative subspace clustering. In: In MultiClust at KDD. Citeseer (2010)
17. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann (2001)
18. Hu, Q., Che, X., Zhang, L., Yu, D.: Feature evaluation and selection based on neighborhood soft margin. Neurocomputing **73**(10), 2114–2124 (2010)
19. Jagota, A.: Novelty detection on a very large number of memories stored in a hopfield-style network. In: IJCNN-91-Seattle International Joint Conference on Neural Networks, 1991, vol. 2, pp. 905–. IEEE (1991)
20. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Inc. (1988)
21. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. ACM Computing Surveys (CSUR) **31**(3), 264–323 (1999)
22. Kabir, M., Islam, M., et al.: A new wrapper feature selection approach using neural network. Neurocomputing **73**(16), 3273–3283 (2010)
23. Khan, S., Ahmad, A.: Cluster center initialization algorithm for *k*-means clustering. Pattern Recognit. Lett. **25**(11), 1293–1302 (2004)
24. Kriegel, H., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. Knowledge Discov. Data (TKDD) **3**(1), 1–58 (2009)
25. Kruskal, J., Wish, M.: Multidimensional Scaling, Quantitative Applications in the Social Sciences. Beverly Hills (1978)
26. Liu, Y., Liu, Y., Chan, K.: Dimensionality reduction for heterogeneous dataset in rushes editing. Pattern Recognit. **42**(2), 229–242 (2009)
27. Moise, G., Zimek, A., Kröger, P., Kriegel, H., Sander, J.: Subspace and projected clustering: Experimental evaluation and analysis. Knowl. Inf. Syst. **21**(3), 299–326 (2009)
28. Ng, R., Han, J.: Clarans: A method for clustering objects for spatial data mining. IEEE Trans. Knowl. Data Eng. **14**(5), 1003–1016 (2002)
29. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: A review. ACM SIGKDD Explorations Newsletter. **6**(1), 90–105 (2004)

30. Parsons, L., Haque, E., Liu, H., et al.: Evaluating subspace clustering algorithms. In: Workshop on Clustering High Dimensional Data and its Applications, SIAM International Conference on Data Mining, pp. 48–56. Citeseer (2004)
31. Pearson, E.: Studies in the history of probability and statistics. XX: Some early correspondence between W.S. Gosset, R.A. Fisher and Karl Pearson, with notes and comments. Biometrika **55**(3), 445–457 (1968)
32. Puri, C., Kumar, N.: Projected Gustafson-Kessel clustering algorithm and its convergence. Trans. on Rough Sets XIV, 159–182 (2011)
33. Rajput, D., Singh, P., Bhattacharya, M.: An efficient technique for clustering high dimensional data set. In: 10th International Conference on Information and Knowledge Engineering. pp. 434–440. WASET, USA (July 2011)
34. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
35. Sugiyama, M., Kawanabe, M., Chui, P.: Dimensionality reduction for density ratio estimation in high-dimensional spaces. Neural Netw. **23**(1), 44–59 (2010)
36. Tenenbaum, J., De Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
37. Veenman, C., Reinders, M., Backer, E.: A maximum variance cluster algorithm. IEEE Trans. Patt. Anal. Machine Intell. **24**(9), 1273–1280 (2002)
38. Wang, D., Ding, C., Li, T.: *K*-subspace clustering. Machine Learn. Knowl. Discov. Databases 506–521 (2009)

Multi-label Classification with a Constrained Minimum Cut Model

Guangzhi Qu, Ishwar Sethi, Craig Hartrick and Hui Zhang

Abstract Multi-label classification has received more attention recently in the fields of data mining and machine learning. Though many approaches have been proposed, the critical issue of how to combine single labels to form a multi-label remains challenging. In this work, we propose a novel multi-label classification approach that each label is represented by two exclusive events: the label is selected or not selected. Then a weighted graph is used to represent all the events and their correlations. The multi-label learning is transformed into finding a constrained minimum cut of the weighted graph. In the experiments, we compare the proposed approach with the state-of-the-art multi-label classifier ML-KNN, and the results show that the new approach is efficient in terms of all the popular metrics used to evaluate multi-label classification performance.

1 Introduction

Compared with conventional single-label classification, multi-label classification is more general in practice, since it allows one instance to have more than one label simultaneously. For example, a movie can be tagged with film genres *action*, *animation*, *drama* and *comedy* the same time. Because of its generality, multi-label classification problem has attracted much attention from researchers. Existing approaches on multi-label classification can be categorized into two main groups:

G. Qu (✉) · I. Sethi
Computer Science and Engineering Department, Oakland University, Rochester, MI 48309, USA
e-mail: gqu@oakland.edu

I. Sethi
e-mail: isethi@oakland.edu

C. Hartrick
Anesthesiology Research, School of Medicine, Oakland University, Rochester, MI 48309, USA
e-mail: chartrick@beaumont.edu

H. Zhang
State Key Laboratory of Software Development Environment, School of Computer Science, Beihang University, Beijing 100191, China
e-mail: hzhang@nlsde.buaa.edu.cn

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_5

program transformation method and algorithm adaptation method [23]. Program transformation method transforms a multi-label classification problem into multiple single-label classification problems. There are two typical transformation methods, *Binary Relevance* (BR) and *Label Powerset* (LP) [22, 24]. In BR, labels are assumed to be independent [8, 9, 12, 14, 21, 27]. A single-label classifier is built for each individual label. During testing, confidence value is calculated for each label and compared with a threshold to determine the relevance of the label. In LP-based methods, multi-labels are considered as new single labels. An apparent problem of this kind of approaches is that some generated labels are supported by very few data instances. Be different from the program transformation method, algorithm adaptation method extends traditional single-label classification approaches for handling multi-label classification directly. Many single-label classification methods have been extended, such as logistic regression [2], K-nearest neighbors (KNN) [20, 32], decision trees [28], and support vector machine (SVM) [6, 7, 10, 17]. Ensemble methods have also been applied for multi-label classification [4, 15, 16, 18, 26]. Though there exist so many efforts, two issues remain critical in improving multi-label classification performance.

The first issue is how to deal with the correlations among labels. A unique characteristic of multi-label classification is that there may exist correlations among labels [5]. Some previous works have shown that considering the correlations has positive impact on improving the multi-label classification performance [2, 9]. Particularly, Cheng and Hüllermeier assumed that the correlations among labels do not change in different contexts [2]. Ghamrawi and McCallum presented a different view on the label correlations [9]. In their CMLF model, a tri-tuple $\langle feature, label_1, label_2 \rangle$ is defined to indicate the relationship between features and label pairs, and the correlation between two labels is closely related with the context (feature values). For example, when consider four film labels *action*, *animation*, *drama* and *comedy*, for the movie ***megamind***, there exists more strong correlation between labels $animation$, $action$ and $comedy$. While on the other hand, for the movie ***unknown***, then the correlation between $action$ and $drama$ will lead the other combinations. In our approach, we utilize the conditional probability of features and labels for considering the context's impact on label correlations.

Another challenge in multi-label classification is what strategy will be used to select the final labels for a data instance. Threshold based filtering has been widely used in the literature. For example, *Label Ranking* (LR) method compares the confidence value of each label to the customized threshold in making the decision on whether the label will be selected into the final label set. Since the existence of the correlations among labels, when we consider the selected labels, we need to evaluate the merit of the label set rather than each individual label.

In this article, we present a novel approach to transforming the multi-label classification problem into a constrained minimum cut problem in that the merit of selected labels is considered from a global optimization perspective. We define a pair of events associated with each label l: positive event l^+ to denote that the label will be chosen; negative event l^- to denote that the label won't be chosen. Naturally, the

two events (*chosen*, *not chosen*) for an individual label are exclusive. For the classification of a testing data instance d, the multi-label result corresponds to positive events. Using the previous example, there are eight events in total. A multi-label of $\{animation, action\}$ means that labels *animation* and *action* were chosen, while the labels *drama* and *comedy* were not. The multi-label classification results divide the total events into two sets: one is $\{animation^+, action^+, comedy^-, drama^-\}$, the other is $\{animation^-, action^-, comedy^+, drama^+\}$.

In our approach, a weighted graph is used to represent the relationships among all the *chosen* and *not chosen* events, which are represented by the vertices and the weight of an edges quantifies the degree of correlation between the two events. The multi-label classification problem then can be converted into a partition problem on all the possible events. In this way, all candidate events are partitioned into two groups: the occurrence event group and the non-occurrence event group, and with the constraint that two events associated with an individual label will appear in different groups. The desired situation is that these two groups have low external connections, and the occurrence event group has high internal connections. Based on this assumption, we propose a multi-label classification approach with a constrained minimum cut. Courant-Fischer Theorem is used to solve the constrained minimum cut problem, and the group with higher internal connections will be selected to determine what multi-label will be used for the testing data instance. To evaluate the performance of proposed approach, we compare it with ML-KNN [32], which is a state-of-the-art multi-label classification method. The experimental results show that our approach is efficient in terms of all the popular metrics used to evaluate multi-label classification performance.

The rest of article is organized as follows. Related work is presented in Sect. 2. Section 3 describes the detailed methodology of our proposed approach. The experimental design and results are given in Sect. 4, and finally Sect. 5 concludes the article.

2 Related Work

In this section, we first review different multi-label classification methods based on their underlying implementation models, viz., *probabilistic model*, *logistic regression*, *Bayesian*, *K-nearest neighbors*, and *decision tree*.

In *probabilistic model* based multi-label classification methods, each label is generally assumed to follow certain distribution of the features. However, the label correlations are treated quite differently. Some work assume the independency among all the labels, and the multi-label is selected according to the order of individual confidence values of the labels [31]. Other research work utilize a mixture model to combine single label models [9, 12, 14, 21, 27, 29]. In [12], McCallum proposed a Bayesian-based model to represent the relationship between a label and data instance features in that every document is expressed by the mixture model, which guides the creation of the label and its weight distributions. In light of Bayesian rule, the

classification goal is to find out the original label distribution:

$$\mathbf{c}^{+} \approx \arg\max_{\mathbf{c}} P(\mathbf{c}) \prod_{w \in d} \sum_{c \in C} \lambda_c^{(\mathbf{c})} P(w|c), \tag{1}$$

where d denotes a testing document, $\mathbf{c}$ is a label distribution, and $\lambda_c^{(\mathbf{c})}$ is the mixture weight of label c in mixture weight distribution $\lambda^{(\mathbf{c})}$. They also assumed that features are independent, and so are labels.

Logistic regression is another popular technique applied for multi-label classification. Cheng and Hüllermeier presented to combine both instanced-based learning and logistic regression [2]. The instanced-based learning method—K-Nearest Neighbors (KNN), is used to form candidate labels, where they improved KNN by weighting votes through the similarities between an instance and its neighbors. Logistic regression is employed to combine the correlations among labels and calculate the final occurrence probability of each label. In another work [8], Fujino and Isozaki built the binary classifier of each label with logistic regression.

In [31], Zhang et al. proposed a *Naive Bayes* based approach for multi-label classification. In their method, one assumption is that all the features follow Gaussian distribution and a feature selection procedure was conducted to meet Naive Bayes assumption that features are independent.

KNN has been extended for handling multi-label classification in many ways. Wpyromitros et al. computed each label's confidence through KNN, and the multi-label is formed according to these confidence values [20]. Brinker and Hüllermeier used KNN-based binary relevance method for multi-label ranking [1]. Zhang and Zhou proposed ML-KNN to combine both KNN and Bayesian rule [32]. ML-KNN's objective function is:

$$\mathbf{y_t}(l) = \arg\max_{b \in \{0,1\}} P(H_b^l | E^l_{C_t(l)}), l \in Y, \tag{2}$$

Where $\mathbf{C_t}(l)$ denotes the number of t's neighbors having label l, and Y is the entire label set. H_b^l is the event that whether an instance t is labeled l. If b is equal to 1, it means that t has label l; otherwise, l is not assigned to t. Since b has only two options, 0 and 1, then if $P(H_1^l | E^l_{\mathbf{c_t}(l)}) \geq P(H_1^l | E^l_{\mathbf{c_t}(l)})$, l is assigned to t; otherwise, t is equal to 0. According to their experiments, ML-KNN has better performance than Boostexter 18, multi-label decision tree ADTboost.MH [4] and the multi-label kernel method Rank-SVM [6].

Celine Vens et al. analyzed the techniques of decision trees for hierarchical multi-label classification [28]. They classified these techniques into three sorts. One is single-label classification (SC) approach, where each class has one binary classifier. The second is hierarchical single-label classifier (HSC). The third approach offers the labels of one example at once and then uses the hierarchical category to give final multiple labels, named as HMC, which works best according to their experiments. Amanda Clare and Ross D. King improved C4.5 by modifying the formula of *entropy* to deal with multi-label classification [3].

In this work, we use a weighted graph to represent labels and their relations. Secondly, we propose to use a minimum cut model for multi-label classification problem, which makes it is possible to find an optimal multi-label.

3 Methodology

3.1 Multi-Label Minimum Cut Formulation

Let $G = (V, E, W)$ be an undirected weighted graph, where V is a vertex set, E is an edge set, and W is the edge weight set of G, where w_{ij} indicates the weight of edge e_{ij}, and $w_{ii} = 0$. In this context, a cut is a partition of the vertices of a graph into two disjoint subsets. The cut-set of the cut is the set of edges whose end points are in different subsets of the partition. Edges are said to be crossing the cut if they are in the cut-set. In a weighted graph, the weight of a cut is defined by the sum of the weights of the edges crossing the cut [30]. The minimum cut aims to minimize the weight of a cut, whose objective function is:

$$\arg\min\Big(\sum_{v_i \in A, v_j \in B} w_{ij}\Big). \tag{3}$$

In Eq. 3, A, B are two disjoint subsets of a partition, which means $A \cup B = V$ and $A \cap B = \emptyset$ 11. According to the end points of an edge, we classify all edges into three groups: $A - A$, $B - B$ and $A - B$ (the cut-set):

$$\begin{aligned} S &= \sum_{v_i, v_j \in V, v_i \neq v_j} w_{ij} \\ &= \sum_{v_i \in A, v_j \in B} w_{ij} + \sum_{v_i \in A, v_j \in A} w_{ij} + \sum_{v_i \in B, v_j \in B} w_{ij}. \end{aligned} \tag{4}$$

For a given graph G, S is a constant. Therefore, we can expand the objective function in Eq. 3 as follows.

$$\begin{aligned} &\arg\min\Big(\sum_{v_i \in A, v_j \in B} w_{ij}\Big) \\ &= \arg\min\Big(S - \Big(\sum_{v_i \in A, v_j \in A} w_{ij} + \sum_{v_i \in B, v_j \in B} w_{ij}\Big)\Big) \\ &= S + \arg\min\Big(-1 \times \Big(\sum_{v_i \in A, v_j \in A} w_{ij} + \sum_{v_i \in B, v_j \in B} w_{ij}\Big)\Big) \\ &= S - \arg\max\Big(\sum_{v_i \in A, v_j \in A} w_{ij} + \sum_{v_i \in B, v_j \in B} w_{ij}\Big). \end{aligned} \tag{5}$$

Since S is a constant then we the following two equivalent optimization problems.

$$\arg\min\Big(\sum_{v_i \in A, v_j \in B} w_{ij}\Big)$$

$$\Leftrightarrow \arg\max\Big(\sum_{v_i\in A, v_j\in A} w_{ij} + \sum_{v_i\in B, v_j\in B} w_{ij}\Big). \tag{6}$$

Combine Eqs. 5 and 6, we will have the following:

$$\arg\min\Big(\sum_{v_i\in A, v_j\in B} w_{ij}\Big) \tag{7}$$

$$\Leftrightarrow \arg\max\Big(\sum_{v_i, v_j\in A} w_{ij} + \sum_{v_i, v_j\in B} w_{ij} - \sum_{v_i\in A, v_j\in B} w_{ij}\Big).$$

Let X be the indicator vector to presentation the partition of A, B. The element of X corresponds a vertex in the graph, and its value indicates the partition assignment of this vertex. If the value is equal to $+1$, it means the corresponding vertex is assigned to set A, and if the value is -1, then the corresponding vertex is assigned to set B. Specifically, if v_i and v_j are in the same group, then $x_i x_j$ is equal to $+1$; otherwise, its value is -1. With this representation, the right side of Eq. 7 can be rewritten into:

$$\arg\max \frac{X^T W X}{X^T X}$$
$$\text{s.t. } X \in \{-1, 1\}^{|V|}. \tag{8}$$

In Eq. 8, $X^T X$ is a constant, whose value is equal to $|V|$. $X^T W X$ is equal to $\sum_{i,j} w_{ij} x_i x_j$.

3.2 Constrained Minimum Cut

We define a constraint c as a two-tuple $(\alpha(c), \beta(c))$, where $\alpha(c)$ and $\beta(c)$ are two vertex sets. $\alpha(c)$ means the vertices with the value of $+1$ in X, and $\beta(c)$ includes the vertices having the value of -1 in X. In other words, all vertices in $\alpha(c)$ should be in the same vertex set, and all vertices in $\beta(c)$ should be in the other vertex set. The constraint is that $\alpha(c)$ and $\beta(c)$ have the same number of elements. Recall from the introduction part, for each label there are two events associated, we use constraint c to explicitly to prohibit the occurrence of them together. To combine all constraints for all the labels, we form a *constraint matrix*, denoted as M, where each row indicates a constraint and each column corresponds to one label. $M(i, j)$ denotes the value of label v_j in constraint c_i, and it is defined as:

$$M(i,j) = \begin{cases} 1 & \text{if } v_j \in \alpha(c_i) \cup \beta(c_i) \\ 0 & \text{Others} \end{cases}. \tag{9}$$

Add the constraint matrix M to Eq. 8, then we can formulate the multi-label classification problem into a constrained minimum cut system as:

$$\arg\max \frac{X^T W X}{X^T X}$$

$$\text{s.t. } MX = \mathbf{0}$$
$$X \in \{-1, 1\}^{|V|}. \tag{10}$$

In this system, the objective function is in non-linear form, which is difficult to solve with linear solvers. On the other side, there are special characterics with this system. One is that W is a n-by-n matrix and it is symmetric; the second is the objective function has the same format of *Rayleigh quotient* [13]. In order to solve the system, we proposed to eliminate the constraint of $MX = \mathbf{0}$ with the help of matrix kernel properties and solve the system utilizing *Courant-Fischer Theorem* [19]. To be self-contained and complete, we describe briefly the Courant-Fischer Theorem and the property of matrix kernel in the following.

Theorem 1 (***Courant-Fischer Theorem***) *Let W be an n-by-n symmetric matrix and let $1 \leq k \leq n$, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of W, then*

$$\lambda_k = \min_{S\, of\ dim\ k} \max_{x \in S} \frac{x^T W x}{x^T x} \tag{11}$$

and

$$\lambda_k = \min_{S\, of\ dim\ n-k-1} \max_{x \in S} \frac{x^T W x}{x^T x}. \tag{12}$$

Property 1 *If MX=0, and K is M's normalized kernel, then*

$$K^T K = \mathbf{1}. \tag{13}$$

Moreover, there exists a matrix Y, which satisfies

$$X = KY. \tag{14}$$

As stated in Theorem 1, W's biggest eigenvalue(λ_n) is the maximum value of $\frac{x^T W x}{x^T x}$, and x is the corresponding eigenvector of W.

Apply Eqs. 13 and 14 to the objective function in Eq. 10, we have

$$\begin{aligned}
&\arg\max \frac{X^T W X}{X^T X} \\
&= \arg\max \frac{(KY)^T W (KY)}{(KY)^T (KY)} \\
&= \arg\max \frac{Y^T K^T W K Y}{Y^T K^T K Y} \\
&= \arg\max \frac{Y^T (K^T W K) Y}{Y^T (K^T K) Y} \\
&= \arg\max \frac{Y^T (K^T W K) Y}{Y^T \mathbf{1} Y}
\end{aligned}$$

$$= \arg\max \frac{Y^T(K^T W K)Y}{Y^T Y}. \tag{15}$$

Relax X as a real value vector, and then we get a new system:

$$\arg\max \frac{Y^T(K^T W K)Y}{Y^T Y}$$
$$\text{s.t. } Y \in \mathbb{R}^{|V|}. \tag{16}$$

Now, Eq. 16 has the standard format of the system given in Theorem 1, so we can solve Eq. 16 in light of Theorem 1. In this system, we only care the maximum value, so our solution is the biggest eigenvector of $K^T W K$.

In conclusion, the core part of the proposed multi-label classification method can be described in Algorithm 1.

Algorithm 3: Constrained Minimum Cut

1: **Input**:
W - correlation matrix among vertices;
M - constraint matrix.
2: **Output**:
X - vertex assignment vector.
3: **Process**:
4: Get M's kernel K;
5: $Y \leftarrow$ the eigenvector according to the biggest eigenvalue of $K^T W K$;
6: $X^* = KY$;
7: Form X by setting the items in X^* with the biggest $\lfloor |V|/2 \rfloor$ values as $+1$, and all the others as -1.

In handling multi-label classification, we limit a constraint c's two components, $\alpha(c)$ and $\beta(c)$, can only have one element, i.e. $\forall c \in M, |\alpha(c)| = |\beta(c)| = 1$. In this specified system, each label will be represented by two vertices, and X represents the classification result. One issue about the result is that even we solve the system to acquire X. It does not specify whether the vertices in the $+1$ group are the resulting labels or the vertices in the -1 group. So we have to judge which group is better to represent the multi-label, which will be discussed in Sect. 3.4. Before that, we need to build a weighted graph for representing the correlations among label events, which will be described in Sect. 3.3.

3.3 Weighted Label-Graph Construction

In our approach, we assume there exist dependency among labels. A weighted graph will be deployed to represent labels and their correlations. In this graph, we use two vertices to denote each label and each edge indicates the correlation between its two end-point events. In building the weighted graph, we consider three aspects: weighting each individual label, computing label correlations and normalization.

The normalization process take consideration of both label importance values and original label correlations.

Let $F = \{f_1, f_2, \ldots, f_t\}(t \in \mathbb{R})$ be a feature set, $d = (d_1, d_2, \ldots, d_n) \in F^n(n \in \mathbb{R})$ be one instance, and $L = \{l_1, l_2, \ldots, l_m\}(m \in \mathbb{R})$ be a label set. The importance of selecting a label l_i ($l_i \in L$) denoted as ($h_{l_i^+}$) is calculated by Bayesian rule:

$$\begin{aligned} h_{l_i^+} &= P(l_i|d) \\ &= \frac{P(l_i, d)}{P(d)} \\ &= \frac{p(l_i)P(d|l_i)}{P(d)} \\ &= \frac{P(l_i)}{P(d)} \times P((d_1, d_2, \ldots, d_n)|l_i) \\ &= \frac{P(l_i)}{P(d)} \times \prod_{t=1}^{n} P(d_t|l_i). \end{aligned} \tag{17}$$

Similiarly, we use l_i^- to indicate that label l_i is not selected, then we can have the following equation in computing its value.

$$\begin{aligned} h_{l_i^-} &= P(l_i^-|d) \\ &= \frac{P(l_i^-)}{P(d)} \times \prod_{t=1}^{n} P(d_t|l_i^-). \end{aligned} \tag{18}$$

Given the testing data instance d, $P(d)$ is a constant. We also note here that the label importance can also be calculated by single-label classifiers, such as SVM or logistic regression, besides the Bayesian rule we employed in this article.

After calculating each single label's importance, the following step is to compute the correlations among labels. Let Q be the correlation matrix, then Q is a 2 m-by-2 m matrix, since there are m labels, each label has two events and we assume there exist correlation between any pair of two events. The correlation between l_i and l_j coined as Q_{l_i,l_j} can be calculated as follows:

$$\begin{aligned} Q_{l_i,l_j} &= P(\{l_i, l_j\}|d) \\ &= \frac{P(\{l_i, l_j\})}{P(d)} \times \prod_{t=1}^{n} P(d_t|\{l_i, l_j\}). \end{aligned} \tag{19}$$

Q_{l_i,l_i}, Q_{l_i,l_i^-} and $Q_{l_i^-,l_i^-}$ are all equal to 0, $\forall i, 1 \leq i \leq m$.

Now, all vertices and all edges have their own weights, which is shown in Fig. 1a.

As shown in Fig. 1a, l_i and l_j both have their own importances, h_{l_i} and h_{l_j}, and they also have their correlation weight Q_{l_i,l_j}. Finally, we normalize this weighted graph to let only edges be weighted, and the normalization formula is given by:

$$W_{l_i,l_j} = h_{l_i} \times Q^*_{l_i,l_j} + h_{l_j} \times Q^*_{l_j,l_i}, \tag{20}$$

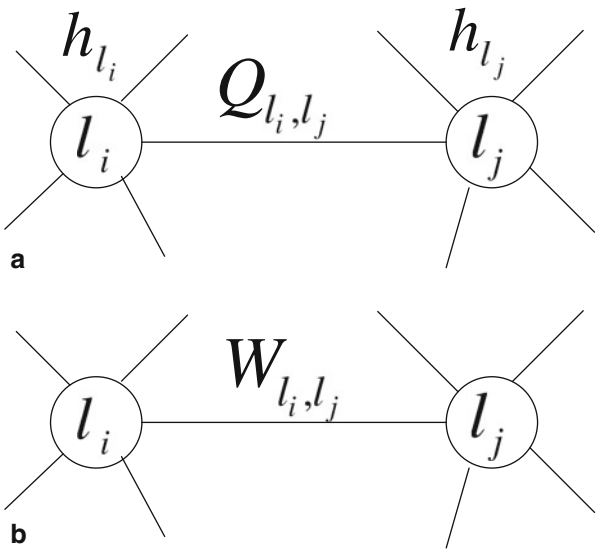


Fig. 1 Weighted graph normalization

where W_{l_i,l_j} is the normalized correlation between the labels l_i and l_j, and

$$Q^*_{l_i,l_j} = \frac{Q_{l_i,l_j}}{\sum_{l\in\{l_1,l_1^-,\ldots,l_m,l_m^-\}} Q_{l_i,l}}. \tag{21}$$

Figure 1b shows the result of Fig. 1a after the normalization. Given the normalized weighted graph, we can employ Algorithm 1 to do the first step multi-label classification—label vertices partitioning.

3.4 Partition Selection

As discussed in Sect. 3.1, the vertex assignment vector X splits vertices into two groups, where the vertices of one group have the value of $+1$, and the vertices of the other group have the value of -1. In this section, we will describe how to choose a group of vertices to determine the final multi-label for the testing instance.

We continue use the notations A and B to denote the two vertex groups. Since the two events of each label should appear separately, then for label l_i, its two events (l_i and l_i^-) should satisfy $l_i \in A$ and $l_i^- \in B$, or $l_i \in B$ and $l_i^- \in A$. Consequently, $|A| = |B| = m$.

While determine which partition will be used to finalize the labels, our aim is to select the partition with higher internal correlations, which is described by:

$$\arg\max_{Z\in\{A,B\}} \sum_{z_i,z_j\in Z, i\neq j} W_{z_i,z_j}. \tag{22}$$

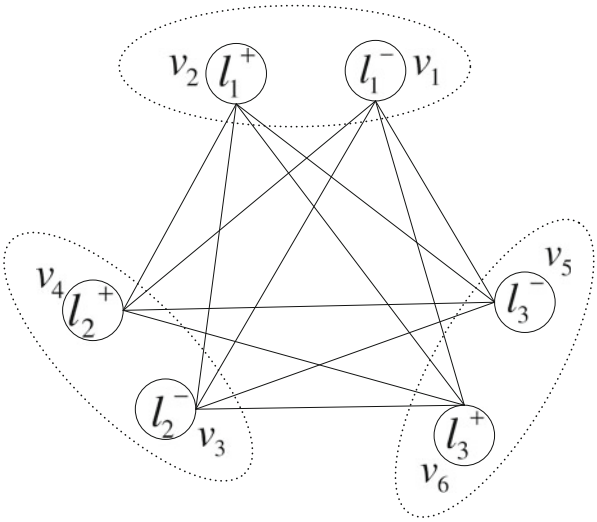


Fig. 2 One example of weighted label-graph

That is, the partition with higher internal correlations is selected to form the multi-label.

3.5 One Example

In this section, we will illustrate proposed approach with synthetic data. In this example, we skip the process of building the weighted label-graph, and assume that we have a weighted label graph shown in Fig. 2.

In Fig. 2, each vertex represents an event (positive or negative). There are three labels, l_1, l_2 and l_3, and each label has two events where l_i^+ means l_i will be chosen and l_i- means label l_i will not be chosen. The correlation matrix of the weighted label graph in Fig. 2 is shown as follows:

$$W = \begin{pmatrix} 0 & 0 & 0.1 & 0.2 & 0.4 & 0.5 \\ 0 & 0 & 0.2 & 0.3 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0 & 0 & 0.3 & 0.4 \\ 0.2 & 0.3 & 0 & 0 & 0.2 & 0.3 \\ 0.4 & 0.3 & 0.3 & 0.2 & 0 & 0 \\ 0.5 & 0.4 & 0.4 & 0.3 & 0 & 0 \end{pmatrix} \tag{23}$$

W is a symmetric square matrix. Each label has a constraint, so there are three constraints in the constraint matrix, which is given by:

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{24}$$

As shown in Eq. 24, the constraint in the first row of M indicates the for label l_1 its two events, l_1^+ and l_1^-, cannot be in a same group. We further get M's kernel:

$$K = \begin{pmatrix} 0.5 & -0.3536 & -0.3536 \\ -0.5 & 0.3536 & 0.3536 \\ -0.5 & -0.3536 & -0.3536 \\ 0.5 & 0.3536 & 0.3536 \\ 0 & 0.5 & -0.5 \\ 0 & -0.5 & 0.5 \end{pmatrix} \tag{25}$$

We use Y to denote the eigenvector corresponding to the biggest eigenvalue in $K^T W K$. Then we have:

$$Y = (0.8918, 1, -0.3851)^T. \tag{26}$$

With $X^* = KY$, we get

$$X^* = (0.2285, -0.2285, -0.6633, 0.6633, 0.6925, -0.6925)^T. \tag{27}$$

By discretizing the values in X^* using the method in Algorithm 1, we will get

$$X = (+1, -1, -1, +1, +1, -1)^T. \tag{28}$$

Refer to indicator vector X, the two vertex groups are $A = \{v_1, v_4, v_5\}$ and $B = \{v_2, v_3, v_6\}$. Use the approach given in Sect. 3.4, we can have

$$\sum_{a_i, a_j \in A, i \neq j} w(a_i, a_j) = 1.6 < \sum_{b_i, b_j \in B, i \neq j} w(b_i, b_j) = 2 \tag{29}$$

Since the intra-correlation in partition B is is stronger than A, so we will use B to form the multi-label, which is $\{l_1, l_3\}$.

4 Experimental Evaluation

4.1 Evaluation Metrics

On evaluating performance of multi-label classification methods, lots of criteria have been proposed [10, 18]. As there is no ranking score of each label in multi-labels predicted by our approach, then we focus on example-based metrics [25], including $Hamming\ loss$, $Precision$, $Recall$, $F1$ and $Accuracy$. To describe these metrics clearly, we define some necessary notations firstly. Let $T = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_q, Y_q)\}$ $(q \in \mathbb{R})$ be a test data set, where x_i $(1 \leq i \leq q)$ is a test instance, and $Y_i \in C^*$ is x_i's multi-label. In addition, let $h(x_i)$ be the predicted multi-label for x_i by one multi-label classifier.

- *Hamming loss* calculates the percentage of mis-predicted labels:

$$\mathrm{HL}(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{1}{m}|h(x_i)\Delta Y_i|, \tag{30}$$

  where Δ is the notation for differentiating two sets. As shown in Eq. 30, the mis-predicted labels for x_i include the labels, which appear in Y_i but not in $h(x_i)$, and the labels, which are in $h(x_i)$ but not in Y_i.
- *Precision* comes from the metrics for single-label classifiers in information retrieval (IR):

$$P(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{|Y_i \cap h(x_i)|}{|h(x_i)|} \tag{31}$$

- *Recall* corresponds to *recall* in single-label metrics:

$$R(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{|Y_i \cap h(x_i)|}{|Y_i|} \tag{32}$$

- *F1* combines *precision* and *recall*:

$$F1(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{2 \times P(x_i) \times R(x_i)}{P(x_i) + R(x_i)} \tag{33}$$

- *Accuracy* is similar to *accuracy* in single-label metrics:

$$A(T) = \frac{1}{q}\sum_{i=1}^{q}\frac{|Y_i \cap h(x_i)|}{|Y_i \cup h(x_i)|} \tag{34}$$

Among these five metrics, only $Hamming\ loss$ follows this pattern that if the value of $Hamming\ loss$ is lower, the classifier's performance is better. And all the other four metrics obey the rule that if the metrics's value is higher, the classifier's performance is better.

4.2 Medical Application

Multi-label classification can be extensively applied to solve different problems in medical informatics domain. A typical scenario is that one patient has some symptoms, a doctor needs to make a treatment plan for the patient according to his/her symptoms. Before deciding the treatment plan, a necessary decision should be made for the doctor is to predict which kinds of adverse effects the patient will have for the planned treatment. Since some effects are related with each other, then the doctor should not only consider the occurrence possibility of each effect individually, but also take into account the correlations among effects. As shown in Sect. 3.3, our weighted label-graph is built on both label importances and label correlations, which makes our approach good for handling this kind of scenarios. In the following, we will test our approach's performance on a real medical data set.

4.2.1 Evaluation Procedure

Subjects aged 18–60 years undergoing arthroscopic shoulder surgery will be randomized into three groups. Subjects and observers will be blinded as to group assignment. Unblinded investigators will administer the injection and establish the infusion in the preoperative holding area. In the operating room the quality of the block will be assessed by surgical manipulation. Sedation with intravenous propofol will be established. General anesthesia will be established if needed (based upon the response to manipulation and the extent of hypesthesia over the C5 and C6 dermatomes) according to standard practice with propofol for induction, a laryngeal mask airway (LMA) and sevoflurane for anesthetic maintenance. Supplemental postoperative analgesia will be provided as needed using intravenous fentanyl, intravenous ketorolac (30 mg, one time dose), and oral hydrocodone/acetaminophen as per the WBH acute pain algorithm (WBH-RO Form 6459). The Stryker Pain Pump II will be used for continuous infusion per WBH policy (WBH-RO: Anesthetic Infusion Device: Pain Pump; Policy 480, part II: pp. 2–7) with the following settings: Ropivacaine 0.2 %; bolus: 3 ml; continuous infusion: 4 ml/h; lock-out: 20 min. A single dose of dexamethasone will be allowed for postoperative nausea and vomiting (PONV) or PONV prophylaxis.

4.2.2 Interscalene Block

Ultrasound-guided placement (high frequency probe; 10–12 Hz) of an interscalene block and catheter will be performed at the ipsilateral C5 level. Observation of spread around the C5 root and upper trunk of the brachial plexus will confirm correct placement and document distant spread outside the interscalene space between the anterior and middle scalene muscles. The presence or absence of spread medially to the vicinity of the carotid artery will be documented. Ropivacaine 0.75 % will be used for the initial block and to facilitate catheter placement. Patients will be randomly assigned to receive 5, 10, or 20 ml bolus. This randomization will be determined

Table 1 Performance comparison in the format of '*mean* ± *variance*'

	ML-KNN	Our proposed approach
Hamming loss	0.0129 ± 91.3333e–006	*0.0112 ± 1.2554e–004*
Precision	0.9028 ± 0.0023	*0.9236 ± 0.0064*
Recall	0.9028 ± 0.0023	*0.9236 ± 0.0064*
F1	0.9028 ± 0.0023	*0.9236 ± 0.0064*
Accuracy	0.9028 ± 0.0023	*0.9236 ± 0.0064*

prior to start of study using standard randomization techniques. Continuous infusion will be started per the previously mentioned standard protocol immediately after catheter placement.

In this data set, there are 36 patient records, and each record describes a patient's symptoms before and after a treatment (catheter placement). We call the symptoms before a treatment as features, and the symptoms after a treatment as adverse effects. There are 16 features and 4 adverse effects. Each adverse effect has two optional values: 1 or 0. 1 means this adverse effect takes place. The four adverse effects include *Dysphonia*, *Hand Weakness*, *Horner's Syndrome* and *Feeling Jittery*.

Our task is to predict which adverse effects will occur given a symptom set. In our solution, each adverse effect is treated as a label, and whether an adverse effect occurs or not is indicated by its corresponding label's two events, which is similar to the example given in Sect. 3.5.

Our experiments are tested with nine-fold cross-validation. We compare our approach with ML-KNN using the code shared in [33]. The performance comparisons of three classifiers are given in Fig. 3.

As shown in Fig. 3, labels *Dysphonia*, *Horner's Syndrome*, *Hand Weakness* and *Feeling Jittery* are represented as 1, 2, 3 and 4, respectively. According to Fig. 3, our approach's performances are much better than those of ML-KNN in all five metrics. To get an overview idea, we summarize the performances in Table 1, which shows that our proposed approach works better than ML-KNN.

4.3 Discussion

In this section, we discuss further the related issues between our approach and other existing methods.

Compared with LP, our approach has more candidate multi-labels. LP requires each generated multi-label to appear in the training data set. There is no such requirement in our approach. Furthermore, our approach focuses on the correlation between two labels, which remedies to some extent the disadvantage of LP that the generated multi-labels do not have enough data instances associated with.

Another advantage of our approach is that we do not need to specify any parameter during learning. In BR-based methods, users have to specify a confidence threshold

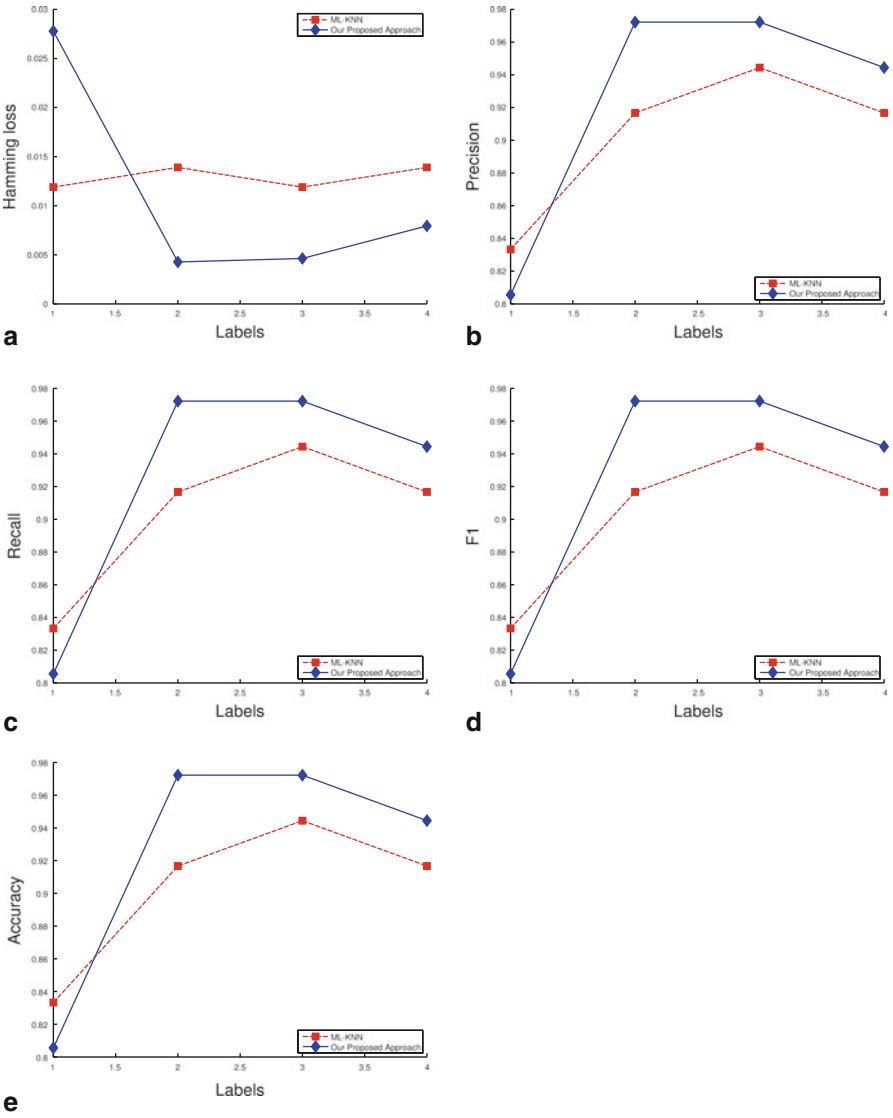


Fig. 3 Experimental results

for label selection. In ML-KNN, the value of K has to be assigned before training. This parameter-free characteristic makes our approach easy to use.

In our approach, we make the assumption that all labels are dependent. One way to enhance is to utlize a pre-processing module to evaluate the dependency between two labels. For two labels, l_i and l_j, if the mutual information between l_i and l_i is low in proportion to either l_i's or l_j's entropy, then these two labels are independent. The

other issue is that our approach only considers the correlation between two labels. The ideal situation is to consider the dependency between any number of labels, which is computational prohibitive.

5 Conclusion

In this article, we proposed a novel constrained minimum cut model based approach to multi-label classification. In this approach, choosing a label or not is associated with two exclusive events respectively, and a multi-label is a combination of these events. We use a weighted label graph to represent the labels and their correlations. Multi-label classification problem is then transformed into finding a minimum cut problem. Compared with existing approaches, our approach starts from a global optimization perspective in choosing multi-labels. The experimental results show the effectiveness of our approach.

References

1. Brinker, K., Hüllermeier, E.: Case-based multilabel ranking. In: M.M. Veloso, M.M. Veloso (eds.) IJCAI, pp. 702–707. (2007).
2. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learn. **76**(2–3), 211–225. http://dx.doi.org/10.1007/s10994-009-5127-5 (2009). doi:10.1007/s10994-009-5127-5
3. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: PKDD '01: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 42–53. Springer-Verlag, London, UK (2001)
4. De Comité, F., Gilleron, R., Tommasi, M.: Learning multi-label alternating decision trees from texts and data. pp. 251–274. http://dx.doi.org/10.1007/3-540-45065-3_4 (2003). doi:10.1007/3-540-45065-3_4
5. Dembczynski, K., Waegeman, W., Cheng, W., H¡§ullermeier, E.: On label dependence in multi-label classification. In: MLD 2010 : 2nd International Workshop on learning from Multi-Label Data (2010)
6. Elisseeff, A., Weston, J.: Kernel methods for multi-labelled classification and categorical regression problems. In: Advances in Neural Information Processing Systems 14, pp. 681–687. MIT Press (2001)
7. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Annual ACM Conference on Research and Development in Information Retrieval, pp. 274–281. http://citeseerx.ist.psu.edu/viewdoc/summary? (2005). doi:10.1.1.18.24 23
8. Fujino, A., Isozaki, H.: Multi-label classification using logistic regression models for ntcir-7 patent mining task. In: Proceedings of NTCIR-7 Workshop Meeting (2008)
9. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 195–200. ACM, New York, NY, USA (2005). doi:http://doi.acm.org/10.1145/1099554.1099591
10. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 22–30. Springer (2004)

11. Gross, J., Yellen, J.: Graph Theory and its Applications. CRC Press, Boca Raton (1998)
12. McCallum, A.K.: Multi-label text classification with a mixture model trained by EM algorithm. http://citeseer.ist.psu.edu/mccallum99multilabel.html (1999).
13. Nakos, G., Joyner, D.: Linear algebra with applications, pp. 472–473. Brooks/Cole Publishing Company. Pacific Grove, California, United States. (1998)
14. Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 248–256. Association for Computational Linguistics, Singapore. http://www.aclweb.org/anthology/D/D09/D09-1026 (2009).
15. Read, J., Pfahringer, B., Holmes, G.: Multi-label classification using ensembles of pruned sets. In: ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, vol. 0, pp. 995–1000. IEEE Computer Society, Washington, DC, USA. http://dx.doi.org/10.1109/ICDM.2008.74 (2008) doi:10.1109/ICDM.2008.74.
16. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: ECML PKDD '09: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 254–269. Springer-Verlag, Berlin, Heidelberg. (2009)
17. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based learning of hierarchical multilabel classification models. J. Machine Learn. Res. **7**, 1601–1626 (2006)
18. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learn. **39**(2/3), 135–168. http://citeseerx.ist.psu.edu/viewdoc/summary? (2000). doi:10.1.1.33.16 66
19. Spielman, D.: Spectral graph theory and its applications. Foundations of Computer Science, 2007. 48th Annual IEEE Symposium on FOCS '07, pp. 29–38 (2007)
20. Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An empirical study of lazy multilabel classification algorithms. In: SETN '08: Proceedings of the 5th Hellenic conference on Artificial Intelligence, pp. 401–406. Springer-Verlag, Berlin, Heidelberg (2008) doi:http://dx.doi.org/10.1007/978-3-540-87881-0_40
21. Streich, A., Buhmann, J.: Classification of multi-labeled data: A generative approach. pp. 390–405. http://dx.doi.org/10.1007/978-3-540-87481-2_26 (2008). doi:10.1007/978-3-540-87481-2_26
22. Tenenboim, L., Rokach, L., Shapira, B.: Identification of label dependencies for multi-label classification. In: MLD 2010 : 2nd International Workshop on learning from Multi-Label Data (2010)
23. Tsoumakas, G., Katakis, I.: Multi label classification: an overview. Int. J. Data Warehousing Mining **3**(3), 1–13. http://mlkd.csd.auth.gr/publication_details.asp?publicationID =219 (2007).
24. Tsoumakas, G., Katakis, I., Vlahava, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data mining and knowledge discovery handbook, 2nd edn, pp. 667–685. Springer New York (2010)
25. Tsoumakas, G., Katakis, I., Vlahavas, I.: A review of multi-label classification methods. Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD 2006) (2006)
26. Tsoumakas, G., Vlahavas, I.: Random *k*-labelsets: An ensemble method for multilabel classification. In: ECML '07: Proceedings of the 18th European Conference on Machine Learning, pp. 406–417. Springer-Verlag, Berlin, Heidelberg (2007). doi:http://dx.doi.org/10.1007/978-3-540-74958-5-38
27. Ueda, N., Saito, K.: Parametric mixture models for multi-labeled text. http://citeseer.ist.psu.edu/ueda03parametric.html (2002)
28. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learn. **2**(73), 185–214. http://dx.doi.org/10.1007/s10994-008-5077-3 (2008). doi:10.1007/s10994-008-5077-3
29. Wang, H., Huang, M., Zhu, X.: A generative probabilistic model for multi-label classification. In: ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on

Data Mining, pp. 628–637. IEEE Computer Society, Washington, DC, USA. http://dx.doi.org/10.1109/ICDM.2008.86 (2008). doi:10.1109/ICDM.2008.86
30. Wikipedia: Cut (Graph theory) (9 Jan 2011). http://en.wikipedia.org/wiki/Cut_(graph_theory)
31. Zhang, M.L., Pe na, J.M., Robles, V.: Feature selection for multi-label naive Bayes classification. Inf. Sci. **179**(19), 3218–3229 (2009). doi:http://dx.doi.org/10.1016/j.ins.2009.06.010
32. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognit. **40**(7), 2038–2048. http://dx.doi.org/10.1016/j.patcog.2006.12.019 (2007). doi:10.1016/j.patcog.2006.12.019
33. Zhang, M.L., Zhou, Z.H.: Ml-knn codes. http://lamda.nju.edu.cn/datacode/MLkNN.htm (2009)

On the Selection of Dimension Reduction Techniques for Scientific Applications

Ya Ju Fan and Chandrika Kamath

Abstract Many dimension reduction methods have been proposed to discover the intrinsic, lower dimensional structure of a high-dimensional dataset. However, determining critical features in datasets that consist of a large number of features is still a challenge. In this article, through a series of carefully designed experiments on real-world datasets, we investigate the performance of different dimension reduction techniques, ranging from feature subset selection to methods that transform the features into a lower dimensional space. We also discuss methods that calculate the intrinsic dimensionality of a dataset in order to understand the reduced dimension. Using several evaluation strategies, we show how these different methods can provide useful insights into the data. These comparisons enable us to provide guidance to users on the selection of a technique for their dataset.

1 Introduction

It is a challenge to understand, interpret, and analyze high-dimensional data, where each example or instance is described by many features. Often, only a few features are important to the analysis task, or the data naturally lie on a lower-dimensional manifold. To reduce the dimension of the dataset, we can either identify a subset of features as important using techniques such as filters [10] and wrappers [19]. Or, we can transform the data into a reduced dimensional representation while preserving meaningful structures in the data. These methods include linear projections, such as principal component analysis (PCA) [28], as well as several non-linear methods that have been proposed recently [22].

To fully benefit from this wealth of dimension reduction techniques, we need to understand their strengths and weaknesses better so we can determine a method appropriate for a dataset and task, select the parameters for the method suitably, and

Y. J. Fan (✉) · C. Kamath
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA
e-mail: fan4@llnl.gov

C. Kamath
e-mail: kamath2@llnl.gov

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_6

interpret the results correctly to provide insights into the data. Some techniques, such as PCA, filters, and wrappers, have been studied extensively and applied to real problems. Others, such as the recent non-linear dimension reduction (NLDR) techniques, have been explained, and their benefits demonstrated, through the use of synthetic datasets, such as the three-dimensional Swiss roll data. While the simplicity of these datasets is useful in visually explaining the techniques, it is unclear how they perform in real problems where the dimensionality is too high for visualization. Additional guidance is needed to ascertain if these newer techniques are more appropriate than other approaches in their ability to represent the data in the lower-dimensional space, their computational cost, and the interpretability of the results.

In this article, we present a series of carefully designed experiments with real datasets to gain insights into the different dimension reduction methods. We consider data from three science domains: astronomy, wind power generation, and remote sensing, where these techniques are used to identify features important to the phenomenon being observed, to build more accurate predictive models, to reduce the number of features that need to be measured, and to reduce the number of samples required to explore the feature space of a problem.

To provide guidance to a practitioner, we focus on three aspects of the task of dimension reduction. First, we evaluate the techniques using datasets with properties that arise commonly in practice, such as data with noise features, with labeling based on different criterion, or with very high dimensionality. These data may also have other unknown properties, such as inherent lower dimensional manifolds. Second, we consider the task of setting the dimensionality of the lower dimensional space. This important issue is rarely discussed in the context of real datasets whose high dimensionality prevents visualization to understand their properties. And finally, we consider ways in which we might interpret the results obtained using the different methods.

This paper is organized as follows: we start by briefly describing data transformation methods and feature subset selection techniques in Sects. 2 and 3, respectively. Next, in Sect. 4, we discuss how we can obtain the intrinsic dimensionality of the data by exploiting the information provided by these methods. In Sect. 5, we describe the scientific problems of interest, followed by our evaluation methodology for the dimension reduction techniques in Sect. 6. The experimental results are discussed in Sect. 7. In Sect. 8 we describe related work and conclude in Sect. 9 by summarizing our guidance for practitioners.

The notation used in this paper is as follows: $X \in \mathbb{R}^{n \times D}$ represents the dataset in the high-dimensional space, that is, X consists of n data points, X_i, each of length D, the dimension of the data. We want to reduce the dimension of these points resulting in the dataset, $Y \in \mathbb{R}^{n \times d}$, where $d < D$.

2 Dimension Reduction Using Transformation

We next briefly describe the transform-based techniques, including PCA and four popular NLDR techniques: Isomap, locally linear embedding (LLE), Laplacian eigenmaps, and local tangent space alignment (LTSA) [24, 27]. These methods

share the use of an eigendecomposition to obtain a lower-dimensional embedding of the data that is guaranteed to provide global optimality.

2.1 Principal Component Analysis (PCA)

PCA [28] is a linear technique that preserves the largest variance in the data while decorrelating the transformed dataset. An eigenvalue problem to the data covariance matrix, C, is formulated as $CM = \lambda M$. The eigenvectors, M, corresponding to the significant eigenvalues, λ, form a basis for linear transformation that optimally maximizes the variance in the data. The low-dimensional representation is expressed by $Y = XM$ and the eigenvalues can be used to determine the lower dimensionality, d.

PCA does not require any parameter to be set. It has a computational cost of $O(D^3)$ and requires $O(D^2)$ memory for $n > D$.

2.2 Isomap

The Isomap method [35] preserves pairwise geodesic distances between data points. It starts by constructing an adjacency graph based on the neighbors of each point in the input space. These neighbors can be either the k-nearest neighbors or points which lie within an ϵ-neighborhood. Next, the geodesic distances [5, 7] between all pairs of points are estimated by computing their shortest path distances over the graph. Let $D_G = \{d_G(i, j)\}_{i,j=1,\ldots,n}$ be the matrix of geodesic distances, where $d_G(i, j)$ is the distance between points i and j. Isomap then constructs an embedding in a d-dimensional Euclidean space such that the pair-wise Euclidean distances between points in this space approximate the geodesic distances in the input space. Let $D_Y = \{d_Y(i, j)\}_{i,j=1,\ldots,n}$ be the Euclidean distance matrix and $d_Y(i, j) = \|Y_i - Y_j\|_2$. The goal is to minimize the cost function $\|\tau(D_G) - \tau(D_Y)\|_2$, where the function τ performs double centering on the matrix to support efficient optimization. The optimal solution is found by solving the eigen-decomposition of $\tau(D_G)$. The Y coordinates are then computed based on the d largest eigenvalues and their corresponding eigenvectors.

Isomap requires one parameter k (or ϵ), has a computational cost of $O(n^3)$ and requires $O(n^2)$ memory.

2.3 Locally Linear Embedding (LLE)

The LLE method [30] preserves the reconstruction weights ω_{ij} that are used to describe a data point X_i as a linear combination of its neighbors X_j, $j \in \mathcal{N}(i)$, where $\mathcal{N}(i)$ is the set of neighbors of point i. The optimal weights for each i are obtained by

minimizing the cost function, $\min_\omega \quad \{\|X_i - \sum_{j\in\mathcal{N}(i)} \omega_{ij} X_j\|^2 \big| \sum_{j\in\mathcal{N}(i)} \omega_{ij} = 1\}$. LLE assumes that the manifold is locally linear and hence the reconstruction weights are invariant in the low-dimensional space. The embedding Y of LLE is the solution of minimizing the cost function $\sum_i \|Y_i - \sum_j W_{ij} Y_j\|^2$, where W is the reconstruction weight matrix with elements $W_{ij} = 0$ if $j \notin \mathcal{N}(i)$; $W_{ij} = \omega_{ij}$ otherwise. Y can be obtained from the eigenvectors corresponding to the smallest d nonzero eigenvalues of the embedding matrix, defined as $M = (I - W)^\mathsf{T}(I - W)$, where I is an identity matrix.

LLE requires one parameter k (or ϵ), has a computational cost of $O(pn^2)$ and requires $O(pn^2)$ memory, where p is the fraction of non-zero elements in the sparse matrix.

2.4 Laplacian Eigenmaps

This method provides a low-dimensional representation in which the weighted distances between a data point and other points within an ϵ-neighborhood (or k-nearest neighbors) are minimized [2]. The distances to the neighbors are weighted using the Laplacian operator $(S - W)$, where $W_{ij} = \mathrm{e}^{-\frac{\|X_i - X_j\|^2}{t}}$ and $S_{ii} = \sum_j W_{ij}$.

Here, $t = 2\sigma^2$, where σ is the standard deviation of the Gaussian kernel. The objective is to find: $\arg\min_Y \{\mathrm{tr}(Y^\mathsf{T}(S - W)Y) | Y^\mathsf{T} S Y = 1\}$. The representation of Y is computed by solving the generalized eigenvector problem: $(S - W)v = \lambda S v$. Only the eigenvectors (v) corresponding to the smallest nonzero eigenvalues (λ) are used for the embedding.

Laplacian Eigenmaps requires two parameters k (or ϵ) and t, has a computational cost of $O(pn^2)$ and requires $O(pn^2)$ memory.

2.5 Local Tangent Space Alignment (LTSA)

The LTSA method [41] applies PCA on the neighborhood of each data point, forming a local tangent space that represents the local geometry. The space is denoted by the local coordinates $\theta_j^{(i)}$, $j = 1, \ldots, k$ that are the k nearest neighbors of point i, for each point $i = 1, \ldots, n$. Those local tangent spaces are then aligned to construct the global coordinate system of the underlying manifold. The global coordinates should preserve the local geometry determined by the $\theta_j^{(i)}$ as much as possible. Therefore, to construct the global coordinates $Y_i, i = 1, \ldots, n$, in the low-dimensional feature space, LTSA seeks to find the local affine transformations L_i to minimize the reconstruction errors, $\sum_i \|E_i\|^2 = \sum_i \|Y_i(I - \frac{1}{k} ee^\mathsf{T}) - L_i \Theta_i\|^2$, where I is an identity matrix, e is a column vector of ones, and $\Theta_i = [\theta_1^{(i)}, \ldots, \theta_k^{(i)}]$.

LTSA requires one parameter k and the determination of d before applying the method. It has a computational cost of $O(pn^2)$ and requires $O(pn^2)$ memory.

3 Dimension Reduction Using Feature Subset Selection

We consider four methods which are applicable when the dataset is labeled.

3.1 Stump Filter

A stump is a decision tree with only the root node; the stump filter ranks features using the same process as the one used to create the root node. Decision trees split the data by examining each feature and finding the split that optimizes an impurity measure. To search for the optimal split for a numeric feature x, the feature values are sorted ($x_1 < x_2 < \cdots < x_n$) and all mid-points $(x_i + x_{i+1})/2$ are evaluated as possible splits using a given impurity measure. The features are then ranked according to their optimal impurity measures. In our work, we use the Gini index [3] as a measure of the impurity.

3.2 Distance Filter

This filter calculates the class separability of each feature using the Kullback-Leibler (KL) distance between histograms of feature values. For each feature, there is one histogram for each class. In a two class problem, if a feature has a large distance between the histograms for the two classes, then it is likely to be an important feature in differentiating between the classes. We discretized numeric features using $\sqrt{|n|}/2$ equally-spaced bins, where $|n|$ is the size of the data. Let $p_j(d = i|c = a)$ be an estimate of the probability that the j-th feature takes a value in the i th bin of the histogram given a class a. For each feature j, we calculate the class separability as $\Delta_j = \sum_{a=1}^{c} \sum_{b=1}^{c} \delta_j(a, b)$, where c is the number of classes and $\delta_j(a, b)$ is the KL distance between histograms corresponding to classes a and b: $\delta_j(a, b) = \sum_{i=1}^{B} p_j(d = i|c = a) \log\left(\frac{p_j(d=i|c=a)}{p_j(d=i|c=b)}\right)$, where B is the number of bins in the histograms. The features are ranked simply by sorting them in descending order of the distances Δ_j (larger distances mean better separability).

3.3 Chi-squared Filter

The Chi-squared filter computes the Chi-square statistics from contingency tables for every feature. The contingency tables have one row for every class label and the columns correspond to possible values of the feature (see Table 1, adapted from [14]). Numeric features are represented by histograms, so the columns of the contingency table are the histogram bins. The Chi-square statistic for feature j is $\chi_j^2 = \sum_i \frac{(o_i - e_i)^2}{e_i}$,

Table 1 A 2×3 contingency table, with observed and expected frequencies (in parenthesis) of a fictitious feature f_1 that takes on three possible values (=1, 2, and 3)

Class	$f_1 = 1$	$f_1 = 2$	$f_1 = 3$	Total
0	31 (22.5)	20 (21)	11 (18.5)	62
1	14 (22.5)	22 (21)	26 (18.5)	62
Total	45	42	37	124

where the sum is over all the cells in the $r \times c$ contingency table, where r is the number of rows and c is the number of columns; o_i stands for the observed value (the count of the items corresponding to the cell i in the contingency table); and e_i is the expected frequency of items calculated as: $e_i = \frac{(\text{column total}) \times (\text{row total})}{\text{grand total}}$. The variables are ranked by sorting them in descending order of their χ^2 statistics.

3.4 ReliefF

ReliefF [29] estimates the quality of features by calculating how well they distinguish between instances close to each other. It starts by taking an instance i at random and identifies its nearest k hits (H_i) and misses (M_i), which are the closest instances of the same and different classes, respectively. Then, it obtains the quality estimate of a feature s, which for a two-class dataset is defined as: $Q_s = \sum_{i=1}^{n} \left\{ \sum_{m \in M_i} \frac{\|X_{is} - X_{ms}\|}{nk} - \sum_{h \in H_i} \frac{\|X_{is} - X_{hs}\|}{nk} \right\}$ where X_{is} is the value of feature s for instance i. By increasing the quality estimate when the selected point and its misses have different values of feature s, and decreasing it when the point and its hits have different values of the feature, ReliefF ranks the features based on their ability to distinguish between instances of the same and different classes.

4 Determining the Reduced Dimensionality of the Data

An important issue in dimension reduction is the choice of the number of dimensions for the low-dimensional solution. While many algorithms require the reduced dimensionality of the embedding be explicitly set, only a few provide an estimate of this number.

In feature subset selection methods, we can easily identify the number of features to select by considering the metric used to order the features and disregarding features ranked lower than a certain threshold value. Or, we can include a noise feature and disregard any features ranked lower than this noise feature.

In the case of PCA, an adequate number of principal components is identified by ordering the eigenvalues and selecting the top d significant principal components; the remainder describes the reconstruction error: $E_d = \sum_{j=d+1}^{D} \lambda_j$ [39]. Many selection criteria have been developed based on the magnitude of eigenvalues. In our work, we use the number of eigenvalues that exceed a fixed percentage of the largest eigenvalue

[8]. For example, we use d^{10} to indicate the number of eigenvalues that exceed 10% of the largest eigenvalue.

For nonlinear methods, the use of the eigen-spectrum only works when the data lie on a linear manifold [32]; so, we need to consider other methods. One such approach applicable to Isomap and LLE is based on the *elbow test* using a lack-of-fit measure. We first determine the property that the NLDR technique is trying to preserve. The deviation between the property in the low-dimensional space and the input space is plot against the dimensionality and the intrinsic dimension is chosen at the "elbow" in the plot where, after a certain number of dimensions, the lack-of-fit value is not reduced substantially. For Isomap, the lack-of-fit measure is the residual variances of the two geodesic distance matrices evaluated in the representation space and in the input space. For LLE, we use the reconstruction error. The reconstruction weights are updated using the embedding vectors Y_i and then applied to the input data X_i. The intrinsic dimensionality d can be estimated by the values of reciprocal cost function [30], defined as $f(W^{(d)}) = \sum_i \|X_i - \sum_j W_{ij}^{(d)} X_j\|^2$, where $W^{(d)}$ is the reconstruction weight matrix computed using the d-dimensional representation vectors Y_i.

Finally, we investigate ideas from the intrinsic dimensionality literature to determine if they can provide insights. The intrinsic dimension of the data is the minimum number of variables necessary to represent the observed properties of the data. A statistical estimation [37] can be calculated based on the assumption that the topological hypersurface in a local region can be approximated by a linear hypersurface of the same dimensionality. We start by calculating the distances between all points. Then, for each point i, we find the closest neighbor j_0; the vector connecting i to j_0 forms a subspace of dimension one. We then consider the next closest neighbor j_1 to i, and consider the angle between the vector connecting i and j_1 and the subspace. These vectors connecting i and its l closest neighbors form an l-dimensional space. We continue increasing the size of l until, for a certain dimension, d, the mean of the angles taken over all points is less than a threshold.

An alternate approach is to determine the locally linear scale using simple box counting. Let $C(r)$ indicate the number of data points that are contained within a ball of radius r centered on a data point. If the data are sampled over a d-dimensional manifold, then $C(r)$ is proportional to r^d for small r. The intrinsic dimensionality at the locally linear scale is $d = \frac{\partial \ln C(r)}{\partial \ln r}$. Since datasets have finite samples in practice, we can obtain the estimate by plotting $\ln C(r)$ versus $\ln r$ and measuring the slope of the linear part of the curve [18].

5 Datasets Used in the Evaluation

We evaluate the dimension reduction techniques using classification problems in three science domains: astronomy, wind energy, and remote sensing.

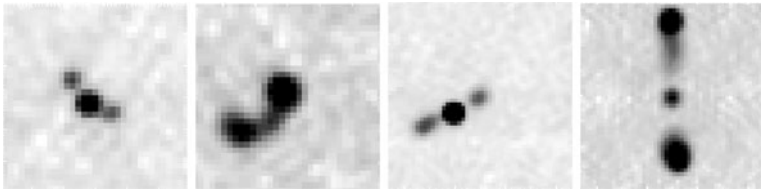


Fig. 1 Examples of bent-double (*left two*) and non-bent double (*right two*) radio-emitting galaxies

5.1 Astronomy Dataset

This dataset [16] is used to build a model to classify radio-emitting galaxies into two classes—one with a bent-double morphology (called 'bents') and the other without (called 'non-bents') (Fig. 1). These data are from the Faint Images of the Radio Sky at Twenty-cm (FIRST) survey [1]. The astronomers first processed the raw image data to create a 'catalog' by fitting two-dimensional, elliptic Gaussians to each galaxy. Each entry in the catalog corresponds to a Gaussian and includes information such as the location and size of the Gaussian, the major and minor axes of the ellipse, and the peak flux. This catalog was then processed to group nearby Gaussians into galaxies and extract features, such as angles and distances, that represented each galaxy. The focus was on galaxies composed of three Gaussians and the features included those obtained by considering each Gaussian individually, considering the Gaussians taken two at a time, and considering all three Gaussians.

This dataset, which we refer to as the *First* dataset, is quite small, consisting of 195 examples, with 167 bents and 28 non-bents, each described by 99 features, of which 9 are non-numeric. In addition, we also consider a derived dataset, which we refer to as *FirstTriples*, containing only the 20 numeric features for all three Gaussians. The astronomers thought this subset to be a better representation of the bent galaxies.

5.2 Wind Energy Dataset

Our next application area is wind power generation. The task is one of using the weather conditions provided by meteorological towers in the region of the wind farms to classify days which will have ramp events [15]. A ramp event occurs when the wind power generation suddenly increases or decreases by a large amount in a short time (Fig. 2). These events make it difficult for the control room operators to schedule wind energy on the power grid. If we can use the weather conditions to predict if a day will have a ramp event, the grid operators can be better prepared to keep the grid balanced in the presence of these events.

In this dataset, we have 731 examples representing the data for the days in 2007–2008. The features are the daily averages of different variables, such as wind speed, wind direction, and temperature, at three meteorological towers in the Tehachapi

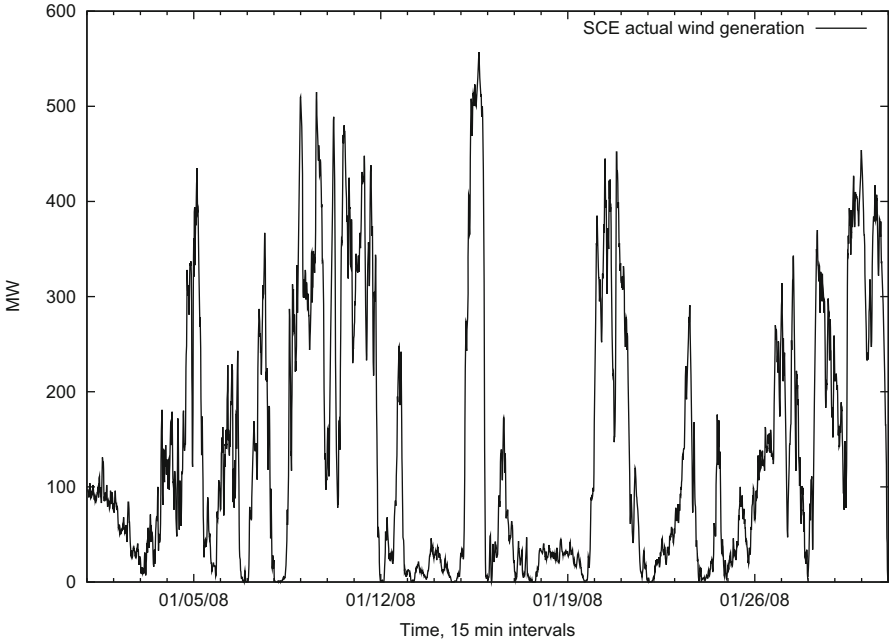


Fig. 2 Wind power generation from the wind farms in the Tehachapi Pass region in Southern California for January 2008

Pass region. Each tower provides 7 features, for a total of 21 features. Each day is assigned a binary class variable, indicating if a ramp event exceeding a certain magnitude occurred in any 1 h interval during that day. There are two datasets, *Wind115* and *Wind150*, which correspond to ramps with magnitudes exceeding 115 and 150 MW, respectively. That is, in the *Wind115* dataset, a day is assigned a label of 1 if during any 1 h interval, the wind power generation increased or decreased by more than 115 MW.

5.3 Remote Sensing Dataset

Our third application area is remote sensing, where the task is to classify tiles in satellite images of the earth as being inhabited or uninhabited (Fig. 3; data from the IKONOS satellite (www.geoeye.com)). The data are available as 4-band multi-spectral (near-infrared, red, green, and blue) images at 4 m ground sample distance. An image is divided into non-overlapping tiles of size 64×64 pixels. Each tile is represented by several texture features as the domain experts believed that texture could indicate man-made structures, such as houses or parking lots where there is certain regular structure that can be represented as a 'texture'. However, as they were not sure which texture feature was the most appropriate, they extracted several,

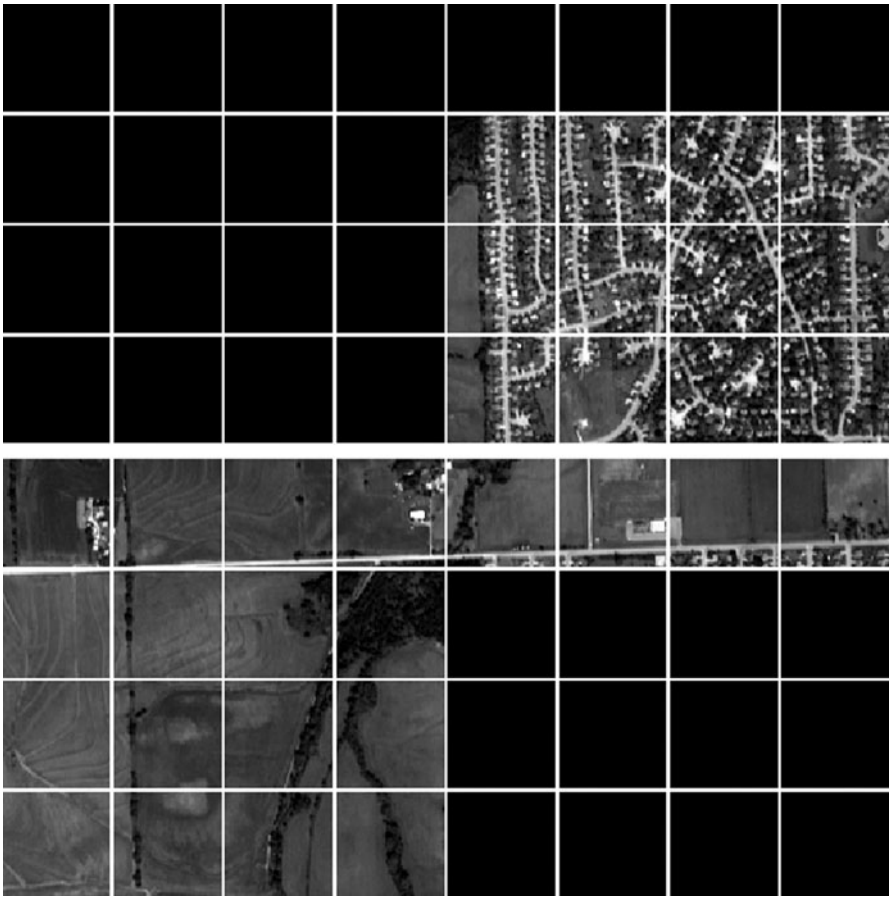


Fig. 3 Example of a region in satellite imagery illustrating the ground truth with inhabited tiles (*top*) and uninhabited tiles (*bottom*). Original satellite image by GeoEye (formerly Space Imaging)

including the Grey Level Co-occurrence Matrix (GLCM), the power spectrum texture features, the wavelet texture features, and the Gabor texture features [11, 25, 26]. Further, as it was not clear which of the 4-bands had the most relevant information, the domain experts extracted the texture features for each band and concatenated them, resulting in a long feature vector of 496 features (124 from each band), representing a tile. In our experiments, we use a dataset consisting of 2000 examples, distributed equally among inhabited and uninhabited tiles. We refer to this as the *Remote* dataset.

6 Evaluation Methodology

We evaluate the effectiveness of the dimension reduction methods using the classification accuracy of the transformed or selected features relative to the accuracy using all the original features. In our work, we consider decision tree classifiers

as their results, being easily interpreted, can be explained to domain scientists. Also, decision tree classifiers utilize the order of the significance of features [31], making them suitable for our use as the features in the lower dimensional space are ordered using either the magnitude of eigenvalues or a metric that determines the discriminating ability of a feature. We could have also used other classifiers such as support vector machines or neural networks, but their results are not as easily interpreted. We could have also used sparse methods which incorporate feature selection [36, 42], but they are more suitable for regression problems.

In our work, we used the ensemble approach proposed in [17] as it gives more accurate results than bagging or boosting. This approach creates ensembles by introducing randomization at each node of the tree in two ways. It first randomly samples the examples at a node and selects a fraction (we use 0.7) for further consideration. Then, for each feature, instead of sorting these examples based on the values of the feature, it creates a histogram, evaluates the splitting criterion (we use Gini [3]) at the mid-point of each bin of the histogram, identifies the best bin, and then selects the split point randomly in this bin. The randomization is introduced both in the sampling and in the choice of the split point. The use of the histograms speeds up the creation of each tree in the ensemble. We use 10 trees in the ensemble. Using the first d transformed features, we report the percentage error rate obtained for five-fold cross validation repeated five times and evaluate how this error rate changes as the number of features is increased.

We observe that our use of a classifier for evaluation may favor the feature selection methods as they exploit the class label in the ordering of features by importance. In contrast, data transformation methods are unsupervised, trying to find hidden structure of the data without knowledge of the class labels. Despite this drawback of the data transformation methods, we expect that in comparison to the error rate using all original features, the transform based methods should provide an improvement.

In addition to classification accuracy, we also evaluate the dimension reduction methods on the insights they can provide into the data. The major advantage of feature subset selection is that the methods identify the important original features, which can be used to understand scientific phenomenon. In contrast, for the data transformation methods, it is not easy to explain what forms the features in the new space. In the case of PCA, since it transforms the original data using linear combinations of the top d eigenvectors, we can consider the values of elements of the eigenvectors for insights. The absolute values of elements in the eigenvector weigh the importance of the original features for the corresponding principal component, while the sign of the elements indicates the correlation among the features. We use a biplot [9] to interpret PCA results, although it is limited to top two or three features on the plot. Points shown in a 2D plot are observations represented by the top two dimensions of PCA, and lines reflect the projections of the original features on to the new space. The length of the lines approximates the importance of the features. To avoid a large number of overlaps, we only show lines of features whose elements in the eigenvector have absolute values that are larger than 0.2.

For the nonlinear transformation methods, the reduced dimension has been explained in the case of datasets such as visual perception, movement and handwriting

Table 2 Intrinsic dimension using PCA

Dataset	d^{10}	d^{5}	d^{1}
First	21	26	36
FirstTriples	9	12	13
Wind115 and Wind150	5	8	15
Remote	2	4	11

[35]. The data points are displayed as images that are interpolations along straight lines in the representing coordinate space. This task becomes impossible for scientific datasets that are not necessarily images and consist of a large number of features extracted from low-level data. Hence, we are limited to evaluating the linear correlation between the projected dimensions and the original features to gain insights into the data.

7 Experimental Results and Discussion

We next present the experimental results for the four feature selection methods and the five transform methods on the datasets from the three problem domains. For the low-dimensional data representations using the four NLDR techniques, we experimented with several parameter settings. Isomap, LLE and Laplacian Eigenmaps have a parameter k or ϵ, depending on whether we consider the k-nearest neighbors or an ϵ-neighborhood. Laplacian Eigenmaps has an additional parameter t used in the Gaussian kernel. LTSA has only a parameter k, but requires a determination of d before applying the method. We tested $k = 3, 5, 7, \ldots, 29$; ϵ that ranges from 1.2 to 20.0; and $t = 1, 5$ and 10. We then obtained the percentage error rates for the decision tree ensemble classifier as outlined earlier in Sect. 6. The same approach was used for the four feature subset selection methods, where we obtained the percentage error rate using the first d features. In the classification error rate plots presented in the rest of this section, we include the best results for each of the four NLDR methods, the results for PCA and the four feature subset selection methods, and the error rate for the decision tree ensemble applied to the original input data with all features, which is displayed as a constant horizontal line on the plots.

Table 2 summarizes the intrinsic dimension estimation using eigenvalue spectrum of PCA on all datasets. They are discussed together with the results of all other intrinsic dimensionality estimations in the following sections.

7.1 Experiments on First and FirstTriples Datasets

Figure 4 presents the classification accuracy of dimension reduction methods applied on the *First* dataset. We observe that using the reduced representations is not guaranteed to provide better classification performance than using the original input

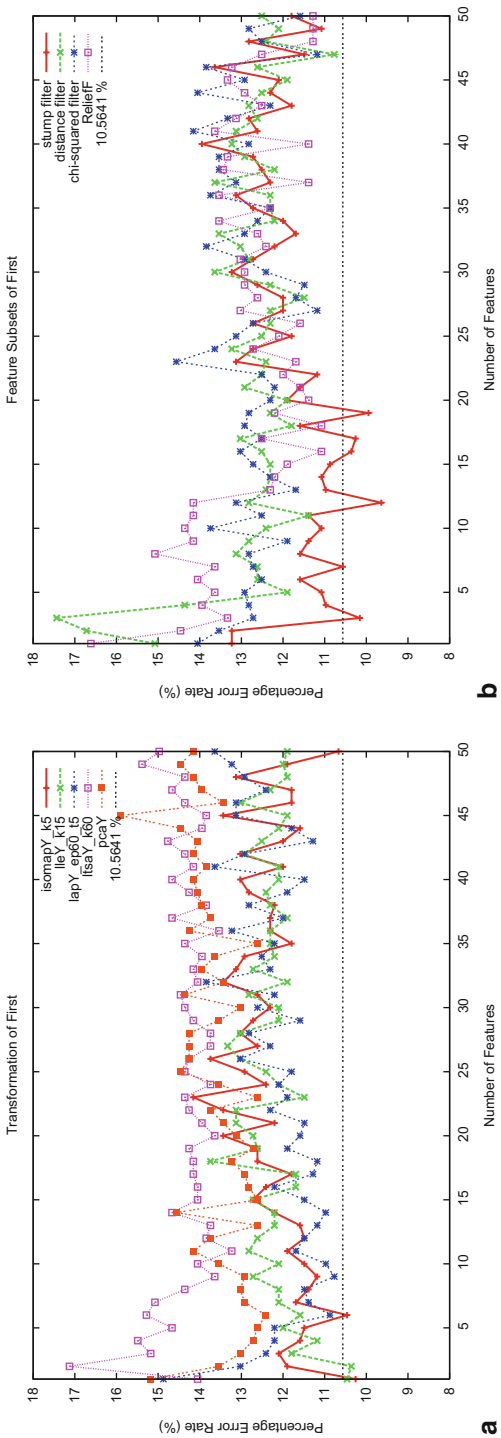


Fig. 4 Classification error rates using decision tree classifiers on the transformed features (*left*) and the selected features (*right*) for the *First* dataset

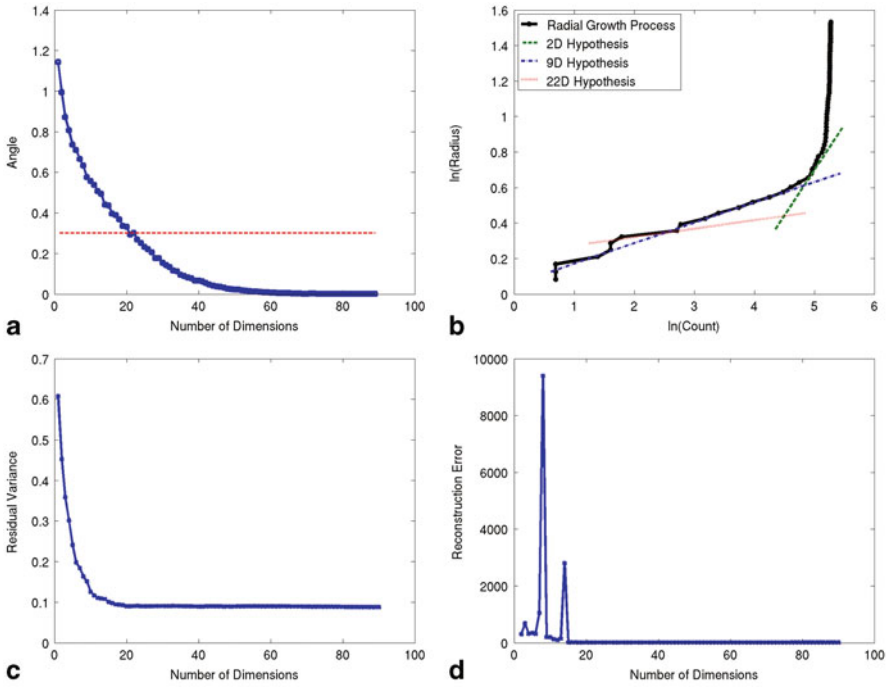


Fig. 5 Intrinsic dimensionality estimation on *First* dataset. (a) Statistical approach. ($d = 21$). (b) Locally linear scale. ($d \approx 9 \sim 22$). (c) Isomap with $k = 5$. ($d \approx 18$). (d) LLE with $k = 15$. ($d \approx 15$).

data. Among the data transformation methods, only the representation of Isomap with $k = 5$ and LLE with $k = 15$ gives a smaller error rate than the original input data when using the first few features. On the other hand, in the results with the feature subset selection methods, only the stump filter gives error rates below the horizontal line, indicating an improvement over using all original features. Its best performance is better than the data transformation methods.

Figure 5 shows the intrinsic dimensionality of the *First* dataset estimated by the four different methods. This dataset contains 90 features and there is some variation in the estimates. In Fig. 5a, the angles obtained by the statistical approach are plotted against the number of dimensions. The dotted line is a threshold; we estimate the dimension as the value where the angle falls below the threshold, that is at $d = 21$. This is the same as PCA d^{10}. The locally linear scale in Fig. 5b indicates that the intrinsic dimensionality falls approximately between 9 and 22 dimensions. Using the elbow test on residual variances of Isomap, the estimate is about 18. The plot of reconstruction error in Fig. 5d obtained by LLE does not have an elbow shape, making it difficult to identify the intrinsic dimensionality.

In contrast to *First*, the results for *FirstTriples* shown in Fig. 6 indicates that PCA, Isomap, LLE, and Laplacian Eigenmaps improve the error rates. The best performance is obtained using the top 2 features from PCA and from Isomap, top 12

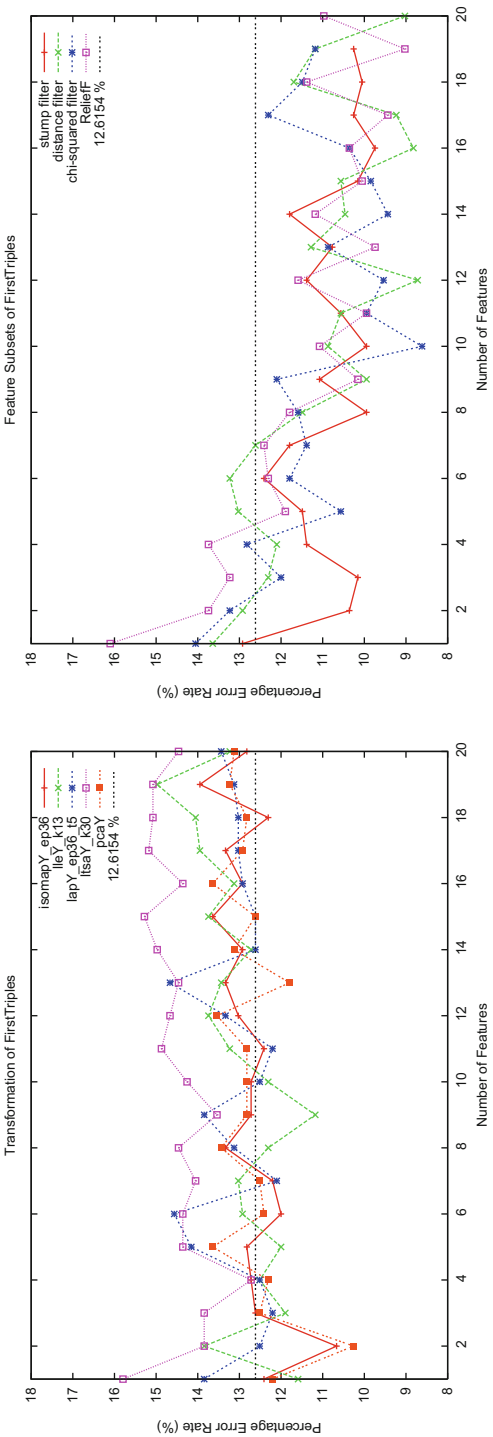


Fig. 6 Classification error rates using decision tree classifiers on the transformed features (*left*) and selected features (*right*) for the *FirstTriples* dataset

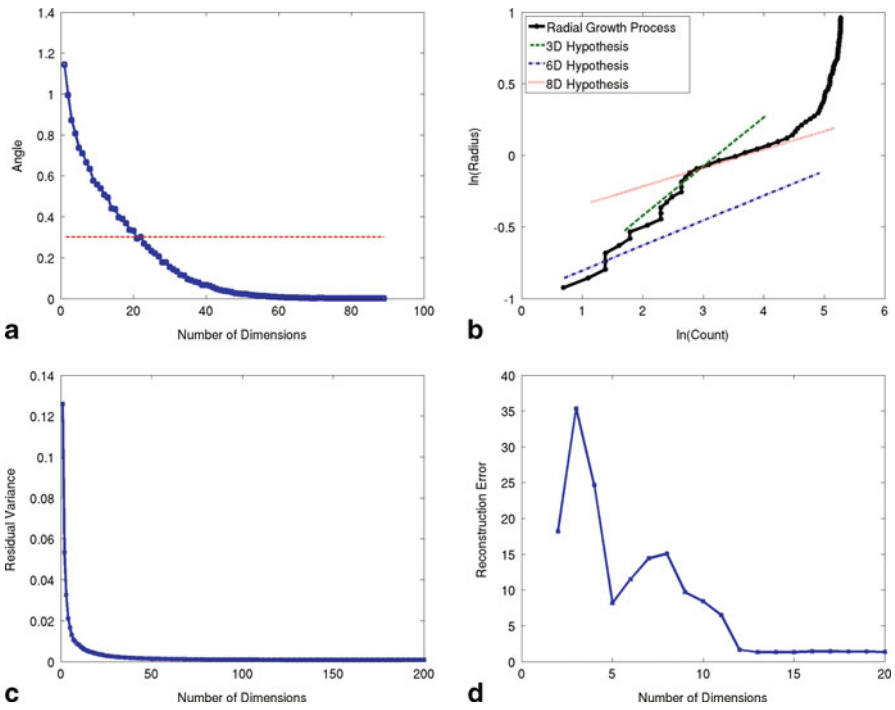


Fig. 7 Intrinsic dimensionality estimation on *FirstTriples* dataset. (a) Statistical approach. ($d = 9$). (b) Locally linear scale. ($d \approx 8$). (c) Isomap with $\epsilon = 3.6$. ($d \approx 9$). (d) LLE with $k = 12$. ($d \approx 12$).

features from LLE, and top 7 features from Laplacian Eigenmaps. However, none of the NLDR techniques perform better than the four feature subset selection methods. In addition, since the *FirstTriples* dataset is derived from *First*, and all methods give lower error rates on *FirstTriples*, it indicates that the dataset is less noisy, confirming the scientists expectation.

We observe that the *FirstTriples* dataset, with fewer features, has a smaller variance in the estimation of the intrinsic dimensionality in comparison to the *First* dataset. This may due to the small ratio of the set cardinality n to the number of dimensions D. In order to obtain an accurate estimation of the dimensionality, it has been proven that the inequality $D < 2\log_{10} n$ should be satisfied [34]. The number of data points needed to accurately estimate the dimension of a D-dimensional data set is at least $10^{\frac{D}{2}}$. So, in practice, if the sample size of a dataset is small, we should try reducing the number of features using domain information prior to determining its intrinsic dimensionality (Fig. 7).

Figure 8 is a biplot of the *First* dataset. Feature CoreAngl (indicated as 8 in the plot) has positive projection on to both the first and the second dimensions. There is a negative correlation between CoreAngl (8) and a subset of features related to angles (9, 10 and 13) and symmetry (20). This subset of features have negative projection on to both the first and the second dimensions. The bents observations are clustered at low values of the distance features (45, 16, 90 and 33). The non-bents observations

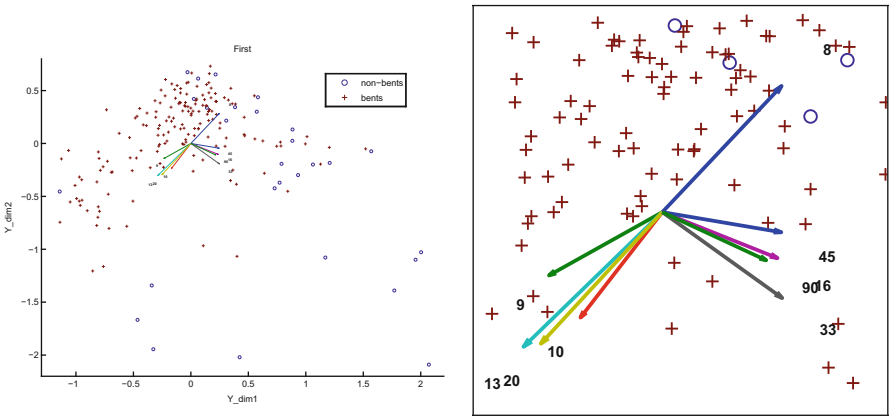


Fig. 8 Left: PCA biplot of the *First* dataset. Right: zoomed-in view

forms a cluster at the near highest CoreAngl (8) values, near lowest symmetries (20) and near lowest other angle values (at 9, 10 and 13) as well as the high distances (at 45, 16, 90 and 33). We may interpret both the first and second PC dimensions as distance-angle dimensions. These observations support the visual labeling process used by astronomers, where symmetry is an important feature of bent-doubles, and angles are an important discriminating feature.

Figure 9 shows the linear correlation between Isomap dimensions and the original features. Seven out of the top 20 features that are highly correlated to the first Isomap dimension are also among the top 9 features rated highly by PCA. Although many features are highly correlated to the second Isomap dimension, none of them are among the top PCA features. These are the linear relationships that we can explain. Nonlinear relationships among the features are still unknown. In the decision tree classification, using only one dimension of Isomap can give a better classification performance than the original dataset. It indicates that the first dimension of Isomap captures a property of the data that reflects the class labels.

The feature subset selection methods rank highly the features of the *First* dataset which are related to symmetries and angles, consistent with what PCA has captured for the top two PCs.

Figure 10 displays the PCA biplot of *FirstTriples*. Seven out of the nine features whose elements are significant in either one of the top two eigenvectors, are among those that PCA chooses for the *First* dataset. These features are also consistent with the highly ranked features from filter methods. The result emphasizes that PCA can be a good measure for removing noise features, although the PCA representations of *First* data do not improve the classification.

There is again a negative correlation between feature CoreAngl (8) and a subset of features related to angles (9, 10, 13 and 14) and symmetry (20). These features are parallel to the first PC coordinate, which tells us that the first dimension is an angle coordinate. We can also see clusters of non-bents fall around the extreme values

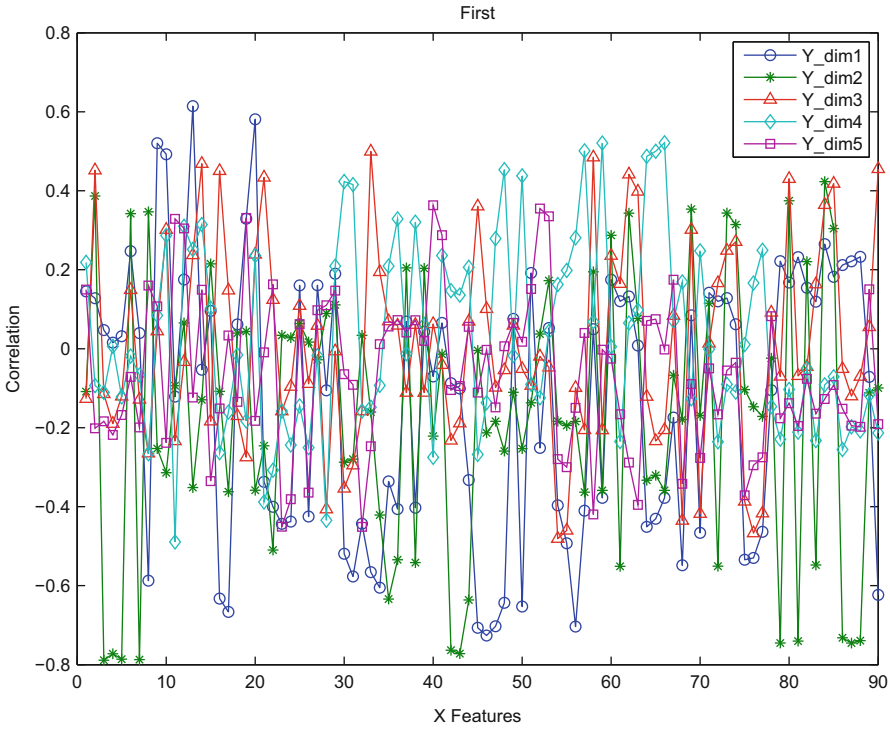


Fig. 9 Correlation between top five Isomap reduced dimensions and all original features for *First* dataset

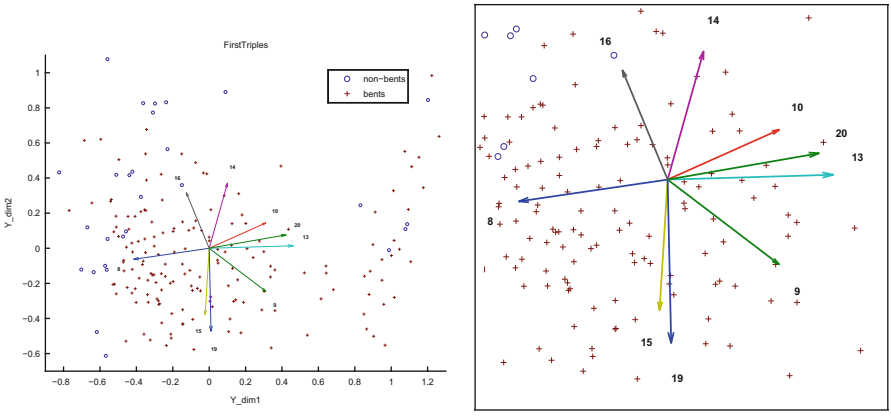


Fig. 10 Left: PCA biplot of the *FirstTriples* dataset. Right: zoomed-in view

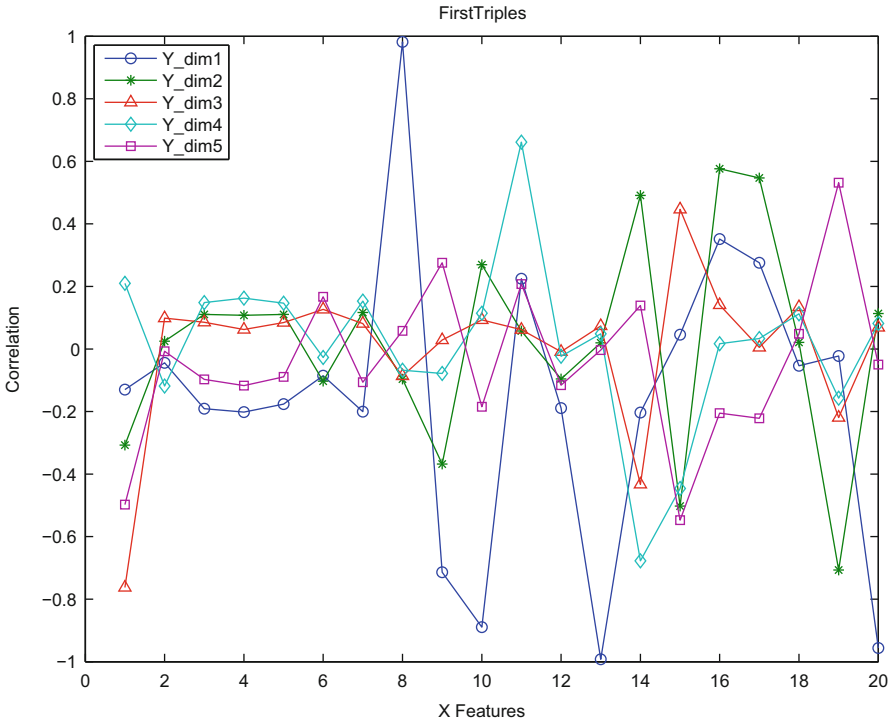


Fig. 11 Correlation between top five Isomap reduced dimensions and all original features for *FirstTriples* dataset

of angle features, while clusters of bents have medium angle values. The second dimension represents the feature #19 (AriSym, one of the symmetries) v.s. feature #14 (ABAngleSide, one of the angles) and feature #16 (SumComDist, one of the distances). We can see a cluster of bents that has small values of feature #16, large values of feature #19 and medium values of angles. There is also a non-bent cluster at the near highest CoreAngl (8) values, near lowest symmetries (20) and near lowest other angle values (at 9, 10 and 13). This fact is similar to what PCA gets from *First* dataset. The two PCs together shows that there exist no non-bents at the corner area where the AriSym (19) is high and CoreAngl (8) is low.

Figure 11 shows that Isomap dimensions have linear correlations with a subset of features that are similar to the PCA dimensions. The first dimension of Isomap has significantly linear correlations with a subset of features. Similar to PCA, in the first dimension there is a negative correlation between feature CoreAngl (8) and a subset of features related to angles (9, 10 and 13) and symmetry (20). The second dimension of Isomap tells its linear correlation with feature #19 (AriSym), which is also a main feature in the second dimension of PCA. In the decision tree classification shown in Fig. 6, both PCA and Isomap obtain best classification performance using their top two features. This similarity in performance of PCA and Isomap strengthens the

possibility that Isomap captures the linear properties in the *FirstTriples* dataset, and it is unlikely there is a nonlinear manifold underlying the data.

Finally, we observe that the subset selection methods, PCA, and Isomap all selected features of the *FirstTriples* dataset related to symmetries and angles.

7.2 Experiments on Wind115 and Wind150 Datasets

Figure 12 displays the classification results for the *Wind115* dataset. It is significant that the feature subset selection techniques outperform the data transformation methods. All data transformation techniques do not reach the accuracy of the original data.

The performance on *Wind150* dataset, which is labeled differently, is shown in Fig. 13. We observed that all methods give lower error rates than on *Wind115*, which indicates that labeling the data according to 150 MW ramps can help identify events more significantly. Again, the feature selection methods outperform the data transformation techniques. Isomap and PCA are the only two data transformation techniques that, in comparison to the original data, slightly improve the classification. The best performance of PCA is at $d = 9$, and the best of Isomap is at $d = 15$ and $\epsilon = 3.2$ (displayed as isomapY_ep32). At $d = 14$ the chi-squared filter reaches its lowest error rates, and at $d = 18$ the distance filter has the lowest error rate. However, we observe that when the number of features is less than 9, the data transformation methods are more accurate than the feature selection methods.

Both *Wind115* and *Wind150* are the same dataset, but with different labeling criteria. Hence, they have the same intrinsic dimensionality shown in Fig. 14. The estimate of statistical approach is $d = 11$, while locally linear scales give $d = 4$. The elbow test on residual variances of Isomap gives $d \approx 9$, close to the statistical approach. The dimensionality according to the elbow test on reconstruction error of LLE gives $d \approx 10$ to 15. All are near the range of $d \approx 5$–15 estimated by PCA.

The PCA biplot of *Wind150* shown in Fig. 15 shows that the first coordinate has two subsets of features that are negatively correlated. One is the humidity features at three weather sites (7, 14, 21). The other subset contains the temperature features (6, 13, 20) and the solar radiation features (2, 9, 16) at three weather sites. The second principal component is about wind direction vector (4, 11, 18) and speed (10, 12, 17, 19) that are positively correlated.

There are two clusters that are dense. One contains observations that have low wind speeds and low wind direction vector degrees. The other cluster is at high temperature, high solar radiation and low humidity. These characters represent non-ramp events, which are consistent with the labels shown on the graph. There exist no clusters of ramp events that are obviously dense.

The linear correlation between the first five Isomap dimensions and all original features for *Wind150* is shown in Fig. 16. Like PCA, the first dimension of Isomap is linearly correlated to features of humidity, temperature, and solar radiation at three weather sites. Similarly, the second dimension of Isomap is linearly correlated

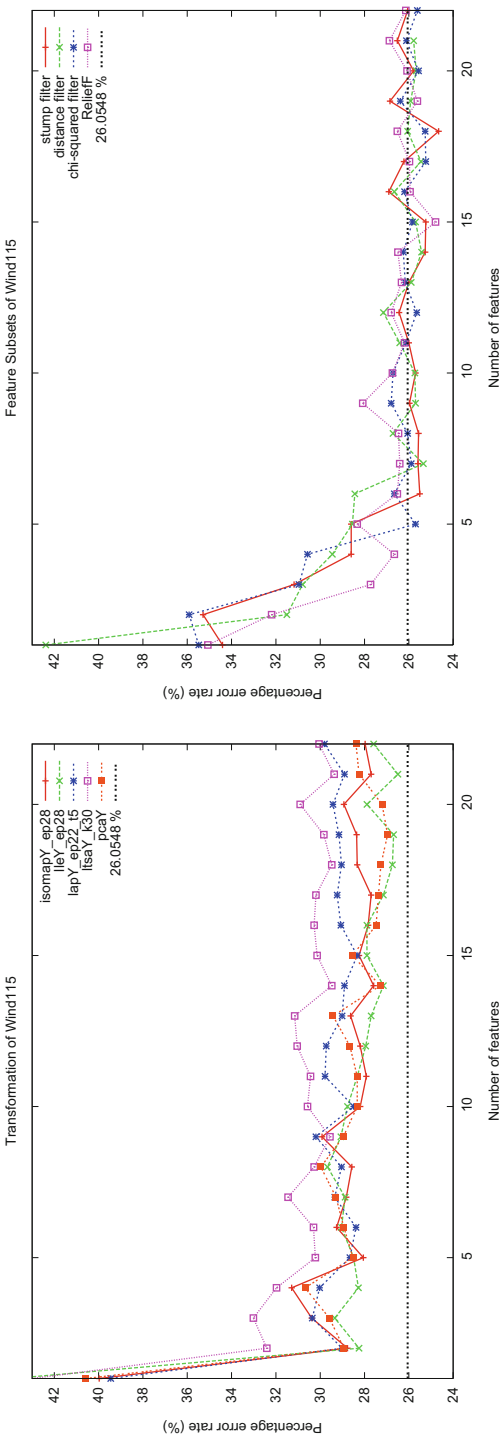


Fig. 12 Classification error rates using decision tree classifiers on the (*left*) transformed features and (*right*) selected features for the *Wind115* dataset

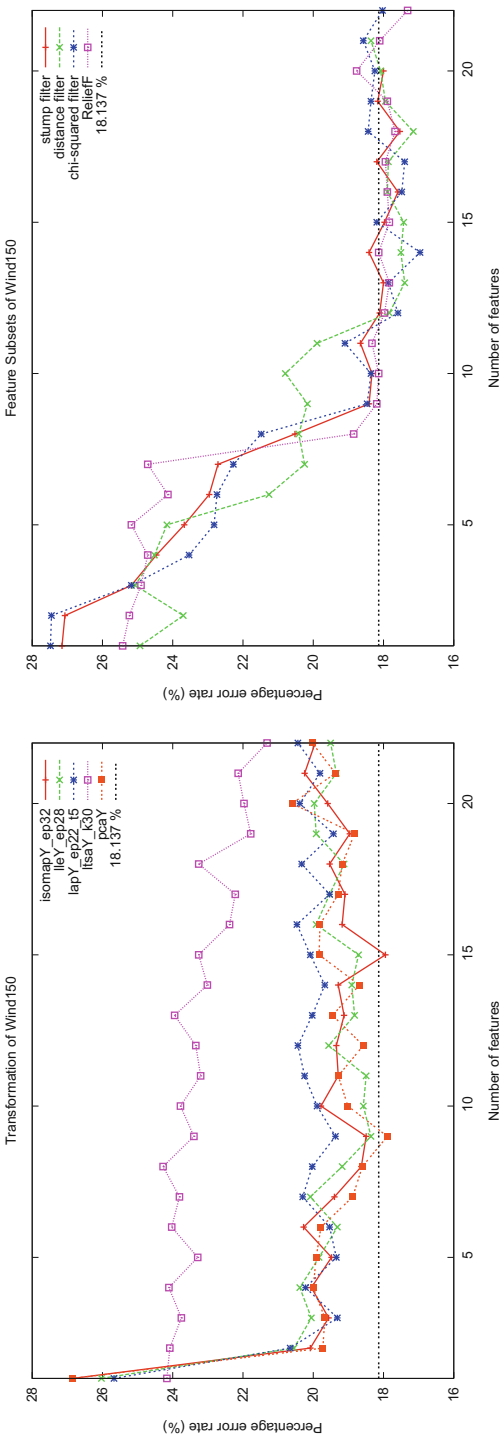


Fig. 13 Classification error rates using decision tree classifiers on the (*left*) transformed features and (*right*) selected features for the *Wind150* dataset

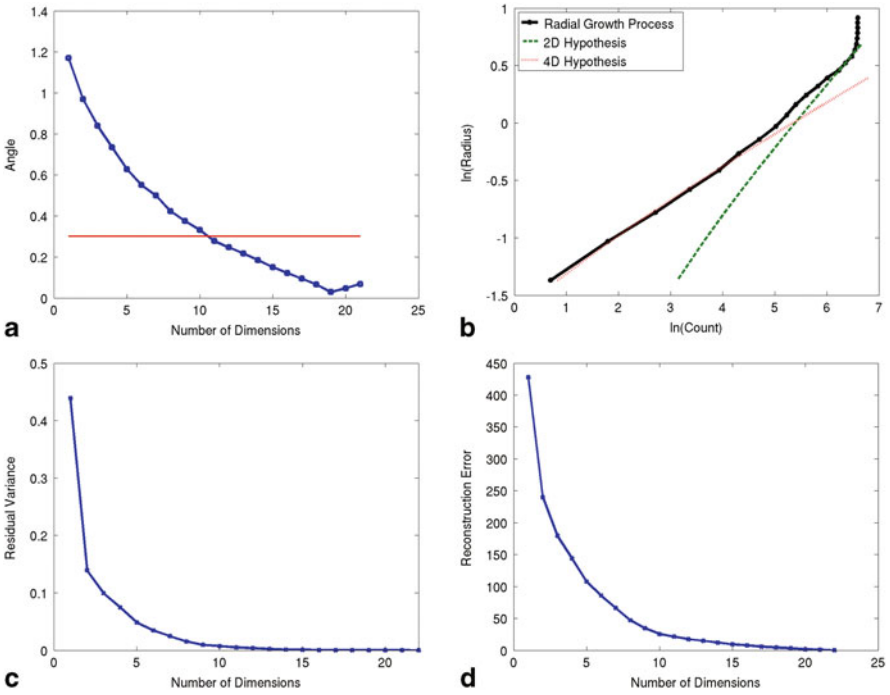


Fig. 14 Intrinsic dimensionality estimation on *Wind115* dataset. (a) Statistical approach. ($d = 11$). (b) Locally linear scale. ($d = 4$). (c) Isomap with $\epsilon = 3.2$. ($d \approx 9$). (d) LLE with $\epsilon = 3.2$. ($d \approx 10 \sim 15$).

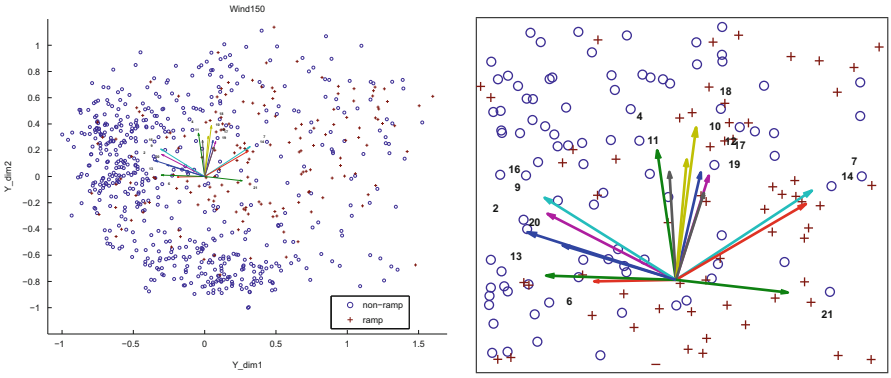


Fig. 15 *Left*: PCA biplot of the *Wind150* dataset. *Right*: zoomed-in view

to wind direction vector degrees and speeds. This implies that the isomap captures the linear relations of the data. However, it is not straight-forward to determine the existence of any nonlinear relations.

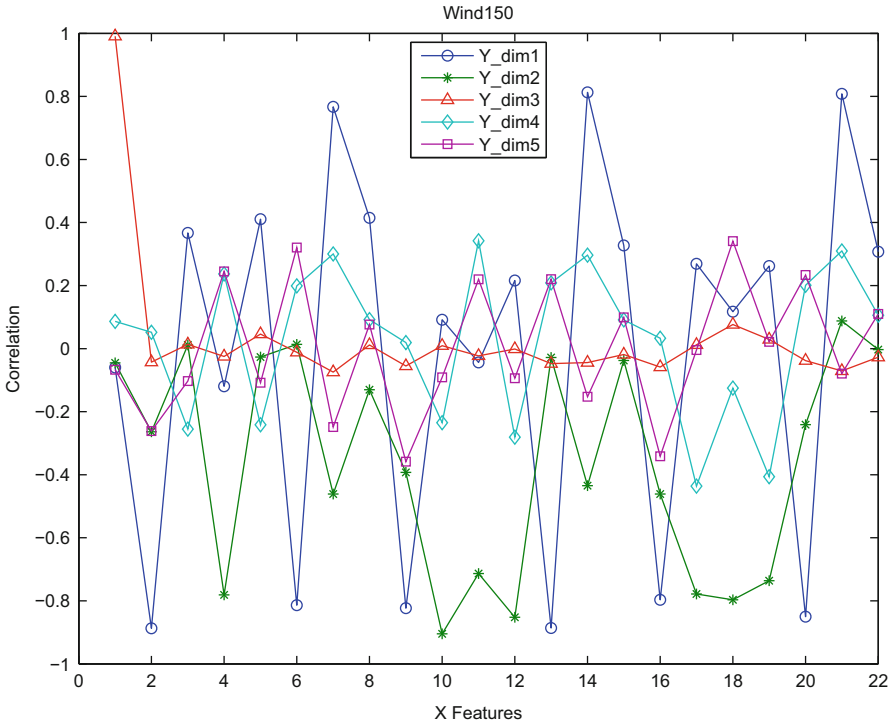


Fig. 16 Correlation between top five Isomap dimensions and all original features for *Wind150* dataset

Finally, the top six common features that are ranked highly by all three filters are also the wind speed, temperature, and humidity. This consistency shows the success of filters, PCA, and Isomap in dimension reduction.

7.3 Experiments on Remote Dataset

Figure 17 shows the classification error rates for the *Remote* dataset. Only 50 of the 496 features are displayed because the rates become almost constant when large numbers of features are used for all methods. Though the feature subset selection methods still outperform the data transformation techniques, all methods perform better than using all features. This could be due to the actually high-dimensional data with a large number of samples ($n = 2000$). Thus, the lower dimensional structure exist and the data transformation methods can find them. Isomap, LLE and PCA have similar performance and reach the minimum error rate at 6–9 dimensions. In contrast, Laplacian Eigenmaps reaches its best performance at $d = 34$ and LTSA at $d = 26$.

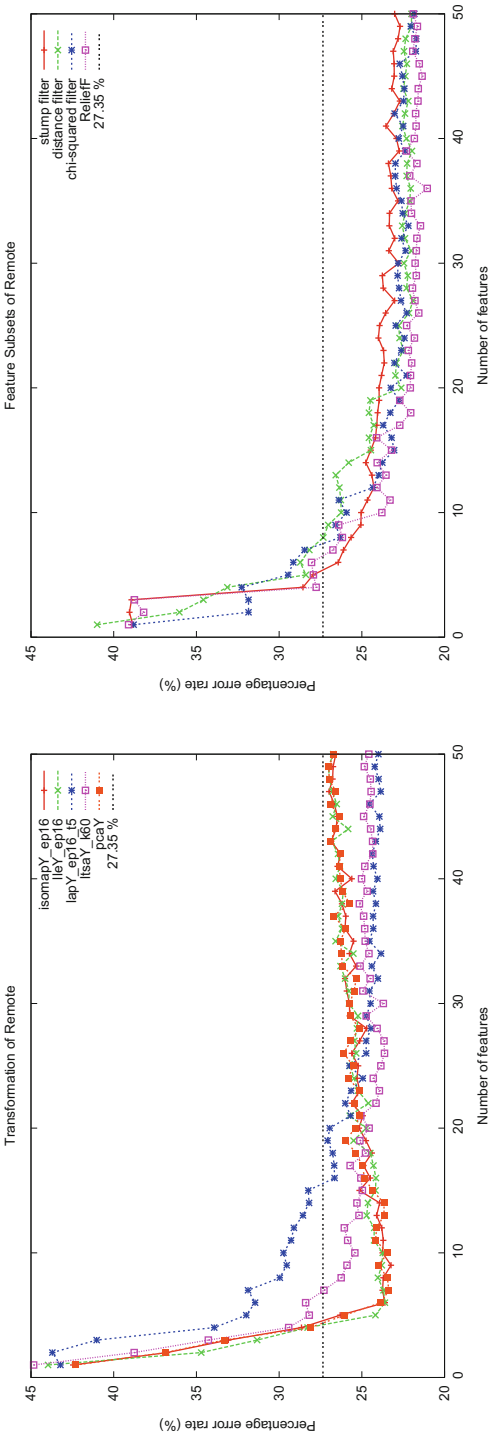


Fig. 17 Classification error rates using decision tree classifiers on the (*left*) transformed features and (*right*) selected features for the *Remote* dataset

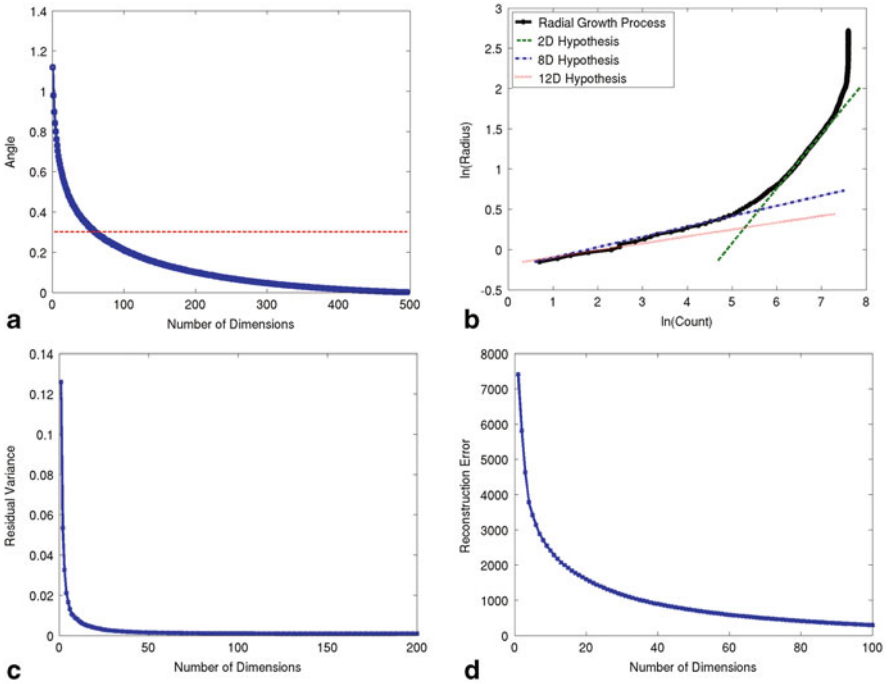


Fig. 18 Intrinsic dimensionality estimation on *Remote* dataset. (a) Statistical approach. ($d = 60$). (b) Locally linear scale. ($d = 8$). (c) Isomap with $\epsilon = 16$. ($d \approx 6$). (d) LLE with $\epsilon = 16$.

For *Remote* dataset, the statistical approach gives an intrinsic dimensionality estimate of $d = 60$, which is quite different from $d = 8$ estimated using locally linear scale. PCA gives small numbers of estimation as well. Elbow test on Fig. 18c shows that $d = 6$ is right below the cliff and the flat region begins at around $d = 20$. Combining the results given in Figs. 17 and 18c, we can see that Isomap with $\epsilon = 16$ gives the minimum error rate at dimensionality close to the estimate of $d \approx 6$. LLE reconstruction error seems not a reliable indicator for estimating intrinsic dimensionality.

In Fig. 19, the PCA biplot of *Remote* shows a large number of features that project observations in the first two PCs. Features pointing to the bottom right are all features from entropy in GLCM category, and most are green bands. Features pointing to the left are all features from inverse difference moment in GLCM category with green, blue and red bands. These two subsets of features are negatively correlated. They determine the first coordinate. Most features that point to the top are features from Gabor and wavelet categories. They are all features of near-infrared bands. Features pointing to the bottom are features of GLCM category with near-infrared, green, blue and red bands. They can be used to explain the second PC. The observations form a funnel on the plot, indicating that one dimension affects the variance of another orthogonal dimension. It means that high GLCM values are similar in their entropy and inverse difference moment, while low GLCM values are more varied.

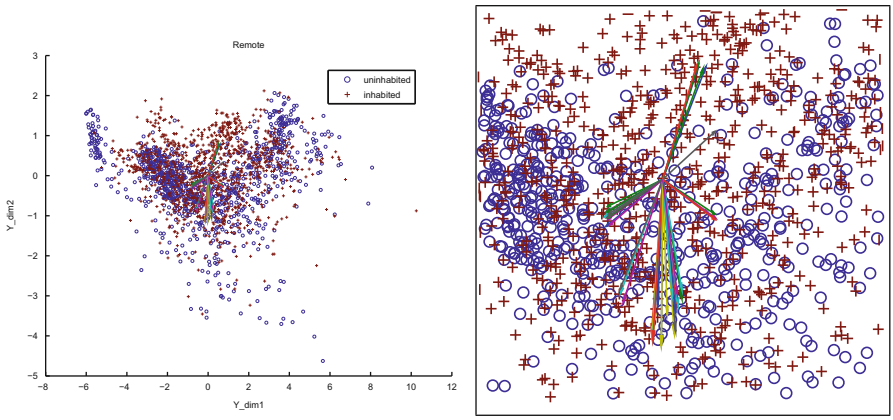


Fig. 19 *Left*: PCA biplot of the *Remote* dataset. *Right*: zoomed-in view. Vectors on the graph are 10 times larger than their original sizes for easier identification

The feature selection methods rank highly the features in the green and near-infrared bands rather than the blue and red bands. The majority of the top ten features are from the GLCM category, while the wavelet and Gabor features are selected less frequently. Power spectrum features are rarely selected. The GLCM features selected most often in top ten are entropy and inverse difference moment. These results agree with what PCA suggests. The linear correlations between Isomap dimensions and the original features are again similar to PCA.

8 Related Work

In this article, we have focused on a few popular data transformation methods for dimension reduction: PCA, Isomap, LLE, Laplacian Eigenmaps and LTSA. Many other techniques have also been proposed, including structure preserving embedding [33], maximum variance unfolding [40], Hessian eigenmaps [6], neighborhood preserving methods [13, 21], and diffusion maps [4], as well as techniques that reduce the data to two dimensions for visualization, such as t-distributed stochastic neighbor embedding (tSNE) [23], self-organizing maps [20], and neural network-based approaches [12].

Much of the work in NLDR techniques has focused on the algorithmic aspects, with experiments on artificial datasets illustrating the benefits of these methods. However, a recent comparative study [24] on several NLDR techniques applied to both artificial and real datasets concluded that the strong performance of these techniques on the artificial Swiss roll data does not generalize to more complex, artificial datasets, such as those with disconnected manifolds or manifolds with high intrinsic dimensionality. In addition, most nonlinear techniques do not outperform PCA on real datasets. Another comparative study of dimension reduction techniques

[38] also shows that for data visualization purposes, NLDR techniques generally perform better on the synthetic data than on the real-world data, and the overall best performing algorithm is Isomap.

Our study does not focus on data visualization, but on practical scientific data analysis. The experiments presented in this article also support the conclusions from these comparative studies. However, there are successful applications of NLDR on real world datasets [27], and methods, such as *t*SNE, when used for visualization, have been shown to provide insights into the inherent structure in high-dimensional data [23]. It appears that the best NLDR technique depends on the nature of the input data and on the use of the reduced representation [27].

9 Conclusions

In this article, we describe a series of carefully-designed experiments that test, in a useful and impartial manner, how dimension reduction methods work in practice. We investigate two types of techniques: data transformation methods and feature subset selection techniques. Using classification problems in five scientific datasets, each exhibiting different data properties, we compare the error rates for the original dataset with those obtained for the reduced representations resulting from the data transformation methods as well as feature selection techniques. We also evaluate the intrinsic dimensionality of the data using estimates obtained from PCA and two of the NLDR methods (Isomap and LLE), in addition to two classical techniques, one based on a statistical approach and the other on a locally linear scale.

Our experiments indicate that, while the supervised feature subset selection techniques consistently improve the classification of all datasets, the data transformation methods do not. However, it is possible to use them to find properties of the data related to class labels. Our experiments show that both PCA and Isomap are able to find representations that improve data classification. Since both PCA and Isomap employ the eigenvectors corresponding to the largest eigenvalues, they seem to perform better than methods which use the eigenvectors corresponding to the smallest non-zero eigenvalues, such as LLE, Laplacian Eigenmaps, and LTSA. This result is consistent with the comparative study in [24]. Like PCA, when the data tend to have strong linear properties, Isomap can identify these properties. Isomap can also capture some kind of nonlinear properties that PCA cannot find. Although there exists applications indicating that PCA is better than Isomap in terms of classification [24], our experiments indicate a different conclusion. We also observe that the ability to interpret the reduced dimension made by data transformation methods is very limited.

Since feature subset selection techniques are computationally inexpensive, we suggest using them first, especially as they could provide insights into the dataset by indicating which of the original features are important. If a dataset contains noise features, the use of feature subset selection techniques to identify and remove possible noise features prior to the application of the data transformation methods could also

be helpful. If the sample size of a dataset is small, we should try reducing the number of features using domain information prior to determining its intrinsic dimensionality. Among the feature subset selection techniques, the filter-based methods give more consistent results. The estimation of intrinsic dimensionality of the dataset may vary, depending on the method used. However, the estimate could be meaningful if it is close to the number of features that give the best performance. For an NLDR method, this may also imply that the method finds the lower-dimensional manifold on which the data lie, something which is not possible with linear feature subset selection.

Acknowledgments LLNL-TR-531131. This work performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. We thank the domain scientists for sharing their data and the reviewers for their insightful comments.

References

1. Becker, R.H., White, R.L., Helfand, D.J.: The FIRST survey: Faint images of the Radio Sky at Twenty-cm. Astrophys. J. **450**, 559 (1995)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. CRC, Boca Raton (1984)
4. Coifman, R.R., Lafon, S.: Diffusion maps. Appl. Comput. Harmonic Anal. **21**(1), 5–30 (2006)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
6. Donoho, D.L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proc. Natl. Acad. Sci. **100**(10), 5591–5596 (2003)
7. Floyd, R.W.: Algorithm 97: Shortest path. Commun. ACM. **5**, 345 (1962)
8. Fukunaga, K., Olsen, D.: An algorithm for finding intrinsic dimensionality of data. IEEE Trans. Comput. **C-20**(2), 176–183 (1971)
9. Gabriel, K.R.: The biplot graphic display of matrices with application to principal component analysis. Biometrika **58**(3), 453–467 (1971)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Machine Learn. Res. **3**, 1157–1182 (2003)
11. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **3**, 610–621 (1973)
12. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall PTR, Upper Saddle River (1998)
13. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: 10th IEEE International Conference on Computer Vision, vol. **2**, pp. 1208–1213 (2005)
14. Huang, S.H.: Dimensionality reduction on automatic knowledge acquisition: A simple greedy search approach. IEEE Trans. Knowl. Data Eng. **15**(6), 1364–1373 (2003)
15. Kamath, C.: Associating weather conditions with ramp events in wind power generation. In: Power Systems Conference and Exposition (PSCE), IEEE/PES, pp. 1-8, 20-23 (2011). http://ckamath.org/publications_by_project/windsense. Accessed date March 2011
16. Kamath, C., Cantú-Paz, E., Fodor, I.K., Tang, N.: Searching for bent-double galaxies in the first survey. In: Grossman, R., Kamath, C., Kegelmeyer, W.P., Kumar, V., Buru, R.N. (eds.) Data Mining for Scientific and Engineering Applications, pp. 95–114. Kluwer, Boston (2001)

17. Kamath, C., Cantú-Paz, E., Littau, D.: Approximate splitting for ensembles of trees using histograms. In: Proceedings, 2nd SIAM International Conference on Data Mining, pp. 370–383 (2002)
18. Kegl, B.: Intrinsic dimension estimation using packing numbers. In S. Becker, S. Thrun, and K. Obermayer, editors, Advances in Neural Information Processing Systems 15, Cambridge, MA, MIT Press (2003)
19. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif. Intell. **97**(1–2), 273–324 (1997)
20. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biol. Cybern. **43**, 59–69 (1982)
21. Kokiopoulou, E., Saad, Y.: Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12), 2143–2156 (2007)
22. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, New York (2007)
23. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Machine Learn. Res. **9**, 2579–2605 (2008)
24. van der Maaten, L., Postma, E., van den Herik, J.: Dimensionality reduction: a comparative review. Tech. Rep. TiCC TR 2009–005, Tilburg University (2009)
25. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. IEEE Trans. Pattern Anal. Mach. Intell. **18**(8), 837–842 (1996)
26. Newsam, S., Kamath, C.: Retrieval using texture features in high-resolution, multi-spectral satellite imagery. In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology, VI, Proceedings of SPIE, vol. 5433, pp. 21–32. SPIE Press (2004)
27. Niskanen, M., Silvén, O.: Comparison of dimensionality reduction methods for wood surface inspection. In: Proceedings of the 6th International Conference on Quality Control by Artificial Vision, pp. 178–188 (2003)
28. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Phenomenol. **2**(6), 559–572 (1901)
29. Robnik-Sikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learn. **53**, 23–69 (2003)
30. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
31. Sabato, S., Shalev-Shwartz, S.: Ranking categorical features using generalization properties. J. Machine Learn. Res. **9**, 1083–1114 (2008). http://dl.acm.org/citation.cfm?id=1390681.1390718. Accessed date June 1, 2008
32. Saul, L.K., Roweis, S.T., Singer, Y.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. J. Machine Learn. Res. **4**, 119–155 (2003)
33. Shaw, B., Jebara, T.: Structure preserving embedding. In: Proceedings of the 26th International Conference on Machine Learning (2009)
34. Smith, L.A.: Intrinsic limits on dimension calculations. Phys. Lett. A **133**(6), 283–288 (1988)
35. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
36. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B **58**(1), 267–288 (1996)
37. Trunk, G.V.: Statistical estimation of the intrinsic dimensionality of a noisy signal collection. IEEE Trans. Comput. **C-25**(2), 165–171 (1976)
38. Tsai, F.S.: Comparative study of dimensionality reduction techniques for data visualization. J. Artif. Intell. **3**(3), 119–134 (2010)
39. Valle, S., Li, W., Qin, S.J.: Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. Ind. Eng. Chem. Res. **38**(11), 4389–4401 (1999)

40. Weinberger, K., Saul, L.K.: An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In: Proceedings of the National Conference on Artificial Intelligence, pp. 1683–1686. Boston, MA (2006)
41. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimension reduction via local tangent space alignment. SIAM J. Sci. Comput. **26**, 313–338 (2002)
42. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B **67**(2), 301–320 (2005)

Relearning Process for SPRT in Structural Change Detection of Time-Series Data

Ryosuke Saga, Naoki Kaisaku and Hiroshi Tsuji

Abstract This study proposes a relearning process for a prediction model after detecting structural change points. There are three problems with the detection of structural change points in time-series data: (1) how to generate a prediction model, (2) how to detect a structural change point rapidly, and (3) how the prediction model should relearn after detection. This article targets the third problem and proposes five relearning methods and a process that embeds the relearning process in the sequential probability ratio test. Two experiments, one using 20 generated data sets and the other TOPIX, which consists of 1104 time-series data points between 1991 and 2012, show that using past and future data after detecting the structural change points is helpful.

1 Introduction

Detecting structural change points in time-series variations is important. For example, we can discover sales and management strategies by recognizing the structural change points. Moreover, we can identify the trend changes in stock and exchange markets by detecting the structural changes. In network monitoring, there is a growing need to predict and act promptly after changes in throughput and network load have been detected.

The change points confirm trend conversions and occurrences of abnormal status. Similarly, the idea of structural points has been applied to online learning (called stream mining) of classification and clustering such as in neural network, support vector machines and k-means [11, 13, 15, 16]. Thus, the rapid detection of structural change points has grown some importance over the years.

R. Saga (✉) · N. Kaisaku · H. Tsuji
Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Nakaku, Sakai, Osaka, Japan
e-mail: saga@cs.osakafu-u.ac.jp

N. Kaisaku
e-mail: kaisaku@cs.osakafu-u.ac.jp

H. Tsuji
e-mail: tsuji@cs.osakafu-u.ac.jp

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_7

There exist several methods to detect structural change points including Bayesian [6] and statistical test techniques. The basic idea is simple: detecting whether the structure changes at some point. The representative method is the Chow test. This method uses the F-test on time-series data and judges whether the data structure before some point is the same as that of the data after the point. Note that the Chow test needs to know the test point. To detect unknown structural points, sequential probabilistic ratio tests such as CUSUM [5] and MOSUM [8] have been used. These methods can be classified into three categories on the basis of the test concept proposed by Zeileis et al. [19]: (1) F statistics [7, 18], (2) fluctuation tests [5, 8], and (3) maximum likelihood scores [10, 14].

The problems regarding structural change points in time-series data are divided into three categories [3, 4]. The first is how to generate a prediction model to represent the characteristics of the time-series data. The second is how to detect a structural change point as quickly and correctly as possible in the time-series data [2, 9]. The third is how to rebuild the model after the structural change has been detected. This article targets the third problem.

The third problem is equivalent to how the existing model should be modified. For example, let us assume that we must separate a cluster during stream mining. Then, the way we separate and modify the cluster corresponds to the model modification. On the other hand, a certain amount of data may be required to accomplish the model modification. More specifically, we need to know how much data is required and how this data should be sampled, after a structural change is detected. Intuitively, it appears better to use the data for relearning after a structural change. However, it is possible to miss new change points for relearning resulting in an unreliable model. There is no research regarding the sampling process for rebuilding.

For these reasons, we target the third problem and propose and validate a relearning method on the basis of the sequential probability ratio test (SPRT), which is the basic method for detecting structural change points [12, 18]. We embed the relearning process in existing processes, apply the method to generated data and real-world data, and compare the detected results of no relearning and various relearning processes. Note that the prediction model of SPRT is generated using a regression model to define the problem regarding the amount of data that SPRT uses for relearning after detecting the structure change points. The remainder of this article is organized as follows. Section 2 gives an overview of the underlying technologies of the SPRT. In Sect. 3, we discuss the relearning model process for SPRT. Section 4 describes the two experimental environments used for the empirical analysis and analyzes the results. Finally, this study is concluded in Sect. 5.

2 Structural Change Detection Based on SPRT

2.1 Sequential Probability Ratio Test

SPRT is a classical statistical hypothesis test that is used to determine whether the ratio of the observed probability events is higher than a given reference value. There are several instances where SPRT can detect structural change more rapidly than

the Chow test and where SPRT does not require data distribution after detecting structural points, unlike Bayesian approaches.

In general, SPRT utilizes a pair of hypotheses according to the statistical hypothesis tests: null hypothesis H_0(e.g., the quality is under the prespecified limit of 1 %) and alternative hypothesis H_1 (e.g., the quality is above the prespecified limit of 1 %). SPRT calculates the likelihood ratio λ, which accumulates the probability ratio from the observed time-series data by using Eq. 1 when data can be monitored:

$$\begin{aligned}\lambda_i &= \frac{P(Z_1|H_1)P(Z_2|H_1)P(Z_3|H_1)\cdots P(Z_i|H_1)}{P(Z_1|H_0)P(Z_2|H_0)P(Z_3|H_0)\cdots P(Z_i|H_0)} \\ &= \lambda_{i-1}\frac{P(Z_i|H_1)}{P(Z_i|H_0)}. \end{aligned} \tag{1}$$

Here, $P(Z_i|H_0)$ indicates the probability of Z_i under the null hypothesis H_0 and $P(Z_i|H_1)$ denotes the probability of Z_i under the alternative hypothesis H_1. To detect structural change from the time-series data, null hypothesis H_0 attaches the significance that structural change does not occur, while alternative hypothesis H_1 indicates that structural change occurs. Note that "structural change does not occur" means that the probability of going beyond the error tolerance is below a probability θ_0, while "structural change occurs" means that the probability of going beyond the error tolerance is above a probability θ_1. Also $\theta_1 \gg \theta_0$

SPRT defines the termination condition by using the following steps:

1. if $\lambda_i < C_1$ then accept H_0
2. if $\lambda_i > C_2$ then accept H_1
3. otherwise (that is, $C_1 \leqq \lambda_i \leqq C_2$), continue monitoring.

Here, parameters $C_1 = \beta/(1-\alpha)$ and $C_2 = (1-\beta)/\alpha$; and α and β ($0 < \alpha < 1$, $0 < \beta < 1$) indicate Type I and II errors relatively and are used for significance level of test(generally $\alpha(\beta) \leq 0.05$). These parameters α, β, θ_0, and θ_1 need to be set before test.

2.2 *Process of Structural Change Detection by SPRT*

Figure 1 shows the detection of structural change by SPRT, and structural change is detected in details by using the following steps.

Step 1 Build a prediction model and set a tolerance band(a)
Build a prediction model using observed data, calculate the error distribution σ^2, and set the tolerance band. Here we assume that the prediction model has an error distribution in itself such as a regression model like $y_i = \Sigma_j \beta_j x_{ij} + C + \epsilon_j$

Step 2 Configure null hypothesis H_0 and alternative hypothesis H_1 and initialize parameters.

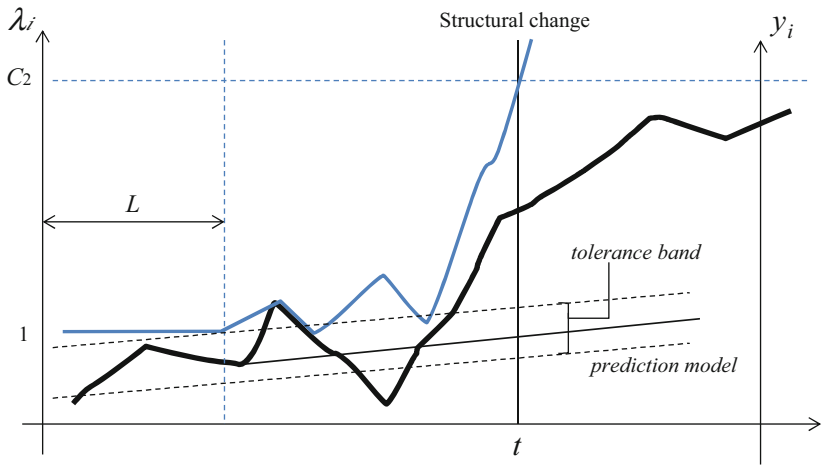


Fig. 1 An image of structural change by SPRT

Set two hypotheses H_0 and H_1, and initialize parameters i and λ_i =1 where i corresponds to the index of data (that is, y_i).

Step 3 Monitor data
Increment i and observe a new data y_i and an error ϵ_i. Here ϵ_i corresponds to Z_i in Eq. 1 (that is $Z_i = \epsilon_i$).

Step 4 Evaluate ϵ_i
Evaluate whether ϵ_i is within the tolerance band. If so, the process returns to the previous step.

Step 5 Calculate statistics value λ_i
Update λ_i by Eq. 1 according to whether the error is within the tolerance band. Here, if ϵ_i is within the tolerance band, then $P(Z_i|H_0)$ and $P(Z_i|H_1)$ are given as θ_0 and θ_1, respectively. If ϵ_i is out of the tolerance band, $P(Z_i|H_0) = 1 - \theta_0$ and $P(Z_i|H_1) = 1 - \theta_1$.

Step 6 Test
As we mentioned in Sect. 2.1, we test by using the following steps:

a. If λ_i is greater than $C_2(=(1-\beta)/\alpha)$, dismiss H_0, adopt H_1, and end the process.
b. If λ_i is less than $C_1(=\beta/(1-\alpha))$, dismiss H_1, adopt H_0, set λ_i=1, and return to Step 3.
c. Otherwise, progress to Step 7

Step 7 Continue monitoring
Increment i (i=i+1) and observe a new data y_i. Evaluate the ϵ_i and returns to Step 5.

This process does not to rebuild the prediction model even thought the structural change occurs. Therefore, we add a new step as Step 8 to the above process in order to relearn a predication model (Fig. 2).

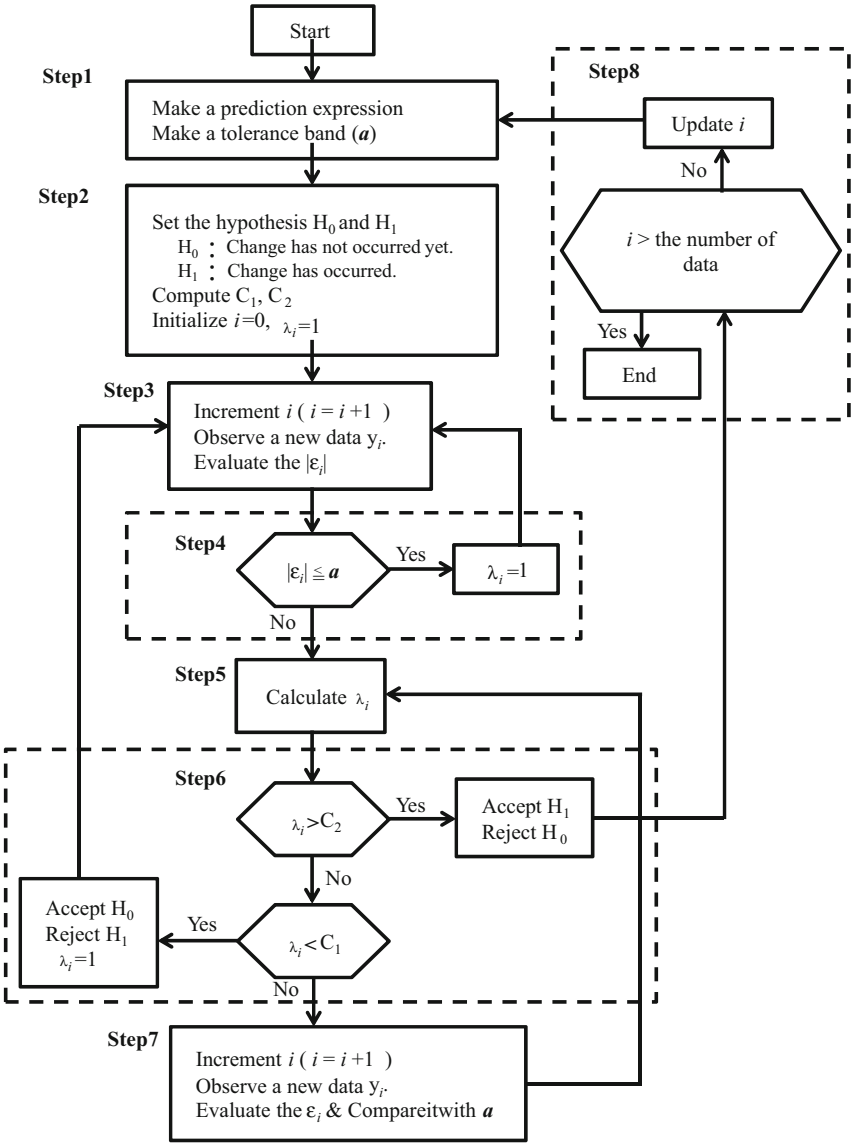


Fig. 2 Relearning process in detecting structural change by SPRT

3 Relearning Process Embedded in SPRT

Even though a structural change occurs, there is a probability that data continue to stream and the tendency of data changes. Therefore, when a structural change occurs, that is, $\lambda_i \leqq C_2$, we need to rebuild a new prediction model.

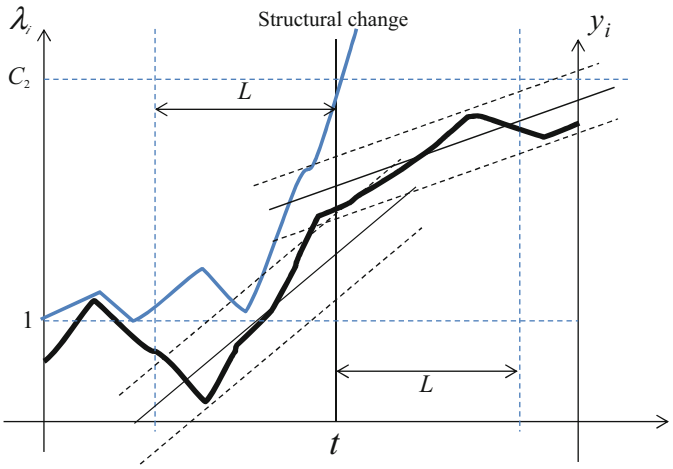


Fig. 3 Relearning Method 1 and 2

We propose five relearning methods for relearning a new prediction model after the detection of a structural change point. Note that the relearning process is conducted under the assumption that the learning period is L and the time of the structural change point is t.

- *[Method 1]* Relearn immediately after detecting a structural change point (Fig. 3)
  We relearn a prediction model from the data appearing soon after detecting a structural change point. In this case, if the structural change occurs at time t, SPRT is rebuilt after $L+t$. Note that this method cannot detect structural change points between t and $L+t$.
- *[Method 2]* Go back to L and relearn (Fig. 3)
  In this method, we go back to L and relearn a new prediction model. This method can detect the model from $t+1$.
- *[Method 3]* Return to the minimum loop count and relearn (Fig. 4)
  In this case, the relearning period starts at the time of going back to the minimum loop count from Steps 5 to 7 to exceed C_2 . In other words, the minimum integer n satisfying $C_2 < (\theta_1/\theta_0)^n$ is backed from t. By using this method, a new prediction model can reflect the past data and future data even though the method needs the $(L-n)$ future data and cannot detect structure changes between t and $t+L-n$.
- *[Method 4]* Date back to the past time when λ_i changes from 1 and relearn (Fig. 5)
  This method involves relearning from the past time when λ_i begins to change from 1. In other words, relearn at the first time when λ_i changes $P(Z_i|H_1)/P(Z_i|H_1)$ from 1 (this time is called t_s for convenience). This method intends to consider the data regarded as structural change starts. Note that the method needs the future data from t in the case of $t - t_s < L$.
- *[Method 5]* Date back to the past time up to L (Fig. 6)
  This method uses the data for relearning based on the fourth method. Note that in this method, we limit the time to go back up to L. This method may be also regarded as the combination between the second and fourth methods.

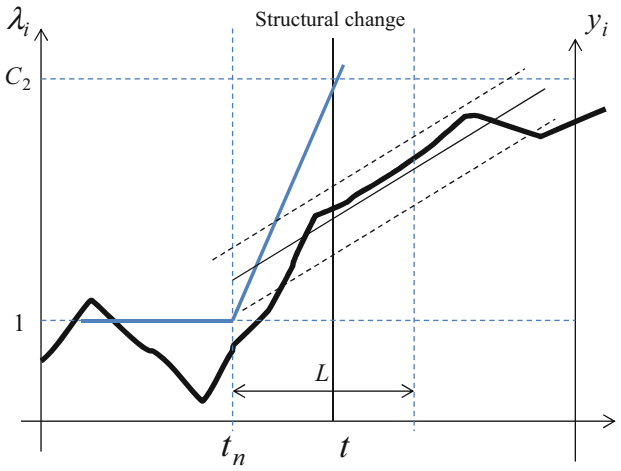


Fig. 4 Relearning Method 3

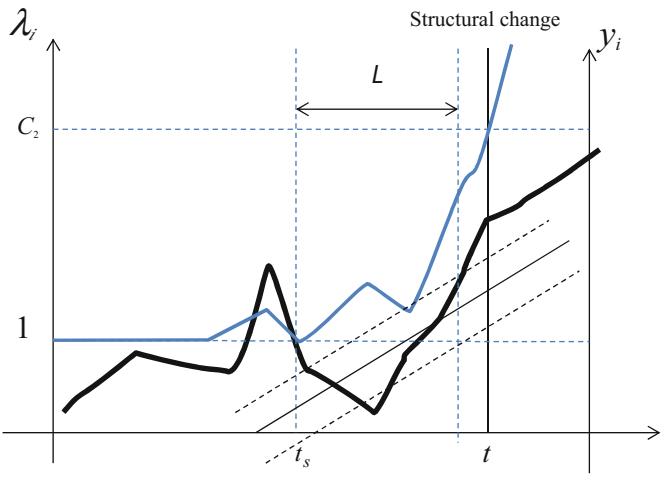


Fig. 5 Relearning Method 4

4 Experiment

We conducted two experiments to verify which of the proposed relearning methods are useful. In the first experiment, we use artificially generated data sets, carries out sensitivity analysis and identifies the value of parameters, and in the second experiment, we use real data.

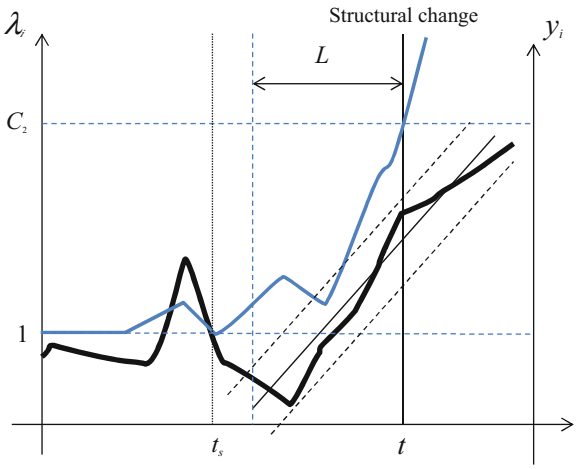


Fig. 6 Relearning Method 5

4.1 Experiments with Generated Data Sets

In this experiment, we generated 20 data sets in accordance with a simple linear regression. Each data set has 1000 observations, which are generated on the basis of the following simple linear regression:

$$y_i = ax_i + b + \epsilon_i, \tag{2}$$

here a is the tendency and ϵ_i the error following $N(0, \sigma^2)$ for point i. Note that we changed the a and b every 200 observations, that is, we artificially designated points 200, 400, 600, and 800 as structural change (for convenient, the period between neighbouring points is called a segment). In addition, we generated 10 data sets of two types of $\sigma^2(30, 150)$ on the assumption that two patterns fluctuate calmly or wildly, i.e. 20 data sets in total.

We utilized simple linear regression model as the prediction model in SPRT. We can expect that multi-regression model and other methods with SPRT works well from the references [12, 17] when we can gain the good evaluation in simple linear regression model. Therefore, in this experiment, we carry out simple linear regression model. In SPRT, we set the parameters as $\alpha = \beta = 0.001$, 0.005, 0.01, and 0.05, $\theta_0, \theta_1 = (0.1, 0.9)$ and (0.2,0.8), and $L = 20, 40, 60, 80$, and 100. Additionally, we used OLS-CUSUM [5, 8, 19] as a reference method because it is a standard model implemented to R (package: strucchange) [20].

To evaluate the experiment, we first introduce the Precision and Recall defined as

$$\text{Precision} = \frac{|D \cap T|}{|D|}, \tag{3}$$

$$\text{Recall} = \frac{|D \cap T|}{|T|}, \tag{4}$$

where D is a set of detected structural change points(i.e. D is equal to the sum of true positive and false negative) and T is a set of correct structural change points (i.e. T is equal to true positive and in this experiment the value is 4 (200, 400, 600, and 800)). In addition, when a method can detect a structural change point after a correct structural change occurs, we count a true positive. On the other hand, when a model cannot, the model misses the structural change point so that we count a false negative. When a model detects structural change points except the point of true positive before structural change occurs, we regard the points as needless points and count a false positive. That is, Precision can be regarded as the probability that the detected structural change points are correct, and Recall can be regarded as the probability that the existing structural changed points are detected. For total estimation, we use the harmonic mean of Precision and Recall as F-measure defined as

$$F\text{-measure} = \frac{2\,\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

In addition, as another criterion, we used the average lag between the correct change point and detected point. That is, the lag is calculated by the difference between the correct change point and the first detected point after the correct change point. If a particular method cannot find structural change points by next structure change, the lag is taken as the period between neighbouring points (i.e. in this experiment the lag is taken as 200 in that segment). If there is no detected change points, we take the F-measure as 0.

For example, when a model detects points in 150, 210, 220, 400, 420, and 630, true positive is 3 (210, 400, and 630), false positive is 3 (150, 220, and 420) and false negative is 1 because there are no points in segment from 800 on. Therefore, Precision $= 3/6 = 0.5$ and Recall $= 3/4 = 0.75$, and as a result, F-measure $= (0.75 \times 0.5 \times 2)/(0.75 + 0.5) = 0.6$. The average lag $= ((210 - 200) + (400 - 400) + (630 - 600) + 200)/4 = 60$. As another example, when a model detects the points in 150, 400, 640, and 800, Precision $= 3/4 = 0.75$, Recall $= 3/4 = 0.75$ and average lag $= (200 + 0 + 40 + 0)/4 = 60$.

4.2 Result of Generated Data Set

Figure 7 shows the experimental results for the generated data sets. Each cell shows the result of the F-measure shown as line graph and the average lag shown as bar graph for each of the relearning methods in each L with the $\alpha(\beta)$ and $\theta_0(\theta_1)$ corresponding to the rows and columns, respectively. And in the line graphs of F-measure, diamond shows Method 1, cross shows Method 2, triangle shows Method 3, square shows Method 4, star shows Method 5 and circle shows OLS-CUSUM. In the bar graphs of average lag, the bars shows Methods 1–5 and OLS-CUSUM from left on each of L.

From the figure, we can see that the F-measure improves and the average lag becomes large as $\alpha(\beta)$ becomes small, especially L was smaller. This reason is

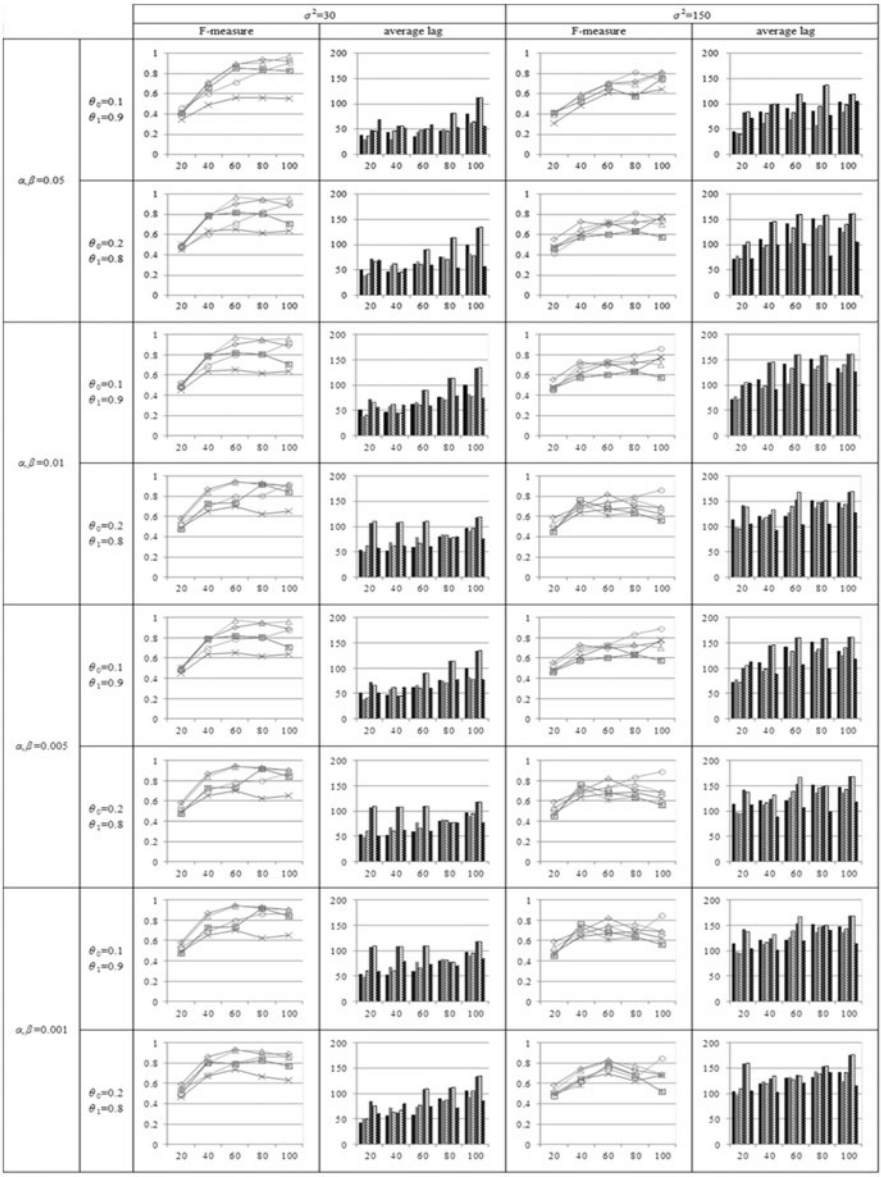


Fig. 7 Experiment result for generated data

simply that the parameter influences the error(i.e. Type I(II) error). As the value becomes small, each method tends to respond sensitively to data fluctuations, so that the F-measure tends to be in error but the structural change points tend to be detected more quickly.

Additionally, θ_0 and θ_1 perform a similar function for α. These parameters are also derived from the error, and they influence the response speed because λ of SPRT is calculated by θ_0 and θ_1. And the change of the parameters strongly influences the average lag rather than F-measure from the figure. Regarding σ^2,the difference of average lag and F-measure between the methods is large for $\sigma^2 = 30$ but becomes small for $sigma^2 = 150$, and the F-measure and average lag of small σ^2 are superior to those of large σ^2.

When comparing the proposed methods, the F-measure of each method increases in many cases up to L = 60. After that, the F-measure changes differently from method to method. Method 1 and 3 are often the best with regard to the F-measure and method 2 is the worst. The reason is that the change points themselves do not adjoin, so that the strategy of the go back method does not work very well. Methods 4 and 5 yield larger values than the others for the average lag.

4.3 Experiments with Real-World Data Set

In the second experiment, we used the data set of the Tokyo Stock Price Index called TOPIX provided by Yahoo! Finance Japan [1], from the period between January 1991 and February 2011 (Fig. 8). We observed the 1104 data points in weekly periods. For this data set, we considered the time events that force changes in an economy, such as the collapse of the bubble economy and Lehman's fall, as structural change points. The list of structural change points is shown in Table 1. Note that it is actually difficult to set structural change but the enumerated changes are not only Japanese but also world's important events and have effects on the Japanese stock. From the consideration, we set them as structural changes although other structural change points my exist in this data.

We utilized a simple linear regression model and set the parameters as $\alpha = \beta = 0.005$, $(\theta_0, \theta_1) = (0.1, 0.9)$ and $L = 60$ as in the first experiment. This choice is based on the following arguments: (1) For $L = 60$, many methods have good F-measures and average lags and the amount of data set is similar to this data set. (2) The criteria are good on $\alpha = 0.005$ and $\theta = 0.1$ independently of σ. As criteria, we used Recall, Precision, F-measure and the average lag.

4.4 Result of Real-World Data Set

The experimental results are shown in Table 2. For comparison, the result of OLS-CUSUM is also included.

From Table 2 it can be seen that most methods have a high Recall and that the best Recall method is Method 2, which has only 1.00. We deduce the reason that this method uses only past data so that it passes over structural change points. On the other hand, Methods 3–5 have high precision. These three methods also have a high

Fig. 8 TOPIX between 1991.1 and 2012.2

Table 1 Configuration of structural change points in TOPIX

Time	Content
1991.10	Collapse of bubble economy in Japan
1995.1	Great Hanshin earthquake
1999.1	Beginning of use of the euro in daily life
2003.5	Jump in oil price
2003.7	Jump in oil price
2007.7	Collapse of the US subprime mortgage market
2008.9	Lehman's fall

Table 2 The result of the experiment for TOPIX

Methods	Precision	Recall	*F*-measure	Average lag
Method 1	0.667	0.857	0.750	37.429
Method 2	0.389	1.000	0.560	19.286
Method 3	0.750	0.857	0.800	20.286
Method 4	0.750	0.857	0.800	21.286
Method 5	0.500	0.857	0.800	20.429
OLS-CUSUM	0.500	0.857	0.632	26.714

Recall, so that their *F*-measures are also better than those of other methods. Method 2 is the quickest in detecting structural change. After that, Method 3, Method 5, Method 4, OLS-CUSUM, and Method 1 are the quickest, in that order. Therefore, we deduce that the best methods are Methods 3 and 5, followed by Method 4.

4.5 Summary of Experiments

Summarizing, we can see the followings:

- The parameters of α, θ_0 etc. affect on F-measures and average lags.
- Each method tends to improve the criteria and hits a peak at some point when the learning period becomes large. In the generated data set experiment, $L = 60$ is remarkable one of the peaks.
- In the generated data experiment, the lags of the proposed methods are larger than those of OLS-CUSUM. However, the proposed methods are superior to OLS-CUSUM. Therefore, it is necessary to examine more data sets under several different conditions.
- The results obtained with Methods 3 and 5 were the best. Both methods use past and future data at the structural change points. Therefore, hybrid-type learning using past and future data may be most useful.

5 Conclusion

In this study, we proposed five relearning methods for a new prediction model after the detection of structural change points. These methods illustrate how a new prediction model uses past and/or future data in relearning. Two experiments were conducted to validate the features of each method. One experiment used 20 generated data sets and the other a real-world data set of TOPIX between 1991 and 2012. The following limitations were identified:

- The generated data in the first experiment is based on normal distribution.
- Only a simple linear regression model is employed in SPRT.
- Only one real-world data set (TOPIX) is used.

From the above limitations, We have to apply and validate the relearning method to other prediction model such as multi-regression model and autoregressive model as well as another test method on other generated and real-world data sets.

References

1. *: Yahoo! finance japan. http://finance.yahoo.co.jp/ (2012). Accessed 18 Aug 2014
2. Blostein, S.: Quickest detection of a time-varying change in distribution. IEEE Trans. Inf. Theory **37**(4), 1116–1122 (1991)
3. Box, G., Jenkins, G.: Time Series Analysis. Forecasting and Control. Prentice Hall, Englewood Cliffs (1976)
4. Brockwell, P., Davis, R.: Introduction to Time Series and Forecasting. Springer, New York (2003)
5. Brown, R.L., Durbin, J., Evans, J.M.: Techniques for testing the constancy of regression relationships over time. J. R. Stat. Soc. Ser. B (Methodol.) **37**(2), 149–192 (1975)

6. Bruzzone, L., Fernandez Prieto, D.: A Bayesian approach to automatic change detection. Proceedings of the 1999 IEEE International Geoscience and Remote Sensing Symposium pp. 1816–1818, Hamburg, Germany (1999)
7. Chow, G.: Tests of equality between sets of coefficients in two linear regressions. Econometrica. **28**(3), 591–605 (1960)
8. Chu, C.S., Hornik, K., Kaun, C.M.: Mosum tests for parameter constancy. Biometrika **82**, 603–617 (1995)
9. Han, C., Willet, P., Abraham, D.: Some methods to evaluate the performance of page's test as used to detect transient signals. IEEE Trans. Signal Process. **47**(8), 2112–2127 (1999)
10. Horvath, L.: The maximum likelihood method for testing changes in the parameters of normal observations. Ann. Stat. **21**, 671–680 (1993)
11. Hulten, G.: Mining time-changing data streams. Proceeding of the 2001 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97–106, San Francisco, USA (2001)
12. Kawano, H., Hattori, T., Nishimatsu, K.: Structural change point detection method of time series using sequential probability ratio test. IEEJ Trans. Electron. Inf. Syst. **128**, 583–592 (2008)
13. Kivinen, J., Smola, A.J., Williamson, R.C.: Online learning with kernels. IEEE Trans. Signal Process **52**, 2165–2176 (2004)
14. Nyblom, J.: Testing for the constancy of parameters over time. J. Am. Stat. Assoc. **84**, 223–230 (1989)
15. Saad, D.: On-Line Learning in Neural Networks. Cambridge University Press, Cambridge (2009)
16. Shah, R., Krishnaswamy, S., Gaber, M.: Resource-aware very fast *k*-means for ubiquitous data stream mining. Proceedings of the 2nd International Workshop on Knowledge Discovery in Data Streams, Porto, Portugal (2005)
17. Takeda, K., Hattori, T., Izumi, T., Kawano, H.: Extended sprt for structural change detection of time series based on a multiple regression model. Artif. Life Robot. **15**, 417–420 (2010)
18. Wald, A.: Sequential Analysis. Wiley, New York (1947)
19. Zeileis, A.: A unified approach to structural change tests based on ml scores, f statistics, and ols residuals. Econom. Rev. **24**, 445–466 (2005)
20. Zeileis, A., Leisch, F., Hornik, K., Kleiber, C.: Strucchange. An R package for testing for structural change in linear regression models. http://epub.wu.ac.at/1124/. Accessed 18 Aug 2014

Part II
Business and Management Tasks

K-means Clustering on a Classifier-Induced Representation Space: Application to Customer Contact Personalization

Vincent Lemaire, Fabrice Clérot and Nicolas Creff

Abstract When the marketing service has to contact customers to propose them a product, the probability that these customers will buy this product is calculated beforehand. This probability is calculated using a predictive model. The marketing service contacts the clients having the highest probability of buying the product. In parallel and before the commercial contact it may be interesting to realize a typology of the customers who will be contacted. The idea is to propose differentiated campaigns by group of customers. This article shows how it is possible to build such a typology so that it respects the nearness of the customers with respect to their appetency score.

1 Introduction

1.1 Industrial Problem

Data mining consists in methods and techniques which allow the extraction of information and knowledge from data. Its use makes it possible to establish correlations between data and, for example within the framework of customer relationship management, to define types of customer's behavior.

One common task is to find the relationships or correlations between a set of input or explanatory variables and one target variable. This knowledge extraction is often based on the building of a model which represents these relationships. Faced with a classification problem, a probabilist model (B) estimates the probabilities of occurrence of each target class for all instances of the database given the values of the explanatory variables. These probabilities, or scores, are used for example in customer relationship management to evaluate the probability that a customer will buy a new product (appetency).

The scores are then exploited by marketing services to personalize the customer relationship. Customers are sorted out according to the value of their score, and only

V. Lemaire (✉) · F. Clérot · N. Creff
Orange Labs, 2 avenue P. Marzin, 22300 Lannion, France
e-mail: vincent.lemaire@orange.com

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_8

the most appetent customers (named "top scores"), i.e. those having the strongest probability to buy the product, are contacted.

In parallel or before the commercial contact, it can be interesting to construct a typology of the customers who will be contacted. This typology is often constructed using a clustering method (G). The idea is to propose marketing campaigns differentiated by customer segments. A sales leaflet is built for every group of customers after analysis of the characteristics of the group: age, CSP, detained offers. For practical reasons (time constraints) the analysis of the group generally amounts to the analysis of the center (or representative customer) of the group. It is important note that this clustering is supposed to have a long lifetime, comparable to the marketing strategy time-scales, and that the same clustering will be re-used for successive marketing campaigns.

Marketing services will then use, for each "top score customer", two pieces of information: the score given by the probabilist model (B) and the characteristics of this customer given by a partitioning method (G). But since there is no link between B and G two problems are generally observed (on Orange campaigns):

1. there is no link, no proximity, between the scores of customers belonging to the same cluster: a cluster can contain customers with a high appetency and customers with a low appetency. The analysis of the center of the group returns an erroneous sales leaflet (as seen above, building a new clustering on the "top scores" after every scoring step is not a viable option).
2. the created clusters are not stable in time when the classifier is deployed successively during several months on the same campaign perimeter (see both criteria Sect. 4.3).

So to resolve the aforementioned problems this article proposes to construct a typology by means of a partitioning method taking into account the knowledge stemming from the classifier which calculates the scores. The purpose is to elaborate a clustering method which preserves the nearness of customers having the same scores.

The second section of this article describes the process which led to choose the algorithm of the k-means as the clustering algorithm. Provided with the choice of the algorithm, Sect. 3 details how to use a classifier-dependent metric, which depends on the classifier used to calculate the scores, during the clusters calculation. Section 4 will present the results obtained before concluding with the last section.

2 Choice of a Technique Among the Various Methods of Clustering Based on Partitioning

Clustering is the process of partitioning a database in groups called clusters. The purpose of clustering is to find groups of similar elements in the sense of a similarity measure. There are thus two main elements to be chosen: the method of groups creation and the metric used during the groups creation.

Notations which will be used below in this article are:

- a training database, $\mathcal{D}$, containing N instances, M explanatory variables and one target variable which has J modalities (the classes to be predicted are noted C_j);
- every data instance, D, is a vector of numerical or categorical values $D = (D_1, D_2, \ldots, D_M)$;
- k is used to designate the desired number of groups.

2.1 Introduction

There are four principal partitioning method which can be used to cluster the elements of a database: a gravity center (the empirical average): the k-means [17]; a geometrical median: the k-medians [3]; a center containing the most frequent modes: the k-modes [12]; a medoid (medoids are representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal): the k-medoids [15].

The choice of one of these algorithms depends on: (i) the nature of the data to which it must be applied; (ii) the desired result (mean, medoid . . .); (iii) the available time and therefore the complexity of the algorithm.

In addition, each of these algorithms depends on the initial selected "center", the value of k, the criterion used to evaluate the quality of the partitioning (cohesion of obtained clusters), the similarity measure and the data representation used at the input of the algorithm.

These points are discussed below in the industrial context of the study.

2.2 Influence of the Nature of the Initial Data

In this study we are in a specific industrial context. Data are from the Orange information system. The explanatory variables which are placed at the input of the classifier (B) used to calculate the appetency probabilities are numerical or categorical variables (with a large number of modalities) and there are missing values. The reader can find a description of these data in [8]. This kind of data representation orients the choice of the partitioning technique towards the technique of the k-prototypes [11] which is a mix of the k-means and k-modes methods. However, the data may also contain a certain number of atypical customers (or erroneous data) which in this case would lead to the choice of k-medoid method that is inherently less sensitive to outliers.

2.3 Influence of the Desired Result

The result of partitioning should allow marketers to build a sales pitch by cluster. A sales pitch is a structured set of arguments that has the characteristics of a product/service as benefits to the customer. It requires detailed knowledge of the product (characteristics), but also of the needs and motivations of the customer. Therefore one would like the "center" of the clusters to represent a "real" customer and not an average customer. It is difficult to extract knowledge from, for example, the average of two genders, several terminal and tariff plans. This desideratum tipped the choice of the partitioning method in favor of the k-medoid method.

2.4 Influence of the Metric

A number of factors must be taken into account when choosing the metric. On the one hand the form of clusters obtained depends on the metric used. On the other hand each of the algorithms described above is dedicated to minimize a particular metric: k-means the L2 norm, k-median the L1 norm [13] ... Although clustering algorithms based on partitioning work with almost any type of distance function (or similarity measure) the same guarantees are not obtained considering the metric used. For example, the Huygens theorem which shows that the sum of intraclusters inertia and interclusters inertia is constant is valid only if one uses the Euclidean distance. In our case we want to adapt the metric to the one which is naturally induced by the classifier (B) used to calculate the appetency probabilities. This adaptation is described in Sect. 3 below. At this point of the article and for understanding the rest of this section, we just indicate that a weighted L1 norm will be used.

2.5 Influence of the Algorithmic Complexity

The algorithmic complexities of the different partitioning methods vary greatly depending on the partitioning method itself but also on the implementation. Readers can find in [10] different implementations of k-median, in [15] different implementations of the k-medoids (PAM, Partitioning Around Medoids; CLARA, Clustering Large Applications; and CLARANS, Clustering Large Applications based upon RANdomized Search). From lowest to highest complexity the algorithms are the k-means, k-mode, k-medoids and finally the k-median.

The marketing campaign involved in this study use databases containing hundreds of thousands of customers, each potentially described by several (tens of) thousands of explanatory variables. After training the classifier (B, which performs a step of variable selection) and retaining only customers with the highest probabilities, databases of tens of thousands of customers described by several hundred variables are obtained. These are databases that are used to build the partitioning. Therefore some of the classical algorithms mentioned above are difficult to use because of the volumetry.

2.6 Influence of the Pretreatment

The classifier used by Orange (in the framework of this study) to calculate the appetency probabilities is Khiops™ (within the PAC platform [6]). Khiops[1] incorporates a Naive Bayes classifier [16] after an optimal pretreatment step on the explanatory variables. Khiops discretizes numeric variables and construct modalities groupings for categorical variables. At the end of the pretreatment process numeric and categorical variables are recoded: each attribute m is recoded in a qualitative attribute values containing I_m recodings. Each instance of data is then recoded as a vector of discrete modalities: $D = D_{1i_1}, D_{2i_2}, \ldots, D_{Mi_M}$. D_{mi_m} represents the recoding value of D_m on the m attribute, with the discrete mode index i_m. After application of the Naive Bayes classifier, the initial explanatory variables are all represented in numerical form as a vector of $M * J$ components: $P(D_{mi_m}|C_j)$.

This pretreatment eliminates the choice of an algorithm like the k-modes, since all variables after the pretreatment step are numeric. It also reduces the advantage of the k-medians/k-medoid regarding the "outliers" because after this type of pretreatment not outliers in terms of a single variable value are present in the data (outliers in terms of variable combinations can still exist).

2.7 Influence of Missing Values

In our case the pretreatment step using Khiops eliminates the missing values. Before the discretization and the grouping of modalities, the missing values for numerical attributes are replace by the values $-\infty$ and those for the categorical attributes are considered as a supplementary value. Then Khiops discretizes numeric variables and construct modalities groupings for categorical variables. Then the K-means algorithm described below is applied on data without missing values.

2.8 Discussion

The above discussion shows the constraints which affect the choice of a the partitioning algorithm most adapted to our industrial context. For example, the computational complexity and nature of the preprocessing performed makes the k-means algorithm very suited to our problem but makes the algorithm less suitable because of the use of a L1 norm and the desire to have real customers as cluster centers.

The k-median algorithm seems more appropriate to the metric used and the nature of the data after preprocessing but its computational complexity makes it unsuitable for our data.

[1] www.khiops.com

The k-medoid algorithm also seems very appropriate but its complexity remains too high (several hours of computing for small databases data even with optimized algorithms such as CLARANS). Other algorithms [19] slightly modify the algorithm of k-medoid to make it closer to the k-means in terms of complexity but need to store the matrix of distances between customers.

Finally, the approach taken in this study is to use the k-median algorithm by taking an approximation of the median as a prototype under the assumption of independent variables and adding a final step after convergence. The assumption of independent variables allows the use of the "component-wise median" [14], a fast version of the median calculation. The step performed after the convergence of the algorithm consists in replacing each prototype by the "real" customer (from this cluster) that is closest to the prototype. The proximity between the customer and the true prototype of the cluster is calculated using a distance L1 norm. This step may slightly degrade the results of the partitioning but it can reach all the objectives given in Sect. 1.1 above.

3 *K*-means Based on Classifier-Induced Representation Space

3.1 Introduction

This section shows that it is possible to insert knowledge coming from the classifier (B) in the metric to be used for the elaboration of a k-means. In our case (the Khiops software) the classifier is obtained from the Averaging of Selective Naive Bayes Classifiers. The purpose is to build a new representation called "supervised representation" (or "classifier-induced representation") so that two instances close in this supervised representation according to the L1 metric should have similar scores (similar appetency probabilities).

The following section describes this supervised representation space for the naive Bayes classifier. Section 3.3 presents how weights are associated with the explanatory variables and how these weights modify the distance.

3.2 Distance Depending on the Target Class

From the naive Bayes predictor and using the log formulation, one has for each target class:

$$log\,(p(C_j|D)) = \sum_{m=1}^{M} log\,\left(p(D_{mi_m}|C_j)\right) + log\,(p(C_j)) - log\,(p(D)) \qquad (1)$$

with $D = (D_m)_{m=1,\ldots,M}$ an instance.

The Bayesian decision corresponds to the target class C_j maximizing the above formula. We define the distance between two instances, d^1_{NB} as follows:

$$d^1_{\mathrm{NB}}(D, D') = \sum_{m=1}^{M} \sum_{j=1}^{J} \left| log\ \left(p(D_{mi_m}|C_j)\right) - log\ \left((p(D'_{mi_m}|C_j)\right)\right|. \tag{2}$$

Each instance can then be encoded in a new representation space as a vector of $M * J$ components, as shown in Eq. 3 for $J = 2$:

$$\begin{aligned}(&log\,(p(D_{i1_1}|C_1)), log\,(p(D_{i1_1}|C_2)), \ldots,\\ &\ldots, log\,(p(D_{Mi_M}|C_1)), log\,(p(D_{Mi_M}|C_2))).\end{aligned} \tag{3}$$

The proposed distance is the L1 norm for this classifier-induced representation. Two instances close in the sense of this representation will be close in the sense of their behavior for the class to predict. Indeed if we define the distance between the predicted class distributions as follows:

$$\Delta^1(D, D') = \sum_{j=1}^{J} \left| log\,(p(C_j|D)) - log\,(p(C_j|D'))\right| \tag{4}$$

and use the following majorization:

$$\Delta^1(D, D') \leq \left[d^1_{\mathrm{NB}}(D, D') + J\,\left|log\,(p(D)) - log\,(p(D'))\right|\right], \tag{5}$$

two instances of the same overall probability close in the sense of d^1_{NB} will be close in the sense of predicting the target class probabilities (two instances with close recoding in the supervised representation will have similar probabilities to have been generated by the recoding model).

3.3 Distance Weighting

The building phase of the weights of the variables used by the Naive Bayes classifier is fully described in [2]. It includes two key steps: a step of variable selection (Sect. 3.5 of [2]) and an averaging step (Sect. 6.2 of [2]). The variable selection step allows the classifier to avoid unnecessary variables or explanatory variables unrelated to the classification problem. The averaging step allows weighting the variables so that the Eq. 1 becomes:

$$\begin{aligned}log\,(p(C_j|D) = &\sum_{m=1}^{M} W_m log\ \left((p(D_{mi_m}|C_j)\right)\\ &+ log\,(p(C_j)) - log\,(p(D)),\end{aligned} \tag{6}$$

where W_m is the weight of the variable m whatever is the target class.

Every instance is then recoded on a vector with $M * J$ components but where each component is weighted. The distance (Eq. 2) is then weighted according to the variables weights and the majorization presented in Eq. 5 remains true.

3.4 Discussion: Modified k-means Algorithm

From here the representation coming from the passage of the initial training database towards a representation where every instance is represented on a vector of $M * J$ components (as shown in the Eq. 3) is called "supervised representation"; where each variable is weighted with its weight W_m.

The result presented above (Eq. 5) provides the guarantee that if the k-means algorithm is used on the supervised representation with the L1 norm, we obtain clusters where two individuals close in the sense of the distance, d^1_{NB}, will be close in the sense of their probability to belong to the target class.

The modified k-means algorithm proposed in this article is called "modified" because it uses (i) a supervised representation of the data, (ii) the L1 norm, (iii) an approximation of the median, (iv) a step of post-processing to select real customers as centers. These four changes are expected to achieve the original objectives of the study as presented in the introduction to this article.

This algorithm assumes that the training data and test data have not different distributions. If this assumption is not relevant the reader may be interested by the following references: [5, 21, 22].

4 Experimental Results

4.1 Preamble

Initialization: Most of the initialization methods mentioned in [18] have been tested. In our case (supervised representation and L1 norm) no significant difference has been found between the results. Results presented below are obtained using a random initialization of the prototypes.

Cross Validation: In each of the experimental phases (and for all values of k) databases were split into ten bags to achieve a cross-validation. The results present in tables and figures are the mean test AUC (Area Under ROC Curve). The score of membership to the target class of an example is defined as the proportion of elements of the target class of the cluster of this example. In case where the number of target classes is greater than 2 the test AUC expectancy is given.

Table 1 Phase 1—AUC: Mean test results (the suffix '-s' indicates the use of the supervised representation)

	Iris	Phoneme	Shuttle	Letter
PAM	0.959	0.926	–	–
PAM-s	0.951	0.935	–	–
K-means	0.946	0.910	0.902	0.711
K-means-s	0.966	0.919	0.929	0.787
J	2	5	7	26
N	150	2554	58000	20000
M	4	256	9	16

NA not applicable

4.2 First Experimental Phase

A first experimental phase was conducted in order to (i) measure the impact of supervised representation on the k-means algorithm and (ii) measure the difference between the results obtained from the k-medoid algorithm PAM (that works directly on the "true" customers) and the step of post-designation included in the modified k-means algorithm.

Khiops software was tested using (i) native data and (ii) data preprocessed to obtain their supervised representation. The tested values of k are in the range of 1 to $\sqrt{N}$ for instance $k \in \mathcal{A} = \{1, 2, \ldots, 9, 10, 20, 40, 80, \ldots, \sqrt{N}\}$. To compare the results with those obtained using PAM the volumetry was limited by using "small" databases from the UCI [1]. The sum of the squares errors (SSE) has not been used to evaluate the results because it is inappropriate here as two different representations (native and supervised) have been tested. The test AUC was then chosen because it gives an indication of purity of the clusters in the sense of a target class.

Table 1 compares the results obtained with (i) the supervised representation to the results obtained with (ii) the native representation for (a) PAM and (b) the modified algorithm on the databases Iris and Phoneme.

For the databases Letters and Shuttle PAM did not provide a result in an acceptable time (for the different tested values of k and the tenfold cross-validation) therefore only results for k-means are presented. This table present a mean test results calculated using individual values obtained as a function of k and the tenfold (f) cross validation $\left(\text{AUC} = \frac{1}{|\mathcal{A}|10} \sum_{k \in A} \sum_{f=1}^{10} \text{AUC}(k, f)\right)$. This mean result corresponds to the area under the Learning Curve which has been recently used as test measure in challenges [9]. Only several representative results [increasing in size of (J, N, M)] of the tests are presented in this article but the interested reader can find more details in [4]. This table shows that the use of a supervised representation exhibits good behavior and gives interesting results.

Figures 1 and 2 show the obtained results on the dataset Abalone ($N = 4177, J = 28$) and Titactoe ($N = 958, J = 2$) using only the supervised representation. In

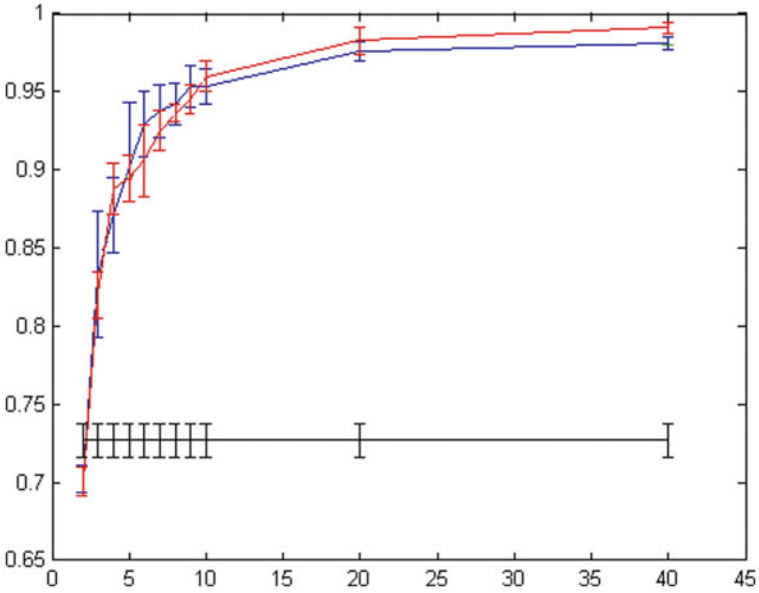


Fig. 1 Abalone: test AUC versus k

these two figures the red (+), blue (•) and black (■) curves represent respectively the classification results obtained for PAM clustering and the modified k-means clustering proposed above (both acting on the supervised representation) and the Averaging of Selective Naive Bayes Classifiers (SNB) for the Khiops software (acting on the native representation).

These illustrative results and those presented in [4], show that the modified k-means algorithm using supervised representation induced by the naive Bayes classifier Khiops is very competitive. We also observe that, for high values of k, it can even achieve a better performance classification than the SNB.

4.3 Second Experimental Phase

Several databases have been available to us for this phase. Three bases of 200,000 customers from March, May, and August 2009 for a churn problem for an Orange product were used. These databases contained around 1000 variables. The database of March was used to construct the classifier (B). The March top scores were used to construct the partition in k groups using the modified k-means algorithm. The databases of May and August correspond to the test sets. The evaluation criteria were calculated for each month (March, May, and August).

Usually the value of k is fixed using a cross validation process. In that case since we are interested by supervised criterion, the criteria describe in [7] will be appropriate.

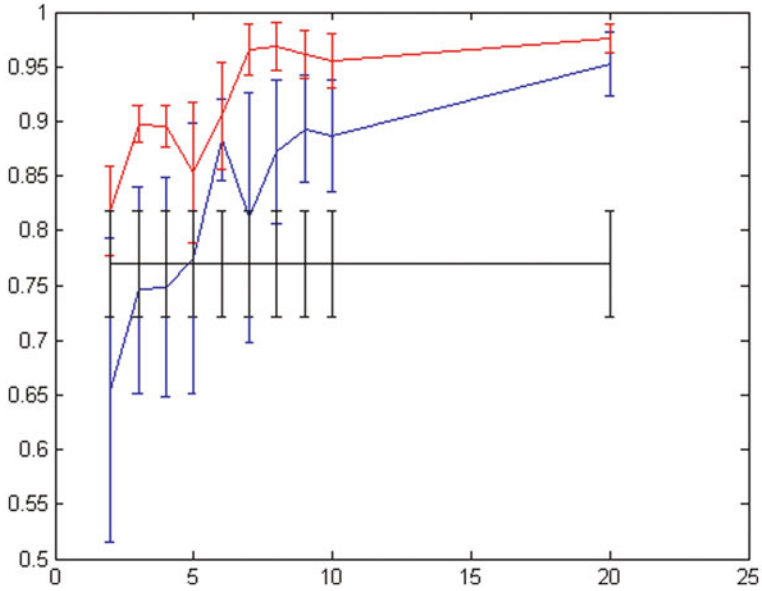


Fig. 2 Titactoe: test AUC versus k

But in our industrial context, users of the clustering algorithm want to set the value of k according to their own requirements and expertise. After consultation with the concerned Orange entity, three k values were tested: 4, 10, and 20. For space reasons only the results with $k = 4$ are presented below; the conclusions remain valid for $k = 10$ or 20 (the complete results are available in [4]).

At the time of the tests, a commercial software solution could be used within the company to achieve this type of campaign. But it was rarely used because the groups obtained were too different from month to month. The modified k-means algorithm proposed in this article was therefore evaluated using a criterion of stability along two dimensions:

- The first dimension is the percentage of customers in each cluster. For each month T, the percentage of customers in each customer is measured. The operation is repeated the following months using new customers (without a new elaboration of the clustering). On a monthly basis the proportions of customers belonging to a cluster should remain the same so that we can consider the solution as stable according to this criterion.
- The second dimension is the evolution of the distribution of the target class within the clusters. For each the month T, the percentage of the target class is measured for each cluster. The operation is repeated the following months using new customers. If the allocation of customers remains the same from month to month then we can consider the clustering method to be stable over time.

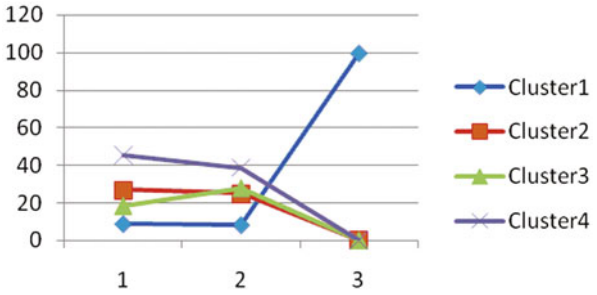


Fig. 3 Percentage of customers per cluster with the current software

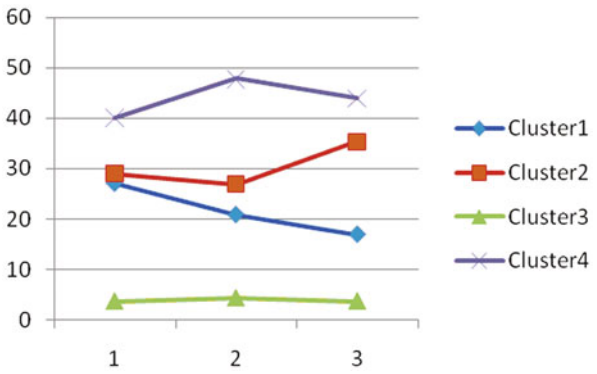


Fig. 4 Percentage of customers per cluster with the proposed algorithm

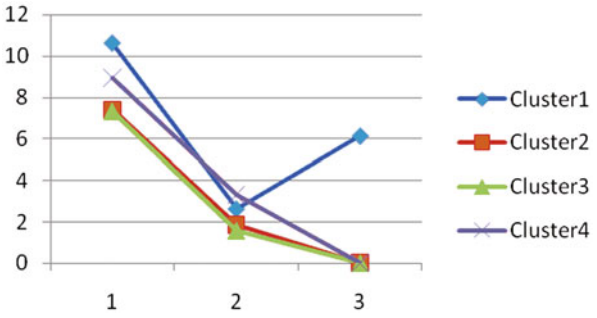


Fig. 5 Percentage of customers (churn = 1) with the current software

The results on these two dimensions are presented Figs. 3, 4, 5, and 6. The x-axis represents the months ($T = 1$ = March, $T = 2$ = May, $T = 3$ = August) and the y-axis percentages. In Figs. 3 and 4, the percentages sum to 100 % and correspond to the top scores. On the other hand, in Figs. 5 and 6, the percentages do not sum to 1 because they represent the proportion of customers in each cluster with the label churn = 1.

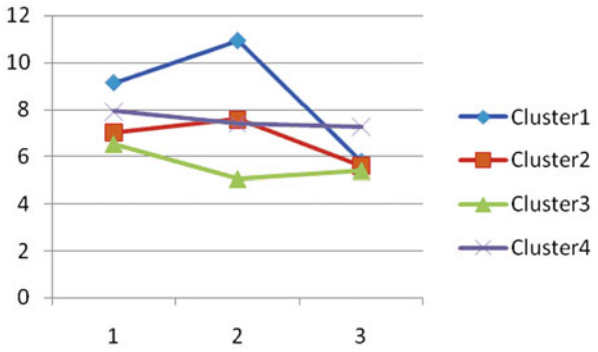


Fig. 6 Percentage of customers (churn = 1) with the proposed algorithm

These four figures show that modified k-means algorithm acting on the supervised representation reaches its goal: the clusters found using supervised representation, which depend on the classifier built in the month T, are much more stable over time (Figs. 4 and 6 as compared to Figs. 3 and 5). We also know that customers in a cluster have similar churn scores.

4.4 Discussion: A Constraint Clustering with Score Proximity

Supervised representation coming from supervised pretreatment (supervised discretization or supervised grouping) allows the use of the result presented Eq. 5 in the case of the classifier is the naive Bayes. This equation provides a guarantee that if we use the k-means algorithm, using the L1 norm and the supervised representation (Eq. 3) we obtain clusters where two individuals close in the sense of the supervised representation will be close in the sense of their probability of belonging to the class target.

However Eq. 5 indicates only $\Delta^1(D, D') \leq d^1_{\text{NB}}(D, D')$. So if two instances D and D' are far in the supervised space we have only the guarantee that the distance between their scores will be smaller. The distance between the scores of two distant instances in the supervised representation can be large.

It would be interesting, in the supervised representation, to force the k-means algorithm to cluster only instances that are further away from a threshold value (denoted by ϵ). An algorithm like Xmeans [20] could be used to cut a cluster where the maximum distance between two instances is greater than ϵ. This constraint would give the guarantee to have no cluster with a diameter greater than ϵ. This guarantee could improve the modified k-means algorithm proposed in this article and automatically set the value of k.

We can also note that the supervised representation built before the clustering could be used with other clustering methods. The Kohonen maps which respect the topology of the space of variables and allow intuitive visualization of the data could be used.

5 Conclusion

This article has shown how to build a typology respecting the knowledge coming from an initial classifier. It was shown that it is possible to elaborate a supervised representation using a naive Bayes classifier. This supervised representation allows a partition that preserves the proximity of samples with the same probability to belong to the target classes. This technique has been used successfully in a customer scoring application. The experimental results show good behavior in terms of measured AUC but also in terms of stability of the typology over time.

The modified k-means algorithm has been operationally deployed and is now used by the Orange business unit which raised the initial problem.

References

1. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html. http://archive.ics.uci.edu/ml/. (1998) Accessed 15 Sept 2010
2. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. J. Mach. Learn. Res. **8**, 1659–1685 (2007)
3. Bradley, P.S., Mangasarian, O.L., Street, W.N.: Clustering via concave minimization. In: Advances in Neural Information Processing Systems -9, pp. 368–374. MIT Press, Cambridge (1997)
4. Creff, N.: Clustering à l'aide d'une représentation supervisée. Master's thesis, Epita, rue Voltaire 94276 Kremlin Bicêtre Cedex, pp. 14–16 (2011)
5. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Self-taught clustering. In: Proceedings of the 25th International Conference on Machine Learning, pp. 200–207 (2008)
6. Féraud, R., Boullé, M., Clérot, F., Fessant, F., Lemaire, V.: The orange customer analysis platform. In: Proceedings of the 10th Industrial Conference on Data Mining, pp. 584–594. Springer Verlag, Berlin, Germany (2010)
7. Ferrandiz, S., Boullé, M.: Bayesian instance selection for the nearest neighbor rule. Mach. Learn. **81**(3), 229–256 (2010)
8. Guyon, I., Lemaire, V., Boullé, M., Dror, G., Vogel, D.: Analysis of the KDD cup 2009: Fast scoring on a large orange customer database. JMLR: Workshop and Conference Proceedings **7**, 1–22 (2009). Data available on http://www.kddcup-orange.com
9. Guyon, I., Cawley, G., Dror, G., Lemaire, V.: Results of the active learning challenge. JMLR W&CP, Workshop on Active Learning and Experimental Design, collocated with AISTATS, Sardinia, Italy, vol. **10**, 1–26 (2010)
10. Har-peled, S., Mazumdar, S.: Coresets for k-means and k-median clustering and their applications. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, pp. 291–300. Chicago, Illinois, USA (2003)
11. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Pacific Asia Knowledge Discovery and Data Mining Conference, pp. 21–34. World Scientific, Singapore (1997)
12. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. **2**, 283–304 (1998)
13. Jajuga, K.: A clustering method based on the l1-norm. Comput. Stat. Data Anal. **5**(4), 357–371 (1987)
14. Kashima, H., Hu, J., Ray, B., Singh, M.: K-means clustering of proportional data using l1 distance. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4 (2008)

15. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
16. Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. In: Proceedings of the 10th National Conference on Artificial Intelligence, pp. 223–228. MIT Press, San Jose, California, USA (1992)
17. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. **1**, pp. 81–297 (1967)
18. Meila, M., Heckerman, D.: An experimental comparison of several clustering and initialization methods. In: Machine Learning, pp. 386–395 (1998)
19. Park, H.S., Jun, C.H.: A simple and fast algorithm for *k*-medoids clustering. Expert Syst. Appl. **36**(2), 3336–3341 (2009)
20. Pelleg, D., Moore, A.W.: X-means: Extending *k*-means with efficient estimation of the number of clusters. In: Proceedings of the 17th International Conference on Machine Learning, ICML '00, pp. 727–734. Morgan Kaufmann, San Francisco, California, USA (2000)
21. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. MIT Press (2009)
22. Zliobaite, I.: Learning under concept drift: An overview. CoRR abs/1010.4784 (2010)

Dimensionality Reduction Using Graph Weighted Subspace Learning for Bankruptcy Prediction

Bernardete Ribeiro and Ning Chen

Abstract Bankruptcy prediction is an extremely actual and important topic in the world. In this complex problem, dimensionality reduction becomes important easing both tasks of visualization and classification. Despite the different motivations, these algorithms can be cast in a graph embedding framework. In this paper we address weighted graph subspace learning methods for dimensionality reduction of bankruptcy data. The rationale behind re-embedding the data in a lower dimensional space that would be better filled is twofold: to get the most compact representation (visualization) and to make subsequent processing of data more easy (classification). To achieve this, two graph weighted subspace learning models are investigated, namely graph regularized non-negative matrix factorization (GNMF) and spatially smooth subspace learning (SSSL). Through an affinity weight graph matrix, the geometric properties are embedded explicitly into the submanifold lying in the high-dimensional data, consequently, the resulting subspace models allow compact representations able to enhance visualization, clustering and classification. The experimental results on a real world database of French companies show that the graph weighted subspace learning models used in a supervised learning manner are very effective for bankruptcy prediction.

1 Introduction

Bankruptcy prediction is an extremely actual and important topic in the world, where the recently developed data mining techniques have been successfully applied for more powerful predictive model and better understanding of the financial data. In

B. Ribeiro (✉)
CISUC, Department of Informatics Engineering, University of Coimbra,
Coimbra, Portugal
e-mail: bribeiro@dei.uc.pt

N. Chen
GECAD, Instituto Superior de Engenharia do Porto,
Porto, Portugal
e-mail: ningchen74@gmail.com

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_9

many real world problems, reducing the dimensionality of data has benefits for visualizing the intrinsic structure of data and it is also an important pre-processing step prior to the pattern recognition stage [54]. In recent years, there has been a raising interest in discovering the manifold of the data lying in a high-dimensional space. Classical techniques such as principal component analysis (PCA) work well when the manifold is embedded linearly in the data space. Among the geometrically nonlinear approaches are manifold learning algorithms, such as ISOMAP [48], local linear embedding (LLE) [45], Laplacian Eigenmaps (LE) [6] which discover the nonlinear structure of data. These algorithms use the locally invariant concept [22] where the close points are likely to have similar embeddings. In the setting of a financial bankruptcy problem, we presented in [43] an enhanced supervised approach to the ISOMAP algorithm (ES-ISOMAP) where the prior knowledge of a variable (indicating bankruptcy risk) is incorporated into a dissimilarity matrix in which the differences between data samples are heightened while irrelevant dimensions are disregarded. Once the low-dimensional manifold is estimated, the embedded mapping is learned using a generalized regression neural network. A classifier is then used in this reduced space for testing new points. Both the algorithm computational complexity and the simplicity of matrix factorization methods motivated an approach based on non-negative matrix factorization (NMF) [31, 42] where the discriminative features incorporating semantics are extracted for subsequent failure prediction. By linking both ideas in this article we seek to encode the intrinsic geometric information in the matrix factorization by using graph regularized non-negative matrix factorization (GNMF) [11]. Furthermore, new research on projection methods which have gained noticeable interest, led to a new subspace learning algorithm that explicitly introduces a Laplacian penalized functional in the objective function. The spatially smooth subspace learning (SSSL) which takes into account the spatial relationships between data points has been shown to excel in face recognition [10].

In this article, we extend our previous work [39, 41, 43] in the field of financial analysis by embedding explicitly the geometric properties into the submanifold lying in the high-dimensional data, and by using the resulting subspace learning models, which allow compact representations able to enhance visualization, clustering and classification.

The rest of the article is organized as follows. Section 2 describes background knowledge on bankruptcy prediction providing related work in the area. In Sect. 3, we describe the subspace models, namely, GNMF and SSSL, aimed at capturing semantic and geometric information from data. In Sect. 4, the data from a case study of the French Market including historic healthy (and bankrupt) firms is described. Results of clustering, visualization and classification are demonstrated in this section. We also provide discussion including parameter selection and comparison with baseline methods. Finally, in Sect. 5 we present the conclusions and point out further lines of work.

2 Background on Bankruptcy Prediction

Credit risk is an extremely actual and important topic in the world which is observing one of the most severe financial crises ever observed. While in the past the small medium (and micro) companies had higher propensity of bankruptcies, in the recent past the number of large bankruptcies are systematically announced. At the heart of the present global recession there is an inappropriate evaluation of credit risk and most of governments were forced to implement rescue plans for the banking systems. Given the devastating effects of the financial distress of firms, now, more than ever, there is a need to identify (and anticipate) this kind of issues. Boosting the accuracy of credit risk methodologies used by banks and financial institutions may lead to considerable gains and have a critical impact on economics.

The real world bankruptcy prediction problem is stated as follows: given a set of parameters (mainly of financial nature) that describe the situation of a company over a given period, predict the probability that the company may become bankrupt during the following year. A large variety of methods have been proposed to provide appealing solution to the task [7]. Many recent studies suggest the use of data mining techniques, including neural networks (NNs), fuzzy set theory (FS), decision trees (DT), case-based reasoning (CBR), support vector machines (SVM), and soft computing [30]. Various prediction models have been proposed using a wide range of intelligent methods. Neural Networks have actively been used in bankruptcy prediction yielding reasonably accurate models able to help on bank lending decisions and profitability [5, 14]. Their properties make them often used in financial applications due to their excellent capability to treat non-linear data [13, 25, 51]. A multi-layer perceptron (MLP) obtains 80 % predictive accuracy on Taiwan and United States markets [26]. Likewise, it predicts the bankruptcy of Iranian companies with desirable outcome [37]. In [16], a stable credit rating model based on learning vector quantization (LVQ) is successfully applied in corporate failure prediction and credit risk analysis. A review of the topic of bankruptcy prediction with emphasis on NN models is given in [4]. SVM is introduced into the bankruptcy prediction problem since the earliest work [33]. It has been proven to yield sound predictive performance with a relatively small amount of data [57].

The recent efforts have shown that the performance of predictive models can be significantly enhanced by means of hybrid and ensemble computing [53]. A hybrid system exploits several approaches (e.g., heuristic techniques and classification algorithms) aiming for optimizing the prediction performance. Following this direction, evolutionary algorithms such as genetic algorithm (GA), annealing simulation (AS), particle swarm optimization (PSO), ant colony optimization (ACO), tabu search (TS) are extensively employed in conjunction with machine learning methods [32]. The typical usages include tuning the architecture of a particular model (such as the connected weights of MLP [27], the parameters of SVM [35]), selecting the relevant features [34], and refining the samples for learning [1]. From another viewpoint, ensemble approach combines several models and aggregates the output in some rules. It has been shown that a well designed ensemble-based system can outperform a single

predictor, inheriting advantages and avoid disadvantages of employed methods [29]. Many attempts have been pursued in aggregating classifiers of different architectures to build an ensemble for bankruptcy prediction, e.g., the ensemble which combines MLP, RBF, PNN, SVM, CART, fuzzy rule-based classifier, PCA-MLP, PCARBF and PCA-PNN [38], the ensemble of LDA, LR, MLP, SVM, and CBR [47], and the one that aggregates DT, MLP and SVM [28]. In [53], a comprehensive review of hybrid and ensemble-based soft computing techniques applied to bankruptcy prediction is presented.

In bankruptcy detection the probability of a corporate failure is of major importance to all stakeholders, since it can help to prevent the adverse effects that such event can provoke. In [40], a probabilistic Bayesian framework for bankruptcy detection based on a relevance vector machine (RVM) is described. It is shown therein that the classifier can yield a decision function that leads to a significant reduction in the computational complexity while the prediction accuracy is competitive. In [44], an SVM+ learning paradigm [52] which finds a classifier with a low generalization error in the decision space is discussed. The model incorporates additional information by grouping the heterogeneous financial ratios according to the firms size and annual turnover, yielding good results in corporate distress prediction. By taking an holistic view of the overall process variables, the learning inductive process is enhanced and the prediction performance improved.

Although these models have actively been investigated in the last decades, still the pioneer statistical techniques are worth mentioning such as multivariate discriminant analysis (MDA), risk index models, and conditional probability model [2, 3]. These techniques aim at finding an optimal linear combination of explanatory input variables, such as, solvency and liquidity ratios, in order to model, analyze and predict corporate default (bankrupt) risk. Unfortunately the financial ratios violate the assumptions of linear separability, multivariate normality and independence of the predictive variables. Therefore, the models overlook the complex nature, boundaries and interrelationships of the financial ratios. Nonetheless, statistical method are still widely used even with some advanced tools, in particular as the baseline for performance comparison. For example, CBR achieves a higher accuracy than MDA on Korean bond-rating data [46]. LVQ is superior than MDA in failure prediction of Turkey banks [8].

As a closely related issue, dimensionality reduction is a crucial component of financial analysis and receives a lot of interest in recent studies. Many dimensionality reduction methods have been proposed, such as t-test, correlation matrix, factor analysis, principal component analysis (PCA), independent component analysis (ICA), and discriminant analysis (DA). In a linear pre-processing stage, PCA and ICA are capable to improve the discriminating power of classifiers [15, 36]. However, nonlinear projection methods are demonstrated particularly applicable to solve the high-dimensional financial data. Among these, manifold learning methods receive much attention. Manifold is an abstract mathematical space in which every point has a neighborhood which resembles the spaces described by Euclidean geometry. Instead of working with points in a high-dimensional space, classification and prediction algorithms can be easily applied in the low-dimensional spaces sought

from the embedded learning process. Regardless of different nonlinear projections (e.g., ISOMAP, local linear embedding (LLE), Laplacian Eigenmaps, and diffusion map), they share the common rationale that the high-dimensional data is cast into low-dimensional manifolds with few degrees of freedom and embedded intrinsic geometry. Ribeiro et al. [41, 43] incorporate the class information in an Enhanced Supervised ISOMAP (ES-ISOMAP) algorithm to uncover the embedding geometry structure of finance data. With the same goal, non-negative matrix factorization (NMF), a multivariate analysis technique for part-based data representation under the non-negative constraint, is used in [42] to extract the most discriminative features, and subsequently construct a classification model for failure prediction. There have been some attempts to encompass various dimensionality reduction algorithms within a unified framework from different perspectives, such as the kernel interpretation [23] and the embedding graph [56]. The former interprets KPCA, ISOMAP, LLE, and Laplacian Eigenmap by a common formulation using different kernel definitions, whereas the latter intends to explain theoretically most of popular dimensionality reduction algorithms as a specific intrinsic and penalty graph varying in the type of embedding.

In bankruptcy prediction problem, few variables are usually insufficient due to the complexity of financial statements. Although dimensionality reduction is not carefully concerned in the literature of bankruptcy prediction, even not considered in some previous studies [21, 50], a recent empirical study shows that the selection of representative variables certainly increases the performance of prediction [49]. Reducing the number of variables was found to be one of the key components in the successful prediction of bankruptcy, not only simplifying the model structure but also improving the discriminative power [12]. Inspired by the results of compact representations that are able to discover the intrinsic discriminant or geometrical structure of the low-dimensional submanifolds lying in data and motivated by the recognition performance in computer vision [10, 55] we extend our earlier work [39] for a better analysis of the financial data. An expanded literature on bankruptcy prediction and manifold learning is introduced. To underscore the latter point, an approach for Enhanced Supervised ISOMAP is schematically illustrated as well as the algorithmic description. The clustering and visualization of the five methods used in the article (PCA, ES-ISOMAP, NMF, GNMF and SSSL) are examined and compared in terms of some evaluation criteria. Moreover,the statistical significance test on different models is provided.

3 Subspace Learning Models

Subspace methods have become rather appropriate to be used in many problems where high-dimensional representations of data are approximately low-dimensional. Among the linear subspace methods single value decomposition (SVD), principal component analysis (PCA), Fisher linear discriminant (FLD), locality preserving projection (LPP) [24] are mostly used.

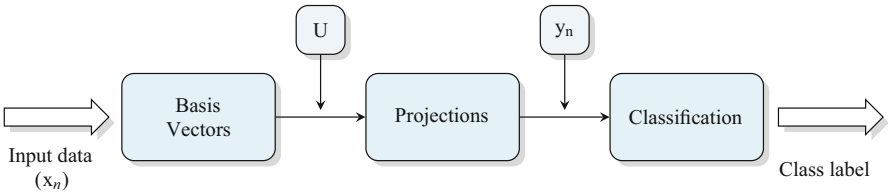


Fig. 1 Subspace techniques as pre-processing stage prior to classification

All the decomposition and projective space techniques (see Fig. 1) share the common goal of describing the data with reduced dimensionality by extracting meaningful components while retaining the geometric structure of raw data.

In the two approaches described below, namely, GNMF and SSSL, we build an affinity weight graph matrix by incorporating geometric neighborhood information of the bankruptcy data set. In the former technique, the geometric based regularizer is considered directly in the objective function and, in the latter, a Laplacian penalising functional is introduced while a regularization parameter controls the smoothness of the basis vectors approximation. Then the transformation matrix is built which maps the data points to a subspace. Once the compact representations are obtained, we seek a subspace learning model where clustering and classification can be effectively performed.

We briefly review in the next subsection the manifold learning methods in particular the ES-ISOMAP algorithm [41, 43] for the sake of comparison with both approaches, GNMF and SSSL.

3.1 Manifold Learning

Manifold methods include a number of nonlinear approaches to data analysis that exploit the geometric properties of the manifold on which the data is supposed to lie. These include algorithms like ISOMAP [48], LLE (Local Linear Embedding) [45], Laplacian Eigenmaps [6] and their variants. They form a neighborhood-preserving projection that projects a point $\mathbf{x}$ from the high-dimensional observation coordinates onto the point $\mathbf{y}$ in the internal coordinates on the manifold. In [43], we looked at the ISOMAP visualization capability in the setting of a bankruptcy financial data under the assumption that it has support on (or is near to) a submanifold. In this regard, a d-dimensional submanifold of a Euclidean space $\mathbf{R}^m$ is a subset of $\mathcal{M}^d \subset \mathbf{R}^m$ which locally looks like a flat d-dimensional Euclidean space [48].

We proposed an enhanced supervised approach (see Fig. 2) to the ISOMAP algorithm (ES-ISOMAP) by incorporating the prior knowledge of a variable (indicating bankruptcy risk) into the dissimilarity weight matrix W. In this manner, the mappings get significantly improved due to the data cluster nature (Healthy and Distressed

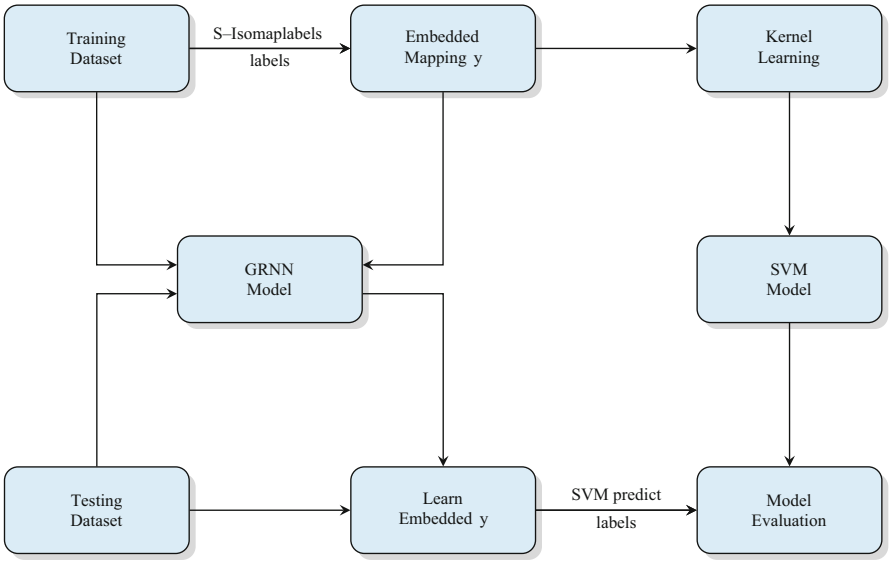


Fig. 2 ES-ISOMAP approach

Enhanced Supervised ISOMAP
input: $X \in \mathrm{IR}^{m \times n}, k, c_i, i = 1,2$
1. Compute Dissimilarity Matrix using Class Labels in the Distance Matrix
2. Run ISOMAP
2.1 Construct Neighborhood Graph G
2.2 Compute Shortest Path Computation Djkstra's (or Floyd's) Algorithm
2.3 Finding the Low Embedding Map Y using MDS
3. Learning the Embedded Mapping
3.1 Generalized Regression Neural Network (or Kernel Regression)
3.2 Project the testing data set to the manifold
4. SVM Testing on New Points Data

Fig. 3 Enhanced supervised ISOMAP algorithm

firms). Moreover, from the assumption that different features of the data can be captured by different dissimilarity measures (Euclidean, Cosine, Correlation, Spearman, Kendal-τ), in our algorithm (summarized in Fig. 3) important differences between data samples are heightened while irrelevant dimensions are disregarded. Following the step 1. (dissimilarity weight matrix construction described above), we next run ISOMAP [48] and multidimensional scaling (MDS) [18], as shown in step 2, for uncovering the manifold embedded in the data. Once the low-dimensional manifold is estimated, the embedded mapping is learned using a generalized regression neural network (or by kernel regression) as indicated in step 3. Finally, in step 4, a classifier in this reduced space is found for testing new points.

3.2 Graph Regularized Non-negative Matrix Factorization (GNMF)

Given a matrix $X \in \mathbf{R}^{m \times n}$, NMF [31] aims to decompose X into two non-negative matrices $U \in \mathbf{R}^{m \times k}$ and $V \in \mathbf{R}^{n \times k}$ so that $X \approx U * V^T$. The squared Euclidean distance (F-norm) is the commonly used objective function, defined as:

$$O = ||X - UV^T||^2. \tag{1}$$

In real applications, usually $k \ll m$ and $k \ll n$, thus each data vector $\mathbf{x}_j (j = 1, \ldots, n)$ is approximated by a linear combination of the columns of U, weighted by the components of V.

In GNMF, the intrinsic geometrical structure of the data is approximated by manifold rather than the Euclidean space. As demonstrated by manifold learning, the local geometric structure can be modeled by a nearest graph to which a weight matrix W is specified by the p-nearest neighbors manner and some weighting schemes.

Afterwards, the smoothness of low dimensional representation is measured as Eq. 2, where $\mathbf{z}_j = \{v_{j1}, \ldots, v_{jk}\}, j = 1, \ldots, n$ denotes the row vector of V, and W_{jl} denotes the weight of the edge between point $\mathbf{x}_j$ and $\mathbf{x}_l$ on the graph.

$$R = \frac{1}{2} \sum_{j,l=1}^{n} ||\mathbf{z}_j - \mathbf{z}_l||^2 W_{jl}. \tag{2}$$

In summary, GNMF is formulated as an optimization problem to minimize the following objective function, where $\lambda \geq 0$ is the regularization parameter. Particularly, GNMF becomes NMF when $\lambda = 0$.

$$\begin{array}{ll} Min_{U,V} & ||X - UV^T||^2 + \frac{\lambda}{2} \sum_{j,l=1}^{n} ||\mathbf{z}_j - \mathbf{z}_l||^2 W_{jl} \\ st. & u_{ij} \geq 0, v_{ij} \geq 0. \end{array} \tag{3}$$

The update rules minimizing the objective function can be derived:

$$u_{ik} = u_{ik} \frac{(XV)_{ik}}{(UV^TV)_{ik}}; \quad v_{jk} = v_{jk} \frac{(X^TU + \lambda WV)_{jk}}{(VU^TU + \lambda DV)_{jk}}. \tag{4}$$

The GNMF optimization is solved by updating U and V alternatively through an iterative process.

1. Construct a p-nearest neighbor graph, taking each column vector of matrix X as a point;
2. Assign the weight matrix W of the graph;
3. Initialize the matrix U and V as small values;
4. Fix U, optimize V to minimize the objective function, then fix V, optimize U;
5. Repeat from 4 until it converges.

3.3 Spatially Smooth Subspace Learning (SSSL)

Given a graph G with n nodes, each node representing a data point, let W be a symmetric $n \times n$ matrix where W_{ij} is the connection weight between node i and j. Each node of the graph is represented as a low-dimensional vector and the similarities between pairs of data (in the original high-dimensional space) are preserved. The corresponding diagonal matrix and the Laplacian matrix [17] are defined as:

$$L = D - W, D_{ii} = \sum_{j \neq i} W_{ij} \quad \forall i, \tag{5}$$

where D is a diagonal matrix whose entries are sums of columns (or rows) of the matrix W. Let the low-dimensional embedding of the nodes be $\mathbf{y} = [y_1 y_2 \cdots y_n]$, where the column y_i vector is the embedding for the vertex $\mathbf{x}_i$. Direct graph embedding [56] aims to maintain similarities among vertex pairs by following the graph preserving criterion (7):

$$\mathbf{y}^* = \arg\min_{\mathbf{y}^T D\mathbf{y}=1} \sum_{i \neq j} ||y_i - y_j||^2 W_{ij} \tag{6}$$

$$= \arg\min_{\mathbf{y}^T D\mathbf{y}=1} (\mathbf{y}^T L\mathbf{y}) = \arg\min \frac{\mathbf{y}^T L\mathbf{y}}{\mathbf{y}^T D\mathbf{y}}. \tag{7}$$

The similarity preservation property of the graph G follows the idea that if the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is high, then the distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ should be small to minimize Eq. (7); on the other hand, if the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is low, the distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ should be large. Hence, the similarities and differences (among vertex pairs) in the graph are preserved in the embedding [55].

The above optimization problem has the equivalent form below given that $L = D - W$:

$$\mathbf{y}^* = \arg\max \mathbf{y}^T W\mathbf{y} = \arg\max \frac{\mathbf{y}^T W\mathbf{y}}{\mathbf{y}^T D\mathbf{y}}. \tag{8}$$

It is clear that the matrices W and D have a major influence in the graph embedding. We follow the notation in [10] to denote the graph embedding as $G(W, D)$ with maximization problem $\max(\mathbf{y}^T W\mathbf{y})/(\mathbf{y}^T D\mathbf{y})$. The graph embedding provides the mappings for the training set. In classification a mapping for all samples, including the test examples, is required. Let $\mathbf{u}$ be the transformation vector and $\mathbf{y}_i = \mathbf{u}^T\mathbf{x}_i$. Equation (8) becomes:

$$\mathbf{u}^* = \arg\max \frac{\mathbf{u}^T XWX^T\mathbf{u}}{\mathbf{u}^T XDX^T\mathbf{u}}. \tag{9}$$

The optimal $\mathbf{u}^*$ are the eigenvectors corresponding to the maximum eigenvalues of the decomposition problem:

$$XWX^T\mathbf{u} = \lambda XDX^T\mathbf{u}. \tag{10}$$

The approach, known as *Linear Graph Embedding*, has nicely been extended to include the spatial smoothness of the basis vectors using the Laplacian penalized functional [10]. The resulting Spatially Smooth Subspace learning (SSSL) uses the graph structure with the weight matrix W and solves the following optimization problem:

$$\mathbf{u}^* = \arg\max \frac{\mathbf{u}^T X L}{(1-\alpha)\mathbf{u}^T X D X^T \mathbf{u} + \alpha \mathcal{L}}, \tag{11}$$

where $\mathcal{L}$ is the discretized Laplacian regularization function and α is the parameter that controls the smoothness of the approximation. The vectors $\mathbf{u}_i (i = 1 \cdots l)$ that maximize the objective function (11) are the solutions of the eigenvalue problem:

$$X W X^T \mathbf{u} = \lambda((1-\alpha) X D^T X + \alpha \Delta^T \Delta)\mathbf{u}, \tag{12}$$

where Δ is a $m \times m$ matrix giving a discrete approximation for the Laplacian $\mathcal{L}$ [10].

3.4 Building the Affinity Graph Matrix

The affinity graph matrix W is built by assuming that each ith node corresponds to a given firm $\mathbf{x}_i$. In the p-nearest neighbor graph we then put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are nearby points, i.e., if $\mathbf{x}_i$ is among the p-nearest neighbors of $\mathbf{x}_j$ and $\mathbf{x}_j$ is among the p-nearest neighbors of $\mathbf{x}_i$. We also considered the supervised mode where the class information is available. Notice that, in this case, we put an edge between the data points belonging to the same class.

Once the affinity graph is constructed, the weight matrix W can be specified by means of weighting schemes such as binary, heat kernel and dot-product as defined in [11]. The weight matrix models the local structure of the data set manifold.

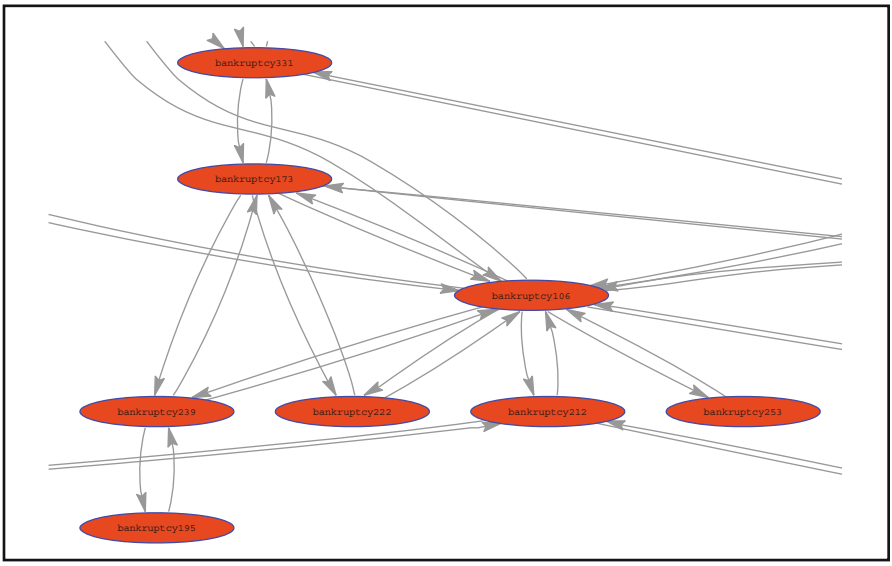
1. Binary weighting. $W_{ij} = 1$ if and only if nodes i and j are connected by an edge, otherwise $W_{ij} = 0$.
2. Heat kernel weighting (with σ the kernel width). The scheme for assigning weights between nodes i and j is:

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same class;} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

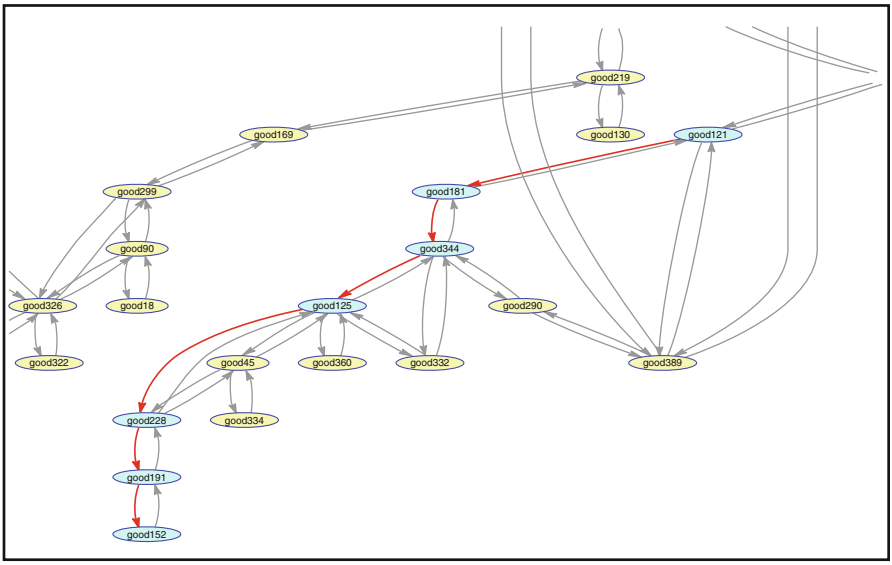
3. Dot-product weighting.

$$W_{ij} = \begin{cases} \frac{\mathbf{x}_j^T \mathbf{x}_i}{||\mathbf{x}_i|| \cdot ||\mathbf{x}_j||}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same class;} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Figure 4 illustrates (parts of) the graphs corresponding to the bankrupt (and healthy) companies constructed with the heat kernel ($\sigma = 0.5$) and (p-neighbors =5) in the supervised mode, i.e., with class label information.



(a)



(b)

Fig. 4 *W* matrix graph with supervised mode and heat kernel. Screen shot for distressed (bad) and healthy (good) companies

4 Experimental Results

This section describes the data set, indicates the preprocessing procedure, and then presents the results and discussion. The metrics for performance evaluation as well as the statistical significance tests for validation of results are also introduced.

4.1 Data Description

We used Diane database which contains financial statements of French companies. One of the problem goals is to find a model able to predict the class (healthy, bankrupt) in a correct manner. Therefore, bankruptcy prediction is handled as a binary class problem. The initial sample contained about 60,000 financial statements from industrial French companies (during the years of 2002–2006) with at least ten employees. In these companies, about 3000 were declared bankrupt in 2007 (or presented a restructuring plan to the court for approval by the creditors). Due to the large number of missing values existed in the companies (particularly in bankrupt companies), we select 600 companies with at most ten missing values from the bankrupt group. It was known that the classification tends to favor the majority class (non-default companies) under the highly skewed distribution of the original database. We then sampled randomly 600 non-default companies in order to generate a balanced data set for experiments. After pre-processing the bankruptcy data set contains 1200 French companies, 600 examples distressed in 2007, and the remainder are healthy. The 30 financial ratios produced by Coface[1] are described in Table 1. These financial predictors allow to describe the firms in terms of its financial strength, liquidity, solvability, productivity of labor and capital, margins, net profitability and return on investment. Some of the variables have small discriminatory capabilities for default (bankrupt) prediction with linear statistical models, whereas non-linear approaches extract relevant (and discriminatory) information improving the classification. In the experiments we took the historical data consisting of 90 inputs spanning three years before bankruptcy with 1200 balanced data samples.

4.2 Results and Discussion

In this section, we investigate the performance of subspace learning algorithms, GNMF and SSSL, in terms of clustering, visualization and prediction for bankruptcy analysis. Moreover, we compare the results with PCA, ES-ISOMAP and NMF methods.

[1] Coface is one of largest financial groups in France providing Credit Insurance, the Factoring Information & Ratings and Debt Recovery.

Table 1 Financial ratios of French DIANE database

Variable description			
x_1-	Number of employees previous year	x_{16}-	Cashflow/turnover
x_2-	Capital employed/fixed assets	x_{17}-	Working capital/turnover days
x_3-	Financial debt/capital employed	x_{18}-	Net current assets/turnover days
x_4-	Depreciation of tangible assets	x_{19}-	Working capital needs/turnover
x_5-	Working capital/current assets	x_{20}-	Export
x_6-	Current ratio	x_{21}-	Added value per employee in k EUR
x_7-	Liquidity ratio	x_{22}-	Total assets turnover
x_8-	Stock turnover days	x_{23}-	Operating profit margin
x_9-	Collection period days	x_{24}-	Net profit margin
x_{10}-	Credit period days	x_{25}-	Added value margin
x_{11}-	Turnover per employee k EUR	x_{26}-	Part of employees
x_{12}-	Interest/turnover	x_{27}-	Return on capital employed
x_{13}-	Debt period days	x_{28}-	Return on total assets
x_{14}-	Financial debt/equity	x_{29}-	EBIT margin
x_{15}-	Financial debt/cashflow	x_{30}-	EBITDA margin

4.2.1 Clustering and Visualization

Our previous studies on the same problem (and same data) of French distressed companies show that ES-ISOMAP and NMF are powerful methods for clustering. Herein, they will be used for comparison together with PCA. In Fig. 5, clustering and visualization results with the five methods considered ((a) PCA, (b) ES-ISOMAP, (c) NMF, (d) GNMF, (e) and (f) SSSL) are illustrated for the case $k = 2$ (rank value).

As one can see the ES-ISOMAP and the non-negative matrix factorization based methods, both NMF and GNMF, outperform PCA which suggest the superiority of parts-based methods with (or without) manifold regularization in discovering hidden factors. GNMF Fig. 5d uses a p-nearest graph to capture the local geometric structure of the data distribution. The success of GNMF relies on the assumption that two neighboring data points share the same label. As for the SSSL a powerful discrimination over the bankrupt (and healthy) companies is obtained as can be observed in Figs. 5 e, f. Some further explanation for the strength of SSSL method is given below in terms of the smooth regularization parameter α. In summary, by using the weight graph matrix W which accomodates the geometrical properties of the data, it is clearly shown that GNMF and SSSL outperform the other methods.

We next study the clustering capability of the five methods. In this experiment three evaluation criteria, namely purity, rand index and normalized mutual information, are used to measure the match between the real clustering C and resulting clustering C'.

Purity is the percent of the correctly assigned instances, for which majority voting is the usually used labeling method. In the following definition, $c'(x_i)$ is the assigned

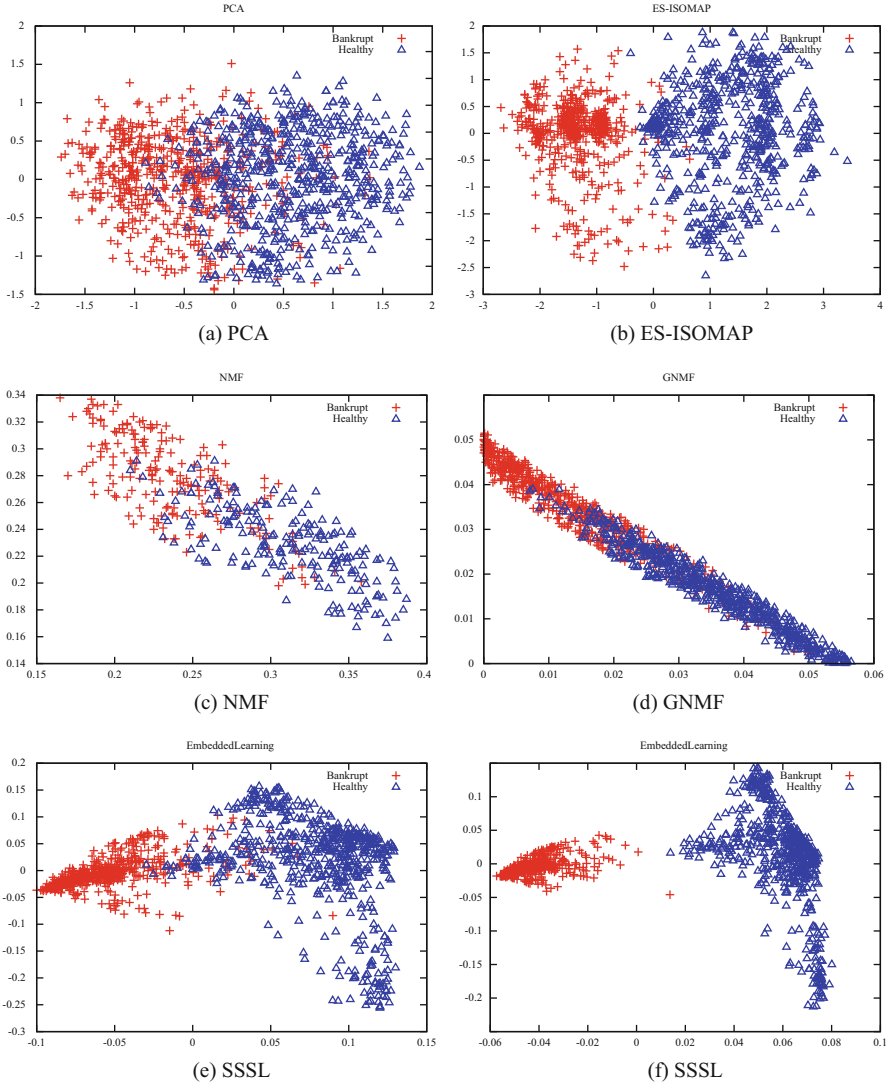


Fig. 5 Visualization with subspace learning for rank $k = 2$. *Red dots* corresponding to bankruptcy; *blue dots* to healthy companies

label of the cluster to which x_i belongs, $c(x_i)$ is the real label of x_i, δ is the indicator function taking the value 1 when the condition satisfies, otherwise 0.

$$P = \frac{\sum_{1 \le i \le n} \delta(c(x_i) = c'(x_i))}{n}. \tag{15}$$

Rand index measures the similarity between two data clustering. Specifically, it calculates the proportion of the instance pairs that belong to the same or different clusters simultaneously in real and resulting clustering respectively. Let a be the number of instance pairs that are in the same cluster in both C and C', b the number of instance pairs that are in different clusters in both C and C', c the number of instance pairs that are in the same cluster in C and different clusters in C', and d the number of instance pairs that are in the same cluster in C' and different clusters in C. Rand index can be defined as:

$$RI = \frac{a+b}{a+b+c+d}. \tag{16}$$

Normalized mutual information is a measure of the correlation between two clustering using entropy. Let $P(C_i)$ be the probability of instances selected from the ith cluster of C, $P(C'_i)$ be the probability selected from the ith cluster of C', $P(C_i \bigcap C'_j)$ be the probability selected from ith cluster of C and jth cluster of C' simultaneously, the mutual information (MI) is defined as:

$$MI = \sum_{C_i \in C} \sum_{C'_j \in C'} P\left(C_i \bigcap C'_j\right) \ log_2 \frac{P(C_i \bigcap C'_j)}{P(C_i)P(C'_j)}. \tag{17}$$

The normalized mutual information (NMI) can be defined in several forms. Here we take the one used in [9], where H_C and $H_{C'}$ are the entropy of real and resulting clustering respectively.

$$NMI = \frac{MI}{max(H_C, H_{C'})}. \tag{18}$$

Table 2 illustrates the clustering results yielded by running the five methods under the setup conditions indicated in column 2. Moreover, 30 test runs conducted in each experiment. The mean and the standard error of the performance are reported in Table 2. It is clear shown that the best methods for clustering considering rank = 2, i.e., two clusters, are the GNMF and SSSL which incorporate the graph structure into the partition found by each method. All the methods are quite stable as indicated by the low standard deviations.

Table 3 describes in more detail the parameters setup for the research design. We followed the graph weight construction for (B) as indicated in [41]. The Djkstra's Algorithm (see step 2.2, Fig. 3) was also used. As for (C)–(F) we used Sect. 3.4 for building the weight matrix [9]. More specifically, to run the experiments we have chosen the Euclidean (or Cosine) metrics. For two data points these distances evaluate the "closeness" between them. The NeighborMode indicates how to construct the graph and the available choices are K-nearest neighbor (KNN) and supervised mode (SUP). In KNN mode, the number of p-nearest neighbors means either a complete graph is constructed ($p = 0$) or an edge between two nodes is put if and only if they are among the $p > 0$ nearest neighbors of each other. In SUP mode, an edge

Table 2 Clustering results for rank = 2

Metrics	Methods setup	P (%)	RI (%)	NMI (%)
PCA	(A)	76.14 ± 1.75	63.64 ± 1.87	20.89 ± 3.01
ES-ISOMAP	(B)	78.13 ± 0.04	65.80 ± 0.05	25.82 ± 0.64
NMF	(C)	84.19 ± 0.25	73.36 ± 0.34	37.29 ± 0.57
GNMF	(D)	84.66 ± 0.41	74.00 ± 0.58	38.47 ± 1.02
SSSL	(E)	90.68 ± 3.64	83.36 ± 5.78	64.44 ± 8.92
SSSL	(F)	*92.34 ± 1.39*	*85.89 ± 2.32*	*68.17 ± 3.79*

Table 3 Clustering methods setup conditions

Methods setup	Metrics	Neighbor mode	Weight scheme	Parameters
(A)	Euclidean	–	–	$k = 2$
(B)	Euclidean	KNN ($p = 5$)	Dot-product	$\alpha = 0.8$
(C)	Euclidean	KNN ($p = 5$)	Heat Kernel ($\sigma = 0.5$)	$\lambda = 0$
(D)	Euclidean	SUP ($p = 2$)	Dot-product	$\lambda = 100$
(F)	Euclidean	KNN ($p = 2$)	RBF kernel ($\sigma = 0.1$)	$\alpha = 0.05$
(E)	Euclidean	KNN ($p = 2$)	Poly kernel ($d = 3$)	$\alpha = 0.05$

between two nodes is added if and only if they belong to the same class ($p = 0$), or if they belong to same class and they are among the ($p > 0$) nearest neighbors of each other. Regarding the weight scheme which indicates how to assign the weights in the graph we followed the procedure indicated in Sect. 3.4. Therefore, to build the graph weight matrix we indicate whether an HeatKernel, dot-product or simply binary mode is used. With respect to the free parameters in each method, for instance in (A) the PCA method, k indicates the number of principal components. In the case of (B) α is the parameter used to calculate the distance metrics for building the dissimilarity matrix in the ES-ISOMAP approach (see [41]). In the case of NMF (C) and GNMF (D) λ is the parameter for embedding the geometry of the data into the weight matrix. Finally, in the SSSL (E) and (F) α is the regularization parameter for better accomodating the projection of data. Therefore, α is an essential parameter in SSSL model which controls the smoothness of the estimator.

4.2.2 Parameter Determination

In order to further investigate the GNMF method on the problem data, we run several experiments for different values of parameter λ and for various rank values k changing from $2 \longrightarrow 81$. The results are illustrated in the histogram bars of Fig. 6. The colors in each bar of the histogram are indicated for increasing values of λ, namely, ($\lambda = 1$, $\lambda = 10$, $\lambda = 100$, $\lambda = 1000$, $\lambda = 10000$). This parameter has a key role on the model selection and it would be interesting to be automatically determined. In

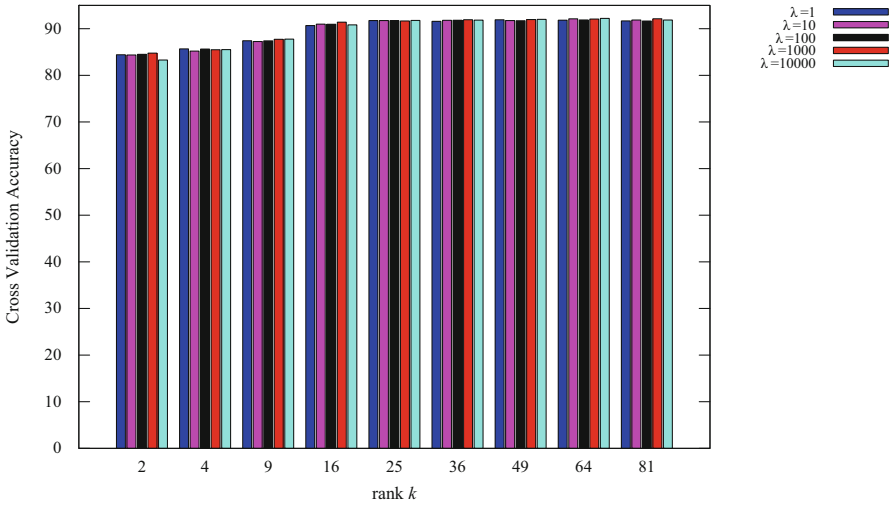


Fig. 6 The performance of GNMF for varying λ with heat kernel weighting scheme, $\sigma = 0.5$, and p-neighbors $= 5$

the experiments it is observed that the best mean results are attained with $\lambda = 1000$ corresponding to the red bar.

With respect to SSSL, we investigate the model performance by varying the regularization constant α (see Eq. 11) from 0.0 to 1. This parameter affects the Laplacian penalty to constrain the problem features to be spatially smooth. In other words, it represents the degree of smoothness of the projection vectors in the approximation. An interesting discussion on how α influences the subspace learning approach is given in [10]. Basically if $\alpha = 0$, the SSSL model will reduce to the ordinary subspace learning approach which totally ignores the spatial relationship between firms across the dataset. When $\alpha \to \infty$, the SSSL will wholly ignore the manifold structure of the handled firms dataset. We used cross-validation to select the best α parameter. In Fig. 7, we illustrate the performance accuracy yielded by using five fold cross-validation and SVM in the classifier stage. As shown the best value is $\alpha = 0.05$. Further details on the classification are given below.

4.2.3 Supervised Subspace Learning

To understand how effective were the subspace learning models, the classification step was performed using an SVM. Since we built a balanced data set we used accuracy as the performance measure. The model selection was performed with fivefold cross validation and we averaged the results by running each model ten times. Table 4 illustrates the mean results (and standard deviations) with five methods (a) PCA, (b) ES-ISOMAP, (c) NMF, (d) GNMF, (e) SSSL. In all the experiments we decided to use RBF kernel since it was shown to be the best in previous empirical

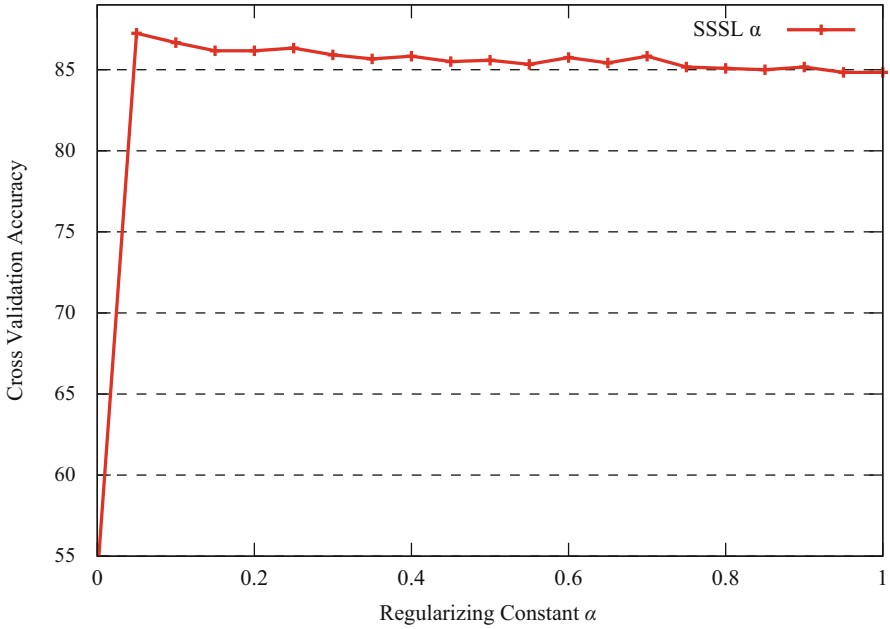


Fig. 7 The performance of SSSL for varying α with heat kernel weighting scheme, $\sigma = 10$, and p-neighbors $= 5$

results running in the same data set [42, 43]. The main parameters for each model are indicated, selected from the previous study. SSSL outperforms the other methods although GNMF also shows good behavior. In average for all the rank values SSSL is better than (i) GNMF by 0.98 %, (ii) PCA by 1.94 %, (iii) NMF by 3.4 % and (iv) ES-ISOMAP by 5.06 %. The choice of Laplacian penalty in SSSL allows to incorporate the prior information that relate neighboring points across the firms data. Once we obtain compact representations of the firms behavior, classification and clustering can be effectively performed in the lower dimensional subspace.

It is interesting to notice that ES-ISOMAP is powerful in the visualization and clustering as observed, respectively, in Fig. 5b and Table 2 (second row). However, in the final classification stage, the performance of ES-ISOMAP (combined with SVM) is lower when compared to the other methods. The reason might be that during manifold learning the projection map is not explicitly found. Therefore, as described in the step 3. of ES-ISOMAP algorithm (see Sect. 3.1 and Fig. 3) the mapping should be learned by regression, training a NN (or by kernel regression), which thus propagates errors to the projected test data. Consequently, during the recall phase, the classification performance is degraded. To better illustrate the performance of SSSL method the results yielded by varying the rank (for each method) are comparatively plotted in Fig. 8.

Table 4 Cross-validation results of classification methods: (a) PCA-SVM, (b) ES-ISOMAP-SVM ($p = 5$), (c) NMF-SVM ($\lambda = 0$), (d) GNMF-SVM ($p = 5$, $\lambda = 1000$), (e) SSSL-SVM ($\alpha = 0.05$)

Rank	PCA-SVM	ES-ISOMAP-SVM	NMF-SVM	GNMF-SVM	SSSL-SVM
k	Acc Std (r_i^j)	Acc Std (r_i^j)	Acc Std (r_i^j)	Acc Std (r_i^j)	Acc Std (r_i^j)
2	84.18 ± 0.93 (4)	85.18 ± 0.88 (2)	83.93 ± 0.84 (5)	84.89 ± 0.23 (3)	*87.60± 0.12* (1)
4	86.08 ± 0.10 (2)	84.80 ± 0.79 (4)	84.32 ± 0.68 (5)	85.65 ± 0.42 (3)	*87.90± 0.12* (1)
9	88.23 ± 0.18 (3)	85.15 ± 0.46 (5)	85.60 ± 1.12 (4)	88.70 ± 0.54 (2)	*90.45± 0.23* (1)
16	89.90 ± 0.57 (3)	86.33 ± 0.95 (5)	86.92 ± 1.39 (4)	91.40 ± 0.54 (2)	*91.92± 0.20* (1)
25	90.75 ± 0.14 (3)	86.10 ± 0.26 (5)	88.59 ± 1.09 (4)	91.66 ± 0.26 (2)	*92.15± 0.23* (1)
36	90.63 ± 0.07 (3)	86.10 ± 0.26 (5)	89.51 ± 0.88 (4)	91.93 ± 0.47 (2)	*92.33± 0.18* (1)
49	90.65 ± 0.14 (3)	86.45 ± 0.51 (5)	89.83 ± 0.91 (4)	91.97 ± 0.29 (2)	*92.28± 0.22* (1)
64	90.75 ± 0.13 (3)	86.75 ± 0.82 (5)	89.99 ± 0.72 (4)	92.07 ± 0.62 (2)	*92.33± 0.12* (1)
81	90.63 ± 0.10 (3)	86.90 ± 0.16 (5)	89.94 ± 0.69 (4)	92.14 ± 0.56 (2)	*92.30± 0.21* (1)
Average	89.09	85.97	87.63	90.05	*91.03*
R_j	3	4.5556	4.2222	2.2222	*1*

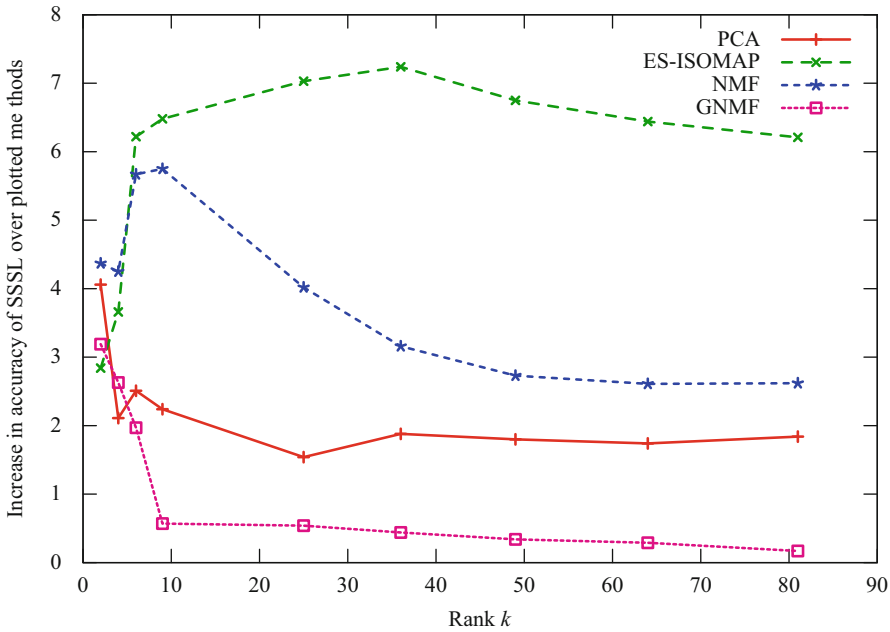


Fig. 8 Classification result comparison of SSSL-SVM with **a** PCA-SVM, **b** ES-ISOMAP-SVM, **c** NMF-SVM, and **d** GNMF-SVM

Friedman test [19] is a non-parametric equivalent of ANOVA for the comparison of multiple classifiers. It is used for repeated measures analysis of variance by ranks to detect the differences among the treatments. Friedman test is particularly suitable for machine learning studies when the assumptions (independency, normality, and homoscedasticity) do not hold or are difficultly verified for a parametric test [20]. For each problem, the classifiers are ranked separately, where the best performance has rank 1, the second has rank 2, etc. If there are ties, the average rank is assigned to the tied values. We denote r_i^j as the rank of jth classifier on the ith problem. The Friedman test then compares the average ranks $R_j = \sum_i r_i^j / n$ of the investigated classifiers. The null hypothesis states that all classifiers behave similarly so that the average ranks should be equal. Therefore, Friedman statistic $\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]$ is distributed according to χ_F^2 with $k-1$ degrees of freedom.

From the average ranks in Table 4, we have $\chi_F^2 = 30.6667$, and the resulting p-value is 3.5801e−6, indicating that the null hypothesis is rejected. It means that there is significant difference among the performance of the five classifiers. Next, we perform a post-hoc Nemenyi test for pairwise comparison. The performance of two classifiers is significantly different at 5 % (or 10 %) significance level if their average ranks differ at least the critical value $\mathrm{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6n}} = 2.0333$ ($\alpha = 0.05$) or 1.8328 ($\alpha = 0.1$) respectively. Table 5 shows the difference of average ranks. We can conclude that (e) outperforms (b) and (c) at the level 5 % with the difference

Table 5 Nemenyi tests: (a) PCA-SVM, (b) ES-ISOMAP-SVM ($p = 5$), (c) NMF-SVM ($\lambda = 0$), (d) GNMF-SVM ($p = 5$, $\lambda = 100$), (e) SSSL-SVM ($\alpha = 0.05$)

	b	c	d	e
a	−1.5556	−1.2222	0.7778	2*
b		0.3333	2.3333**	3.5556**
c			2*	3.2222**
d				1.2222

*significance at 10 % level; **significance at 5 % level

3.5556 and 3.2222 respectively greater than the critical value 2.0333, as well as (a) at the level 10 % with the difference 2 greater than the critical value 1.8328. On the other hand, the performance of (d) is significantly better than (b) (with the difference 2.3333) at the level 5 % and (c) (with the difference 2) at the level 10 %, whereas there is no significant difference between (d) and (e). It verifies the potential of GNMF and SSSL as the projection method for bankruptcy prediction combined with SVM.

5 Conclusion and Future Work

We investigated the subspace learning models with a proper constructed graph weight matrix in the setting of a financial problem. The embedded graph of the bankruptcy data is cast under several parameters for better filled space. The experiments show that the properties of the projected data yield meaningful and appealing visualization and clustering of data. Furthermore the accuracy obtained by cross-validation yields good results as compared with our previous work on the same data. Namely, both the non-negative matrix factorization in embedded graph (GNMF) and spatially smooth Laplacian regularization (SSSL) used in a supervised learning manner demonstrate that these methods are very effective for this problem. In the future work, some limitations will be addressed. First, although the models studied are viable, a further step including not only the class information but also heterogeneous information could foster a closer insight on the firms behavior. In particular, it might allow to detect default drifts along time, possibly avoiding catastrophic losses by stakeholders. Second, an extensive evaluation will be conducted to validate the generalizability of the results using more data sets, other dimensionality reduction methods and state-of-the-art data mining models. Third, the effect of parameters involved in the employed models will be investigated to achieve an optimal performance.

References

1. Aha, H., Kim, K.: Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. Appl. Soft Comput. **9**(2), 599–607 (2009)
2. Altman, E.: Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J. Financ. **23**(4), 589–609 (1968)

3. Altman, E.: Corporate Financial Distress and Bankruptcy: A Complete Guide to Predicting and Avoiding Distress and Profiting from Bankruptcy, 2nd edn. Wiley, New York (1993)
4. Atiya, A.: Bankruptcy prediction for credit risk using neural networks: A survey and new results. IEEE Trans. Neural Netw. **12**(4), 929–935 (2001)
5. Baek, J., Cho, S.: Bankruptcy prediction for credit risk using an auto-associative neural network in Korean firms. In: Proceedings of International Conference on Computational Intelligence for Financial Engineering, Hong Kong, pp. 25–29. IEEE (2003)
6. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems 14, pp. 585–591. MIT Press, Cambridge, MA (2002)
7. Bellovary, J., Giacomino, D., Akers, M.: A review of bankruptcy prediction studies: 1930 to present. J. Financ. Educ. **33**(4), 1–43 (2007)
8. Boyacioglu, M., Kara, Y., Baykan, O.: Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: A comparative analysis in the sample of savings deposit insurance fund (SDIF) transferred banks in Turkey. Expert Syst. Appl. **36**(2, Part 2), 3355–3366 (2009)
9. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Trans. Knowl. Data Eng. **17**(12), 1624–1637 (2005)
10. Cai, D., He, X., Hu, Y., Han, J., Huang, T.: Learning a spatially smooth subspace for face recognition. In: Proceedings of International Conference on Computer Vision and Pattern Recognition Machine Learning (CVPR'07), Rio de Janeiro, Brazil, pp. 1–7. IEEE (2007)
11. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1548–1560 (2011)
12. Cao, Y.: Aggregating multiple classification results using Choquet integral for financial distress early warning. Expert Syst. Appl. **39**(2), 1830–1836 (2012)
13. Chandra, D., Ravi, V., Ravisankar, P.: Support vector machine and wavelet neural network hybrid: Application to bankruptcy prediction in banks. Int. J. Data Min. Model. Manag. **2**(9), 1–21 (2010)
14. Charalambous, C., Charitou, A., Kaourou, F.: Application of feature extractive algorithm to bankruptcy prediction. In: Proceedings of International Joint Conference on Neural Networks, Como, Italy, vol. **5**, pp. 303–308. IEEE (2000)
15. Chen, N., Vieira, A.: Bankruptcy prediction based on independent component analysis. In: Proceedings of International Conference on Agents and Artificial Intelligence (ICAART09), pp. 150–155 (2009)
16. Chen, N., Vieira, A., Ribeiro, B., Duarte, J., Neves, J.: A stable credit rating model based on learning vector quantization. Int. J. Intell. Data Anal. **15**(2), 237–250 (2011)
17. Chung, F.: Spectral Graph Theory, 1st edn. American Mathematical Society, Providence (1997)
18. Cox, T., Cox, M.: Multidimensional Scaling, 1st edn. Chapman & Hall, London (1994)
19. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
20. Garcia, S., Fernandez, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inf. Sci. **180**(10), 2044–2064 (2010)
21. Gestel, T., Baesens, B., Suykens, J., Poel, D., Baestaens, D.E., Willekens, M.: Bayesian kernel based classification for financial distress detection. Eur. J. Oper. Res. **172**(3), 979–1003 (2006)
22. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proceedings of Computer Vision and Pattern Recognition Conference (CVPR'06), pp. 1735–1742. IEEE (2006)
23. Ham, J., Lee, D., Mika, S., Scholkopf, B.: A kernel view of the dimensionality reduction of manifolds. In: Proceedings of International Conference on Machine Learning, Alberta, Canada, pp. 47–54 (2004)

24. He, X., Niyogi, P.: Locality preserving projections. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16, pp. 153–160. MIT Press, Cambridge (2004)
25. Huang, F.: A genetic fuzzy neural network for bankruptcy prediction in chinese corporations. In: Proceedings of International Conference on Risk Management & Engineering Management (ICRMEM08), pp. 542–546 (2008)
26. Huang, Z., Chen, H., Hsu, C., Chen, W., Wu, S.: Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis. Support Syst. **37**(4), 543–558 (2004)
27. Huang, K., Kuo, Y., Yeh, I.: A novel fitness function in genetic algorithm to optimize neural networks for imbalanced data sets. In: Proceedings of 8th International Conference on Intelligent Systems Design and Applications. pp. 647–650 (2008)
28. Hung, C., Chen, J.: A selective ensemble based on expected probabilities for bankruptcy prediction. Expert Syst. Appl. **36**(3), 5297–5303 (2009)
29. Kima, M., Kang, D.: Ensemble with neural networks for bankruptcy prediction. Expert Syst. Appl. **37**(4), 3373–3379 (2010)
30. Kumar, P.R., Ravi, V.: Bankruptcy prediction in banks and firms via statistical and intelligent techniques - a review. Eur. J. Oper. Res. **180**(1), 1–28 (2007)
31. Lee, D., Seung, H.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems 13. pp. 556–562. MIT Press, Cambridge, MA (2001)
32. Lin, S., Shiue, Y., Chen, S., Cheng, H.: Applying enhanced data mining approaches in predicting bank performance: A case of taiwanese commercial banks. Expert Syst. Appl. **36**(9), 11543–11551 (2009)
33. Min, J., Lee, Y.: Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Syst. Appl. **28**(4), 603–614 (2005)
34. Min, S., Lee, J., Han, I.: Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Syst. Appl. **31**(3), 652–660 (2006)
35. Min, J., Jeong, C., Kim, M.: Tuning the architecture of support vector machine - the case of bankruptcy prediction. Int. J. Manag. Sci. **17**(1), 1–116 (2011)
36. Pai, G.R., Annapoorani, R., Pai, G.V.: Performance analysis of a statistical and an evolutionary neural network based classifier for the prediction of industrial bankruptcy. In: Proceedings of International Conference on Cybernetics and Intelligent Systems, vol. **2**, Singapore, pp. 1033–1038. IEEE (2004)
37. Rafiei, M., Manzari, S., Bostanian, S.: Financial health prediction models using artificial neural networks, genetic algorithm and multivariate discriminant analysis: Iranian evidence. Expert Syst. Appl. **38**(8), 10210–10217 (2011)
38. Ravi, V., Kurniawan, H., Thai, P.N.K., Kumar, P.R.: Soft computing system for bank performance prediction. Appl. Soft Comput. **8**(1), 305–315 (2008)
39. Ribeiro, B., Chen, N.: Graph weighted subspace learning models in bankruptcy. In: Proceedings of International Joint Conference on Neural Networks (IJCNN), San Jose, USA, pp. 2055–2061. IEEE (2011)
40. Ribeiro, B., Silva, C., Neves, J.: Sparse Bayesian models: Bankruptcy-predictors of choice? In: Proceedings of International Joint Conference on Neural Networks, Vancouver, Canada, pp. 3377–3381. IEEE (2006)
41. Ribeiro, B., Vieira, A., Carvalho das Neves, J.: Supervised isomap with dissimilarity measures in embedding learning. In: Ruiz-Shulcloper, J., Kropatsch, W. (eds.) Progress in Pattern Recognition, Image Analysis and Applications, Lecture Notes in Computer Science, vol. 5197, pp. 389–396. Springer, Berlin (2008)
42. Ribeiro, B., Silva, C., Vieira, A., Neves, J.: Extracting discriminative features using non-negative matrix factorization in financial distress data. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science, vol. 5495, pp. 537–547. Springer, Berlin (2009)

43. Ribeiro, B., Vieira, A., Duarte, J., Silva, C., das Neves, J., Liu, Q., Sung, A.: Learning manifolds for bankruptcy analysis. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) Advances in Neuro-Information Processing, Lecture Notes in Computer Science, vol. 5506, pp. 723–730. Springer, Berlin (2009)
44. Ribeiro, B., Silva, C., Chen, N., Vieira, A., Neves, J.: Enhanced default risk models with SVM+. Expert Syst. Appl. **39**(11), 10140–10152 (2012)
45. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
46. Shin, K., Han, I.: A case-based approach using inductive indexing for corporate bond rating. Decis. Support Syst. **32**(1), 41–52 (2001)
47. Sun, J., Li, H.: Listed companies financial distress prediction based on weighted majority voting combination of multiple classifiers. Expert Syst. Appl. **35**(3), 818–827 (2008)
48. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
49. Tsai, C.F.: Feature selection in bankruptcy prediction. Knowl. Based Syst. **22**(2), 120–127 (2009)
50. Tsai, C.F., Wu, J.W.: Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Syst. Appl. **34**(4), 2639–2649 (2008)
51. Tseng, F., Hu, Y.: Comparing four bankruptcy prediction models: Logit, quadratic interval logit, neural and fuzzy neural networks. Expert Syst. Appl. **37**(3), 1846–1853 (2010)
52. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural Netw. **22**(5–6), 544–557 (2009)
53. Verikas, A., Kalsyte, Z., Bacauskiene, M., Gelzinis, A.: Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey. Soft Comput. Fus. Found. Methodol. Appl. **14**(9), 995–1010 (2010)
54. Verleysen, M.: Learning high-dimensional data. In: Limitations and Future Trends in Neural Computation, pp. 141–162. IOS, Netherlands (2003)
55. Yan, S., Liu, J., Tang, X., Huang, T.: A parameter-free framework for general supervised subspace learning. IEEE Trans. Inf. Forensics Secur. **2**(1), 69–76 (2007)
56. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. **29**(1), 40–51 (2007)
57. Yang, Z., You, W., Ji, G.: Using partial least squares and support vector machines for bankruptcy prediction. Expert Syst. Appl. **38**(7), 8336–8342 (2011)

Part III
Fraud Detection

Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft

Brendan Kitts, Jing Ying Zhang, Gang Wu, Wesley Brandi, Julien Beasley, Kieran Morrill, John Ettedgui, Sid Siddhartha, Hong Yuan, Feng Gao, Peter Azo and Raj Mahato

Abstract Microsoft adCenter is the third largest Search advertising platform in the United States behind Google and Yahoo, and services about 10 % of US traffic. At this scale of traffic approximately 1 billion events per hour, amounting to 2.3 billion ad dollars annually, need to be scored to determine if it is fraudulent or bot-generated [32, 37, 41]. In order to accomplish this, adCenter has developed arguably one of the largest data mining systems in the world to score traffic quality, and has employed them successfully over 5 years. The current paper describes the unique challenges posed by data mining at massive scale, the design choices and rationale behind the technologies to address the problem, and shows some examples and some quantitative results on the effectiveness of the system in combating click fraud.

1 What is Click Fraud?

Pay Per Click (PPC) auctions are a significant engine for the online advertising economy. They have taken Google from a revenueless start-up company to a giant making $ 37 billion per year [17]. PPC Auctions show many remarkable properties including a tendency towards increased relevance with increased density [19, 20]. Conversion rates can also be extremely high because the keywords are typed by a user looking for the product or service that they are typing [24].

Unfortunately PPC has an Achilles Heel. Click fraud is the term used to describe artificial clicks generated on advertisements to either create direct or indirect financial gain from the PPC payouts [27]. Click Fraud strikes at the heart of PPC's economic model. Advertisers pay for clicks that don't convert, leading them to need to lower bids. Ad networks generate reduced ROI, resulting in fewer advertisers, and innocent publishers receive lower payouts because of revenue being diverted to cheaters [25, 33–36]. Click fraud is an area which requires significant investments in detection technology and a constant arms race with attackers in order to ensure that the

B. Kitts (✉) · J. Y. Zhang · G. Wu · W. Brandi · J. Beasley · K. Morrill · J. Ettedgui · S. Siddhartha · H. Yuan · F. Gao · P. Azo · R. Mahato
Microsoft Corporation, One Microsoft Way, Redmond, WA, USA
e-mail: bkitts@excite.com

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_10

economics of PPC work to provide value for advertisers, users and publishers [13, 15, 16, 31, 45, 46].

2 Examples of Click Fraud Attacks

A wide range of Click Fraud attacks have been documented in the literature [1, 6, 7, 9, 10, 12, 18, 29, 30, 39, 40, 42].

One of the earliest was a human clicking operation that was uncovered and sued by Google in 2004. Auctions Experts Limited had used 12 employees to click on ads [40].

Leyden [29] reported on a 115 computer click botnet that was designed to execute a low frequency click fraud attack. The controller of the botnet used a Graphical User Interface to manage its slave computers. Each slave computer was configured to click no more than 15 times per day and target specific paid listings.

A much larger scale botnet was uncovered by Panda Labs [9]. ClickBotA was a 100,000 computer botnet designed to execute another sophisticated, low frequency click fraud attack. Machines were infected by downloading a popular screensaver. Infected machines randomly queried a lexicon from a MySQL database, and fired these against the search engine. The infected machine then selected a listing from the ad-results and asked its Central BotNet Controller whether it "canClick"? If the Central BotNet Controller counted less than 20 clicks against that ad-link in the day, then it responded that it could click.

At Microsoft we filed a law suit against Eric Lam and Supercontinental LLC for their alleged activities running the WOW Botnet [7]. The WOW Botnet executed a click fraud attack across hundreds of thousands of IPs. However in this case the intent wasn't to directly generate revenue from the clicks, but to actually target advertisers. The objective was to deplete the budget of advertisers, eliminating them from the auction, and allowing the attacker—who was actually an advertiser on the same keywords—to monetize high quality human traffic.

3 Overview of Paper

In this paper we will discuss the unique data mining systems needed for combating this major problem at one of the largest companies in the world [22, 28, 31]. Because click fraud is an area with real financial implications and adversaries trying to attack the system—some of whom may be reading this article—we will not be able to discuss the specific algorithms and signatures being used to successively combat fraud. We will instead focus on the unique technology needed to operate at massive scale, and what we have learned over 5 years in this challenging data mining problem. The lessons that we learned developing this system should be helpful at other very large-scale data mining initiatives, particularly those that are searching for rare events and facing adversarial attackers.

4 Why Click Fraud Detection is Hard

Click Fraud is an adversarial detection problem [14]. Attackers exploit sophisticated methods to cloak their activities including mimicking human behavior and sometimes hijacking legitimate human traffic. They also evolve—after deploying countermeasures, we've watched as different strains start appearing and attempt to break through. The challenges of the click fraud detection problem can be summarized as (a) throughput requirements, (b) rapidity of model updates needed to combat attackers, (c) low frequency nature of attacks (d) user anonymity, (e) programmability of attacks, (f) accuracy requirements, and (g) the need to detect and eliminate the effects of fraud within milliseconds.

At the current scale of traffic serviced by Microsoft, approximately 1 billion events per hour need to be scored to determine if it is fraudulent or bot-generated. To provide a sense for scale, this is 300 times more events than US credit card transactions [43]. Similarly the variables in the online space are massive. There are 4 billion possible Internet Protocol (IP) v4 addresses and, as IP v6 is adopted, there will be 3 x 1038 IP v6s [11]. adCenter currently detects over half a million IPs per hour. Looking for combinations of IP behavior against thousands of publishers and millions of keywords creates major computational challenges.

This enormous scale is in stark contrast to the number of fraudulent events. Often these events are spread across large numbers of IPs. For example ClickBotA was configured to click fewer than 20 times per IP [9] and PandaLabs which was configured to click less than 15 times per IP [29]. Low frequency attacks are designed to blend in with statistical noise to avoid detection.

Ad Networks also need to intercept and contain attacks in real-time. This is necessary to prevent disruption to the economics of the auction including depletion attacks [6, 26, 35] and ad clickthrough rate prediction spoofing [36]. This creates enormous challenges for computational infrastructure and informs the architecture that needs to be fielded.

5 Filtration System Principles

5.1 Lossless Processing

There are many hard lessons learned in the development of our traffic quality systems and we believe that these could help inform how other attacker detection systems might be effectively designed.

In 2006, adCenter's filtration system was set up to filter traffic by physically discarding records as soon as they underwent certain quality tests. There were a series of these "stages" since different information was available at different points in processing. The intuition behind this was that "if the traffic is bad why bother spending CPU cycles to process it?"

However because records were being dropped it meant that the number of records coming out from processing was significantly less than the number of records going in. This led to repeated executive escalations due to the concern that adCenter may be dropping traffic. Each of these escalations required a time-consuming investigation to resolve.

When we re-platformed the system in 2007, we ensured that the new design was non-lossy—every impression, click and conversion would be processed by our systems so that we can see filtered and unfiltered data [31]. A useful analogy is to think of this like a "Conservation Law for Clicks". Clicks would be neither created nor destroyed, however could be "transformed" from one classification to another [23]. There are rare circumstances in which traffic may need to be dropped due to a denial of service attack, however we will discuss later that a lighter-weight component is designed for looking for this and we maintain a minimal rate of sampling in order to continue to evaluate the traffic when drastic action is necessary.

5.2 Rapid Updates

Drops in early stage processes led to another undesirable phenomenon. Because logic was scattered across different systems, it became difficult to update the filtration logic. Simulating the system also required changes to multiple systems, making testing difficult.

adCenter's filtration logic is now isolated in a single, centralized filtration decision point called the Minerva Module. Filtration decisions are then fed to all downstream systems. This facilitates maintenance, troubleshooting, and rapid updates to the filtration logic.

Rigorous test and deployment systems have been created around this one module so that the model can be rapidly updated. Because of the ability to completely test this component, hotfix model updates can be deployed to production when needed. The centralized architecture has dramatically increased our speed in responding to attacks.

5.3 Rules Representation

It is well known that a variety of machine learning algorithms could be used to solve a particular classification problem. However what is not as widely known, is the impact of different algorithms on daily operations. For example, one of adCenter's early systems utilized Naïve Bayes to predict clickthrough rate on keywords. This seemed like a good idea, but led to a global model with hundreds of thousands of weights. When inevitable issues with bad ad selections emerged, there were many possible causes and it was difficult to isolate the problem and fix the bad ads. In

designing the filtration system, we intentionally chose a rules representation. Rules have a number of advantages for large-scale fraud detection:

- *Every filtration decision has a reason.* Every rule can be identified with a ruleID. When the rule fires, the model outcome (e.g. filter/don't filter) as well as the ruleID that fired, can then both be output for reporting.
- *Bad rules can be discretely identified.* It is possible to report on each ruleID that fired along with that rule's accuracy in identifying bot traffic. As a result, every rule can be clearly measured. Rules that are performing poorly can be easily removed without upsetting the rest of the model.
- *Machine-induced and Human Expert rules both supported seamlessly.* Rules can of course be induced using a variety of induction techniques including decision trees. In addition, if an analyst is able to identify some traffic that should be filtered, they can also add those rules into the system, along with a ruleID just like all of the other rules. Their rules can then be measured for accuracy in exactly the same way.
- *Decisions are auditable.* By recording the reason for a traffic classification, it also becomes possible to easily audit the system. In the IAB/MRC audit it is possible to verify that the system is filtering traffic for the reasons that we expect [16, 31, 45].
- *Ease of troubleshooting.* In addition, if a publisher's traffic is being filtered—perhaps inappropriately—the exact rule which fired is recorded. This makes it possible to rapidly debug any issues with the filtration system.
- *Ease of interpretation.* When we do find cases where publishers are pereptarating fraud, being able to see the rules that are detecting them helps greatly in understanding the kind of exploit that they are using.
- *Updatability*. Because the rules are understandable and discrete, it is very easy and fast to update rules. There are no global interactions that need to be considered.
- *Ease of integrating findings from other teams.* One of the really nice features of rules is that they allow researchers, investigators, and others to develop rules or discover methods for detecting attacks which can then be simply plugged in. We solicited rules from a variety of fraud teams including our investigation team, and promised to name the rules they developed after their inventor. Each person who came up with rules could track their rule's revenue impact and fraud detection performance, and they could try to lead in detecting attackers.

5.4 Rule Bitmaps ("Multiple Rules")

As is true of other rules based systems, multiple rules may trigger for a given input. For example, both an IP blacklist and bot detection rule may trigger. When this occurs it is typical in Expert Systems literature for a conflict resolution algorithm to determine the winning rule which should fire [2].

Because of the selection algorithm, some rules may be "shadowed" by other rules which tend to trigger with a higher priority—for example the IP blacklist may have higher priority than the bot detection rule. When this happens it can cause problems for measuring rule quality, since the bot detection rule may "appear" to be performing poorly, however this is because of other rules which are "taking credit" for the traffic. Without the ability to accurately measure the performance of each rule, it is possible to inappropriately remove "poor performing" rules without awareness of the value of the rule or its interactions.

adCenter's approach to this problem is to both create a single "winning" rule for basic reporting purposes, as well as to record every rule which was triggered for detailed analysis with full awareness of rule interactions.

adCenter achieves this by creating a bitmap output for each record. Each bit position on the bitmap corresponds to a rule. For example the following bit representation 01000101 means that rules 1, 3, and 7 all triggered. The unique number for this configuration of rules is 69. An example of a bitmap translated into human-readable format is below, showing that five rules triggered for a particular input is below.

```
;;;;;;;;;;;;;;;;RULEA1;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;RULEC2;;;;;;;;;;
;;;;;;;;;;;RULED1;;;RULEA2;DUP;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

The bit position doubles as the rule priority number, with lower bits having higher priority. In the example above, the bit priority means that rule 1 actually is the "winning" rule. In the example above "DUP" is recorded as the "winning" rule for basic OLAP Cube reporting, and this is what is presented to Business users.

Bitmaps allow rule performance to be calculated without overlap from other rules. They also allow the construction of any new model using existing rules. Let's say that we have tested a model, and found that two of the rules appear to be "malfunctioning". We would like to promote the model to production, but without the problematic rules. Unfortunately, there could be unanticipated overlap effects—when the two rules are removed, other existing rules may suddenly soak up the same traffic. Using bitmap it is possible to calculate a new model using any configuration of existing rules—just using the rule output that we have generated from the production system and without having to re-process any data through the new model. The desired rule configuration simply needs to be chosen, and then the rules that are being deactivated simply "zeroed out" from the bitmap. Next the records can be aggregated based on any bit having triggered, resulting in the new filtration performance. Thus we have a "soft simulation" that we can produce for business users that has absolutely perfect fidelity with the production system had we altered the rule mix in code.

5.5 *Model Flighting ("Multiple Models")*

adCenter Filtration Models (the logic governing filtration) are "flighted". This is done by executing multiple parallel "Candidate" models all at the same time as the Deployment algorithm is running. Each of the models then outputs its filtration results including rule reason for its decision, and these parallel assessments are passed downstream for reporting.

Flighting algorithms in this way makes it possible to explicitly score one filtration model as better than another, and also makes it possible to examine the impact of filtration as opposed to no filtration. For instance, it is possible to compare the true positive and false positive rates of two different algorithms.

Creating the parallel candidate models has also proven one of the key methods for being able to rapidly deploy model updates in a safe manner and observe them before "promoting" them to production. A final benefit is the potential of using the simultaneously evaluating models to actively execute genetic optimization of rules using each flighted model as an instance of the population.

5.6 *Redundant Keys ("Multiple Keys")*

We have sometimes been asked "what is the definition of a user in the system?" In general, the problem with creating a single user key is that it is possible to defeat any such definition.

Combinations of keys or information also can fall victim to fraudulent attackers. For example, the fraudster may generate random strings for their useragent, and approach similar to "cache busting" but designed to bust statistical models. In such a simple approach, any key built off useragent string and IP address would fail [11]. As a result effective identification of users can only be possible by deploying multiple redundant definitions of the user, and ensuring that the detection system is capable of looking for suspicious behavior at many levels and across many variables.

6 Architecture

Microsoft's filtration system has been developed to meet the incredible challenges outlined in the previous section. A schematic of the system is shown in Fig. 1. The system entails both real-time components as well as offline systems operating near real-time.

The process starts when a user visits a publisher site (1), and their browser executes a HTTP request that calls to adCenter for ads (2). This sends a request to the adCenter delivery engine (3). Within 2 ms that request is sent to the ARTEMIS (adCenter Real-Time Extendible Model Impression Scoring) real-time scoring system (4) which determines billability in real-time (Filtration) as well as calculating any

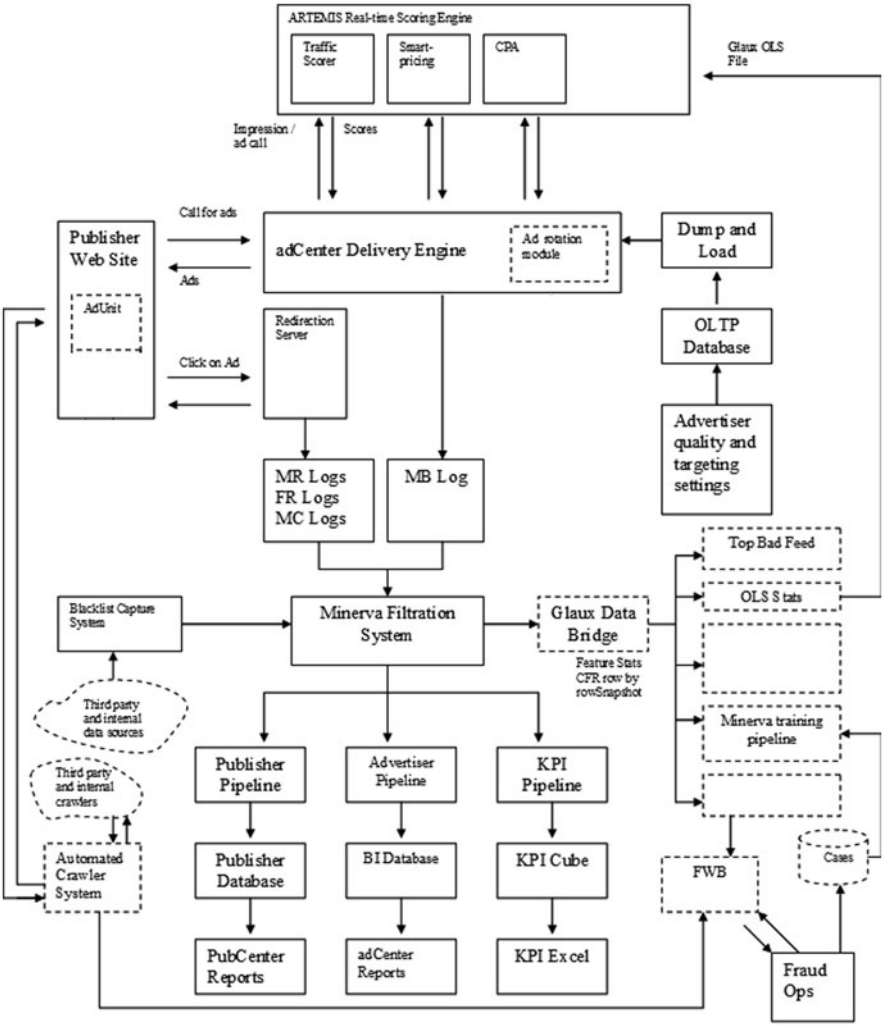


Fig. 1 adCenter traffic quality architecture

price adjustments that are needed (Smartpricing [38]) (5). ARTEMIS is also capable of supporting advertiser-controlled real-time bidding based on traffic quality as proposed in [22], and can also send instructions back to reduce the Delivery Engine's level of service specifically to that traffic if the system is under significant attack.

Assuming typical traffic quality, the adCenter Delivery Engine proceeds to hold its auction and return a set of ads to the user (7).

One might think that if traffic is known to be fraudulent, then ads should stop being served back. In fact this is almost never done. The reason is because if the Ad Server changes its behavior and stops serving ads, then attackers can use this

behavior as a kind of training signal for rapid training of their attacking programs. By continuing to serve back traffic, it both allows more data to be collected about the attacker, and also impairs the ability of attackers to probe the filtration system. In Email Spam attackers also set up accounts in order to test spam attacks, and the same techniques are used in click fraud [14].

After the ads are served back, the ads can be improperly copied, pasted, and so on since fundamentally they are HTML. Because of this the adCenter click link is encrypted and contains information about where the ad was served, and the originating request [31].

When a user clicks on those ads (8) they are redirected through the adCenter redirection server (9). This server unencrypts the link payload and records the click (10) and sends the user to their ultimate advertiser landing page. If the user then purchases a product they may execute a conversion script (11) which calls back to an adCenter server which logs the conversion event.

The click, impression, and conversion events are recorded in weblogs. These are loaded in batch fashion within the hour for processing by the Minerva offline filtration system (12). Minerva (MINing and Retroactive Analysis) is a very large, 1000 machine grid, that is designed to be non-lossy and develop the most complete picture possible of the impression, click or conversion event, the events leading up to it, and whether it is billable. In order to preserve the dynamics of the auction, Minerva respects all non-billable decisions made by ARTEMIS, and will itself only re-classify from billable to non-billable [22], but is able to bring significantly more resources to bear on traffic quality and is the final decision-maker for the system. Minerva renders the final decision on billability and flows to all downstream reporting systems so that advertiser reports (13), publisher reports (14), and internal reports (15) all show filtered data. As a result of this architecture, within milliseconds attacks are detectable using ARTEMIS, and after just a little over an hour after an impression and click was recorded, the advertiser is able to see "final" billing results.

A variety of other systems are also important for detection purposes and are used near real-time.

The Fraud workbench (16) allows human Fraud Ops team to review customers and disable their accounts. It also includes an automated machine learning module which runs every hour and creates a probability of fraud which is then provided to the human Fraud Ops team. If the customer is new and the probability is high enough, the account will be paused for a specified number of hours to allow the human Fraud team time to review the customer account. The Fraud workbench is designed in a similar fashion to the very large-scale click filtration system in that it is rules based and each decision is made visible to the Fraud Ops team. The Fraud Ops team can in turn decide to initiate action against a suspected fraudster.

adCenter's Blacklist Capture System (17) (described in [31]) was developed to pull in 3rd party IP data to help assess whether these IPs are legitimate and billable. The system is currently operational and pulling lists every 15 min.

adCenter deploys crawlers (18) to publisher sites to analyze their content and operation and compare against data collected in the course of serving ads. These crawlers analyze everything from site keywords, to looking for deceptive practices

and links to other known fraudsters. Bot instrumentation (20) is technology described in the adCenter Description of Method [31] and provides telemetry for the detailed investigation of traffic sources.

Packet sniffers (21) (internally known as "MH logs") are also described in the adCenter Description of Method [31] are special weblogs that sample 100 % of the HTTP protocol headers in the request received by adCenter. This allows for extremely deep analysis into the origin of the traffic as well as the likelihood of it being produced by automated processes. Packet sniffers operate automatically to collect detailed information on a sample of traffic, and can also be configured to collect all records from particular data sources.

7 Metrics

There are two major ways to define traffic quality (a) true-positive, false-positive, detection rates of confirmed fraudsters, and (b) overall traffic quality metrics that encompass a lot of information about the traffic. adCenter utilizes both approaches.

7.1 Case Base

adCenter is fortunate to be able to collect confirmed fraudulent cases because of the its highly expert and dedicated investigation team. Approximately 40 traffic quality investigation tickets per month are collected. These cases are saved into a "case base" which is literally a copy of logs, but where they are known to be generated by a particular bot. Historical logs with these known tags can then be replayed against the filtration system to see if it detects the known attack.

7.2 Traffic Quality Metrics

We also define Q1 and Q2 metrics which each measure "value per click" compared to typical traffic where 1.0 indicates standard traffic, and values greater than 1.0 indicate less valuable traffic. These metrics do not attempt to measure fraud per se, but instead measure the traffic quality or marketplace health for advertisers on the system.

8 Detection Techniques

The rules used for detection fall into seven major categories described below:

8.1 Bot Signatures

Every week the engineering team reviews with the support team to look at new types of attacks. The behavior of these attacks is analyzed, and if necessary new rules are developed to combat them. The key is to store literally the electronic format data that the system would have processed, but to separately have a case label for these records. It is then possible to replay the traffic through the system and determine its effectiveness in detecting the historical attack.

8.2 Distribution Tests

Daswani et al. [8] describe detection in which an expected distribution is compared against an actual distribution. To the degree to which the distributions diverge, the traffic may be artificially generated.

8.3 Scale Families and Reference Curves

We often get questions about how we handle proxy IPs. These are IPs such as AOL or Mobile carrier IPs which service a large amount of traffic. Might not we filter an awful lot of good traffic if we are applying our basic frequency caps and other rules?

The basic approach for handling these IPs is to create a family of versions of a rule but at different scale. For example, say that we have a rule BOT1 which is working very well to identify traffic with less than 20 clicks. We can create another version of this rule that works at 200, 2000, and 20,000 clicks. At these levels of scale the anomaly or signature that is being detected is usually much more subtle and so it doesn't need to register comparatively very high, however with the large number of clicks it is statistically significant. The end result is a family of rules eg. BOT1-A..BOT1-E that are designed to operate at different levels of scale, and which naturally handle proxy IPs as well as other large sources of traffic.

The "scale family" is a discretized version of a significance test which takes into account the number of observations when determining whether a variable is statistically significant/significantly different from the norm.

8.4 Traps

Traps are special tests designed to identify bot activity. These generally utilize "active" methods outlined by Daswani et al. [8].

8.5 Extremata

Extramata are rules designed to identify and remove extreme behavior [3, 4]. In general trying to identify the unusual extremes can help to remove activity that is not typical of focused human browsing behavior as well as pre-emptively "protecting" the system from robotic behavior that has yet to be encountered. In general, many extremata can be pre-loaded into the system to catch highly anomalous conditions.

8.6 Key Families

In order to be effective, fraudsters need to camouflage their attack perfectly in multiple dimensions. This leads to an important strategy for success. Detection should not just use one or two criteria. It should use as many as possible.

In practice often an effective rule can be developed that looks for activity from an attribute of interest such as an IP. Similar rules can often be developed which look for the same kind of unusual behavior coming a different levels of aggregation from other attributes—such as publisher, advertiser and so on. We call these key families. Therefore even if an attacker cloaks some of their activity they are unlikely to remain hidden in every dimension and the redundant key family rules will tend to detect them.

8.7 Machine-Induced Decision Trees

The final kind of rule is of course the machine learning induced rule. adCenter currently uses a C4.5-like induction method described in [21]. Although these rules are effective, they are less interpretable and are used more in cases where traffic quality is being assessed rather than specific bot signatures.

9 Results

Microsoft adCenter's filtration system has been in operation for over 5 years. In this time the systems have been steadily improved and enhanced.

9.1 Automated Detection Performance

We can summarize the overall performance of the system by looking at filtration rates. These provide some information on how much traffic is being automatically

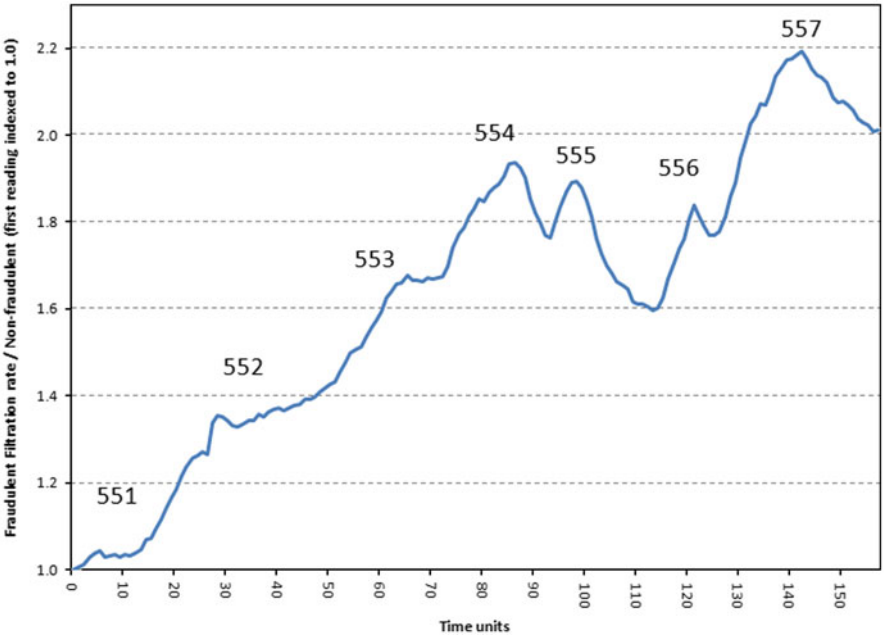


Fig. 2 Filtration rate of fraudulent publishers versus non-fraudulent publishers, expressed as a ratio of the fraudulent rate to non-fraudulent. *x*-axis is time, *y*-axis is ratio, and indexed to the ratio on the first date in the timeseries above. The numbers indicate major product releases

flagged. We would expect that fraudsters should show higher filtration rates, although it is also true that high filtration rate does not necessarily imply fraudulent activity. There are some cases in which advertisements may have been placed improperly attracting lots of accidental clicks for example.

Figure 2 shows the filtration rate ratio for fraudulent publishers versus non-fraudulent over the same time period. This shows clearly that fraudulent publishers are being filtered more aggressively.

Figure 3 shows that fraudsters are shifted in terms of their filtration rate with a mode that is 1.8x normal.

Table 1 shows if the publisher is a fraudster, then they are likely to be filtered at a rate that is 1.49x higher than normals, with the 10th and 90th percentiles ranging between 0.9x and 1.94x. The difference in distribution of filtration rates between fraud and non-fraud is statistically significant with $p < 0.01$ under Wilcoxon Rank-Sum Test.

What do fraudulent publishers look like? Although we cannot go into details, we can show what happened during some of those releases. Figure 4 shows the days leading up to, and just after, one of our model updates. Four fraudulent publishers suddenly had their filtration rates go to 100 %. The investigation team was alerted to these cases because of the high filtration rate, and proactively investigated them to determine the cause of the filtration. They confirmed that this was indeed fraudulent

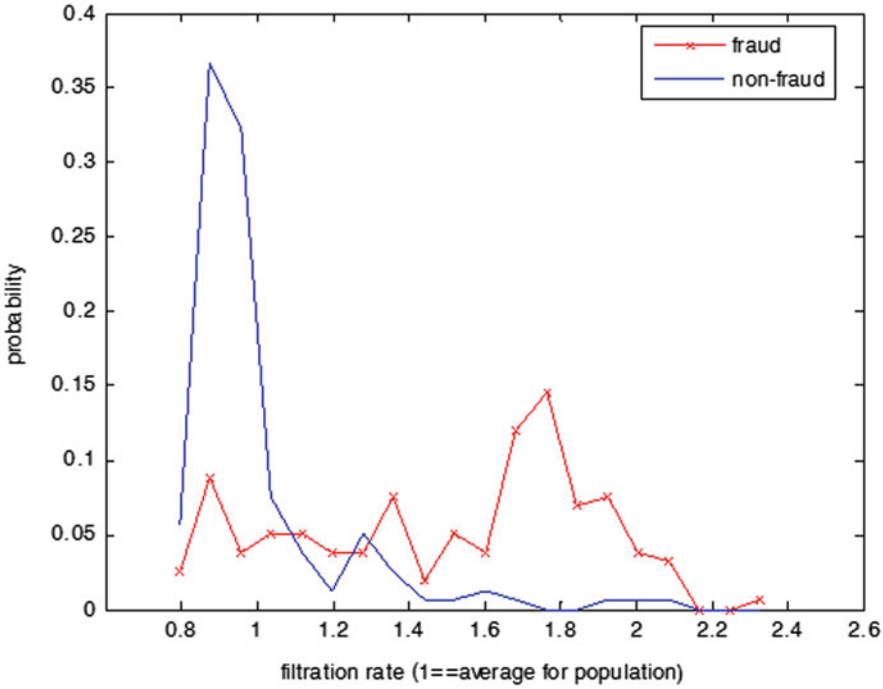


Fig. 3 Filtration rate ratio for fraudulent versus non-fraudulent publishers

Table 1 Filtration rate for fraudulent versus non-fraudulent publisher (1.0 = average for population)

Filtration rate (ratio vs. population)	Fraudulent publisher	Normal publisher
Mean	1.4942	1.0000
Variance	0.1456	0.0479
90th pctl	1.9404	1.2794
10th pctl	0.9038	0.8508

activity, and the rules that were triggered were some specifically geared to particular bots. The support team tried to reach out to these publishers but found that their accounts had been abandoned, and the publishers stopped requesting traffic less than a week after their filtration went to 100 %.

9.2 Click Fraud Investigation Team

Table 2 shows reasons for fraudulent account take-downs provided by the human support team, as well as the true positive rate for those reasons. True positives are

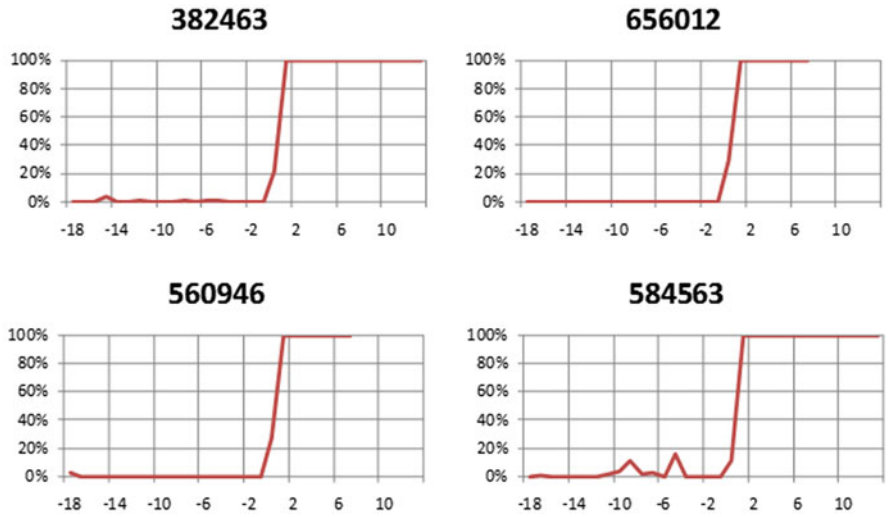


Fig. 4 Filtration rates for four fraudulent publishers. After a rule update their filtration rates went to 100 %. The time-axis shows days leading up to a model update and following the model update

Table 2 Click quality investigations team reason for investigation and true positive rate

Percent (%) of true fraud revenue detected	TP rate (%) (accounts)	Reason
61.09	75	High invalid click rate
18.85	50	R1
9.88	63	R2
8.79	75	R3
1.09	50	R4
0.29	33	R5

finalized only after sometimes lengthy investigations in which the investigation team is able to determine whether a customer is engaging in fraud or is not.

Most of the fraudulent revenue identified and remitted by the investigation team was found after seeing a high invalid click rate (61 %). The true positive rate for investigations triggered by high invalid click rate is also around 75 %, which makes it the top performing detection category given the volume of fraudulent activity being removed. In addition, the high quality of automated detection has not only improved the fraud team's accuracy and speed of detection, but has also freed it up to spend more time conducting deep investigations.

Table 3 adCenter rule categories and Q1 and Q2 traffic quality metrics

Category	Percent (%) clicks	Q1	Q2
Billable	82.63	1.0	1.0
Double click	6.48	1.9	1.8
Known bot	2.47	937.8	1773.6
Staleness	2.13	2.4	3.5
User freq cap	1.92	7.2	12.7
Suspicious	1.92	2.5	3.3
Bad proxy	1.08	84.3	71.2
Business non-billable	0.70	1.6	1.8
Refractory	0.35	3.1	1.9
Outlier	0.15	3.2	4.2
Defective	0.12	1.3	0.9
IAB browser list	0.04	2.8	3.3
IAB robots and spiders	0.02	1.7	2.4

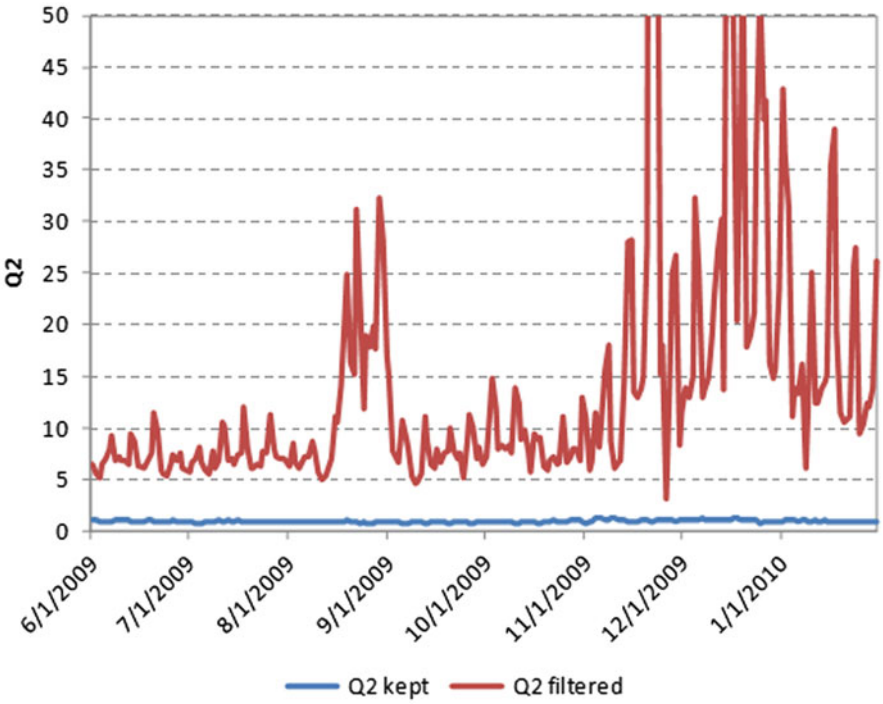


Fig. 5 Q2 statistic for filtered (*upper irregular line*) versus kept traffic (*lower line*). The upper line has Q2 rates ranging from 5 to over 50. This activity is variable because of ongoing fraud and bot activity. The lower line which is billed traffic is extremely stable. adCenter is filtering out the activity in the upper line and trying to maintain good performance for advertisers

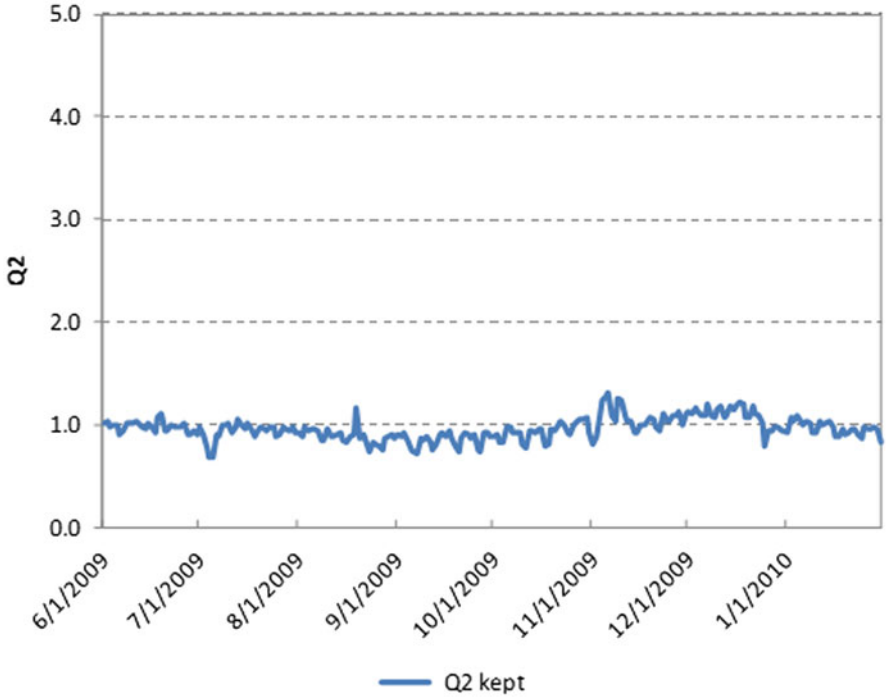


Fig. 6 Close-up of the Q2 timeseries for kept traffic. The filtration system helps to ensure that the metric varies by only a few percent from its average value across 6 months of campaigns, despite rampant click fraud attacks that are underway

9.3 Traffic Quality

Although the effectiveness in detecting known fraudsters is promising, it remains to be seen what is the overall effect on the advertiser in their day-to-day advertising? We can measure traffic quality using our Q1 and Q2 metrics that we introduced earlier.

Table 3 shows traffic quality for major classes of rules used in adCenter.

Known bots clearly have very bad traffic (e.g. Q2 = 1773). A range of other rules are also shown, some of which may not necessarily be fraudulent, but for which the business has decided not to bill such as Defective traffic (which is actually a little better than the norm at Q2 = 0.9; so is likely human traffic, but for which we can't bill due to errors in the click request or out-of-date account information). Interestingly, the IAB Robots and Spiders List [5]—which represents a commendable industry effort to track and catalog bots for companies to implement industry standard filtration—produces only 0.02 and 0.04% additional filtration in our system (Q2 = 3.3 and 2.4).

Figure 5 shows traffic quality for filtered traffic as well as kept traffic. The filtered traffic undergoes massive spikes in the Q2 measure, from being consistently about 5x

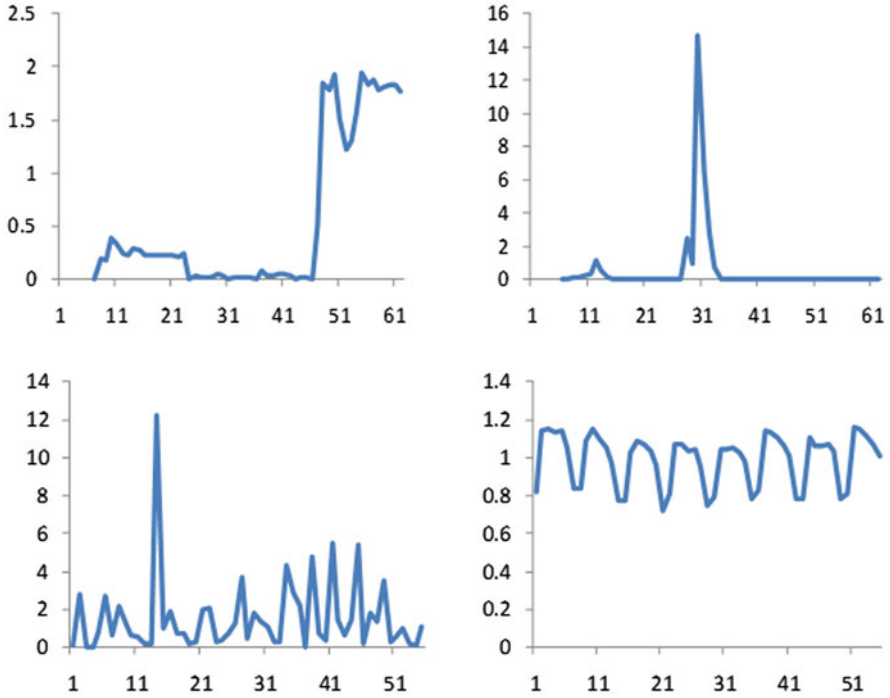


Fig. 7 Sixty-day timeseries for four rules. The first three are all discrete rules designed to look for bot activity. The top left is a clearly robotic process that runs at different levels of aggressiveness. The top right is a burst attack that was observed at day 31 and then disappeared. The bottom left shows continuous attack activity. The final graph on the bottom right shows the post-filtration timeseries. By removing the bot activity and leaving in place a clean, human timeseries, Microsoft is better able to optimize its ad engine for users, advertisers and publishers

worse than billable traffic to sometimes as high as 50x. In addition, the irregularity of the traffic shows that this traffic would be extremely disruptive for advertisers that expect a consistent standard of traffic and value.

The kept traffic line (Fig. 6), in contrast, is extremely stable. Indeed, the metric has shifted by only a few percent over the period. This shows that adCenter is delivering good value to advertisers, protecting them from bad traffic, and is maintaining consistent value over a long period of time.

Figure 7 (bottom right subplot) shows post-filtration, billable clicks, which is the same data that are reported throughout Microsoft, sent to Advertisers on their billing reports, reported to Publishers, and so on. The constituent bot traces, artificial level changes and spikes are gone, leaving a human-looking profile. By removing the bot activity and leaving in place a clean, human timeseries, Microsoft is better able to assess the performance of its online services, as well as ensuring that machine learning systems throughout the platform are learning on human data, allowing adCenter to work to maximize user search relevance, advertiser performance and publisher value [44].

10 Conclusion

Click Fraud is a major challenge for online advertising. Effective execution requires constant investment and development. We have discussed the design of Microsoft's click filtration systems, and the choices that were needed to operate at massive scale, and to detect sophisticated adversarial attackers. We believe that the design that we have developed—including real-time and near-real-time components, rapid update capabilities, and so on—should translate well to other large-scale fraud detection domains.

Acknowledgments We would like to thank Raj Mahato, Albert Roux, Ron Mills, Brandon Sobottka, Matthew Rice, Sasha Berger, Jigar Mody, Dennis Minium, Kamran Kanany, Tudor Trufinescu, Dinesh Chahlia, Ken Pierce, Hank Hoek, Tao Ma, Karl Reese, Narayanan Madhu, Dimitry Berger, Rageesh Maniyembath, Meena, Joseph Morrison, Kiran Vemulapalli, Anthony Crispo, Matthew Bisson, Igor Chepil, Matthew Ford, Sachin Ghani, Amjad Hussain, Steve Marlar, Bill Morency, Gerry Moses, Steve Sullivan and many others.

References

1. Boyd, C.: IE used to launch instant messaging and questionable clicks. http://blog.spywareguide.com/2006/10/ie_used_to_launch_instant_mess.htm (2006)
2. Buchanan, B., Shortliffe, E.: Rule-Based Expert Systems. Addison-Wesley, Reading (1984)
3. Buehrer, G., Stokes, J., Chellapilla, K.: A large-scale study of automated web search traffic. Proceedings of the Fourth International Workshop on Adversarial Information Retrieival on the Web (AIRWEB) (2008)
4. Buehrer, G., Stokes, J., Chellapilla, K., Platt, J.: Classification of automated search traffic. In: King, I., Baeza-Yates, R. (eds.) Weaving Services and People on the World Wide Web, pp. 3–26. Springer, Berlin (2008)
5. Bureau, I.A.: Iab/abce international spiders & bots list. http://www.iab/iab_products_and_industry_services/1418/spiders (2010)
6. Claburn, T.: Microsoft sues three for click fraud. InformationWeek (June 2009)
7. Court, U.S.D.: Microsoft vs Eric Lam et. al. Civil Case Number CO 9-0815. http://graphics8.nytimes.com/packages/pdf/business/LamComplaint.pdf (2009)
8. Daswani, N., Mysen, C., Rao, V., Weis, S., Gharachorloo, K., Ghosemajumder, S.: Online advertising fraud. In: Crimeware : understanding new attacks and defenses, Chap. 11. Symantec Press (2008)
9. Daswani, N., Stoppelman, M.: The anatomy of clickbot a. Usenix HotBots 2007 (2007)
10. Edelman, B.: The spyware—click-fraud connection—and yahoo's role revisited. http://www.benedelman.org/news/040406-1.html#e1 (2006)
11. Fielding, R. et al.: Hypertext transfer protocol – http/1.1. Tech. Rep. RFC 2616, Network Working Group (1999)
12. Gandhi, M., Jakobsson, M., Ratkiewicz, J.: Badvertisements: Stealthy click-fraud with unwitting accessories. In: Online Fraud, Part I J. Digital Forensic Pract., vol. **1**, Special Issue 2 (2006)
13. Ghosemajumder, S.: Findings on invalid clicks. http://googleblog.blogspot.com/2006/03/update-lanes-gifts-v-google.html (2006)
14. Goodman, J.: Spam filtering: Text classification with an adversary (2003)

15. Google: Google ad traffic quality resource center. http://www.google.com/adwords/adtrafficquality/ (2010)
16. Google: Google iab click measurement description of method. http://adwords.google.com/support/aw/bin/answer.py?hl=en&answer=153707 (2010)
17. Google: Google Form 10-Q Quarterly Report Pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934 (2012)
18. Jackson, C., Barth, A., Bortz, A., Shao, W., Boneh, D.: Protecting browsers from dns rebinding attacks. Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 421–431 (2007)
19. Jansen, B.: The comparative effectiveness of sponsored and non-sponsored results for web ecommerce queries. ACM Trans. Web **1** (2007)
20. Jansen, B., Flaherty, T., Baeza-Yates, R., Hunter, L., Kitts, B., Murphy, J.: The components and impact of sponsored search. Computer **42**, 98–101 (2009)
21. Kitts, B.: Regression trees (2000), unpublished manuscript
22. Kitts, B.: Click fraud protector. US Patent Application, (2006)
23. Kitts, B.: Introducing adcenter clickids. http://community.microsoftadvertising.com/blogs/advertiser/archive/2009/06/17/introducing-adcenter-clickids.aspx. (June 2009)
24. Kitts, B., Laxminarayan, P., LeBlanc, B.: Cooperative strategies for keyword auctions. First International Conference on Internet Technologies and Applications (2005)
25. Kitts, B., Laxminarayan, P., LeBlanc, B., Meech, R.: A formal analysis of search auctions including predictions on click fraud and bidding tactics. ACM Conference on E-Commerce Workshop on Sponsored Search (2005)
26. Kitts, B., LeBlanc, B.: Optimal bidding on keyword auctions. Electron. Markets Int. J. Electron. Comm. Bus. Media **14** (2004)
27. Kitts, B., LeBlanc, B., Laxminarayan, P.: Click fraud. American Society for Information Science and Technology Bulletin, pp. 20–23 (December 2006)
28. Kitts, B., Najm, T., Burdick, B.: Identifying automated click fraud programs. US Patent Application, (2006)
29. Leyden, J.: Botnet implicated in click fraud scam. http://www.theregister.co.uk/2006/05/15/google_adword_scam/. (May 2006)
30. Leyden, J.: Click-fraud menace spreads using IM. http://blog.spywareguide.com/2006/10/ie_used_to_launch_instant_mess.html. (Oct 2006)
31. Microsoft: Microsoft adcenter click measurement description of method. https://adcenterhelp.microsoft.com/Help.aspx?market=en-US&project=adCen ter_live_Std&querytype=topic&query=MOONSHOT_CONC_ClickMethod.htm (2009)
32. Microsoft: Microsoft Form 10-Q Quarterly Report Pursuant to Section 13 or 15(d) of the Security Exchange Act of 1934 (2010)
33. Mungamuru, B., Garcia-Molina, H.: Managing the quality of cpc traffic. Proceedings of the 10th ACM Conference on Electronic Commerce, pp. 215–224 (2008)
34. Mungamuru, B., Garcia-Molinja, H.: Predictive pricing and revenue sharing. Proceedings of the 4th International Workshop on Internet and Network Economics, pp. 53–60 (2008)
35. Mungamuru, B., Weis, S.: Competition and fraud in online advertising markets. In: Tsudik, G. (ed.) Financial Cryptography and Data Security, pp. 187–191. Springer, Berlin (2008)
36. Mungamuru, B., Weis, S., Garcia-Molina, H.: Should ad networks bother fighting clickfraud (yes, they should.). Technical Report 2008-24, Stanford InfoLab (2008)
37. Nielsen: Nielsen reports December U.S. search rankings. http://blog.nielsen.com/nielsenwire/online_mobile/nielsen-reports-december-u-s-search-rankings/ (2010)
38. Rey, B., Kannan, A.: Conversion rate based bid adjustment for sponsored search auctions. WWW. (April 2010)
39. Schonfeld, E.: The evolution of click fraud: massive Chinese operation DormRing1 uncovered. http://techcrunch.com/2009/10/08/the-evolution-of-click-fraud-massive-chinese-operation-dormring1-uncovered/ (2009)

40. Weinberg, N.: Google wins click-fraud case vs auction experts. http://www.webpronews.com/topnews/2005/07/05/google-wins-clickfraud-case-vs-auction-experts. (July 2005)
41. Whitney, L.: Bing grabs 10 percent of search market. http://news.cnet.com/8301-10805_3-10354394-75.html. (Sept 2009)
42. Wikipedia: Cross-site request forgery. http://en.wikipedia.org/wiki/Cross-site_request_forgery (2012)
43. Woolsey, B., Schulz, M.: Credit card statistics, industry facts, debt statistics. http://www.creditcards.com/credit-card-news/credit-card-industry-facts-personal-debt-statistics-1276.php (2010)
44. Wu, G., Kitts, B.: Experimental comparison of scalable online ad serving. In: Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1008–1015 (2008)
45. Yahoo: Yahoo search marketing click measurement guidelines description of method (2009)
46. Yahoo: Yahoo traffic quality center. http://searchmarketing.yahoo.com/trafficquality/ (2010)

A Novel Approach for Analysis of 'Real World' Data: A Data Mining Engine for Identification of Multi-author Student Document Submission

Kathryn Burn-Thornton and Tim Burman

Abstract In this article we describe a data mining engine which makes use of a new approach to plagiarism detection. The new approach which we have taken identifies student submissions which have been produced by more than one author and hence provides a starting point for investigation of a student submission which may contain plagiarized material. The approach, which this engines uses, has great potential for use by those marking submissions from two types of student bodies. Namely large class size with whose written styles they may not be familiar and students following online courses who they may not ever meet. The approach which we have taken is new in that other approaches endeavor to match the submitted material with material existing elsewhere whereas our approach attempts to determine multiple author styles in the submission and hence provide an indication that the submission contains information from more than one source. The implications of the use of author styles for identification of future suspect submissions, and for comparison with future submissions by the same student, are discussed.

1 Introduction

Government cuts in Higher Education funding have provided a greater driver for larger university class sizes, both face-to-face and online [6]. With online delivery of the lectures, rather than blended learning, the lecturer, or marker, may not ever meet the student face-to-face [19, 22]. Class sizes greater than 50 can also result in a similar problem in that those marking an essay style submission may be unaware of the written style of the students and, in many cases, unable to put a name to a face [3, 4].

This lack of knowledge of the student puts the marker at a great disadvantage and provides a window of opportunity for those who are aware of the situation and who

K. Burn-Thornton (✉)
OUDCE, University of Oxford, Oxford OX2 7DD, UK
e-mail: kathryn.burn-thornton@conted.ox.ac.uk

T. Burman
School of Computing and Engineering Science, University of Durham, Durham, UK
e-mail: tim.burman@dur.ac.uk

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_11

are keen to achieve grades/credits by the easiest route and reuse material which may have been created by others i.e. those who are willing to plagiarize existing material.

Such activity is readily facilitated by the virtual society which now makes it possible for students to access material from all over the world and with which the marker may not reasonably be expect to be aware [4].

Approaches which have been taken to ameliorate this problem include continual assignment and changes of the assessment topics and sub-topics. However, with the current proliferation of HE, and FE institutions, it is not possible to ensure that they do not overlap with others set somewhere else in the world [3]. Despite this problem, identification of whether the student's submission contains a duplication of information which may be found elsewhere on the superhighway is an approach to solving the plagiarism problem which may be ideally suited to a software tool [13].

Although identical duplicate documents, paragraph content, to those of some student submissions may be readily found by making use of a simple search engine [9], submissions which contain modification of documents from many sources are harder to detect by this approach and a more sophisticated approach must be used to identify these. This is particularly so when the focus for many is now the qualification which is achieved, at the end of a period of study, rather than gain subject knowledge en route to the qualification.

This change in student aspirational focus resulted in a 100-fold growth, over the last 10 years, in published papers which outline approaches, and software tools, which may be used to provide aid in the detection of student plagiarism by universities [7].

However, the results from the use of such plagiarism tools are often hard to act upon using formal university procedures because of their determination of degree of commonality between the student submissions and other documents which are available [1, 2, 8, 9]. In addition, the tools do not always provide those investigating the submission of interest with an indication of whether, or not, the submission is individual original work.

An approach which has not been used to identify student submissions of interest, which may emanate from more than one author, is document signature style. This approach has an added advantage in that it is easier to follow up the results obtained using formal university procedures if required.

This article describes a novel approach, based upon a data mining engine, which enables documents to be identified which have been written by more than one author by virtue of identification that the document is composed of more than one signature style.

Section 1 describes current approaches which are used to identify 'suspect' student submissions. This is followed by a discussion of two possible solutions which would enable document signature styles to be determined and a description of techniques which may be employed in order to achieve each of the potential solutions. Algorithms which may be gainfully employed in achieving each solution are then described. Then an overview of the sub-tasks carried out by the algorithms which have been used to implement the proposed solution follows. The remaining sections

discuss the investigations which were carried out in order to determine the effectiveness of the approach, the metrics which were used to determine the effectiveness and the results of the investigations for the CSS type solutions—the ASS based solution. Conclusions regarding the results of the investigations are then drawn with future profitable avenues for investigation being discussed.

2 Existing Approaches

The vast majority of tools, in common use in a university environment, which enable the investigation of submission of non-original work such as TurnitinUK and Viper [2, 7] appear to make the simplest assumption that the submission of non original work by a student falls into the category of potential plagiarism. With this prior assumption that determination of plagiarism may achieved by comparing the student's submission with all other submissions, and documents, which are available throughout the world.

The process is readily suited to current pattern matching algorithms, and methods, especially if paragraphs similarity between documents is to be considered. It is a type of pattern matching engine which underpins plagiarism tools which are commonly used in a university environment to identify potential plagiarized submission [2, 9].

Despite their speed of document comparison most tools of this type present those investigating the student submission of interest with a problem. Namely, that unless the submission is a 'simple' combination of existing work many of the current plagiarism tools do not provide sufficiently large a percentage match between the student's submission and documents which may be available on the web in order to pursue further the investigation of the lack of originality, or degree of multi authorship, in the submission using formal university approaches [5].

However, another approach to detection of non-original work in the student submission could prove profitable when the pattern matching approach fails, that of the author signature style [10, 16, 23] (ASS) since all student submissions should emanate from one student so should contain only one ASS or a variant on the same ASS.

3 Document Signature Style

Document signature style makes the assumption that each individual has a unique writing style which is characterized by their individual use, and combination, of nouns, verbs and a other features which include referencing [10, 17, 23]. If the document signature style were to vary throughout a document's paragraphs, pages and chapters this could provide an indication that the submitted document originated from more than one author and was not the submission from one individual. Such variation in style could be used as a basis for a formal university approach as the student submissions profess that the word is their individual work, in other words from

only once source. This approach could, if sufficiently accurate, prove to enable the task to be achieved faster, and hence enable more student submissions to be checked, because all the information in cyberspace is not be trawled for each submission. The following section outlines how such a solution can be achieved.

3.1 Extraction of Signature Style

In order to determine the unique author signature(s) present in the electronic submissions it necessary to determine key elements of documents written by student which can be used to determine a unique documents signature created by each student.

Initial analysis of over 300 submissions, from 50 individual students, each having a maximum set word count of 10,000 words, written in English, in one university School [5] suggested that there are eight key features of the signature required in order to determine whether, or not, a document emanates from one author. These key feature are number of words in a sentence, number of lines in a paragraph, paragraph formatting, degree and use of grammar, type of language used, word spelling and referencing style.

In order to determine these key elements of the signature style 30 elements of the documents were assigned values for each of the 50 students and then an investigation was carried out in order determine the minimum number of these elements, and which elements, it was possible to use in order to assign each of the six pieces of work to the student who had written it. A grid, part of which is shown in Fig. 1, was produced to aid in the visualization of the results.

The figure shows that using variables Var2–5, 7–8, 11–12 that all submissions are accurately attributed to the students who submitted them. These variables numbers correspond to number of words in a sentence, number of lines in a paragraph, paragraph formatting, degree and use of grammar, type of language used, word spelling and referencing style.

The figure also shows that using more than eight key elements results in not all pieces work being assigned to accurately to the students and that the use of less than eight elements also being inaccurately assigned to the students. In other words there are eight key feature values of a ‘student signature’.

These key signature features of the ‘student signature’ are concomitant with those proposed at ICADPR for general documents, for instance those in [17] and [20], but differ because of approach used to determine commonality in key features, the type of document being considered and also the language focus of interest.

The work described in ICADPR predominantly focuses upon both signature features required for a plethora of languages and also for those describing documentation available in the outside world. The signature features are also considered for both type written and hand written documents.

However, our approach focuses upon the work produced by the individual and does not compare it with other documents available, only those submitted by that student. In effect our approach determines self-consistency within the documentation

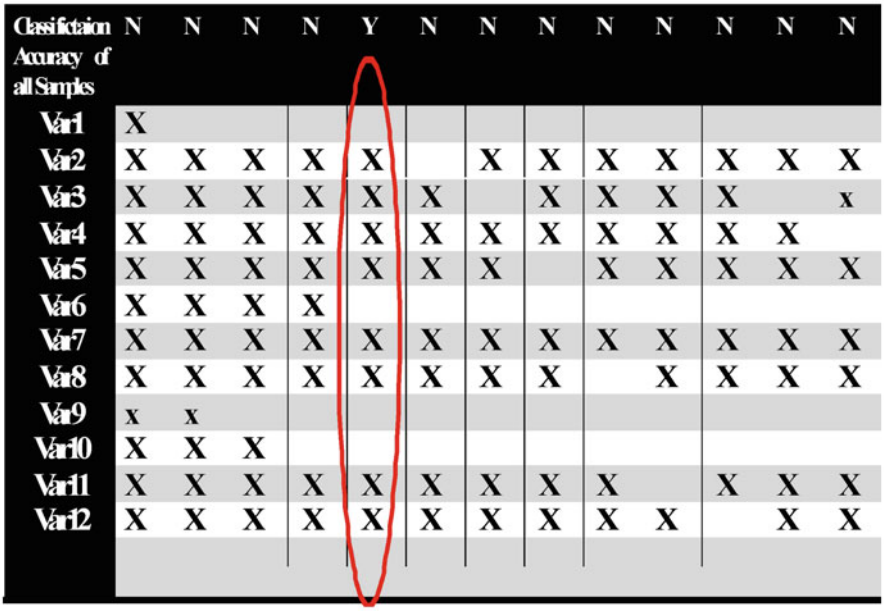


Fig. 1 Portion of grid showing features and classification accuracy of samples: *Y* denotes all samples correctly attributes, *N* denotes not all samples correctly attributed. Var2–5, 7–8, 11–12 correspond to those used

submitted by each student. Our work is also only concerned with electronic text which is nominally written in 'UK' English.

The first two elements of the 'student' signature are self explanatory but the others may require some clarification. Degree and use of grammar to include the manner in which infinitives are used; use of, and types, of punctuation; use of plurality. Type of language is taken to mean language style which is used in different types of English for instance UK and US. However, word spelling includes not only language spelling differences such as those found between UK and US, for example as in counsellor and counselor, but also frequency of typographical errors and spelling mistakes. The referencing style required by different bodies, and institutions, vary and can provide an indication of material which originates from more than one source.

A solution to analyzing this information would be an approach which is able to extract the key signature elements, and their values, from paragraphs, and pages, and compare them with others in the same document and with those extracted from other documents. It could also be helpful if the approach used could be used, during any subsequent university formal procedures, to show how the document would have appeared if written by a sole author. Such documentation would prove useful if additional proof of multi-authorship was required.

The following section describes two possible variants on such an approach.

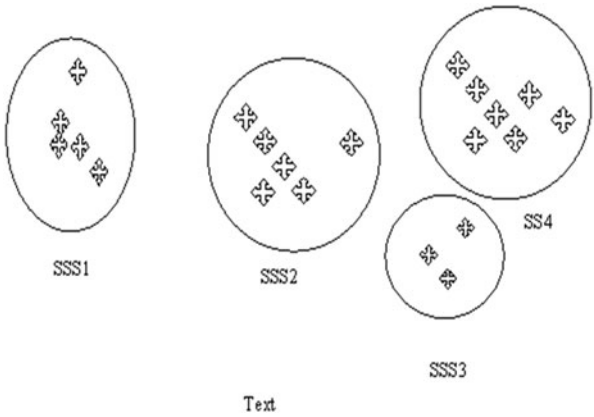


Fig. 2 Pictorial representation of clustering of a document into four ASSs, the cluster boundary being depicted by a *solid line*

4 Possible Approaches

Both of the possible approaches suggested in this section make use of a modification of the approaches which we used in our web site maintainability tool [15]. The approaches make use of Cascading Style Sheets (CSS) or a combination of the eXtensible Markup Language (XML) in combination with the eXtensible Style Language (XSL) [26].

These approaches make use of information extraction and representation. Some commonality can be observed between the first steps of the approaches, which are described in the next section, and that of Ghani [18] and Simpson [24].

4.1 CSS

If a CSS-based approach was used, a named author signature style (ASS) could be defined which would describe the values assigned to the key signature features. Once the ASS files were created, the signature of style of the author could not only be compared with others within the same document but it could also be applied to any document section and the output compared with that contained within the current, or other, submitted document. By using this approach the speed of investigation of submitted documents could be minimized by the reduction in the size of file which is required in order achieve comparison [25].

In practice, each section of the document being investigated could be converted directly to a section of ASS containing the feature values. Such an approach would require the use of a measure of uncertainty when mapping the samples of document and related ASS code to named signature styles. Figure 2 provides a visual representation how a page of text may be converted using such an approach.

Figure 2 shows that four ASSs have been produced from a sample document with ASSs 1–4 having respective membership of 5, 6, 3 and 7 document sub-samples.

Data mining would appear to be able to provide a solution to this problem by making use of modified clustering techniques. The only drawback to this approach is that a library of assignable values for each key signature feature will need to be defined initially. However, this library could be updated as part on an electronic submission process.

4.2 XML

For an XML approach all content information would be contained in an XSL file with its companion XML file containing the ASS feature information which would be recursively applied to the XSL document.

Using the example from Fig. 2 this approach would result in the production of a XML file containing a section of text that would be marked up as a reference name, and the XSL file would contain a template which could be applied to reference names in that document. Such an approach would readily facilitate comparison of documents because it would be relatively easy to target comparison of documents by investigation of specific signatures, ASSs.

Rigid definitions do not exist for XML tags which means that any appropriately defined names will have to be used in the XML file as well as a library of attributable values of the signature features, as in the CSS approach. However, a major drawback of this approach would be the need of consistency for XML tags and the possibility ongoing modification to a centrally accessed XML tag dictionary.

The requirements which will need to be fulfilled for the XML/XSL solution suggest that the CSS-based solution may be the more accurate approach to use for the comparison of signature styles in documents. This is because even a slight variation is XML tags could result in a large discrepancy in ASS and hence identification of a document as containing information from more than one author when it does not.

The following section provides an introduction to data mining, which will be used as the basis of the CSS, or ASS, approach.

5 Data Mining

Data mining finds novel, potentially useful and ultimately understandable patterns from mountains of data [14] and has been used to mine data from diverse domains including the medical domain [12], pharmaceutical [11] and, as such, appears to be the ideal solution for finding the patterns of information contained within the files extracted from (and contained within) the student submissions. This is an approach which we used in our web maintainability tool [15].

Data mining can determine the patterns by clustering the data according to variable values contained in the data [21]. Figure 2 shows how clustering could be carried out using pre-determined CSS, or ASS, files and unclassified student submissions shown

in red. In this example each sample in the classifier, ASS 1–4, is marked with a cross indicating the document page giving rise to the sample, and the CSS section (ASS section) that was generated from the document. Samples that have similar values, appear to have been produced by the same author, are given the same classification. The figure shows that ASSs 1–4 having respective membership of 5, 6, 3 and 7 document sub-samples.

In clustering, each CSS section would be classified in turn. If it is sufficiently similar to other previously classified sections, ASS, it is added to the same classification (class) as these other sections. If it is not sufficiently similar to another section, a new classification, a new ASS, is created. In this particular case the 'unknown' sample is sufficiently similar to ASS2 to be classified as ASS2 or as belonging to the student whose submission contains ASS2 feature values.

There are many different classes of Data mining algorithm which can perform clustering with each class possessing different properties. It is these different properties which make each class suitable for analyzing different types of data [21]. The class of algorithms which appear to be particularly appropriate for mining the type of data of which CSS files are composed belong to the statistical and machine learning classes of algorithms. More information regarding this may be found from the results of the STALOG project [21]. These classes of algorithms are described, briefly, in the following section.

5.1 Suitable Data Mining Algorithms

The suitability of algorithms chosen from the statistical and machine learning classes, namely: k nearest neighbors, linear (k-NN), quadratic and logistic discriminants, k means, rule based, decision trees and Bayesian classifiers are described and their appropriateness for the task in hand. These are the same algorithms which were discussed for the task of website maintainability [15]. The reasons behind the choice of algorithm for the task are discussed in the final sub-section.

The most appropriate algorithm for the conversion from student document to CSS, ASS, from those listed above, is the k-NN algorithm, or a variant of such. The other algorithms are not appropriate because they either require too many samples with which to build an effective model from which to work effectively in this application (decision trees, Bayesian classifiers), require numerical data (Fisher's linear discriminants), or require prior knowledge of the classes (K means).

However, k-NN can work effectively with a small number of samples, can work with categorical data given an appropriate function to compare two samples, and does not require any prior knowledge of the number of classes, or authors. This is particularly important, and useful, because of the small cohort size of niche cohorts which may be found in postgraduate courses.

The following sections describe the implementation of the CSS solution which has been described in this section.

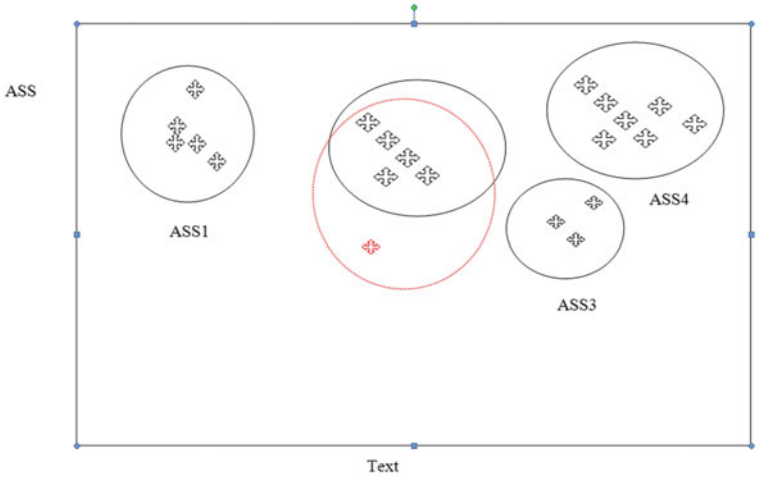


Fig. 3 Visual representation of ks-NN classifying an 'unknown' submission against four 'known' ASSs

6 CSS Solution

In order to implement the k-NN algorithm, or variant therein which we are calling ks-NN, some means of finding a numeric difference between a sample of student document and an existing ASS is required. This can be achieved by determining the signature features in one sample which are also present in an existing ASS, as well as the values which have been assigned to the signature features in each sample.

A visual representation of the approach used to determine the difference between the ASS signature features values and those present in each in each 'unknown', or unattributed, sample signature, may be seen in Fig. 3. This figure shows that the 'unknown' sample is nearest, within the threshold distance, to the signature feature values in ASS2.

In order to achieve this each section of student document submission needs to be represented by equivalent signature features and their values. In the same manner as presentation tags in HTML code these can be represented as signature tags. It is these adjacent signature tags which form clusters of tags and can be represented by a single ASS.

The first stage of the implementation of the ks-NN algorithm is to create the signature tags from the original document and then each cluster of signature tags is converted to a ASS sample using a set of rules that are defined in a data file. This can be changed by the user as the ASS evolves, but a standard set of rules.

Each line is in the format: Tag-name ASS-equivalent Value

After each cluster is converted to an ASS the algorithm iterates through each sample and compares it to any that have already been classified. At the start of the loop, none will have been classified. Otherwise, a list of the other classified samples

is created and ordered by difference to the new sample. If no sample is within a threshold distance, it is assumed that the new sample is not sufficiently similar to any previous classification, and so the user is prompted for a new classification for this sample. Otherwise, the closest k samples are taken from this list and the new sample is assigned the same classification as the majority of these k samples. An appropriate value of k can be found through trial and error during initial investigations.

For the final conversion of the classifications to a style sheet, an arbitrary sample from each classification is used to supply the definition of the style, and the name assigned to the classification is used as the name of the style. As each sample in the class should be very similar, it should not matter which sample is used for the style definition.

A slight modification was made to the ks-NN class so that it could be used to create an example document from an existing signature style. This modification was that a new author signature is not created if no close match among the previously classified samples is found i.e. if multi authorship style exists in the document. The contents of the style sheet are read in and set as the classified samples to provide the classification.

The same approach is used for finding groups of pages with the same style. The major differences in this case is that the methods used to represent each page, and the differences between them—as well as the automatic naming procedure of a process which is to all intents and purposes completely unsupervised.

Each page, or paragraph, is represented by a set of feature information, including a list of the number of times each one is used, and the distribution of the feature tags throughout the page or paragraph. The combination of this set of information gives a good overall impression of the written signature style of the author.

The difference between two sets of information, 'unknown sample' and existing ASS, is found by the number of features, and values, that are not present in one set of information and is present in the other. The table distributions are compared using the chi-squared test. Each distribution is composed of 100 values, indicating the number of signature tags in that 100th of the section. The chi-squared value is calculated as the sum of the squares of the differences of each of these values, as given by the following formula [15]:

$$\chi^2 = \sum_{i=1}^{100} \frac{(x_i - y_i)^2}{y_i},$$

where x is the distribution of table tags in Sect. 1 and

y is the distribution of table tags in Sect. 2

The set of this information provides an overall value for the difference between the two pages, or paragraphs. This can then be directly compared to the value for any other pages. Again, if the page. or paragraph, being classified is not sufficiently similar to any previously classified section, a new classification, or ASS, is created for it.

The following section describes investigations which were carried out, using the new algorithm, to determine the effectiveness of the CSS methods to facilitate comparison of author signature styles (ASS) in the paragraphs comprising the students.

7 Investigations

In order to determine the effectiveness of the approach used, a set of metrics were defined which enabled the effectiveness of the solution to be determined on a wide range of submitted documents. This section describes the metrics used and the wide range of documents used.

7.1 Measures of Effectiveness: Metrics Used

The effectiveness of the solution was determined by the ease, and effectiveness, of extraction of file information from the source page into a separate author signature style sheet and the degree to which the content of the original pages remained unaltered once it has been produced by use of the style sheet.

The metrics of: Relative time for comparison of author signature styles in paragraphs contained within student submission by the new algorithm with that taken by a human carrying out the same task. Number of author signature styles produced and number of differences between the signature style features in the original page, or paragraph, in the submission and that created using the ASS were also used to determine the effectiveness of the solution.

The following sections describe in detail the metrics and provides a justification for their use.

7.1.1 Metric 1 Relative Time for Paragraph Authorship Comparisons

This test investigates whether the authorship of all paragraphs, and pages, contained within the submission to be carried out more rapidly by the algorithm than by a human. This is tested by calculating an index which provides a value representing the relative times to carry out authorship comparison of paragraphs by hand and by the new algorithm.

This metric test is of importance because the algorithm would not be useful if it were not possible to identify possible multi-author submission within a timescale in which university procedures could be instigated, and completed, during an academic year.

Table 1 Examples of submission types

Sample	Student origin	First language	Number	Authorship
1	UK	English	100	Known single
1	UK	English	100	Assumed single
1	EU	–	100	Assumed single
1	Not EU	Not English	100	Assumed single
1	UK	English	20	Known multi

7.1.2 Metric 2 A Count of the Author Signature Styles Produced

Sections of document paragraphs which are slightly different could potentially be converted to the same ASS style, because the data mining approach used allows for some fuzziness in the classification in line with author styles varying slightly within the paragraphs of a document. However, document paragraphs which vary greater than observed with one author should result in different ASS styles. This should be indicated by the number of styles produced. Therefore the number of styles produced is also an important measure of how easy it will be to determine commonality in author signature style within paragraphs contained within a document.

7.1.3 Metric 3 Information Differences

In addition to improving speed of author signature comparisons between pages, and paragraphs, of a submission the key author signature features pages created by the system, using the appropriate ASS, should be identical to those contained within the original submission. This is tested by measuring the number of differences between the original and newly produced pages, assigning a score to each type of difference, and adding these scores together.

It is important that the submitted, and re-created, document paragraphs do not exhibit any differences because the ASSs of each student, created from their first submission, will be compared with future submissions. Any differences present in this metric would question the validity of the tests, for multi-authorship, of future submissions by each student.

7.2 Documents Investigated

Table 1 provides examples of the wide range of student submissions which were investigated.

These submissions were chosen as examples of their wide range of document pages to which the new algorithms can be applied because they represent a cross section of the variation in author styles contained with documents submitted at this university.

Sample 1 containing documents known to have been written by one author. Sample 2 contains UK students whose first language is English whilst Sample 3 contains those from EU. Sample 4 contains non EU students who are required to take TOEFL and who have all passed the level required to be admitted to the university. Sample 5 contains documents which are known to contain multiple authorship. The sample size reflects all those available in this category in the year in which the investigations were carried out. A decision was made to only make use of samples of each type available in the same year to ensure that external, unknown, factors which could impact upon the submissions were the same.

This range of documents should enable the performances of the new algorithm on different written styles of pages to be determined.

The following section describes the results from applying the metrics to the wide range of test documents.

8 Results

Simple plots were used to visualize the results. Figures 4, 5, 6 show the results of investigation of the three metrics.

8.1 Relative Time for Paragraph Authorship Comparisons

The results of these investigations are shown in Fig. 4. The figure shows that the new algorithm was able to perform comparison up to 1000 times faster than the person carrying out the same task. The figure also shows that the time taken for the non-algorithm based comparison of the signature style in each paragraph varied from person to person and also from sample type to sample type. There was no difference in the comparison time for the new algorithm because of the short time in which this was achieved, all within 1 s.

Half of the people carrying out the task were unable to complete the comparison for any of sample size 4 because of the fluency in the written style of the student submissions.

8.2 A Count of the Author Signature Styles Produced

The number of signature styles produced is dependent upon the written content of each page.

Figure 5 shows that, on average, two styles are produced from a page known to have been written by one author. The figure also shows that, on average, two styles are produced from a page of unknown authorship, irrespective of language background.

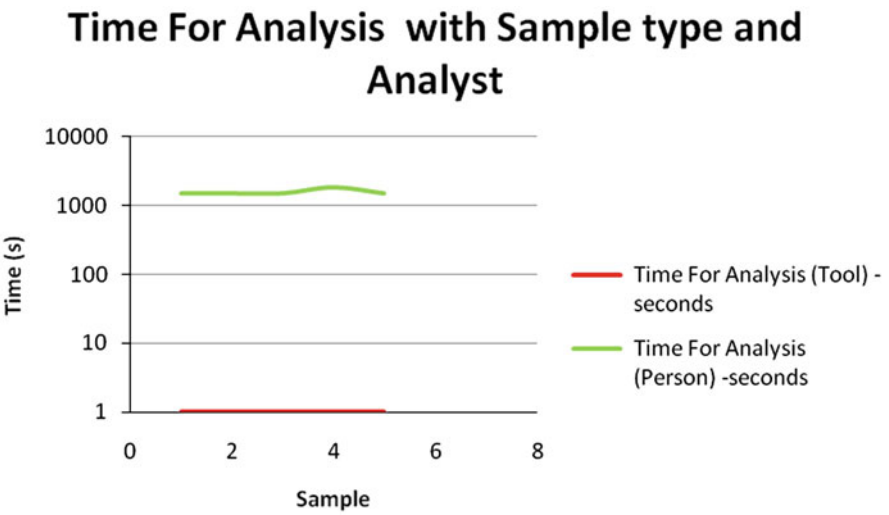


Fig. 4 Relative time for paragraph authorship comparisons

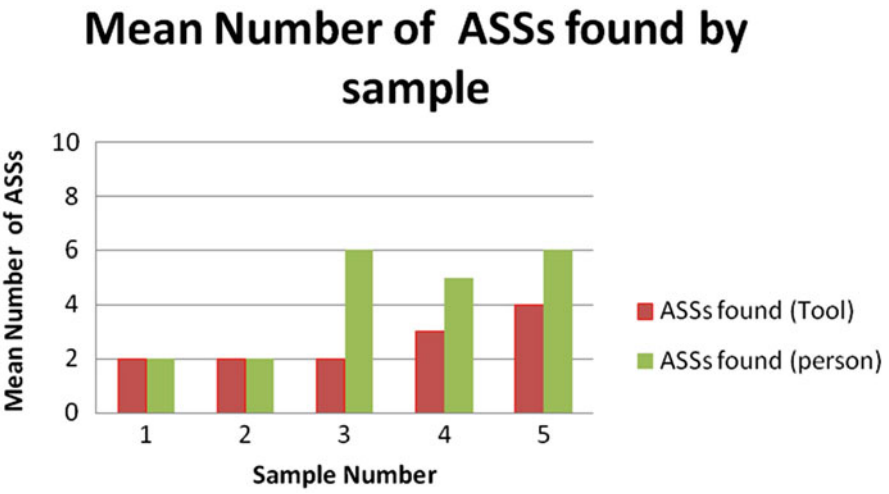


Fig. 5 A count of the author signature styles produced

The algorithm identifies the presence of more authors, four, in the known multi-author submissions. However, the figure shows that human determination was less accurate—especially for samples 3 and 4, those for which English was not a first language.

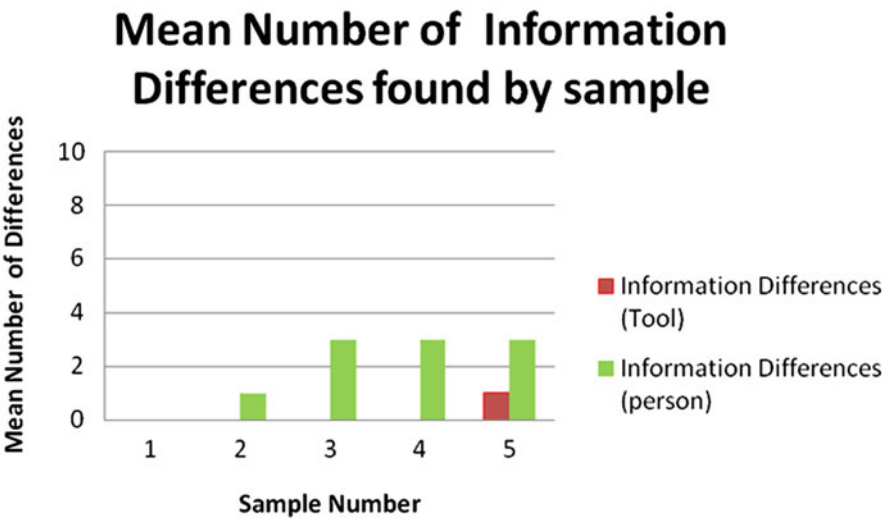


Fig. 6 Information differences between original and reformed text

8.3 Information Differences

These results shown in Fig. 6 are consistent with the results of the ASS investigations in that information differences observed between the original, and key features of the, document are strongly correlated with the error in determining authorship number.

Thus suggesting that if the ASSs contained in the document can be determined then it is possible to reform key features of the original document for comparison with other student submissions and with future work by the same student.

9 Conclusions and Future Work

We have described an approach which enables investigation of the plurality of the authorship of documents submitted by students. This approach makes use of Data Mining based clustering methods and the assumption that plagiarism can occur when a document contains written material from more than one author.

As a by-product of the production of the author signature styles, during the investigation of the first submission, it is possible to compare future submissions with the signature styles shown in their initial submissions and hence, by identifying a change in submission style, potential plagiarism.

The results presented in Sect. 8 show that the approach used facilitates accurate investigation of the authorship of student document submission. Such results have the potential to be used in formal university procedures for those students suspected

of submitted plagiarized material. This is especially so because of the accuracy in production of reformed documents/paragraphs using the ASSs.

It is intended that further work will be carried out investigating the three key metrics in submission from other Faculties and universities in the UK. Work will also be carried out to modify the Data mining algorithm to maintain accuracy of Multi Author determination across this new range of submissions.

Acknowledgements Acknowledgement is made to Mark Carrington for his original project work in 2002 which led to development of this work and article.

References

1. cs.stanford.edu/~aiken/moss/. Accessed 10 Feb 2012
2. Turnitin UK. www.submit.ac.uk/. Accessed 10 Feb 2012
3. http://www.alluniversities.com/index.php. Accessed 10 Feb 2012
4. http://www.articlesnatch.com/Article/Uk-Academic-Writing-Service/1239456. Accessed 10 Feb 2012
5. http://www.brunel.ac.uk. Accessed 10 Feb 2012
6. http://www.hefce.ac.uk. Accessed 10 Feb 2012
7. http://www.ithenticate.com/press-releases/leading-organizational-plagiarism-checker-reports-record-growth. Accessed 10 Feb 2012
8. www.plagiarism.phys.virginia.edu/software.html. Accessed 10 Feb 2012
9. www.scanmyessay.com. Accessed 10 Feb 2012
10. Brink, A., Schomaker, L., Bulacu, M.: Towards explainable writer verification and identification using vantage writers. In: Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007, vol. II, pp. 824–828. IEEE Computer Society, Washington DC (2007)
11. Burn-Thornton, K.E., Bradshaw, J.: Mining the organic compound jungle—A functional programming approach. In: Bramer, M.A. (ed.) Knowledge Discovery and Data Mining. IEEE, Washington DC (1999)
12. Burn-Thornton, K.E., Edenbrandt, L.: Myocardial infarction—Pinpointing the key indicators in the 12 lead ECG using data mining. J. Comput. Biomed. Res. **31**, 293–303 (1998)
13. Burn-Thornton, K.E., Cattrall, D.M., Simpson, A.: A novel data mining tool for ATM networks. In: Proceedings of the 1st International Conference on Practical Aspects of Knowledge Management (PAKM'96), vol. 1, Basel, 30–31 Oct 1996
14. Burn-Thornton, K.E., Thorpe, S.I., Attenborough, J.: A Method for Determining Minimum Data Set Size Required for Accurate Domain Analysis, pp. 161–169, PADD'00, International Data Mining Conference, Manchester-ISBN I 902426088 (2000)
15. Burn-Thornton, K.E., Carrington, M., Burman, T.: A data mining based method for web site maintenance. Intell. Data Anal. **10**(6), 555–581 (2006)
16. Cai, J., Paige, R., Tarjan, R.: More efficient bottom-up multi-pattern matching in trees. Theor. Comput. Sci. **106**, 21–60 (1992)
17. Chaski, C.: Multilingual forensic author identification through N-Gram analysis. Paper presented at the annual meeting of The Law and Society Association, Berlin, 25 Jul 2007
18. Ghani, R., Jones, R., Mladenic, D., Nigam, K., Slattery, S.: Data mining on symbolic knowledge extracted from the web. In: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD-2000), Workshop on Text Mining, Boston, 20–23 Aug 2000
19. Hewitt, J., Brett, C.: The relationship between class size and online activity, and patterns in asynchronous computer conferencing environments. Comput. Educ. **49**, 1258–1271 (2007)

20. Kövesi, B., Boucher, J.M., Saoudi, S.: Stochastic K-means algorithm for vector quantization. Pattern Recognit. Lett. **22**, 603–610 (2001)
21. Michie, D., Spiegelhalter, D.J., Taylor, C.C. (eds.): Machine Learning, Neural and Statistical Classification. Ellis Horwood, Chichester (1994)
22. Shadbolt, N.: Caught up in the web IEEE Intelligent systems (2001)
23. Siddiqi, I., Vincent, N.: Writer identification in handwritten documents. In: Proceedings of the 9th International Conference on Document Analysis and Recognition, vol. 1, Curitiba, 23–26 Sept 2007
24. Simpson, S.: http://www.comp.lancs.ac.ucomputing/users/ss/websitemgmt. Accessed 10 Feb 2012
25. Sommerville, I.: Software Engineering, 5th edn. Addison-Wesley, Boston (1996)
26. Wilde, E.: Wilde's WWW: Technical Foundations of the World Wide Web. Springer, Berlin (1999)

Data Mining Based Tax Audit Selection: A Case Study of a Pilot Project at the Minnesota Department of Revenue

Kuo-Wei Hsu, Nishith Pathak, Jaideep Srivastava, Greg Tschida and Eric Bjorklund

Abstract We present a case study of a pilot project that was developed to evaluate the use of data mining in audit selection for the Minnesota Department of Revenue (DOR). The Internal Revenue Service (IRS) estimated the gap between revenue owed and revenue collected for 2001 to be approximately $345 billion, of which they were able to recover only $55 billion, and the estimated gap for 2006 was approximately $450 billion, of which the IRS was able to recover only $65 billion. It is critical for the government to reduce the gap and the fundamental process for doing so is audit selection. We present a data mining based approach that was used to improve the audit selection process at the DOR. We describe the manual audit selection process used at the time of the pilot project for Sales and Use taxes, discuss the data from various sources, address issues regarding feature selection, and explain the data mining techniques used. Results from the pilot project revealed that the data mining based approach can increase efficiency in the audit selection process. We also report results from actual field audits performed by auditors at the DOR, and results validated the usefulness of the data mining based approach for audit selection. The impact of the pilot project would be a refinement of the manual audit selection process and tax assessment procedures for other types of taxes.

K.-W. Hsu (✉)
Department of Computer Science, National Chengchi University, Taipei, Taiwan (ROC)
e-mail: hsu@cs.nccu.edu.tw

N. Pathak · J. Srivastava
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
e-mail: npathak@cs.umn.edu

J. Srivastava
e-mail: srivasta@cs.umn.edu

G. Tschida
Department of Revenue, State of Minnesota, St. Paul, MN, USA
e-mail: greg.tschida@state.mn.us

E. Bjorklund
Computer Sciences Corporation, Falls Church, VA, USA
e-mail: ebjorklu@csc.com

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_12

1 Introduction

The Internal Revenue Service (IRS) uses the concept of *tax gap* to estimate the amount of non-compliance with tax laws.[1] The tax gap measures the difference between the amount of tax that taxpayers should pay and the amount that taxpayers actually pay on time. The estimated tax gap is stable over time, ranging between 16 and 20 % of tax liability [29]. For Tax Year 2001, the estimated tax gap was roughly $345 billion and only a small portion was eventually collected. The IRS recovered roughly $55 billion, 15.9 % of the total tax gap, reducing it to $290 billion for Tax Year 2001. For Tax Year 2006, the estimated tax gap was about $450 billion, while the net tax gap (which could not be collected through enforcement or late payments) was about $385 billion. Tax evasion is problem that all modern economics have to face and solve [34]. As former IRS Commissioner Mark W. Everson indicates,[2]

> "*The magnitude of this tax gap highlights the critical role of enforcement in keeping our system of tax administration healthy.*"

Also as Andreoni et al. indicates [1],

> "*Characterizing and explaining the observed patterns of tax noncompliance, and ultimately finding ways to reduce it, are of obvious importance to nations around the world.*"

Improving government efficiency is important for effective governance, while improving tax assessment efficiency is essential for economic activities. This is especially critical in a tough economy.

Since tax is the primary source of revenue for the government, it is imperative for the government to reduce the tax gap. The first step in doing so is to understand the sources of this tax gap. These include non-filing of tax returns, underreporting of tax, and underpayment of tax. Underreporting of tax is the single largest factor, and the amount underreported is much larger than the sum of non-filing and underpayment. As Toder indicates [29],

> "*Underreporting of tax liability is a much bigger source of the tax gap ($285 billion) than either underpayment ($33 billion) or non-filling ($27 billion).*"

Thus, underreporting of taxes is an important challenge presented to the government; however discovering these cases requires considerable effort and work from multiple departments. The problem for the government is how to efficiently discover individuals or businesses that potentially owed taxes [11]. The data mining community has made contributions towards solving this problem. Related work includes using artificial neural network (ANN) to determine if an audit case requires further audit [35], using classification techniques to assist in strategies for audit planning [4, 5], and using machine learning and statistical methods to identify high-income individuals taking advantage of abusive tax shelters [14].

[1] http://www.irs.gov/newsroom/article/0,,id=158619,00.html
[2] http://www.irs.gov/newsroom/article/0,,id=154496,00.html

Most research models underreporting of tax as fraudulent behavior and discovery of unreported taxes as a fraud detection problem. Business or finance related fraud detection problems have been receiving a lot of attention from the data mining community [3, 5, 24, 25, 31, 32]. In the pilot project presented in this article, in contrast, we focused on a different problem, viz. audit selection. An audit is an exploration into records of a taxpayer with the goal of finding out if the taxpayer properly reported tax liabilities, and the goal of an audit selection strategy is not to track down all tax evaders but to use the resources in a more efficient way in order to help the government get better returns on investments [19].

In particular, in the pilot project, we focused on a specific form of tax called Use tax. Use tax is similar to but differs from Sales tax in that Use tax is on the use of taxable goods and services [13]. However, the data mining based approach presented in this article also helped audit selection for Sales tax, since audits for Sales and Use taxes are usually conducted together. Cornia indicated that Sales tax could contribute to 30 % of the annual revenue of states that have employed Sales tax (and most states have done so), and also that the estimated loss in Use tax revenue could be more than $55 billion by 2011 [13]. While the audit selection process for Sales and Use taxes is an important part of tax administration, so far (to the best of our knowledge) limited research in the areas of data mining applications has addressed it.

Bots and Lohman discussed the added value of data mining in a tax collection agency [6]. Gupta and Nagadevara studied the application of data mining to audit selection strategy and provided a review of some published research articles [19]. Cleary used data mining to help select better targeting cases for audits and further to reduce costs of audits [12]. What distinguishes the work we have done for the pilot project from the work done by others is that we focused on Sales and Use taxes in the USA, and that we not only used data from actual field audits for model training but also reported results from the deployment of the data mining based approach in a real audit project.

Figure 1 illustrates the big picture for the problem addressed in the pilot project: The y-axis is the ratio of the average benefit (or revenue) obtained from an audit case to the average cost of an audit; the x-axis is the number of audits that are ranked according to some criteria, such as the business size. Large businesses are usually selected because they are few in number and typically have higher B/C (Benefit/Cost) values, therefore resulting in higher potential auditing revenue. In contrast, small businesses are selected more or less at random. Improvement of audit selection can be obtained by 1) getting higher B/C values (i.e. creating a lift) in the tail area of this graph, or 2) extending the curve with the same B/C values (i.e. making an extension) in the tail area. In the pilot project, we adopted the first method. That is, we aimed at increasing B/C values for (relatively) small businesses.

In the pilot project, data mining was used to analyze a collection of candidate cases filtered from the database and to identify a smaller set of more profitable candidate cases. The identified candidates would be good cases for field audit. In the pilot project, this was posed as a classification problem, and we used supervised learning techniques with historical data for training.

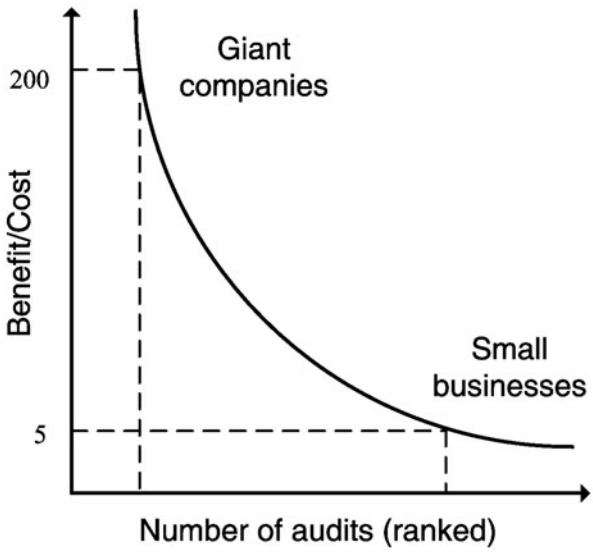


Fig. 1 The big picture for the audit selection problem

Focusing on a particular tax type and a specific group of taxpayers, in this article we describe the approach used in the pilot project, report validation results, and analyze tax audit data and results collected during a certain period of time. The tax audit data reflects taxpayer behavior and poses certain challenges for data mining. This data contains latent subgroups and suffers from the imperfect nature of real-world data, such as missing values and noise. Models trained on this data are tested using data from actual field audits conducted during a subsequent period of time. It must be noted that the approach used has been validated by the DOR, based on actual field audits performed by auditors (which means that for interested cases, auditors reviewed their business and tax records, visited their business locations, and determined their compliance with the tax laws). The data mining based approach has been used in a real audit project. Thus, this paper presents a unique and valuable case study of a pilot project for mining tax data.

The rest of this paper is organized as follows: Section 2 briefly introduces domain knowledge and background information. Section 3 describes our approach to use data mining in audit selection, and Sect. 4 reports the results for data mining on real-world data. Section 5 reports validation results from actual field audits and Sect. 6 concludes this paper with a discussion on the impact of the pilot project.

2 Background

The DOR is responsible for executing and enforcing the tax laws defined by the legislative process. Enforcement of the tax laws is one key piece of this process. Carrying out audits to identify taxpayers that are furthest from tax compliance is a

central component of this process. The DOR has a limited number of resources to allocate to this process, thus it is of interest to find better ways to identify taxpayers furthest from compliance and therefore allocate resources more efficiently. There are opportunities within the DOR to improve the efficiency of compliance activities and subsequently increase revenue. Data mining can identify these, as demonstrated in the pilot project presented in this article. In looking at the existing compliance efforts, there are essentially two avenues for improving efficiency and increasing revenue: Cost savings and revenue collection.

Cost savings could be understood through the following example. One of the most successful audit projects in the end did not generate sufficient revenue for the DOR. That is, the work on the project returned less than what the DOR invested into the project. If successful in reducing costs for the project, data mining has an even greater potential to reduce costs for audit projects that generate a higher number of unsuccessful audits. Moreover, revenue collection is another essential way to improve efficiency and increase revenue. For every given audit executed, there is potentially a better audit candidate that is not being audited that could provide a higher return for the DOR than the audit this is being performed.

The major issue for audit selection is that of effectively selecting audit cases from a pool of candidates, such that the selected cases will result in substantial revenue gains. Audit selection is the very first step in any tax audit project (no matter whether data mining is used or not). Improving the efficiency of audit selection is a key strategic priority to drive government revenue growth. The more frequently the tax collection agency selects potentially profitable cases for audits, fewer unsuccessful audits could be expected, more cost savings will be achieved, and more revenue will be brought into the government. Audit selection is thus important for all audit projects and requires intensive efforts as well as knowledge from experts.

It is impractical to audit all taxpayers with the limited time and resources provided. Moreover, there is always a cost associated with an audit and the generated revenue might not cover it. Due to these factors, in any audit project experts evaluate certain audit cases and determine taxpayers who are at potential risk for underreporting or underpaying taxes. The final results (i.e. the revenue generated by audit cases) are highly dependent on the quality of the pool of selected taxpayers or audit cases. Although there is a systematic selection approach, it serves more as a guideline and audit cases are generally evaluated by experts based on their experience. Nevertheless, there is room for improvement and data mining has potential to improve the audit selection process.

At the time of the pilot project, the process for audit selection was human-intensive and depended heavily on the experience of experts.[3] To begin, rules derived from tax research were used to filter out several thousand candidate cases from a database

[3] Some workflows have been changed since the pilot project was completed, and part of the data mining based approach (presented in this article) has been changed because of the introduction of the comprehensive Integrated Tax system, the advances in the analytic capability, and the amount of data available. Nevertheless, the objective of this paper is to share our experiences of using data mining to improve audit selection for the DOR.

or data warehouse (the first stage). This list of candidate cases was then refined and several hundred were selected for field audits (the second stage). In the refinement stage, experts evaluated candidate cases based on pre-specified rules, but evaluation was mostly based on their experience and expertise. If audit cases were well selected in the previous two stages, there was a higher likelihood of generating cost savings and additional revenue. If an audit case turned out to be unsuccessful (i.e. to generate revenue less than the efforts associated with the audit process), the cost was not only time but also the loss of an opportunity to work on a successful case. The goal of the pilot project was to use data mining to improve audit selection, particularly the second stage. As for the first stage, where an initial pool of candidates was generated, the audit selection criteria and process are confidential.

Data mining is a solution that enables increased efficiency and revenue collection and is applicable to most or all of compliance activities. If we view the data mining based approach as a functional component in the whole audit selection process, its input is the initial pool of candidate cases prepared by experts and its output is a collection of labels, each of which corresponds to a case and is with one of the two possible values *Good* or *Bad*. Those labeled as *Good* would be cases for field audit. Nevertheless, data mining is not the only solution to improve audit selection at the DOR.

In the rest of this section, we first introduce some types of taxes, especially Sales and Use taxes. Then, for each tax type we briefly describe the special part in its audit process (used at the time of the pilot project). Given that each tax type has unique characteristics and varying degrees of reliance on data sources other than its own tax return, the audit process varies across divisions at the DOR. All divisions at the DOR are motivated to increase efficiency throughout the audit process, so the approach developed in the pilot project would help the DOR develop the data mining audit processes for other types of taxes.

Sales and Use Taxes Nearly all returns of Sales and Use taxes were processed electronically which allowed the DOR to make use of audit selectors as the return was being processed. Experts used a few dozen selectors. There was room to increase this number but not from the return itself. Once the returns were in the tax system, experts requested worksheets listing subsets of taxpayer financial information. They then sorted this list and added other sources as necessary to look up business taxpayers who were furthest from tax compliance. Audit selection process used both pre-defined queries and ad-hoc queries. However, most audit selection work was delegated to each region to extract independently using the data available in spreadsheets and the related system.

Individual Income Tax Part of the Individual Income Tax audit workload was reduced by filtering out non-compliance that could be identified by looking at the returns alone or in comparison with available federal return data as the return was being processed.

Corporate Franchise Tax Processing the Corporate Tax Returns and getting the paper filed returns into electronic entry format was a difficult task. The returns came

in many different formats and it was a challenge for data entry personnel to fit the data into the standard tax return data structure. The electronic data was therefore somewhat suspect at that time.

Partnership, Estate, Fiduciary, S-Corporation (PEFS) Tax Their filings were entered into the corresponding tax system. Some audit selectors called *audit flags* were implemented. Proven queries and investigative queries played a part in audit selection. Federal, Individual, and other data were merged with PEFS data to identify audit candidates. Additional staff and additional data sources could lead to more investigative audits and ultimately better audit selection quality.

In this paper, we present the work of a pilot study that we have done at the DOR, and the focus was on Sales and Use taxes. The DOR defined the problem addressed in the pilot project a typical binary classification problem to keep the pilot study simple. The reason why the DOR used average cost (and revenue) rather than individual costs (and revenues) was also to keep the pilot study simple. The results, as reported later in this article, showed that the classification models built under such simple settings were beneficial in a real audit project.

3 Approach

Data mining has been applied to finance and accounting [23, 39]. Zhang and Zhou pointed out the challenges of applying data mining to finance [39], and those discussed in this section include choosing data mining techniques, integrating multiple data mining techniques, and using heterogeneous and distributed data sources. The approach developed in the pilot project is based on supervised learning (i.e. classification), while Yang et al. proposed an approach based on unsupervised learning (i.e. clustering) to analyze tax returns [37]. Since we have data sets with labels given by experts at the DOR (and labels themselves are informative), it is reasonable to use supervised learning.

Now let us consider the audit process (for Sales and Use taxes) used at the time of the pilot project. In the final pool of field audits, about half of the audits were fixed irrespective of their outcome. These included the largest companies in every state zone[4] that were audited regularly. Also included was a group of audits dedicated to research conducted internally by experts. The other half was handpicked by experts using the audit selection process. These hand-picked audits fell under a general category called APGEN (i.e. *Audit Plan—General*), and typically consisted of cases involving small to medium scale sized businesses. Figure 2 presents the process for audit selection for the APGEN category used at the time of the pilot project. Initially, experts posed a database query in order to select several thousand candidate cases out of all the businesses in the state. The query depended on tax type and other

[4] State zones are divisions of the state, created and used internally by the DOR for efficient work-load balancing.

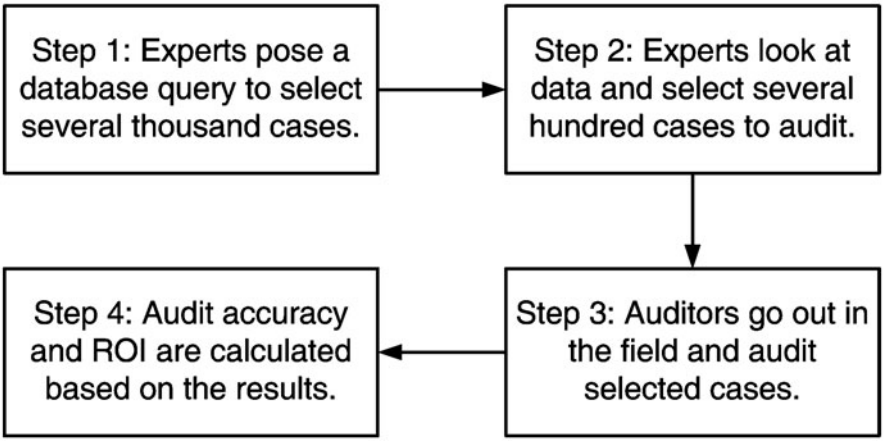


Fig. 2 The manual audit selection process for audit selection used at the time of the pilot project

information. Additionally, experts could potentially refine the query for candidates satisfying certain other criteria (most likely criteria depending on the state of the economy, industries and other situational factors at the time candidates filled tax returns). Next, experts examined some of the candidates in more detail and selected roughly 10 % for further examination and audit. Experts at the DOR have a method to rank cases, but its confidentiality is protected by law. They would probably also select certain cases based upon suspicions, recommendations from other experts, tip-offs, etc. The final selection was based on case by case subjective evaluations by experts at the DOR. Finally, field audits were conducted on those selected cases and success was measured using audit accuracy along with return on investment (ROI, which represents efficiency and whose definition is given later in this article). In other words, the first two steps in Fig. 2 respectively correspond to the two stages of the process for audit selection (at the time of the pilot project), as introduced in Sect. 1. The more *potentially good* audit cases that were selected in the first two stages, the higher the revenue from the audits. If an audit case turned out to be unsuccessful, then it would be a loss in two respects: (1) time and effort put in the audit was wasted, and (2) the resources could have been directed at a successful case, and thus, potential revenue from a successful audit was lost.

In the pilot project, data mining was applied to the audit selection process (used at the time of the pilot project) and was used to select field audits from the pool of several thousand candidate cases resulting from the database query. That is, the focus of the pilot project was on Step 2 in Fig. 2. As previously mentioned, the selection criteria to generate the initial pool is strictly confidential, thus the goal of the pilot project was to analyze the generated pool rather than to generate such a pool of candidates. Nevertheless, the approach developed in the pilot project could help us 'data mining practitioners' and the DOR construct models used to generate the initial candidates. In the pilot project, we focused on Sales and Use taxes and our goal was

to identify candidate cases having higher chances of success of an audit for Sales and Use taxes. The pilot project focused only on audits in the APGEN category (since audits in all other categories were pre-determined), so a binary definition of goodness of an audit was used and defined as the following: Greater than $500 per year during the audit period is *Good*; less than $500 per year during the audit period is *Bad*. Audit period is the number of years in the past (starting from the interested fiscal year) for which tax compliance is checked. In almost all cases the audit period is 3 years. In the pilot project, a Use tax assessment of $1500 was considered a successful audit. This criterion was determined by experts at the DOR. The threshold implied that an audit generating revenue of $1600 would be viewed as successful as an audit generating revenue of $16,000. It was what the DOR used to evaluate the selected audits, however. We used it because we intended to make the data mining based approach compatible with the whole audit selection process used by the DOR (at the time of the pilot project). Moreover, it would be useful to consider the cost of individual audits, since the effort to audit a large corporation is more than that to audit a small corporation. Using an average cost for all audits was also what the DOR used in evaluation (at the time of the pilot project). We used it for the same reason: compatibility.

Taxpayer behavior varies across many diverse factors and as a result, even though the focus of the pilot project was on Use tax (while audits for Sales and Use taxes are usually conducted together); data sources from other tax returns were considered as features. Figure 3 illustrates these data sources, including business registration, income, returns of Sales and Use taxes. Use tax field audits and their results conducted over the last 3 years were used to construct the training and test data. Multiple data sources were used for audit selection. This is because business tax data are complicated and related, with certain data sources having potential information regarding Use tax compliance. The training data set consisted of APGEN Use tax audits and their results for the years 2004–2006. The test data consisted of APGEN Use tax audits conducted in 2007 was used to test or evaluate models built on the training data set, while validation was done by actually conducting field audits on predictions made by models built on 2007 Use tax return data (processed in 2008). These three data sets are listed below:

- Training: 2004–2006
- Test or evaluation: 2007
- Validation: 2007

Please note that the two 2007 data sets are different (and there are no common records between the two 2007 data sets).

For data preparation, we started with cleaning the training data set by removing inadequate cases (i.e. cases with none or, at most, one year of tax return data). These cases were generally new businesses or businesses that did not file tax returns, and they had no values (i.e. nulls) for most of the features necessary and were removed from the training data. The data set originally consisting of 11,083 cases was cut down to 10,943 cases after this step. Experts helped us select an initial list of more than 220 features from the various data sources, as shown in Fig. 3. After iterative cycles of

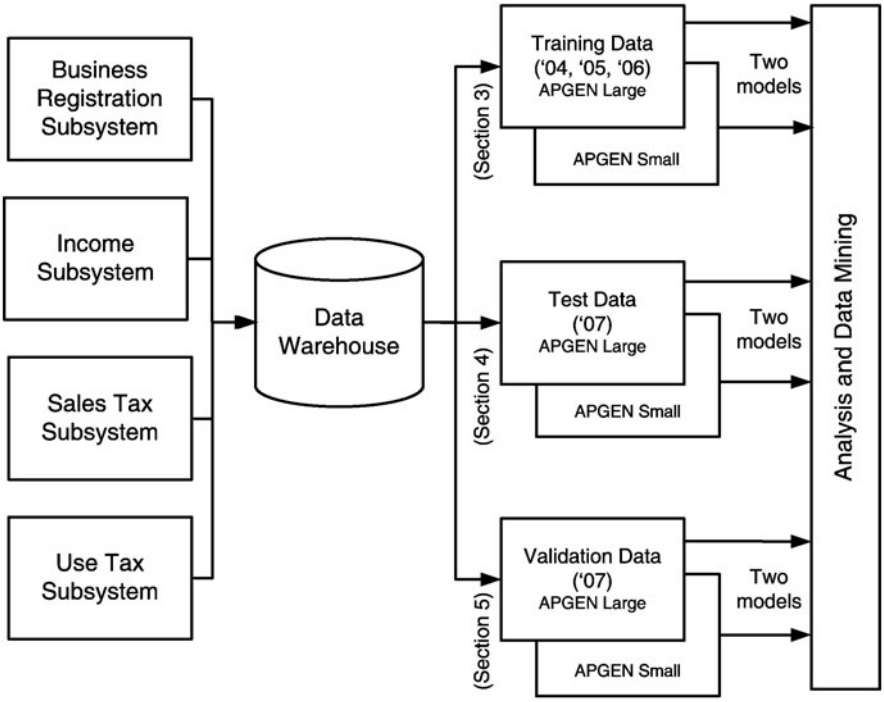


Fig. 3 Data sources for data mining

feature selection and expert consultations a handful of features were selected. These features consisted of two categories: (1) Features related to business characteristics obtained from business registration data, geographic location and type of business, and (2) features correlated with the size of the business, for the three years of the audit period. Nevertheless, details of features that were actually used in the pilot project could not be reported here due to the confidential nature of the tax audit process. Doing so can increase the potential for re-engineering of the audit process, which is clearly undesirable and unlawful.

Initial classification models built using the refined feature set were leaning heavily towards rule-sets that predicted successful audits for larger businesses. During the evaluation stage (using n-fold cross-validation on training data as well as results on test data) it was observed that the initial classification models did well for roughly half the population which consisted of relatively larger businesses, but poorly on the other half which consisted of smaller businesses. The pattern correlating business size with audit success was so dominant that almost all other patterns did not have sufficient relative support to be detected. Therefore, it was decided to divide the original modeling task into two parts: (1) Building one classification model for audit prediction on (relatively) larger businesses (for which the initial classification models seemed to be doing well), and (2) building a second classification model for

(relatively) smaller businesses. We chose to label these two categories as APGEN Large and APGEN Small respectively.

As shown in Fig. 3, we used the same procedure to divide the original three data sets: The training data set was split into an APGEN Large data set and an APGEN Small data set. These two data sets were disjoint. Likewise, the test or evaluation data set was divided into two data sets: one was for APGEN Large and the other was for APGEN Small. The validation data set was also divided into such two data sets.

Businesses in the training set were ranked from largest to smallest, with average annual withholding amount being used as an indicator for business size. The annual withholding amount was directly related to the number of employees and was a very strong indicator of business size. Statistical t-tests were used to determine a withholding amount threshold such that for all businesses below it, there was no significant difference between annual withholding amounts of good and bad audit cases. For cases larger than the threshold value, the business size played an important role in picking good audits (these were mainly the larger businesses in the data set).

The actual withholding amount determined for such a division was determined by using statistical t-tests with various values (to check whether the two sets after a division were really different) and it served as a constant threshold in the pilot project. Thus, the threshold was used to divide the data set into the APGEN Large and APGEN Small categories. Again, the value of the threshold is confidential.

Next, feature selection was performed for APGEN Large and Small individually and different feature sets were obtained for each. Figure 4 presents the feature selection process. Starting from the original feature set, a working feature set was constructed and used as training data to build classification models. We used C4.5 (a decision tree algorithm) [27], Naïve Bayes, multilayer perceptron (an ANN algorithm), support vector machine (SVM) [10, 16], and some others to build classification models. For a set of features in which we were interested, we averaged performance over all the models (excluding those performing significantly poorly). If the results were sufficiently good (on training data and measured using n-fold cross-validation), evaluating the model with test data was performed next. Here, good results were defined as those achieving reasonably high precision and recall with improved estimated ROI (as defined later in this article). In addition, the feature sets corresponding to good results were expected to be consistent with the knowledge and experience of experts. However, at any point where results were not adequately good, the models and their results along with help from experts were examined to identify and remove inadequate features and/or derive new ones. This process was repeated iteratively in order to best use information embedded in the data. Deriving new features was suitable for this purpose and provided a chance for the classification algorithm to analyze the data from different perspectives.

Please note that what is shown in Fig. 4 is not the process used to train models but that used to select features. The test or evaluation data set was used to test or evaluate the feature sets selected to train models, and the results were fed back to the training process. That is, no cases in the test or evaluation data set were used to train a model.

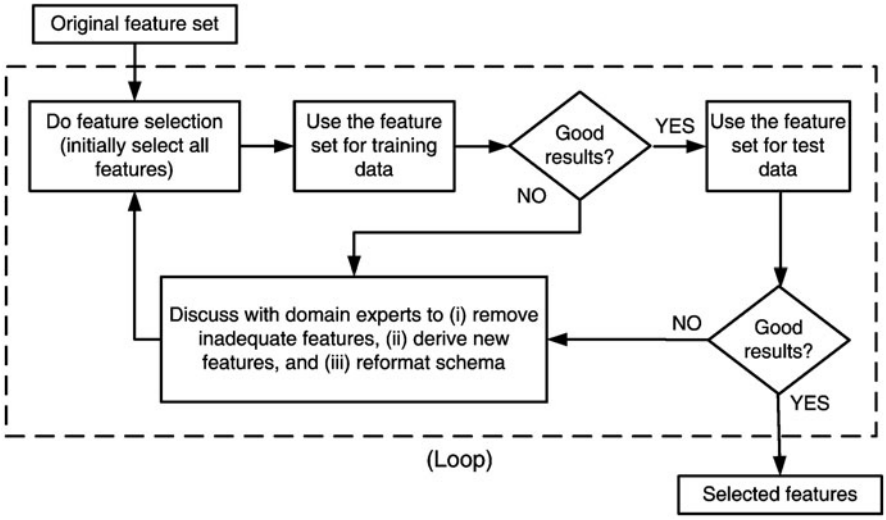


Fig. 4 The feature selection process

We used the same process to select features for APGEN Large and Small, while the set of selected features for APGEN Large was different from that for APGEN Small. Nevertheless, some features were important for both. It is unlawful to reveal the details of the selected features, but we believe that the feature selection process presented in Fig. 4 is valuable and can be used in many other real-world data mining projects.

With the help of experts at the DOR, two special categories of features were recognized for data cleaning and schema reformatting. One special category was composed of features from pre-audit information, which was compiled before auditors performed field audits; the special other category consisted of features from post-audit information, which was collected in or after field audits. However, post-audit information could not be used since it was not available until after auditors performed the field audits. If erroneously used, it could misguide the classification algorithms and created models that succeeded in training (and test or evaluation, probably) but failed in validation or real-world deployment. In fact, any model created using post-audit information was not a predictive model but a descriptive one. While such models were found to be useful to help auditors better understand cases that had already been audited, they were unsuitable for the pilot project's final objective and therefore, not explored further.

The data cleaning process early on filtered out cases that had very little or no data associated with them. The training data set still consisted of quite a few missing values. The data set also consisted of some noise, primarily arising due to errors in reporting from businesses filing their returns or errors due to incorrect data recording from the DOR (especially before the DOR employed the new tax systems). However, this was not a significant issue as the expected fraction of noisy records was very

low. Hence, it was suggested by experts at the DOR to ignore missing values and focus on classification models that were robust towards handling missing values and noisy records.

We reformatted schema (for all data sets) by replacing absolute timestamps with relative timestamps. For example, in the original training data set, three features represented the business statuses of a case in 2004–2006. If a case was audited in 2005, the feature corresponding to its business status in 2006 would have no value. If we intended to use a model in which these three features were referenced to classify a case processed in 2007 (that is, if we intended to *port* the model to a different time frame), we could not find any value in the feature representing the business status of the case in 2004. What is worse, the training data set tracked data only between 2004 and 2006. To solve the problem (or to make the built models *portable*), we used Y, Y-1, and Y-2 for such features. In the first situation given above, there were values for Y and Y-1, and the value for Y-2 was null. In the second situation given above, there were values for Y, Y-1, and Y-2. Then, we could use cases in the training data set even though base values of Y were different. Deligianni and Kotsiantis used a similar way to denote years but for the presentation of the results of forecasting corporate bankruptcy [15].

For modeling, we used WEKA [17, 20], an open source data mining package. Several algorithms had been experimented with and the two with the best performance are reported here. The performance was measured using n-fold cross-validation (on training data and test or evaluation data), and we were looking for (relatively) stable and robust models that would be less prone to overfitting. We intended to have models that would show us the smallest difference between performance on known and performance on unknown data. Having and using such classification models is why and how the data mining based approach performed well in validation where the data set was different from training and test or evaluation data sets. For APGEN Large, MultiBoosting [33] using Naïve Bayes [22, 36] as the base algorithm was used, and for APGEN Small, Naïve Bayes (without MultiBoosting or any other algorithms) was used.

We chose MultiBoosting and Naïve Bayes through experiments, in which we also tried C4.5, multilayer perceptron, SVM, and some others. C4.5 generated interpretable models, but in our experiments it did not demonstrate satisfactory performance. In our experiments, multilayer perceptron and SVM were sensitive to changes in parameters. For both classification algorithms, the best set of parameters obtained on training data was usually different from that obtained on test or evaluation data. Moreover, these two classification algorithms did not perform as well as we expected. In addition, the data sets for APGEN Large and Small might not be sufficiently large for multilayer perceptron and SVM.

Naïve Bayes assumes independence among the features. Despite that this assumption is unrealistic in many situations, classification models built using Naïve Bayes have been successfully used in many real-world applications [28, 36, 38]. Moreover, after we finished the feature selection process, the features we used were less dependent than the original features. MultiBoosting is an ensemble technique that forms a committee (i.e. a group of classification models) to use *group wisdom* to make a

decision. It is different from other ensemble techniques in the sense that it forms a committee of sub-committees (i.e. a group of groups of classification models). Each sub-committee is formed using AdaBoost [18], and wagging (weight aggregation) [2] is used to further combine all these sub-committees into a single committee. It manipulates the given data set to generate different training sets and construct diverse member classification models. The most important characteristic of MultiBoosting is that it exploits the bias reduction capability from AdaBoost as well as the variance reduction capability from wagging. Bias is the average difference between a created model and the underlying model (that generates data and in practice is unknown), while variance represents the average difference among created models. Here the difference arises from using different training sets and it can be measured using error rates. AdaBoost has been shown to have effective bias as well as variance reduction, while it is primarily used for bias reduction. On the other hand, wagging is a variant of bagging [9] (bootstrap aggregation) and is used to reduce variance. Thus, MultiBoosting reduces bias as well as variance [7]. It leverages both techniques and forms a committee that is close to the underlying model and is stable i.e. less variance or more consistent results on unseen data.

Naïve Bayes has high bias and low variance [7, 8]. Thus, it is reasonable to use MultiBoosting with Naïve Bayes (as the base algorithm). Boosting Naïve Bayes was applied to claim fraud diagnosis [31]. Studies showed that MultiBoosting performed well in the prediction of customer choice [30], the prediction of financial distress [26], and the assessment of customer credit quality [21].

4 Evaluation

Classification models were trained or built on tax audit data collected in 2004–2006, while they were tested or evaluated by predicting goodness of audits for 2007 APGEN audit cases. The pilot project used some instead of all such cases because of certain restrictions (mainly those imposed by tax laws). The same withholding amount threshold was used to split the 2007 APGEN data set into (relatively) large and small businesses and the corresponding models were used. The predictions made by models on both large and small businesses were compared to the actual audit results. Figure 5 illustrates the evaluation procedure for data mining based audit selection on the APGEN Large data set for 2007 (which is different from the APGEN Large data set for 2007 used in validation).

APGEN Large for 2007, before the audits[5], experts predicted 878 cases (out of the initial pool) to be good audits, and after the audits, 495 of them turned out to be good audits. On applying the classification model to the same data set and comparing

[5] For evaluation, all cases in the APGEN Large and Small data sets were processed and audited. We simulated the audit case selection process and used them as the ground truth. In the simulation, no actual field audits were conducted for the cases that were not selected by experts or the classification models (while we had all the results).

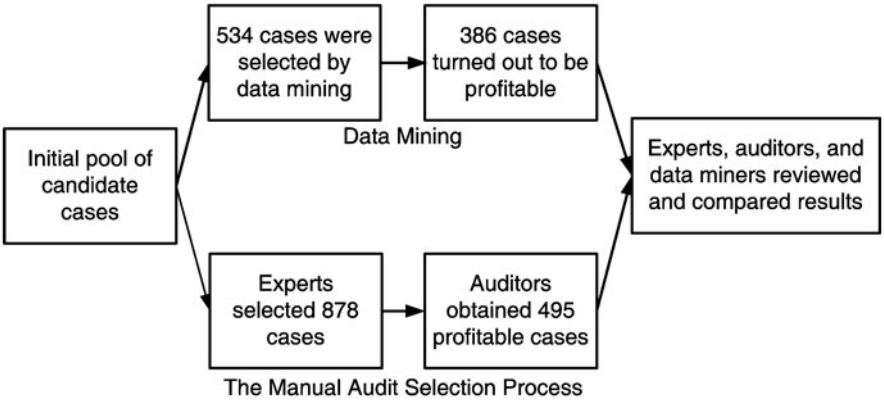


Fig. 5 Evaluation for APGEN Large for 2007 (test or evaluation)

the results to those from actual field audits, it was observed that the classification model predicted 534 cases (out of the initial pool) to be good audits, out of which 386 cases (or 72.3 %) were actually good audits. To sum up, for the same pool of candidate cases for APGEN Large for 2007, experts suggested to audit 878 cases (and 495 of them were good), while the classification model suggested to audit 534 cases (and 386 of them were good). Results were evaluated on the ROI metric. ROI is used as a measure of efficiency, as shown in Eq. 1, which is suggested by experts at the DOR.

$$\text{Efficiency} = \text{ROI} = \frac{\text{Total revenue generated}}{\text{Total collection cost}}. \tag{1}$$

Simplicity and compatibility are the reasons why we did not use ROC (receiver operating characteristic) as the performance measure. ROC was not used by experts at the DOR. Moreover, in some audit projects, individual results were not be available (while only average results were available) and hence ROC was not applicable.

Business analysis is important for any practical data mining application. Cleary and Tax reported their experience in using data mining to help select better targeting cases for audits with little business analysis [12]. In the pilot project, we followed the suggestions given by experts at the DOR to perform business analysis and reported the results as follows.

Table 1 summarizes results from the manual audit selection process at the time of the pilot project, while Table 2 presents results using the data mining based approach. In both tables, the first row (disregarding the header) indicates the number of audits (and corresponding dollar amounts) that were selected by the process. For evaluation purposes the following were used (these were estimates suggested by experts and not the actual numbers): the average number of hours spent conducting a Use tax audit is 23 (h), while the average pay of a tax specialist is $20 per hour. Thus, the collection cost of k audits was $460k. The value seems low, but this was how experts

Table 1 Business analysis for the manual audit selection process used at the time of the pilot project for APGEN Large for 2007 (test or evaluation)

	Good audits	Bad audits	Total
Number of audits	495	383	878
	(56.4 %)	(43.6 %)	(100 %)
Revenue generated	$6,502,724	$170,849	$6,673,573
	(97.4 %)	(2.6 %)	(100 %)
Collection cost	$227,700	$176,180	$403,880
	(56.4 %)	(43.6 %)	(100 %)

Table 2 Business analysis for the classification model created for APGEN Large for 2007 (test or evaluation)

	Good audits	Bad audits	Total
Number of audits	386	148	534
	(72.3 %)	(27.7 %)	(100 %)
Revenue generated	$5,577,431	$72,744	$5,650,175
	(98.7 %)	(1.3 %)	(100 %)
Collection cost	$177,560	$68,080	$245,640
	(72.3 %)	(27.7 %)	(100 %)

at the DOR suggested to calculate the cost for the pilot project. In Tables 1 and 2, the second and the third rows report revenue generated and collection cost, respectively.

The ROI value for the manual audit selection process used at the time of the pilot project for APGEN Large is 1652 % and the same for the data mining based approach is 2300 %. This represents a 39.2 % increase in efficiency. Figure 6 illustrates audit resource deployment efficiency (as does Fig. 8). In Fig. 6, the x-axis and y-axis respectively represent the number of audits performed (i.e. audit effort) and the number of audits that are successful and generate revenue. Theoretically, the best situation is that all audits performed turn out to be profitable. This is captured by the left-most (solid) line. Additionally, the manual audit selection process is represented by the right-most (solid) line. Based on these two lines, the space is divided into three regions, as shown in both figures: the left-most region (A) represents the impossible situation of having more successful audits than the actual number of audits conducted. The right-most region (C) represents the situation where, compared to the manual audit selection process, fewer successful audits are obtained. Any model in this region is less efficient than the manual audit selection process. The data mining based approach is located in the region B. Any solution in region B represents a method better than the audit selection process and is closer to the theoretically best process.

From Fig. 6, the theoretically best process will find 495 successful audits when 495 audits are performed, while the manual audit selection process used at the time of the pilot project will need 878 audits in order to obtain the same number of successful audits. If one projects the (solid) line presenting the data mining based approach, it is observed that in order to obtain 495 successful audits, the number of audits performed will be lower than 878 (which is better than the manual audit

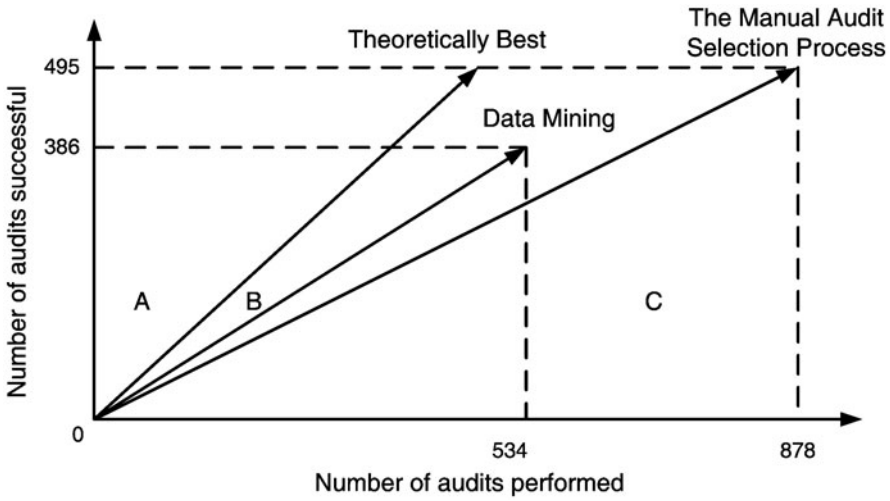


Fig. 6 Audit resource deployment efficiency for APGEN Large for 2007 (for Tables 1 and 2)

Table 3 The confusion matrix for APGEN Large for 2007 (test or evaluation); *R* revenue, *C* collection cost

	Predicted as good	Predicted as bad
Actually good	386 (Use tax collected) R = $5,577,431 (83.6 %) C = $177,560 (44 %)	109 (Use tax lost) R = $925,293 (13.9 %) C = $50,140 (12.4 %)
Actually bad	148 (costs wasted) R = $72,744 (1.1 %) C = $68,080 (16.9 %)	235 (costs saved) R = $98,105 (1.4 %) C = $108,100 (26.7 %)

selection process). It can be estimated as $534 \times 495 \div 386$, which is approximately 685. Alternatively, if the manual audit selection process selects only 534 cases, the number of successful audits will be lower than 386, which is found by the data mining based approach. It can be estimated as $495 \times 534 \div 878$, which is approximately 301. The former number shows that with data mining, less effort is required for the same degree of tax compliance, while the latter number shows that higher tax compliance is achievable for the same effort. Furthermore, Table 3 presents the confusion matrix for the classification model on the APGEN Large data set (for 2007). Columns and rows are for predictions and actual results, respectively. Revenue and collection cost associated with each element is also reported. The top-left element indicates Use tax assessment collected, the top-right element indicates Use tax assessment lost (i.e. cases predicted as bad turning out to be good), the bottom-left element indicates collection costs wasted due to audits incorrectly predicted as good, and the bottom-right element indicates collection costs saved when predicted bad audits are not assessed. Notice that the data mining based approach eliminated cases that

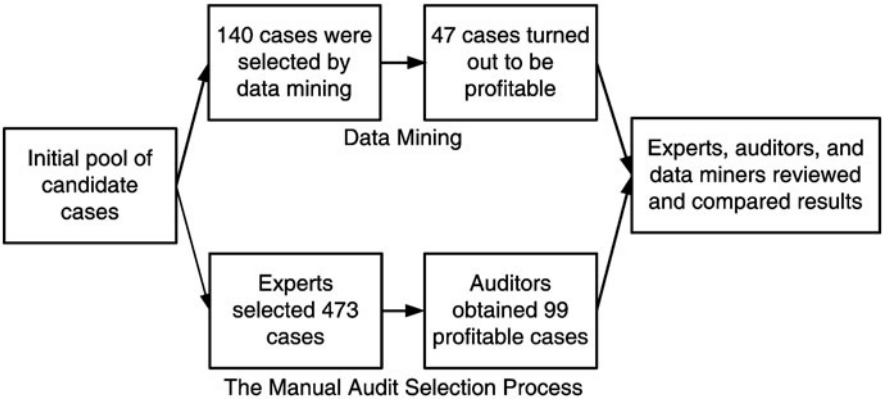


Fig. 7 Evaluation for APGEN Small for 2007 (test or evaluation)

Table 4 Business analysis for the manual audit selection process used at the time of the pilot project for APGEN Small for 2007 (test or evaluation)

	Good audits	Bad audits	Total
Number of audits	99 (20.9 %)	374 (79.1 %)	473 (100 %)
Revenue generated	$527,807 (98.7 %)	$93,259 (1.3 %)	$621,066 (100 %)
Collection cost	$45,540 (20.9 %)	$172,040 (79.1 %)	$217,580 (100 %)

consumed 26.7 % of collection resources but generated only 1.4 % of revenue, thus, significantly improving efficiency.

Figure 7 illustrates the audit selection process used at the time of the pilot project and the data mining based approach for APGEN Small. Here, experts predicted 473 cases as good audits. After conducting field audits, only 99 of them were actually good. For businesses in this subgroup, only one-fifth of cases selected by the audit selection process generated revenue greater than the pre-defined threshold value. In contrast, 47 out of 140 cases (33.6 %) selected by the classification model were truly good audits.

Table 4 reports results for the manual audit selection process used at the time of the pilot project, and Table 5 summarizes results for the data mining based approach. Apart from the increase in precision, the ROI value for the manual audit selection process for APGEN Small is 285 % and the same for the data mining based approach is 447 %, indicating a 57 % increase in efficiency.

Similar to Fig. 6, Fig. 8 illustrates audit resource deployment efficiency. From Fig. 8, when 99 audits are performed the theoretically best process will find 99 profitable audits. However, the manual audit selection process will need 473 audits in order to obtain the same number of successful audits. For using the data mining based approach to obtain 99 successful audits, the number of audits performed will

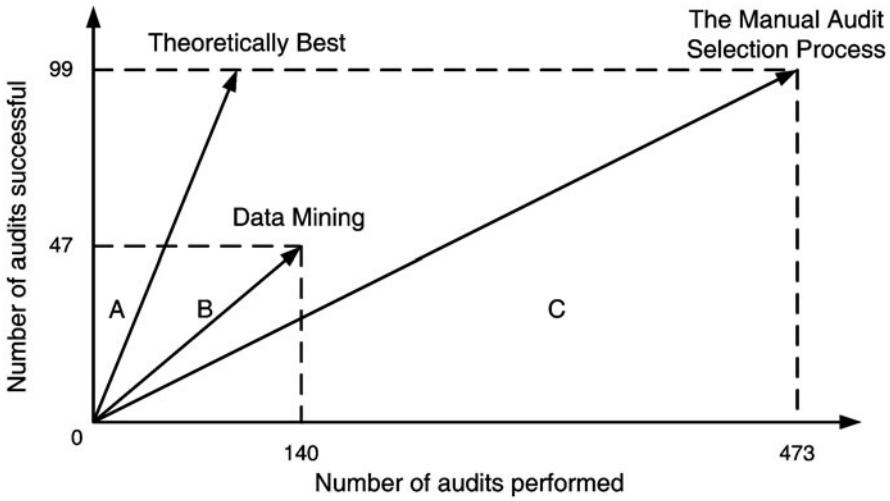


Fig. 8 Audit resource deployment efficiency for APGEN Small for 2007 (for Tables 4 and 5)

Table 5 Business analysis for the classification model created for APGEN Small for 2007 (test or evaluation)

	Good audits	Bad audits	Total
Number of audits	47 (33.6 %)	93 (66.4 %)	140 (100 %)
Revenue generated	$263,706 (91.5 %)	$24,441 (8.5 %)	$288,147 (100 %)
Collection cost	$21,620 (33.6 %)	$42,780 (66.4 %)	$64,400 (100 %)

be lower than 473. It can be estimated as $140 \times 99 \div 47$, which is approximately 295. This number shows that with data mining less effort is required to obtain the same degree of tax compliance. Furthermore, 47 out of 140 cases selected by the data mining based approach turn out to be successful audits. If the manual audit selection process is used to select 140 audits, the number of successful audits will be lower than 47. It can be estimated as $99 \times 140 \div 473$, which is approximately 29. This number shows that with data mining higher tax compliance is achievable for the same effort.

The confusion matrix for the classification model for APGEN Small is presented in Table 6. The 47 good audits correctly identified correspond to cases that consume 9.9 % of collection costs but generate 42.5 % of revenue. Note that the 281 bad audits correctly predicted by the classification model represent notable collection cost savings. These are associated with 59.4 % of collection costs generating only 11.1 % of the revenue.

Table 6 The confusion matrix for APGEN Small for 2007 (test or evaluation); *R* revenue, *C* collection cost

	Predicted as good	Predicted as bad
Actually good	47 (Use tax collected) R = $263,706 (42.5 %) C = $21,620 (9.9 %)	52 (Use tax lost) R = $264,101 (42.5 %) C = $23,920 (11 %)
Actually bad	93 (costs wasted) R = $24,441 (3.9 %) C = $42,780 (19.7 %)	281 (costs saved) R = $68,818 (11.1 %) C = $129,260 (59.4 %)

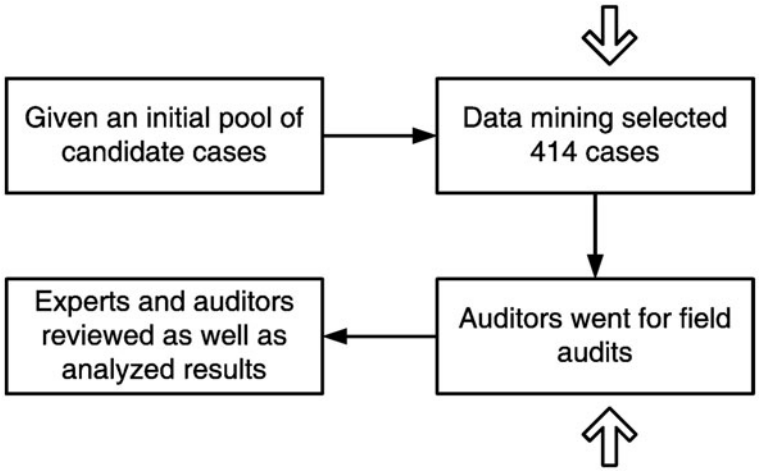


Fig. 9 Validation for the data mining based approach

5 Validation

This section reports results from actual field audits conducted by auditors at the DOR. In order to evaluate the pilot project, the DOR validated the data mining based approach by using them to select cases for actual field audits in a real audit project. The audit selection criteria and process to generate the initial pool of candidate cases are confidential due to DOR regulations. Thus, the data mining based approach was used to select cases from the pool for actual field audits. Figure 9 illustrates the process used to validate the data mining based approach.

The DOR used the classification models to select 414 tax cases for which auditors conducted actual field audits. For the pilot project, the DOR defined a productive audit as an audit resulting in an assessment of at least $500 per year, or $1500 per case (for a 3-year audit period). The classification models were used to analyze the collected tax data and the top 414 most likely predicted good audits were selected. For these selected cases, auditors reviewed their business and tax records, visited

Table 7 Validation results in success rate

	Pre data mining (%)	Data mining predicted (%)	Actual (%)
Sales	29 %	38 %	37 %
Use	39 %	56 %	51 %

Table 8 Validation results

	Pre data mining ($)	Data mining predicted ($)	Actual ($)
Sales	6497	11,976	8186
Use	5019	8623	10,829

their business locations, and determined their compliance with the tax laws. The data mining approach used in the pilot project had been used in a real audit project and therefore the DOR did not carry out the manual audit selection process in parallel. Consequently, there was no direct comparison between the data mining based approach used here and the manual audit selection process. However, as shown in the last step of Fig. 9, audit results from the data mining based approach were reviewed by experts.

The DOR reviewed the results of the actual field audits and compared them to the predicted ones. Table 7 reports results in success rate (i.e. accuracy) while Table 8 reports results in dollars. Both tables present results for Use tax and Sales tax even though the focus of the pilot project was on Use tax (as in earlier discussion). On the one hand, auditors would simultaneously do Use tax and Sales tax when they decided to audit a taxpayer, no matter if the decision is based on their analysis for Use tax or Sales tax for the taxpayer (and no matter whether data mining is used or not). On the other hand, auditors would probably concentrate on Sales tax even though the initial decision was from their analysis of Use tax. Such a decision depended on their experience, the possible collection cost, and the potential profits of these selected audits.

As one can see from Table 7, only 29 % of audits for Sales tax were thought to be profitable by the manual audit selection process (used at the time of the pilot project), named *pre data mining* process, while 38 % of audits were predicted as profitable by the data mining based approach. After auditors performed actual field audits, 37 % of audits turned out to be successful and generated revenue for the DOR. Similarly, 56 % of audits for Use tax were predicted as profitable by the data mining based approach, while only 39 % of audits were thought to be profitable by the manual audit selection process. For Use tax, the actual success rate was 51 %, which was closer to the rate predicted by the data mining based approach. These numbers validated that the data mining based approach had better accuracy and consequently better efficiency.

Table 8 reports revenue in dollars predicted by the manual audit selection process (used at the time of the pilot project) and by the data mining based approach, and it also reports the revenue actually collected by auditors after they performed actual field audits. The *pre data mining average* dollars collected was from historical data (e.g. from auditors' experience) but not from conducting the manual audit selection

Table 9 Validation results of 414 actual filed audits for different categories

	Overall total assessed ($)	Overall average assessed ($)
Large use and sales	1,399,436	19,437
Small use and sales	72,605	2504
Large sales	6,229,248	23,776
Small sales	101,895	1998
Combined totals	7,803,184	18,848

process in parallel. Experts provided qualitative assessment of the cases that were selected by the data mining based approach before auditors performed actual field audits, as the column *data mining predicted* suggests. The results from actual field audits were in the last column. The detailed assessment results are not reported here since they are protected by law. Results in dollars for different categories are reported in Table 9. As described earlier, auditors would concentrate their attention on Use tax, Sales tax, or both once they decided to conduct field audits. Therefore, there are different categories shown in Table 9. These results clearly show that the data mining based approach generated more revenue for the DOR. For example, in Table 9, the DOR assessed Sales tax of $23,776 in average for relatively large businesses. Please recall that the threshold for being a profitable audit was set to $1500 per case (for a 3-year audit period). The result achieved by the data mining based approach clearly proved that it was able to not only save costs and efforts but also generate more revenue. What is more, the manual audit selection process (used at the time of the pilot project) was struggling with relatively small businesses and usually most cases generated less than $1500. Nevertheless, the average amount of assessed Sales and Use taxes achieved by the data mining based approach was $2504. Furthermore, if auditors decided to concentrate on Sales tax, the average assessed amount for relatively large businesses was $23,776 while that for relatively small businesses was $1998. Considering the threshold was set to $1500 per case, and the average assessed amount of dollars was $18,848, revenue generated by the data mining based approach was over 12 times of the threshold that was associated with the average collection cost. These results demonstrated that data mining had the potential to efficiently and effectively perform more sophisticated tax audit selection.

6 Conclusions

We have presented a case study of a pilot project and shared our experiences of using data mining to improve audit selection for the Minnesota Department of Revenue (DOR). Additionally, we have described some practical challenges when applying data mining to audit selection. Improving the efficiency of audit selection and further the productivity of the tax assessment process is an essential component of driving revenue growth for the DOR as well as the government. The audit selection process

is a human-intensive process required of knowledgeable experts. However, apart from being cumbersome, the process is also inefficient. Bad audits not only waste auditors' time and resources but also erode revenue. The approach developed in the pilot project is to use the data mining based approach (i.e. classification models) for the purpose of improving audit selection. Since data play a vital role in any data mining project, considerable attention was paid to data pre-processing, cleaning, and reformatting. Models were trained and tested using real-world data. The results of the pilot project showed that the data mining based approach achieved an increase of 63.1 % in efficiency. The most important part of the pilot project is the validation from actual field audits which demonstrated the usefulness of data mining for improving audit selection in terms of accuracy as well as revenue generated. Improving government efficiency is important for effective governance, while improving tax assessment efficiency is essential for economic activities. This is especially critical in a tough economy. The pilot project provided a further impact of increased interest among the government for effective applications of data mining. The direct impact is a reexamination and refinement of other tax assessment processes that are in use but may be inefficient.

Acknowledgements The research was supported in part by a grant from the Minnesota Department of Revenue. This collaboration would not have been possible without the sponsorship and support of a number of organizations and individuals. Specifically, the authors would like to thank the Minnesota Department of Revenue and the University of Minnesota for providing the institutional support for this work. Finally, we would like to acknowledge the support of the Office of Enterprise Technology, State of Minnesota, and the vision and leadership of Commissioner Gopal Khanna for creating a novel relationship which allows University faculty and students to interact with government departments to mutually collaborate on using advanced technologies to address the State of Minnesota's information technology needs. The opinions expressed in this article, however, are solely those of the authors, and do not represent, directly or by implication, the policies of their respective organizations. The authors would also like to thank anonymous reviewers for their valuable comments and suggestions.

References

1. Andreoni, J., Erard, B., Feinstein, J.: Tax compliance. J. Econ. Lit. **36**(2), 818–860 (1998)
2. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learn. **36**(1), 105–139 (1999)
3. Bhowmik, R.: Detecting auto insurance fraud by data mining techniques. J. Emerg. Trends Comput. Inf. Sci. **2**(4), 156–162 (2011)
4. Bonchi, F., Giannotti, F., Mainetto, G., Pedreschi, D.: A classification-based methodology for planning audit strategies in fraud detection. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, pp. 175–184 (1999)
5. Bonchi, F., Giannotti, F., Mainetto, G., Pedreschi, D.: Using data mining techniques in fiscal fraud detection. In: Proceedings of the 1st International Conference on Data Warehousing and Knowledge Discovery, Florence, Italy, pp. 369–376 (1999)
6. Bots, P.W.G., Lohman, F.A.B.: Estimating the added value of data mining: A study for the Dutch Internal Revenue Service. Int. J. Technol. Policy Manag. **3**(3/4), 380–395 (2003)

7. Brain, D., Webb, G.I.: On the effect of data set size on bias and variance in classification learning. In: Proceedings of the 4th Australian Knowledge Acquisition Workshop, Sydney, Australia, pp. 117–128 (1999)
8. Brain, D., Webb, G.I.: The need for low bias algorithms in classification learning from large data sets. Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, Helsinki, Finland, pp. 62–73 (2002)
9. Breiman, L.: Bagging predictors. Machine Learn. **24**(2), 123–140 (1996)
10. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27 (2011)
11. Chen, Y.S., Cheng, C.H.: A Delphi-based rough sets fusion model for extracting payment rules of vehicle license tax in the government sector. Expert Syst. Appl. **37**(3), 2161–2174 (2010)
12. Cleary, D.: Predictive analytics in the public sector: Using data mining to assist better target selection for audit. Electron. J. e-Gov. **9**(2), 132–140 (2011)
13. Cornia, G.C., Sjoquist, D.L., Walters, L.C.: Sales and use tax simplification and voluntary compliance. Public Budget. Financ. **24**(1), 1–31 (2004)
14. DeBarr, D., Eyler-Walker, Z.: Closing the gap: Automated screening of tax returns to identify egregious tax shelters. ACM SIGKDD Explor. Newslett. **8**(1), 11–16 (2006)
15. Deligianni, D., Kotsiantis, S.B.: Forecasting corporate bankruptcy with an ensemble of classifiers. In: Proceedings of the 7th Hellenic Conference on Artificial Intelligence, pp. 65–72 (2012)
16. EL-Manzalawy, Y., Honavar, V.: WLSVM: Integrating LibSVM into Weka environment. http://www.cs.iastate.edu/∼yasser/wlsvm (2005). Accessed 17 Feb 2012
17. Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: Weka—A machine learning workbench for data mining. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 1269–1277. Springer, Berlin (2010)
18. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, pp. 148–156 (1996)
19. Gupta, M., Nagadevara, V.: Audit selection strategy for improving tax compliance—Application of data mining techniques. In: Agarwal, A., Venkata Ramana, V. (eds.) Foundations of E-government. Computer Society of India, Hyderabad (2007)
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. ACM SIGKDD Explor. Newslett. **8**(1), 10–18 (2009)
21. Huang, S.C., Wu, C.F.: Customer credit quality assessments using data mining methods for banking industries. Afr. J. Bus. Manag. **5**(11), 4438–4445 (2011)
22. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence, pp. 338–345 (1995)
23. Kirkos, E., Manolopoulos, Y.: Data mining in finance and accounting: A review of current research trends. In: Proceedings of the 1st International Conference on Enterprise Systems and Accounting, pp. 63–78 (2004)
24. Kirkosa, E., Spathisb, C., Manolopoulosc, Y.: Data mining techniques for the detection of fraudulent financial statements. Expert Syst. Appl. **32**(4), 995–1003 (2007)
25. Kotsiantis, S., Koumanakos, E., Tzelepis, D., Tampakas, V.: Forecasting fraudulent financial statements using data mining. Int. J. Comput. Intell. **3**(2), 104–110 (2006)
26. Liu, H., Huang, S.: Integrating GA with boosting methods for financial distress predictions. J. Qual. **17**(2), 131–158 (2010)
27. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
28. Rish, I.: An empirical study of the naïve bayes classifier. Tech. rep., IBM. http://researchweb.watson.ibm.com/people/r/rish/papers/RC22230.pdf (2001). Accessed 17 Feb 2012
29. Toder, E.: Reducing the tax gap: The illusion of pain-free deficit reduction. Tech. rep., Tax Policy Center. http://www.taxpolicycenter.org/UploadedPDF/411496_reducing_tax_gap_revised.pdf (2007). Accessed 17 Feb 2012
30. van Wezel, M., Potharst, R.: Improved customer choice predictions using ensemble methods. Eur. J. Oper. Res. **181**(1), 436–452 (2007)

31. Viaene, S., Derrig, R.A., Dedene, G.: A case study of applying boosting Naïve Bayes to claim fraud diagnosis. IEEE Trans. Knowl. Data Eng. **16**(5), 612–620 (2004)
32. Wang, J., Yang, J.G.S.: Data mining techniques for auditing attest function and fraud detection. J. Forensic Invest. Account. **1**(1) (2009). http://www.bus.lsu.edu/accounting/faculty/lcrumbley/jfia/Articles/FullText/2009v1n1a8.pdf
33. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine Learn. **40**(2), 159–196 (2000)
34. Webley, P., Cole, M., Eidjar, O.P.: The prediction of self-reported and hypothetical tax-evasion: Evidence from England, France and Norway. J. Econ. Psychol. **22**(2), 141–155 (2001)
35. Wu, R.C.F.: Integrating neurocomputing and auditing expertise. Manag. Audit. J. **9**(3), 20–26 (1994)
36. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**(1), 1–37 (2008)
37. Yang, Y., Ge, E., Barns, R.: Towards effective and efficient identification of potential tax agent compliance risk: A stratified random sampling approach. e-J. Tax Res. **9**(1), 116–137 (2011)
38. Zhang, H.: The optimality of naïve Bayes. In: Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, Miami Beach, FL, USA (2004)
39. Zhang, D., Zhou, L.: Discovering golden nuggets: Data mining in financial application. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **34**(4), 513–522 (2004)

Part IV
Medical Applications

A Nearest Neighbor Approach to Build a Readable Risk Score for Breast Cancer

Émilien Gauthier, Laurent Brisson, Philippe Lenca and Stéphane Ragusa

Abstract According to the World Health Organization, starting from 2010, cancer has become the leading cause of death worldwide. Prevention of major cancer localizations through a quantified assessment of risk factors is a major concern in order to decrease their impact in our society. Our objective is to test the performances of a modeling method that answers to needs and constraints of end users. In this article, we follow a data mining process to build a reliable assessment tool for primary breast cancer risk. A k-nearest-neighbor algorithm is used to compute a risk score for different profiles from a public database. We empirically show that it is possible to achieve the same performances as logistic regressions with less attributes and a more easily readable model. The process includes the intervention of a domain expert, during an offline step of the process, who helps to select one of the numerous model variations by combining at best, physician expectations and performances. A risk score made of four parameters: *age*, *breast density*, *number of affected first degree relatives* and *breast biopsy*, is chosen. Detection performance measured with the area under the ROC curve is 0.637. A graphical user interface is presented to show how users will interact with this risk score.

É. Gauthier (✉) · S. Ragusa
Statlife company, Institut Gustave Roussy, 114 rue Édouard Vaillant,
94805 Villejuif Cedex, France
e-mail: emilien.gauthier@statlife.fr

L. Brisson · É. Gauthier · P. Lenca
UMR CNRS 6285 Lab-STICC, Institut Telecom, Telecom Bretagne,
Technopôle Brest Iroise CS 83818, 29238 Brest Cedex 3, France

Université Européenne de Bretagne, France
e-mail: laurent.brisson@telecom-bretagne.eu

P. Lenca
e-mail: philippe.lenca@telecom-bretagne.eu

S. Ragusa
e-mail: stephane.ragusa@statlife.fr

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_13

1 Introduction

As cancer is becoming the leading cause of death worldwide, prevention of major types of cancer through a quantified assessment of risk is a major concern in reducing its impact in our society. Physicians have to inform patients about risk factors and have to detect fatal diseases as soon as possible in order to treat them as quickly as possible. Nowadays, this detection is led by prevention programs designed to target highest-risk subsets of the population. For example, women over 50 years old in France and over 40 in USA are recommended to perform a mammography every two years to detect breast cancer; mammography being the primary method for detecting early stage breast cancer which is the most common cause of cancer for women [17]. As a consequence, our society could benefit from a widely used risk score in order to give more accurate counseling on how cancer is impacted by risk factors and to target smallest subset of the population with higher risks. For example, using age at first mammogram as an actionable variable, screenings programs for breast cancer could be extended: younger women with high risk profiles could be offered more frequent screenings in order to decrease death risk [26].

Even if some women may have genetic predisposition for breast cancer, environmental factors have a large impact on the risk according to Lichtenstein [21]. Because of this impact and due to acquisition cost and easyness-to-use constraints, we have decided to focus on environmental factors as attributes to compute a risk for women who never had breast cancer.

As pointed out by Testard-Vaillant [27], "*information, dialog and more patient involvement in the decision-making process*" are key words in dealing with cancer, therefore a major challenge in the field of medical counseling is to provide physicians and radiologists with adequate tools to help them to assess their patients breast cancer risk and to show easily how risk factors impact global risk. For many years, risk scores built upon statistical models were not adopted in medical counseling domain despite their performance. This may be because end-users of these tools are not oncologists nor clinicians and underlying models are too complex and too difficult to use during a medical consultation. Thus, to build a new risk score tool, we need to consider the model readability and the current medical decision process. Moreover, we will have to consider the obligation to use imbalanced datasets with missing data. To the best of our knowledge, no one has been interested in analyzing, with a mining approach, data from women who never had cancer in order to create a risk score with a prevention purpose.

Showing similar cases may improve communication with the patient, therefore increase its involvement in the prevention and decision process. Because core concept of k-nearest-neighbor algorithm is to gather similar profiles using a distance computation, we use it with help of a domain expert in order to build a tool to predict breast cancer risk and measure its performances.

The article is organized in seven sections. Section 2 provides an overview of related works on risk models; Sect. 3 presents our approach of the data mining process we follow; Sect. 4 summarizes needs and constraints of users for the final tool; Sect. 5 describes source data and Sect. 6 reports results, discuss them and present future works.

2 Breast Cancer Risk Scores

2.1 Statistical Approaches

We present studies focusing on prevention and the use of environmental factors such as reproductive and medical history. One major risk prediction model emerges in the statistical field.

Based on an unstratified, unconditional logistic regression analysis, the most commonly used model was developed by Gail et al. [15] using data from the *Breast Cancer Detection Demonstration Program*. Risk factor information was collected during a home interview and the analysis was based on approximately 6000 cases and controls. Among 15 risk factors obtained through patient interviews, only 5 were chosen: age, age at menarche (first natural menstrual period), number of previous breast biopsies, age at first live birth and number of first-degree relatives with breast cancer. Gail's risk score was validated on the population of United States with the *Cancer and Steroid Hormone Study* (CASH) by Costantino et al. [6] and in Italy on the *Florence-EPIC Cohort Study* by Decarli et al. [8]. Chen et al. [5] enhanced the Gail model by modeling the risk with a new equation that includes the breast density. Both regression equation parameters and coefficients are very differents than Gail's ones. It does not facilitate praticians understanding of risk evolution when adding new risk factors as attributes to describe the risk level.

Barlow et al. [3] also built a risk prediction model using a logistic regression on the *Breast Cancer Surveillance Consortium* (BCSC) database (see Table 1 and download data from *http://breastscreening.cancer.gov*) which contains 2.4 millions screenings mammograms and associated self-administered questionnaires (see Sect. 5). Two logistic regression risk models were built with 4 or 10 risk factors depending on the menopausal status. Compared to Gail's model, it gains the use of breast density and hormone therapy. As we will use the same database, it is worth highlighting that reported area under ROC curve (see performance measurement in Sect. 3.4.1) was 0.631 for premenopausal women and 0.624 for postmenopausal women.

Primary goal of these studies was not readability, but rather highest risk detection performances and impact levels of each risk factors.

2.2 Data Mining Approaches and Imbalanced Data

Most similar data mining approaches dealt with slightly imbalanced data, mostly used to predict a cancer relapse as a result of the *Surveillance, Epidemiology and End Results* (SEER) database use. Here, we present two significant related studies involving both medical data and mining algorithm.

Endo et al. [11] implemented common machine learning algorithms to predict survival rate of breast cancer patient. This study is based upon data of the SEER program with high rate of positive examples (18.5 %). Authors did not used ROC curve to assess performances results but accuracy, specificity and sensitivity. Logistic

regression had the highest accuracy, artificial neural network showed the highest specificity and J48 decision trees model had the best sensitivity.

Jerez-Aragonés et al. [19] built a decision support tool for the prognosis of breast cancer relapse. They used similar attributes as Gail (like age, age at menarche or first full time pregnancy, see Sect. 2.1) but also biological tumor descriptors. A method based on tree induction was conceived to select the most relevant prognosis factors. Selected attributes were used to predict relapse with an artificial neural network by computing a Bayes *a posteriori* probability in order to generate a prognosis system based on data from 1035 patients of the oncology service of the Malaga Hospital in Spain.

Such studies show how mining approaches can be used to built classification tools on medical databases while dealing with missing data and business processes. But they do not consider problems (such as readability) encountered by patients who never had cancer nor physicians in their day to day interactions. Moreover, these approaches aim at predicting a class for unlabeled data (e.g. cancer relapse or not) while our goal is to provide a risk level without making the decision (breast cancer or not) in place of the physician.

To build a risk score that helps to detect highest risk profiles among general population, the mining algorithm has to provide a risk value without labeling a woman profile. Dealing with general population means we are facing highly imbalanced data with a breast cancer incidence rate lower than 1000 new cases for 100,000 women. Dealing with such imbalanced data can be done at both algorithmic [20] and data levels [28, 29]. At data level by choosing a different cost or rebalancing positives or negatives examples. At algorithmic level, it is possible to make a k-nearest-neighbor algorithm more sensitive to the minority class by modifying the neighborhood boundaries [20] or by using a class confidence weight [22] to handle imbalanced data during the labeling step.

3 Proposed Process to Build a Risk Score

3.1 Main Objectives

The main objective of our approach is to provide physicians with a tool to assess a cancer risk level for their patient and to promote dialog between them. As statistical models spread with difficulty in the physician community, we aim to find models with good scoring performance and good readability. In our case, we say a model has a good readability if it allows a physician to explain the risk score to his patient:

- it has to be quickly readable by a physician during a medical appointment
- and has to give access to understanding the score.

Furthermore, we have other constraints: physicians have *a priori* ideas about good attributes of a model, patients need actionable attributes to change their lifestyle, both of them want immediately usable score (i.e. very low cost of data acquisition). In addition, a generic algorithm that can be easily adapted to various pathologies is desirable.

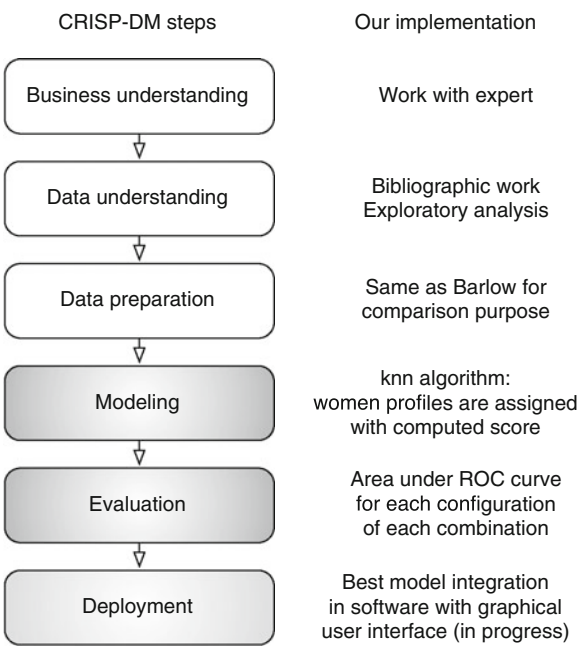


Fig. 1 General process based on CRISP-DM methodology—gray steps identify our major contributions

3.2 General Process

Our approach follows the CRoss Industry Standard Process for Data Mining (CRISP-DM) [4] data-mining methodology. Figure 1 shows the six steps of this process where gray ones identify our major contributions. Business and data understanding steps are not impacted because we want to work on the same data as [3] to be able to compare our results.

3.2.1 Business Understanding

An expert with knowledge of the needs of physicians help us to prioritize our objectives (see Sect. 3.1) and to assess the situation. We decide to focus on a scoring task (no classification or prediction).

3.2.2 Data Understanding

Despite limitations described in Sect. 5, the BCSC database contains most of the known breast cancer personal factors. It is the largest database publicly available that includes breast density information.

3.2.3 Data Preparation

To deal with data imbalance, we can apply rebalancing algorithms on this data but it is not the focus of the article. We do want to minimize modification of data in order to compare our results with Barlow's. The only modification we apply is normalization. It was decided to keep the same split between training and validation set.

3.2.4 Modeling

Several data mining algorithms were considered at first, but domain expert suggested to use a k-nearest-neighbor algorithm because it uses a concept of similarity which is easily understandable by end-users without explaining a complex formula. Moreover, such algorithm is able to deal with imbalanced data if there is enough positive examples among neighbors. We generate models and search for the best combination of attributes by performing an exhaustive search (see Sect. 3.3) on a limited set of combinations. The reason is that the expert issued a recommendation of using a restricted number of factors to make the risk score easy to use. Obviously, for large combinations, computation time can increase sharply, but it is not a problem as models are generated offline only once by us (see Sect. 4.2), when a physician uses the final software, no computation is necessary.

3.2.5 Evaluation

Generated models can be evaluated from a discrimination or a calibration perspective. Discrimination is needed to assess if women with breast cancer from the validation set are given higher scores than women without breast cancer. We use the Receiver Operating Characteristic method using Area Under Curve (AUC) in order to sorts models by scoring performance. Calibration is needed to assess if the number of predicted breast cancer cases is in line with the observed number of breast cancer cases in the validation set. We use the ratio of expected cases number to observed cases number to compare models. Explanation for both evaluation criteria are given in Sect. 3.4.

3.2.6 Deployment

We are currently working to incorporate selected model configuration into a computer software tool for physicians. It will come with a graphical explanation of the concept of nearest neighbor. But it will not embed the database.

3.3 Focus on k-Nearest-Neighbor Implementation

To provide experts with several interesting models, k-nearest-neighbor algorithm (see [7, 14]) is used with various size of attributes combinations (from 1 to 6 attributes), several Minkowski generalized distance measure ($p = 1$–5) and several k values were used (see Sect. 6). Performance for each of hundreds of generated combinations is tested for each values of k.

We implement the k-nearest-neighbor algorithm in two steps:

- *Selection of neighborhood*: For a combination of attributes (e.g. age and breast density), a score value has to be computed for each combination of values (e.g. age = 5 and breast density = 3). To compute such score value, a neighborhood has to be defined for each values combination. To determine if a profile of the database belong to the neighborhood of a combination of values, an Euclidean distance is used to compute the distance between a combination of values and every single record of the training set using a normalized version of the coding values of the BCSC database. Thus, at least k of the nearest records of the database are included in the neighborhood. The neighborhood may not have always the same size because for a given group at the same distance, if k is not reached yet, all neighbors at the same distance are added to the neighborhood.
- *Scoring function*: The score of a combination of values, is the ratio between the number of breast cancer cases (i.e. positive examples) and the size of the neighborhood. In epidemiology, the rate of individuals having a disease in a population is called prevalence. This rate was chosen because it is well known by physicians, easily explainable to a patient and it is directly built on the number of patient diagnosed with breast cancer among patients with a similar profile.

To deal with missing data, we keep the same decision as Barlow, i.e. assign a high value when missing. It will prevent a record with a missing value to be integrated in the neighborhood.

3.4 Focus on Evaluation

Mostly two kinds of evaluation are performed for epidemiological scores: discrimination and calibration. We explain why and how we use them.

3.4.1 Discrimination Using ROC Evaluation

The Receiver Operating Characteristic (ROC) [10] is used to measure discrimination due to the continuous nature of our classifier: performance has to depict how positive instances are assigned with higher scores than negative ones. The ROC curve allows to measure detection performances using a moving threshold to classify examples of the validation set. Moreover, it allows direct comparison with Barlow's results and epidemiological-based scores in general.

Negative examples labeled as positive by the algorithm are called a false positives whereas positive examples labeled as positives are called true positives. The ROC curve is plotted with the false positive rate on the *X* axis and the true positive rate on the *Y* axis [13], both rates being calculated for a given threshold. It can be summarized in one number: the Area Under the ROC Curve (AUC). The area being a portion of the unit square, its value is in the [0,1] interval. The best classifier will have an AUC of 1.0 (i.e. all positive examples are assigned with higher score than negative ones) whereas an AUC of 0.5 is equivalent to random score assignment. The AUC can also be seen as the probability that randomly chosen positive and negative examples will be correctly ranked.

3.4.2 Calibration Using E/O Ratio

The *E*xpected cases number to *O*bserved cases number ratio is used to measure the calibration of a model. Women from the validation set are sorted by scoring value and the validation set is split in ten groups. In each group, the mean score is computed and converted to an expected number of cases. The sum of the ten expected numbers of cases is then compared to the observed number of the validation set using a ratio. The best E/O ratio is 1.0, meaning that the model predict the same number of cancer cases than the actual number of cases.

To help the expert to choose the best model, each k value of each combination of attributes is assigned with an AUC and a E/O ratio value.

4 A Mediation Tool for Physicians and Patients

Providing physicians with a tool to assess a cancer risk level for their patient and promoting dialog between them, we identified constraints that arise from the users needs, we describe a solution and a we show a graphical user interface prototype that fits users needs.

4.1 Users Needs and Impacts on the Tool

As pointed out in the introduction, the risk score is not only used to compute a risk level, but it has to be a way to promote dialog between the patient and the physician. These constraint has two majors impacts on the process that lead to the risk score construction.

First, the risk score has to be readable in how it operates. The basics of the modeling method have to be understandable by both patients and physicians: readability impacts the choice of the algorithm used to compute a score. Need of readability also impacts attributes chosen to characterize a profile. The process to build the risk score

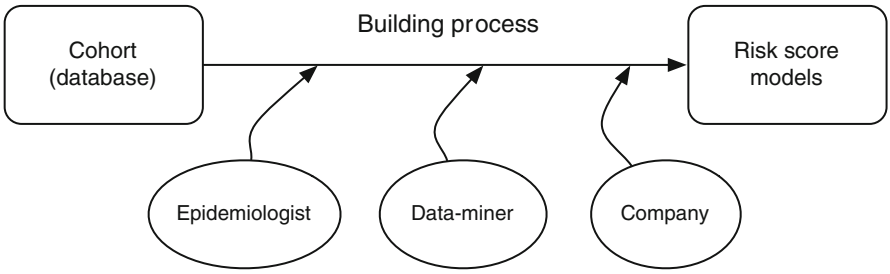


Fig. 2 Offline process with stakeholders intervention

has to allow intervention from domain expert: he will choose the best combination in terms of high risk profile detection, attributes acceptability for end users and capacity to promote dialog between patient and physician, using attributes actionability for example.

Second, the risk score has to be provided in real time. To promote dialog and allow quick appropriation by users, the risk score has to be displayed instantly on a computer screen. The need of immediacy impacts the building process. The chosen algorithm has to be used in a way that allow results to be instantly available. Need of immediacy also impacts the way attributes have to be chosen depending on their time and price of acquisition. For example genetic or blood sample tests are excluded, while questions about lifestyle and women relatives are allowed.

4.2 An Offline Process to Create the Risk Score

Three major constraints affect our process to build a risk score in a way that results in building our risk score in an offline manner:

- As explained in Sect. 4.1, the risk score level has to be displayed in almost real time. Computing all profiles risk scores offline makes instant display very easy, especially when using a k-nearest-neighbor algorithm may lead to large computation time (see modeling step in Sect. 3.2).
- Very often, epidemiology databases are not publicly available because health data are sensitive and their collection are expensive. Offline computation of risk scores prevents making data available in a k-nearest-neighbor based software.
- All stakeholders have to intervene in the process of building the risk score models (see Fig. 2). Having the attributes selection and modeling steps done offline allows to implement in our process the domain expert, the contractor and the data-miner recommendations.

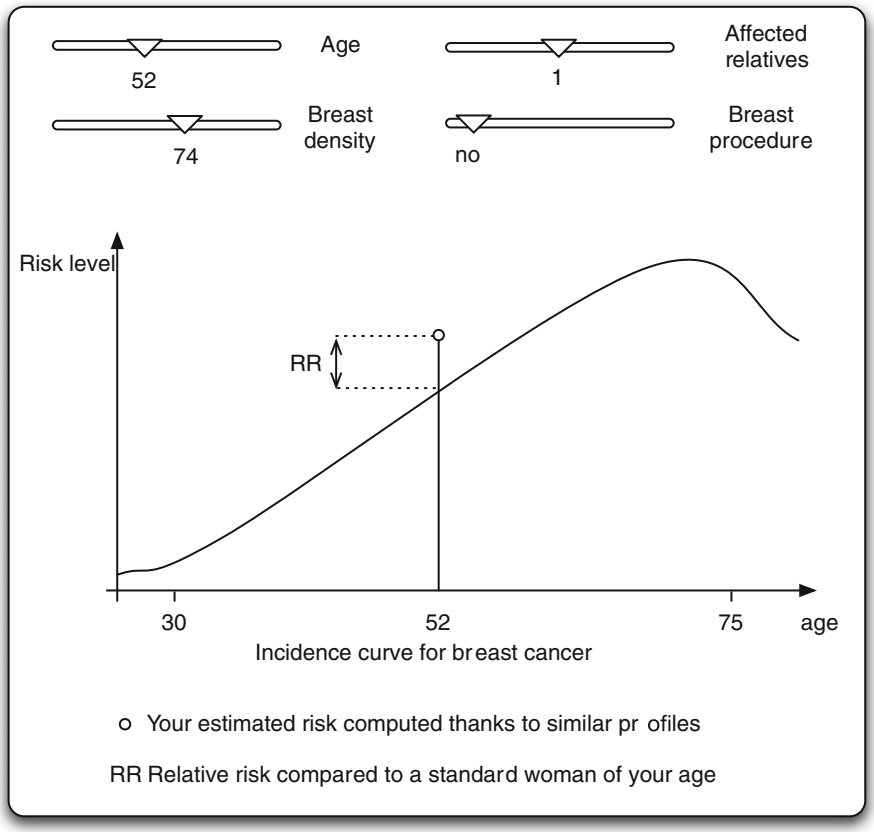


Fig. 3 A graphical user interface prototype

4.3 An Online Graphical User Interface Prototype

As all computation will be done offline, risk score values will be displayed instantly through a responsive graphical user interface (see Fig. 3).

On the graphic, the curve represents the standard incidence of breast cancer depending on the age of the woman. The curve does not evolve when using the software. At the top of the vertical line, the circle represents the woman risk: if the circle is over the curve, the relative risk to standard women is over 1.0 meaning the risk is higher than the average woman of her age. If under, the relative risk is under 1.0, meaning that the risk is lower than the average woman of her age.

Each time a cursor is moved, the graphic will be instantly updated to reflect the risk of the profile. It means that the appropriation of the evolution of the risk level will be made easier for users. Patients or physicians will be able to enter the profile of a woman thanks to the sliders at the top of the interface in order to display the risk level based on real data sources used during the offline part of the process.

Table 1 BCSC database publicly available attributes

Full name	Short name	Description and coding
Menopausal status	menopaus	Premenopausal or postmenopausal
Age group	agegrp	10 categories from 35 to 84 years old
Breast density	density	BI-RADS breast density codes
Race	race	White, Asian/Pacific Islander, Black, Native American, Other/Mixed
Being hispanic	hispanic	Yes or no
Body mass index	bmi	4 category from 10 (underweight) to 35 and more (obese)
Age at first birth	agefirst	Before or after 30 at first live birth or nulliparous (i.e. no children)
First degree relatives	nrelbc	Number of first degree relatives with breast cancer 0, 1 or more than 2
Had breast procedure	brstproc	Prone to breast biopsy, yes or no
Last mammogramm	lastmamm	Last mammogram was negative or false positive
Surgical menopause	surgmeno	Natural or surgical menopause
Hormone therapy	hrt	Being under hormone therapy
Cancer status	cancer	Diagnosis of invasive breast cancer within 1 year, yes or no

This kind of graphical user interface will be tested through a platform in the biggest health center dedicated to oncology in Europe, the Gustave Roussy Institute, using french data [16].

5 Data Source

To build such a graphical tool and to ensure result reproducibility, we have to run the offline part of the process and therefore choose a public database with environmental factors to compute risk levels. The Breast Cancer Surveillance Consortium (BCSC) makes available a database that fits these major constraints. Each of the 2,392,998 lines match to a screening mammogram for a woman. This publicly available database provides 12 attributes to describe the woman including cancer status.

5.1 BCSC Database: Data Collection

Originally, the consortium was conceived to enhance understanding of breast cancer screening practices [2]. The consortium aims at establishing targets for mammography performance and a better understanding of how screenings affect patients in term of actions taken after the mammography. Domain experts from the surveillance

Table 2 Missing data level by attribute

Attribute	Missing data level (%)
Body mass index	55.9
Age at first birth	55.5
Surgical menopause	52.1
Hormone therapy	41.0
Breast density	26.3
Last mammogramm	23.4
Being hispanic	20.3
Race	15.9
First degree relatives	15.2
Had breast procedure	10.5
Menopausal status	7.6
Age group	0
Cancer status	0

consortium identified critical data elements for evaluating screenings performances reaching a consensus on a standard set of core data variables. Then, from 1996 to 2002, data were collected in seven centers across the United States: mammograms and their detailed analysis were collected and, at the same time, women were asked to complete a self-administrated questionnaire.

BCSC database provides personal factors (see Table 1) such as factual factors (age, race, body mass index), reproductive history (age at first birth, menopausal status, hormone therapy) and medical history (number of first degree relatives with breast cancer or type of menopause). In addition, breast density was recorded when the classic Breast Imaging Reporting and Data System (BI-RADS) [25] was used by the radiologist. To ensure good quality of data, exclusion rules were set: for example, women who have undergone cosmetic breast surgery were excluded as well as women with previous breast cancer and women with no known prior mammogram.

Eventually, breast cancer cases were identified by linking cancer registries to BCSC database, i.e. for each record of the database, the class of the example is positive if the corresponding women was diagnosed with breast cancer within one year after the mammogram and completing the questionnaire and negative otherwise.

5.2 BCSC Database: Exploratory Analysis

Among the 2,392,998 records of the database, 9314 cases of invasive breast cancer were diagnosed in the first year of follow up. We are facing highly imbalanced data with a positive class accounting for only 0.39 % of all records.

We also observe a high level of missing data (see Table 2). Two main reasons explain missing data:

Table 3 Breast cancer incidence rate per 100,000

Age category	SEER rate (2003–2007)	BCSC rate (1996–2002)
35–39	58.9	142.7
40–44	120.9	168.1
45–49	186.1	250.5
50–54	225.8	360.7
55–59	280.2	436.4
60–64	348.9	478.5
65–69	394.2	512.3
70–74	410.0	575.1
75–79	433.7	632.0
80–84	422.3	709.4
85+	339.2	Unavailable

- Data were collected in different registries with non-standardized self-reported questionnaire: some questions were not asked and for any question, each woman had the possibility not to answer.
- Collection of some risk factors did not start at the same time. For example, height and weight were added later, explaining such a high rate of missing data for the body mass index.

Last, one has to notice that data of the BCSC are not representative of the USA breast cancer incidence rate (number of new cases during a specified time for a given population). Table 3 offers a comparison between the BCSC and the SEER incidence rate [1] by age categories.

Indeed, depending on data sources, the breast cancer incidence usually increase slowly from approximatively 60–80 years old and starts to decrease after 80 years old. But such a slower increase or decrease does not occur in the BCSC database.

6 Experimental Results

6.1 Scoring Performances

An experiment set was designed to test how the k-nearest-neighbor algorithm perform on the BCSC data. As one of our constraint is to build a readable risk score (see Sect. 3.1), we select all combinations with a size s of 1–6 attributes among $n = 12$ available attributes, meaning we have $\sum_{s=1}^{6} \frac{n!}{s!(n-s)!} = 2509$ combinations to test. A first way of assessing results of these combinations is to look at the best combinations by size (see Table 4). These results are obtained in an euclidian space using a 2-norm Euclidian distance as they are not significantly better, when improved, using another p-norm measures.

Table 4 Best discrimination performances by combination size

Metrics for all combinations by size					Metric for one combination	
Size	Combinations	AUC mean	AUC Std deviation	AUC median	Best combination (see Table 1)	AUC
1	12	0.536	0.030	0.529	agegrp	0.614
2	66	0.563	0.031	0.553	agegrp+density	0.635
3	220	0.581	0.029	0.601	agegrp+density+brstproc	0.641
4	495	0.593	0.026	0.597	agegrp+density+brstproc+lastmamm	0.642
5	792	0.602	0.023	0.586	agegrp+density+brstproc+lastmamm+menopaus	0.642
6	924	0.607	0.019	0.603	agegrp+density+brstproc+lastmamm+hrt+nrelbc	0.637

Among one attribute combinations, *agegrp* is by far the best factor to score breast cancer risk in the BCSC database with an AUC of 0.614, while the next best attribute (not shown), *menopaus* for menopausal status, performs only at 0.563. This result confirms expert knowledge since it is widely known that age is a major breast cancer risk factor.

For combinations size from 1 to 3 attributes, mean, median and best AUC rise, whereas for sizes of 4 and 5 attributes, maximal performances level off around 0.64 with a slight decrease with 6 attributes for best combinations. It is interesting to obtain the best results using less possible attributes to improve model readability. Furthermore, our three attributes *agegrp, density, brstproc* combination has an AUC of 0.641 while in Barlow's results (see Sect. 2.1), at least four attributes are needed to achieve an AUC of 0.631 on a subset of data that includes only premenopausal women only.

A first list of all possible combinations (from 1 to 6 attributes), is produced and sorted by performances (see Table 5-A). We observe that with an AUC of 0.642, the *agegrp, density, brstproc, lastmamm* combination perform better than the two specialized regression models obtained on pre- and postmenopausal women by [3].

6.2 Use of Expert Knowledge

As stated in Sect. 3.1, besides scoring performances, our main objectives also include readability and integration of *a priori* ideas from physicians. This step of the process involves contribution from a domain expert (see Sect. 3.2). From our domain expert point of view, when considering Table 5-A, it appears that the result of the last mammogram is a costly piece of information to obtain from women during a counseling appointment with a physician compared to performance improvement. Domain expert chooses to reduce his choices list to available combinations without *lastmamm*. Top 15 performances measures without *lastmamm* attribute are shown in Table 5-B.

Based on his domain knowledge, the expert highlights that the number of first degree relatives affected by breast cancer (*nrelbc*) is widely recognized as an important factor in breast cancer risk whereas other risk factor, like the body mass index (*bmi*), are not that important compared to others. According to this expert, a good candidate for our risk score would be the *agegrp, density, brstproc, nrelbc* combination with an AUC of 0.637. In addition, this performance is equivalent to the best performances of Barlow's logistic regression models (AUC of 0.624 to 0.631 depending on menopausal status). This combination uses relevant attributes for physicians according to our expert and performance loss, from 0.642 to 0.637, is acceptable. Compared to the *agegrp*, the chosen combination is a valuable performance increase. Moreover the domain expert states that the acceptability of the *agegrp, density, brstproc, nrelbc* combination by physicians, is better than the acceptability of a risk score based on *agegrp* only. It is worth highlighting that on a french database, being specifically built for breast cancer studies, the age of woman attributes only performs a 0.552 [16].

Table 5 Top 15 performance results before and after expert advice

A. Best combinations before expert advice	AUC	B. Best combinations after expert advice	AUC
agegrp, lastmamm, density, brstproc	*0.642*	agegrp, density, brstproc	0.641
menopaus, agegrp, lastmamm, density, brstproc	0.642	menopaus, agegrp, density, brstproc	0.641
agegrp, density, brstproc	0.641	bmi, agegrp, density, brstproc	0.640
menopaus, agegrp, density, brstproc	0.641	agegrp, hispanic, density, brstproc	0.640
bmi, agegrp, density, brstproc	0.640	agegrp, density, brstproc, agefirst	0.639
bmi, agegrp, lastmamm, density, brstproc	0.640	bmi, agegrp, density, brstproc, race	0.638
agegrp, hispanic, density, brstproc	0.640	menopaus, agegrp, hispanic, density, brstproc	0.638
agegrp, density, brstproc, agefirst	0.639	agegrp, density, brstproc, race	0.638
agegrp, hispanic, lastmamm, density, brstproc	0.639	menopaus, agegrp, surgmeno, density, brstproc	0.638
bmi, agegrp, density, brstproc, race	0.638	agegrp, hispanic, density, brstproc, agefirst	0.638
menopaus, agegrp, hispanic, density, brstproc	0.638	bmi, agegrp, hispanic, density, brstproc	0.638
hrt, agegrp, lastmamm, density, brstproc	0.638	menopaus, agegrp, density, brstproc, agefirst	0.638
agegrp, density, brstproc, race	0.638	bmi, agegrp, density, brstproc, agefirst	0.637
agegrp, surgmeno, lastmamm, density, brstproc	0.638	menopaus, hrt, agegrp, density, brstproc	0.637
agegrp, lastmamm, density, brstproc, race	0.638	*agegrp, density, brstproc, nrelbc*	*0.637*

Calibration results shows that the chosen combination (*agegrp, density, brstproc, nrelbc*) has an E/O ratio of 1.01. It is better than the 1.02 E/O ratio of both top combinations *agegrp, density, brstproc* and *agegrp, lastmamm, density, brstproc* (Table 5). It is also better than the 1.02 E/O ratio of *agegrp* alone.

6.3 Performances with Respect to k

In order to run a k-nearest-neighbor algorithm, the size of neighborhood has to be set. Since only k closest neighbors are used to compute the ratio healthy vs. diseased, risk score value depends on k value. If the neighborhood is too small, few breast cancer cases are included and if the neighborhood is too large, patient profiles are too different: in both cases the risk score is not reliable. For each of the 2509 combinations of attributes, we tested the scoring function with 40 values of k from 100 to 100,000.

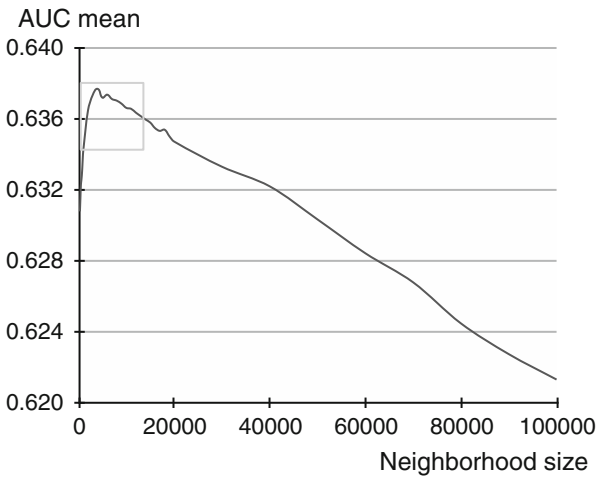


Fig. 4 Performances of top 15 combinations from Table 5-B

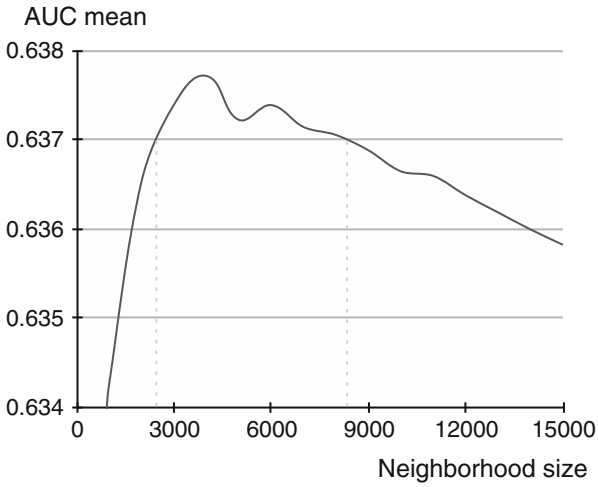


Fig. 5 Zoom on performances of top 15 combinations from Table 5-B

Using, as an example, the top 15 combinations from Table 5-B, we plotted the evolution of the performance (using the AUC mean) depending on the size of the neighborhood (see Fig. 4). With an undersized neighborhood, performances are low but then, as k increases, performances increase with a maximum of 0.638. From 2500 to 8400 neighbors (see Fig. 5), performances are always higher than 0.637 meaning that the algorithm is relatively stable depending on k and ultimately on the number of positive examples in the neighborhood. Eventually, as k increases, performances decrease because using a larger neighborhood leads to compute a ratio with increasingly dissimilar profiles and poor targeting.

It means that performance of the combination is not obtained with a local maximum for a single value of k. It rather depicts overall prediction ability of a

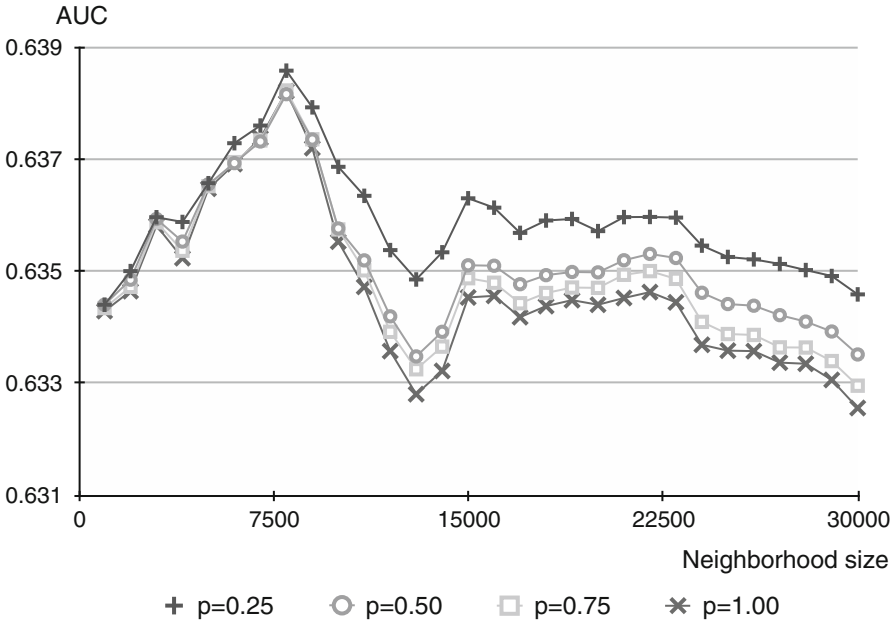


Fig. 6 Performances of best combination depending on k and the weighting function

combination independently of the value of k as long as the size of the neighborhood is large enough to be statistically reliable (according to the law of large numbers) and stringent enough to eliminate too dissimilar profiles.

Eliminating dissimilar profiles can be done using a weighting function [9] to decrease neighbors weight (w_i) in the prevalence computation (see Sect. 3.3 in which w_i implicitly worth 1) depending on the Euclidean distance (d_i):

$$w_i = \left(\frac{d_i}{d_{max}}\right)^p$$

with d_{max}, the greatest distance among the neighborhood and $p \in \{0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 4.0\}$.

The prevalence is computed for the *agegrp, density, brstproc, nrelbc* combination selected by domain expert with $k \in [1000; 30,000]$ neighbors. AUC performance results are plotted in Fig. 6 only for $p = 0.25$, $p = 0.5$, $p = 0.75$ and $p = 1.0$ because performances curves for $p = 2.0$, $p = 3.0$ and $p = 4.0$ weighting functions are indistinguishable from curve for $p = 1.0$. Maximal performances peak is not significantly enhanced as AUC increase is less than 0.001. But when p tends to decrease, mean value of AUC increases for k in $[1000; 30,000]$. It suggests that the choice for the k value in the k-nearest-neighbor algorithm is less critical when using a weighting function because the stability range, where performances are upon a minimal value, is larger. Optimal value of k can be found more easily, making the use of the k-nearest-neighbor algorithm more independent from k.

6.4 Discussion

As statistical risk scores are not commonly used in the medical community, we think there is a possibility to improve risk scores to offer both readability in its elaboration and possibility for experts to integrate their knowledge (regarding end users expectations and the disease itself) in the process. A standard methodology called *CRISP-DM* was followed in the process of building such a risk score. The database from the BCSC was selected because a regression-based score was already built upon it and because the database itself was publicly available. We chose to run extensive test with a k-nearest-neighbor algorithm to score profiles with different combinations of attributes. Every combinations with 1–6 attributes were tested, each for several values of k neighbors. Thus, we were able to allow experts to establish rules to keep or reject combinations by weighting between performance versus attributes usefulness and risk factors expected by physicians.

Nevertheless, our study has some limitations. First, on one hand, even if we selected one of the few databases large enough to be representative of the targeted population, findings from database of volunteers require cautious extrapolation to general population. On the other hand, as we use prevalence to link a profile to a risk level, even if some profiles are under or over-represented compared to general population, it has limited impact on the risk score because we used the prevalence as a score value. Second, if the concept of similarity used in the algorithm is easy to understand for everyone, performances may be limited due to imbalanced data and the constraint of not modifying data used in this article in order to be able to compare results. However, options are available to improve steps of the process. Better performances may be obtained using another algorithm, potentially with balance of data in the data preparation step [12, 18, 23], or by combining k-nearest-neighbor with another algorithm [20, 22, 24]. Use of expert knowledge could be improved by selecting models which are provided to the expert to avoid complications due to the size of the list of combinations.

Increased acceptability could be reached by integrating actionable attributes. Indeed, to make more interactive softwares and increase patient involvement in the risk measurement process, actionable risk factors as attributes may improve the prevention process with the goal to lower the risk. It implies close work with epidemiologists who lead data collection.

Since k-nearest-neighbor algorithm gives good results, we will continue to test this process on another database that include continuous attributes that were not discretized. For example age or breast density are some of the most predictive attributes and more specific data should improve performances. Higher risk profiles should be more accurately targeted leading to increased performances.

In the same time, we are developing softwares for physicians use based on prototype presented in Sect. 4.3.

7 Conclusion

On a medical dataset, we obtain good results on readability on the modeling method with a *k*-nearest-neighbor algorithm easy to understand for physicians and patients. In addition, the score is very easy to use for end-users with only four attributes needed through a prototype of a graphical user interface. Thanks to our offline process, we also allow the expert to choose a combination that has not necessarily the best detection performance, but show qualities like physician acceptance and inclusion of performant attributes recognized by the community.

Our approach is innovative and successful because we have shown that it is possible to build a simple and readable risk score model for primary breast cancer prevention that performs as good as widely used logistical models.

References

1. Howlader, N., Noone, A.M., Krapcho, M., Garshell, J., Miller, D., Altekruse, S.F., Kosary, C.L., Yu, M., Ruhl, J., Tatalovich, Z.,Mariotto, A., Lewis, D.R., Chen, H.S., Feuer, E.J., Cronin, K.A. (eds). SEER Cancer Statistics Review, 1975-2011, National Cancer Institute. Bethesda, MD (2010)
2. Ballard-Barbash, R., Taplin, S., Yankaskas, B., Ernster, V., Rosenberg, R., Carney, P., Barlow, W., Geller, B., Kerlikowske, K., Edwards, B., Lynch, C., Urban, N., Chrvala, C., Key, C., Poplack, S., Worden, J., Kessler, L.: Breast cancer surveillance consortium: a national mammography screening and outcomes database. Am. J. Roentgenol. **169**(4), 1001–1008 (1997)
3. Barlow, W.E., White, E., Ballard-Barbash, R., Vacek, P.M., Titus-Ernstoff, L., Carney, P.A., Tice, J.A., Buist, D.S.M., Geller, B.M., Rosenberg, R., Yankaskas, B.C., Kerlikowske, K.: Prospective breast cancer risk prediction model for women undergoing screening mammography. J. Natl. Cancer Inst. **98**(17), 1204–1214 (2006)
4. Chapman, P., Clinton, J., Kerber, R., Khabaza, T.: CRISP-DM 1.0 step-by-step data mining guide. Tech. Rep., The CRISP-DM Consortium (2000)
5. Chen, J., Pee, D., Ayyagari, R., Graubard, B., Schairer, C., Byrne, C., Benichou, J., Gail, M.H.: Projecting absolute invasive breast cancer risk in white women with a model that includes mammographic density. J. Natl. Cancer Inst. **98**(17), 1215–1226 (2006)
6. Costantino, J., Gail, M., Pee, D., Anderson, S., Redmond, C., Benichou, J., Wieand, H.: Validation studies for models projecting the risk of invasive and total breast cancer incidence. J. Natl. Cancer Inst. **91**(18), 1541–1548 (1999)
7. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
8. Decarli, A., Calza, S., Masala, G., Specchia, C., Palli, D., Gail, M.H.: Gail model for prediction of absolute risk of invasive breast cancer: Independent evaluation in the Florence-European prospective investigation into cancer and nutrition cohort. J. Natl. Cancer Inst. **98**(23), 1686–1693 (2006)
9. Dudani, S.A.: The distance-weighted *k*-nearest-neighbor rule. IEEE Trans. Syst. Man Cybern. **6**(4), 325–327 (1976)
10. Egan, J.P.: Signal detection theory and ROC analysis. Academic Press series in cognition and perception. Academic (1975)
11. Endo, A., Shibata, T., Tanaka, H.: Comparison of seven algorithms to predict breast cancer survival. Biomed. Soft Comput. Hum. Sci. **13**(2), 11–16 (2008)

12. Fan, X., Tang, K., Weise, T.: Margin-based over-sampling method for learning from imbalanced datasets. In: Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Lecture Notes in Computer Science, Springer (2011)
13. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**(8), 861–874 (2006)
14. Fix, E., Hodges, J.L.: Discriminatory analysis, non-parametric discrimination: consistency properties. Tech. Rep., USAF Scholl of Aviation and Medicine, Randolph Field (1951)
15. Gail, M.H., Brinton, L.A., Byar, D.P., Corle, D.K., Green, S.B., Schairer, C., Mulvihill, J.J.: Projecting individualized probabilities of developing breast cancer for white females who are being examined annually. J. Natl. Cancer Inst. **81**(24), 1879–1886 (1989)
16. Gauthier, E., Brisson, L., Lenca, P., Clavel-Chapelon, F., Ragusa, S.: Challenges to building a platform for a breast cancer risk score. In: Sixth International Conference on Research Challenges in Information Science, pp. 1–10. IEEE (2012)
17. IARC: World Cancer Report. IARC Publications. http://www.iarc.fr/en/publications/pdfs-online/wcr/2008/wcr_2008_1.pdf (2008)
18. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intell. Data Anal. **6**(5), 429–449 (2002)
19. Jerez-Aragonés, J.M., Gómez-Ruiz, J.A., Ramos-Jiménez, G., Muñoz-Pérez, J., E., A.C.: A combined neural network and decision trees model for prognosis of breast cancer relapse. Artif. Intell. Med. **27**(1), 45–63 (2003)
20. Li, Y., Zhang, X.: Improving *k* nearest neighbor with exemplar generalization for imbalanced classification. In: Huang, J., Cao, L., Srivastava, J. (eds.) Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science, vol. 6635, pp. 321–332. Springer, Berlin (2011)
21. Lichtenstein, P., Holm, N.V., Verkasalo, P.K., Iliadou, A., Kaprio, J., Koskenvuo, M., Pukkala, E., Skytthe, A., Hemminki, K.: Environmental and heritable factors in the causation of cancer, analyses of cohorts of twins from Sweden, Denmark, and Finland. N. Engl. J. Med. **343**(2), 78–85 (2000)
22. Liu, W., Chawla, S.: Class confidence weighted knn algorithms for imbalanced data sets. In: Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining. Lecture Notes in Computer Science, vol. 6635, pp. 345–356. Springer, Berlin (2011)
23. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. B **39**(2), 539–550 (2009)
24. Pham, N.K., Do, T.N., Lenca, P., Lallich, S.: Using local node information in decision trees: coupling a local labeling rule with an off-centered entropy. In: The International Conference on Data Mining, pp. 117–123. Las Vegas, Nevada, USA. CSREA Press (2008)
25. D'Orsi, C.J., Sickles, E.A., Mendelson, E.B., Morris, E.A., et al.: ACR BI-RADS® Atlas, Breast Imaging Reporting and Data System, Reston, VA, American College of Radiology (2013)
26. Teams, F.C.: Mammographic surveillance in women younger than 50 years who have a family history of breast cancer: tumour characteristics and projected effect on mortality in the prospective, single-arm, fh01 study. Lancet Oncol. **11**(12), 1127–1134 (2010)
27. Testard-Vaillant, P.: The war on cancer. CNRS Int. Mag. **17**, 18–21 (2010)
28. Visa, S., Ralescu, A.: Issues in mining imbalanced data sets–a review paper. In: Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, MAICS-2005, Dayton, pp. 67–73 (2005)
29. Weiss, G.M., Provost, F.: Learning when training data are costly: the effect of class distribution on tree induction. J. Artif. Intell. Res. **19**, 315–354 (2003)

Machine Learning for Medical Examination Report Processing

Yinghao Huang, Yi Lu Murphey, Naeem Seliya and Roy B. Friedenthal

Abstract Vast amounts of human medical documents contain rich knowledge that can be used to facilitate a broad range of medical research and clinical study. One important application is to automatically categorize medical documents into specific categories. However, those medical documents usually contain names and identities of patients and doctors that are not allowed to be disclosed due to patient privacy and regulation issues concerning medical data. In this article, we address two issues, automatic name entity detection, and automatic classification of medical reports. We present a name entity recognition system, MD_NER_NCL, and a text document classification system, C_IME_RPT for medical report processing and categorization. The MD_NER_NCL contains an innovative segmentation algorithm, called HBE segmentation, that segments a medical text document into the Heading, Body and Ending parts, and a statistical reasoning process that utilizes knowledge of three entity lists: people name prefix list, people name suffix list, and false positive prefix list. The C_IME_RPT is developed based on Self Organizing Maps (SOM) and a machine learning process. Both systems have been evaluated using Independent Medical Examination (IME) reports provided by medical professionals. The proposed system MD_NER_NCL made a significant improvement over the well-known text analysis software, OpenNLP, for people name entity detection. The C_IME_RPT system attained a 89.9% classification accuracy, which is very good in clinical record classification. We also present an in-depth empirical study on the effectiveness of

Y. Huang (✉) · N. Seliya
Computer and Information Science, University of Michigan—Dearborn,
Dearborn, MI 48128, USA
e-mail: yinghaoh85@gmail.com

N. Seliya
e-mail: nseliya@umich.edu

Y. L. Murphey
Electrical and Computer Engineering, University of Michigan—Dearborn,
Dearborn, MI 48128, USA
e-mail: yilu@umich.edu

R. B. Friedenthal
Central Orthopedics, 820 S. White Horse Pike, Hammonton,
NJ 08037, USA
e-mail: roy@comcast.net

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_14

parameters associated with the SOM learning process and text mining, and their effects on classification results.

1 Introduction

Human medical document classification is one of the most rewarding, yet very difficult, applications in data mining. According to Cios and Moore [11], about three quarter billions of people living in North America, Europe, and Asia have at least some of their medical information collected in an electronic form. These documents contain rich knowledge that facilitates research on medical informatics and clinical study. Because of the large volume of medical data, automatic knowledge discovery algorithms are very much in demand in medical research [18]. However in most of medical documents, patients' private information is recorded by physicians, nurses, and other medical staff in plain text documents. There are ethical, legal and social constraints imposed on medical data collection, distribution, and analysis due to private information embedded in medical documents. The most important information needs to be protected is the names of patients, physicians and organizations. Therefore one of our research focuses is to develop machine learning algorithms for training an automated system that can detect name entities and encrypt them before the documents are being distributed for further processing. The second focus of this research is to develop text mining technologies for automatic classification of medical document based on user (physicians and nurses) defined groups.

The medical documents of our interest are Independent Medical Examination (IME) reports that are in the form of correspondences between physicians and patients or various third parties such as insurance companies, patients' employers, and attorneys [20]. These IME reports contain descriptions of patients' ailments and are written in various style and format. An automatic system to categorize these IME reports is very useful for many medical researchers. For example, when interested parties want to make decisions regarding medicine based on the medical examination reports from physicians, it is arduous and time consuming for human experts to review and analyze thousands of hundreds of medical reports, and classify them based on specific user defined criterions.

In this paper, we present a name entity detection model, MD_NER_NCL and a SOM (Self-Organizing-Map) based machine learning system, C_IME_RPT, for medical report classification. The MD_NER_NCL consists of an automatic document segmentation process and a statistical reasoning process to accurately detect and classify name entities. We will show that the proposed algorithm provides much improved recall and precision for name detections in the medical text documents in our case study than the OpenNLP program. The Self-Organizing-Map-based machine learning system consists of a vector space model for representing medical text documents and a SOM learning process to generate a classification system for the automatic categorization of the IME reports. The evaluation of various training parameters is presented and their effects on the classification model are analyzed.

The performance of proposed IME report classification system, C_IME_RPT is compared with six well-known text clustering and classification methods, K-NN, K-means, Hierarchical K-means, Naïve Bayesian, Random Forest and SVM. The results show that the C_IME_RPT has better accuracy in categorizing documents over all categories and, in particular, in minority categories.

The remainder of this article paper is organized as follows. Section 2 gives an overview of the technologies developed for the two main topics addressed in this article, name entity detection, and medical document classification. Section 3 and 4 introduce, respectively, the proposed name entity detection algorithm and the machine learning algorithm for medical document classification. Section 5 presents the results of our case study conducted on medical IME reports, and Sect. 6 concludes the article.

2 Overview of Related Works

2.1 Research in Name Entity Recognition

Name entity recognition (NER) is an important process in many information retrieval and natural language processing (NLP) applications that require privacy protection and/or correlating name entities with specific groups of people. Various techniques of NER have been developed. The most popular techniques are Conditional Random Fields (CRF), entropy based, and classification based.

McCallum and Li introduced a method that combines Conditional Random Fields (CRF) with feature induction and Web-augmented lexicons for NER [34]. With this method, they were able to obtain 84.04 % measured in F1 on the Routers Corpus provided by the CoNLL-2003 name entity recognition (NER) Shared Task. F1 is defined as $F1 = 2 \times \frac{precision \times recall}{precision + recall}$. The Maximum Entropy based technique has been used by a number of researchers for NER. Chieu and Ng presented a Maximum Entropy-based NER model that makes use of both local and global information features to classify each word [10]. They obtained 93.27 and 87.24 % in F1 measure on news documents in the MUC-6 and MUC-7 collections respectively. Bender et al. developed a system that is also based on the Maximum Entropy Model [3]. Starting with an annotated corpus and a set of entropy features they first built a baseline NE recognizer, which was then used to extract the name entities and their context information from the non-annotated data. Their system obtained an overall 83.92 % measured in F1 on the NER test set provided by the CoNLL-2003 Shared Task. Many NER systems were built upon well-trained classifiers. MayField and McNamee developed a SVM based system to solve the NER problem [33]. The performance of their NER system reached 84.67 % for the English data provided by the CoNLL-2003 named entity recognition (NER) shared task. Florian and Ittycheriah proposed an experimental framework of four classifiers for named entity recognition [16]. They developed four classifiers based on robust linear classification, maximum entropy, transformation-based learning, and hidden Markov model. The NER detection system attained a performance of 91.6 % in F1 measure.

Our research focuses on NER in correspondence documents, which have not been addressed by other researchers. Our approach is to first segment documents into three different parts, Heading, Body and Ending, and then apply different strategies to these parts for NER. The precision of the system on person name reached 98.05 %.

2.2 Research in Medical Document Classification

In text document classification, documents are often transformed from a full text version to a document vector which describes the contents of the document. The most used technique is to represent document contents in terms and frequencies. Machine learning techniques have been actively explored for text document classification. Among these are neural networks [8, 7, 15, 37], support vector machines [2, 22], genetic programming [42]. Kohonen type self-organizing maps [21], hierarchically organized neural network built up from a number of independent self-organizing maps [35], fuzzy k-means [4], hierarchical Bayesian clustering [30], Bayesian network classifier [25], and naïve Bayes classifier [36]. In this article paper, we focus on medical document categorization.

Research in the field of medical text document retrieval and classification has been conducted by a number of researchers. Stephen et al. developed several text analysis systems for medical documents [40, 41] including BADGER, CRYSTAL and WHISK. BADGER is a system that identifies concepts embedded in a text based on linguistic context. Cases are represented in concept nodes, each of which is a set of syntactic and semantic constraints. CRYSTAL is a system that automatically induces a dictionary of "concept-node definitions" sufficient to identify relevant information in a training corpus. The learning strategy is mostly rule based and high in computational cost because of the extensive semantic analysis, and requires manually labeled concepts and annotations. "WHISK" is a text mining system designed to learn text extraction rules from modified regular expressions and relationship between isolated facts. The system focuses on information extraction from clinical records at sentence level and is capable of processing both structured and semi-structured text. However, as the authors pointed out that it does not perform well when document context has high variation. Claster et al. developed a system to analyze the radiology department records of children who had undergone a CT scan in 2004 [12]. They used a technique based on self organizing map to identify keywords with significance values within the narratives of the medical records that could predict whether a CT scan will be beneficial in the diagnosis/management of children and, thereby, lower the number of unnecessary CT requests by clinicians. A SOM network was used to discover associations among different key words for decision making.

Zhou et al. describe a MEDical Information Extraction (MedIE) system that extracts and mines a variety of patient information from free-text clinical records [47]. Three approaches are proposed to solve different IE tasks. A graph-based approach which uses the parsing result of link-grammar parser was developed for relation extraction. A simple but efficient ontology-based approach was adopted to extract

medical terms of interest. Finally, an NLP-based feature extraction method coupled with an ID3-based decision tree was used to perform text classification. Manine, Alphonse and Bessieres proposed a rich modeling of gene ontology, and showed that it could be used within an IE system [31]. The ontology was seen as a language specifying a normalized representation of text. Inference rules were learnt with an inductive logic programming (ILP) algorithm, using the ontology as the hypothesis language and its instantiation on an annotated corpus as the example language. Learning is set in a multi-category setting to deal with the multiple ontological relations.

While text mining has been applied to a few medical applications as discussed above, new techniques applied to clinical contents are still in demand [13]

3 Name Entity Detection

In this section, we introduce a name entity recognition algorithm developed to detect and encrypt private information such as names and addresses embedded in medical reports. Since medical documents should not be viewed and analyzed with private information, the very first step in any medical document processing is to detect and encrypt the name entities contained in the documents. The name entity recognition problem addressed in this article serves as a preprocessing stage for a data encryption process. The basic requirements for the encryption process are as follows: all names, persons or organizations, should be encrypted, the same names should be encrypted with the same code word, and different names should be encrypted with different code word. The encryption requirements dictate the following performance requirements of a NER system:

- Minimizing the missing rate,
- Minimizing partial recognition of entities,
- Minimizing false positives, and
- Classifying recognized names into the pre-code categories for proper encryption. For example, patients may have a pre-code as PP, doctors as PD, hospitals as OH, etc.

MD_NER_NCL is developed to meet these requirements. As illustrated in Fig. 1, the MD_NER_NCL consists of three major computational algorithms, HBE segmentation, sentence extraction and tagging, name entity detection and classification. The HBE segmentation algorithm partitions a document into three categories: heading, ending, and text body part. Then individual sentences from each document part are extracted and used as input to the name entity detection algorithm to detect names of people and organization. The following subsections give detailed description about these algorithms.

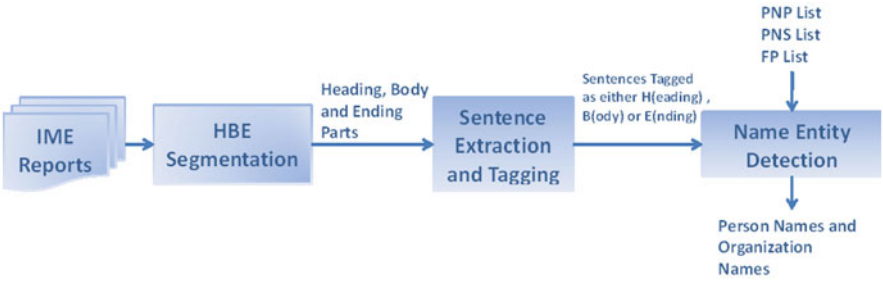


Fig. 1 Major computational steps in MD_NER_NCL

Fig. 2 General format of medical correspondence

Date Line (e.g. August 13, 1998)

Address Lines (e.g. CXX PXX and CXXX Companies,
XXX WXX HXX Road, P.O Box XXXX,
VXX, Michigan, 00000)

Receiver Lines (e.g. ATTN: MXX MXXX
Claims Representative)

Patient Lines (e.g. RE: KXX KXXX
FILE#: 000 A 00 00 00-0/000)

Body Part (e.g. Line begins with "Dear Mr./Mrs. MXXX")

Writer Lines (e.g. Sincerely yours, RXX F. FXX, M.D.)

3.1 HBE Segmentation and Sentence Extraction

A typical IME report contains a sequence of correspondences. Each correspondence (see Fig. 2) contains a Heading, a Body and an Ending part. A Heading part includes the date line, as well as the name and address of the received entity. The Body part immediately follows the Heading and ends before the solution line. The Ending part includes the solution line and the sender's name. The Heading parts can contain a variety of names, e.g. person names, organization names, street names, city names,

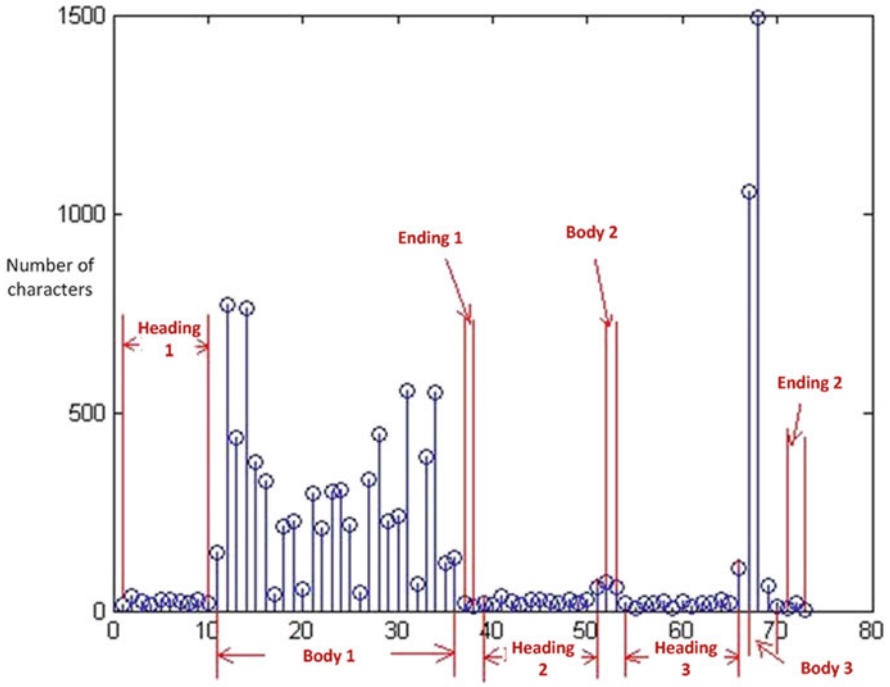


Fig. 3 Line projection of an IME document

etc., and these names often appear quite differently as the names appearing in the body and ending parts. The sentences in the heading and the ending parts are not separated by punctuations as those in the body part, and every line in the heading contains some sort of names. We noticed during our empirical study that the OpenNLP model has difficulties in detecting names particularly embedded in the header sections of IME documents. A document segmentation algorithm, named HBE (Heading, Body, Ending) algorithm is developed based on the analysis of the line projection of a document. The line projection of a document, D, is defined as the histogram of characters for every line entity in D. A line entity is defined as a sequence of words between two line breaks in a document. Figure 3 shows the line projection of an IME document that contains multiple correspondences. In general the Heading and Ending parts contain line entities that are much shorter than a full document line, while the Body parts contain mostly long line entities that occupy multiple document lines. Therefore the line projection of an IEM document provides rich knowledge for segmenting the Heading, Body and Ending parts in the IME document. The HBE segmentation algorithm uses a parameter, *Max_length*, to quantify the difference of line entity in the Body parts and in the Heading and Ending parts. *Max_length* is selected based on the examination of a number of IME reports. For example, the number of characters in a full document line in a word document with font Times New Roman and size equal 9 is about 119, and for font size 12 about 86. In order

to accommodate different font types and sizes, it is safe to set the maximum length of a full line in a document to be less than 150 characters, i.e. *Max_length* = 150. If the length of a line entity is greater than *Max_length*, then the line entity is within a Body part, since it occupies more than one document lines. The major computational steps in the HBE algorithm are described as follows.

1. For an input text document, segment the text into "line entities" through line breaks and generate the line projection histogram.
2. Search the line projection histogram line by line starting by setting the first line entity as the current line entity.
3. For the current line entity do the following:

   (a) If the current line entity has no period at the end and its length is less than 50 % of *Max_length*, it is a heading line.
      (i) If the previous line entity is also a heading line, add the current line to the current Heading.
      (ii) If the previous line entity is not a heading line, generate a new Heading and mark the current line entity as the beginning of a new heading.
      (iii) Make the next line entity in the document as the current line entity and repeat Step 3.
   (b) If the current line entity ends with a comma and has the keyword "sincerely" and its length is less than 50 % of the *Max_length*, it is an ending line entity.
      (i) Mark the line entity as the beginning line of a new Ending part.
      (ii) Mark the last line entity as the closing line of the current Body part.
      (iii) Make the next line entity in the document as the current line entity and add it to the Ending part.
      (iv) Make the next line entity in the document as the current line entity and repeat Step 3.
   (c) If the current line entity does not meet the conditions stated in (a) and (b), it belongs to the Body part.
      (i) Add it to the current body part.
      (ii) Make the next line entity in the document as the current line entity and repeat Step 3.

The output of the HBE algorithm is a sequence of Heading, Body and Ending parts in the input IME document. After the HBE segmentation process, the sentence entities in the Heading and Ending parts are extracted by searching for line breaks, and in the Body parts by looking for punctuation periods.

3.2 Name Entity Detection

The name entity detection algorithm consists of two major processes, name candidate detection and statistical reasoning to recover missing names and eliminate false name candidates.

The name candidate list, **NC_L**, for a document D is generated as follows. First we apply the HBE and Sentence Extraction algorithm to D to obtain a sequence of Heading, Body and Ending parts and the sentence entities. For each of the Heading and Body parts, the extracted sentence entities are sent to OpenNLP Maxent Model for name candidate detection. For each of the Ending parts, since they are all homogeneous "Writer Lines" as shown in Fig. 2, we search in every sentence entity for "sincere". If there is a match, we extract writer's name from the next sentence entity. Since a writer's name is often followed with a suffix such as "M.D," we check these possible suffixes against the suffix list **PNS_L**. If there is a match, then the suffix is removed. If no "sincere" and name suffix is found in a sentence entity, we send it to OpenNLP Maxent Model for name entity detection. **NC_L** contains all the name candidates detected by this process.

We noticed that the OpenNLP software could miss name entities and also generate false name entities. For example, words at the beginning of the sentences such as "Medical", "There", "Straight" and etc. were mistakenly classified as person names, while some of the person names that have middle name such as "Lori S. Kingsler", "Larry D. Rosenberg" were only partially recognized. Some names were entirely missed, such as "Marybeth Molloie", "Pamela Heenan", etc. We designed the following statistical reasoning process to recover missed names and reduce false positives. The statistical reasoning process uses three reference lists, person name prefix list (**PNP_L**), person name suffix list (**PNS_L**), and false positive prefix list (**FP_L**), that are generated through the following machine learning process.

A training data set of IME documents is collected and all name entities in these documents should be labeled. The HBE segmentation algorithm is applied to the training data to extract the Headings, Body and Ending parts, and then the OpenNLP program is applied to these three types of document parts for name entities detection using the process described at the beginning of this subsection. The detected name entities along with the labeled name entities are used to generate the three reference lists using the following procedure.

The **PNP_L** is generated by extracting all the name prefixes from the training documents and calculating the risk factor of each prefix using the formula: $\alpha_i^{pnp} = \frac{nf_i^{pnp}}{np_i^{pnp}+nf_i^{pnp}}$, where nf_i^{pnp} is the number of the no-name entities in the training documents that occurred after the ith prefix on the **PNP_L**, and np_i^{pnp} is the number of true name entities occurred after the ith prefix. The **PNS_L** is generated by extracting all person name suffixes, from the training data set and calculate the risk factors using the similar formula above: $\alpha_i^{pns} = \frac{nf_i^{pns}}{np_i^{pns}+nf_i^{pns}}$, where nf_i^{pns} is the number of the no-name entities in the training documents that occurred before the ith suffix on the **PNS_L**, and np_i^{pns} is the number of true name entities occurred before the ith suffix. The false positive prefix list, **FP_L**, is generated by storing all the prefixes of false names generated by the OpenNLP and their risk factors. The risk factor for a false prefix is calculated using formula $\alpha_i^{fp} = \frac{np_i^{fp}}{np_i^{fp}+nf_i^{fp}}$, where np_i^{fp} is the number of the true name entities in the training documents that occurred after the ith prefix on the **FP_L**, and nf_i^{fp} is the number of the false name entities after

the ith prefix. These prefix, suffix and FP lists are used as the knowledge base in the following statistical reasoning process to recover missing names and eliminate false names.

In this statistical reasoning, we use a Part-Of-Speech Tagger (POS Tagger) developed by the Stanford Natural Language Processing Group to tag the terms, and so the search for prefix and suffix of terms is carried out only on terms tagged as "noun" to reduce the searching space. The POS tagger is a Java implementation of the log-linear part-of-speech taggers [43], and it uses the Penn Treebank [32] to tag terms.

3.2.1 Statistical Reasoning Algorithm

Let **NC_L** be the name candidate list generated from an input document D by the name candidate detection process described above.

1. Apply POS Tagger to input document D obtain a term-tag list, denoted as **TT_L**.
2. For every term t_i in **TT_L**, if it is tagged as a noun, then find prefix and suffix of t_i: $p\_\text{term}_i, s\_\text{term}_i$, and execute Steps 3 through Step 5.
3. If $p\_\text{term}_i \in$ **PNP_L** and α_i^{pnp} is lower than a threshold, add t_i into **NC_L**.
4. If $s\_\text{term}_i \in$ **PNS_L** and α_i^{pns} is lower than a threshold, add t_i into **NC_L**.
5. If $p\_\text{term}_i \in$ **FP_L** and α_i^{fp} is lower than a threshold, put t_i into a flase term list, denoted as **FT_L**.
6. Remove redundant terms from **NC_L** using a hash table.
7. Remove false terms from **NC_L** that occur in **FT_L** using a hash table.
8. Output **NC_L**, the detected name list of people and organizations.

The above algorithm is computationally efficient. It requires only one-time search of the input document to generate the name list.

The name list, **NC_L**, is then encrypted by encryption software. The encrypted documents are then distributed for further process such as document classification.

4 SOM Based Medical Document Classification

Our approach to IME report classification consists of multiple stages, extracting paragraphs of interest, generating an algebraic model for document representation, SOM learning procedure and a procedure for evaluating and selecting the optimal SOM model for IME report classification. Figure 4 illustrates the data flow between these major computational components in the proposed machine learning process. The following subsections describe the detail of these components.

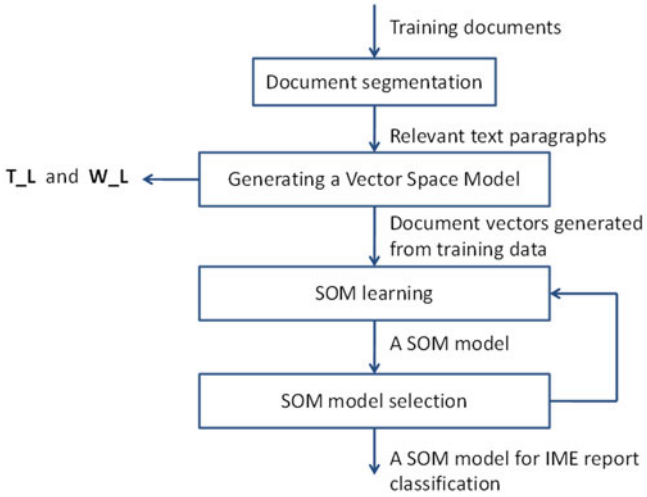


Fig. 4 A SOM based machine learning process for medical document classification

4.1 Document Segmentation

In many medical practices, an IME document contains only a few paragraphs that are relevant to the classification problem, and these paragraphs of interests often contain some relevant keywords. The document segmentation process is to extract paragraphs of interests from IME reports to be used for classification. The first step is to apply the HBE algorithm to the IME reports to extract the body parts of these reports. If there is no domain knowledge about the paragraphs of interests, the body parts of these reports are used as the relevant text paragraphs in classification. However, in many applications, domain knowledge about the paragraphs of interests is available. For examples, a list of subtitles that the paragraphs of interests are placed after, and/or a list of keywords often contained in these paragraphs. If such knowledge is available, the document segmentation process will extract paragraphs under these subtitles and the paragraphs containing any of these keywords. For example for the application problem addressed in this article, the medical experts provided the following lists, subtitle list **S_L** = {Assessment:, Impression:, Treatment:, Discussion:, Summary:, Conclusion:}, and the keyword list **K_L** = {assessment, diagnostic, impression, discussion, summary, conclusion, treatment, permanent, permanency, disability}. Only the paragraphs either associated with these subtitles or containing any of these keywords were extracted for further processes.

4.2 A Vector Space Model for Medical Document Classification

In text classification and retrieval, a text document is usually represented in the form of a vector model [27, 39], which is a meaningful and concise way for computation and analysis. A vector space model for representing text documents should be

built based on carefully selected terms and weighting schemes [39]. A vector space model can be defined by a term list, **T_L** = $\{t_1, t_2, \ldots, t_K\}$ and a weight list **W_L** = $\{w_1, w_2, \ldots, w_K\}$, where t_i is the ith important term, w_i is the product of a local weight and a global weight for term t_i, and K is the number of words or terms that are important for text document classifications. Our vector space model is built through the following machine learning process.

For a given set of training documents, $Tr = T_1 \cup T_2 \cup \cdots \cup T_{\text{N\_C}}$, where N_C is the number of document categories, and T_c is the set of documents that belong to category c, $c = 1, \ldots,$ N_C. Please note that the training set contains important paragraphs extracted by the document segmentation process discussed earlier. First we generate an ordered index term list **T_L** that contains all the words and phrases extracted from all the documents in Tr. These words and terms are then filtered through a number of processes including word stemming and stopping word removal. In order to keep only content bearing words on **T_L**, we also remove the words that have either too low or too high frequency in Tr [19] by using the following strategy.

For an IME report, D_j, let its vector representation be $W_j = (tf_{1j} * g_1, \ldots, tf_{ij} * g_i, \ldots, tf_{Kj} * g_K)$, where tf_{ij} is the occurrence frequency of term t_i within D_j, and g_i is a global weight, which should be determined through the analysis of its occurrences in the training documents in Tr. A number of term weighting schemes can be found in [19].

Two well known global weight schemes used in text mining are inverse term frequency(itf) and inverse document frequency(idf) [28], which are defined as: $itf_i = \sqrt{\frac{1}{\sum_j tf_{ij}^2}} + 1$, and $idf_i = \log_2 \frac{N\_d}{df_i} + 1$, where df_i is the document frequency, i.e., the total number of documents in the document collection that contain term t_i; and N_d is the total number of documents in the training data set. When itf or idf is used as the global weight function, we have $g_i = itf_i$ or $g_i = idf_i$, respectively. However, based on our observation, important term words or their synonyms such as those identified by the physicians sometimes appear frequently in documents in a specific category. These two global weight functions do not give heavy weights to these types of words. For example, in our case study the classification criterion is "Whether or not the patient has permanent disabilities sustained from an accident", and the significant term words for classification would be "permanent", "permanency", "disability", as well as some of the negations such as "no", "not", etc. As a result, we developed the following entropy-based global weight function, E_GW.

1. For each term t_i on the term list **T_L**, calculate the proportion of the documents in Tr that contain t_i within N_C different categories, $p_{ij} = \frac{N\_c_{ij}}{N\_c_j}$, $j = 1, 2, \ldots, N\_C$, $\alpha_{ij} = \frac{p_{ij}}{\sum_{j=1}^{N\_C} p_{ij}}$, where N_c_{ij} is the number of documents within the jth categories that contains t_i, and N_c_j is the total number of documents in the jth category.
2. Calculate the entropy with respect to term t_i, $E_i = \sum_{j=1}^{N\_C} -\alpha_{ij} \log_2 \alpha_{ij}$. The entropy measure is a good indicator of how term t_i is distributed over different document categories. The higher the entropy, the less important item t_i is, since it is more evenly distributed among the document categories.

3. Calculate the global weight $E\_GW(i)$ for term t_i: $E\_GW(i) = 1 - \frac{E_i}{\log_2 N\_C}$, where N_C is the number of categories. This global weight function gives more weights to terms that have small entropy values.

We will show in Sect. 5 that the entropic based global weight function performs better than both, the inverse term frequency (*itf*) and inverse document frequency (*idf*) method.

4.3 SOM Learning

The Self Organizing Maps (SOM) was initially proposed by Kohonen et al. [23] as an unsupervised learning method for solving massive document collection problems [24]. It has been widely used in applications ranging from full text mining, financial data analysis, pattern recognition, image analysis, process monitoring and control to fault diagnosis [38]. A SOM is a two-dimensional lattice with M nodes. These nodes are usually arranged in a rectangular or hexagonal grid. A node u is represented by a weight vector, $\mathbf{W}_u$, which has the same dimension as an input vector. A SOM training algorithm continuously updates the weight vectors of each selected node and its neighboring nodes so that the map is organized in such a way that neighboring nodes on the grid have similar weight vectors. Effectiveness of SOM very much depends on the proper selection of SOM learning parameters, including map size M, lattice shape, learning algorithm, etc. [23].

In this article, we focus on training a SOM as a classifier to predict the categories of medical documents. The SOM classifier is obtained by applying the following learning algorithm to the training data Tr. Please note, the documents in Tr now are all represented in vectors in the form of the space model discussed in the last subsection. Let the weight vectors for all nodes at iteration Tr be represented by $\mathbf{W}(r) = \{\mathbf{W}_1(r), \ldots, \mathbf{W}_M(r)\}$.

1. Initialize weight vectors in $\mathbf{W}(0)$ for all nodes with randomly selected real numbers between 0 and 1, and initialize the radius of neighbors to a reasonable number, e.g. $\sigma(0) = \frac{1}{2}S$, where S is the width of the map.
2. Randomly choose a data sample from Tr, present it as an input vector $\mathbf{x}$.
3. Calculate the distance between $\mathbf{x}$ and every node in the map. We use the Euclidean distance to calculate the distance between the ith node's weight vector $\mathbf{W}_i(r)$ and the input vector $\mathbf{x}$: $\text{dist} = \|\mathbf{x} - \mathbf{W}_i(r)\|$.
4. The node has the smallest distance to the input vector is the "winner", denoted as BMU (Best Matching Unit).
5. For every node u within the radius $\sigma(r)$ of the BMU, adjust its weight vector $\mathbf{W}_u(r)$ according to the following learning rule:

$$\mathbf{W}_u(r+1) = \mathbf{W}_u(r) + \Theta_{uc}(r)L(r)(\mathbf{x} - \mathbf{W}_u(r)),$$

where $\Theta_{uc}(r) = \exp\left(-\frac{G_{uc}^2}{2\sigma^2(r)}\right)$, c is the BMU of $\mathbf{x}$, G is the distance between node u and c, $\sigma(r) = \sigma(0)\exp\left(-\frac{r\log_2\sigma(0)}{\Gamma}\right)$, $L(r) = L(0)(\frac{0.005}{L(0)})^{\frac{r}{\Gamma}}$ [45], $L(0)$ is an initial learning rate, which is set to 0.5, and Γ is the maximum number of iteration.

6. Increment r by 1, and goto step 2 until $r = \Gamma$ or the radius σ shrinks to 1.

After the SOM is trained, each training data sample is assigned to its BMU. These training data samples are then used in IME report classification, as described in Sect. 4.5.

4.4 SOM Model Selection

In order to make the above SOM learning algorithm effective, the following learning parameters should be carefully selected.

1. *Map size:* While there is no theoretical rule of for optimum map size M [5], there are quantitative indicators to help people determine the map size. Wang et al. [46] suggest using $M = 5\sqrt{|\mathrm{Tr}|}$ to use as the map size. However we will show in the empirical study section that this size is not effective.
2. *Map shape:* Kohonen introduced two types of SOM lattice shapes: hexagonal and rectangular. Because of the effect that different shapes may have on the neighborhood radius, these two can result in different organized maps, different quantitative error (QE) and topographic error (TE).
3. *Training algorithm:* There are two types of widely used training algorithms: sequential training and batch training [23]. The difference between these two is how often the weight vectors of the nodes are updated. The sequential training algorithm updates the weight vectors for every training sample presented to it. This is the training algorithm we used in the SOM learning algorithm presented above. A batch training algorithm updates the weights after the evaluation of all the training data.

Any combination of the learning parameters discussed above results a SOM model, and some models are better than others. We developed the following procedure for selecting an effective SOM model for classification.

1. For every combination of the learning parameters, namely, word frequency thresholds, map size, map shape, and the two types of training algorithms, use the SOM learning algorithm to generate a SOM model.
2. Evaluate these SOM models through a 3-fold cross validation process on the training data using the following performance criteria; QE, TE, and classification accuracy.
3. Output the best SOM based on the evaluation generated at Step 2 as the IME report classification system.

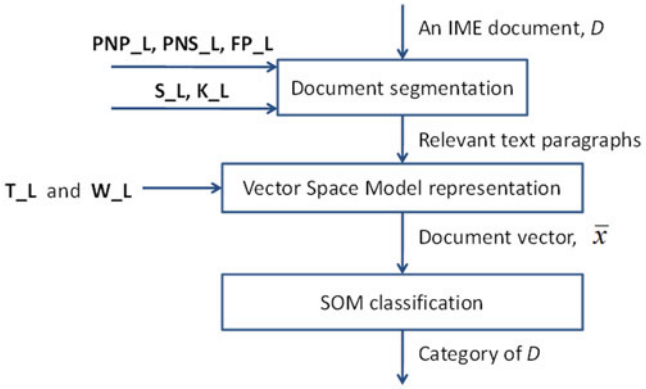


Fig. 5 C_IME_RPT: a system for IME report classification

The QE (quantitative error) and TE (topographic error) are two well-known criteria used in SOM evaluation 44. The QE is measured by the average distance between each training sample $\mathbf{x}_i$ and its best matching unit $\mathbf{B}_i$, $\text{QE} = \frac{1}{|Tr|}\sum_{i=1}^{|Tr|} \|\mathbf{x}_i - \mathbf{B}_i\|$. The TE is measured by the proportion of the training vectors such that their first and second BMU are not adjacent in the map, $TE = \frac{1}{|Tr|}\sum_{i=1}^{|Tr|} \xi(\mathbf{x}_i)$, where $\xi(\mathbf{x}_i) = 1$ if the top 2 matched nodes with $\mathbf{x}_i$ are not neighbors in the map, otherwise $\xi(\mathbf{x}_i) = 0$.

4.5 C_IME_RPT: An IME Report Classification System

An IME report classification system, C_IME_RPT, is developed based on the SOM model generated by the above machine learning process. Figure 5 illustrates computational steps in C_IME_RPT. When an IME document D is submitted to the system, it is first segmented by the document segmentation procedure described in Sect. 4.1. The document segmentation procedure uses the three name prefix and suffix lists generated by the process described in Sect. 3.2 to extract the body part in D, and the domain knowledge represented in subtitle list and the keyword list to extract the relevant paragraphs from the body part in D. The relevant text paragraphs are then represented in the document vector $\mathbf{x}$ based on the term list and the weight list obtained through the machine learning process described in Sect. 4.2. The SOM classification process consists of the following steps.

1. Find the best matching node N_c with $\mathbf{x}$.
2. If all the training documents belonging to N_c have the same category label, ρ, then the category of D is ρ.
3. If the training documents belonging to N_c have multiple category labels and there is a clear majority label, ρ, then the category of D is ρ.

4. If the training documents belonging to N_c have multiple category labels and there is no clear majority, then the label of the document in N_c that is closest to $\mathbf{x}$ is used as the category of D.

5 Empirical Case Study

In this section, we evaluate the algorithms presented in Sects. 3 and 4 through a case study. All training and test documents are IME reports of patients with orthopedic related ailments. Our task is to detect the name entities and encrypt them, and then classify the IME documents into three categories, "NP" representing "The patient has no permanent injuries sustained", "P" representing "The patient has permanent injuries sustained", and "NS" representing "The physician is not sure about whether the patient has permanent injuries or not".

5.1 Experiments Involving Name Entity Recognition

A set of 450 files of IME reports of patients with orthopedic related ailments have been collected. Each file contains several correspondences. These documents contain 10,309 people names and 1806 organization names. All the names have been labeled manually with different name entity categories for the purpose of training and evaluation processes.

Experiments were conducted using a 9-fold process; each fold uses a training data set of 50 documents and a testing set of 400 documents. In each fold we use the training data to generate the three reference lists, **PNP_L**, **PNS_L** and **FP_L**. In order to evaluate our system properly three experiments were conducted and the results are summarized in Table 1. In the first experiment, we applied the OpenNLP program to the test data in each of the 9-fold without any preprocessing. As shown in Table 1, this method gave very low precision in recognizing person and organization name entities, 49.87 and 68.62 % respectively. In the second experiment, we applied our HBE segmentation algorithm to each test document, and then applied the OpenNLP to the heading, body and ending parts separately. The segmentation algorithm, HEB, has helped reduce false positives significantly, and also increased the name entity recognition accuracy. As a result, the precision has increased to 97.33 % and the recall to 77.29 % for people name entity. The precision for organization name entities is also boosted up to 90.26 %. In the third experiment we applied the proposed MD_NER_NCL system illustrated in Fig. 1 to the test data. The recognition precision on people name entities has increased to 98.05 % and the recall to 91.08 %, which is a significant improvement over the other two methods. This performance is also substantially better than the published work on name entity detection [6, 16, 34]. More statistics are shown in Tables 2 and 3. Table 2 shows the examples of prefixes extracted from the training data in one of the nine folds. Table 3 shows the

Table 1 NER performances based on a 9-fold cross-validation process

Approaches	Entity type	Precision (%)	Recall (%)	F1 measure (%)
OpenNLP only	People name Organizations	49.87 68.62	67.67 73.54	57.42 71.00
HBE segmentation + OpenNLP	People name Organizations	97.33 90.26	77.29 73.22	86.16 80.85
MD_NER_NCL	People name Organizations	98.05 90.26	91.08 73.22	94.44 80.85

Table 2 Examples of prefixes on PNP_L and FP_L

Positive name prefix	Attention:, ATTN:, Dr., Drs., Mr., Mrs., Ms., RE:
False name prefix	PMH:, S., SYMPTOMS:, Mt.

Table 3 Detailed performance of three NER methods

Approaches	Entity type	All	Correct	Partial	Missed	FP
OpenNLP only	People name Organizations	9601 1753	6474 1288	949 168	2178 297	6641 586
HBE Segmentation + OpenNLP	People name Organizations	9601 1753	7418 1282	725 174	1458 297	178 140
MD_NER_NCL	People name Organizations	9601 1753	8738 1282	725 174	138 297	178 140
Approaches	Entity type	Precision (%)	Recall (%)	F1 measure (%)		
OpenNLP only	People name Organizations	49.36 68.73	67.43 73.47	57.00 71.02		
HBE Segmentation + OpenNLP	People name Organizations	97.65 90.15	77.26 73.13	86.27 80.75		
MD_NER_NCL	People name Organizations	98.00 90.15	91.01 73.13	94.38 80.75		

statistics of correctly recognized, partially recognized, missed and false positive (FP) name entities, and three performance measures: precision, recall and F1-measures. The proposed algorithm, MD_NER_NCL gave significantly better performances in recognizing people name entities over the other two methods. For the organization name entities, both MD_NER_NCL and HBE Segmentation + OpenNLB gave much improved performances over the OpenNLP method.

5.2 Experiments Involving IME Report Classification

We randomly sampled 495 IME documents from a collection of over 8000 documents to evaluate the IME report classification system. In these 495 documents, 453 documents belong to the "NP" category, 21 documents belong to the "NS" category and 21 documents belong to the "P" category. The data were partitioned into training

Table 4 Evaluating word frequency thresholds

Fold - 1										
Frequency threshold	1	2	3	4	5	6	7	8	9	10
QE	1.46	1.36	1.25	1.18	1.1	1.07	1.05	1	0.97	0.95
TE	0	0	0.01	0.01	0	0.01	0	0.01	0	0.01
Testing accuracy (%)	85.7	84.4	83.7	85.9	84.1	86.6	83	88.5	89.4	89
Fold - 2										
Frequency threshold	1	2	3	4	5	6	7	8	9	10
QE	1.54	1.39	1.25	1.15	1.07	1	0.97	0.95	0.94	0.93
TE	0	0.01	0.01	0.01	0	0	0.01	0	0	0
Testing accuracy (%)	87.1	84	86.5	84.8	85.2	82.6	83.5	86.8	89.8	87.4
Fold - 3										
Frequency threshold	1	2	3	4	5	6	7	8	9	10
QE	1.47	1.32	1.2	1.12	1.05	0.99	0.95	0.92	0.9	0.89
TE	0	0	0.01	0.01	0	0.01	0	0.01	0	0.01
Testing accuracy (%)	84.4	85.8	85	82.1	84.8	85.5	81.9	87.1	90.4	89.3

and test sets through random sampling, and experiment is conducted using a procedure of 3-fold cross validation. Each fold uses a training data set of 330 documents and a testing set of 165 documents. In this empirical study, we focus on the study of SOM learning parameters, and the two critical factors in the vector space model, term frequency threshold and the global weight function. The SOM learning parameters being studied include map size, map shape, and the type of training algorithms. We also conducted a comparative study on the proposed SOM classification system and the six well-known text classification systems, K-NN, *K*-Means, Hierarchical *K*-Means, Nnaïve Bayesian, Random Forest, and SVM.

5.2.1 Evaluation of Vector Space Models and SOM Learning Parameters

A term frequency threshold is used to remove words that are used often and not uniquely important to individual categories. To evaluate the term frequency thresholds, we used the batch training algorithm, a hexagonal lattice with map size 20×16, which is approximately four times as big as the map size suggested in [46]. Table 4 presents the experiment results on three folds generated by C_MED_RPT using term frequency thresholds 1 through 10. Our criterion to select the best model is to minimize the QE and TE, and maximize the testing accuracy. It appears that the C_MED_RPT system used term frequency threshold of 9 gave the best overall performances: high in accuracy and low in QE and TE, as shown in Figs. 6 and 7.

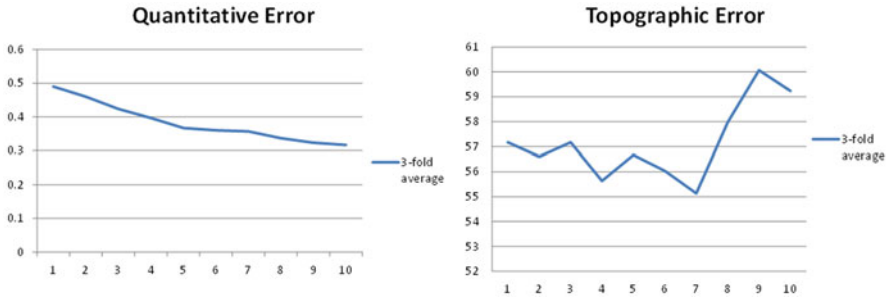


Fig. 6 TE and QE on different word frequency thresholds

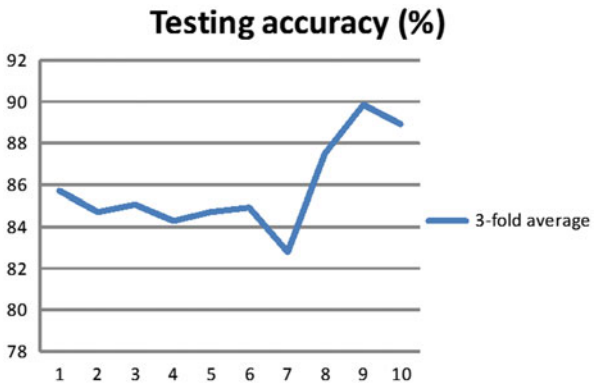


Fig. 7 Classification accuracy on different word frequency thresholds

Four different global weight functions that used in the vector space model were evaluated: binary, itf, idf, and E_GW. The *binary* function is defined as:

$$bin_{ij} = \begin{cases} 1, & tf_{ij} > 0 \\ 0, & otherwise. \end{cases}$$

where tf_{ij} is the occurrence frequency of term t_i in IME report category j. Figure 8 presents the classification accuracies of these four weighting functions used in the C_MED_RPT system. The proposed global weight scheme, E_GW, yielded the highest (89.9 %) 3-fold average classification accuracy.

The next step in the model selection procedure is to use the selected vector space model to evaluate the SOM learning parameters, map size, lattice shape and training algorithm. Three map sizes were evaluated, large, medium and small. The large map size is chosen as $a \times b$, where the product of a and b is approximately chosen as $20\sqrt{|\mathrm{Tr}|}$ and the ratio of a and b should be approximately 1.25, and Tr denotes the set of training data. Since the training data size used in the experiment is 330, a map of 20×16 is used as the large size. The medium size map is $a/2 \times b/2$, which is 10×8, and small is $a/4 \times b/4$, which is 5×4. Two map shapes were evaluated, hexagonal, denoted as "Hexa," and rectangle, denoted as "Rect." Two

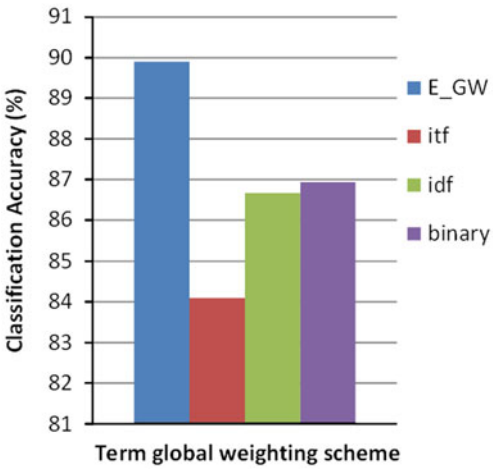


Fig. 8 Classification accuracy generated using different global weight schemes

training algorithms were used in the evaluation, batch and sequential. The results are presented in Table 5. It appears that the C_MED_RPT systems used large SOM map size are the best according to the three criteria. Note that the medium size map used in the experiment matches the map size $M = 5\sqrt{|Tr|}$ suggested by other researchers [46]. In this application, this map size does not appear to be effective. As for the map shape, the experiment results showed that C_MED_RPT systems used the SOMs with hexagonal shape generated less topographical error than the SOMs with rectangular shape. The batch learning algorithm generated less quantitative errors than the sequential learning algorithm. By looking at all three criteria, i.e., QE, TE and the validation accuracy, the best combination of the SOM learning parameters is: the large map, a hexagonal shape and the batch learning algorithm. These SOM learning parameters were used in training the final IME report classification system, C_MED_RPT system, which was used in the comparative study presented in the next subsection.

5.2.2 Comparative Study with Other Classification Algorithms

In this section, we compare the performance of the C_MED_RPT system with the six well-known text document classification systems, K-nearest neighbor (K-NN) [29], K-means algorithm [9], a hierarchical K-means [1], naïve Bayesian [17], RF(Random Forest) [26] and SVM [14]. K-means and hierarchical K-means are used to generate clusters of documents with similar feature vectors through an unsupervised learning. Each cluster is then labeled with the specific category based on a majority vote.

For the K-NN algorithm, since the training data contain many "NP" documents (over 90 %), it is prudent to use a smaller K value in the K-NN algorithm to avoid too many "NP" labeled documents falling into the top K closest documents. As a result we evaluated the performances of K-NN with $K = 1$ through 10. Figure 9 presents the classification accuracy for each category on testing set. It shows systems with K

Table 5 Evaluating SOM parameters and training algorithms

Map size	Shape	Training algorithm	QE	TE	Testing accuracy (%)
large	Hexa	Batch	1.105	0.003	89.9
medium	Hexa	Batch	1.396	0.022	88.76
small	Hexa	Batch	1.651	0.065	87.79
large	Rect	Seq	1.238	0.028	87.76
medium	Rect	Seq	1.489	0.033	88.05
small	Rect	Seq	1.715	0.047	86.53
large	Rect	Batch	1.089	0.016	87.87
medium	Rect	Batch	1.403	0.059	87.09
small	Rect	Batch	1.695	0.047	85.44
large	Hexa	Seq	1.212	0.005	88.12
medium	Hexa	Seq	1.502	0.008	89.10
small	Hexa	Seq	1.619	0.015	87.92

= 3–10 completely missed the “P” category. For the “NS” category, all achieved less than 15 % recognition rate. Systems with K = 8–10 missed “P” and “NS” categories completely.

The classification results of the other five classifiers along with the proposed SOM classification system are presented in Fig. 10. The K-means algorithm only has 4.76 % accuracy on “P” category, which is the most important category for this application. The Hierarchical K-means did even worse; it has 4.76 % accuracy on both “P” and “NS” categories. The Naïve Bayesian missed the “P” category completely. The RF system was able to recognize more than 23 % of the documents in the “P” category, and more than 19 % in the “NS” category. The SVM is better than all previous classifiers in terms of recognize “P” and “NS” documents, while having slightly lower accuracy on “NP” documents. The C_MED_RPT system gave the best overall classification results: it classified correctly 52.38 % of the documents in“P”, more than 57.14 % in the “NS” categories and over 93 % of the documents in the “NP” category.

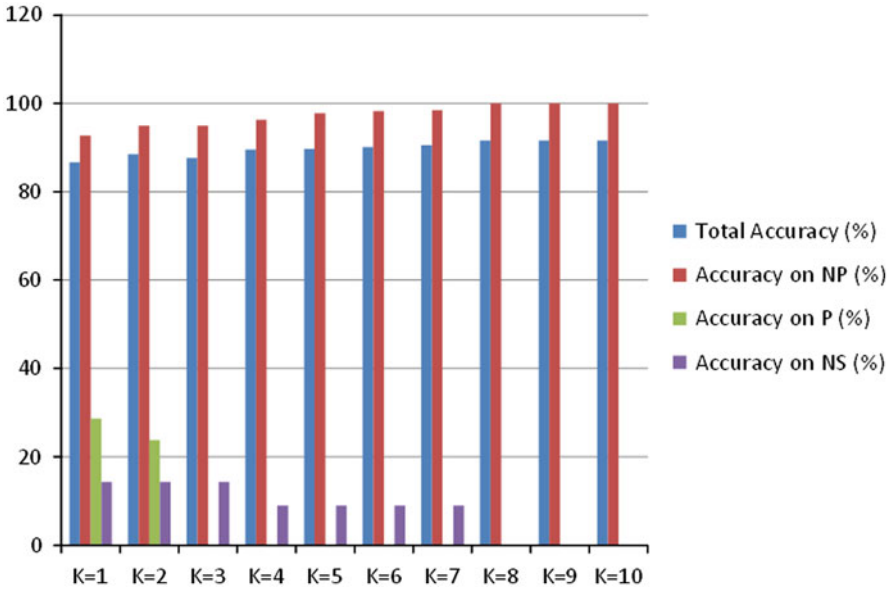


Fig. 9 Classification accuracy of *K*-NN

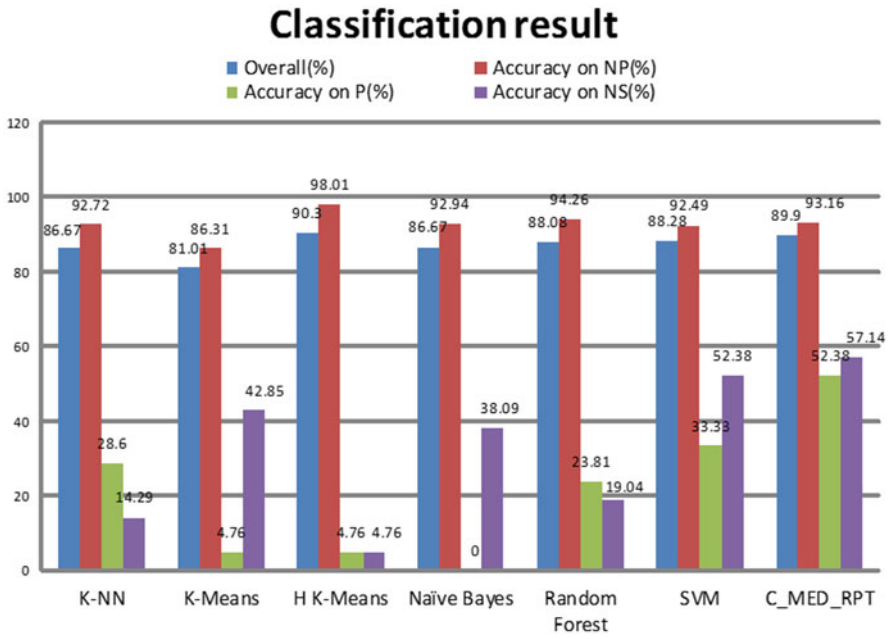


Fig. 10 Classification accuracy of the C_MED_RPT and five other text classification and clustering systems

6 Conclusion

This article presents two medical IME report processing systems, MD_NER_NCL for name entity recognition, and C_MED_RPT system for classifying IME reports, and the machine learning processes used to train the two systems. The MD_NER_NCL system consists of a document segmentation process (HBE segmentation algorithm), and a statistical reasoning process. C_MED_RPT system consists of HBE document segmentation, vector space modeling and SOM classification. The C_MED_RPT system is trained through a machine learning process, which includes the generation of an effective vector space model, SOM learning and model selection. We evaluated the two systems through a case study of the IME reports of the patients with orthopedic related ailments. Based on the results generated by the case study, we can conclude that:

- MD_NER_NCL system is effective in detecting name entities in correspondence documents. Its performance is significantly superior over the OpenNLP software: MD_NER_NCL yielded 98.05 % in precision and 91.08 % in recall, while the OpenNLP process only 49.87 % in precision and 67.67 % in recall for people name recognition on the same case study data.
- The machine learning process used to train the classification system, C_MED_RPT, is effective. The machine learning process provides an automatic approach to train a document classification system that is capable of accurately categorizing input documents. The evaluation results generated in the case study show that the C_MED_RPT system gave the best overall performance in comparison with the six other classification system. More importantly it is less sensitive to unbalanced data. In the case study, the C_MED_RPT system was able to classify the IME reports in the minority classes, i.e., the "P" and "NS" categories much more accurately than any of the other six classifiers.

References

1. Arai, K., Barakbah, A.: Hierarchical k-means: an algorithm for centroids initialization for *k*-means. Rep. Fac. Sci. Eng. **36**(1), 25–31 (2007)
2. Basu, A., Walters, C., Shepherd, M.: Support vector machines for text categorization. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, Los Alamitos, California, USA, 2003, pp. 7– IEEE (2003)
3. Bender, O., Och, F., Ney, H.: Maximum entropy models for named entity recognition. In: Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 148–151, Edmonton, Canada. Association for Computational Linguistics (2003)
4. Benkhalifa, M., Bensaid, A., Mouradi, A.: Text categorization using the semi-supervised fuzzy c-means algorithm. In: Fuzzy Information Processing Society, 1999, NAFIPS. 18th International Conference of the North American. pp. 561–565, New York, USA. IEEE (1999)
5. Céréghino, R., Park, Y.: Review of the self-organizing map (som) approach in water resources: commentary. Environ. Model. Softw. **24**(8), 945–947 (2009)
6. Chang, Y., Sung, Y.: Applying name entity recognition to informal text. Recall **1**, 1 (2005)

7. Chen, Z., Ni, C., Murphey, Y.L.: Neural network approaches for text document categorization. In: IEEE International Joint Conference on Neural Networks, Vancouver, BC, Canada (2006)
8. Chen, Z., Huang, L., Murphey, Y.L.: Incremental neural learning for text document classification. In: International Joint Conference on Neural Networks, Orlando, Florida, USA (2007)
9. Cheung, Y.: *k**-means: a new generalized *k*-means clustering algorithm. Pattern Recognit. Lett. **24**(15), 2883–2893 (2003)
10. Chieu, H., Ng, H.: Named entity recognition: a maximum entropy approach using global information. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, pp. 1–7. Association for Computational Linguistics, Taipei, Taiwan (2002)
11. Cios, K., William Moore, G.: Uniqueness of medical data mining. Artif. Intell. Med. **26**(1), 1–24 (2002)
12. Claster, W., Shanmuganathan, S., Ghotbi, N.: Text mining of medical records for radiodiagnostic decision-making. J. Comput. **3**(1), 1–6 (2008)
13. Collier, N., Nazarenko, A., Baud, R., Ruch, P.: Recent advances in natural language processing for biomedical applications. Int. J. Med. Inform. **75**(6), 413–417 (2006)
14. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learn. **20**(3), 273–297 (1995)
15. Farkas, J.: Generating document clusters using thesauri and neural networks. In: Canadian Conference on Electrical and Computer Engineering, 1994, Conference Proceedings 1994, pp. 710–713, New York, NY, USA. IEEE (1994)
16. Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named entity recognition through classifier combination. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, vol. 4, pp. 168–171. Association for Computational Linguistics, Edmonton, Canada (2003)
17. Ho, T.: Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995, vol. 1, pp. 278–282, Montreal, Canada. IEEE (1995)
18. Holzinger, A., Geierhofer, R., Mödritscher, F., Tatzl, R.: Semantic information in medical information systems: Utilization of text mining techniques to analyze medical diagnoses. J. Univ. Comput. Sci. **14**(22), 3781–3795 (2008)
19. Huang, L., Murphey, Y.: Text mining with application to engineering diagnostics. Advances in Applied Artificial Intelligence, pp. 1309–1317 (2006)
20. Huang, Y., Seliya, N., Murphey, Y.L., Friedenthal, R.B.: Named entity recognition and classification in medical text documents. In: The 5th International Conference on Data Mining, Las Vegas, Nevada, USA (2009)
21. Hyotyniemi, H., et al.: Text document classification with self-organizing maps. STeP'96, Genes, Nets and Symbols, pp. 64–72 (1996)
22. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. Machine Learning: ECML-98, pp. 137–142, Chemnitz, Germany (1998)
23. Kohonen, T.: Self-organizing maps, vol. 30. Springer, Berlin, Germany (2001)
24. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection. IEEE Trans. Neur. Netw. **11**(3), 574–585 (2000)
25. Lam, W., Low, K.: Automatic document classification based on probabilistic reasoning: Model and performance analysis. In: Systems, Man, and Cybernetics, 1997. IEEE International Conference on Computational Cybernetics and Simulation, 1997, vol. **3**, pp. 2719–2723. IEEE (1997)
26. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: Proceedings of the National Conference on Artificial Intelligence, pp. 223–223, Menlo Park, CA, USA. Wiley (1992)
27. Lee, D., Chuang, H., Seamons, K.: Document ranking and the vector-space model. IEEE Softw. **14**(2), 67–75 (1997)
28. Luhn, H.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)

29. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. California, USA (1967)
30. Makoto, I., Takenobu, T.: Hierarchical bayesian clustering for automatic text classification. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95), Montreal, Quebec, Canada (1995)
31. Manine, A., Alphonse, E., Bessières, P.: Learning ontological rules to extract multiple relations of genic interactions from text. Int. J. Med. Inform. **78**(12), e31–e38 (2009)
32. Marcus, M., Marcinkiewicz, M., Santorini, B.: Building a large annotated corpus of english: The penn treebank. Comput. Linguist. **19**(2), 313–330 (1993)
33. Mayfield, J., McNamee, P., Piatko, C.: Named entity recognition using hundreds of thousands of features. In: Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 184–187. Association for Computational Linguistics (2003)
34. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 188–191, Edmonton, Canada. Association for Computational Linguistics (2003)
35. Merkl, D.: Text classification with self-organizing maps: Some lessons learned. Neurocomputing **21**(1), 61–77 (1998)
36. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em algorithm. Machine Learn. **39**(2), 103–134 (2000)
37. Ou, G., Murphey, Y.L., Feldkamp, L.: Multicategory pattern classification using neural networks. In: International Conference on Pattern Recognition, Cambridge, UK (2004)
38. Pölzlbauer, G.: Survey and comparison of quality measures for self-organizing maps. In: 5th Workshop on Data Analysis (WDA 2004), pp. 67–82 2004
39. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
40. Soderland, S.: Learning information extraction rules for semi-structured and free text. Machine Learn. **34**(1), 233–272 (1999)
41. Soderland, S., Aronow, D., Fisher, D., Aseltine, J., Lehnert, W.: Machine learning of text analysis rules for clinical records. TE-39: University of Massachusetts, Center for Intelligent Information Retrieval Technical Report (1995)
42. Svingen, B.: Using genetic programming for document classification. Diane J. Cook (1998)
43. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 173–180. Association for Computational Linguistics (2003)
44. Uriarte, E., Martín, F.: Topology preservation in SOM. Int. J. Appl. Math. Comput. Sci. **1**(1), 19–22 (2005)
45. Vesanto, J., et al.: Technical report on SOM toolbox 2.0. Espoo, Finland (2000)
46. Wang, J., Delabie, J., Aasheim, H., Smeland, E., Myklebost, O.: Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study. BMC Bioinform. **3**(1), 36 (2002)
47. Zhou, X., Han, H., Chankai, I., Prestrud, A., Brooks, A.: Converting semi-structured clinical medical records into information and knowledge. In: 21st International Conference on Data Engineering Workshops, 2005, pp. 1162–1162, Tokyo, Japan. IEEE (2005)

Part V
Engineering Tasks

Data Mining Vortex Cores Concurrent with Computational Fluid Dynamics Simulations

Clifton Mortensen, Steve Gorrell, Robert Woodley and Michael Gosnell

Abstract Computational fluid dynamics (CFD) simulations present a variety of data mining challenges. At the forefront, CFD computations can require weeks of computation on expensive high performance clusters, delaying investigation of results until a fully converged solution is obtained. Also, advanced modeling can create large data sets that risk concealing rather than revealing useful flow information. 21st Century Systems, Inc. and Brigham Young University have been collaborating on a concurrent agent-enabled feature extraction project designed to provide intelligent feedback to researchers while a CFD simulation is executing. This approach can extract flow features while a simulation is running and then project their expected probability for a complete simulation. This article gives a detailed outline of our approach and then shows the results of our implemented approach on two sample CFD data sets. The results show vortex core features can be successfully extracted while a simulation is running and provide information as much as 50 % earlier than waiting for complete simulation convergence.

1 Introduction

Data mining is quickly becoming recognized as an important tool to analyze large data sets generated by high-fidelity computational fluid dynamics (CFD) simulations. CFD simulations numerically solve the governing equations of fluid motion. Some examples are simulations of ocean currents, atmospheric turbulence, combustion, aircraft, rotorcraft, and ship hydrodynamics. Very large time-accurate,

C. Mortensen (✉) · S. Gorrell
Brigham Young University, Provo UT 84604, USA
e-mail: clifton.mortensen@engineering.ucla.edu

S. Gorrell
e-mail: sgorrell@byu.edu

R. Woodley · M. Gosnell
21st Century Systems, Inc., 6825 Pine Street Suite 141, Omaha NE 68106, USA
e-mail: robert.woodley@21csi.com

M. Gosnell
e-mail: mike.gosnell@21csi.com

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_15

three-dimensional computational models risk concealing rather than revealing the physics of interest.

The use of parallel codes and supercomputers has allowed CFD simulations to increase in grid resolution and numerical accuracy to a point of correctly simulating highly complex fluid flow problems. Many of these advanced simulations are run on multi-node computing clusters requiring many weeks to reach full convergence. List et al. [13] and Yao et al. [20] have run unsteady Reynolds-averaged Navier-Stokes (URANS) simulations of gas turbine engine transonic fan stages with 166 million grid points and entire fans with over 300 million grid points respectively. These types of simulations typically run on a thousand or more processors, take hundreds of thousands of CPU-hours on expensive computing clusters to obtain converged solutions, and generate terabytes of raw data.

The time to analyze massive CFD data sets can be equivalent to the wall time of computing the solution—sometimes hundreds of hours. To post-process large data sets, there are a variety of software programs and techniques which can be quite discipline dependent. One approach is simply to slowly sift through data to find useful information based on intuition and previous experience. Other approaches spanning many disciplines utilize software concepts and packages such as Evita [19], Intelligent Light's FieldView [7], and Kitware's ParaView [11]. These types of programs are meant to post-process and visualize massive data sets and commonly include techniques such as feature extraction, construction of iso-surfaces, and automated visualization.

With such massive simulation sizes and analysis requirements, there is an increasing burden to effectively mine the data. One attractive new method which offers the potential to save vast amounts of resources are technologies to automatically and intelligently mine CFD data concurrently with the evolving simulation. The risk is that before traditional convergence, features may not exist or conform to their accepted mathematical definitions. However, the tradeoff is that if certain features could be detected with appropriate levels of confidence, the CFD researcher might be able to obtain enough information to forego the continuation of the solution, saving CPU-hours and allowing for new design approaches to be considered much earlier than if the solution had run to convergence.

The Concurrent Agent-Enabled Feature Extraction (CAFÉ) concept was proposed to provide such a mechanism and is being developed to aid a CFD researcher in effectively dealing with mining features in massive partially-converged and completely converged CFD simulations. This approach is unique due to the transient nature of the solution space—attempting to mine relevant CFD features essentially before they appear. Figure 1 shows a conceptual view of the CAFÉ concept, trading off additional expense of concurrent feature detection with potential benefit of not requiring the CFD simulation to completely converge before items of interest are identified.

This article shows that extracting flow features from CFD data sets before a simulation has reached full numerical convergence is possible and can be done early enough in the simulation to be of use. This article also gives a methodology for obtaining information from the extracted features. When a feature has been extracted before simulation convergence, it is possible to find the expected probability of

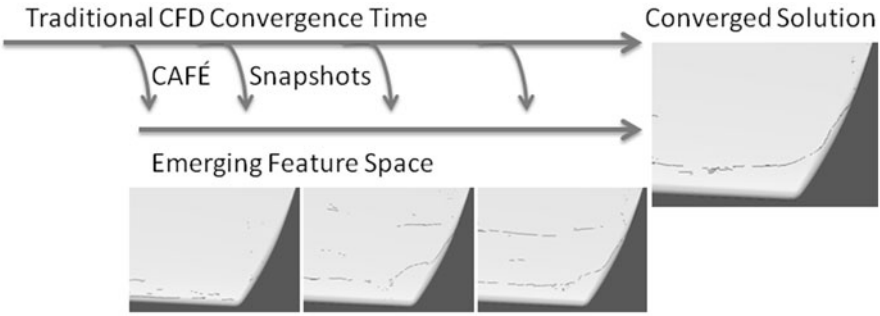


Fig. 1 CAFÉ concept showing concurrent feature mining

whether or not that feature will exist when a simulation is complete. The methodology could potentially be used in many physics based numerical simulations that have features extracted from primitive variables. A researcher can be shown the expected probability of these features as a simulation progresses rather than *only* when a simulation reaches numerical convergence.

Presented here is real world experience implementing the CAFÉ concept with respect to detecting vortex cores. Section 2 introduces CAFÉ and describes the relevant architectural components. Section 3 describes implementation of vortex core feature extraction and reasoning within the CAFÉ architecture. Results of vortex core feature extraction on blunt fin and delta wing test cases is addressed in Sect. 4 with concluding remarks in Sect. 5.

2 CAFÉ Overview

Ultimately, the goal of CAFÉ is to assist the researcher in performing CFD simulations. The entire scope of CAFÉ extends beyond the scope of this research, spanning from CFD pre-processing, concurrent feature extraction and analysis, through to solution post-processing. The approach gains innovation as the solution was framed around an agent-based structure designed for decision support software applications. This structure allowed all aspects of the CFD solution process to be included within the scope of CAFÉ, with the following high-level goals:

1. Provide concurrent feature extraction
2. Provide intelligent reasoning about extracted features
   (a) Be able to incorporate multiple feature extraction algorithms
   (b) Determine the believability of features
3. Utilize detected features and results
   (a) Hone future search space to reduce resource waste
   (b) Incorporate machine learning to generalize solutions and provide more intelligent initial conditions

Goal 3 focuses primarily on CFD pre- and post-processing aspects and is beyond the realm of this discussion. Goal 1 focuses on one of the two main elements of this investigation, concurrent feature extraction. Typical CFD simulations largely ignore the iterative solution data occurring before the required convergence. Concurrent aspects of CAFÉ utilize some of this intermediate data for analysis which is investigated while the CFD simulation continues toward a solution. Unlike final post-processing and analysis, CAFÉ must be able to make decisions on the feature extractions without the assistance of user input. This aspect is addressed in Goal 2 which utilizes results from multiple feature extraction algorithms along with knowledge of the solution space to provide intelligence about the detected features.

By providing rules for extraction and a mathematically rigorous method for evaluating the uncertainty in feature extraction, CAFÉ attempts to determine when a feature is true or simply an artifact of an unconverged simulation. An agent-based architecture, based on decision support software, allows for these capabilities as addressed in the following sections.

2.1 The CAFÉ Architecture

As mentioned previously, CAFÉ spans the CFD domain including pre-processing and post-processing aspects. However, the focus of this study is on the concurrent aspects and associated empirical findings. As such, the complete CAFÉ architecture is omitted here, with additional details available in [14, 15]. The relevant aspects of the CAFÉ architecture for this work are presented in Fig. 2, showing the information flow from the concurrent data extraction through processing and decision support presentation. Initially from the raw data, agents running feature extraction algorithms identify possible features. These individual feature analyses are aggregated and evaluated at the feature aggregation level, with the capability to analyze multiple features as well as multiple analyses of the same feature. Finally, the feature opinions from the aggregation level are collected and incorporated within decision support agents to provide the desired analysis to the CFD researcher. Each of these component areas is discussed below.

2.2 Feature Extraction

CAFÉ's design allows for incorporation of multiple algorithms for any given feature of interest. Furthermore, the architecture is extensible to include additional features beyond what is initially implemented. Initial CAFÉ development has implemented three basic flow features: vortices, shock waves, and separation and attachment lines. This work concentrates on vortices.

Vortices are common occurrences in many types of engineering flows. They arise where there are large amounts of vorticity, or flow rotation. A vortex contains two

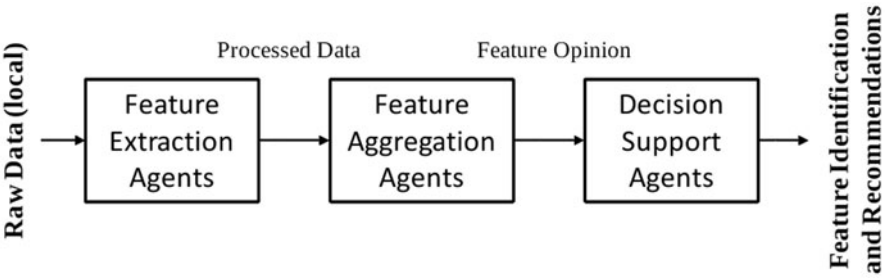


Fig. 2 Information flow diagram for CAFÉ

interdependent parts: the vortex core line and the swirling fluid motion around the core. Many feature extraction algorithms have been developed to locate vortex core lines. Unfortunately, when extracting vortex core lines, there is not one markedly superior algorithm that correctly extracts all features within the spatiotemporal flow domain. Rather, there are multiple algorithms per feature that have been optimized for specific flow conditions. Roth [16] states that

> none of the [vortex extraction] methods is clearly superior in all the tested data sets.

This leaves a researcher with the significant problem of having to run a data set through multiple extraction algorithms and parse through the data output to find relevant features. This is also where CAFÉ's feature aggregation will come into play as discussed in the next section.

The initial CAFÉ work has implemented two vortex core extraction algorithms. The first vortex extraction algorithm selected is the Sujudi-Haimes (SH) algorithm [18]. The SH algorithm was designed as a robust vortex core line detection algorithm for use in large 3D transient problems. It is commonly used in CFD post-processing software packages such as EnSight 9 [2] and pV3 [5]. The second vortex core extraction algorithm is the Roth-Peikert (RP) algorithm [16, 17]. The RP algorithm is specifically designed to extract fluid vortices in turbomachine simulations. What makes the RP algorithm unique and well suited for complex flow fields is the fact that it is designed to locate curved rather than straight vortex core lines. Each algorithm is strongly suited to a different domain.

All feature extraction algorithms implemented in CAFÉ have the same basic composition: perform computations on subsets of the flow domain, apply some filtering mechanism, and aggregate the remaining selected regions into features. This structure has two immediate consequences. First, since the agents operate on the feature before it has been aggregated, the parts of a feature with a high believability can easily be selected while disregarding other parts with a low believability. Second, it allows a combination of features that have been extracted by multiple algorithms. In other words, if nodes are extracted close together by two separate extraction algorithms, they can be operated on by a feature aggregation agent and then combined into one distinct feature rather than two disjoint features. This modular approach

also creates the framework for including additional feature extraction algorithms as desired.

2.3 Feature Aggregation

One of the key functionalities of CAFÉ is the ability to identify features and reason with combinations of features. As mentioned above, certain feature extraction algorithms work better than others for a given situation. Additionally, since CAFÉ is trying to perform concurrent extraction, some extracted features may actually be incorrect. As a result, CAFÉ needed a way to select features according to the believability of the feature based on when in the simulation the feature occurs, what conditions the feature is extracted under, and if previous iterations contain the same feature.

The tool employed to quantify the believability of a feature is encapsulated within subjective logic developed by Jøsang [8, 10]. This ternary logic captures belief (b), disbelief (d), and uncertainty (u) as an opinion, and intrinsically handles these in an algebraic space. These three elements are defined in [8] along with relative atomicity a to form an opinion or belief tuple ω with a belief about x as shown in Eq. (1). Relative atomicity is used to give an a priori weight to a system's uncertainty. The common assumption of $a = 0.5$ is used here:

$$\omega_x = (b(x),\ d(x),\ u(x),\ a(x))\,. \tag{1}$$

To form an opinion, each component of the belief tuple is given a numerical value, allowing the opinion to have an exact representation. To maintain uniformity and provide for mathematical constructs, the summation of an opinion's belief, disbelief, and uncertainty components is always equal to one as in Eq. (2):

$$b + d + u = 1. \tag{2}$$

Furthermore, belief, disbelief, and uncertainty can only take on values between 0 and 1. These basic prerequisites provide much of the framework necessary for working with opinions in a mathematically rigorous fashion. Adhering to these fundamentals, subjective logic provides additional resources for incorporating opinions within a reasoning framework referred to as a trust network. For example, Fig. 3 shows a simple trust network where individual A has trust in individual B, but does not have an opinion about C. Individual B trusts C with some opinion denoted $\omega_{\mathrm{C}}^{\mathrm{B}}$ and can then 'refer' C to A, thus giving A derived trust in C. In the trust network, individuals are sometimes called 'agents' and the means by which trust is quantitatively transferred between agents is subjective logic.

Subjective logic is extremely attractive for incorporating the inherent uncertainty present during CFD execution as opinions are not forced to identify belief or disbelief. In other words, opinions are not strictly forced one way or another in the presence of uncertainty. An agent can find, based on given information, how probable an outcome

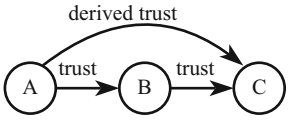


Fig. 3 Simple trust network showing A's derived trust in C from B

is rather than simply reducing the outcome to a binary TRUE or FALSE. In addition to the initial opinion formulation for feature detection, missing or incomplete data can also be incorporated within subjective logic by adjusting belief, disbelief, and uncertainty accordingly.

Another reasoning mechanism commonly used in data mining applications is fuzzy logic so it's worth highlighting the differences between fuzzy logic and subjective logic. A fuzzy logic approach incorporates multiple fuzzy categories which are typically partially overlapping. The goal measures addressed with fuzzy logic are typically crisp, such as representing a precisely measurable temperature through fuzzy states of hot, warm, or cold. In subjective logic, opinions constitute a crisp belief frame with mutually exclusive states. The opinion measures as a whole express uncertainty, thus the opinions themselves are "fuzzy" within a subjective logic belief frame. Since the desire is to express the presence or absence of features under some uncertainty, these two mutually exclusive properties map naturally within the subjective logic context and seemed the most natural choice for development.

Once initial opinions of features are formulated adherent to the belief tuples of subjective logic, two key operators are utilized to formulate trust networks: the discounting operator and the consensus operator. The discounting operator is used when agents in a trust network lie along the same path as in Fig. 3. The discounting operator is defined by Jøsang [8], and uses the symbol $\otimes$ giving

$$\omega_x^{AB} = \omega_B^A \otimes \omega_x^B, \tag{3}$$

where the superscripts represent an agent having the trust and the subscripts represent an agent, or piece of information, on which the trust is based. Conceptually, the discounting of opinions allows individual, independent beliefs to be transferred along a chain of agents. The counterpart to the discounting operator is the consensus operator. The consensus operator is used when multiple opinions are held about the same agent, or piece of information, and a single opinion is desired. The consensus operator is defined by Jøsang [9], and uses the symbol $\oplus$ giving

$$\omega_x^{AB} = \omega_x^A \oplus \omega_x^B, \tag{4}$$

which follows the same syntax as the discounting operator. With supporting opinions, the consensus operator has the effect of reducing uncertainty.

Subjective logic attempts to remove strict notions of TRUE and FALSE. Thus, instead of specifically stating if a feature is present, an opinion of a detected CFD feature can express if a feature has a high expected probability of occurring. When evaluating an opinion, Jøsang [9] provides probability expectation (E) to give the expected probability of an outcome and is calculated from

$$E = b + au. \tag{5}$$

The probability expectation identifies what an agent expects the probability to be and is not an exact measure of probability. However, there exist mappings which allow opinions to be expressed as probabilistic distributions. For a more thorough analysis of subjective logic and its capabilities, see [9].

2.4 Decision Support

With an overarching goal of providing decision support to the CFD user, CAFÉ utilizes the feature extraction algorithms along with feature aggregation capabilities utilizing subjective logic opinions and the trust network framework. These key components allow for extracting features concurrent with CFD simulations and provide intelligent analysis of the feature space prior to convergence. While early feature extraction may contain large variations in the solution space, multiple sets of the solution space, taken many iterations apart, can also be incorporated within the trust network. As will be addressed within the context of the vortex core extraction in the following sections, these components aid in overall CFD analysis with CAFÉ, providing information on the identified features, what filters to use to better visualize the feature, comparisons of detected features (size, strength, etc.), and do so concurrently with the simulation.

3 Vortex Core Extraction Method

3.1 Trust Network for Vortex Core Extraction

A graphical representation of the vortex core extraction trust network is shown in Fig. 4. The algorithm agent (AA) contains actual feature extraction algorithms with subscripts 1 and 2 denoting separate algorithms. The master agent (MA) combines information from multiple AAs to form its opinion. R refers to a grid point contained in the extracted core line under inspection by the agents to find whether or not the vortex core is probable. The end goal is for the MA to form an opinion on R, meaning that the MA will have some belief, disbelief, and uncertainty about the vortex core feature contained in R.

Each AA forms its own opinion on R denoted by $\omega_{\mathrm{R}}^{\mathrm{AA}_1}$ and $\omega_{\mathrm{R}}^{\mathrm{AA}_2}$. This notation states the agent forming the opinion as the superscript with the opinion reflecting belief, disbelief, and uncertainty as the subscript. The MA forms an opinion on each AA in use given by $\omega_{\mathrm{AA}_1}^{\mathrm{MA}}$ and $\omega_{\mathrm{AA}_2}^{\mathrm{MA}}$. Once the initial opinions are formed, they can be combined into a final opinion, $\omega_{\mathrm{R}}^{\mathrm{MA}}$, on the existence of a feature in R as

$$\omega_{\mathrm{R}}^{\mathrm{MA}} = \left(\omega_{\mathrm{AA}_1}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_1}\right) \oplus \left(\omega_{\mathrm{AA}_2}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_2}\right). \tag{6}$$

While Fig. 4 displays two AAs, any number of AAs may be incorporated into the agent structure allowing the use of any number of vortex core extraction algorithms.

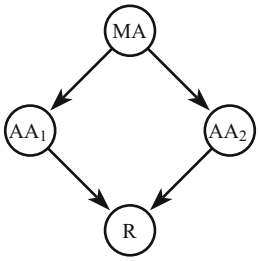


Fig. 4 Graphical representation of two algorithm trust network

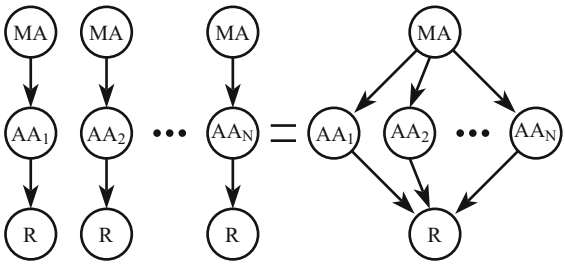


Fig. 5 Graphical representation of modular trust network

Figure 5 shows how each algorithm plays a role in only one of the transitive trust paths, allowing a modular handling of multiple algorithms. A transitive trust path can be visualized as any one path from the MA to R. Any algorithm's path may be added or removed from the trust network and the network will still function properly. This allows the agent structure to easily handle new and updated vortex core extraction algorithms. For example, if a new vortex core extraction algorithm is defined, it can be encapsulated in an agent and easily inserted without requiring a change of the architecture. Equation (7) uses the consensus and discounting operators to give the final opinion as a combination of all opinions for any number of AAs. With an increased number of algorithms there are more feature sets, allowing the agents to search through an increased amount of vortex cores giving more information on what cores are probable and what are not. Also, with added algorithms vortex cores that were not previously extracted could possibly be extracted. An agent cannot select a vortex core if it is not in one of the available feature sets.

$$\omega_{\mathrm{R}}^{\mathrm{MA}} = \left(\omega_{\mathrm{AA}_1}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_1}\right) \oplus \left(\omega_{\mathrm{AA}_2}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_2}\right) \oplus \cdots \oplus \left(\omega_{\mathrm{AA}_N}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_N}\right) \tag{7}$$

3.1.1 Algorithm Agent Opinion Dichotomy

The first agent opinions to generate are the AA, or algorithm agent, opinions. Recall that in a two AA structure there are two feature extraction algorithms that output two separate feature sets. It is important to recognize that each feature set is separate, having been extracted by a different extraction algorithm and thus by a different AA. Consider Fig. 6 containing two hypothetical separate line-type feature sets produced

Fig. 6 Two separate simple sets of line-type features hypothetically extracted by AA_1 (black) and AA_2 (gray)

by AA_1 (black) and AA_2 (gray). The black line comprises feature set 1 and the gray line comprises feature set 2. While these line-type feature sets are displayed together they are two separate sets.

With these two feature sets, corresponding opinions of AA_1 and AA_2 for feature set 1 may be defined. AA_1 needs to form an opinion at each point contained in each line in feature set 1. Also, AA_2 needs to form an opinion at each point contained in each line in feature set 1. Why does AA_2 need to form an opinion on feature set 1 even if it does not extract the features, or the exact points, contained in the feature? This follows from Fig. 4 and the resulting Eq. (6). If AA_2 does not form an opinion at each point, or each R, then the left-hand-side of Eq. (6) cannot be evaluated for there will be no values in $\omega_R^{AA_2}$. Both AA_1 and AA_2 need to form an opinion at each point in each feature set, leading to a dichotomy for defining the algorithm agents. AA_1 extracts the features in feature set 1 so it is termed the extracting algorithm agent (AA_E). AA_2 does not extract the features in feature set 1 so it is termed the non-extracting algorithm agent (AA_{NE}).

After opinions are defined for feature set 1, AA_1 and AA_2 need to define their opinions for feature set 2. This changes how opinions are set from feature set 1. In feature set 1, AA_1 was the extracting algorithm agent and AA_2 was the non-extracting algorithm agent. Now the roles are reversed for feature set 2. AA_2 extracts the features in feature set 2 so it is AA_E, and AA_1 is AA_{NE}.

This dichotomy between extracting and non-extracting algorithm agent opinions works just as well with multiple algorithms contained in the trust network as shown in Fig. 5 and Eq. (7). At each created feature set there will be one AA_E and the rest of the algorithm agents will be non-extracting algorithm agents. With this dichotomy in place, it is now possible to define the AA_E and AA_{NE} opinions.

It should be noted before moving further that during a simulation the entire data set is known at the instant flow features are extracted. This is not the data set of the converged solution but rather an iterant of the converged solution. This allows us to obtain information about features such as curvature, flow rotation and any other flow characteristic at that instant. It is with this information that we can set the belief tuple to project how the features will behave in future solution iterants. The flow does not need to be known *a priori* but rather information can be obtained from the data set at that instant.

3.1.2 Extracting Algorithm Agent Opinion

The belief tuple set for AA_E is constructed as follows: belief is set by extraction algorithm strengths, disbelief is set by extraction algorithm weaknesses, and uncertainty is set by flow feature characteristics. Belief set by algorithm strengths means

that each strength is given to the agent and a belief is set based on whether or not the extracted region has the strength characteristics. For example, work done by Roth [16] showed that the SH algorithm adequately extracts vortex core lines when they are close to straight and when the vortex has high strength. These two strength characteristics are given to AA_E and a high belief of one is set if the region has both characteristics, a low value near zero is set if the region has neither characteristic and any value in between the high and low values may be given for all other cases.

Disbelief is set similar to belief except the weaknesses, or situations where a feature extraction algorithm may spuriously extract a feature, govern the value. The weakness characteristics may be the exact opposite of the strength characteristics. Continuing the example used for belief, the SH algorithm does not work well for curved vortices or vortices with a low strength. So if a vortex has both of these weakness characteristics the disbelief will be set high, if neither characteristic is present then the disbelief is zero and for other cases a disbelief value may be set between the high and low values.

Uncertainty is a measure of the unknowns in an outcome, set from scientifically known characteristics of the flow feature. Missing or incomplete data can be taken into account in an agent's uncertainty. Some of the unknowns may positively affect an outcome while some may negatively affect an outcome and are encapsulated within the uncertainty calculation.

The main strength of setting the AA_E opinion values in this manner is that this template can easily be adapted to any feature with corresponding feature extraction algorithms. For example, if a feature has three feature extraction algorithms then as long as the strengths of each algorithm, the weaknesses of each algorithm, and some information about the physical formation of the feature are known they can be added to agents, allowing decisions about the expected probability of extracted features.

3.1.3 Non-extracting Algorithm Agent Opinion

After defining the AA_E's opinion a definition can be given for the AA_{NE}'s opinion. The belief tuple set for the AA_{NE} is defined as follows: belief is set by extraction algorithm strengths, disbelief is set by extraction algorithm weaknesses, and uncertainty is set by the distance from the closest extracted region.

The uncertainty is set according to the minimum distance between any region extracted by the AA_{NE} and the region under consideration. For example, if there are two feature sets and the region under inspection is contained in feature set 1 then the minimum distance would be measured between that region and the closest region contained in feature set 2. The idea is when the AA_{NE} extracts a region close to the AA_E, the AA_{NE} is more certain about the region so its uncertainty is near zero. When the AA_{NE} does not extract a region close to the region under inspection, it is uncertain about the AA_E's extracted region meaning that its uncertainty will be high.

Table 1 AA_E opinion values set for the SH vortex core extraction algorithm

AA_E	Sujudi-Haimes
b	straight core, high strength, low quality
d	curved core, low strength, high quality
u	distance from possible trip point

Table 2 AA_E opinion values set for the RP vortex core extraction algorithm

AA_E	Roth-Peikert
b	curved core, low strength, low quality
d	straight core, near zero strength, high quality
u	distance from possible trip point

3.2 Sujudi-Haimes Strengths, Weaknesses and Feature Characteristics

Table 1 gives the strengths, weaknesses and feature characteristics used for the SH vortex core extraction algorithm. Strength referes to the amount of flow rotation about the core and quality is a vortex characteristic originally defined by Roth [16]. In this research quality is the angle between a vortex core line and its associated velocity vector.

The weakness characteristics for the SH algorithm are the exact opposite of the strength characteristics. Curved core, low strength, and high quality are all characteristics that negatively affect the correct extraction of vortex core lines. While there are many possible feature characteristics, the only feature characteristic initially used in this research is the distance from a possible vortex trip point.

3.3 Roth-Peikert Strengths, Weaknesses and Feature Characteristics

The strengths, weaknesses and feature characteristics used for the RP vortex core extraction algorithm are given in Table 2. Two of the weakness characteristics for the RP algorithm are the opposite of the strength characteristics: straight core and high quality. Setting a straight core as a weakness characteristic might be misleading because the RP algorithm does not extraneously extract straight vortex core lines. A straight core is a weakness characteristic because when it comes to straight core lines there is more belief that the SH algorithm will extract them correctly than the RP algorithm. Using the SH and the RP algorithms together in this fashion helps us to match each algorithms strengths with the flow situations for which they were designed. The last weakness for the RP algorithm is a near zero strength. This simply means that there is some minimum threshold on vortex strength for which the RP algorithm correctly extracts vortices.

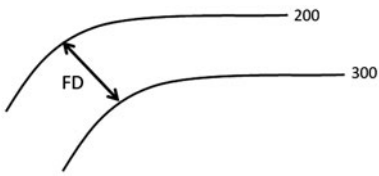


Fig. 7 Two line-type features extracted at 200 and 300 iterations show that feature displacement is found between a similar point on each line

The feature characteristic used for the RP algorithm uncertainty is the same as the feature characteristic used for the SH algorithm which is distance from a possible vortex trip point. When using multiple feature extraction algorithms, the same feature characteristics are used for all algorithms since feature characteristics are not algorithm dependent.

3.4 Master Agent Opinion

The MA can be thought of as the governing, or controlling, agent. It has the most influence on the believability of extracted features. Its job is to synthesize information from multiple AAs and provide a final decision on the extracted features. The MA's belief tuple is based on the idea that as a simulation converges to a final solution, so too will a feature converge from some beginning spatial location to a final location. This is implemented through a displacement measure called feature displacement (FD). Feature displacement is a measure of the displacement, or movement, of a region between any number of iterations. FD is divided by a reference length which nondimensionalizes the FD making it easier to work with across separate simulations with large variations in length scales. For line-type features the reference length is the line length. Equation (8) gives the FD when the region is a point. Here the subscript i refers to the iteration under investigation and $i - 1$ refers to the previous iteration where features were extracted which could be any number of iterations completed by the simulation:

$$FD_i = \frac{|\boldsymbol{P}_i - \boldsymbol{P}_{i-1}|}{\ell_i}. \tag{8}$$

Figure 7 gives an example of feature displacement between two line-type features. One of the lines is extracted at 200 iterations with the other extracted at 300 iterations. If a similar point is taken from each line the feature displacement at that point is defined as the magnitude of the distance between the two points divided by the length of the line at 300 iterations. Each point contained in the 300 iteration core line has a feature displacement based on the 200 iteration core line.

Another quantity used to define the MA's opinion is the change in feature displacement (ΔFD). Change in feature displacement is defined as:

$$\Delta FD_i = \frac{|FD_i - FD_{i-1}|}{number\ of\ iterations}. \tag{9}$$

Table 3 Raw data used to form final opinions

Point	Velocity ($\frac{m}{s}$)	Vortex	Strength ($\frac{1}{s}$)	FD	$\omega_R^{AA_E}$	ω_R^{MA}
26679	(7.6, 40.5, –5.8)	1	1132.6	0.03	(0.97,0.03,0.00)	(0.96,0.03,0.01)
30475	(136.6, 183.4, –1.9)	1	848.7	0.17	(0.99,0.01,0.00)	(0.98,0.01,0.01)
39116	(–61.5, 56.3, 12.9)	1	293.4	18.6	(0.91,0.09,0.00)	(0.05,0.01,0.94)
36244	(–19.8, 71.6, 5.9)	1	393.9	21.7	(0.91,0.09,0.00)	(0.00,0.00,1.00)

With feature displacement and change in feature displacement defined, the belief tuple for the MA is specified as follows: belief is set by feature displacement and change in feature displacement, disbelief is set by feature displacement and uncertainty is set by change in feature displacement.

The belief will be high, or close to 1, when the feature displacement is small accompanied with small changes in feature displacement. The belief will be low, or close to zero, when the feature displacement and the change in feature displacement are high. This says that the MA believes a feature when the feature has moved only a small amount between iterations under investigation and has a trend that suggests the feature will not move substantially with more iterations. The disbelief will be low when feature displacement is low and high when feature displacement is high. The uncertainty is high when change in feature displacement is high and low when change in feature displacement is low. This says that the MA is uncertain about the feature if the feature could move substantially with more iterations.

3.5 Method Overview

Table 3 gives some selected sample data where the described method has been used. The column 'point' denotes the identifying integer value for each computational grid point. The column 'velocity' is the velocity of the fluid at the given point. From the velocity components the two vortex feature extraction algorithms either select the point or not. A 1 in the column 'vortex' denotes a positive selection by either algorithm. If the point is not selected by a feature extraction algorithm then it would be given a value 0 in the 'vortex' column and dropped from further consideration. After a point has been selected, characteristics like vortex strength, curvature, FD, etc. are computed. From these the opinions $\omega_R^{AA_E}$, $\omega_R^{AA_{NE}}$, $\omega_{AA_E}^{MA}$, $\omega_{AA_{NE}}^{MA}$ can be formed to compute the final opinion, ω_R^{MA}.

All four points in Table 3 have been selected by the RP algorithm. Points 26679 and 30475 have a high strength and were selected by RP where its strengths were met resulting in the high belief of the opinion $\omega_R^{AA_E}$. Some of the data is missing from the table such as the opinion $\omega_R^{AA_{NE}}$, but it can be seen that when everything is accounted for the resulting final opinion ω_R^{MA} leads to the belief that this is an actual vortex core. For the two points 39116 and 36244 the strength is lower and FD is high. After combining all the information this results in a poor final opinion and the

point is not believed to be a vortex core. The results from this method are best seen graphically and are given in Sects. 4.1 and 4.2.

4 Concurrent Feature Extraction Results

Two CFD simulations were run on separate geometries to verify that concurrent feature extraction is possible and to validate the vortex core extraction method with subjective logic. The two geometries are common in CFD feature extraction research: a blunt fin and a delta wing. The blunt fin was selected as an initial test case that had well defined vortex cores and was relatively simple to grid and solve. The delta wing data set was selected as it had a more complex flow field that would help validate the vortex core extraction method.

It should be noted that neither of these simulations are of the magnitude where concurrent feature extraction would be useful. However, both simulations are representative pieces of larger, more complex simulations. For example, the blunt fin simulation is meant to compute the flow around a probe that protrudes from the outside of an aircraft. A larger simulation that would benefit from CAFÉ would simply compute the entire flowfield around the aircraft including the protruding probe. Larger simulations may not have more complex features but will have much more data to examine.

One crucial piece of information needs to be clear for proper interpretation of results. When agents form opinions on extracted cores, they have information from the current iteration of the simulation and previous iterations only. They do not use information from the fully converged simulation, or any iterations beyond the current iteration, to form opinions on extracted cores. Belief, disbelief, uncertainty, and expected probability of vortex cores can be determined without requiring a final converged solution giving information about a final simulation's expected vortex cores *before* a simulation is 100 % converged.

4.1 Blunt Fin

A CFD simulation was run of a blunt fin [6] geometry using the steady RANS equations solved with Fluent 6.3 to verify that concurrent feature extraction is possible and to validate the vortex core extraction method with subjective logic. The computational domain was generated as a structured curvilinear grid with 44,000 nodes. The Reynolds number based on the fin diameter was 630,000 and the one equation Spalart-Allmaras method was used to model turbulence. The inlet boundary condition was a pressure-inlet condition with freestream $M = 2.95$. The outlet boundary condition was a pressure-outlet condition, the top boundary condition was a symmetry condition. The fin and the lower boundary were modeled as walls. The solution reached full convergence at 900 iterations.

Concurrent feature extraction was replicated by exporting and saving to hard disk the entire flow field data set every 45 iterations from start to convergence. Each of these saved data sets were input into the vortex core extraction method where vortex core lines were extracted using the RP and the SH algorithms resulting in two feature sets per saved data set. Agents then produced final opinions on all features in both data sets and a final feature set was selected per data set.

4.1.1 Vortex Cores in Converging Data Sets

Figure 8 displays the vortex core extraction results obtained from the RP algorithm. Flow is moving from bottom right to top left. Extraneous cores have already been filtered out to make the images easier to understand. Black lines represent extracted vortex core lines from the converged data set and gray lines represent extracted core lines from the converging data sets. The percent convergence is based on the number of iterations at full solution convergence (900).

There are two vortex core lines for the blunt fin data set: the horseshoe vortex core line and the fin vortex core line. The core line that forms around the front of the fin in a horseshoe like shape is called the horseshoe line. The core line near the side of the fin is called the fin line. Experimental results of a blunt fin verify the formation of these two features [3].

Figure 9 shows a graph of the feature displacement for the endpoints of the horseshoe core line and the fin core line extracted by the RP algorithm and displayed in Fig. 8a–d. The two vortex core lines exhibit similar behavior when they are extracted by the SH algorithm. The start point is defined as the farthest upstream point and the end point is defined as the farthest downstream point. At 60 % converged, all but the end point of the fin line has a non-negligible feature displacement. This shows that at 60 % converged the entirety of the horseshoe vortex core line is very close to the same position it will be in at full solution convergence (see Fig. 8d).

The start point of the fin line has the same behavior as the horseshoe line. The end point of the fin line has a feature displacement within 2.5 %, and 10 % from 55 % converged to full solution convergence. This tells us that the end point of the fin line does not find a fixed position, but rather continues to move slightly every 45 iterations between 55 % converged and fully converged. This behavior suggests that the simulation is not fully converged as the RP algorithm takes two spatial derivatives of velocity to locate vortex cores which makes it very sensitive to variations in the velocity field solution.

Making sure that the feature displacement is zero for all features in the spatiotemporal flow domain extracted by the RP algorithm could aid in determining if a CFD simulation has reached full solution convergence. If the feature displacement is not zero then the features are continuing to move suggesting that the flow solution has not converged.

It is interesting to note that the upstream start point moves to its final location sooner than the downstream end point for both core lines. For the horseshoe line the *FD* at the start point is 0.8 % at 35 % converged and the *FD* at the end point is 0.8 % at

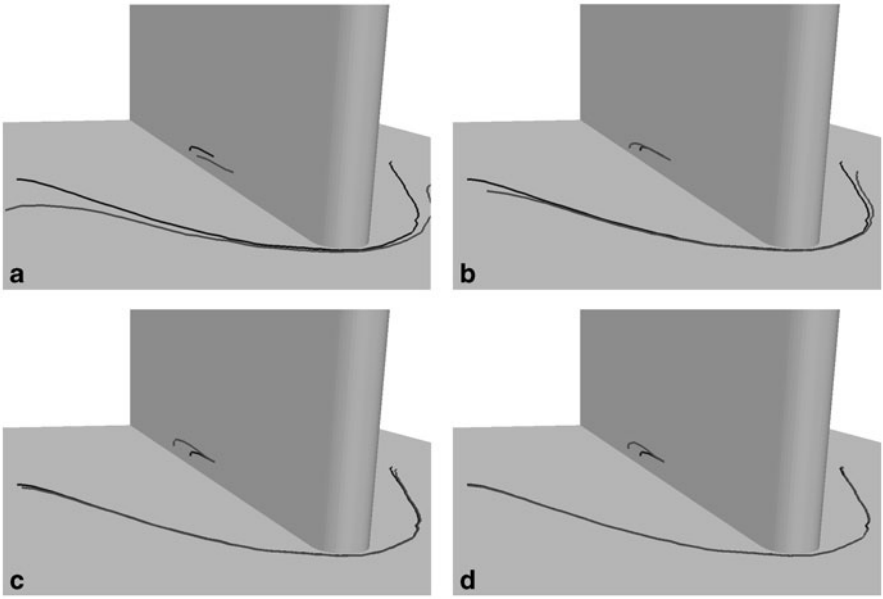


Fig. 8 Comparison of RP extracted vortex core lines from the converged data set (black) and converging data sets (gray). **a** At 30 % converged the horseshoe line begins to take shape upstream. **b** At 40 % converged the horseshoe line and the fin line are almost correctly resolved. **c** At 50 % converged the end point of the fin line moves downstream. **d** At 60 % converged the horseshoe line is spatially correct but the fin line is not

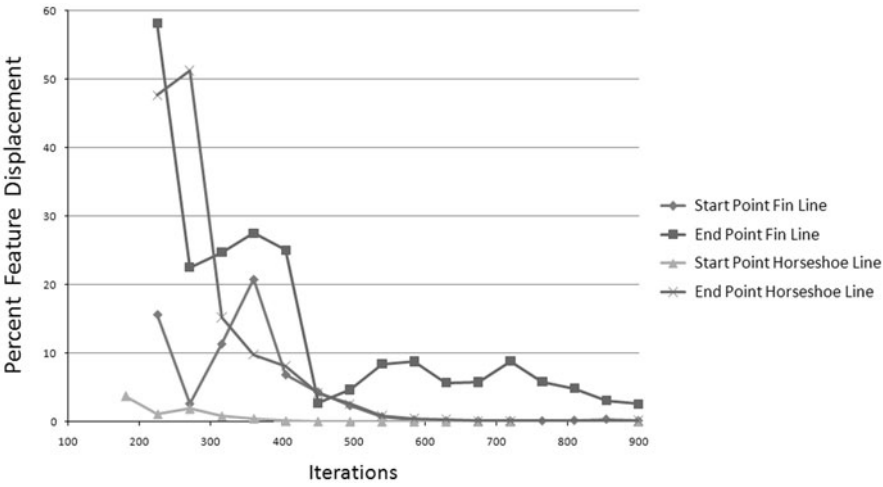


Fig. 9 Percent feature displacement for the endpoints of the horseshoe line and the fin line extracted by the RP algorithm

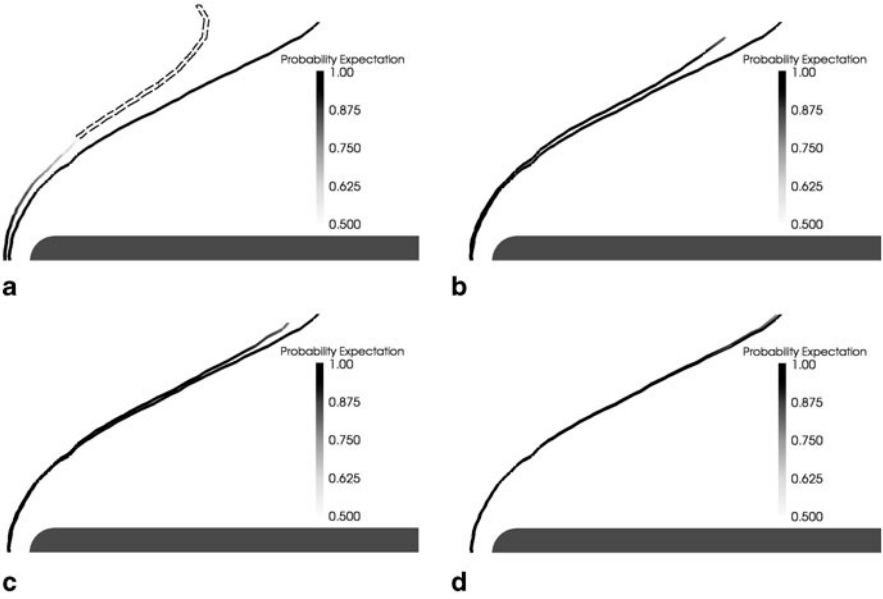


Fig. 10 Comparison of horseshoe core lines extracted by RP algorithm at 10 % convergence increments. The black line represents the extracted core line from the final converged solution. Flow is moving from left to right

60 % converged. This suggests that the vortex core lines are convected downstream as the solution converges. This convection can also be seen in Fig. 10.

4.1.2 Vortex Cores in Converging Data Sets Processed by Agents

Figure 10 is a comparison of the probability expectation between four separate core lines extracted by RP at 30, 40, 50, and 60 % converged. Recall that the probability expectation defined in Eq. (5) gives what one would expect the probability of a feature to be. In these figures, the flow is moving from left to right. The converged line is included as a reference and it is colored black and located downstream of the converging core. It is this core that the we are trying to match.

At 30 % converged, the probability expectation value is close to 1 at the start point and then quickly transitions to 0.5 at the downstream end point, which tells us that only the area near the start point has a high expected probability. We have judged a high expected probability as approximately 0.85 and above. At 40 and 50 % converged, the probability expectation value is close to 1 at the start point and stays close to 1 until near the end point which indicates that these lines are highly probable. This is a correct analysis by the agents since both lines are close to the final line. The 60 % converged core line is almost identical spatially to the fully converged core line but the agents have only given most of the line an expected probability of 0.90

or above and the rest is lower with the end reaching an expected probability of 0.75. The reason for this is that near the end point of the horseshoe line the vortex strength has a low value, which is one of the input criterion for belief in a feature. This low value drives the belief down at the end of the horseshoe line and therefore drives the probability expectation value down. In summary, the agents correctly identify the core line at 30 % convergence as having a low expected probability and correctly identify the core lines at 40, 50, and 60 % as having a high expected probability.

4.1.3 Comparison of Vortex Cores Processed by Agents from Converged Solution

Figure 11 is a comparison of the horseshoe line extracted at full solution convergence by the RP algorithm and the SH algorithm after agents have formed final opinions. This particular situation represents a case where both feature extraction algorithms have extracted a feature in a similar location so one more probable feature must be selected. Referring to Table 2, it can be seen that a belief criteria for the RP algorithm was curvature. This core line contains high curvature so the corresponding belief value for AA_E when RP extracts the line will be high which is shown in Fig. 11a. Weaknesses for the RP algorithm were not found in the extracted core which makes for the low disbelief in Fig. 11b. From Table 1, a criterion to set the disbelief for SH is curvature so the corresponding disbelief for AA_E when SH extracts the line will be high which is shown in Fig. 11f. Only some strengths were found in the SH extracted core line giving a belief around 0.75 for most of the horseshoe core line. The uncertainty for both algorithms in Fig. 11c, g are low, showing that the distance from a vortex trip point is low and that the horseshoe core line is converged as the FD and ΔFD must also be low for the uncertainty to be low.

For the horseshoe core line, the RP horseshoe line was selected as most probable because the probability expectation value throughout the line was higher as well as the belief. The probability expectation values are shown in Fig. 11d, h. Based on the strengths and weaknesses input criteria, agents correctly selected the RP horseshoe line as the feature with the highest expected probability.

4.2 Delta Wing

A CFD simulation was run of a delta wing using the steady Reynolds-averaged Navier-Stokes (RANS) equations solved using Fluent 6.3. The computational mesh was acquired from CGNS [1]. The inlet and outlet boundary conditions were pressure-far-field with $M = 0.3$ and $\alpha = 20.5°$. The delta wing was modeled as a wall with a symmetry plane through the delta wing center line. A pressure based compressible flow solver was used and the flow was modeled as laminar. The simulation reached full convergence at 300 iterations. This simulation was designed to match the experimental results of Kjelgaard [12] and the numerical results of

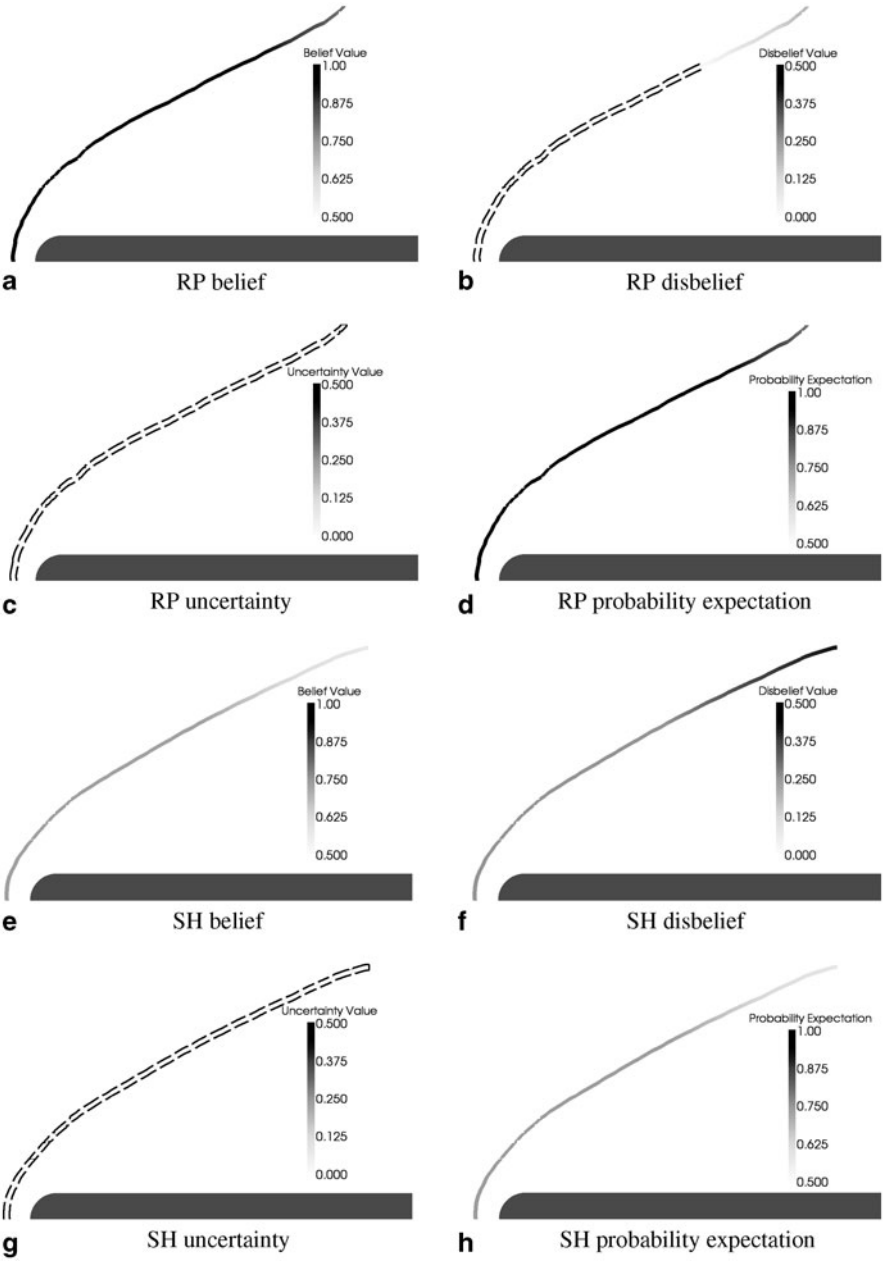


Fig. 11 Comparison of the belief tuple values and probability expectation for the horseshoe core line from the final opinion $\omega_{\mathrm{R}}^{\mathrm{MA}}$ of the converged data set extracted by the RP and SH algorithms at full solution convergence. Flow is moving from left to right

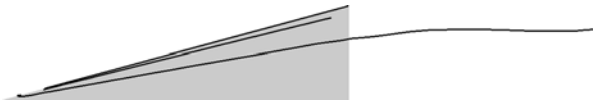


Fig. 12 Delta wing data set showing primary, secondary, and tertiary core lines extracted by RP

Ekaterinaris [4]. The main difference between this simulation and Kjelgaard and Ekaterinaris is that this delta wing has zero thickness. Feature extraction was replicated the same as with the blunt fin data set except data sets were saved every 30 iterations.

4.2.1 Definition of Extracted Vortex Cores

Figure 12 shows a top down view of the delta wing along with its vortex core lines extracted by the RP extraction algorithm. There are three vortex core lines in the delta wing data set which are also seen in the numerical results of Ekaterinaris and numerical results from CGNS. Starting at the delta wing centerline and moving up towards the edge of the wing the cores are named as follows: primary, secondary, and tertiary.

4.2.2 Vortex Cores in Converging Data Sets

Figure 13a–c shows the primary core line extracted by SH at 40, 60, and 80 % converged where the gray lines represent the converging primary cores and the black lines represent the primary core extracted from the converged simulation. At 40 % converged the primary core is slightly below and towards the centerline from the converged primary core in the upstream portion of the wing. Near the end of the wing the 40 % converged core oscillates and is moved towards the centerline. At 60 % converged the primary core is in the same spatial location as the converged primary core from the front of the wing to near the end of the wing. At the end of the wing there are some oscillations of the 60 % converged core. At 80 % converged the spot where the oscillations start moves downstream leaving the core extracted in the same spatial location as the converged core slightly downstream of the wing edge. From 40 to 80 % the point where the oscillations of the converging start moves downstream suggesting that the primary core is resolving downstream. Also, even at 80 % the primary core is still converging unlike the horseshoe core which reached convergence near 60 %. The section of the primary core that is not converged is in the delta wing wake region suggesting that the CFD code is still converging in the wake region. A more stable solution may exist over the wing where the core is not converging spatially at 80 %.

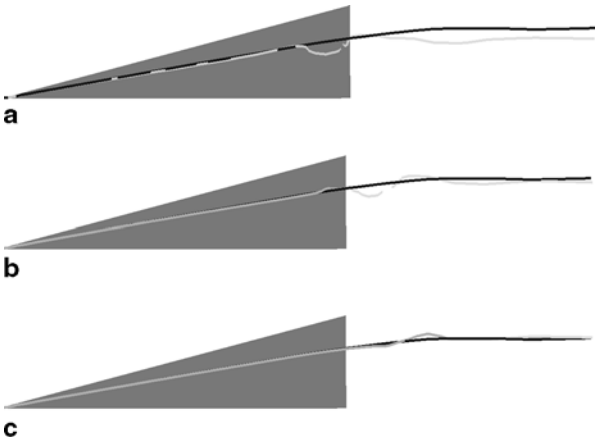


Fig. 13 Converging cores from the SH algorithm extracted at 40, 60, and 80 % converged. Gray is the converging data set. Black is fully converged

4.2.3 Vortex Cores in Converging Data Sets Processed by Agents

To understand the extraction method it is informative to visualize the MA belief of the primary core as it is converging. Figure 14a–c shows the MA belief for the SH extracted primary core at 40, 60, and 80 % converged. Recall that MA belief is set by FD and $\Delta\, FD$ requiring that $b = 1$ when $FD = \Delta FD = 0$. At 40 % converged $b = 0.8$ in a downstream section of the core. This is expected as there are larger variations in the solution in the wake region of the wing especially early in the simulation. At 60 % there are low belief regions that are also only contained in the wake region. At 60 and 80 % it can be seen that belief is near unity for almost the entire length of the wing. This shows that the primary core has not been changing spatially in this region as the solution converges.

The spacing of the final opinion depends strongly on the MA belief. If there is no belief in the MA opinion, then the final opinion will be dominated by uncertainty or disbelief without taking into account the AA_E and AA_{NE} opinions. If the MA belief is clustered too close to 1, then the AA_E and AA_{NE} opinions will dominate and FD and ΔFD will not properly be taken into account. Figure 14a–c shows that the MA belief may be clustered fairly close to 1, but the current settings seem to work well for the simple test cases tried here.

4.2.4 Comparison of Vortex Cores Processed by Agents from Converged Solution

Figure 15 shows the expected probability of the vortex cores extracted by RP and SH where extraneous cores have been removed from the figures. Figure 15a shows the primary core extracted by RP as having an expected probability near unity at the upstream end of the core. This is due to all of the strength characteristics from Table 2 being met. At the downstream end of the primary core the expected probability drops to 0.83 due to a lower quality and lower vortex strength.

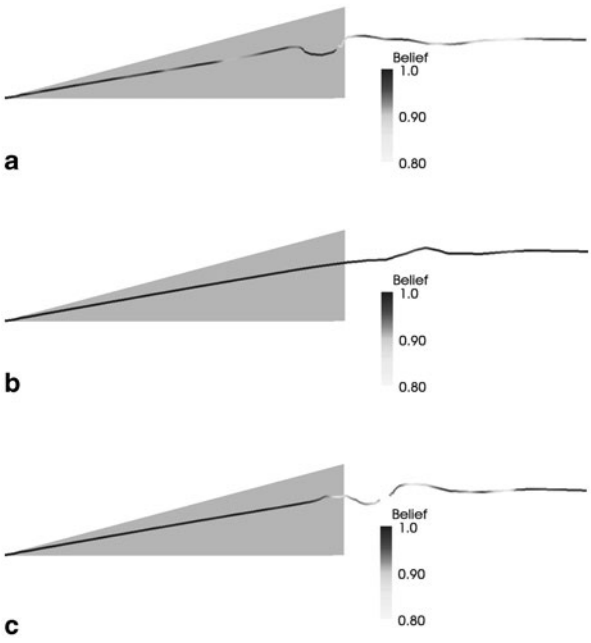


Fig. 14 Converging primary cores extracted by the SH algorithm at 40, 60, and 80 % converged. The cores are colored by the MA's belief

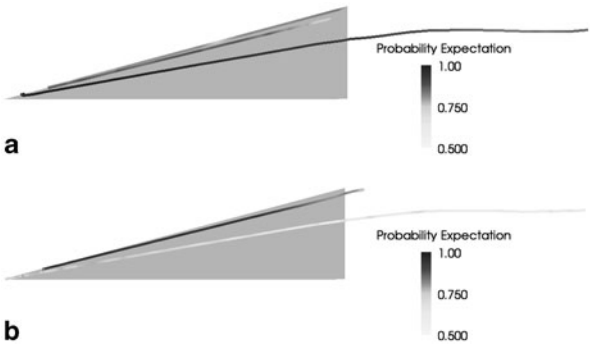


Fig. 15 Probability expectation values of converged cores for SH and RP

The primary core extracted by SH has a lower expected probability than the RP primary core throughout the entire length of the core even though it is extracted in almost exactly the same spatial location. This is due to the high curvature of the primary core which is not seen in this top down view. An idea to take from this is the MA is not saying necessarily that the core extracted by SH is incorrect but rather that the core extracted by RP is expected to be more probable based upon the input strengths and weakness information.

Unlike the primary core, there is some distinct difference between the secondary core extracted by RP and SH. The RP secondary core is extracted in the same spatial location as the SH secondary core except it ends before the end of the wing where the SH secondary core extends beyond the edge of the wing. Figure 15b shows an

expected probability near unity for most of the SH secondary core. This is due to all the strength characteristics from Table 1 being met. The RP secondary core has a lower expected probability than the SH secondary core throughout the entire core. It is the near zero curvature that gives the RP secondary core its lower expected probability.

The tertiary core extracted only by RP has $0.75 \leq E \leq 0.83$ throughout the entire core. These expected probabilities are due to a core line with near zero curvature, which is not one of RP's strengths, and a low vortex strength. A tertiary core that is only extracted by one extraction algorithm is an example of the benefits of incorporating multiple algorithms per feature.

5 Conclusion

We have presented results from the conceptual idea of using an agent based data mining system for extracting computational fluid dynamics (CFD) features in extremely large data sets concurrent with simulation. The agent-based approach allows for extensibility and incorporation of parallel extraction techniques, thereby maximizing the use of computing clusters. The CAFÉ concept utilizes subjective logic to evaluate the viability of extracted features. This culminates in a tool that is able to analyze the data concurrently with the CFD simulation and provide the researcher with the decision support capabilities to maximize research time and effectiveness.

The vortex core extraction method was validated with two well known vortex feature data sets: a blunt fin and a delta wing. For processing each data set, CAFÉ selected the most probable extracted vortex cores based on the coded strengths and weaknesses of each algorithm. When all the strengths of the vortex detection algorithm are met, the probability expectation value approaches one. Agents correctly selected which vortex detection algorithm had the highest expected probability, and showed how the expected probability increased as the simulation progressed from 30 to 60 % convergence. These test cases showed that probability expectation values can approach one before a solution is completely converged.

Feature displacement and change in feature displacement were defined to measure the movement of a vortex core between any number of CFD simulation iterations. These can be another metric to help determine when a simulation is completely converged. For a steady simulation, these measures monitor how much a feature is changing location as the simulation progresses and appeared to aid in the CAFÉ decision support aspect of feature detection.

With vortex cores, an important measure was curvature, as it tended to select between the Roth-Peikert and Sujudi-Haimes algorithms when vortex cores were extracted in similar spatial locations. As a byproduct, concurrent feature extraction shows how much a feature changes spatially as a solution converges. For the two validation cases, vortex cores tended to resolve spatially downstream as the solution converged.

Additional work is investigating concurrent extraction of other features and appropriate measures to provide belief and disbelief of those features. Continued research has extended the vortex core feature recognition to unsteady CFD simulations and will include analysis of larger, more complex computational data. These initial findings show the usefulness of pursuing CAFÉ and begin to highlight the capabilities which it may provide CFD researchers.

Acknowledgements This material is based upon work supported by the United States Air Force under Contract No. FA9550-10-C-0035 and is sponsored by the Air Force Research Laboratory (AFRL). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

1. CFD General Notation System (CGNS): Example CGNS files. http://cgns.sourceforge.net/CGNSFiles.html (2006). Accessed 1 Sept 2010
2. Computational Engineering International, Inc., 2166 N. Salem Street, Suite 101, Apex, NC 27523: EnSight User Manual for Version 9.0. http://www.ensight.com (2008). Accessed 1 Sept 2010
3. Dolling, D., Cosad, C., Bogdonoff, S.: An examination of blunt fin-induced shock wave turbulent boundary layer interactions. AIAA Paper 79-0068, New Orleans, LA, Jan 15–17 (1979)
4. Ekaterinaris, J., Schiff, L.: Vortical flows over delta wings and numerical prediction of vortex breakdown. AIAA Paper 90-0102, Reno, NV, Jan 8–11 (1990)
5. Haimes, R.: pV3: A distributed system for large-scale unsteady CFD visualization. AIAA Paper 94-0321, Reno, NV, Jan 10–13 (1994)
6. Hung, C., Buning, P.: Simulation of blunt-fin-induced shock-wave and turbulent boundary–layer interaction. J. Fluid Mech. **154**, 163–185 (1985)
7. Intelligent Light: Fieldview (version 13.1) [computer software] (2011)
8. Jøsang, A.: A logic for uncertain probabilites. Int. J. Uncertain. Fuzz. Knowl. Based Syst. **9**(3), 279–311 (2001)
9. Jøsang, A.: The consensus operator for combining beliefs. Artif. Intell. J. **141**(1–2), 157–170 (2002)
10. Jøsang, A.: Subjective evidential reasoning. In: Proceedings of the International Conference on Information Processing and Management of Uncertainty, Annecy, Fance, 1–5 July (2002)
11. Kitware: Paraview (version 3.12.0) [computer software]. http://www.paraview.org/paraview/resources/software.html (2011). Accessed 1 Sept 2010
12. Kjelgaard, S., Sellers, W.: Detailed flow-field measurements over a 75° swept delta wing. NASA TP 2997 (1990)
13. List, M., Gorrell, S., Turner, M.: Investigation of loss generation in an embedded transonic fan stage at several gaps using high fidelity, time-accurate CFD. ASME Paper GT2008-51220, Berlin, Germany (9–13 June 2008)
14. Mortensen, C.: A computational fluid dynamics feature extraction method using subjective logic. Master's thesis, Brigham Young University (2010)
15. Mortensen, C., Woodley, R., Gorrell, S.: Concurrent agent-enabled extraction of computational fluid dynamics (CFD) features in simulation. In: Proceedings of The 2009 International Conference on Data Mining, pp. 90–96, Las Vegas, NV (July 2009)
16. Roth, M.: Automatic extraction of vortex core lines and other line-type features for scientific visualization. Ph.D. Dissertation, Swiss Federal Institute of Technology (2000)

17. Roth, M., Peikert, R.: A higher-order method for finding vortex core lines. In: Proceedings of IEEE Visualization, pp. 143–150, Research Triangle Park, NC, USA (18 Oct 1998)
18. Sujudi, D., Haimes, R.: Identification of swirling flow in 3-D vector fields. AIAA Paper 95-1715, San Diego, CA (June 1995)
19. Thompson, D., Nair, J., Venkata, S., Machiraju, R., Jiang, M., Craciun, G.: Physics-based feature mining for large data exploration. IEEE Comput. Sci. Eng. **4**(4), 22–30 (2002)
20. Yao, J., Wadia, A., Gorrell, S.: High-fidelity numerical analysis of per-rev-type inlet distortion transfer in multistage fans–Part II: Entire component simulation and investigation. ASME Paper GT2008-50813, Berlin, Germany (9–13 June 2008)

A Data Mining Based Method for Discovery of Web Services and their Compositions

Richi Nayak and Aishwarya Bose

Abstract Due to the availability of huge number of web services, finding an appropriate Web service according to the requirements of a service consumer is still a challenge. Moreover, sometimes a single web service is unable to fully satisfy the requirements of the service consumer. In such cases, combinations of multiple inter-related web services can be utilised. This paper proposes a method that first utilises a semantic kernel model to find related services and then models these related Web services as nodes of a graph. An all-pair shortest-path algorithm is applied to find the best compositions of Web services that are semantically related to the service consumer requirement. The recommendation of individual and composite Web services composition for a service request is finally made. Empirical evaluation confirms that the proposed method significantly improves the accuracy of service discovery in comparison to traditional keyword-based discovery methods.

1 Introduction

Web services (WSs) have been widely accepted as a standard for building next generation Web-based systems [7]. Unlike system-specific local applications, WSs are interoperable software components that can be used in application integration and component-based application development [11]. The tremendous popularity of Web services over the years can be shown by the trend of number of WSs in one of the largest online API directories and resources, ProgrammableWeb (http://www.programmableweb.com/). It took about 8 years, 18, 9 and 6 months to reach the milestone of listing 1000, 2000, 3000 and 4000 WSs respectively in their directory [16]. A report published in 2009 states that over 440,000 developers have registered to use Amazon Web Services and there is an increase in registration by 10 % comparing it with last quarter results [17]. A similar rising trend can be seen on usage of WSs. Twitter attracts a transaction of 13 billion WS calls per day and Google attracts 5 billion calls per day for its WSs [9].

R. Nayak (✉) · A. Bose
School of Electrical Engineering and Computer Science, Science and Engineering Technology, Queensland University of Technology, Brisbane, Australia
e-mail: r.nayak@qut.edu.au

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_16

Web service discovery (WSD) is a core mechanism in achieving the full functionality of Web services. WSD is the process of finding the locations of providers of Web services which satisfy service consumers' requirements. The proliferation of Web services over the web makes it extremely difficult to discover Web services that can perform specialized tasks [12]. Furthermore, the complexity of a WSD mechanism increases with the existence of heterogeneity among services. Heterogeneity can be in the form of different service description languages, as well as different ontologies that are used to add semantics in WSs. The majority of WSD mechanisms use *traditional attribute-based match-making algorithms*, and *simple search engines* which provide only simple keyword-searches on WS descriptions. These algorithms fall short of understanding the underlying semantics of WSs [21] and/or of understanding the full scope of consumers' requirements. Consider the following example. A WS request is made for "weather" of a place. The WSs dealing with "humidity" and/or "rainfall" of that place along with "temperature" should also be discovered because they are semantically related.

To overcome the shortcomings of non-semantic WS discovery, semantic WS (SWS) that annotate the elements in WSs using an ontology language such as OWL-S, DAML-S and RDF-S can be used [10]. However WSD systems can only support matching WSs consumers and WS providers which use the same ontology [24]. Since WSs are heterogeneous, autonomous and developed independently, it becomes necessary to match/discover SWSs using different description languages, SWSs that are based on different ontologies, and/or SWSs against non-SWSs [21]. An important assumption behind this paper is the need to exploit semantic relationships among WSs (both semantic and non-semantic) in order to enhance the precision of WS discovery.

With the advent of service-oriented architecture (SOA), many organizations are using WSs to orchestrate their business workflow processes. A major problem is finding from a set of services the right WS or a combination of WSs which best suit the given agenda of a service consumer. During the process of service discovery, many services are found that can partially satisfy the request. Linking these relevant services in a systematic fashion can lead to achieving the overall objective of finding a combination of WS(s) which will meet the consumer's needs.

Consider an example related to discovery of WSs that are needed to resolve multiple service invocations. A WS request is made for "Automobile Hire in United Kingdom, payment converted to Australian Dollars". This request should invoke the services that deal with rental "car" or "automobile" or "vehicle" hiring as well as a currency conversion service. This example illustrates the importance of finding semantic service and then linking them for an efficient WS discovery.

Researchers have developed several approaches to achieving the final goal of effective WSD. These approaches can be categorized into information-retrieval, data mining, machine learning, semantic-based and Quality of Services [19, 20]. All these methods mainly recommend a single WS or present a list of ranked WSs in response to a WS request. They do not consider any possibility of linking multiple WSs if the request cannot be fulfilled by a single WS and requires a combination of services. It was noted that the problem is not the lack of WSs, but rather a lack

of infrastructure to link WSs to form an aggregated service [26]. A theoretical WS composition methodology has been proposed using Linear Logic theorem proving [22], but currently, to our knowledge, there exists no thoroughly evaluated approach to link and recommend multiple semantically similar WSs which address the consumer's service agenda. A number of approaches to WS compositions have been developed; however, they are primarily based on workflow techniques [2, 3, 13, 25] in which the WSs that form the components are known in advance. Most of these techniques support process modeling at the syntactic level and are unable to support reasoning at a conceptual level [23].

This paper presents the concept of linking WSs from the consumer's needs and perspective as described in the search criteria for relevant services. In this paper, we conceptualize WSD as a matching process that fully satisfies a service consumer's requirements by finding the most relevant services or service combinations. The proposed method uses the functional aspects of WSs such as inputs, outputs and descriptions to analyze semantic heterogeneity and linking operability. We first utilize a WS discovery module using semantic similarity derived from a "trained semantic kernel" so that it can identify semantically similar WSs for a service consumer [6]. The proposed Web composition algorithm models semantically similar WSs as nodes of a graph and then selects the best option for invoking multiple services according to an all-pair shortest-path algorithm. We then develop a fusion algorithm that integrates the single services recommended from the support-based semantic kernel and the service compositions recommended by the WS composition module. A thorough empirical evaluation of the proposed methodology has been performed with real-world datasets. The proposed system is able to recommend multiple interrelated WSs that match the consumer's criteria if a single WS fails to satisfy his/her requirements.

The rest of the paper is organised as follows. The next section presents the details of the proposed WSD methodology. Experiments and analysis of the results including comparison with existing methods are discussed in next section. Finally, the paper is concluded with a note on future research.

2 The Proposed WSD Method

Let $\{WS_1, WS_2, \ldots, WS_p\}$ be a set of p WSs available in a registry. Let q be a service consumer. A service consumer can be an application, a software module, a user or another service that requires a service. The task is to find WSs that fully satisfy the service consumer's requirements expressed as keywords. Let us first utilize a WSD module utilizing semantic similarity derived from a "trained semantic kernel" so that semantically similar WSs can be determined for the service consumer [5]. Let a set of individual WSs be $\{WS_1, WS_i, \ldots, WS_j, WS_s\}$ with $i, j, s \in p$, that are semantically associated to the service request q. In case of complex requests, a single WS may not be able to fully satisfy the requirements. In such situations, a composition of multiple WSs needs to be suggested. The proposed link analysis

module is applied on the semantically associated services and a set of WSs compositions of the form $\{(WS_1 \rightarrow WS_3 \rightarrow WS_5), (WS_4 \rightarrow WS_7), (WS_i \rightarrow WS_{j,} \rightarrow WS_s)$ is returned. The proposed fusion algorithm is applied to integrate the semantically related WSs and their compositions. Finally, the final set of ranked WSs consisting of individual WSs and combinations of WSs, $\{(WS_1), (WS_{i+1}), (WS_i \rightarrow WS_{j,} \rightarrow WS_s)\}$ are recommended to close the service consumer's given agenda.

2.1 Finding Semantically Similar WSs

A latent semantic kernel is constructed with a general-purpose dataset [8] so that it can support diverse domains and different topics of consumer requests. The trained semantic kernel is used in match-making the service request and WSDLs using the process explained in [5] . In order to create the kernel, the document Importance (DI) (based on support) is computed for each processed document. On the basis of DI, training documents are assigned into bins such that each bin contains equally important documents. Documents of a bin are then merged to form a single document representing the contents of the bin. This type of compression allows the reduction of the matrix size in terms of the number of documents, keeping the number of terms from the original corpus intact. The reduced size matrix facilitates the matrix factorization method, Singular Value Decomposition, in creating the latent semantic kernel representing a diverse range of domains. The constructed support-based semantic kernel is then used to find the similarities between the processed WSDL documents and a consumer request.

The degree of similarity between the service request and the WSs in the form of the input and output parameters (P'), the operation name (N') and the description (D') of the WS is computed using the following formula:

$$S_{sum} = w_1 \times sim(Q, P') + w_2 \times sim(Q, N') + w_3 \times sim(Q, D'),$$

where, w_1, w_2 and w_3 are the assigned weights for the parameter (P'), operation name (N') and description (D') components of the WS respectively. The constructed semantic kernel (SK) is utilised to find the similarity between a service request (Q) and a component of the WS (W) using the following model:

$$sim(Q, W) = \cos(Q, W) = \frac{Q^{\mathrm{T}}(SK)P^{\mathrm{T}}W}{\left\| Q^{\mathrm{T}}(SK) \right\| \left\| (SK)^{\mathrm{T}}W \right\|}.$$

After careful consideration of various scenarios, equal weight is assigned to each of the components such that:

$$w_1 = w_2 = w_3 \quad \text{and} \quad w_1 + w_2 + w_3 = 1.$$

The use of a semantic kernel helps to locate semantically similar WSs that were otherwise not found using traditional retrieval/matching methods. For example, the

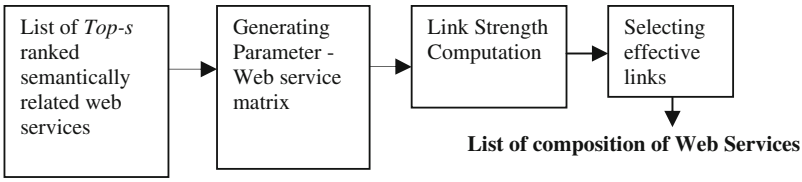


Fig. 1 Overview of the link analysis module

provider of service request of 'automobile' may also find WSs related to 'car' or 'vehicle' useful since they are semantically related.

The WSDLs (Web Services) present in the repository are then ranked in the descending order of similarity value to the service request. A list of top-*s* WSs is provided as an input to the fusion algorithm for final recommendations as well as they are used for link analysis if the service request includes more than two request terms. There is a high probability that two or more terms in a request can be interrelated. Consider the request 'Weather by postcode'. Let there exist two independent WSs, 'weather by location' and 'location by postcode'. A single WS will not be able to provide the information on weather, given a postcode. This request can only be fulfilled by an integrated solution of WSs related to 'weather' and 'postcode'.

2.2 Link Analysis for Finding Compositions

The Link Analysis module (as depicted in Fig. 1) deals with finding the possible combinations among a set of WSs that are found by the semantic kernel in response to a service request. Link analysis is not performed on the entire WSs in the repository; rather, it is only applied to semantically associated WSs responded to a service request, as it is time consuming to perform link analysis on the entire repository. The link analysis module analyzes the input and output parameters matching between different functional methods that exist in a pair of WSs. A parameter-WS matrix is created using the names of input and output parameters of each method with binary representation (whether the parameter name exists in the Web Service) where each row represents a unique parameter name and each column represents the input or output of a method in the WS as shown in Table 1. The link strength is computed for all methods that can be linked in two WSs. The optimum paths between different sets of source and destination are obtained using an all-pair shortest-path algorithm and are ranked on the basis of composition strength. The output is a list of compositions of interrelated WSs.

Table 1 A sample of parameter-WS matrix

Parameters	$WS_1(M_{11})$		$WS_1(M_{12})$		$WS_2(M_{21})$		$WS_3(M_{31})$		$WS_4(M_{41})$	
	I	O	I	O	I	O	I	O	I	O
A	1	0	0	0	0	0	0	0	0	0
B	1	0	0	0	0	0	0	0	0	0
C	0	1	0	0	1	0	0	0	0	0
D	0	1	1	0	1	0	0	0	0	0
E	0	0	0	1	1	0	1	0	0	0
F	0	0	0	0	0	1	1	0	0	0
G	0	0	0	0	0	0	0	1	1	0
H	0	0	0	0	0	0	0	0	0	1

2.2.1 Performing Link Analysis

Let $WS = \{WS_1, WS_2, \ldots, WS_s\}$ be the top-s selected Web services semantically associated to a service consumer request. Suppose a WS_i contains the maximum of m methods ($M_{i1}, M_{i2}, \ldots, M_{im}$). Each method contains a maximum of r input parameters ($I_{m1}, I_{m2}, \ldots, I_{mr}$) and o output parameters ($O_{m1}, O_{m2}, \ldots, O_{mo}$). Let $P = \{P_1, P_2, \ldots, P_t\}$ be the parameter set containing t unique input and output parameter names in all methods of WS. A Parameter-WS matrix is derived in which each row represents a unique parameter name and each column represents the input or output of a method in the WS. Each cell contains the value denoting the presence or absence of an input/output parameter in the method. Table 1 represents a sample of the parameter-WS matrix. It contains 8 rows containing the unique parameter names (A–H) and 10 columns denoting the input and output parameters of the methods ($M_{11} - M_{41}$) of 4 WSs.

A WS link $l_{(WS_i \to WS_j)}$ is said to exist if output parameter of a method of WS_i matches with the input parameter of a method of WS_j. Only the output parameters that match with the input of another WS are defined as 'matching-parameter'. The matching-parameter is represented as $P_{kM(WS_i \to WS_j)}$. A WS link is defined as 'self-link' if it exists between two methods of the same WS. A self-link can be represented as $l_{(WS_i M_i \to WS_i M_j)}$, where M_i and M_j are two different methods of the WS WS_i. Table 2 shows the WS links and self links obtained based on the data in Table 1.

The strength of a link demonstrates the compatibility between the various WSs to be linked. It is determined by considering the number of matching parameters between two different methods of a pair of WSs and the number of input parameters present in the methods of the destination WS, as follows:

$$\text{Link strength } (l_{(WS_i \to WS_j)}) = \frac{\sum_{k=1}^{t} P_{kM(WS_i \to WS_j)}}{\sum_{i=1}^{r} I_{Mi(WS_j)}}.$$

Table 2 All possible methods and WS links

Method link	WS link	Self link
$M_{11} - M_{12}$	WS_1	Yes
$M_{11} - M_{21}$	$WS_1 - WS_2$	No
$M_{12} - M_{21}$	$WS_1 - WS_2$	No
$M_{12} - M_{31}$	$WS_1 - WS_3$	No
$M_{21} - M_{31}$	$WS_2 - WS_3$	No
$M_{31} - M_{41}$	$WS_3 - WS_4$	No

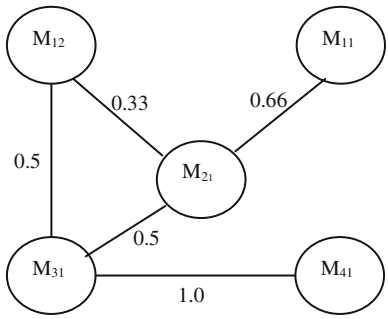


Fig. 2 Graph representation of linking scenario

The links with link strength higher than a threshold (Ψ) are used for finding realistic compositions. The threshold (Ψ) value is determined empirically. Once the WS links between different methods are obtained, the WS composition can be modelled using a graph. A graph is denoted as $G = (V, E, f)$, where V is the set of methods of WSs; E is the set of edges in the graph G; and f is a mapping function $f: E \rightarrow V \times V$. The value of the mapping function is obtained by link strength. Figure 2 shows a graph linking four WSs presented in Table 2.

The problem now is traversing the graph in order to find the shortest path between any pair of vertices. The Floyd Warshall algorithm [6] which is an all pair shortest path algorithm is adopted to calculate the shortest path from each method to all other methods of WSs using the existing links. This algorithm suffices for our problem of finding the shortest-path with undefined source and destination for a WS composition. The input for the Floyd-Warshall algorithm is a $n \times n$ matrix with W representing the edge weights of an n-vertex graph with $W_{ij} = (w_{ij})$ where

$$w_{ij} = \begin{cases} 0, & \text{if } i = j, \\ l_{(WS_i \rightarrow WS_j),} & \text{if } i \neq j \text{ and } (i, j) \in E, \\ \infty, & \text{if } i \neq j \text{ and } (i, j) \notin E. \end{cases}$$

Let $d_{ij}^{(k)}$ be the weight of a shortest path from vertex i to vertex j with all intermediate vertices in the set $\{1, 2, \ldots, k\}$. When $k = 0$, no intermediate vertices exists

Table 3 Overview of link analysis algorithm

1.	Select a set of top-s WSs from the list of WSs obtained from the support-based semantic kernel sorted in descending order of semantic similarity
2.	Create the parameter-WS matrix
3.	Perform matching operation between outputs of a method to the inputs of all other methods within the set of WSs under consideration
4.	Calculate the strength of a link between two WS methods
5.	If the link is not a self link and the link strength is greater than a threshold value (Ψ), accept the link, else reject the link
6.	Model a graph of WS connectivity
7.	Identify the shortest path for the WS composition from the graph
8.	Compute the strength of composition for each of the different WS compositions
9.	Sort the WS compositions in descending order of composition strength and Select a list of top-l unique compositions

between vertex i and vertex j. Thus, it has at most one edge, and hence $d_{ij}^{(0)} = w_{ij}$. A recursive definition is given by

$$d_{ij}^{(k)} = \begin{cases} w_{ij}, & \text{if } k = 0, \\ \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k)} + d_{kj}^{(k-1)}\right), & \text{if } k \geq 1. \end{cases}$$

The matrix of shortest-path weights gives the final solution. Based on the recurrence relation given by above equation, the values of $d_{ij}^{(k)}$ are calculated in order of increasing values of k. The composition strength for each shortest-path is computed as the average of link strength for all links in the compositions.

$$\text{Composition strength}(l_{(WS_i \to WS_j \to WS_k)}) = \frac{\sum\limits_{\forall i,j,k} Link\ strength(l_{(WS_i \to WS_j)})}{Number\ of\ links}.$$

Table 3 presents the overview of the link analysis algorithm.

It can be noticed that the link analysis phase directly relies on the name comparison between input/output parameters of two WSs. Ambiguity in parameters names may affect the matching process in this phase. The link analysis phase, however, is conducted after the semantic analysis and a number of semantically related WSs are retrieved. Inclusion of first phase reduces the possibility of having a WS that may contain parameters those are semantically unrelated to the query terms. The ambiguity problem can be handled by considering the equality of synonyms between names (e.g. car → automobile, movie → film) as well as similarity of names based on common string edit distance operation (e.g. chtitle → title). A thesaurus such as WordNet can be used to exploit synonyms (e.g., movie → film) and the user-defined dictionaries can be used to identify abbreviations (e.g.Emp → Employee), acronyms (e.g. DOB → Date of Birth), and user-defined synonyms.

Table 4 Fusion algorithm

1.	Let N be the exhaustive set of WSs containing WSs from both the preceding phases that is a list of individual WSs and their possible compositions
2.	Let l be the unique WS compositions obtained from the link analysis module only
3.	if $((N - l)! = 0)$ Select $(N - l)$ individual WSs that have not occurred in any of the WS compositions from the list of ranked WSs obtained from the semantic kernel
4.	else Select a maximum of l unique WS compositions
5.	Compute the similarity between the service request and the l unique composite WSs using cosine similarity with kernel
6.	Sort the list consisting of individual and composite WSs in descending order of similarity value
7.	Recommend the top-n WSs most relevant to the service consumer

2.3 Fusion of Web Services and Their Composition

The next task is to combine the list of top-s WSs sorted in descending order of semantic similarity and the list of top-l unique compositions sorted in descending order of composition strength. The fusion algorithm (shown in Table 4) is developed to create an exhaustive list of WSs by integrating the list of semantically associated individual WSs and the list of composed WSs.

There may be situations where the link analysis module is unable to produce relevant links. This may be due to the use of a single term in the service request or due to the absence of links in WSs. In these cases, the fusion algorithm provides the list of semantically similar WSs. The fusion algorithm plays a vital role in reordering the results if there are inputs from both semantic analysis and link analysis phases.

3 Empirical Evaluation

3.1 The Datasets

A general-purpose Wikipedia dataset [8] is used in creating semantic kernels in order to allow the kernels to represent the knowledge of many domains that a WS possibly may belong to. The kernel constructed from this dataset is used to find the similarity between a service request and the WSs (or WSDLs). Live WSs have been chosen to provide accurate and reliable results. The WS dataset contains 873 WSDL documents from XMethods [18] and QWS Dataset [1]. The WSDL documents represent a variety of WSs such as literature, communication, language translation, sports, finance, news, geographic location etc.

In order to evaluate the proposed methodology, a set of service requests was required. After carefully analysing the domain and content of the Web services (WSDL documents), 50 queries (or service requests) were synthetically created. A variety of topic has been considered in creating the query since in real life the queries are

related to any topic. The objective of structuring the queries was to evaluate both the semantic analysis and the link analysis phases. In order to evaluate the link analysis phase queries have been designed such that they have the potential to be linked; i.e. only one WS would not be able to satisfy the requirement of the whole query. A set of 14 queries have been designed so that the link analysis phase can be exclusively evaluated. The remaining queries have the potential to evaluate the semantic analysis as well as the combination of both in the fusion engine. Queries cover a variety of topics such as finance, sports, ciphers, barcode, email verification, spell checking, fax etc. Some examples are 'weather by postcode', 'residence information by phone number', 'music audio song', 'currency conversion', 'fraud detection in credit card service', 'image conversion' etc. The query data set should be considered unbiased as it was created by analysing the Web services (WSDL documents) before the proposed methodology was implemented. Each query is having at least two terms and the average query size is three. This dataset is available on http://applieddatamining.info/datasets/WebServices.

3.2 Evaluation Criteria

To evaluate the accuracy of the proposed WS discovery method, F-score measures is used. F-score is the harmonic mean of precision and recall. Precision is defined as the ability to provide the relevant WSs from a set of retrieved WSs. Recall is defined as the ability to provide the maximum number of relevant WSs from a set of relevant WSs.

$$\text{Precision} = \frac{Number\ of\ relevant\ WSs\ retrieved}{Total\ number\ of\ retrieved\ WSs\ from\ the\ dataset}$$

$$\text{Recall} = \frac{Number\ of\ relevant\ WSs\ retrieved}{Total\ number\ of\ relevant\ WSs\ in\ the\ dataset}$$

$$\text{F-score} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The reported results of these measures are averaged over 50 service requests. While evaluating the approach, the precision at n has been evaluated. The value of n has been selected as 5, 10 and 20 for all experiments.

3.3 Experiment Design

Before the link analysis is conducted, web services that (semantically) match the service request are identified by the proposed method. A latent semantic kernel

Table 5 All possible methods and WS links

Web service matching methods	Precision Top 5	Recall Top 5	F-score Top 5	F-score Top 10	F-score Top 20
LSK (support-based selection) ($U_k = 300$)	*60.4*	*63.24*	*44.98*	*54.84*	*61.79*
LSK (random selection) ($U_k = 300$)	58	61.85	43.96	54.3	59.86
VSM tf $\times$ idf (with Wordnet enhancement)	53.2	56.9	40.47	47.93	55.12
VSM tf $\times$ idf keyword-based	51.6	54.68	39.08	46.73	53.1

is constructed to support semantic discovery of the services utilising the method presented in [5]. Two types of kernels are constructed utilising the term-document matrix using support-based distribution and random distribution. Standard text pre-processing such as stop-word removal and stemming is performed on the kernel training data and the WSDL documents. The use of words that are hybrid in nature is common in WSDL documents. A hybrid word can be a compound word (e.g. news-paper) or a composite word (e.g. StockPriceCheck) or a joint word with a connected symbol such as hyphen (e.g. stock-price). Hence, hybrid words are converted into multiple standard dictionary words wherever possible.

Experiments are also conducted with the simple keyword matching using the VSM (vector space model), calling this method as tf$\times$idf method. We also added an experiment by expanding the service request by adding semantically similar terms using Wordnet [15] to improve the precision of matching, calling this method as tf$\times$idf with Wordnet.

A set of experiments have been performed to find whether linking has any positive effect in recommending multiple interrelated services, in comparison to recommending the WSs independently after semantic analysis (i.e. without linking them). Experiments have been carried out to find the exact number of WSs that should be analysed to determine service compositions. Each of the sets containing the top 5, 10, 15, 20, 30 and 40 WSs is considered as a possible candidate for finding compositions. Experiments have also been carried out to analyse the impact of threshold value, which can be set on link-strength, on the accuracy of the retrieved WSs. Experiments have also been designed to analyse the time required for linking multiple WSs by varying the number of linked WSs. Time analysis helps to achieve an optimal solution to find the number of WSs that should be considered for linking.

Experiments have also been performed to find if it is worthwhile to fuse the results from the previous two phases to achieve higher accuracy compared to the results obtained from an individual phase.

3.4 Results Evaluation and Discussion

Results in Table 5 show that the WSD utilizing the semantic kernel outperforms the vector space model based discovery methods.

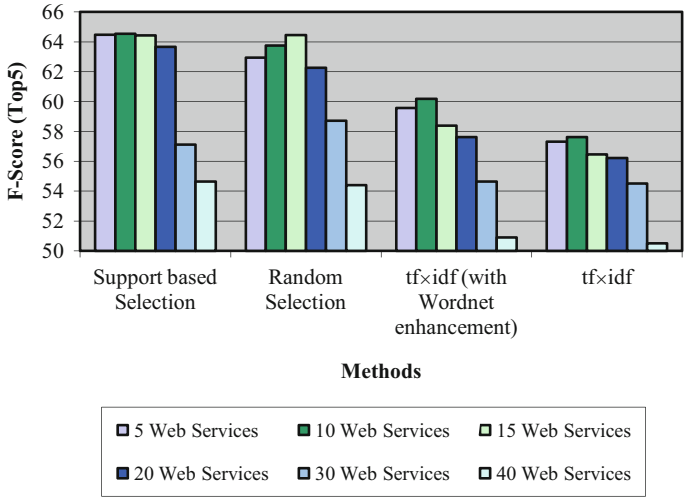


Fig. 3 Effect on *F*-score value for increasing number of WSs considered for linking

Results presented in Figs. 3 and 4 show how many semantically associated WSs to a service request should be analyzed in order to find out the WSs compositions. As the number of WSs increases beyond 20, accuracy (*F*-score) of WSD decreases (Fig. 3). Consideration of larger numbers of WSs in forming compositions leads to service compositions irrelevant to the service request. Also, consideration of only 5 WSs for finding a composition leads to comparatively poor performance due to having a smaller number of possibilities. An optimum solution can be found by linking the WSs in the range of 10–15. Results in Fig. 3 also show that this behavior is true for the set of WSs obtained by using any method.

Figure 4 presents the response time analysis results. Response time is defined as the time elapsed between the service request submission and the WSs returned to the consumer. For LSK support-based selection, even though there is no significant difference in terms of response time for combining the 5 or 10 or 15 WSs, there is a significant difference when the number of WSs to be linked increases beyond 20. The time required to check if it is possible to link 20 WSs is three times to the time required for checking 15 WSs. As the number of WSs considered for linking increases beyond 30, the time required for composition increases a further five times. The figure also shows that support-based selection performs the best in terms of average response time. The list of WSs discovered by a support-based semantic kernel is more precise in comparison to the WSs list found by other methods and hence the time required for forming a composition is reduced.

Experiments have been performed to analyze the impact of threshold value on the precision of WSD and determine the correct magnitude of the threshold value (Ψ) to select the links relevant for WS composition. Results presented in Fig. 5 show that for higher threshold values (greater than 0.8), there is a rise in *F*-score values. Results also show that the top-5 compositions are most relevant to a service request,

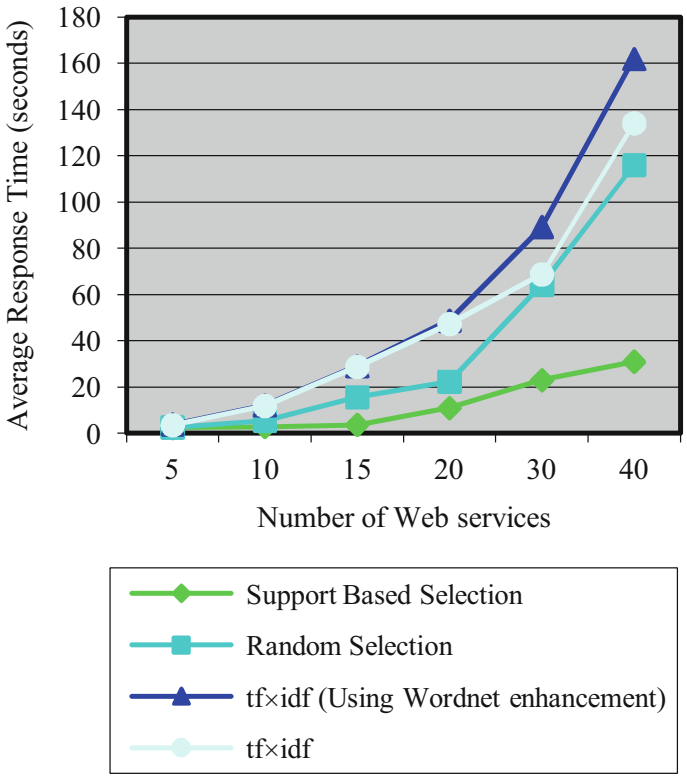


Fig. 4 Average response time for different number of WSs under consideration

a higher top-5 F-score value is obtained in comparison to the top-10 or top-20 F-score values. These results are averaged only for requests that have returned at least one link in the WS composition. A threshold of 0.9 has been used in analysis as the recall value starts decreasing after the value of 0.9. Results in Fig. 6 show that as the value of the threshold increases, the number of links and compositions per request as well as the number of links per composition decreases. However results in Fig. 5 confirm that even though the number of compositions decreases, the F-score value increases. This indicates that by selecting stronger links (using a higher threshold value), a larger number of relevant compositions can be obtained.

Table 6 presents the top-5 F-score value for the outputs obtained only from using the individual services (without linking – semantic analysis only), by considering the composite WSs only, and from the fusion of the results obtained after integrating individual and composite WSs after analysing top-10 WSs. Experiments were conducted with the top-10 and top-15 semantically associated services as a response to service request for forming compositions. There was no significant difference. It can be seen that the F-score value obtained from the fusion outperforms both without

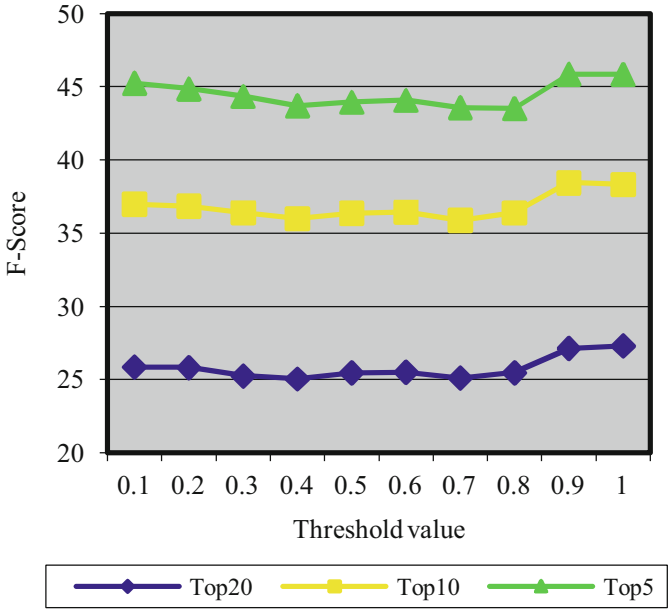
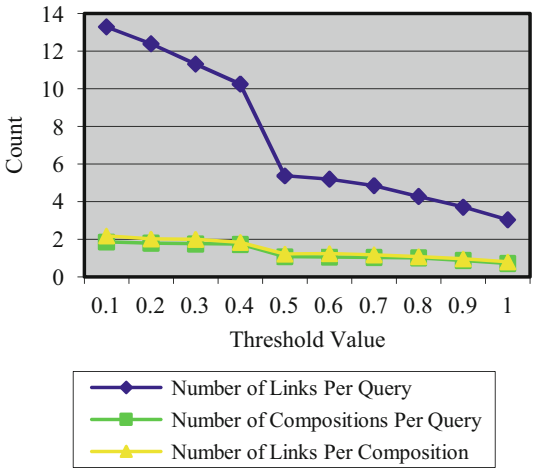


Fig. 5 Effect of threshold value on *F*-Score

Fig. 6 Comparing the effect of the number of links on the threshold value



linking and linking only for all the methods under consideration. The very low *F*-score value obtained when the linking only situation is considered can be attributed to the fact that only 40 % of the total queries under consideration have the potential to be linked. It can be noted, however, when the compositions are fused with the list of WSs, which are semantically related to the service request, overall precision increases. There has been an improvement of at least 4.5 % when linking is performed

Table 6 Comparing top-5 F-score value to analyse the effect of fusion

	Individual services	Service compositions	Fusion (individual and compositions)
LSK (support-based selection)	*61.79*	*17.41*	*66.29*
LSK (random selection)	59.86	19.93	65.86
VSM tf × idf (with Wordnet enhancement)	55.12	22.92	64.32
VSM tf × idf or keyword-based	53.10	22.42	61.6

on the result obtained by the support-based latent semantic kernel. When linking is performed on the results obtained from the random-based latent semantic kernel, vector space model discovery and vector space model with Wordnet, improvements of 6, 8.5 and 9.2 % respectively are evident. This shows that linking boosts the results.

Results obtained from the proposed method have been further analysed based on statistical analysis using analysis of variance (ANOVA) [4, 14].

There is a significant difference in F-score value when different numbers of WSs are considered for linking. The support based kernel is able to find semantically similar WSs in a much superior way compared to all the other methods. This eventually helps in finding better possibility of linking.

A p-value of 7.55E−09 is obtained; the hypothesis is accepted, as the p-value is much smaller than $\alpha = 0.5$. The fusion enhances the accuracy of WSD by combining the outputs from both the semantic analysis and link analysis. A significant improvement in accuracy (F-score) has been accomplished by fusion in comparison to individual semantically associated WSs or Web composites. The Anova test finds a p-value of 1.879E−12, which again is smaller than $\alpha = 0.5$. Thus the above hypothesis can also be accepted. The support based kernel is able to find semantically similar WSs in a much superior way compared to all the other methods. This eventually helps in finding better possibility of linking.

A one-way between group analysis of variance (also known as single-factor ANOVA) was performed to explore the impact of F-score value for different methods (or groups) used to find the semantically similar WSs. Table 7 provides the summary of the parameters that needs to be calculated for performing ANOVA on a sample size of 50. The null hypothesis (H_0) for this test is: the means of all the four different methods are equal. The alternate hypothesis (H_1) states that the means of the different methods are different. The values of F, $F_{critical}$ and p were found to be 9.8, 2.24 and 1.03E−08 respectively. It is evident that $F > F_{critical}$ and the value of p is much smaller than $\alpha = 0.5$. Hence, it can be said that there is a significant difference in F-score values obtained from the different methods. From the average value presented in Table 7 it can be said that support-based selection method is superior to other methods in consideration.

A two factor analysis of variance has been performed to explore the impact on F-score value for linking multiple WSs considering different methods used to link

Table 7 Summary table of one-way ANOVA to analyse F-score values obtained from different methods

Groups	Count	Sum	Average	Variance
LSK-support-based selection	50	2898.52	57.97	985.17
LSK-random selection	50	2799.27	55.99	1122.36
tf × idf with Wordnet enhancement	50	2542.00	50.84	885.75
tf × idf	50	2458.64	49.17	938.04

Table 8 Two factor ANOVA summary of different methods that considers different number of WSs considered for linking

Summary	Count	Sum	Average	Variance
Without linking	6	343.78	57.296	10.016
Linking 40 WSs	6	324.78	54.130	8.959
Linking 30 WSs	6	347.41	57.901	9.078
Linking 20 WSs	6	365.86	60.976	10.866
Linking 15 WSs	6	370.61	61.768	12.549
Linking 10 WSs	*6*	*371.02*	*61.836*	*6.632*
Linking 5 WSs	6	362.76	60.460	7.223
LSK - support-based selection	*7*	*430.67*	*61.524*	*16.291*
LSK - random selection	7	426.41	60.916	12.492
tf × idf with Wordnet	7	396.44	56.634	10.739
tf × idf	7	385.7	55.099	6.649

them. The F-score values used in this analysis is averaged over 50 queries under consideration. Table 8 presents the summary of the values of the parameter that needs to be computed for two-factor analysis. The null hypothesis for the rows (number of WSs considered for link analysis) states that the row mean is equal for all rows irrespective of the number of WSs under consideration. The alternate hypothesis states that they are different. Similarly, the null hypothesis for the columns (different methods for used for link analysis) states that the column means are all equal meaning all the method perform equally well.

The values of F, $F_{critical}$ and p for rows were found to be 18.43, 2.42 and 7.55E−09 respectively. It is evident that $F > F_{critical}$ and the value of p is much smaller than $\alpha = 0.5$. Hence, it can be said that the number of WSs under consideration produces different results. From the average value presented in Table 8 it can be said that considering 10 WSs for linking is provides the best result.

The values of F, $F_{critical}$ and p for columns were found to be 18.42, 2.53 and 2.39E−09 respectively. It is evident that $F > F_{critical}$ and the value of p is much smaller than $\alpha = 0.5$. Hence, it can be said that the semantic analysis method produces different results. It is evident from the average values presented in Table 8 that support-based selection provides the best F-score value.

4 Conclusions

The proliferation of WSs makes it extremely difficult to discover WSs that can perform specialized tasks. In this paper, we have presented a novel approach to find a combination of semantically similar WSs and their compositions to satisfy a service request. An innovative concept of combining the multiple interrelated WSs that individually partially fulfill the consumer request is presented. A fusion module has also been proposed to integrate the semantically associated WSs and composite WSs for re-ranking the results.

Experiments ascertain that the proposed approach of linking multiple inter-related WSs to form WS compositions helps achieve a higher precision in WSD. The proposed method for integrating outputs by using a fusion technique further enhances the results.

In future, this work can be extended by considering the quality of service constraints to link semantically similar WSs. A WS relevancy function which considers non-functional properties of WSs such as response time, throughput, availability, usability etc. may further enhance the reliability of the WS composition and increase the overall accuracy of WSD. Thus an advanced link-mining algorithm can be developed incorporating non-functional features, which may further enhance the accuracy of the composition.

References

1. Al-Masri, E., Mahmoud, Q.H.: Qos-based discovery and ranking of web services. In: IEEE 16th International Conference on Computer Communications and Networks (ICCCN), pp. 529–534 (2007)
2. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business process execution language for web services version 1.1. http://www.ibm.com/developerworks/library/specification/ws-bpel/ (2003). Accessed 11 Aug 2009
3. Arkin, A., Askary, S., Fordin, S., Jekeli, W., Kawaguchi, K., Orchard, D., Pogliani, S., Riemer, K., Struble, S., Takacsi-Nagy, P., Trickovic, I., Zimek, S.: Web service choreography interface (wsci) 1.0. http://www.w3.org/TR/wsci (2002). Accessed 13 Aug 2007
4. Berenson, M.L., Levine, D.M.: Anova and other c-sample tests with numnerical data. In: Basic Business Statistics: Concepts and Applications, 6th edn., pp. 525–543. Prentice Hall, Englewood Cliffs (2011)
5. Bose, A., Nayak, R., Bruza, P.: Improving web service discovery by using semantic models. In: 9th International Conference on Web Information Systems Engineering (WISE 2008), pp. 366–380. Springer, Berlin, Heidelberg, Auckland, New Zealand (2008)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: The Floyd-Warshall algorithm. In: Introduction to Algorithms, 1st edn., pp. 558–565. McGraw-Hill, New York (1990)
7. Cuzzocrea, A.: Knowledge on the web: Models, algorithms, and an effective framework based on web services. In: Studies in Computational Intelligence, vol. 130. Springer, Berlin/Heidelberg (2008)
8. Denoyer, L., Gallinari, P.: The wikipedia xml corpus. ACM SIGIR Forum **40**(1), 64–69 (2006)
9. DuVander, A.: Who belongs to the api billionaires club? (2011)

10. Kuropka, D., Tröger, P., Staab, S., Weske, M.: Semantic Service Provisioning, vol. XII. Springer (2008)
11. Lamparter, S., Schnizler, B.: Trading services in ontology-driven markets. In: 2006 ACM Symposium on Applied Computing, pp. 1679–1683. ACM, Dijon, France (2006)
12. Lecue, F., Leger, A., Superieure, E.N.: Semantic web service composition based on a closed world assumption. In: Fourth IEEE European Conference on Web Services (ECOWS'06), pp. 233–242. Zurich, Switzerland (2006)
13. Leymann, F.: Web services flow language (wsfl 1.0). http://xml.coverpages.org/WSFL-Guide-200110.pdf (2001). Accessed 14 Aug 2007
14. Mendenhall, W., Beaver, R.J., Beaver, B.M.: The analysis of variance. In: Introduction to Probability and Statistics, 10th edn., pp. 451–512. Duxbery Press (1999)
15. Miller, G., Fellbaum, C., Tengi, R., Wakefield, P., Langone, H., Haskell, B.: Wordnet. http://wordnet.princeton.edu (2007). Accessed 23 Dec 2011
16. Musser, J.: Amazon web services make earnings news. http://blog.programmableweb.com/2008/01/31/amazon-web-services-make-earnings-news/ (2009)
17. Musser, J.: Open apis: state of the market . The Glue Conference (2011)
18. n.a.: Xmethods. http://www.xmethods.net/ve2/index.poSeptember (2007)
19. Nayak, R.: Facilitating and improving the use of web services with data mining. In: Research and Trends in Data Mining Technologies and Applications, pp. 309–327. Idea Group Publishers (2007)
20. Nayak, R.: Using data mining in web services planning, development and maintenance. Int. J. Web Serv. Res. **5**(1), 62–80 (2008)
21. Nayak, R., Lee, B.: Web service discovery with additional semantics and clustering. In: 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), pp. 555–558. IEEE Computer Society, Silicon Valley, USA (2007)
22. Rao, J., Kungas, P., Matskin, M.: Application of linear logic to web service composition. In: International Conference on Web Services (ICWS '03), pp. 3–10. CSREA Press, Las Vegas, USA (2003)
23. Richards, D., Splunter, S., Brazier, F., Sabou, M.: Composing web services using an agent factory. In: Extending Web Services Technologies, vol. 13, pp. 229–251. Springer, US (2004)
24. Schumacher, M., Tim Van, P., Ion, C., Alexandre de, O., Boi, F.: Discovering semantic web services in federated directories (2007)
25. Thatte, S.: Xlang: Web services for business process design. http://xml.coverpages.org/XLANG-C-200106.html (2003). Accessed 12 Aug 2010
26. Vizard, M.: Developers must focus on link ing web services to boost net collaboration (2007). http://www.infoworld.com/s/op/xml/01/03/19/010319opvizard.html (2007). Accessed 12 Aug 2007

Exploiting Terrain Information for Enhancing Fuel Economy of Cruising Vehicles by Supervised Training of Recurrent Neural Optimizers

Mahmoud Abou-Nasr, John Michelini and Dimitar Filev

Abstract In this chapter, we present a novel data driven approach based on supervised training of feed forward neural networks for solving nonlinear optimization problems. Then we extend the approach to approximate the solution of deterministic, discrete dynamic programming problems by using recurrent networks. We apply this data driven methodology on a real-world fuel economy application in which we train a neural optimizer to prescribe the optimum cruise speed that minimizes fuel consumption, based on the instantaneous and a limited history of the vehicle speeds and road grades, with no a priori knowledge of the future path. The optimizer is tested in simulation on novel road segments. In simulation tests, the optimizer prescribed grade based modulated speed, has achieved about 8–10.6 % fuel savings over driving with constant cruise speed on the same roads, out of which 3.7–10.6 % were due to exploiting the road grades.

1 Introduction

There is a growing interest especially in the automotive industry, in data driven applications and methodologies, in which algorithms are learned directly from the data acquired in real-life situations e.g. from a machine in a factory for process control or an engine in a vehicle for diagnostics and optimal control. These applications are opportunities for the advancement of more intelligent learning algorithms based on neural networks, support vector machines, fuzzy modeling, etc. Neural networks have been employed in many engineering, scientific, medical and financial applications, mainly for their remarkable ability of universal function approximation.

M. Abou-Nasr (✉) · J. Michelini · D. Filev
Research and Advanced Engineering, Research & Innovation Center, Ford Motor Company, Dearborn, MI, USA
e-mail: mabounas@ford.com

J. Michelini
e-mail: jmichel1@ford.com

D. Filev
e-mail: dfilev@ford.com

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_17

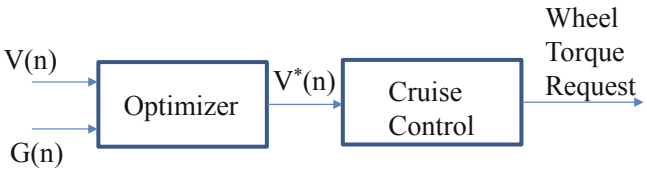


Fig. 1 Block diagram of the system which is comprised of an optimizer neural network that computes the optimum cruise set point for the cruise control subsystem

Werbos in [11, 12] has proposed a framework for neural networks beyond function approximation, in feedback control and approximate dynamic programming (ADP). In particular, the framework for solving ADP problems with adaptive critics neural networks, was a major milestone in the literature of artificial networks [1]. The adaptive critics networks overcome the computational complexities that the dynamic programming formulations of real-world optimal control problems suffer from. Since then, other variants of adaptive critics networks have been proposed and studied, a good account of these networks can be found in [10].

Other frameworks for solving the approximate dynamic programming problem which are based on recurrent neural networks have been presented in [3] and [9]. In [3], the authors utilized recurrent neural networks in the problem of energy managment in a parallel hybrid electric vehicle with an ultra-capacitor. They have trained a recurrent neural controller that prescribes the optimal power split between the engine and the electric motor in that vehicle, which reduces fuel consumption and adheres to the constraints on charging and discharging of the ultra-capacitor. In [9], a recurrent neural network controller was proposed for improving the fuel efficiency of a Toyota Prius hybrid electric vehicle while adhering to the constraints on the battery state of charge.

In this chapter, we present a generic data driven framework for solving nonlinear optimization problems and for approximating the solution of deterministic, discrete dynamic programming problems with neural networks, based on supervised training. We employ this data driven framework in a real-world vehicle cruise control application, in which we train a recurrent neural optimizer to reduce the engine's fuel consumption over a trip utilizing the road grade information. We pose the problem as a dynamic programming problem with constraints and we obtain its solution by supervised training of the recurrent neural optimizer. The trained optimizer does not assume a priori knowledge of the road (terrain preview) that is being travelled nor does it suppose any explicit vehicle model. Rather, it only assumes the knowledge of the instantaneous road grade, a limited history of the previous vehicle speeds, vehicle fuel consumption, and road grades. It utilizes this information to modulate the cruise control set point around the initial speed that is set by the driver, to ultimately achieve better fuel economy. In this work, we constrain the average modulated vehicle speed to be equal to the initial cruise control speed set by the driver.

A block diagram of the neural optimizer and its interface to the vehicle cruise control subsystem is depicted in Fig. 1.

In Fig. 1, V is the vehicle speed, G is the road grade, V^* is the optimum speed (speed at which maximum fuel savings is achieved) and n is the unit of discretization.

The rest of this chapter is organized as follows. In Sect. 2, we state the general nonlinear optimization problem and we present our generic solution with supervised training of feed forward neural networks for this class of problems. In Sect. 3, we employ the same framework presented in Sect. 2 to solve dynamic programming problem with recurrent networks. In Sect. 4, with the methodology described in Sect. 3, we derive our neural optimizer for fuel economy utilizing the road grades. In Sect. 5, we discuss the experimental network training and testing procedures and present sample results. We conclude with a summary and some remarks in Sect. 6.

2 Solving Nonlinear Optimization Problems with Supervised Training of Neural Networks

An instance of the nonlinear optimization problem can be stated as follows:

$$\begin{aligned} &\min y = f(\mathbf{x}) \\ &s.t. \\ &c = g(\mathbf{x}) = 0, \end{aligned} \tag{1}$$

where,
$f(\mathbf{x})$ and $g(\mathbf{x})$ are nonlinear functions of $\mathbf{x}$ respectively,
$\mathbf{x}$ is a vector $\{x_1, x_2, \ldots, x_n\}$,
c is a constraint, n is an integer,
bold typeface letters denote vectors.

This statement of the problem with equality constraints is not restrictive, as generally inequality constraints can be transformed to equality constraints.

Assuming that the function $f(\mathbf{x})$ can be represented by its neural network approximation, the first step in solving this optimization problem is to obtain this representation through the process of learning on pairs of $\mathbf{x}$ and $f(\mathbf{x})$. Consecutively, we train an optimizer which is another neural network that initially has random weights connecting its nodes. The objective of the training is to minimize a cost function that generally can be represented as follows:

$$J = O^2 + \lambda c^2, \tag{2}$$

where,
J is the optimization objective function.
O is the output of the neura network that is approximating the function f,
λ is a Lagrange multiplier,
c is an equality constraint.

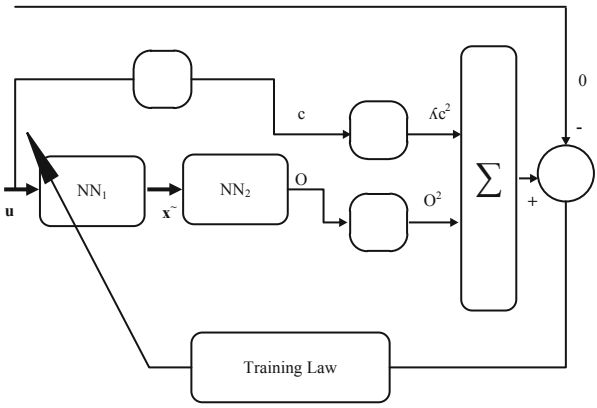


Fig. 2 Block diagram illustrating generic optimization with supervised training of neural networks

The objective function J is quadratic and its minimum is ≥ 0. To minimize the function the neural optimizer has to find $\mathbf{x}^*$ in the range $\mathbf{j} \leq \mathbf{x} \leq \mathbf{k}$, such that the value of $J(\mathbf{x}^*)$ is the closest to zero, where $\mathbf{j}, \mathbf{k} \in \mathrm{R}$ and $\mathbf{k} > \mathbf{j}$. We illustrate the neural optimization process in Fig. 2.

In Fig. 2, NN_1 is the neural optimizer which initially has random weights connecting its nodes. NN_2 is a trained network that represents the function $f(\mathbf{x})$ with fixed weights connecting its nodes. $\mathbf{u}$ is is a vector that has values in the range: $\mathbf{j} \leq \mathbf{u} \leq \mathbf{k}$. $\tilde{\mathbf{x}}$ is the output of the optimizer network NN_1 which is constrained in the range $\mathbf{j} \leq \tilde{\mathbf{x}} \leq \mathbf{k}$ during training. O is the output of NN_2. The objective function is formed by adding O^2 and $\lambda \mathrm{c}^2$.

As the training proceeds, the weights of the NN1 are adapted and the value of the objective function is minimized as it tries to approach zero. At the end of the training, when the objective function J converges to its minimum value, the output of NN1 is $\mathbf{x}^*$, at which value the function $f(\mathbf{x}^*)$ has its minimum.

If we visualize Fig. 2 as representing a network of networks with its rightmost node as its output, then essentially the optimization problem is transformed into a supervised learning problem, in which the supervised learning target is zero.

3 Solving Dynamic Programming Problems with Supervised Training of Recurrent Neural Networks

The same supervised learning based technique can be applied to approximate the solution of deterministic, discrete dynamic programming problems. In this class of problems the objective function to be minimized (or maximized), is the sum of a utility function over a horizon N of sequential events.

$$J = \sum_{n=1}^{N} U(\mathbf{x}(n), \mathbf{u}(n)). \tag{3}$$

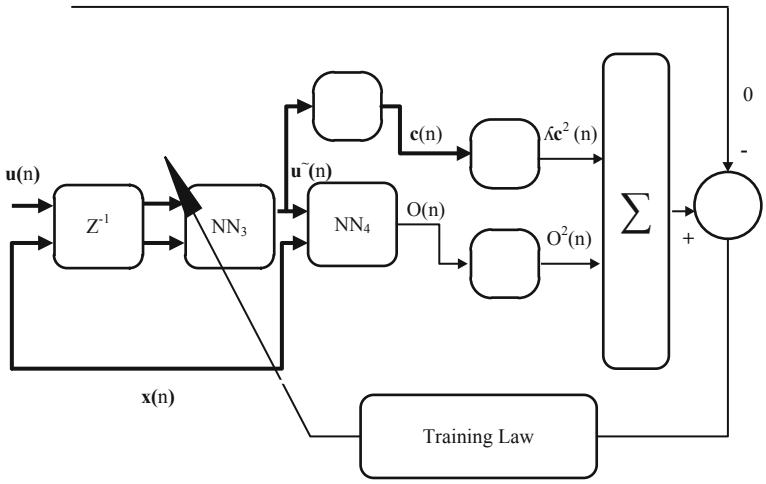


Fig. 3 Block diagram illustrating the solution of a dynamic programming problem with supervised training of recurrent neural networks

Solving the problem means finding the sequence (policy) of controls/decisions that minimizes the sum in (3) as a function of the system states according to the Bellman's principle of optimality recursion (4) where, the angular brackets denote expectation.

$$J(\mathbf{x}(n)) = \max_{\mathbf{u}(n)} \left(U(\mathbf{x}(n), \mathbf{u}(n)) + \langle J(\mathbf{x}(n+1)) \rangle \right). \tag{4}$$

The intuitive meaning of the Bellman's principle of optimality is that in order for the policy to be optimal, all the controls/decisions in the policy have to be optimal. Thus the problem of minimizing the sum of the utility function over all future events can be reduced in dynamic programming to optimizing for one event ahead. Nevertheless, it is seldom practical to obtain solutions to real-world problems with dynamic programming as they suffer from what is known in the literature as the "curse of dimensionality". We can use the same framework that we employed for static optimization in Sect. 2, to solve these dynamic programming problems with a couple of notable exceptions. We will employ recurrent neural networks for representing the utility function and for the optimizer. We also added a unit delay as the leftmost block of the diagram.

The first step for solving a dynamic programming problem with supervised training is to train the NN_4 in Fig. 3. NN_4 is recurrent in this case to represent the utility function U as a function of $\mathbf{x}(n)$, $\mathbf{u}(n)$. For recurrent network modeling, the training data has to be sequential. The second step is to train the neural optimizer NN_3 which is also a recurrent network that has two vector inputs: $\mathbf{u}(n-1)$ and $\mathbf{x}(n-1)$ and one output: $\tilde{\mathbf{u}}(n)$ during training. As the supervised training proceeds, the objective function that is comprised of the sum: $O^2(n) + \lambda \mathbf{c}^2(n)$, approaches the zero target,

thus trying to minimize the utility function and satisfy the constraints at each sequential event. In the dynamic programming problems, the supervised training is utilizing the Bellman's principle of optimality and is essentially approximating the minimum sum of the utility function over the horizon of events, by minimizing the utility function one event ahead. At the end of the training, $\tilde{\mathbf{u}}(n)$ converges to $\mathbf{u}^*(n)$ which is a sequence of controls/decisions that approximates the minimum of the sum of the utility function in (3).

This process for solving dynamic programming problems converts from what is essentially a reinforcement learning problem into a supervised learning problem. In reinforcement learning, no targets are given for the output to follow in the training process. Instead, we introduce a measure by which we can evaluate the output at each step of training. As we did in Sect. 2, if we visualize Fig. 3 as representing a network of networks with its rightmost node as its output, then we have a supervised training problem in which the target (desired output) of the training is zero.

Apparently, this approach is not dependent on the choice and the method of learning of the plant model NN_4. Although we have considered a neural model, it can be replaced by any alternative "black box" type or first principle based model providing an adequate approximation of the plant dynamics.

4 Example Application: Enhancing the Fuel Economy of Cruising Vehicles by Exploiting Terrain Information

The quest for ever improving fuel economy has stimulated research aiming at developing control strategies that exploit the characteristics of a particular terrain, the traffic conditions on particular routes or during a specific part of the day [4, 7] and [8]. In [8], the authors have presented methods based on stochastic dynamic programming (SDP), for deriving control policies that enhance the fuel economy of a cruising vehicle by exploiting the road grades and the traffic conditions. They have also presented a graph depicting a strong functional dependency of the fuel economy on the vehicle speed and the road grade.

We demonstrate the method and framework presented in the previous sections on an application of optimal control of the vehicle speed of a cruising vehicle. Our objective is to develop an algorithm optimizing the fuel economy of the vehicle without significantly affecting the travel time by utilizing the terrain information—the instantaneous road grade. No additional assumptions of a known route or road geometry preview along the driving path are considered. We utilize the framework for approximate dynamic programming based optimization through supervised training of recurrent neural networks described in Sect. 3 to develop a novel approach to the problem of optimal speed control under the assumptions that were stated above. Our solution exploits the idea of training a neural optimizer that instantaneously modulates the vehicle speed about the initial cruise set point, based on the current vehicle speed and road grade, and a limited history that is comprised of few past vehicle speeds and road grades. The goal is to minimize the fuel consumption over

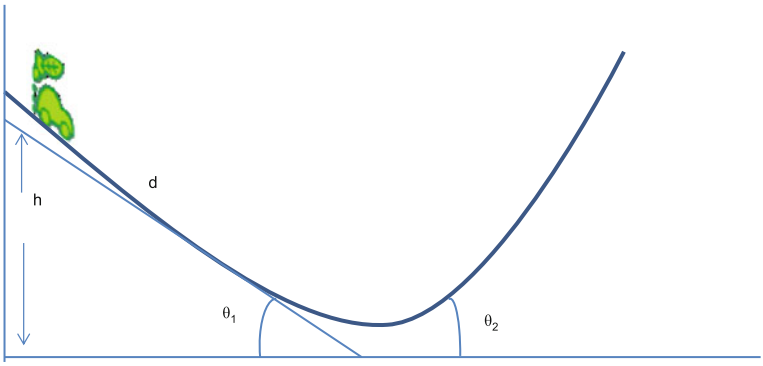


Fig. 4 A section of a road depicting a negative road grade θ_1 and a positive road grade θ_2

an entire trip as compared to the fuel consumption obtained while driving with a constant speed that was initially set by the driver.

Figure 4 depicts a section of a road with a negative road grade θ_1 and a positive road grade θ_2. The road grade is estimated by the following percentage:

$$G = \frac{h}{d} \times 100\%. \tag{5}$$

In (5), d is the distance travelled by the vehicle per sample, h is the road height per sample defined as:

$$h_2 - h_1, \tag{6}$$

h_1 is the road height at the beginning of the sample and h_2 is the road height at the end of the sample.

The problem can be stated as follows. Given a sequential set that is comprised of N samples of road grades $G(n)$ and vehicle speeds $V(n)$, find the sequence of vehicle speeds $V^*(n)$ that minimizes fuel consumption over the path, while maintaining the overall average speed.

This problem can be posed as a dynamic programming problem in which the utility function to be minimized is the sum of the fuel consumption over a horizon of N samples representing the duration of the trip.

$$\begin{aligned} &J = \sum_{n=1}^{N} F(n) \\ &s.t. \\ &\tfrac{1}{N} \sum_{n=0}^{N} V(n) = V_{\text{sp}} \end{aligned} \tag{7}$$

In (7), F is the fuel consumption rate, which is a function of the road grades and the vehicle speed (8)

$$F(n) = f(G(n), V(n)) \tag{8}$$

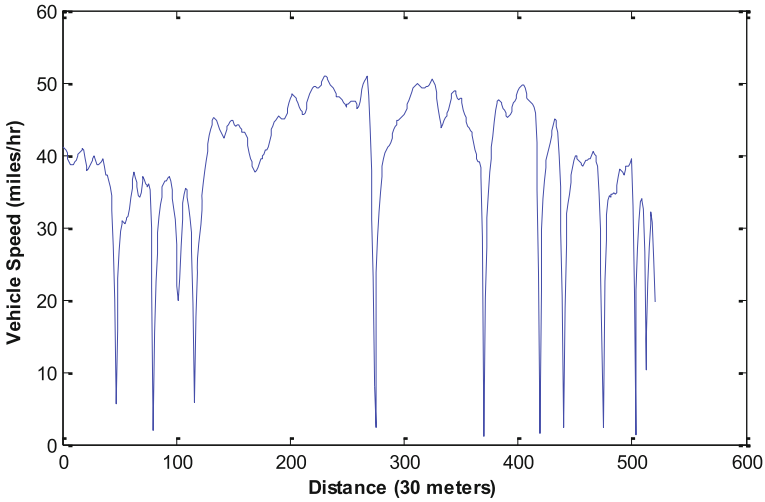


Fig. 5 Sample training data showing the vehicle speed on a segment of a freeway during rush hour

$G(n)$ is the road grade and $V(n)$ is the vehicle speed at sample n. V_{sp} is the cruise control set point which is set by the driver.

The first step in the generic framework that is presented in Sect. 3 for solving an approximate dynamic programming problem is to train a recurrent neural network, to represent F as defined in (8). For this purpose we employed a high fidelity vehicle model that was validated against experimental data. We generated sequential data representing equidistant samples while driving on a freeway. The vehicle speed, the road grade and fuel consumption rate are recorded at each sample. Instances of the data in a training set are shown in the graphs of Figs. 5, 6 and 7.

The second step is to train a recurrent neural network as an optimizer.

As depicted in Fig. 8, we minimize the total fuel consumption over the trip by customizing the generic solution developed in Sect. 3 for dynamic programming problems to the specifics of this problem. In this problem, we only have one component in the vector of controls $\mathbf{u}(n)$ which is the speed $V(n)$ and one component of the state vector $\mathbf{x}(n)$ which is the road grade. The recurrent neural fuel model NN_4 is driven by two inputs, one is the output of the neural optimizer NN_3 and the other input is the road grade $G(n)$. As the supervised training proceeds, the objective function that is comprised of the sum $F^2(n) + \lambda c^2(n)$ approaches the zero training target, thus minimizing the utility function and satisfying the constraint at each sample of the training sequence. As mentioned before, by the Bellman's principle of optimality we are also optimizing the sum of the utility function over the trip horizon (the number of samples in the training set). By the end of the training, the output of the neural optimizer $\tilde{V}(n)$ will converge to the optimum vehicle speed sequence $V^*(n)$.

The constraint for this problem specified in (7) is implemented by subtracting V_{sp} (the vehicle cruise control set point) from a recursively filtered $\tilde{V}(n)$. The exponentially weighted moving average filter (EWMA) is described in (9):

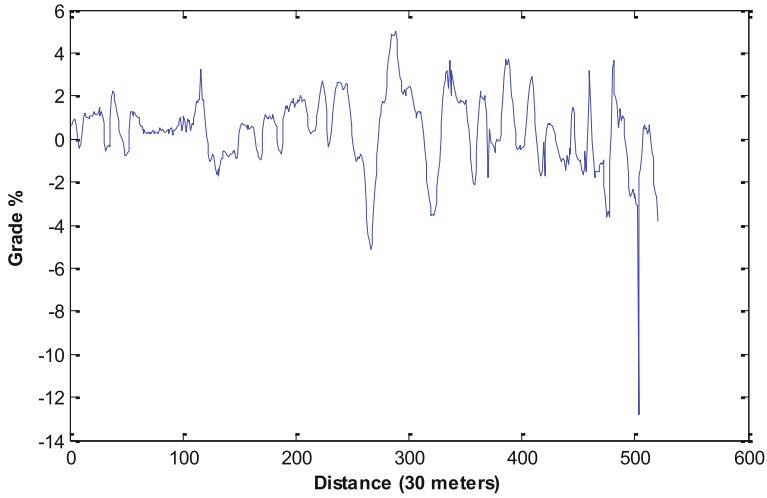


Fig. 6 Road grades for the trip depicted in Fig. 5

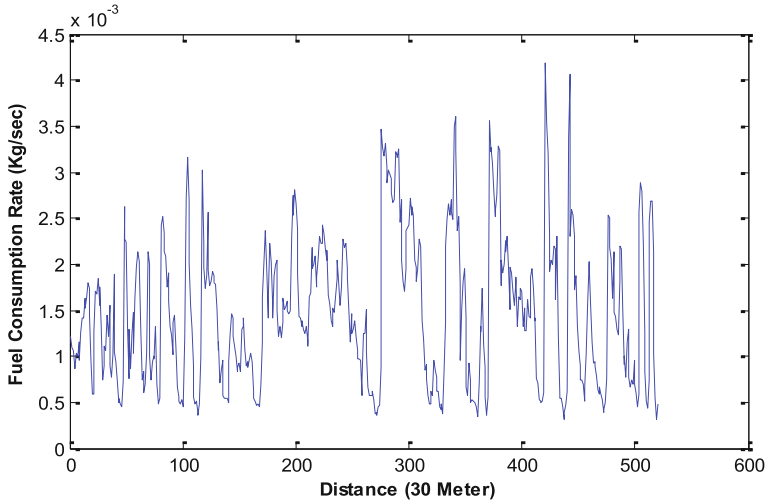


Fig. 7 Fuel consumption rate for the trip depicted in Fig. 5

$$\bar{V}(n) = \alpha \tilde{V}(n) + (1 - \alpha)\bar{V}(n - 1), \tag{9}$$

$\bar{V}(n)$ is the EWMA average of $\tilde{V}$ over $((1/\alpha) - 1)$ samples.

Some experimentation is needed to choose a proper value of the Lagrange multiplier λ but it is straight forward.

One advantage of the neural solution to the approximate dynamic programming problem besides its speed over actual dynamic programming is that we obtain a trained neural optimizer that can be implemented in a vehicle with software. For

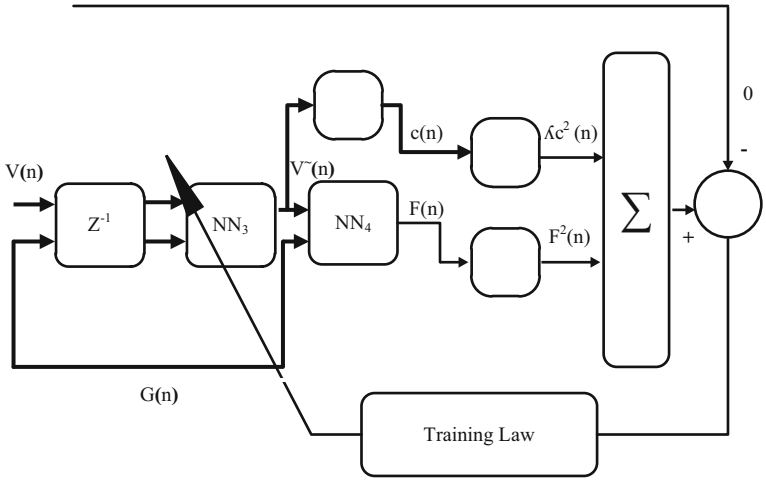


Fig. 8 Block diagram of minimization of fuel consumption over a trip by supervised training of recurrent neural networks

this chapter we test our optimizer with a data set generated by the model that we employed to generate the training data set and was withheld for testing.

5 Experimental Procedures and Results

In this section we first describe the architecture of the recurrent neural networks which were used in modeling the fuel rate of consumption as a function of the grade and vehicle speed, and the optimizer. Then we briefly describe the training method. Next we show the results of testing the neural optimizer on a data set that was not part of the training set. We evaluate the neural optimizer by comparing the fuel consumption over the trips in the testing set with the modulated cruising speeds prescribed by the optimizer to driving the same trip with the speed that was originally set for the cruise control.

5.1 Architecture of the Recurrent Neural Networks in this Work

As mentioned previously, the network NN4 employed for modeling the rate of fuel consumption as a function of the road grade and the vehicle speed is a recurrent network. It has two inputs, six sigmoidal recurrent nodes in the first layer, three sigmoidal nodes in the next (hidden) layer and one output representing the modeled fuel consumption rate. Each recurrent node has its output fed back to its input and to the inputs of each other node in the first layer with a unit delay. We used the same architecture for the optimizer network.

5.2 Network Training

We used the extended Kalman filter (EKF) training [2] for weight adaptation (training) of the fuel model and for the supervised training of the optimizer. The optimizer converges to the optimum set of vehicle speeds on the training set road segments within $\sim$ 20 epochs of training which is one of the advantages of this approximate dynamic programming method.

5.3 Neural Optimizer Testing Procedure

To evaluate the trained optimizer we conducted the following tests with data representing road segments that were not part of the optimizer training set. In each test, we set the input $V(n)$ of the trained optimizer NN_3 to a constant value representing a cruise control set speed and provide a sequence representing the road grades in this particular road segment to the other input of the optimizer: $G(n)$.

$$V(n) = v, \quad \text{for all } n = 1 : N \tag{10}$$

where,

$$v \rightarrow \mathrm{R}, \quad v \,\varepsilon\, [v_{\min}; v_{\max}].$$

$$G(n) = G_{\mathrm{RS}_i}(n) \tag{11}$$

where,

$G_{\mathrm{RS}_i}(n)$ is the grade sequence for road segment i.

The output of the optimizer which is the sequence of optimum speeds $V^*_{S_i v}(n)$ is then used by our high fidelity model (that generated the data) to estimate the fuel consumption over this road segment that we will denote as F^*. The high fidelity model also estimates two other fuel consumptions. The first is the consumption when driving with the set constant cruise speed v that we will denote as F_v. The second is the consumption when driving with a constant speed $\bar{v}$ that we will denote as $F_{\bar{v}}$. The speed $\bar{v}$ is the average speed over the trip and is calculated as follows:

$$\bar{v} = \frac{1}{N} \sum_{n=1}^{N} V^*_{s_i v}(n). \tag{12}$$

Finally we compute the following performance measures:

$$E_T = \frac{(F_v - F^*)}{F_v} \times 100\%, \tag{13}$$

where,

E_T is the percentage of fuel saved when driving with $V^*_{s_i v}$ compared to the fuel consumption when driving with v(the cruise set speed).

Ideally, the average of modulated optimum speed $\bar{v}$ should equal v. However in practice they may be slightly different. One may question whether the achieved fuel savings were due to the average speed reduction or due to exploiting the road grades. In order to estimate the relative contribution due to road grades E_d and the relative contribution due to average speed reduction $E_{\bar{v}}$ to the total fuel savings, we also define the following:

$$E_d = E_T - E_{\bar{v}}, \tag{14}$$

where,

$$E_{\bar{v}} = \frac{(F_v - F_{\bar{v}})}{F_v} \times 100\%. \tag{15}$$

5.4 Experimental Results

The following figures show the results of testing the optimizer on three different road segments. The cruise set speed was 35.8, 40.26 and 51.45 miles/h respectively.

On road segment 1, the cruise set speed v was 35.8 miles/h. The optimum modulation of the speed based on the road grades of segment 1: $V^*_{1v}(n)$ is shown in the top panel of Fig. 9 as dashed and solid lines respectively. The inputs to the optimizer, vehicle speed and the road grade were updated at a rate of one sample per 30 m. The road grades of segment1: $G_{\mathrm{RS}_1}(n)$ is shown in the bottom panel of Fig. 9. The distance travelled in segment 1 is ∼9.7 km. In simulation, when the vehicle was driven at the set cruise speed for the duration of the trip the fuel consumption F_v was 0.389 kg. When it was driven with the speed prescribed by the optimizer the fuel consumption F^* was 0.348 kg. The average modulated speed $\bar{v}$ from (12) is 35.3 mile/h, which is less than the cruise set speed. As described before we also obtained from the model the fuel consumption when driving the vehicle at the constant speed $\bar{v}$ denoted as $F_{\bar{v}}$ which was 0.383 kg. From (11), E_T (the percentage of fuel saved by driving with the optimizer prescribed speed) was 10.4 %. From (15) $E_{\bar{v}}$ (the percentage of fuel saved by driving at$\bar{v}$) is 1.5 %. From (14) the percentage contribution of the grades E_d to the total fuel saved is 8.9 %.

In the two other tests we employed the same sampling rates for the inputs of the optimizer. The graphs depicting the tests also use the same conventions in the top and bottom panels.

On road segment 2, the cruise set speed was 40.26 miles/h. The distance travelled was ∼39 km. In this trip, fuel consumption when driving with the set cruise speed F_v was 1.72 kg. The fuel consumption when driving with the speed modulation prescribed by the optimizer F^* was 1.53 kg. The average modulated speed was

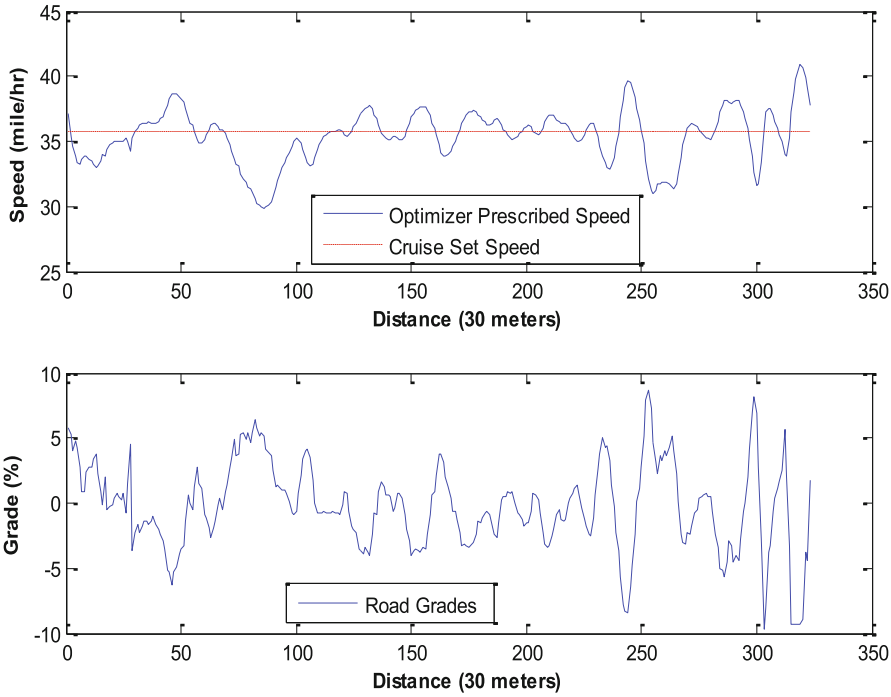


Fig. 9 Road segment 1. Top panel: The optimizer prescribed optimum speed modulation and the cruise control set speed. Bottom panel: Grades of road segment 1. Cruise set speed: 35.8 mile/h

equal to the cruise set speed 40.26 miles/h. Hence $F_{\bar{v}}$ was 1.72 kg, the same as F_v. E_T was 10.6 %, $E_{\bar{v}}$ was 0 % (no speed reduction) and E_d was 10.6 %. All the fuel saving in this case was due to exploiting the grades.

On road segment 3, the cruise set speed was 51.45 miles/h. The distance travelled was $\sim$30 km. F_v was 1.6 kg. F^* was 1.47 kg. The average modulated speed was 50.3 mile/h, which is less than the cruise set speed. $F_{\bar{v}}$ was 1.53 kg. E_T was 8 % and $E_{\bar{v}}$ was 4.3 %. The fuel savings due to exploiting the grades E_D was 3.7 %. We summarize the results in the following table.

We should note that occasionally the optimal speed profile modulated by the optimizer may exceed the maximal speed set by the driver, i.e. the cruise control set speed in Figs. 9, 10 and 11, a situation that might not be desirable given the assumption that maximal speed is defined by the cruise controller set-point. This problem can be alleviated by clipping the optimizer output values that are over the set-point of the cruise controller with the expectation that some of the optimization benefits will potentially be lost:

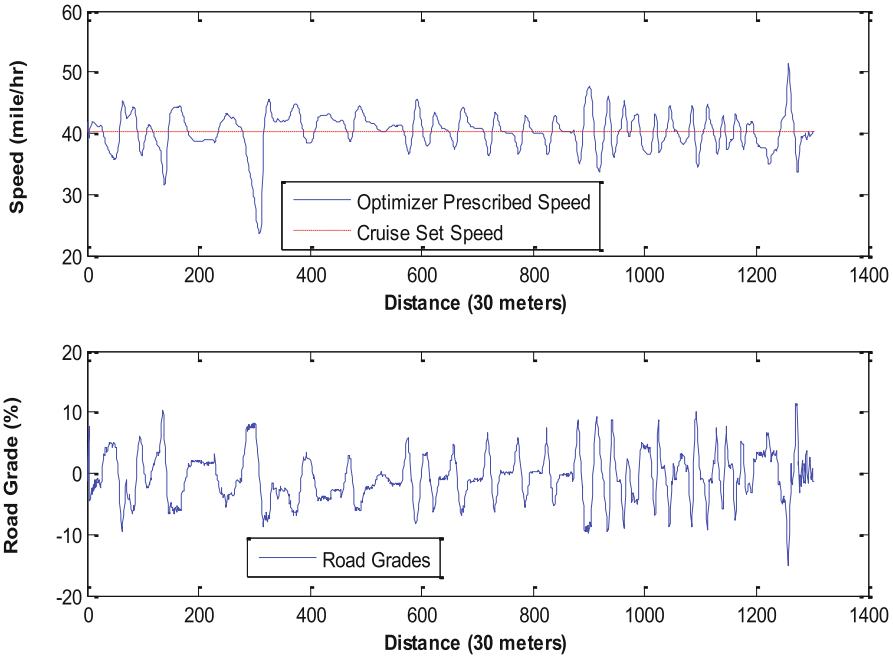


Fig. 10 Road segment 2. Top panel: The optimizer prescribed optimum speed modulation and the cruise control set speed. Bottom panel: grades of road segment 2. Cruise set speed: 40.26 mile/h

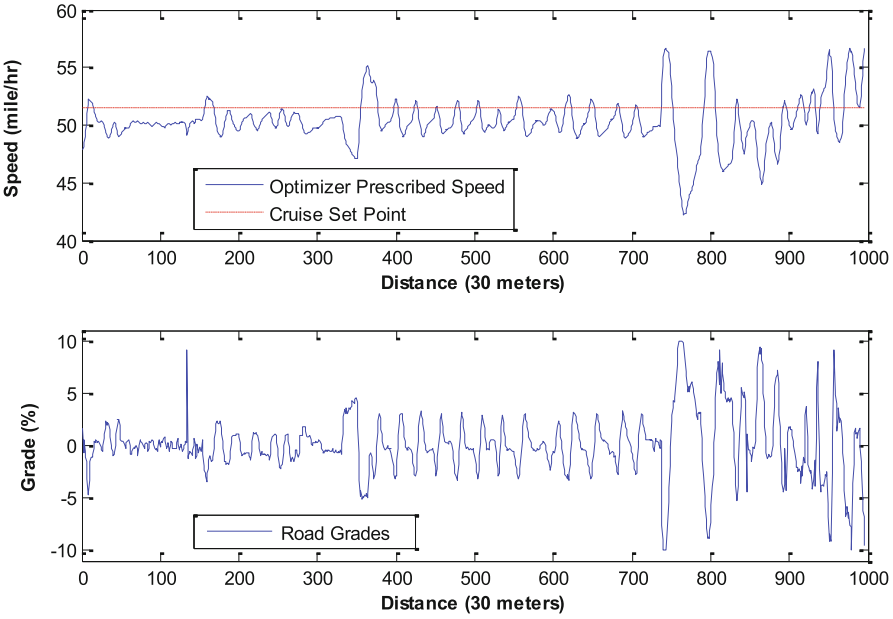


Fig. 11 Road segment 3. Top panel: The optimizer prescribed optimum speed modulation and the cruise control set speed. Bottom panel: Grades of road segment 3. Cruise set speed: 51.45 mile/h

Table 1 Summary of test results

Road segment	Cruise speed (mile/h)	E_T (%)	$E_{\bar{v}}$ (%)	E_D (%)
1	35.8	10.4	1.5	8.9
2	40.26	10.6	0	10.6
3	51.45	8	4.3	3.7

$$V^*(n) := (V_{\text{sp}} + V^*(n) - |V_{\text{sp}} - V^*(n)|)/2. \tag{16}$$

Testing results show that the speeds prescribed by the optimizer, have saved fuel over the constant cruise speed set by the driver. Also, considerable fuel savings were obtained by leveraging the road grades. Another promising aspect is that although the optimizer was not trained on the road grade patterns in the testing road segments, nevertheless it was able to prescribe fuel saving speeds. This has practical implications as it may reduce the need for retuning or retraining the optimizer in the field.

6 Summary and Conclusion

In this chapter, we presented a data driven generic framework for solving nonlinear optimization problems with constraints and for approximating the solution of deterministic, discrete dynamic programming problems with constraints. In this framework, the optimization and the dynamic programming problems are posed as supervised training problems in which the quadratic objective functions are trained to converge to a target of zero.

We demonstrated the data driven methodology in a cruise control application in which our objective was to find the optimum speed modulation based on the road grades, that minimizes the fuel consumption over a trip, with the constraint that the average of the speed modulation should equal the cruise speed set by the driver.

We have tested the recurrent neural optimizer that we obtained by supervised training in simulation on novel road segments that were not part of its training set. The testing results showed that modulated speeds that were prescribed by the recurrent neural optimizer have achieved better fuel economy compared to driving with constant cruise speeds, and that considerable fuel savings were due to leveraging the road grades, as shown in Table 1.

The trained neural optimizer has prescribed fuel savings speed modulation on novel road segments based on the current vehicle speed, road grade and a limited history of their past values in its recurrent nodes. This has practical implications as it may reduce the need for returning or retraining the optimizer in the field.

The recurrent neural network based approach presented in this chapter for optimizing the vehicle cruising speed is different from the approaches presented in [7, 8] which employ stochastic dynamic programming. It is also worth noting that our modeling approach does not explicitly model the gear shifts as inputs to the model, rather it is a approach and the gear shifts are implicitly inferred from the data in the

process of training. Also, it is different from the approaches in [6] and [5] for heavy trucks which employ dynamic programming and hence assume the knowledge of path and road topology over the trip. Our approach only assumes the knowledge of the instantaneous road grade. No additional assumptions of a known route or road geometry preview along the driving path are considered.

Although we did not compare the solution obtained by our approach to a solution that would be obtained by dynamic programming, in [5], the authors have compared solutions obtained by neural networks ADP to DP solutions and have shown that it achieved ∼90 % of the savings obtained by the DP solution in the context of a hybrid vehicle with an ultra-capacitor.

References

1. Balakrishnan, S.N., Ding, J., Lewis, F.L.: Issues on stability of adp feedback controllers for dynamical systems. IEEE Trans. Syst. Man Cybern. B Cybern. **38**(4), 913–917 (2008)
2. Feldkamp, L.A., Puskorius, G.: A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering and classification. In: Proceedings of the IEEE, vol. **86**, pp. 2259–2277 (2009)
3. Feldkamp, L., Abou-Nasr, M., Kolmanovsky, I.: Recurrent neural network training for energy management of a mild hybrid electric vehicle with an ultra-capacitor. In: IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, pp. 29–36 (2009)
4. Hellstrom, E., Froberg, A., Nielsen, L.: A real-time fuel optimal cruise controller for heavy trucks using road topography information. Tech. Rep. 2006-01-0008, SAE Paper (2006)
5. Hellstrom, E., Ivarsson, M., Aslund, J., Nielsen, L.: Look-ahead control for heavy trucks to minimize trip time and fuel consumption. Control Eng. Pract. vol. 17, 245–254 (2009)
6. Huang, W., Bevly, D.M.: Evaluation of 3d road geometry based heavy truck fuel optimization. Int. J. Veh. Autonom. Syst. vol. 8, 39–55 (2010)
7. Kolmanovsky, I., Filev, D.: Stochastic optimal control of systems with soft constraints and opportunities for automotive applications. In: Proc. IEEE Conference on Control Applications, pp. 1265–1270 (2009)
8. Kolmanovsky, I.V., Filev, D.P.: Terrain and traffic optimized vehicle speed control. In: Proceedings of 6th IFAC Symposium on Advances in Automotive Control (2010)
9. Prokhorov, D.: Toyota prius hev neurocontrol. In: International Joint Conference on Neural Networks, pp. 2129–2134 (2007)
10. Prokhorov, D.V., Wunsch, D.C.: Adaptive critic designs. IEEE Trans. Neur. Netw. 8(5), 997–1007 (1997)
11. Werbos, P.J.: Neural networks for control and system identification. In: Proceedings of the IEEE Conference on Decision Control, vol. **1**, pp. 260–265 (1989)
12. Werbos, P.J.: Approximate dynamic programming for real-time control and neural modeling. In: White, D.A., Sofge, D.A. (eds.) Handbook of Intelligent Control, chap. 13. Van Nostrand Reinhold, New York (1992)

Exploration of Flight State and Control System Parameters for Prediction of Helicopter Loads via Gamma Test and Machine Learning Techniques

Catherine Cheung, Julio J. Valdés and Matthew Li

Abstract Accurate estimation of helicopter component loads is an important goal for life cycle management and life extension efforts. In this research, estimates of helicopter dynamic loads were achieved through a combination of statistical and machine learning (computational intelligence) techniques. Estimates for the main rotor normal bending (MRNBX) loads on the Sikorsky S-70A-9 Black Hawk helicopter during two flight conditions (full speed forward level flight and rolling left pullout at 1.5g) were generated from an input set comprising 30 standard flight state and control system (FSCS) parameters. Data exploration using principal component analysis and multi-objective optimization of Gamma test parameters generated reduced subsets of predictors. These subsets were used to estimate MRNBX using neural network models trained by deterministic and evolutionary computation techniques. Reasonably accurate and correlated models were obtained using the subsets of the multi-objective optimization, also allowing some insight into the relationship between MRNBX and the 30 FSCS parameters.

1 Introduction

Operational requirements are significantly expanding the role of military helicopter fleets in many countries. This expansion has resulted in helicopters flying missions that are beyond the design usage spectrum. Therefore, the current life usage estimation for the fatigue critical components may no longer have the required low probability of failure. Due to this change in usage, there is a need to monitor individual aircraft usage to compare with the original design usage spectrum in order to more accurately determine the life of critical components. One of the key elements

C. Cheung (✉) · J. J. Valdés · M. Li
National Research Council Canada, 1200 Montreal Rd, Ottawa, ON K1A 0R6, Canada
e-mail: cathy.cheung@nrc-cnrc.gc.ca

J. J. Valdés
e-mail: julio.valdes@nrc-cnrc.gc.ca

M. Li
e-mail: matthew.li@nrc-cnrc.gc.ca

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_18

of tracking individual aircraft usage and calculating component retirement times is accurate determination of the component loads.

The operational loads experienced by rotary-wing aircraft are complex due to the dynamic rotating components operating at high frequencies. As a result of the large number of load cycles produced by the rotating components and the wide load spectrum experienced from the broad range of manoeuvres, the fatigue lives of many components can be affected by even small changes in loads. While measuring dynamic component loads directly is possible, traditionally through slip rings or telemetry systems, these measurement methods are not reliable and are difficult to maintain. Therefore, an accurate and robust process to estimate these loads indirectly would be a practical alternative or supplement to the existing methods. Load estimation methods can make use of existing aircraft sensors, such as standard flight state and control system (FSCS) parameters, to minimize the requirement for additional sensors and consequently the high costs associated with instrumentation installation, maintenance and monitoring.

There have been a number of attempts at estimating these loads on the helicopter indirectly with varying degrees of success [17]. Preliminary work exploring the use of various computational intelligence techniques for estimating helicopter loads showed that reasonably accurate and correlated predictions for the main rotor normal bending could be obtained for forward level flight at full speed using only a reduced set of FSCS parameters [23, 24]. The results from these models allow some domain expert examination of the relationships between the flight state and control system parameters and component loads, enabling a better understanding of the loads in the critical components. In particular, a better understanding of the specific flight state parameters that are most relevant for particular loads in airframe and dynamic components of the helicopter has been obtained.

This article presents a more comprehensive analysis of the results obtained during data exploration and modeling, examining the relationship between the flight state and control system parameters and the main rotor normal bending on the Australian Army Black Hawk helicopter (shown in Fig. 1) in two flight conditions (forward level flight at full speed and rolling left pullout at $1.5g$). The objectives of this work were as follows: (i) to extend the scope and complexity of the predictions to include more flight conditions, (ii) to evaluate the ability of multi-objective genetic algorithms (MOGA) with the Gamma test to explore the data and identify subsets with high predictive potential, (iii) to extract information from the data that could enable a better understanding of the physical process of the input/output relationship, and (iv) to compare the prediction results obtained by the reduced subset found by the MOGA and Gamma test with those of the full suite of input parameters.

This article is organized as follows: Sect. 1 provides an overview and introduction to the helicopter loads estimation problem; Sect. 2 details the Black Hawk flight loads survey, the input and target variables, and the flight conditions that were examined; Sect. 3 explains the methodology that was followed; Sect. 4 introduces the techniques used for data exploration; Sect. 5 describes the computational intelligence methods used for model building; Sect. 6 provides the experimental settings; Sect. 7 presents

Fig. 1 Australian army Black Hawk

the results from the data exploration and modeling; and Sect. 8 summarizes the findings of this work and offers some conclusions.

2 Test Data

The data used for this work were obtained from a S-70A-9 Black Hawk (UH-60 variant) flight loads survey conducted in 2000 in a joint flight loads measurement program between the United States Air Force and the Australian Defence Force [15]. During these flight trials, 65 hours of useable flight test data were collected for a number of different steady state and transient flight conditions at several different altitudes and aircraft configurations. Instrumentation on the aircraft included 321 strain gauges, with 249 gauges on the airframe and 72 gauges on dynamic components. Accelerometers were installed to measure accelerations at several locations on the aircraft and other sensors captured flight state and control system (FSCS) parameters. Full details of the instrumentation and flight loads survey are provided in [15].

One of the goals of this research was to determine if the dynamic loads on the helicopter could be accurately predicted solely from the FSCS parameters, which are already recorded by the flight data recorder found on most helicopters. The Black Hawk helicopter had thirty such FSCS parameters recorded during the flight loads survey. The thirty FSCS parameters on the Black Hawk that were examined in this work are listed in Table 1. This work focused on estimating the main rotor normal bending (MRBNX) for several flight conditions. From over 50 flight conditions, two were selected for inclusion in this study: forward level flight at full speed and rolling left pullout at $1.5g$. While forward level flight is a steady state manoeuvre that should be relatively straightforward to predict since the parameter values remain steady through the manoeuvre, the rolling pullout manoeuvre is a more severe and dynamic flight condition that should present a greater challenge since there is much more variation in the parameter values through each recording.

Table 1 Black Hawk flight state and control system (FSCS) parameters

Mnemonic	Description	Mnemonic	Description
VCASBOOM	Air speed (boom) Vertical acceleration,	PEDP	Directional pedal position
LOADFACT	Load factor at CG	COLLSTKP	Collective stick position
ATTACK	Angle of attack (boom)	STABLAIC	Stabilator position
SIDESLIP	Sideslip angle (boom)	NR	% of max main rotor speed
PITCHATT	Pitch attitude	ERITS	Retreating tip speed
PITCHRAT	Pitch rate	MRQ	Main rotor shaft torque
PITCHACC	Pitch acceleration	TRQ	Tail rotor drive shaft torque
ROLLATT	Roll attitude	NO1QPCT	No. 1 engine torque
ROLLRAT	Roll rate	NO2QPCT	No. 2 engine torque
ROLLACC	Roll acceleration	NO1T45	No. 1 engine power lever (temp)
HEAD180	Heading	NO2T45	No. 2 engine power lever (temp)
YAWRAT	Yaw rate	HBOOM	Barometric altitude (boom)
YAWACC	Yaw acceleration	FAT	Temperature (Kelvin)
LGSTKP	Longitudinal stick/cyclic position	HD	Altitude (height density)
LATSTKP	Lateral stick/cyclic position	ROCBOOM1	Barometric rate of climb (boom)

A large number of runs for each flight condition were performed during the flight load survey to encompass different altitudes, pilots and aircraft configurations, such as varying gross weight and centre of gravity position. For the manoeuvres examined in this work, there were 26 recordings for forward level flight and 14 recordings for rolling pullout. With a sampling frequency of 52 Hz for the FSCS parameters and each recording lasting about 15 seconds, over 21,000 data points were available for level flight and over 7000 samples for rolling pullout. To obtain models with the broadest application for these flight conditions, the data from all of these runs were used in the training and testing stages of the modeling.

3 Methodology

The overall goal of this work was to develop models to generate accurate predictions for helicopter loads using FSCS parameters. The methodology that was adopted is illustrated in Fig. 2. The application of computational intelligence and machine learning techniques to develop these models occurred in two phases: (i) data exploration: characterization of the internal structure of the data and assessment of the information content of the predictor variables and its relation to the predicted (dependent) variables; and (ii) modeling: build models relating the dependent and the predictor variables.

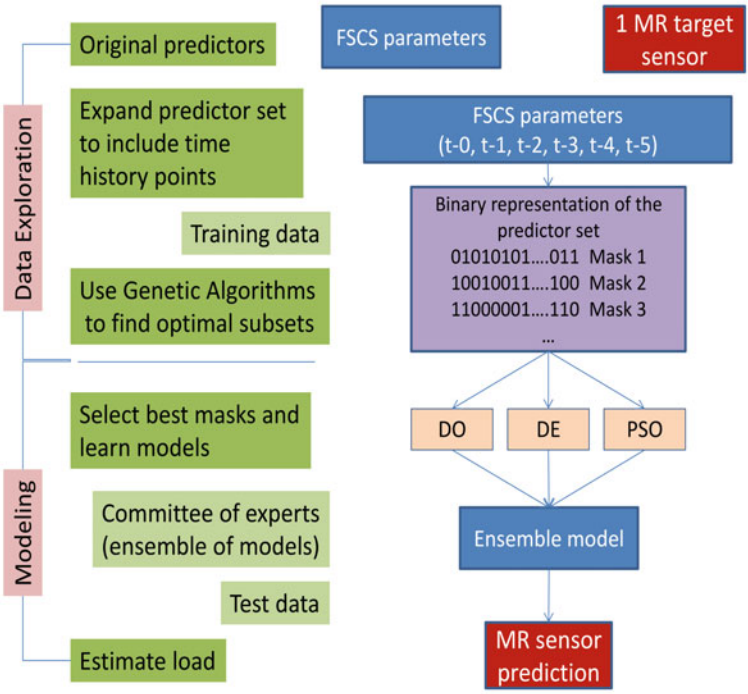


Fig. 2 Experimental methodology. DO, DE and PSO refer to deterministic optimization, differential evolution and particle swarm optimization (and hybrids). They were used for training feed-forward neural network models based on subsets of predictors obtained in the data exploration phase. FSCS: Flight state and control system parameters, MR: Main rotor of the helicopter

For the data exploration stage, phase space methods and residual variance analysis (or Gamma test as described in Sect. 4.2) were used to explore the time dependencies within the FSCS parameters and the target sensor variables, identifying how far into the past the events within the system influenced present and future values. This analysis found that 5 time lags were necessary and therefore the predictor set consisted of the 30 FSCS parameters and their 5 time lags for a total of 180 predictors (described in Sect. 4.3). The Gamma test then steered the evolutionary process used for data exploration. Multi-objective genetic algorithms (MOGA) were used for searching the input space to discover irrelevant and/or noisy information and identify much simpler well-behaved subsets to use as input for modeling. The most promising subsets were then selected and used as a base for model search. A more traditional approach to data exploration would be to use principal component analysis (PCA) to reduce the dimensionality of the data. This method was also explored (described in Sect. 4.1) and generated a subset of predictor variables that could be used for modeling.

During the modeling stage, the models were all feed-forward neural networks that used a number of different computational intelligence techniques (described in

Sect. 5) for building the models relating the target variable to the subset of predictor variables (as identified in the data exploration stage). These techniques included deterministic optimization methods and several evolutionary computation techniques. The network configurations and settings were selected based on either recommended values from literature, previous settings that obtained good results, or simply to cover the allowable parameter range.

3.1 Data Pre-Processing

In multivariate problems it is important to consider the effect of the different units of measurement used for the description of the input variables, which creates semantic incompatibilities. The choice of the units of measurement is specific to each domain, country or user and it is usual that the same magnitude is expressed differently (e.g. a distance expressed in meters or in feet). As a consequence, the actual values of a given physical magnitude may change considerably, thus affecting the computation of indices of similarity, vector distances variables interactions and other multivariate measures.

In order to overcome such unwanted effects, a normalization procedure for all of the variables that will be used for the description of the objects, samples or observations is necessary. There are many normalization procedures. Among them, the conversion of each variable to z-scores transforms the mean of each variable to zero and its standard deviation to 1 (if x_i is a variable with mean $\overline{x}_i$ and standard deviation s_i, $z_i = (x_i - \overline{x}_i)/s_i$). This method is a very convenient one because the values of all variables are measured in units of their own standard deviation, with respect to a common mean of zero, which makes direct comparisons easy. Moreover, since the variance of all variables is the same (one), the influence of each variable in similarities, distances, etc. is the same. It could be argued that this method gives equal influence to variables which may not be related to the target, but the relative influence of the different variables can be assessed with dedicated techniques and later on, weights can be assigned to the individual variables in order to account for the differential influence.

3.2 Construction of the Training and Testing Sets

The training/testing sets used for learning the neural network parameters and for independent evaluation of their performance should come from the same statistical population in order to properly asses the generalization capability of the network models learnt. Common practices in machine learning work with 50–90 % of the available samples for training and the remaining for testing. In the present case, there was a tremendous amount of data available so in order to keep computing times practical, smaller training sets were constructed. In order to compose a training set with manageable size while still containing a statistically representative sample of

the whole dataset, k-means clustering was applied [1]. Accordingly, sets of 2000 clusters were formed using the k-means algorithm with Euclidean distance. Then the data vector closest to each centroid was selected (the so called k-leader) with the set of k-leaders as training sample. Since every data vector is assigned to a cluster and every cluster has a k-leader as its representative, this sampling procedure ensures that every multivariate vector in the original data is represented in the training sample, and at the same time, that it is a reasonably large set for training purposes.

4 Data Exploration Techniques

In this initial phase, the main purpose was to explore the set of 180 potential predictor variables and determine whether proper subsets could be found (much smaller in cardinality) that: (i) could provide an indication of which of the FSCS parameters have greater influence on the helicopter main rotor loads and (ii) would reduce the dimensionality of the supervised learning problem posed by finding suitable prediction models. A brute force approach cannot be considered, as the space of possible predictor subsets is $2^{180} - 1$. Accordingly, other approaches and heuristics were considered. In this case, they were principal components and the multi-objective optimization of residual variance (Gamma test) parameters.

4.1 Principal Component Analysis

Principal component analysis (PCA) (Karhunen–Loève transform) is an unsupervised, non-parametric method of extracting relevant information from data using linear algebra techniques. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components (guaranteed to be independent only if the data set is jointly normally distributed). An important property is that their variance is a monotonically decreasing function of the components. If all of them are retained, 100 % of the original variance is retained. However, if there are correlated variables, a small number of components may capture an important amount of the total variance, allowing a dimensionality reduction via retaining only those components. In such cases, it may reveal the sometimes hidden, simplified structure that often underlie the data. Solving for the principal components is usually done either by diagonalizing a covariance (correlation) matrix or by singular value decomposition techniques [18]. The latter approach was used in this article.

4.2 Gamma Test (Residual Variance) Analysis

The Gamma test is an algorithm developed by [8, 11, 20] as a tool to aid in the construction of data-driven models of smooth systems. It is a technique aimed at estimating the level of noise (its variance) present in a dataset. Noise is understood as any source of variation in the output (target) variable that cannot be explained by a smooth transformation (model) relating the output (predicted or dependent variable) with the input (predictor) variables.

The fundamental information provided by this estimate is whether it is hopeful or hopeless to find (fit) a smooth model to the data. Here a 'smooth' model is understood as one in which the first and second partial derivatives are bounded by finite constants for every point of observation. The gamma estimate indicates whether it is possible to explain the dependent variable by a smooth deterministic model involving the observed input and output variables. Model search is a costly, time consuming data mining operation. Therefore, knowing beforehand that the information provided by the input variables is not enough to build a smooth model is very helpful. It may give an indication that more explanatory variables should be incorporated to the data or that the underlying model may be very complex. If for a given dataset, the gamma estimates are small, it means that a smooth deterministic dependency can be expected. It also gives a threshold in order to avoid overfitting and it can give an indication of how many observations are minimally required in order to build a model which performs with that mean squared error. Overall it gives a measure of the quality of the data.

Let $\mathcal{S}$ be a system described in terms of a set of variables and with $y \in \mathbb{R}$ being a variable of interest, potentially related to a set of m variables $\overleftarrow{x} \in \mathbb{R}^m$ expressed as

$$y = f(\overleftarrow{x}) + r, \tag{1}$$

where f is a smooth unknown function representing the system, $\overleftarrow{x}$ is a set of predictor variables and r is a random variable representing noise or unexplained variation. Despite f being an unknown function, under some assumptions it is possible to estimate the variance of the residual term (r) using available data obtained from $\mathcal{S}$. This will give an indication about the possibility of developing models for y based on the information contained in $\overleftarrow{x}$. Among the most important assumptions are :

- The function f is continuous within the input space.
- The noise is independent of the input vector $\overleftarrow{x}$.
- The function f has bounded first and second partial derivatives.

Let $\overleftarrow{x}_{N[i,k]}$ denote the k-th nearest neighbor of $\overleftarrow{x}_i$ in the input set $\{\overleftarrow{x}_1, \cdots, \overleftarrow{x}_M\}$. If p is the number of nearest neighbors considered, for every $k \in [1, p]$ a sequence of estimates of $\mathbf{E}\left(\frac{1}{2}(y' - y)^2\right)$ based on sample means is computed as

$$\gamma_M(k) = \frac{1}{2M} \sum_{i=1}^{M} \left| y_{N[i,k]} - y_i \right|^2 . \tag{2}$$

In each case, an 'error' indication is given by the mean squared distances between the k nearest neighbors, given by

$$\delta_M(k) = \frac{1}{M} \sum_{i=1}^{M} |\overleftarrow{x}_{N\lfloor i,k \rfloor} - \overleftarrow{x}_i|^2, \tag{3}$$

where **E** denotes the mathematical expectation and |.| Euclidean distance. The relationship between $\gamma_M(k)$ and $\delta_M(k)$ is assumed linear as $\delta_M(k) \to 0$ and then $\Gamma = Var(r)$ is obtained by linear regression

$$\gamma_M(k) = \Gamma + G\ \delta_M(k). \tag{4}$$

A derived parameter of particular importance is the vRatio (V_r), defined as a normalized γ value. Since it is measured in units of the variance of the output variable, it allows comparisons across different datasets:

$$V_r = \frac{\Gamma}{Var(y)}. \tag{5}$$

This vRatio will be the fundamental parameter used in the analysis of the present data. Another important magnitude associated to the Gamma test is the so-called gradient (G), which is proportional to the mean squared value of gradient over the input space and therefore a measure of the complexity of the system under investigation (i.e. the 'roughness' of the unknown surface function f representing the model).

This tool is used in many ways when exploring the data. In the present case, it was used for determining how many time lags were relevant for predicting the future target sensor values, for finding the appropriate number of neighbours for the computation of V_r and G, and very importantly, for determining the subset of lagged FSCS variables with the largest prediction potential (therefore, the best candidates for building predictive models). In this sense, a comprehensive exploration of the datasets for the two flight conditions was made using Gamma test techniques in order to find subsets of the lagged FSCS variables simultaneously featuring minimal V_r (large prediction power), minimal G (low complexity) and small in size (cardinality, denoted by #), thus involving a small number of predictor variables. In order to accomplish this task, a multi-objective optimization framework using evolutionary computation techniques (described in Sect. 5.2.1) was used with $\langle V_r, G, \# \rangle$ as objectives.

4.3 Time Dependencies

One important consideration when working with time-dependent data is the extension of the relationships between the future values with the past values. In purely random processes (e.g. white noise), such dependencies do not exist. In real-world

physical processes, however, events in the past shape in certain ways events of the future. Determining how far into the past the events within a given system exert their influence onto the present and future values and specifically which past events carry information about such dependencies is crucial for an understanding of a system's dynamics as well as for developing predictive models.

In order to explore the structure of the time dependencies within the FSCS parameters and the target sensor variables, phase space methods involving time lags were considered [22]. If T denotes a target sensor, the number of variables (p) in a return map formed by considering an increasing sequence of lags $(t-1), (t-2), \ldots, (t-p)$ which is sufficient to describe the properties of the system must be found. The value of p was determined using the residual variance technique (Gamma test). For the case of the MRNBX sensor, 20 lags were found as meaningful. For the FSCS variables, a more complex setting was constructed in order to capture the nature of the lagged interactions between the whole set of predictors. If $P_k(t)$ denotes the kth FSCS parameter time series (k in [1, 30]), tuples describing the state of the systems in terms of the predictors and the target can be formed as $[P_1(t-\tau), \ldots, P_1(t-1), P_1(t)]$, $[P_2(t-\tau), \ldots, P_2(t-1), P_2(t)], \ldots, [P_{30}(t-\tau), \ldots, P_{30}(t-1), P_{30}(t)], T(t)$, where τ is a maximum embedding lag for the FSCS variables. The Black Hawk flight loads survey provided the data for the FSCS parameters recorded at 52 Hz and the data for the MR target parameters at 416 Hz. Since the FSCS variables were sampled at a 1:8 frequency ratio with respect to the target MR sensors, a value of $\tau = 5$ was chosen to sufficiently cover the time spanned by the target sensor. Therefore the predictor set consisted of the 30 FSCS parameters at both the current time step $(t-0)$ as well as their preceding 5 time lags $((t-1), (t-2), \ldots, (t-5))$ for a total of 180 predictors (30 parameters $\times$ 6 time steps).

5 Machine Learning Methods

5.1 Neural Networks

Neural networks (NN) are universal function approximators that can be applied to a wide range of problems such as classification and model building. It is already a mature field within machine learning and computational intelligence and there are many different NN paradigms. Multilayer feed-forward networks are the most popular and a large number of training algorithms have been proposed. Training neural networks involves an optimization process focused on minimizing an error measure. It is customary to use the mean squared error (MSE) or its root (RMSE) given by $\mathrm{MSE} = \sum_{i=1}^{n} (o_i - p_i)^2/n$, $\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$, where o_i and p_i are the observed and predicted values of the target variable respectively (n is the number of observations). Other error measures have been proposed which have proved to be useful for specific domains. Optimizing the error or performance measure can be done using a variety of approaches ranging from deterministic methods to stochastic, evolutionary computation (EC) and hybrid techniques. In this article deterministic,

evolutionary computation methods, as well as hybrids between them were used. Neural networks are considered black box models and there are many other black as well as white modeling approaches within machine learning. In this article, modeling with feed-forward neural networks is considered, but other approaches for the same problem were introduced by the authors previously [23, 24].

Deterministic optimization (DO) of the RMSE or other error measures is the standard practice when training neural networks. One of the popular and powerful DO techniques is the Levenberg–Marquardt algorithm (LM) [18]. It works with an approximation of the Hessian matrix (of second derivatives of the error function, in this case, mean-squared). The advantage of this method is that it smoothly blends the Newton and the steepest descent approaches into a single way of computing the optimization parameters (the NN weights), ranging a continuum between the two. As the process goes on, an adaptation parameter shifts the process towards favoring a Newton-like or steepest descent-like approach to updating the optimization parameters.

5.2 Evolutionary Computation Methods

An evolutionary computation (EC) algorithm constructs a population of individuals, which evolve through time until a stopping criteria is satisfied. At any particular time, the current population of individuals represent the current solutions to the input problem, with the final population representing the algorithm's resulting output solutions. Genetic algorithms (GA) are the most popular of the EC techniques [2, 10]. In the present case, EC and hybrid techniques were used for searching in subsets of feed-forward neural networks spaces composed of multidimensional tuples of weights defined by a neural network with a fixed layout. Once the architecture of a network is defined in terms of the number of layers, their composition and the kind of aggregation and activation functions used for each layer, the dimensionality of the space is fixed. EC techniques are used for mining in such subspaces for networks optimizing a specified error or fitness function. When pure EC techniques are used, no partial derivatives are involved as with most deterministic training methods. The hybrid EC–DO approaches have the advantages of combining the more global search capabilities of the EC methods with the powerful local search of the DO procedures.

5.2.1 Multi-Objective Optimization using Genetic Algorithms (MOGA)

An enhancement to the traditional evolutionary algorithm is to allow an individual to have more than one measure of fitness within a population. This modification may be applied through the use of a weighted sum of more than one fitness value [3]. MOGA, however, offers another possible way for enabling such an enhancement. In the latter case, the problem arises for the evolutionary algorithm to select individuals for inclusion in the next population, because a set of individuals contained in one

population exhibits a Pareto Front [16] of best current individuals, rather than a single best individual. Most [3] multi-objective algorithms use the concept of dominance.

A solution $\overleftarrow{x}_{(1)}$ is said to dominate [3] a solution $\overleftarrow{x}_{(2)}$ for a set of m objective functions $\langle f_1(\overleftarrow{x}), f_2(\overleftarrow{x}), \ldots, f_m(\overleftarrow{x})\rangle$ if

1. $\overleftarrow{x}_{(1)}$ is not worse than $\overleftarrow{x}_{(2)}$ over all objectives.
   For example, $f_3(\overleftarrow{x}_{(1)}) \leq f_3(\overleftarrow{x}_{(2)})$ if $f_3(\overleftarrow{x})$ is a minimization objective.
2. $\overleftarrow{x}_{(1)}$ is strictly better than $\overleftarrow{x}_{(2)}$ in at least one objective. For example, $f_6(\overleftarrow{x}_{(1)}) > f_6(\overleftarrow{x}_{(2)})$ if $f_6(\overleftarrow{x})$ is a maximization objective.

One particular algorithm for MOGA is the elitist non-dominated sorting genetic algorithm (NSGA-II) [3, 5–7]. It has the features that it (i) uses elitism, (ii) uses an explicit diversity preserving mechanism, and (iii) emphasizes the non-dominated solutions. The procedure is as follows:

1. Create the child population using the usual genetic algorithm operations.
2. Combine parent and child populations into a merged population.
3. Sort the merged population according to the non-domination principle.
4. Identify a set of fronts in the merged population ($\mathcal{F}_i, i = 1, 2, \ldots$).
5. Add all complete fronts $\mathcal{F}_i$, for $i = 1, 2, \ldots, k-1$ to the next population.
6. If there is a front, $\mathcal{F}_k$, that does not completely fit into the next population, select individuals that are maximally separated from each other from the front $\mathcal{F}_k$ according to a crowding distance operator.
7. The next population has now been constructed so continue with the genetic algorithm operations.

Considering that an ideal model would be one approximating the target variable values as much as possible while being simple and depending on as few predictor variables as possible, the multi-objective optimization approach investigated was formulated in terms of a 3-objective MOGA with Residual Variance criteria, aimed at finding subsets of predictor variables that simultaneously: (i) minimize V_r as an approximation to the MSE of the model's residual, (ii) minimize G as a measure of model complexity, and (iii) minimize the ratio of the number of predictor variables in the model with respect to the total number of potential predictors (180) as a normalized cardinality measure of the set of predictors.

5.2.2 Differential Evolution

Differential evolution (DE) [19, 21] is a kind of evolutionary algorithm working with real-valued vectors. Although relatively less popular than genetic algorithms, it has proven to be very effective for complex optimization problems, outperforming other approaches [9, 14]. As in other EC algorithms, it works with populations of individual vectors (real-valued) and evolves them. There are many variants but the general scheme is as follows:

step 0 Initialization: Create a population $\mathcal{P}$ of random vectors in $\Re^n$, and decide upon an objective function $f : \Re^n \rightarrow \Re$ and a strategy $\mathcal{S}$, involving vector differentials.
step 1 Choose a target vector from the population $\vec{x}_t \in \mathcal{P}$.
step 2 Randomly choose a set of other population vectors $\mathcal{V} = \{\vec{x}_1, \vec{x}_2, \dots\}$ with a cardinality determined by strategy $\mathcal{S}$.
step 3 Apply strategy $\mathcal{S}$ to the set of vectors $\mathcal{V} \cup \{\vec{x}_t\}$ yielding a new vector $\vec{x}_{t'}$.
step 4 Add $\vec{x}_t$ or $\vec{x}_{t'}$ to the new population according to the value of the objective function f and the type of problem (minimization or maximization).
step 5 Repeat steps 1–4 to form a new population until termination conditions are satisfied.

There are several variants of DE which can be classified using the notation $DE/x/y/z$, where x specifies the vector to be mutated, y is the number of vectors used to compute the new one and z denotes the crossover scheme. The algorithm is controlled by two parameters: the scaling factor F and the crossover rate $C_r \in \Re$. More than ten particular strategies have been proposed that differ in the way the trial vector is constructed. In this paper, the *DE/rand/*1*/exp* strategy was used as it has worked well for most problems (see Sect. 6 for experimental settings).

5.2.3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic search process, modeled after the social behavior of bird flocks and similar animal collectives [4, 12, 13]. The algorithm maintains a population of particles, where each particle represents a potential solution to an optimization problem. In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each potential solution is referred to as a particle. Each particle i maintains information concerning its current position and velocity, as well as its best location overall. These elements are modified as the process evolves, and different strategies have been proposed for updating them, which consider a variety of elements like the intrinsic information (history) of the particle, *cognitive* and *social* factors, the effect of the *neighborhood*, etc., formalized in different ways. The swarm model used has the form proposed in [25]:

$$\begin{aligned} v_{id}^{k+1} &= \omega \cdot v_{id}^{k} + \phi_1 \cdot (p_{id}^{k} - x_{id}^{k}) + \phi_2 \cdot (p_{gd}^{k} - x_{id}^{k}) \\ x_{id}^{k+1} &= x_{id}^{k} + v_{id}^{k+1} \\ \phi_i &= b_i \cdot r_i + d_i, \quad i = 1, 2 \end{aligned} \tag{6}$$

where v_{id}^{k+1} is the velocity component along dimension d for particle i at iteration $k+1$, and x_{id}^{k+1} is its location; b_1 and b_2 are positive constants; r_1 and r_2 are random numbers; d_1 and d_2 are positive constants to cooperate with b_1 and b_2 in order to confine ϕ_1 and ϕ_2 within the interval $(0.5, 2)$; ω is an inertia weight (see Sect. 6 for settings).

5.2.4 Hybrid Techniques

A common issue of deterministic (gradient-based) techniques is the entrapment in local extrema, which can be mitigated by combining local and global search techniques. In this case, deterministic optimization with evolutionary computation methods, both presented above. Several hybridization approaches are possible: (i) *coarse and refinement stages*: Use a global search technique and upon completion, go to a next step of using a subset of the solutions (proper or not) as initial approximations for local search procedures (e.g. deterministic optimization methods). The final solutions will be those found after this second step; (ii) *memetic*: Embed the local search within the global search procedure. In this case the evaluation of the individual constructed by the global search procedure is made by a local search algorithm. For example, within an evolutionary computation procedure, the evaluation of the fitness of an individual is the result of a deterministic optimization procedure using as initial approximation the individual provided by the evolutionary algorithm. Then, the EC-individual is *redefined* accordingly and returned to the evolutionary procedure for the application of the evolutionary operators and the continuation of the evolutionary procedure. In this paper both hybridization procedures were used.

6 Experiments and Their Settings

6.1 Data Exploration—MOGA Settings

During the data exploration by the MOGAs as guided by the Gamma test, 150 runs were made for each of the flight conditions and training schemes. The experimental settings included allowing elitism, using a number of crossover probabilities (0.3, 0.5, 0.6, 0.8, 0.9), mutation probabilities (0.01, 0.025, 0.05), and 10 random seeds. Each experiment consisted of 1000 objects that were allowed to evolve over 300 generations producing 1000 solutions. Therefore for each case, a total of 15,000 subsets (or masks) were identified.

6.2 Neural Network Configurations

In this study, the models were all feed-forward neural networks that were trained using one of seven different methods: pure LM, pure PSO, PSO with LM, memetic LM-PSO, pure DE, DE with LM, and memetic LM-DE. A number of network configurations were attempted including those with one or two hidden layers with 1–12 neurons in each hidden layer. The output layer used a linear transfer function, while the hidden layers used a $tanh$ transfer function. Each trial was repeated up to three times initialized with a different random seed. In total there were 1872 runs for each

Table 2 Experimental settings for PSO and DE

PSO settings		DE settings	
Parameter	Value	Parameter	Value
No. of particles	10	Population size	20
b_1,b_2	1.5, 1.5	F	0.5
r_1, r_2	*rnd* [0, 1], *rnd* [0, 1]	C_r	0.8
d_1, d_2	0.5, 0.5	r_i	*rnd* ([0,1])
range of x_i^0	$[-3, 3]$	Range of x_i^0	$[-3, 3]$
range of v_i^0	$[1, 5]$	Strategy	DE/rand/1/exp

flight condition with an imposed time limit (3 hours) and maximum number of iterations allowed (10,000). The settings for PSO and DE are described in Table 2. The network configurations and settings were selected based on either recommended values from literature, previous settings that obtained good results, or simply to cover the allowable parameter range.

7 Results and Discussion

7.1 Data Exploration Results

The exploration using PCA obtained the principal components of the data contained by the 180 input parameters. (This process is unsupervised and therefore did not use any information from the target variables.) For level flight, 95% variance of the data was captured by 20 principal components, whereas for rolling pullout 13 components were required. A heuristic was used to select the 'most important' variables from the point of view of their contribution to the linear combination defining each component (i.e. the composition of the eigenvectors). For each eigenvector, variables with associated weights over 95% of the eigenvector's absolute weight range were considered important. Accordingly, for the 20 and 13 principal components for level flight and rolling pullout respectively, there were 125 and 120 predictor variables (from the original 180) that formed the most important subsets required to explain the chosen PCs at the 95% variance threshold. From the 30 FSCS parameters, 24 were included in the PCA subset for level flight and 20 for rolling pullout. The dimensionality reduction ratio is high when the number of principal components are compared with the number of variables, but not quite so when the number of variables required to reconstruct the component are considered, as no component (and particularly the first ones) could be explained with a small number of the original variables.

Searching for subsets of input variables with predictor potential, the exploration using MOGA in combination with the Gamma test (MOGA-Γ) generated 15,000 solutions for each flight condition. As mentioned, the task of searching the input space for suitable subsets was not a simple one since with 180 predictor variables,

there were $2^{180} - 1$ subsets to consider as input variables for modeling. This huge number, however, only includes the potential combinations of variables; the actual number of potential models that could be formed from the different combinations of variables is infinite. Therefore the 15,000 subsets that were generated by the MOGA-Γ represent only a tiny fraction of that space. The goal of the MOGA-Γ was to simultaneously minimize the three objectives of normalized residual variance, complexity and number of predictors $\langle V_r, G, \# \rangle$ (Eqs. 4 and 5 in Sect. 4.2). The search process aimed at optimizing those three objectives would favour subsets of predictor variables leading to accurate (low V_r) and parsimonious models (low G and small cardinality). As is typical in real-world multi-objective optimization problems, no single solution can absolutely optimize all of the objectives. The algorithm produced a set of candidate solutions representing the best tradeoffs between the different objectives (non-dominated solutions).

Each solution generated by the genetic algorithm was a binary string, also called a mask, where the position of the 1-bits indicated whether the corresponding variable from the set of 180 predictors was chosen for assembling a model for the target sensor. Irrelevant parameters were omitted from the MOGA-Γ solutions. The vRatio of that subset indicated the lowest possible mean squared error that could be obtained by building a non-overfitting model using the variables in the subset as predictors. The MOGA-Γ masks provided two important pieces of information for the modeling stage: (i) a reduced input parameter set with irrelevant parameters removed, and (ii) a target error threshold (vRatio) to aim for during training.

Since the overall goal of this process was to create accurate models for sensor prediction, of the three objectives that were optimized (vRatio, complexity, number of parameters) only the vRatio provided an indication of the accuracy of the resultant model using that mask. Therefore in analyzing the MOGA-Γ results, even though the MOGA-Γ sought to simultaneously minimize all three objectives, the ranking of the masks was based on the vRatio, that is, the most accurate within the parsimonious subsets (masks).

The most promising masks were defined as those in the 0.01-percentile of all masks according to vRatio, equating to approximately 150 masks for each case. The results are summarized in Table 3 describing the average number of parameters found in the top masks, the range of vRatio values for those masks, and the most frequently occurring parameters in those masks from the 30 FSCS parameters. The distribution of the frequency of occurrence of the 180 input parameters within the top masks is shown in Fig. 3. These frequencies are not necessarily an indication of their relative weight in the subsequent model, they simply indicate how many times that parameter was included in the top 1% of the MOGA-Γ solutions. However, they indicate relevance in the sense of being present in many masks associated to low noise values, therefore, good to include in a modeling exercise. The effect of a variable in a model also depends on its interaction with the other variables, whereas these frequencies are associated to each variable individually.

As can be seen in the summary table and the frequency distribution, there was a small number of parameters that appear in almost all of the solutions for the two flight conditions, namely the main rotor shaft torque, tail rotor drive shaft torque,

Table 3 Summary of MOGA-Γ results for the MRNBX sensor

	Level flight	Rolling pullout
No. of mask parameters	43	16
vRatio range	0.037−0.105	0.55−0.628
Mask parameters above 50%	MRQ	MRQ
Occurrence frequency	TRQ	TRQ
	YAWACC	
	NR	
	ROLLACC	

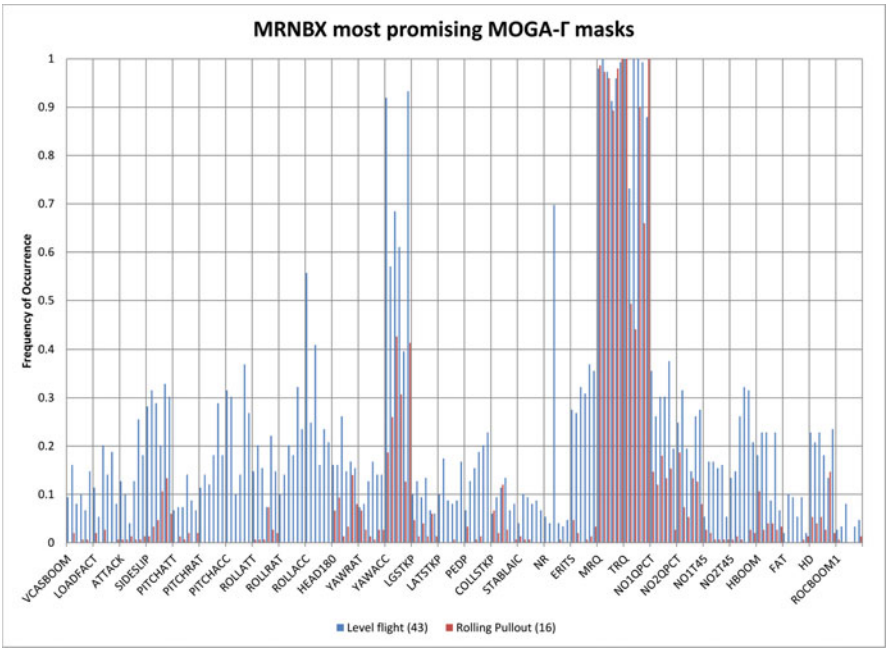


Fig. 3 Frequency distribution of FSCS parameters for the most promising MOGA-Γ masks for the MRNBX sensor under level flight and roll pullout flight conditions

and the yaw acceleration (MRQ, TRQ, and YAWAACC respectively). Some of these parameters appeared multiple times with great frequency to include various time history points. Of the 180 original parameters, the average mask sizes found by the MOGA-Γ were 43 for level flight and 16 for rolling pullout, representing a reduction to 24% and 9% respectively from the original input variable set. The vRatio for the level flight solutions were quite low at 0.037. One would therefore expect quite accurate models to be built for the level flight manoeuvre using any of these top 1% MOGA-Γ solutions. For the rolling pullout flight condition, the MOGA-Γ solutions were dominated by the same three FSCS parameters as seen previously (MRQ, TRQ

and YAWACC). Furthermore, none of the other FSCS parameters appeared in the solutions with frequency above 33%. While the average size of the masks found for the rolling pullout was much smaller (16 parameters), the vRatio of these solutions was much higher (> 0.55). These high vRatio values indicate that the MRNBX sensor readings for that flight condition would be predicted with a much lower level of accuracy with respect to its level flight counterpart. These results are consistent with the fact that rolling pullout is a more complicated flight condition.

Inclusion of an analysis of the least promising masks is necessary to provide some perspective and comparison to the top masks. In the top masks some clear trends emerged with a small number of FSCS parameters appearing frequently in the solutions. If these same parameters were either absent or only marginally present in the worst performing masks, their relevance for prediction would be more clearly established. The least promising masks of the 15,000 generated by the genetic algorithm for the two flight conditions were the ones with the greatest residual variance, greatest complexity and largest number of mask parameters. However, it is worth noting that each mask within the final feasible population represented the end result of an evolutionary process where millions of masks were evaluated and discarded. The least promising masks presented here were defined as those above the 99th percentile of all the masks in each case, corresponding to about 150 masks. The frequency distribution of the FSCS parameters in the least promising masks is shown in Fig. 4. Most notable in these results is the relatively even distribution of parameters at a similar frequency and the lack of high frequency (> 60%) parameters other than NR. While the three FSCS parameters (MRQ, TRQ, and YAWACC) that dominated the top masks still had some representation in the least promising masks for rolling pullout, the frequency of occurrence for these parameters is mostly under 30% like almost all other variables. In fact for level flight, MRQ and YAWACC were virtually nonexistent in the least promising masks. Clearly, these results confirm the findings derived from the analysis of the most promising masks.

The FSCS parameters selected by PCA and MOGA-Γ are shown in Table 4 (definitions of the mnemonics are found in Table 1 in Sect. 2). The MOGA-Γ feature selection procedure resulted in greater dimensionality reduction in terms of both the overall number of predictor variables and the number of FSCS parameters involved. These subsets can be compared from the point of view of a similarity measure defined as $\mathcal{S}(A, B) = \frac{\#\bigcap(A,B)}{\#\bigcup(A,B)}$ where (A, B) are the sets to compare, $\bigcap, \bigcup$ their intersection and union, and # their cardinality. From a similarity perspective, the PCA masks for level flight and rolling pullout were 0.64 similar. That is, PCA saw that a large number of variables are commonly relevant for both flight conditions. For MOGA-Γ, the similarity was only 0.19, which clearly indicates that MOGA-Γ differentiated the two flight conditions more sharply than PCA, as it identified essentially different subsets of variables as relevant for the two flight conditions. When the flight conditions are considered independently, for level flight the similarity between the PCA and the MOGA-Γ masks was 0.64. For rolling pullout, however, the similarity was much smaller (0.16). These results indicate that overall PCA characterized the processes in terms of a larger amount of FSCS parameters than MOGA-Γ (less dimensionality reduction). The two approaches essentially were identifying different subsets as

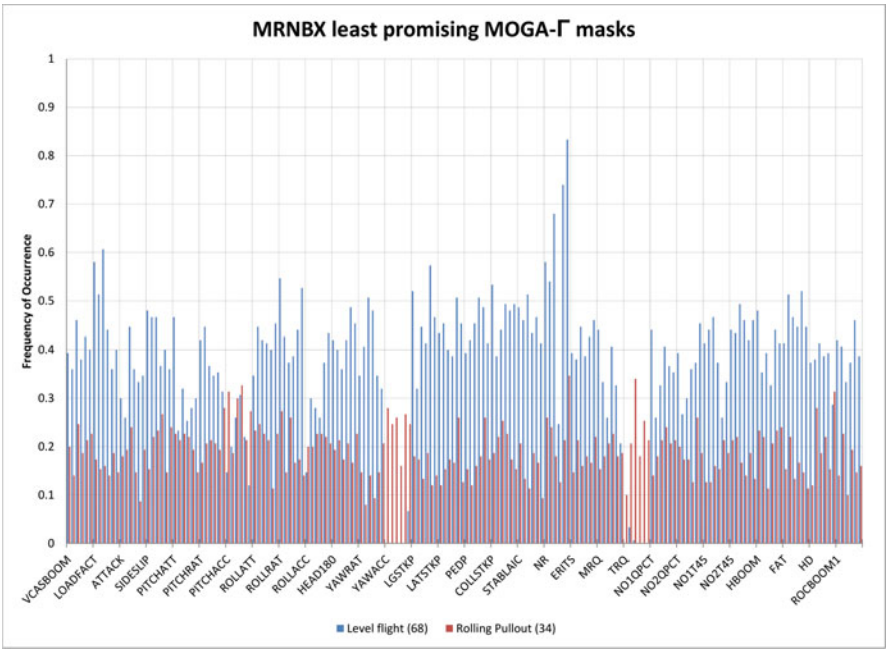


Fig. 4 Frequency distribution of FSCS parameters for the least promising MOGA-Γ masks (MRNBX sensor under level flight and roll pullout flight conditions)

relevant variables as indicated by the limited similarity between the chosen subsets for the two flight conditions. Their differential performance when used for modeling is presented in Sect. 7.2, which also compares the behavior of the full mask (involving all predictors).

For both flight conditions examined in this work, the MOGA-Γ was able to find a large number of solutions with a greatly reduced set of predictor variables, indicating that the original input set contained a large amount of irrelevant and noisy data. There was a small number of parameters that consistently appeared with very high frequency in the top masks for both flight conditions: the main rotor drive torque, the tail rotor drive shaft torque, and the yaw acceleration (MRQ, TRQ, YAWACC respectively). While these parameters also appeared with moderate frequency in the least promising masks, their overwhelming presence in the top masks suggests that these parameters are essential for predicting the target sensors in these flight conditions.

Some discussion regarding the flight conditions examined is relevant here. In forward level flight at full speed the helicopter would require more thrust and therefore would experience more drag. The increased thrust requirement would be manifested in the aircraft pitching forward so that more of the thrust vector is aligned in the direction of travel. High vibratory loads would also be generated as the load cycles increase which means that the reactions at the main rotor hub would increase as well. In a rolling left pullout at $1.5g$, the helicopter would be coming out of a dive and climbing while rolling to the left. The helicopter would be banked and

Table 4 Summary of PCA and top MOGA-Γ mask results for MRNBX. Refer to Table 1 in Sect. 2 for description of FSCS parameter abbreviations

Level flight		Rolling pullout	
PCA	MOGA-Γ	PCA	MOGA-Γ
125 predictors	33 predictors	120 predictors	17 predictors
24 FSCS parameters	20 FSCS parameters	20 FSCS parameters	4 FSCS parameters
VCASBOOM	ATTACK	VCASBOOM	COLLSTKP
LOADFACT	PITCHATT	LOADFACT	MRQ
SIDESLIP	PITCHRAT	SIDESLIP	TRQ
PITCHRAT	PITCHACC	PITCHRAT	NO1QPCT
ROLLATT	ROLLATT	PITCHACC	HD
ROLLRAT	ROLLRAT	ROLLATT	
ROLLACC	ROLLACC	ROLLRAT	
HEAD180	HEAD180	ROLLACC	
YAWRAT	YAWACC	YAWACC	
YAWACC	LGSTKP	LGSTKP	
LGSTKP	LATSTKP	NR	
PEDP	PEDP	ERITS	
COLLSTKP	COLLSTKP	MRQ	
STABLAIC	STABLAIC	TRQ	
NR	NR	NO1QPCT	
ERITS	MRQ	NO2T45	
MRQ	TRQ	HBOOM	
TRQ	NO1QPCT		
NO1QPCT	NO2QPCT		
NO2QPCT	FAT		
NO2T45			
HBOOM			
HD			
ROCBOOM1			

therefore the aerodynamic and centrifugal forces acting on the helicopter would be asymmetric. Some of the characteristics at full speed as mentioned for forward level flight would be expected, such as the high thrust requirement and the high vibratory loads. In addition, the increased gravitational forces on the helicopter would generate increased upward force on the main rotor blades, and the rolling orientation would result in an asymmetric loading. The FSCS parameters that one might expect to reflect these conditions could include roll acceleration (ROLLACC) for the rolling pullout manoeuvre, and for both manoeuvres retreating tip speed (ERITS), engine

Table 5 Comparison of full mask, PCA subset, and MOGA-Γ mask results for MRNBX level flight and rolling pullout

MRNBX Level flight		Training RMSE	Training corr	Testing RMSE	Testing corr
Full mask 180 predictors	PSO	0.429	0.886	0.621	0.786
	DE	0.291	0.962	0.615	0.796
PCA subset 125 predictors	PSO	0.593	0.767	0.791	0.559
	DE	0.591	0.765	0.875	0.521
MOGA-Γ mask 33 predictors	PSO	0.385	0.909	0.668	0.747
	DE	0.291	0.952	0.612	0.795
MRNBX Rolling pullout		Training RMSE	Training corr	Testing RMSE	Testing corr
Full mask 180 predictors	PSO	0.678	0.739	0.884	0.492
	DE	0.327	0.948	0.832	0.586
PCA subset 120 predictors	PSO	0.887	0.430	0.959	0.308
	DE	0.816	0.579	0.936	0.364
MOGA-Γ mask 17 predictors	PSO	0.744	0.426	0.911	0.210
	DE	0.744	0.653	0.898	0.476

torque (NO1QPCT or NO2QPCT), main rotor shaft torque (MRQ), tail rotor drive shaft torque (TRQ), air speed (VCASBOOM), and load factor (LOADFACT). The candidate solutions found by the genetic algorithms were somewhat consistent with the above postulates, but the prominence of the three parameters (MRQ, TRQ, and YAWACC) was quite surprising.

7.2 Modeling Results

Following the data exploration, models were built estimating the main rotor normal bending for the two flight conditions. The inputs to these models were either the full mask of 180 parameters, the subset identified by PCA, or the most promising mask found by the MOGA and Gamma test for each flight condition. The models were all feed-forward neural networks that used various computational intelligence techniques for training, which in this case consist of estimating the neural network's weights using MSE as the objective to minimize. Ensembles of the top models were then formed for each case using either pure LM, pure EC, coarse-refinement DO-EC, or memetic DO-EC (see Sect. 5). Table 5 shows the performance results (root mean squared error (RMSE) and correlation (corr)) of the models for the two flight conditions. Figure 5 plots the estimates for level flight for the testing set using the three different input sets. Figure 6 plots the estimates for rolling pullout for the testing set.

Using the full mask predictions as the baseline result, it is clear that the model predictions using PCA were considerably less accurate and less correlated with the observed sensor loads than those for either the full mask or the ensemble models based on MOGA-Γ masks. Factors that may explain this result are the unsupervised

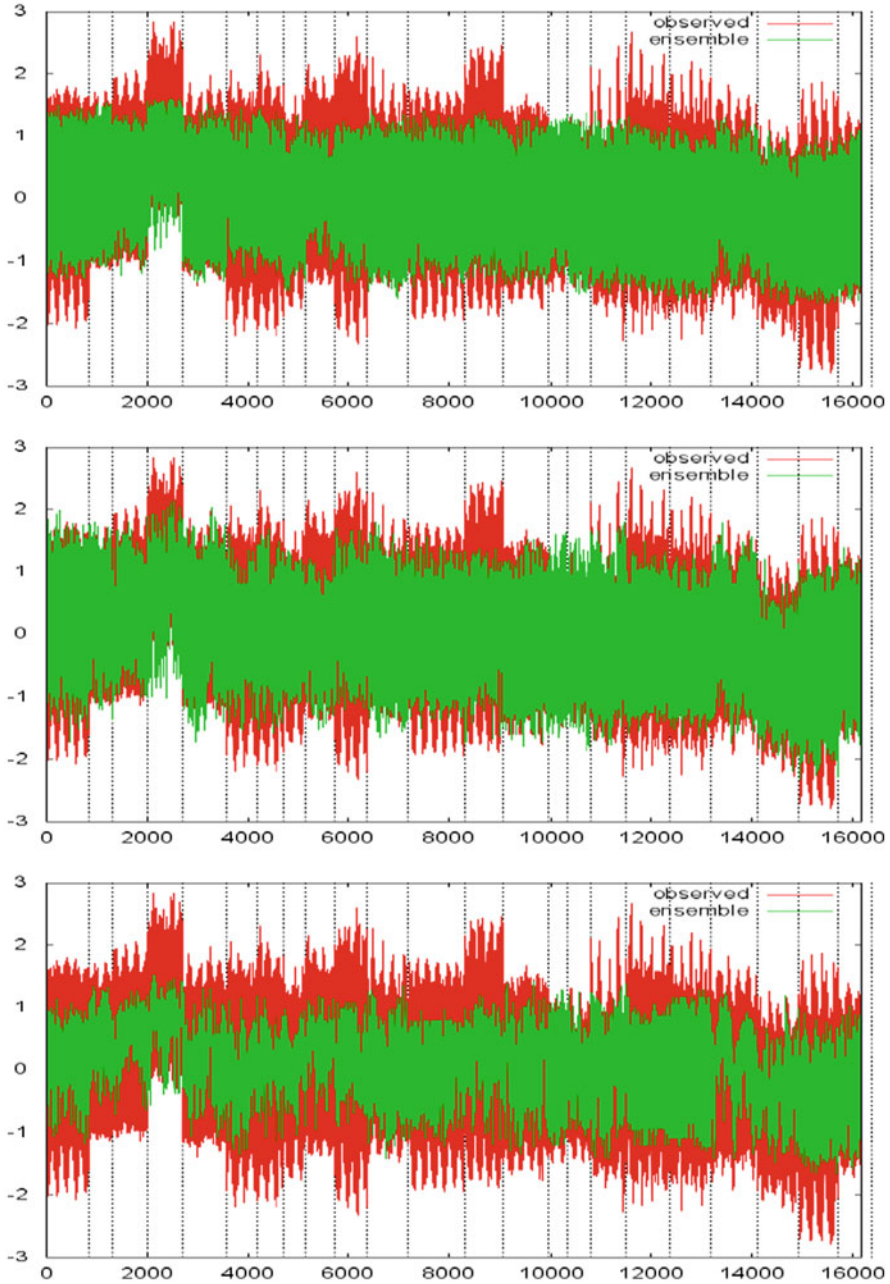


Fig. 5 Predictions for MRNBX level flight using neural network ensembles trained with PSO. *Top*: full mask of 180 predictors. *Middle*: MOGA-Γ mask of 33 predictors. *Bottom*: PCA subset of 125 predictors. The dashed vertical lines indicate the boundary between different flight recordings

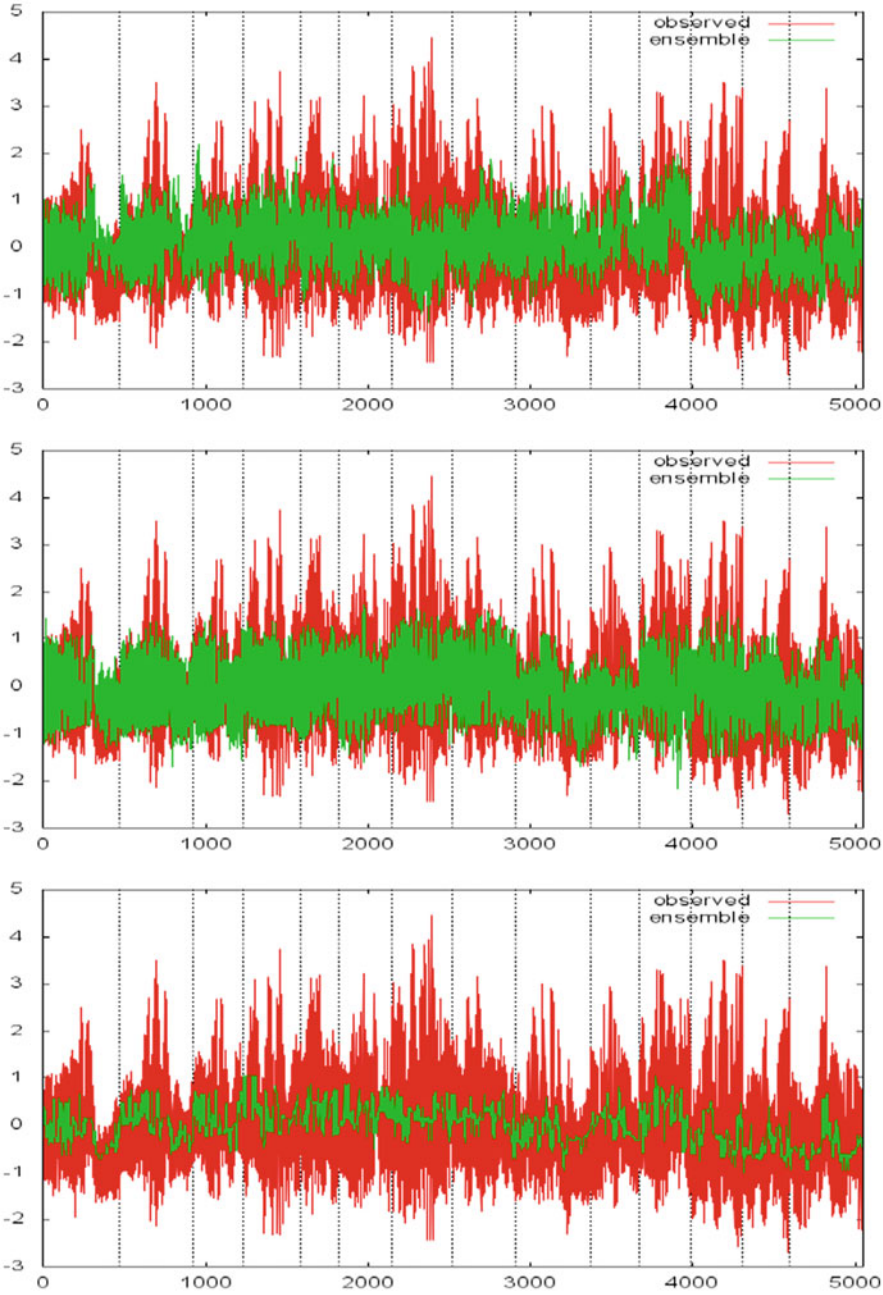


Fig. 6 Predictions for MRNBX rolling pullout using neural network ensembles trained with PSO. *Top*: full mask of 180 predictors. *Middle*: MOGA-Γ mask of 33 predictors. *Bottom*: PCA subset of 125 predictors. The dashed vertical lines indicate the boundary between different flight recordings

nature of the PCs, based only on the structure of the predictor variables, and very importantly, the fact that they only capture linear dependencies between the variables. This characteristic is a great shortcoming when the nature of the relationships is nonlinear, as seems to be the case of the helicopter dynamics (particularly under complex flight conditions). This behavior is suggested by the poor performance of the PCA-based predictions for the rolling pullout flight condition which is a more complicated manoeuver than forward level flight.

The predictions for level flight using the full mask and the MOGA-Γ mask were very similar in performance and both were quite accurate and correlated. The rolling pullout was a more difficult flight condition to predict due to its dynamic nature and this complexity is reflected in the performance results which have higher RMSE and lower correlation as compared to level flight. The plots of the predictions in Figs. 5 and 6 show that improvements could still be attained particularly at the peak values of large magnitude of the target signal. Certainly the upper peaks are underestimated, the lower peaks less so, and overestimation of the values is rare. The main and secondary peaks as well as the phase of the predicted signal match up well with the target signal, which are important features for helicopter load monitoring.

The MOGA-Γ masks contained only 33 variables for level flight and 17 variables for rolling pullout compared to the 180 of the full mask, corresponding to a significant 82% and 91% respectively reduction in size. However, the models generated using these reduced mask yielded predictions with very similar if not better performance than those obtained with the full mask, which is remarkable. Even though the full mask contained the same variables as the most promising mask, it is evident that having to accommodate *all* of the variables, including those the GA deemed superfluous and noisy, detracted from its performance. The use of MOGA and the Gamma test to identify and exclude noisy and irrelevant parameters then was successful. This result is encouraging, as it indicates that simpler models could be constructed using the MOGA-Γ masks without sacrificing performance. Models with good performance were found for both flight conditions demonstrating the effectiveness of the approach. It should be noted that the MOGA-Γ feature selection procedure is computationally much more expensive than PCA, but the results indicate that in the context of the problem investigated here, its use is plainly justified.

From the data exploration phase, the most promising mask found by the MOGA-Γ had an associated vRatio indicating the lowest possible residual variance (or error) based on the training set that could be obtained by building a model from that subset; however, finding the model that will achieve that level of performance is a separate challenge in itself. The vRatio of the MOGA-Γ masks used for modeling for level flight and rolling pullout were 0.037 and 0.55 respectively. The best training MSE values achieved were 0.107 (RMSE = 0.328) for level flight and 0.554 (RMSE = 0.744) for rolling pullout (Table 5). For level flight the training MSE attained was far from the original Gamma test estimates, known to be (optimistic) theoretical MSE levels [8, 11, 20], not always achievable in practice.

One of the challenges in the analysis involved some of the discovered peculiarities of the dataset. As seen in Figs. 5 and 6, the dashed vertical lines indicate the boundary between different flight records for the same manoeuvre so that the variation in the

shape and range of the time signal from one recording to the next is evident. Particularly for the rolling pullout manoeuvre, the main rotor normal bending appeared to vary noticeably depending on the pilot, aircraft configuration, and environmental conditions. Furthermore, within the same flight recording (particularly for the rolling pullout manoeuvre), each individual record did not appear to be homogeneous, that is, each record did not solely consist of data from the labelled flight condition. This observation is likely due to the nature of recording such dynamic manoeuvres, as the recording likely would have included the helicopter's steady state condition just before and after the manoeuvre, the transitions into and out of the manoeuvre, and the manoeuvre itself. While it might be possible to further refine the classification of the flight records to include only those tuples 'rightfully' belonging to the flight condition under examination, any type of classification introduces a subjective element in order to establish crisp boundaries between classes and therefore another source of error would result. This type of 'contamination' is encountered in many classification problems, in fact one could say it is an intrinsic characteristic of a real world data gathering process, and the emphasis in this paper is the methodology and approach developed to utilize computational intelligence techniques to estimate helicopter loads. Needless to say, the presence of these heterogeneities and variations in the dataset, while difficult to avoid or overcome, made the task of data exploration and modelling particularly challenging.

Overall, the models provided a reasonable approximation of the target parameter, more so for the level flight manoeuvre than for the rolling pullout manoeuvre. It is important to keep in mind that the test data were taken from many different flight records (26 for level flight, 14 for rolling pullout) with different aircraft configurations whose variation was not necessarily captured in the 30 FSCS parameters. Furthermore the training set was smaller than the test data set in a ratio of 2:19 for level flight and 2:5 for rolling pullout. The results presented here are promising especially given the relatively small size of the training data set and that the predictor set to obtain these results used less than 20% of the total predictor variables, but improvements to the signal prediction, particularly at the peak values, is still required. Future work will continue to explore methods to improve the sensor prediction while still following the same overall methodology presented here. Other improvements could perhaps be obtained by increasing the size of the training set without demanding overly high computing times, exploring alternative sampling schemes in forming the training set, and trying different error functions in the neural network optimization process.

8 Conclusions

This paper presents the results of data mining on a class of aerospace data including exploration and model building processes to estimate the Black Hawk helicopter main rotor normal bending from 30 recorded flight state and control system parameters during several flights conducted under two different flight conditions (forward level flight at full speed and rolling left pullout at $1.5g$). The approach involved a

combination of statistical and machine learning (computational intelligence) techniques for data exploration and model building. Residual variance analysis (Gamma test) provided the basis for the evolutionary techniques implemented for data exploration. Using multi-objective genetic algorithms that simultaneously sought to minimize the residual variance, complexity and number of predictors, a large number of candidate solutions were generated for both flight conditions. This approach was compared with principal component analysis and was found to yield superior models. The genetic algorithms were able to find solutions with a greatly reduced number of predictor variables, and therefore omit many irrelevant parameters. From analyzing these solutions, some clear trends became apparent. Three flight state and control system parameters in particular appeared prominently in the top masks for all flight conditions and all target sensors: main rotor shaft torque, tail rotor drive shaft torque, and yaw acceleration. Their overwhelming presence in the MOGA-Γ solutions suggest that these parameters are essential in predicting the main rotor normal bending during the two flight conditions examined in this study. While examination of these relationships is still in the early phases, the analysis thus far has enabled a better understanding of the loads in the critical components. Future work will seek to alter the training scheme to increase the chances of finding more accurate solutions particularly for the cases where accurate solutions were not found. Further investigation into the flight state and control system parameters is also still required and a better understanding of the mechanics of the flight conditions could still be achieved.

The predictions for the main rotor normal bending using the reduced masks found by the MOGA and Gamma test were reasonably accurate and correlated and performed similarly to the models built using the full set of input variables. Improvements to the predictions could still be attained, particularly at the peak values of the target signal. The results from the PCA subset showed that PCA-based feature selection was not appropriate for this application. Future work should incorporate other feature selection methods as well as alternative methods to form the training/testing sets within the data exploration phase. At the modeling stage other computational intelligence techniques and different error measures should be considered.

Acknowledgements This work was supported in part by Defence Research and Development Canada (13pt). Access to the data was granted by Australia's Defence Science and Technology Organisation.

References

1. Anderberg, M.: Cluster Analysis for Applications. Wiley, London (1973)
2. Bäck, T., Fogel, D., Michalewicz, Z.: Handbook of Evolutionary Computation. Institute of Physics Publishing and Oxford University Press, UK (1997)
3. Burke, E., Kendall, G.: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Springer Science and Business Media, New York (2005)
4. Clerc, M.: Particle Swarm Optimization. ISTE, London (2006)

5. Deb, K., Agarwal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceeding of Parallel Problem Solving from Nature VI Conference, pp. 849–858. Paris, France, 16–20 Sept 2000
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. Technical Report 2000001, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur (2000)
7. Deb, K., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), pp. 181–197 (2002)
8. Evans, D., Jones, A.: A proof of the gamma test. Proc. R. Soc. Lond. A **458**, 1–41 (2002)
9. Gämperle, R., Müller, S.D., Koumoutsakos, P.: A parameter study for differential evolution. In: WSEAS International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 293–298. (2002, in press)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman, Boston (1989)
11. Jones, A., Evans, D., Margetts, S., Durrant, P.: The gamma test. In: Sarker, R., Abbass, H., Newton, C. (eds.) Heuristic and Optimization for Knowledge Discovery, pp.142–168. Idea Group, Hershey (2002)
12. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proceeding of IEEE International Joint Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
13. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2002)
14. Kukkonen, S., Lampinen, J.: An empirical study of control parameters for generalized differential evolution. Technical Report 2005014, Kanpur Genetic Algorithms Laboratory (KanGAL) (2005)
15. Georgia Tech Research Institute.: Joint USAF-ADF S-70A-9 flight test program, summary report. Technical Report A-6186, Georgia Tech Research Institute (2001)
16. Pareto, V.: Cours D'Economie Politique, vols. **I** and **II**. F. Rouge, Lausanne (1896)
17. Polanco, F.: Estimation of structural component loads in helicopters: a review of current methodologies. Technical Report DSTO-TN-0239, Defence Science and Technology Organisation (1999)
18. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes in C. 2nd edn. Cambridge University Press, Cambridge (1992)
19. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: A practical approach to global optimization. Natural Computing Series. Springer, Berlin Heidelberg (2005)
20. Stefánsson, A., Končar, N., Jones, A.: A note on the gamma test. Neur. Comput. Appl. **5**, 131–133 (1997)
21. Storn, R., Price, K.: Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report tr-09012, ICSI (1995)
22. Takens, F.: Detecting strange attractors in turbulence. Dynamical systems and turbulence. Lecture Notes in Mathematics, vol. 898, pp. 366–381 (1981)
23. Valdés, J.J., Cheung, C., Wang, W.: Evolutionary computation methods for helicopter loads estimation. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC2011). pp. 1589–1596. New Orleans, LA, USA (June 2011)
24. Valdés, J.J., Cheung, C., Wang, W.: Computational intelligence methods for helicopter loads estimation. In: Proceedings of International Joint Conference on Neural Networks (IJCNN2011), pp. 1864–1871. San Jose, CA, USA (Aug 2011)
25. Zheng, Y., Ma, L., Zhang, L., Qian, J.: Study of particle swarm optimizer with an increasing inertia weight. In: Proceedings of World Congress on Evolutionary Computation, pp. 221–226. Canberra, Australia (Dec 2003)

Multilayer Semantic Analysis in Image Databases

Ismail El Sayad, Jean Martinet, Zhongfei (Mark) Zhang and Peter Eisert

Abstract With the availability of massive amounts of digital images in personal and on-line collections, effective techniques for navigating, indexing and searching images become more crucial. In this article, we rely on the image visual content as the main source of information to represent images. Starting from the bag of visual words (BOW) representation, a high-level visual representation is learned where each image is modeled as a mixture of visual topics depicted in the image and related to high-level topics. First, we introduce a new probabilistic topic model, Multilayer Semantic Significance Analysis (MSSA) model, in order to study a semantic inference of the constructed visual words. Consequently, we generate the Semantically Significant Visual Words (SSVWs). Second, we strengthen the discrimination power of SSVWs by constructing Semantically Significant Visual Phrases (SSVPs) from frequently co-occurring SSVWs that are semantically coherent. We partially bridge the intra-class visual diversity of the images by re-indexing the SSVWs and the SSVPs based on their distributional clustering. This leads to generating a Semantically Significant Invariant Visual Glossary (SSIVG) representation. Finally, we propose a new Multiclass Vote-Based Classifier (MVBC) based on the proposed SSIVG representation. The large-scale extensive experimental results show that the proposed higher-level visual representation outperforms the traditional part-based image representations in retrieval, classification, and object recognition.

I. El Sayad (✉) · P. Eisert
Fraunhofer Heinrich Hertz Institute, Berlin, Germany
e-mail: is.elsayad@gmail.com

J. Martinet
Lille 1 University, Villeneuve d'ascq, France
e-mail: jean.martinet@lifl.fr

Z. (Mark) Zhang
Computer Science Department, SUNY at Binghamton, NY 13905, USA
e-mail: zhongfei@cs.binghamton.edu

P. Eisert
e-mail: peter.eisert@hhi.fraunhofer.de

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0_19

1 Introduction

With the increasing convenience of capturing devices and the wide availability of large capacity storage devices, the amount of digital images that ordinary people can reach has become particularly wide. This huge amount is useless if there is no effective way to retrieve the desired images. The usual way to solve this problem consists in describing images by keywords. Since images are annotated manually, this method suffers from subjectivity and textual ambiguity [15]; however, images can be indexed via an automatic description which only depends on their objective visual content. When considering visual documents, the problem of the semantic gap arises. The notion of semantic gap has been defined as the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation [26]. Many techniques make use of several chains of processes, in order to obtain a hierarchal semantic representation of images that can lower this gap. Recently, the Bag of Visual Words (BOW) image representation [31] has drawn much attention, as it tends to code the local visual characteristics towards the object level, which is closer to the perception of human visual systems [40]. Besides the significant performance of the BOW representation, there still are drawbacks to be considered. This article that is an substantially revised extension of [12] aims at addressing some of these drawbacks with the contributions highlighted as follows. Firstly, a new probabilistic topic model, *Multilayer Semantic Significance Analysis* (MSSA), is introduced in order to select *Semantically Significant Visual Words* (SSVWs) from the constructed visual words based on the probability of their distributions to the relevant visual latent topics. This model differs from the pLSA model [19] and LDA [5] model by introducing two layers of latent topics: high and visual latent topics. One layer represents the high-level aspects (i.e., image categories) and the other one represents the visual aspects (i.e., objects, parts of objects or scenes). Secondly, the *low discrimination power* of visual words leads to low correlations between the image features and their semantics [38, 42]. In our work, we build a higher-level representation, namely the *semantically significant visual phrase* (SSVP) from groups of adjacent significant visual words that co-occur frequently and are semantically coherent. Thirdly, the images of the same semantic class can have arbitrarily different visual appearances and shapes. Such visual diversity of object causes one image semantics to be represented by different SSVWs and SSVPs. In this circumstance, the SSVWs and SSVPs become too primitive to effectively model the image semantics, as their efficacy depends highly on the visual similarity and regularity of images of the same semantics. To tackle this issue, We run a distributional clustering for the SSVWs and SSVPs inured to generate a *Semantically Significant Invariant Visual Glossary* (SSIVG) representation. Fourthly, we have conducted large-scale, extensive experimental evaluations regarding the performances of social image retrieval in comparison with various state-of-the-art image representation methods from the recent literature to demonstrate the superiority of the proposed higher-level visual representation methods. The remainder of the article is structured as follows. We review the related work in Sect. 2. In Sect. 3, we propose

the new MSSA model based on different latent topics (visual and high latent topics) in order to generate the SSVWs. We generate the SSVPs in Sect. 4. We introduce the final representation, SSIVG, in Sect. 5. In Sect. 6, we describe the usage of the SSVIG representation in image indexing, retrieval, and classfication. We report the experimental results in Sect. 7, and we give a conclusion to this article in Sect. 7.4.

2 Related Work

2.1 Part-Based Image Representation

Liu et al. [25] provided a thorough survey on the literature of image retrieval systems. The image representation for the previous image retrieval systems can be generally classified into two types: (1) image-based or grid-based global features like color, color moment, shape or texture histogram over the whole image or grid; and (2) part-based bag-of-words features extracted from segmented image regions, salient key points, and blobs. As we mentioned in the introduction, part-based bag-of-visual-words (BOW) representation was reported as more robust than the traditional global features in handling changes in scale, pose, and illumination [27, 36]. Inspired by the success of the vector-space model [30] for text document representation and retrieval, the bag-of-visual-words (BOW) approach usually converts images into vectors of visual words based on their frequencies. In BOW approach, the vocabulary creation process, based on clustering algorithms such as k-means, is quite rude and leads to many noisy words. Such words add ambiguity in the image representation. This problem has been addressed in the first video-Google paper by Sivic and Zisserman [31]. They used stop-lists that remove the most and least frequent words from the collection. Yang et al. [36] pointed out the ineffectiveness of this method and proposed several measures usually used in feature selection for machine learning or text retrieval. Another evident drawback in BOW representation is the spatial information loss. To overcome this, Lazebnik et al. [22] extended the BOW representation to Spatial Pyramid Matching Kernel (SPM) by exploiting the spatial information of location regions. Recently, Yang et al. [37] tackled the two drawbacks (quantization rudeness and spatial information loss) and proposed an extension of SPM by replacing k-means with Sparse Coding. In sparse coding and feature selection techniques, local features are dealt separately. The mutual dependence and interrelation among local features are ignored. However, recent work shows that the relationships among the local features are important for image representation, such as the geometric relationship [35]. Gao et al. [17] introduced Laplacian sparse coding to enhance the sparse coding by constructing a Laplacian matrix, which can well characterize the similarity between local features. This representation, however, lacks semantic learning that would better characterize the semantic relationships between the visual words. To address the discrimination or polysemy problem of visual words, Zheng and Gao [41] made an analogy between image retrieval and text retrieval, and proposed a higher-level representation *visual phrase* for object-based image retrieval.

Visual phrases are defined as pairs of adjacent frequent visual words. Recently, Zhang et al. [40] enhance this approach by selecting descriptive visual phrases from the constructed visual phrases according to the frequencies of their constituent visual word pairs. In these two approaches, the higher-level (visual phrase) is defined as adjacent pairs of visual words which do not necessarily guarantee a truly meaningful descriptive visual representation [38]. In addition, there are ambiguities in visual word lexicons. If the generation of the representation is a pure bottom-up process, the imperfectness in the visual words would never be reduced, and the quantization error would never be corrected without a pre-filtering step for the visual words done at a lower level. Yuan et al. [38] proposed another higher-level representation, i.e., visual phrase pattern, where a visual phrase is a spatially co-occurent group of visual words. The main contribution of this approach is to present a solution to the discovery of significant spatial co-occurent patterns using the frequent item set mining. Zheng et al. [42] proposed a similar approach by constructing another high-level, delta visual phrase, and grouped delta visual phrases according to their similarity to visual synsets. Both approaches evaluated the significance of the visual phrases statistically. Zheng et al. [42] addressed the importance of the semantic factor but they measured the significance of a delta visual phrase based on its frequency as well as the frequencies of its constituent visual words.

2.2 Probabilistic Topic Models for Semantic Learning in Image Databases

Among the existing methods that extract statistical characteristics, the probabilistic topic models play an important role. The key idea behind the applying topic models in images is how to capture the essential statistical characteristics of the visual representation units (i.e. visual words). By capturing and learning the statistical characteristics of the visual representation units, one gives the images a new representation, which is often more parsimonious and less noise-sensitive. Probabilistic topic models extract a set of latent topics from a corpus and as a consequence represent the images in a new latent semantic space. One of the well-known topic models is the Probabilistic Latent Semantic Analysis (pLSA) model proposed by Hofmann [19] for text document semantic analysis and is applied later to images. In pLSA each image is modeled as a probabilistic mixture of a set of topics. Going beyond PLSA, Blei et al. [5] presented the Latent Dirichlet Allocation (LDA) model by incorporating a prior for the topic distributions. In these probabilistic topic models, one assumption underpinning the generative process is that images are independent and one layer of topics are proposed. Lienhart et al. [23] introduced a new model named multilayer multimodal probabilistic Latent Semantic Analysis (mm-pLSA). They derive the training and inference rule for the smallest possible non-degenerated mm-pLSA model: a model with two leaf-pLSAs (here from two different data modalities: image tags and visual image features) and a single top-level pLSA node merging the two leaf-pLSAs. From this derivation, it is obvious how to extend the learning and inference rules to more

modalities and more layers. Even though this approach introduced a new multilayer inference rules, it uses an EM algorithm to derive the different parameters, which costs a high computational power for the parameter initialization and estimation. In addition, this approach did not introduce any criterion to estimate the number of different latent variables. Our framework differs from these approaches by proposing a new probabilistic topic model, Multilayer Semantic Significance Analysis (MSSA) model, to analyze the semantic significance of the visual words in order to overcome the rudeness of quantization. We also utilize MSSA to check the semantic coherence of groups of Semantically Significant Visual Words (SSVWs) that are spatially adjacent and frequently occur with each other in order to construct another higher-level representation.

3 Semantically Significant Visual Words (SSVWs) Using Multilayer Semantic Significance Analysis (MSSA) Model

As we mentioned in the introduction, the vocabulary creation process in the BOW image representation based on clustering algorithms such as K-means, is quite rude and can lead to many noisy visual words. Such visual words add ambiguity in the image representation. Thus, it reduces the effectiveness of the visual representation in retrieval or classification; then a statistical criterion is needed to study the semantic significance of the constructed visual words. In our understanding, every image is assumed to consist of one or more visual aspects, which in turn are combined into the higher-level aspects. This is very natural since images consist of multiple objects or scenes, which belong to different categories or classes. Figure 1 shows an example of different high-level and visual aspects in some images. In this figure, *face* can be a visual aspect and *person* can be the high-level aspect. This leads to designing a new probabilistic topic model that studies the semantic inferences of the visual words and takes in consideration the hierarchal consistency of the image, without adding much complexity in the process of the parameters initialization.

3.1 Visual Words (VWs) Creation

We utilize the SURF [3] descriptor with the Edge Context descriptor [14]. The Edge Context describes the distribution of the edge points in the same Gaussian (by returning to the 5-dimensional color-spatial feature space). It is represented as a histogram of 6 bins for R (*magnitude* of the drawn vector from the interest point to the edge points) and 4 bins for θ (*orientation angle*).

Finally, both descriptors are fused to form a feature vector composed of 88 dimensions (64 from SURF + 24 from the Edge context descriptor). Hence, the new feature vector describes the information on the distribution of the intensity and the

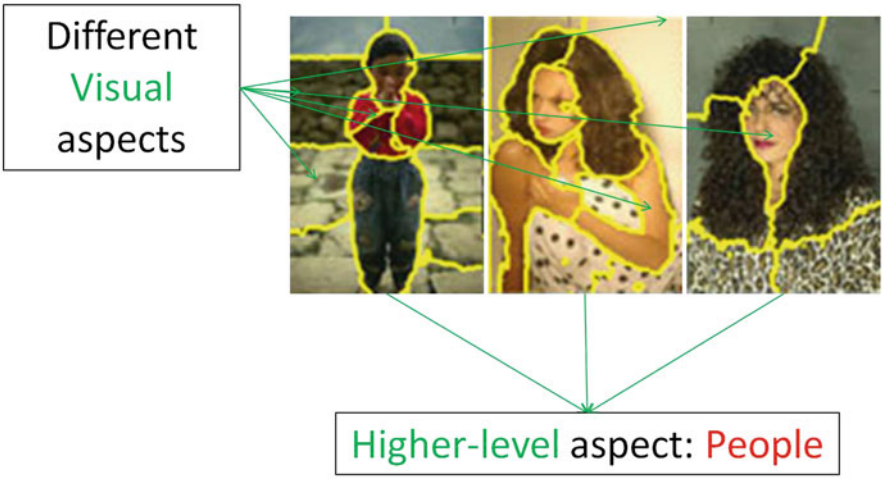


Fig. 1 Examples of different visual and higher-level aspects

edge points of the image. The quantization of the features into visual words is performed by using a vocabulary tree. The vocabulary tree is computed by repeated k-means clustering that hierarchically partitions the feature space [13].

3.2 Generative Process

Suppose that we have N images $\{im_j\}_{j=1}^{N}$ in which M visual words $\{VW_i\}_{i=1}^{M}$ are observed. We introduce two layers of topics. High latent topics represent the high-level aspects (i.e., image categories) and visual latent represent the visual aspects (i.e., objects, parts of objects, or scenes). The following generative process for a given image im_j:

- Choose a high latent topic h_k from $P(h_k|im_j)$, a multinomial distribution conditioned on im_j and parameterized by a $K \times N$ stochastic matrix θ, where $\theta_{kj} = P(h_k = k|im_j = j)$.
- Choose a visual latent topic v_l from $P(v_l|h_k)$, a multinomial distribution conditioned on h_k and parameterized by an $L \times K$ stochastic matrix φ, where $\varphi_{lk} = P(v_l = l|h_k = k)$.
- Generate a visual representation unit VW_i from $P(VW_i|v_l)$, a multinomial distribution conditioned on v_l and parameterized by an $M \times L$ stochastic matrix Ψ, where $\Psi_{il} = P(VW_i = i|v_l = l)$.

This generative process leads to the following conditional probability distribution:

$$P(VW_i|im_j) = \sum_{k=1}^{K} \sum_{l=1}^{L} P(h_k|im_j, \theta) P(v_l|h_k, \varphi) P(VW_i|v_l, \Psi). \quad (1)$$

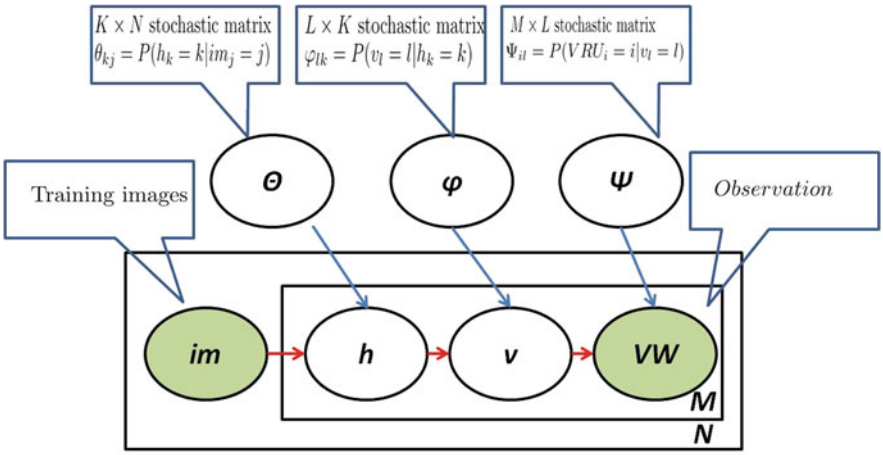


Fig. 2 The semantic model using the plate notation

Following the maximum likelihood principle, one can estimate the parameters by maximizing the log-likelihood function as follows:

$$Li = \sum_{j=1}^{N} \sum_{i=1}^{M} n(VW_i, im_j) \log(P(VW_i|im_j)), \tag{2}$$

where $n(VW_i, im_j)$ denotes the number of the occurance of VW_i in im_j. Figure 2 depicts the generative process using the plate notation.

3.3 Parameter Estimation

The expectation-maximization (EM) algorithm [9] is the standard approach for maximum likelihood estimation in latent variable models. The main difficulty when implementing the EM algorithm in this work is that a four dimensional matrix is required in the E-step because of the two latent variables, which induces a high complexity. However, Gaussier et al. [18] have proven that maximizing the likelihood can be seen as a Non-Negative Matrix Factorization (NMF) problem under the generalized KL divergence. This leads to the following objective function:

$$min_{\theta,\varphi,\Psi} GL(A, \Psi\varphi\theta), \tag{3}$$

where Ψ, φ, and θ are stationary points, A is the observation matrix, and $GL(A, \Psi\varphi\theta)$ is the generalized KL divergence such that:

$$\theta \in \mathbb{R}_{+}^{K \times N}, \theta^T 1 = 1, \tag{4}$$

$$\varphi \in \mathbb{R}_{+}^{L\times K}, \varphi^T 1 = 1, \tag{5}$$

$$\Psi \in \mathbb{R}_{+}^{M\times L}, \Psi^T 1 = 1, \tag{6}$$

$$A_{ij} = \frac{n(VW_i, im_j)}{\sum_{i,j} n(VW_i, im_j)}, \tag{7}$$

$$GL(A, \Psi\varphi\theta) = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(A_{ij}\log\frac{A_{ij}}{[\Psi\varphi\theta]_{i,j}} - A_{ij} + [\Psi\varphi\theta]_{ij}\right). \tag{8}$$

3.3.1 Karush Kuhn Tucker (KKT) Conditions

We use the Karush Kuhn Tucker (KKT) conditions [21] to derive the multiplicative update rules for minimizing (3) since it can be formulated as a constrained minimization problem with the following inequality constraint:

$$\Psi_{il} > 0, \tag{9}$$

$$\varphi_{lk} > 0, \tag{10}$$

$$\theta_{kj} > 0. \tag{11}$$

The necessary KKT conditions for a minimum of the constrained problem stated above are obtained by using the Lagrange multiplier method. Let α_{il}, β_{lk}, γ_{kj} be the Lagrangian multipliers associated with the constraints $\Psi_{il}, \varphi_{lk}, \theta_{kj}$ respectively. The KKT conditions require the following optimality conditions:

$$\alpha_{il} = \frac{\partial GL(A, \Psi\varphi\theta)}{\partial\Psi_{il}} = \sum_{j=1}^{N}\left\{[\varphi\theta]_{lj} - \frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}\right\}, \tag{12}$$

$$\beta_{lk} = \frac{\partial GL(A, \Psi\varphi\theta)}{\partial\varphi_{lk}} = \sum_{i=1}^{M}\sum_{j=1}^{N}\left\{\Psi_{il}\theta_{kj} - \frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}\Psi_{il}\theta_{kj}\right\}, \tag{13}$$

$$\gamma_{kj} = \frac{\partial GL(A, \Psi\varphi\theta)}{\partial\theta_{kj}} = \sum_{i=1}^{M}\left\{[\Psi\varphi]_{ik} - \frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\Psi\varphi]_{ik}\right\}. \tag{14}$$

This leads to the following:

$$\sum_{j=1}^{N}\left\{[\varphi\theta]_{lj} - \frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}\right\} = \alpha_{il}, \tag{15}$$

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\left\{\Psi_{il}\theta_{kj}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}\Psi_{il}\theta_{kj}\right\}=\beta_{lk}, \tag{16}$$

$$\sum_{i=1}^{M}\left\{[\Psi\varphi]_{ik}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\Psi\varphi]_{ik}\right\}=\gamma_{kj}. \tag{17}$$

The following complementary slackness conditions are also required:

$$\alpha_{il}\Psi_{il}=0, \tag{18}$$

$$\beta_{lk}\varphi_{lk}=0, \tag{19}$$

$$\gamma_{kj}\theta_{kj}=0. \tag{20}$$

3.3.2 New Multiplicative Update Rules for NMF

The minimization of the objective function (3), should be done with non-negativity constraints as described in Sect. 3.3.1. A multiplicative updating is an efficient way in such a case since it can easily preserve the non-negativity constraints at each iteration. The proposed multiplicative updating algorithms for NMF associated with the objective functions (3) are given as follows: Multiplying both sides of (15–17) by Ψ_{il}, φ_{lk}, and θ_{kj} respectively, leads to the following:

$$\left[\sum_{j=1}^{N}\left\{[\varphi\theta]_{lj}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}\right\}\right]\Psi_{il}=\alpha_{il}\Psi_{ij}, \tag{21}$$

$$\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\left\{\Psi_{il}\theta_{kj}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}\Psi_{il}\theta_{kj}\right\}\right]\varphi_{lk}=\beta_{lk}\varphi_{ij}, \tag{22}$$

$$\left[\sum_{i=1}^{M}\left\{[\Psi\varphi]_{ik}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\Psi\varphi]_{ik}\right\}\right]\theta_{kj}=\gamma_{kj}\theta_{ij}. \tag{23}$$

Incorporating (21–23) with (18–20), leads to the following:

$$\left[\sum_{j=1}^{N}\left\{[\varphi\theta]_{lj}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}\right\}\right]\Psi_{il}=0, \tag{24}$$

$$\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\left\{\Psi_{il}\theta_{kj}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}\Psi_{il}\theta_{kj}\right\}\right]\varphi_{lk}=0, \tag{25}$$

$$\left[\sum_{i=1}^{M}\left\{[\Psi\varphi]_{ik}-\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\Psi\varphi]_{ik}\right\}\right]\theta_{kj}=0. \tag{26}$$

This suggests the following iterative multiplicative update rules:

$$\Psi_{il}\leftarrow\Psi_{il}\frac{\sum_{j=1}^{N}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}}{\sum_{j=1}^{N}[\varphi\theta]_{lj}}, \tag{27}$$

$$\varphi_{lk}\leftarrow\varphi_{lk}\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\Psi_{il}\theta_{kj}}, \tag{28}$$

$$\theta_{kj}\leftarrow\theta_{kj}\frac{\sum_{i=1}^{M}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}}{\sum_{i=1}^{M}[\Psi\varphi]_{ik}}. \tag{29}$$

A small positive parameter ε, with value 10^{-9}, is added to (27–29) in order to avoid a division by zero as follows:

$$\Psi_{il}\leftarrow\Psi_{il}\frac{\sum_{j=1}^{N}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}}{\sum_{j=1}^{N}[\varphi\theta]_{lj}+\varepsilon}, \tag{30}$$

$$\varphi_{lk}\leftarrow\varphi_{lk}\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\Psi_{il}\theta_{kj}+\varepsilon}, \tag{31}$$

$$\theta_{kj}\leftarrow\theta_{kj}\frac{\sum_{i=1}^{M}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}}{\sum_{i=1}^{M}[\Psi\varphi]_{ik}+\varepsilon}. \tag{32}$$

Also, some normalizing coefficients (λ, μ, and ν) are added to (30–32) with the aim of satisfying the normalization constraints:

$$\Psi_{il}\leftarrow\lambda\Psi_{il}\frac{\sum_{j=1}^{N}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}[\varphi\theta]_{lj}}{\sum_{j=1}^{N}[\varphi\theta]_{lj}+\varepsilon}, \tag{33}$$

$$\varphi_{lk}\leftarrow\mu\varphi_{lk}\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\Psi_{il}\theta_{kj}+\varepsilon}, \tag{34}$$

$$\theta_{kj}\leftarrow\nu\theta_{kj}\frac{\sum_{i=1}^{M}\frac{A_{ij}}{[\Psi\varphi\theta]_{ij}}}{\sum_{i=1}^{M}[\Psi\varphi]_{ik}+\varepsilon}. \tag{35}$$

The application of the final multiplicative update rules (33–35) find at least locally optimal solutions for the objective function (3), where all the different parameters (Ψ, φ, θ) are estimated. Therefore, the semantic inferences of the observed visual representation units (visual words) are known and can be used for further semantic analysis. We would like to highlight that in this form, the proposed multiplicative update rules themselves are extremely easy to implement computationally.

3.4 Number of the Latent Topics Estimation

In many probabilistic models, the number of latent topics is usually not known in advance. In the proposed model, the number of the high-level latent topics, L, and the number of the visual latent topics, K, is determined in advance for the model fitting based on the Minimum Description Length (MDL) principle [29] to maximize

$$Li - \frac{m_k}{\log(NM)}, \tag{36}$$

where the first term is a log-likelihood function expressed in (2), m_k is the number of the free parameters needed for the model, N is the number of the images in the dataset, and M is the visual word vocabulary size. In our model, m_k is expressed as follows:

$$m_k = ML + LK + KN. \tag{37}$$

As a consequence of this principle, when models with different values of K and L fit the data equally well, the simpler model is selected.

3.5 Semantically Significant Visual Words (SSVWs) Generation

After we have estimated the different probability distributions $P(h_k|im_j)$, $P(v_l|h_k)$, and $P(\mathrm{VW}_i|v_l)$ using the MSSA, all the visual latent topics v_l are categorized according to their conditional probabilities with all the high-level latent topics $P(v_l|h_k)$. All the visual latent topics whose conditional probabilities relating to all the high latent topics are higher than a given threshold t_{h_k} are categorized as relevant. Given a set of the relevant visual topics, a Semantically Significant Visual Word (SSVW) is defined as follows.

Definition 1 (**Semantically Significant Visual Word**) *An SSVW is a visual word (VW) whose conditional probability $P(\mathrm{VW}_i|v_l)$ is higher than a given threshold t_{v_l} for at least one relevant visual latent topic.*

From our perspective, all the visual words whose probability distributions $P(\mathrm{VW}_i|v_l)$ are low for every relevant visual topic are irrelevant, since they are not informative for any relevant visual topic. Hence, we propose to keep only the most

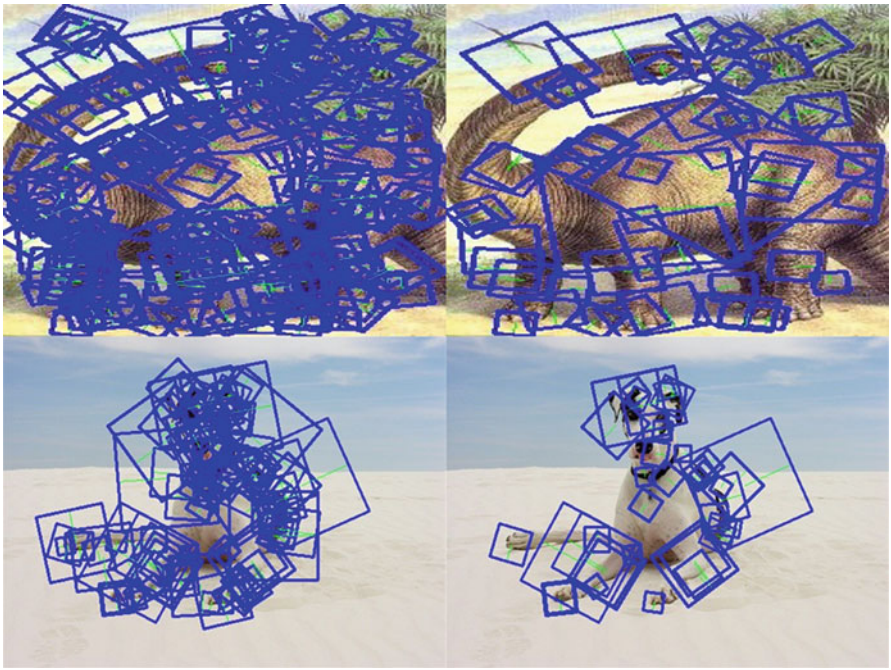


Fig. 3 The left side of the figure is an example of an image represented by VWs and the right side is the same image represented by SSVWs

significant visual words for each relevant visual topic. Figure 3 gives an example of an image represented by VWs and SSVWs.In this figure, it is clear that there is a huge difference in the number of visual words between the images in the right and left side. Moreover, it is obvious that the visual words that represent the images in the right side are deemed to be more semantically since most of them are describing different parts of the main objects (Dinosaur and dog).

4 Semantically Significant Visual Phrases (SSVPs) Generation

The low discrimination power of the SSVWs leads to low correlations between the image features and their semantics. An SSVW represents different semantic meanings in different image contexts. This encumbers the distinctiveness of the SSVWs and leads to a low discrimination. In fact, the discrimination issue is a problem of under-representation [39]. Its consequence is effectively small interclass distances [42]. One of the major reasons for the low discrimination issue is that the regions represented in a visual word might come from the objects with different semantics but similar local appearances. Figure 4 gives an example of two SSVWs that share visual similarity in two different categories (*car* and *motorbike*). The

Fig. 4 An example of the low discrimination power of the SSVWs

SSVW A is, therefore, not able to distinguish $motorbike$ from car. However, SSVW A and SSVW B considered together can effectively distinguish $motorbike$ from car. The discrimination of representation can therefore be improved by mining interrelations among SSVWs in a certain neighborhood region in order to construct a more discriminative higher-level representation. Such low correlation motivates to generate a higher-level discriminative visual representation, named Semantically Significant Visual Phrase (SSVP). The SSVWs and their inter-relationships are the basis for generating SSVPs [11], which are defined as follows.

Definition 2 (**Semantically Significant Visual Phrase**) *We define an SSVP as a set of Semantically Significant Visual Words (SSVWs) that frequently co-occurr together in a spatial local context, involved in strong association rules, and semantically coherent.*

Since it is not easy to define the semantic coherence in a set of SSVWs, we assume the following:

Assumption 1 (**Semantically Coherent Set of SSVWs**) *A set of SSVWs are semantically coherent whenever they have a high probability regarding to at least one common relevant visual latent topic. Their probability distributions are estimated using the MSSA model.*

In this section, we discuss the different processes for generating the SSVPs. These processes start by defining the local neighborhood of a given SSVW, and finish by generating a representation scheme for the constructed SSVP vocabulary.

4.1 Spatial Local Context

Several methods have been proposed to sample spatial neighborhoods within an image. In [8] a sliding-window mechanism samples windows at a fixed location and scale step, followed by a spatial tiling of the windows. The very different approach [32] defines a neighborhood around each region. This is represented as an

unordered set of the k nearest regions, without storing any spatial information (k-neighborhoods). However, the neighborhoods in this case are always of a fixed size. Our approach attempts to combine the best of them. Instead of using a k-neighborhood, we use the *scale* of the center of the local patch to define the size of the neighborhood and all SSVWs (not just pairs of SSVWs) that occur within this context are considered in the SSVP generation process.

4.2 Association Rules and SSVP Generation

After defining the local context, we apply the association rule mining theory [1] to find the frequent item set with the SSVWs. $T = \{T_1, T_2, \ldots, T_n\}$ is the set of all the different sets of SSVWs located in the same context which denotes the set of transactions. An association rule is a relation of an expression $X \Rightarrow Y$, where X and Y are sets of items (sets of one or more of SSVWs that are within the same context). The quality of a rule can be described in the support-confidence framework. The support of a rule measures the statistical significance of a rule

$$support(X \Rightarrow Y) = \frac{|\{T_i \in T | (X \cup Y) \subset T_i\}|}{|T|}. \tag{38}$$

The confidence of a rule measures the strength of the implication $X \Rightarrow Y$.

$$con\ f(X \Rightarrow Y) = \frac{|\{T_i \in T | (X \cup Y) \subset T_i\}|}{|\{T_i \in T | X \subset T_i\}|}. \tag{39}$$

By applying the Apriori algorithm [1] to a set of images, we discover all the strong association rules, which have a support and confidence greater than the predefined thresholds. Finally, all the frequent SSVW sets that occur in the same context and are semantically coherent form SSVPs. Figure 5 shows examples of SSVPs corresponding to different visual aspects. The square resembles a local patch; the red circle around the center of the local patch denotes the local context, and the group of local patches in the same context denotes an SSVP.

5 Semantically Significant Invariant Visual Glossaries (SSIVGs) Generation

Even though studying the co-occurrence and spatial scatter information makes the image representation more distinctive, the invariance power of SSVWs or SSVPs is still low. Returning to text documents, synonymous words (different words with same meaning) are usually clustered into one synonym set to improve the document categorization performance [4]. Such an approach inspires us to partially bridge the visual diversity of the images by clustering the SSVWs and the SSVPs based on their

Fig. 5 Examples of SSVPs appear in different images

probability distributions to all the relevant visual latent topics. After the distributional clustering, *each group of SSVWs that belongs to a given cluster is re-indexed with the same index as the cluster centroid*. This leads to generate Semantically Significant Invariant Visual Words (SSIVWs) which consist of SSVWs that are re-indexed after the distributional clustering. In the same manner we generate the Semantically Significant Invariant Visual Phrase (SSIVP). Finally, both the SSIVWs and the SSIVPs form the Semantically Significant Invariant Visual Glossary (SSIVG) representation.

Definition 3 (**Semantically Significant Invariant Visual Glossary**) *Semantically Significant Invariant Visual Glossary (SSIVG) representation is a higher-level visual*

representation composed two different layers of representations: Semantically Significant Invariant Visual Word (SSIVW) representation and Semantically Significant Invariant Visual Phrase (SSIVP) representation, where an SSIVW (resp. SSIVP) is an SSVW (resp. SSVP) that has been re-indexed after a distributional clustering.

In this section, we discuss the invariance problem. Then, the MSSA model is run one more time with the new observations that are built upon the co-occurrences of the SSVWs and the SSVPs in order to estimate the new probability distributions for both of them. Based on the estimated probabilistic inferences, we cluster SSVs and the SSVPs. Finally, the SSVWs and SSVPs are re-indexed to form the SSIVW and SSIVP respectively.

5.1 Low Invariance of the SSVWs and SSVPs

The images in a given semantic class can have arbitrarily different visual appearances and shapes. Such visual diversity of objects causes one visual aspect to be possibly represented by different SSVWs and SSVPs. This leads to low invariance of SSVWs and SSVPS. The consequence is the large intra-class variations. In this circumstance, the SSVS and SSVPS become too primitive to effectively model the image semantics, as their efficacy depends highly on the visual similarity and regularity of images of the same semantics.

Figure 6 gives an example of the invariance problem in two images of motorcycles. The two different SSVWs ($SSVW_{605}$, $SSVW_{1076}$) occurring in the two images describe the same part of the of motorcycle; however they are different, and therefore, they have different indexes (605 and 1076). Also, the two SSVPs ($SSVP_{148}$, $SSVP_{263}$) describe the same part of the motorcycle (part of the wheels), and they are different indexes (148 and 263). This happens since the two images are for the same object (motorcycle), yet with different shapes and colors. This leads to extracting different low-level features from the two images. In the text domain, when documents of the same topic or category contain different sets of words, the word synset (synonym set) that links words of similar semantics is robust to model them [4]. Inspired by this, we propose that relevance-consistent groups of the SSVWs or SSVPs with similar semantic inferences should have the same index.

5.2 New Generative Process

After generating the SSVWs and the SSVPs, the co-occurrence of both forms new observations. We study the semantic inferences for the SSVWs and the SSVPs after the new observations. The same MSSA model that is introduced in Sect. 3 is run with the co-occurrences of the SSVPs and the SSVWs as the observation matrix. After re-running the MSSA according to the above generative processes, the new probability distributions for SSVWs and SSVPs to different visual latent topics are estimated.

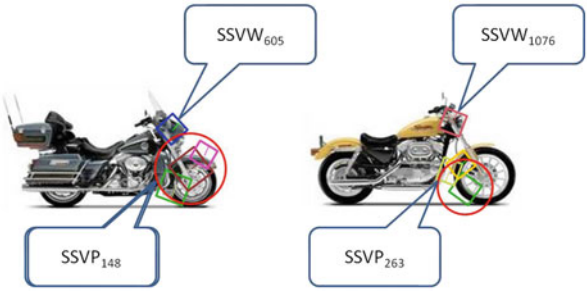


Fig. 6 Illustrations of the invarince problem: similar image regions are indexed with different SSVWs and SSVPs

5.3 *Distributional Clustering for SSVWs and SSVPs*

After estimating the new semantic inferences of the SSVWs and the SSVPs, the next step is to group the SSVWs that are with similar probabilistic inferences. Similarly, the SSVPs that share similar semantic inferences are also grouped. In our approach, we use an information-theoretic framework that was introduced by Dhillon et al. [10].

Algorithm 4: Divisive Information Theoretic Clustering (P,ψ,l,k,W)

Input:
P is the set of distributions, $\mathrm{p}(SSVW_t|v_l) : 1 \le t \le M$, Π is the set of all SSVW priors, $\pi = p(SSVW_t) : 1 \le t \le M$, L is the number of the visual latent topics, K is the number of the desired clusters.
Output: C is the set of word clusters $c_1,c_2,.\psi\psi\psi..,c_K$.

1. Initialization: for every SSVW $SSVW_t$, assign $SSVW_t$ to C_q such that $\mathrm{p}(SSVW_t|v_l) = maxip(SSVW_t|v_l)$.This gives L' initial SSVW clusters; if $Q \ge L$ split each cluster arbitrarily into at least $\lfloor K/L \rfloor$ clusters; otherwise merge the L' clusters to get Q SSVW clusters.
2. For each cluster c_k, compute $\pi(\mathrm{c_k}) = \sum_{\mathrm{g_t} \in c_k} \pi(SSVW_t)$, $\mathrm{p}(c_k|v_l) = \sum_{SSVW_t \in c_k} \frac{\pi_t}{\pi(c_k)} p(SSVW_t|v_l)$.
3. Re-compute all clusters: For each $SSVW_t$, find its new cluster index as $\mathrm{j} * (SSVW_t) = argmin_i KL(p(SSVW_t|v_l), p(c_k|v_l))$, resolving ties arbitrarily. Thus compute the new SSVW clusters $c_k, 1 \le k \le K$, as $c_k = SSVW_t : j * (SSVW_t) = k$.
4. Measure the quality of SSVW clustering by the following objective function:

$$Q(\{\mathrm{c_k}\}_{k=1}^{K}) = I(SSVW_i; v_l) - I(c_k; v_l) = \sum_{k=1}^{\mathrm{K}} \sum_{SSVW_t \in c_k} \pi_t KL(p(SSVW_t|v_l)) (\mathrm{p}(c_k|v_l))$$

5. Stop if the change in the objective function value given by (40) is small (10^{-3}); Else go to step 2.

This framework is similar to Information Bottleneck [7] by deriving a global criterion, that captures the optimality of the distributional clustering. The main criterion is based on the generalized Jensen-Shannon divergence [24] among multiple probability distributions. Algorithm (1) describes the Divisive Information Theoretic Clustering algorithm in details, as it is used in our approach. Dhillon et al. [10] showed

that their algorithm minimizes *within-cluster divergence* and simultaneously maximizes *between-cluster divergence*. Dhillon et al. [10] have proved that this approach is remarkably better than the agglomerative algorithm of Baker and McCallum [2] and the one introduced by Slonim and Tishby [33]. We cluster the SSVPs to Q clusters in the same manner using the same Divisive Information Theoretic Clustering algorithm (1) stated above.

5.4 Semantically Significant Invariant Visual Words and Phrases (SSIVWs and SSIVPs) Generation

After the distributional clustering, groups of SSVWs that tend to share similar probability distributions are grouped in the same cluster c_k and re-indexed with the same index k. In the same manner,groups of SSVPs that share similar probability distributions are clustered into the same cluster c_q and re-indexed with the same index q.

After re-indexing the SSVWs and the SSVPs, they form the Semantically Significant Invariant Visual Words (SSIVWs) and the Semantically Significant Invariant Visual Phrases (SSIVPs), respectively. Both of the SSIVWs and the SSIVPs form the Semantically Significant Visual Glossaries (SSIVGs).

By generating the SSIVG representation, the visual differences of images from the same class can be *partially* bridged. Consequently, the image distribution in the feature space will become more coherent, regular, and stable.

6 Image Indexing, Classification, and Retrieval Using the SSIVG Representation

Inspired by the success of the vector-space model in the text document representation, it is applied recently to the image representation. Each image is represented by a k-dimensional vector of the estimated weights associated with the visual index terms appearing in the image collections. In this section, we describe how we apply the vector space model to the different layers of the proposed SSIVG representation.

6.1 Vector Space Image Model

The traditional Vector Space Model [30] in Information Retrieval [28] is adapted to our representation, and used for similarity matching and retrieval of images. The following doublet represents each image in the model:

$$I = \begin{cases} \overrightarrow{SSIVW_i} \\ \overrightarrow{SSIVW_i} \end{cases}, \tag{41}$$

where $\overrightarrow{SSIVW_i}$ and $\overrightarrow{SSIVW_i}$ are the vectors for the word and phrase representations of a document respectively:

$$\overrightarrow{SSIVW_i} = (SSIVW_{1,i}, \ldots, SSIVW_{n_{SSIVW},i}),$$

$$\overrightarrow{SSIVP_i} = (SSIVP_{1,i}, \ldots, SSIVP_{n_{SSIVP},i}). \tag{42}$$

Note that the vectors for each level of representation lie in a separate space. In the above vectors, each component represents the weight of the corresponding dimension. We use the ***spatial weight scheme*** introduced by El Sayad et al. [14] for the SSIVWs and the standard ***td*×*idf weighting scheme*** for the SSIVPs. In our approach, we use an inverted file [34] to index images. The inverted index consists of two components: one includes the visual index terms (SSIVW and SSIVP), and the other includes vectors containing the information about the spatial weighting of the SSIVW and the $tf \times idf$ weighting of the SSIVP.

6.2 Similarity Measure

The query image is represented as a doublet of SSIVWs and SSIVPs and we consult the inverted index to find candidate images. All candidate images are ranked according to their similarities to the query image. We have designed a simple measure that allows evaluating the contribution of words and phrases. The similarity measure between a query I_q and a candidate image I_c is estimated as:

$$sim(I_q, I_c) = (1-\alpha)RSV(\overrightarrow{SSIVW_c}, \overrightarrow{SSIVW_q}) + (\alpha)RSV(\overrightarrow{SSIVP_c}, \overrightarrow{SSIVP_q}). \tag{43}$$

The Retrieval Status Value (RSV) of two vectors is estimated with the cosine distance. The non-negative parameter $0 < \alpha < 1$ is to be set according to experiment runs in order to evaluate the contribution between the SSIVWs and the SSIVPs.

6.3 Multiclass Vote-based Classifier (MVBC)

We propose a new multiclass vote-based classification technique (MVBC) based on the SSIVG representation. For each SSIVG_i occurring in an image im_j, we detect the high-level latent topic h_k that maximizes the following conditional probability:

$$p(SSVG_i|h_k) = p(v_l|h_k)p(SSIVG_i|v_l). \tag{44}$$

The final voting score VS_{h_k} for a high-level latent topic h_k in a test image imj_j is:

$$VS_{h_k} = \sum_{a=1}^{N_{h_k}^{SSIVG}} p(SSIVG_a|h_k), \tag{45}$$

Table 1 Values of the different vocabulary sizes and the number of latent topics of different datasets

Dataset	M	W	P	K	L
NUS-WIDE	10,000	3248	551	80	325
MIRFLICKR-25000	3000	1248	480	10	325
Caltech101	2500	1480	409	100	325

where $N_{h_k}^{SSIVG}$ is the number of SSVGs voted for h_k in im_j. Finally, each image is categorized according to the dominant high latent topic which is the topic with the highest voting score (the high latent topic and the class labels are mapped in the training dataset).

7 Experiments

This section reports the large-scale, extensive experimental evaluations in comparison with the state-of-the-art literature to demonstrate the superiority of the proposed methods of the higher-level visual representation and the probabilistic semantic learning in image retrieval, classification, and object recognition.

7.1 Dataset and Experimental Setup

Firstly, we evaluate the proposed SSIVG representation on image retrieval using the NUS-WIDE dataset [6], one of the largest available datasets with 269,648 images and the associated tags from Flickr website. We separate the dataset into two parts. The first part contains 161,789 images to be used for training and the second part contains 107,859 images to be used for testing. It contains 81 image categories or high topics. Secondly, we have tested the proposed MVBC and the SSVIG representation on the MIRFLICKR-25000 [20] dataset for classification. The dataset contains 25,000 images that were retrieved from the Flickr website. We have used the 11 general annotations in the experiments. We use 15,000 images as the training dataset from different image classes and the rest 10,000 images for testing. Thirdly, Caltech101 Dataset [16] is used the proposed SSVIG representation in object recognition. It contains 8707 images, which include objects belonging to 101 classes. For the various experiments, we construct the test dataset by selecting randomly ten images from each object category (resulting in 1010 images) and the rest of the collection are used for training. Table 1 shows the different values of the classical visual word vocabulary or clustering size (M), the SSVW vocabulary size (W), the SSVP vocabulary size (P), the number of the high-level latent topics (K), and the number of the visual latent topics (L) of different datasets, respectively.

7.2 *Assessment of the SSIVG Representation Performance in Image Retrieval*

In this section, we study the performance of the proposed higher-level visual representation in retrieval using NUS-WIDE dataset. We compare the performance of different representations: classical bag of visual words (BOW) [31], the enhanced bag of visual words (E-BOW) that is introduced in Sect. 3.1, SSVW, SSVP, SSIVW, SSIVP, and SSIVG that combine the SSIVW and the SSIVP representations. In addition, we compare the performances of the visual glossaries generated from the pLSA and LDA models rather than the MSSA model, and we reference them here as SSIVG-pLSA and SSIVG-LDA representations, respectively. We also extend the performance comparison to several other recently proposed higher-level representation methods specifically visual phrase pattern [38], descriptive visual glossary [40], and visual synset [42]. For all the representation methods, the traditional Vector Space Model of Information Retrieval is adapted using an inverted file structure and the $tf \times idf$ weighting for all the representations except for the SSIVG representation. We use the proposed spatial weighting scheme and the $tf \times idf$ weighting as described in Sect. 6. In addition, the cosine distance is used for the similarity matching between the query image and the candidate images. The evaluation metric used for the different experiments is the mean average precision (MAP).

7.2.1 Individual Contributions of Different Representation Levels in Image Retrieval

Figure 7 plots the mean average precisions for different representations in image retrieval. It is clear that the E-BOW representation (MAP = 0.193) outperforms the classical BOW representations (MAP = 0.142). It is also obvious that SSIVW representation (MAP = 0.225) is better than the E-BOW representation. The SSVW representation outperforms the BOW representation in the 81 categories except in 5 categories (*glacier*, *fire*, *sport*, *flags*, *sand*). We notice that the average number of the classical visual words in these five categories is too small since the number of the detected interest points is too small. Having a smaller number of visual words leads to a fewer number of SSIVWs that are selected from the visual words, which affects the performances of the SSVW representation. When considering only SSVPs (MAP = 0.232), the performance is slightly better than that of SSVW (MAP = 0.225). An SSVP representation contains both spatial and appearance information, which is assumed to be more informative than that of SSVW in many image categories. However, some query images in categories such as *sky* and *waterfall* do not present consistent spatial characteristics and contain very few or even zero SSVPs. Thus SSVPs do not work well for these cases. The re-indexing of the SSVW and SSVP representations leads to the SSIVW and the SSIVP representation that have better performance (MAP = 0.317 for the SSIVW representation and MAP = 0.321 for the SSIVP representation). The combination of SSIVW and SSIVP into the

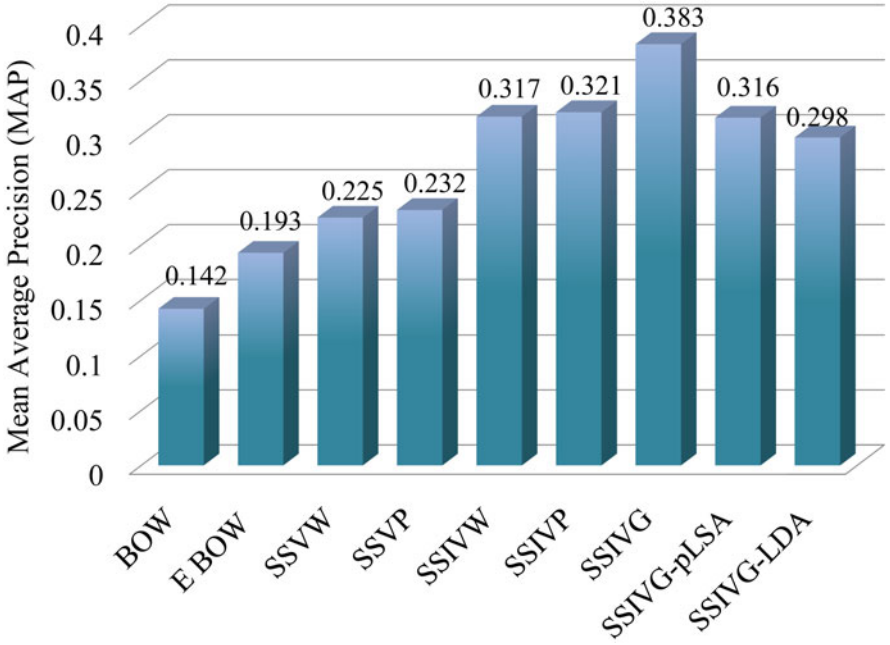


Fig. 7 MAP results for the performances of BOW, E-BOW, SSVW, SSVP, SSIVW, SSIVP, SSIVG, SSIVG-pLSA, and SSIVG-LDA representations in image retrieval

SSIVG representation yields the best results with MAP = 0.383. It also outperforms the SSIVG-pLSA (MAP = 0.316) and SSIVG-LDA (MAP = 0.298) representations especially in the categories that have complicated visual scenes such as *weddings*, *military*, and *coral*.

7.2.2 Comparison of the SSIVG Representation Performance with Other Representation Methods

Figure 8 shows the performance comparison between the SSIVG representation with visual phrase pattern, descriptive visual glossary, and visual synset. SSIVG representation performs better than others and the visual synset has the least performance (MAP = 0.211) compared to others. It is also noted that SSIVG representation outperforms the other representations in most of the 81 classes while the visual phrase pattern representation outperforms SSIVG in only three categories (*dancing, train, computer*) and the descriptive visual glossary representation outperforms SSIVG in only two categories (*fox, harbor*). Having this difference over a data set containing 81 categories and 269, 648 images emphasizes the good performance of the proposed representation.

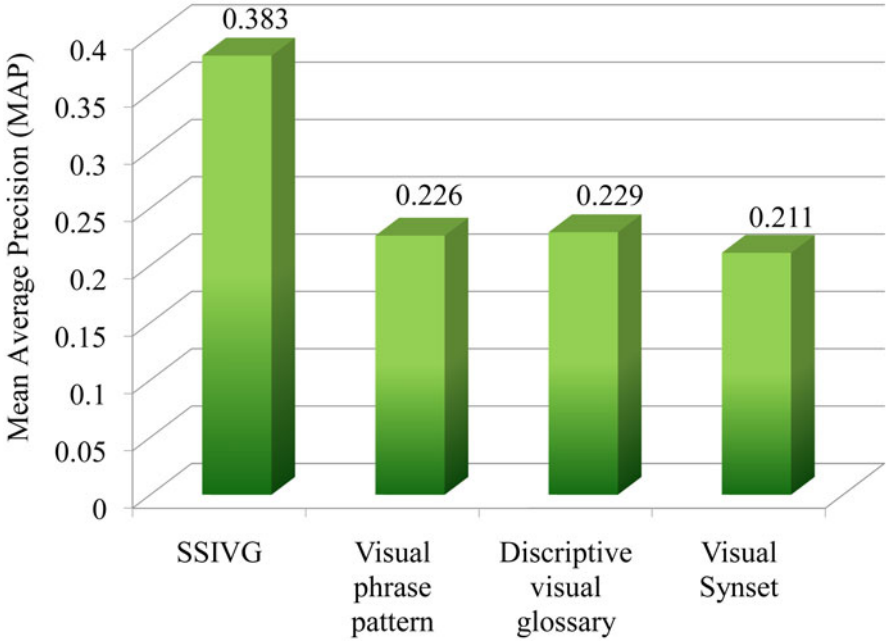


Fig. 8 MAP results for different representations in image retrieval

7.3 Evaluation of the SSIVG Representation and MVBC Performance in Classification

In the following experiments, we study the performance of the SSIVG representation in classification using the vote-based classifier (MVBC). We test the proposed approach (SSIVG+MVBC) performance using MIRFLICKR-25000 data set. We also tested the proposed SSIVG representation using SVM with a linear kernel as a classifier. Again, we compare the classification performance of the SSIVGS+MVBC with those of the other three higher-level visual representations (visual phrase pattern [38], descriptive visual glossary [42], and visual synset [42]) using SVM with a linear kernel as a classifier and $tf \times idf$ as the weighting scheme. Figure 9 plots the average classification precision results for each image class for the different approaches. It is clear that the proposed approach (SSIVG + MVBC) outperforms or performs closely to the SSIVG + SVM approach. SSIVG + MVBC approach also outperforms or performs equally to other approaches. The highest classification performance is obtained in *sky* and *sunset* classes. The different higher-level approaches perform well in these classes except the visual synset representation with SVM which performs worse than the other approaches. It is noted that all the images in both classes contain very specific colors and almost not so much texture. However, this is not always the case, for some *sky* images, there is cloudy sky or just a vague notion of sky somewhere in the images. The least classification performances are in

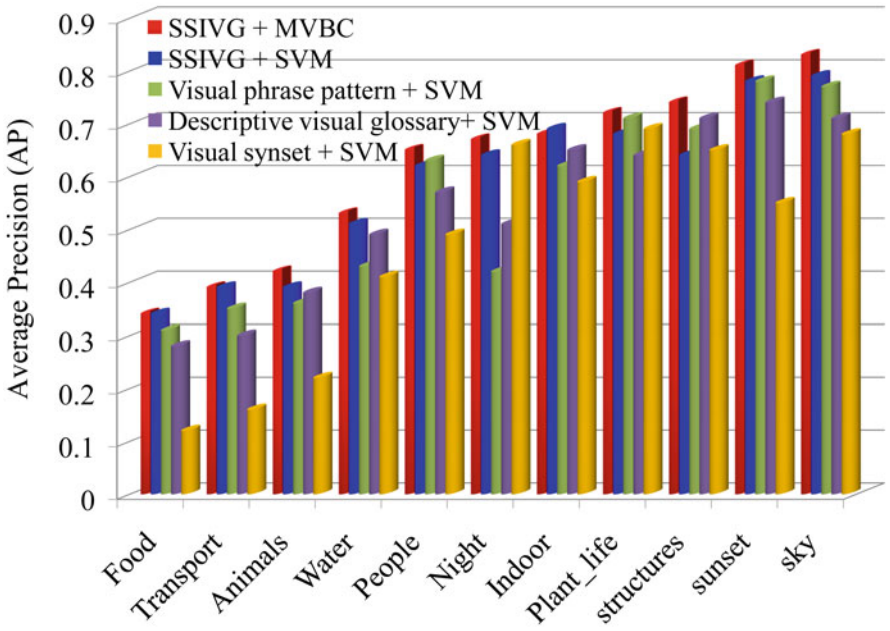


Fig. 9 Classification performance for different approaches

animal, food, and *transport* classes. Note that there is a wide variety of images that can be classified as containing *animal, food,* or *transport*. For example in the *animal* class, not only real animals that are clearly visible, but also hand drawn animals or parts of an animal result into the same class. In addition, in some images, the target object (*animal, food,* or *transport*) does not have to be the subject of the image, but it might also be seen in the background. This makes the classification a challenging problem in these classes.

7.4 Assessment of the SSIVG Representation Performance in Object Recognition

Object recognition has been a popular research topic for many years. Many recently reported efforts show a promising performance in this challenging recognition task. Since the SSIVGs effectively describe certain visual aspects (objects or scenes), it is straightforward that the SSIVGs in each object category should be discriminative for the corresponding object. Consequently, we utilize the object recognition task to illustrate the discriminative ability of SSIVGs. We utilize the Caltech101 dataset for the object recognition task. For each test image, the training image category containing the same object is selected from the image database. In our approach, each test image is recognized by predicting the object class using the SSIVG representation

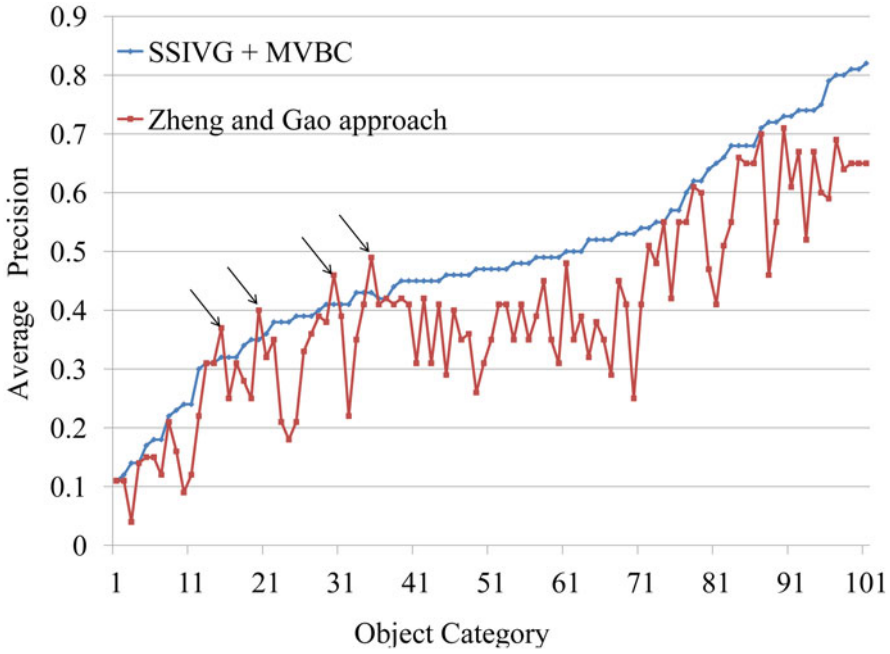


Fig. 10 Object recognition performance for different approaches

and the MVBC. We compare this method with the visual phrase-based approach proposed by Zheng and Gao to retrieve images containing the desired objects. In this approach, each test image is recognized by computing the first 20 retrieved images in the training dataset. Figure 10 shows the average precisions for the two approaches for each object category. We arrange the 101 classes from left to right with respect to the ascending order of average precisions of SSIVG representation in order to get a clearer representation. It is obvious from the results that the proposed approach globally outperforms the other approach, except for four image classes (pyramid, revolver, dolphin, and stegosaurus) out of the 101 classes in the used data set.

Conclusion

In order to retrieve and classify images beyond their visual appearances, we propose a higher-level image representation, semantically significant visual glossary (SSVG). Firstly, we introduce a new multilayer semantic significance model (MSSA) in order to select semantically significant visual words (SSVWs) from the classical visual words according to their probability distributions relating to the relevant visual latent topics in order to overcome the rudeness of the feature quantization process. Secondly, we exploit the spatial co-occurrence information of SSVWs and their

semantic coherency in order to generate a more distinctive visual configuration, i.e., semantically significant visual phrases (SSVPs). Thirdly, we combine the two representation methods to form SSIVG representation. The large-scale, extensive experimental studies have demonstrated the good performance compared with several recent approaches in retrieval, classification, and object recognition. In our future work, we will investigate the usage of such a representation for other applications such as multi-view object class detection and pose estimation.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Record **22**, 207–216 (1993)
2. Baker, L.D., McCallum, A.: Distributional clustering of words for text classification. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 96–103. ACM (1998)
3. Bay, H., Tuytelaars, T., Gool, L.J.V.: Surf: Speeded up robust features. Eur. Conf. Comput Vis. (ECCV) **1**, 404–417 (2006)
4. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter, Y.: Distributional word clusters vs. words for text categorization. J. Mach. Learn. Res. **3**, 1183–1208 (2003)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003). doi:http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993
6. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.T.: Nus-wide: A real-world web image database from national university of singapore. In: ACM International Conference on Image and Video Retrieval (CIVR)
7. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886–893 (2005)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. B. **39**(1), 1–38 (1977)
10. Dhillon, I.S., Mallela, S., Kumar, R.: A divisive information-theoretic feature clustering algorithm for text classification. J. Mach. Learn. Res. **3**, 1265–1287 (2003)
11. El Sayad, I., Martinet, J., Urruty, T., Amir, S., Djeraba, C.: Toward a higher-level visual representation for content-based image retrieval. In: ACM International Conference on Advances in Mobile Computing and Multimedia (ACM MoMM), pp. 213–220 (2010)
12. El Sayad, I., Martinet, J., Urruty, T., Benabbas, Y., Djeraba, C.: A semantically significant visual representation for social image retrieval. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2011). doi:10.1109/ICME.2011.6011867
13. El Sayad, I., Martinet, J., Urruty, T., Dejraba, C.: A semantic higher-level visual representation for object recognition. In: Advances in Multimedia Modeling, Lecture Notes in Computer Science, vol. 6523, pp. 251–261. Springer, Berlin/Heidelberg (2011)
14. El Sayad, I., Martinet, J., Urruty, T., Djeraba, C.: A new spatial weighting scheme for bag-of-visual-words. In: IEEE International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1–6 (2010)
15. El Sayad, I., Martinet, J., Urruty, T., Djeraba, C.: Toward a higher-level visual representation for content-based image retrieval. Multim. Tools Appl. 1–28 (2010)
16. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. Comput. Vis. Image Underst. **106**(1), 59–70 (2007)

17. Gao, S., Tsang, I., Chia, L.T., Zhao, P.: Local features are not lonely—sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3555–3561 (2010). doi: 10.1109/CVPR.2010.5539943
18. Gaussier, E., Goutte, C.: Relation between plsa and nmf and implications. In: The Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 601–602 (2005). doi:http://doi.acm.org/10.1145/1076034.1076148
19. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **42**(1/2), 177–196 (2001)
20. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: ACM International Conference on Multimedia Information Retrieval (ACM MIR). ACM (2008)
21. Kuhn, H.W.: Nonlinear programming: A historical view. SIGMAP Bull. pp. 6–18 (1982). http://doi.acm.org/10.1145/1111278.1111279
22. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. IEEE Conf Comput Vis Pattern Recognit (CVPR). **2**, 2169–2178 (2006)
23. Lienhart, R., Romberg, S., Hörster, E.: Multilayer plsa for multimodal image retrieval. In: ACM International Conference on Image and Video Retrieval (CIVR), p. 9. ACM (2009)
24. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145– (1991)
25. Liu, Y., Zhang, D., Lu, G., Ma, W.: A survey of content-based image retrieval with high-level semantics. Pattern Recognit. **40**(1), 262–282 (2007). doi:10.1016/j.patcog.2006.04.045. http://linkinghub.elsevier.com/retrieve/pii/S0031320306002184
26. Ma, H., Zhu, J., Lyu, M.R.T., King, I.: Bridging the semantic gap between image contents and tags. IEEE Trans. Multim. **12**(5), 462–473 (2010). doi:10.1109/TMM.2010.2051360
27. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. IEEE Conf Comput Vis Pattern Recognit (CVPR). **2**, 2161–2168 (2006)
28. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths (1979)
29. Rissanen, J.: Stochastic Complexity in Statistical Inquiry Theory. World Scientific Publishing Co., Inc. (1989)
30. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM. **18**(11), 613–620 (1975)
31. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision (ICCV), pp. 1470–1477 (2003)
32. Sivic, J., Zisserman, A.: Video data mining using configurations of viewpoint invariant regions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 488–495 (2004)
33. Slonim, N., Tishby, N.: The power of word clusters for text classification. In: In 23rd European Colloquium on Information Retrieval Research (2001)
34. Witten, I.H., Moffat, A., Bell, T.C.: Managing gigabytes: Compressing and Indexing Documents and Images, 2nd edn. Morgan Kaufmann (1999)
35. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 25–32 (2009)
36. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag-of-visual-words representations in scene classification. In: ACM Multimedia Information Retrieval. pp. 197–206. ACM, MIR (2007)
37. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1794–1801 (2009)
38. Yuan, J., Wu, Y., Yang, M.: Discovery of collocation patterns: From visual words to visual phrases. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8. IEEE (2007)
39. Yuan, J., Wu, Y., Yang, M.: Discovery of collocation patterns: From visual words to visual phrases. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)

40. Zhang, S., Tian, Q., Hua, G., Huang, Q., Li, S.: Descriptive visual words and visual phrases for image applications. In: ACM Multimedia, pp. 75–84. ACM, MM (2009)
41. Zheng, Q.F., Gao, W.: Constructing visual phrases for effective and efficient object-based image retrieval. Trans. Multim. Comput. Commun. Appl. **5**(1) (2008)
42. Zheng, Y.T., Zhao, M., Neo, S.Y., Chua, T.S., Tian, Q.: Visual synset: Towards a higher-level visual representation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)

Index

M. Abou-Nasr et al. (eds.), *Real World Data Mining Applications,*
Annals of Information Systems 17, DOI 10.1007/978-3-319-07812-0