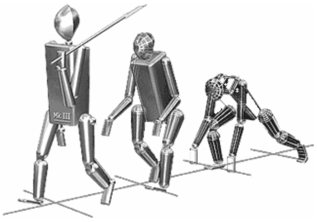
ADM03



Proceedings
Australasian Data Mining Workshop

8th December, 2003, Canberra, Australia

Edited by
Simeon J. Simoff, Graham J. Williams and
Markus Hegland

in conjunction with
The 2003 Congress on
Evolutionary Computation
Canberra – Australia,
8th – 12th December, 2003

University of Technology Sydney
2003

Proceedings of the 2nd Australasian Data Mining Workshop – ADM03, in conjunction with the 2003 Congress on Evolutionary Computation, 8th – 12th December, 2003, Canberra, Australia.

S. J. Simoff, G. J. Williams and M. Hegland (eds).

Workshop Web Site:
http://datamining.csiro.au/adm03/

Published by the University of Technology Sydney

ISBN 0-9751724-1-7

Foreword

The Australasian Data Mining Workshop is a flagship event in the area of discovering meaningful insights in large data sets. The art and science of analytics and data mining have always attracted researchers and industry practitioners in the region. Data mining projects involve both the utilisation of established algorithms from machine learning, statistics, and database systems, and the development of new methods and algorithms, targeted at large data mining problems. Nowadays data mining efforts have gone beyond crunching databases of credit card usage or retail transaction records. The progress in computing technology affects all aspects of human existence. The data mining technologies are becoming the core part of the so-called “embedded intelligence” in business, health care, drug design, biology, design and other areas of human endeavour.

The first edition of the Australasian Data Mining Workshop was a successful event, conducted in conjunction with the 15th Australian Joint Conference on Artificial Intelligence, 2nd - 6th December 2002, Canberra, Australia. The workshop attracted a number of participants from Australian industry, academia, research institutions and centers, in particular, researchers from the ANU Data Mining Group, CSIRO Enterprise Data Mining, and UTS Smart e-Business Systems Lab. The workshop facilitated the links between different research groups in Australia and industry, evidenced by the initiative in the creation of an Australian Research Council Research Network on Improving Australia's Data Mining and Knowledge Discovery Research, and the Institute of Analytic Professionals of Australia. It also strengthened the interconnections between researchers in academic and research organisations, and industry practitioners, who utilise data mining techniques in various business case studies.

This year the workshop builds on these trends. The workshop is expected to broaden and strengthen the links within the analytics community, offering a forum for presenting and discussing latest research and practical experience in the area. The workshop follows a rigid blind peer-review process and ranking-based paper selection process. All papers were extensively reviewed by two to three referees drawn from the program committee. Papers that present comprehensive, completed (or near completion) research work have been allocated larger presentation time slots. The works that present research work in progress or “green house” ideas have been allocated shorter time slots, with more time left for discussion. The organisers have reserved a special presentation session for an overview of on-going initiatives.

Once again, we would like to thank all those, who supported this year’s efforts on all stages – from the development and submission of the workshop proposal to the preparation of the final program and proceedings. We would like to thank all those who submitted their work to the workshop. Special thanks go to the program committee members and other reviewers, for the final quality of selected papers depends on their efforts.

Simeon, J. Simoff, Graham J. Williams and Markus Hegland

November 2003

Workshop Chairs

Simeon J Simoff	University of Technology, Sydney
Graham J Williams	CSIRO Canberra
Markus Hegland	Australian National University

Program Committee

Mihael Ankerst	Boeing Corp., USA
Michael Böhlen	University of Bolzano, Italy
Doug Campbell	SAS Australian and New Zealand
Jie Chen	CSIRO, Canberra, Australia
Peter Christen	Australian National University, Australia
Vladimir Estivill-Castro	Giffith University, Australia
Warwick Graco	Australian Taxation Office
Weiqiang Lin	Australian Taxation Office
Warren Jin	CSIRO, Canberra, Australia
Paul Kennedy	University of Technology, Sydney, Australia
Inna Kolyshkina	Pricewaterhouse Coopers Actuarial, Sydney, Austrlia
Tom Osborn	NUIX Pty Ltd, Australia
Francois Poilet	Parc Universitaire de Laval-Change, France
Chris Rainsford	DSTO Canberra, Australia
David Skillicorn	University of Queens, Canada
Marcel van Rooyen	Hutchison Communications, Australia
John Yearwood	University of Ballarat, Australia
Osmar Zaiane	University of Alberta, Canada

Program for ADM03 Workshop

Monday, 8 December, 2003, Canberra, Australia

9:00 - 9:15 Opening and Welcome

9:15 - 10:30 Session 1

- 09:15 - 09:40 STOCHASTIC PRELIMINARY INVESTIGATIONS INTO STATISTICALLY VALID EXPLORATORY RULE DISCOVERY
  Geoffrey I. Webb
- 09:40 - 10:05 TWO PHASE CLASSIFICATION BY EMERGING PATTERNS
  Ming Fan, Weimei Zhi, Hongjian Fan and Yigui Sun
- 10:05 - 10:30 FEATURE PREPARATION IN TEXT CATEGORIZATION
  Ciya Liao, Shamim Alpha and Paul Dixon

10:30 - 11:00 Coffee break

11:00 - 12:30 Session 2

- 11:00 - 11:25 LEARNING QUANTITATIVE GENE INTERACTIONS FROM MICROARRAY DATA
  Michael Bain and Bruno Gaëta
- 11:25 - 11:50 AN MINING APPROACH USING MOUCLAS PATTERNS
  Yalei Hao, Markus Stumptner and Gerald Quirchmayr
- 11:50 - 12:15 FROM RULE VISUALISATION TO GUIDED KNOWLEDGE DISCOVERY
  Aaron Ceglar, John F. Roddick, Carl H. Mooney and Paul Calder
- 12:15 - 12:30 TEXTURE ANALYSIS VIA DATA MINING
  Martin Heckel

12:30 - 13:30 Lunch

13:30 - 14:50 Session 3

- 13:30 - 13:45 CLUSTERING TIME SERIES FROM MIXTURE POLYNOMIAL MODELS WITH DISCRETISED DATA
  A. J. Bagnall, G. Janacek, B. de la Iglesia and M. Zhang
- 13:45 - 14:00 RELATIVE TEMPORAL ASSOCIATION RULE MINING
  Edi Winarko and John F. Roddick
- 14:00 - 14:25 A STUDY OF DRUG-REACTION RELATIONS IN AUSTRALIAN DRUG SAFETY DATA
  M. A. Mamedov, G. W. Saunders and E. Dekker
- 14:25 - 14:50 MINING GEOGRAPHICAL DATA WITH SPARSE GRIDS
  Markus Hegland and Shawn W. Laffan

14:50 - 15:20 Coffee break

15:20 - 16:40 Session 4

- 15:20 - 15:45 CONGO: Clustering on the Gene Ontology
  Paul J. Kennedy and Simeon J. Simoff
- 15:45 - 16:00 ADAPTIVE MINING TECHNIQUES FOR DATA STREAMS USING ALGORITHM OUTPUT GRANULARITY
  Mohamed Medhat Gaber, Shonali Krishnaswamy and Arkady Zaslavsky
- 16:00 - 16:15 An Analytical Approach for Handling Association Rule Mining Results
  Rodrigo Salvador Monteiro, Geraldo Zimbrão and Jano Moreira de Souza
- 16:15 - 16:40 Modelling Insurance Risk: A Comparison of Data Mining and Logistic Regression Approaches
  Inna Kolyshkina, Peter Petocz and Ingrid Rylander

16:40 - 17:00 Closing Session

- 16:40 - 16:50 AUSTRALIAN RESEARCH COUNCIL RESEARCH NETWORK ON IMPROVING AUSTRALIA'S DATA MINING AND KNOWLEDGE DISCOVERY RESEARCH
  John F. Roddick
- 16:50 - 17:00 INSTITUTE OF ANALYTIC PROFESSIONALS OF AUSTRALIA
  Inna Kolyshkina

Table of Contents

Preliminary Investigations into Statistically Valid Exploratory Rule Discovery

Geoffrey I. Webb

School of Computer Science and Software Engineering, Monash University,
Melbourne, Vic 3800, Australia
webb@infotech.monash.edu

Abstract. Exploratory rule discovery, as exemplified by association rule discovery, is has proven very popular. In this paper I investigate issues surrounding the statistical validity of rules found using this approach and methods that might be employed to deliver statistically sound exploratory rule discovery.

1 Introduction

Association rule discovery has proven very popular. However, it is plagued by the problem that it often delivers unmanageably large numbers of rules. As the current work reveals, not only are the rules numerous, but in at least some cases the vast majority are spurious or unproductive specialisations of more general rules. This paper discusses the issues of spurious and unproductive rules and presents preliminary approaches to address them. Experimental results confirm the practical realisation of the concerns and suggests that the preliminary techniques presented are effective.

2 Exploratory rule discovery

I use the term *exploratory rule discovery* to encompass data mining techniques that seek multiple rather than single models, with the objective of allowing the end-user to select between those models. It is distinguished from *predictive data mining* that seeks a single model that can be used for making predictions.

Exploratory data mining is often applicable when there are factors that can affect the usefulness of a model but it is difficult to quantify those factors in a manner that may be used by an automated data mining system. By delivering multiple alternative models to the end-user they are empowered to evaluate the available models and to select those that best suit their business or other objectives.

Three prominent frameworks for exploratory rule discovery on which I here focus are *association rule discovery* [1], *k-most-interesting rule discovery* [2] and *contrast* or *emerging pattern discovery* [3, 4] as it is variously known. These techniques all discover qualitative rules, rules that represent relationships between nominal-valued variables.

Each such rule $A \rightarrow C$ represents the presence of a (potentially) interesting relationship between the antecedent A and the consequent C, where A is a conjunction of nominal-valued terms and C is a single nominal valued term[1]. The rules are usually presented together with statistics that describe the relationship between A and C.

2.1 Association Rule Discovery

Association rule discovery [1] is the most widely deployed exploratory rule discovery approach. It grew out of market-basket analysis, the analysis of transaction data for combinations of products that are purchased in a single transaction. Association rule discovery uses the so called *support-confidence framework*. It finds all rules that satisfy a user-specified minimum support constraint together with whatever further constraints the user may specify. Essentially, the approach generates all rules that satisfy the minimum support constraint but discards at the final stage any rules that fail the further constraints.

Support is the proportion of records in the training data that satisfy both the antecedent and consequent of the rule.

Initial approaches used a further constraint on minimum *confidence*. To avoid potential confusion with the statistical concept of *confidence* I will hereafter refer to this metric as *strength*.

$$strength = support/coverage \tag{1}$$

where *coverage* is the proportion of records that satisfy the antecedent.

More recent approaches typically use a constraint on minimum lift in preference to a constraint on strength:

$$lift = strength/prior \tag{2}$$

where *prior* is the proportion of records that satisfy the consequent.

The main mechanism available to control the number of rules that are discovered is the value that is specified for minimum support. However, it is usually difficult to anticipate which values of minimum support will result in manageable numbers of rules. Too large a value will result in no rules. Too small a value will result in literally millions of rules. In practice there may be a very narrow range of values of support below which there are extremely few rules discovered and above which there are too many rules discovered [5].

2.2 *K*-Most-Interesting Rule Discovery

K-most-interesting rule discovery [2] addresses this problem by empowering the user to specify both a metric of interestingness and a constraint on the maximum

[1] While association rules are often described in terms of allowing C to be an arbitrary conjunction of terms, in many implementations C is restricted to a single term. In the current work I follow this practice as it greatly reduces the complexity of the rule discovery task while satisfying many rule discovery needs.

number of rules to by discovered. In place of a minimum support constraint, k-most-interesting rule discovery uses these two pieces of information to prune the search space. They return the k rules that optimise the interestingness metric within whatever other constraints the user might specify. It is left up to the user to specify the interestingness metric, the only constraint on the metric being that it must define a partial order on a set of rules given a set of data by which the interestingness of those rules is to be scored.

2.3 Contrast Discovery

Contrast discovery [4] (initially developed under the name *emerging pattern discovery* [3]) seeks rules that identify conditions whose frequency differs between groups of interest. It has been shown that this is equivalent to a form of rule discovery restricted to a consequent that signifies group membership [6].

3 Spurious Rules

A problem for all three of these forms of exploratory rule discovery is that they suffer a high risk of discovering *spurious rules*. These are rules that appear interesting on the sample data but which would not be interesting if the true population probabilities were used to assess their level interestingness in place of the observed sample frequencies.

For example, suppose that there is a rule with coverage of one record and a lift of 2.0. This provides very little evidence that the lift that would be obtained by substituting population probabilities for sample frequencies would have high lift as a rule with one record coverage must have either a support of zero or one record and hence, irrespective of the population lift, the observed lift must either be 0.0 or $1.0/prior$. In other words, when the rule coverage is low, the statistical confidence will be low that the observed relative frequencies are strongly indicative of the population probabilities.

The support-confidence framework of association rule discovery attempts to counter this problem by enforcing a minimum support constraint in the expectation that considering only rules with high support will lead to the observed frequencies being strongly representative of the population frequencies.

4 The Multiple Comparisons Problem

However, this ignores the problem of multiple comparisons [7]. If many observations are made then one can have high confidence that some events that are unlikely in the context of a single observation are likely to occur in some of the many observations that are made. For example, suppose a hypothesis test is applied to evaluate whether a rule is spurious with a significance level of 0.05. Consider a spurious rule $A \rightarrow C$ for which A and C are independent. The probability that this spurious rule will be accepted as not spurious is 5%. If this process were applied to 1,000,000 rules in a context where all rules were spurious (for example, the data were generated stochastically using uniform probabilities) we could reasonably expect that 5% or 50,000 would be accepted as non-spurious despite all being spurious. In practice the rule spaces explored by rule discovery systems are many magnitudes greater than 1,000,000, and hence we should expect many spurious rules to be generated even if we apply a significance test before accepting each one.

5 Filters for Spurious Rules

One response to this problem is to apply a correction for multiple comparisons, such as the Bonferroni adjustment that divides the critical value α by the number of rules evaluated. This is the approach adopted in the contrast discovery context by STUCCO [4]. A problem with this approach is that the search process may require the evaluation of very large numbers of rules and hence α may be driven to extremely low values. The lower the value of α the higher the probability of type-2 error, that is, of rejecting rules that are not spurious.

What is required is an approach that minimises the risk of type-1 error, that is, of accepting spurious rules, without in the process discarding the most interesting non-spurious rules.

6 Unproductive Rules

A further problem for rule discovery is that of unproductive rules. A rule $A \rightarrow C$ is unproductive if it has a generalisation $B \rightarrow C$ such that $strength(A \rightarrow C) \leq strength(B \rightarrow C)$. An unproductive rule will arise when a variable that is unrelated to either B or C is added to B. As the strength is unaltered, the lift of the unproductive rule will equal that of the generalisation. In practice data sets often involve many variables that do not impact upon the rules of interest and hence very large numbers of unproductive rules are generated.

The problem of unproductive rules interacts with the problem of spurious rules. It is straightforward to add a filter to the rule discovery process that discards any rule for which the observed strength is not greater than the observed strength of all its generalisations. However, random variations in the data sample will lead to almost half the unproductive rules appearing to be productive (albeit in many cases only very slightly). A statistical test of significance may be applied, as supported by the Magnum Opus rule discovery system [8], but we again face the multiple comparisons problem.

7 A New Approach

Hypothesis testing is designed for controlling the risk of type-1 error in the context of evaluating a prior hypothesis against previously unsighted data. It is inadequate to the task of both generating hypotheses and evaluating them from the same set of data.

An approach that has been used in other data mining contexts is to use a holdout set for hypothesis testing. Models are inferred from a training set and then evaluated against a holdout set. My proposal is to utilise this framework in an exploratory rule discovery context. The available data will be divided into an exploratory data set from which rules will be discovered. This will be treated as a hypothesis generation process. The rules discovered are treated as hypotheses that are then evaluated against the holdout set. As the holdout data is partitioned from the exploratory data, the huge number of rules considered during rule discovery does not affect the subsequent evaluation. A simple multiple comparisons adjustment need only divide the selected alpha value by the number of rules delivered by the rule discovery phase. Thus the α value need not be set prohibitively low, minimising the problem of type-2 error.

Note that re-sampling methods, such as cross-validation, that serve to evaluate the power of a method for a given set of data are not adequate to this task. Unlike the case where we wish to predict the likely predictive accuracy of a single model produced by a system, here we wish to produce many rules. We want to control the risk of any of these rules being spurious.

I propose the use of k-most-interesting rule discovery for the rule discovery phase rather than association rule discovery, because it is desirable to find a constrained number of rules during the rule discovery phase. If too many rules are discovered the necessary multiple comparisons adjustment will result in a raised risk of type-2 error. If too few rules are discovered then there is a raised risk of failing to discover sufficient interesting rules to satisfy the user. Standard association rule discovery provides only very imprecise control over the number of rules discovered. Tightening or weakening each of the constraints will respectively decrease or increase the number of rules discovered, but typically it is not possible to predict by exactly how much a particular alteration to the constraints will affect the number of rules discovered. In contrast, k-most interesting rule discovery always returns k rules, except in the unusual circumstance that the other constraints applied are satisfied by fewer than k rules.

7.1 Selection of Holdout Data

The proposed generic holdout technique is applicable to two different contexts. In the first context there is a single set of data available, and this data needs to be partitioned. In this context, it would be appropriate for the data to be randomly partitioned, a process that can be readily automated. It is probably desirable that the partitions be of similar sizes. It is important to have as much data as possible for exploratory rule discovery, so as to generate as powerful hypotheses as possible. It is also important to have as much holdout data as possible so as to maximise the power of the statistical tests that are applied.

The second context is one in which there are natural partitions of the data. For example, data may be obtained over time. In such a context it might be valuable to utilise the natural partitions so as to evaluate whether the regularities apparent in the exploratory partition (such as the data from one year) generalise across partitions (such as to the next year).

7.2 Holdout Evaluation Tests

For each rule $A \rightarrow C$ we wish to assess whether the observed $strength(A \rightarrow C)$ is significantly higher than would be expected if there were no relationship between the antecedent and consequent and also whether it is significantly higher than the strength of all its generalisations[2]. I use a binomial test to assess whether an observed strength is signficantly higher than a comparator strength. The large number of subsets of an antecedent containing many conditions would make testing against all generalisations infeasible. In consequence I test $strength(A \rightarrow C)$ against the sample frequency of C (which equals $strength(\emptyset \rightarrow C)$) and against the strength of all its immediate generalisations (rules formed by removing a single condition from A). While it is theoretically possible for a rule to have higher strength than all of its immediate generalisations but lower strength than a further generalisation, to do so requires a very specific type of interaction between four or more variables of a form that might make the resulting rule interesting in its own right despite being unproductive with respect to one of its generalisations.

8 Evaluation

The Magnum Opus [8] k-most interesting rule discovery system was extended to support the form of holdout evaluation described above.

I first sought to evaluate what proportion of rules discovered by a traditional association rule approach to rule discovery might be either spurious or unproductive. To this end I investigated rule discovery performance on two large data sets from the UCI repository [9], covtype (581012 records, 10 numeric fields, 41 categorical fields) and census-income (199,523 records, 7 numeric fields, 34 categorical fields). Numeric fields are discretised into three bins, each containing equal numbers of records.

Each data set was randomly divided into two equal sized subsets, the exploratory data used to discover rules and the holdout data used for holdout evaluation.

I started by seeking to find values of minimum support and minimum lift that resulted in constrained numbers of rules (less than 10,000). After a number of trials I found for the covtype data that a minimum support of 0.25 and minimum lift of 2.75 resulted in 1997 rules of which 1936 (96.9%) were rejected by holdout evaluation. For the census-income data minimum support of 0.4 and minimum lift of 2.0 resulted in 7502 rules of which 7462 (99.4%) were rejected as spurious or unproductive when assessed against the holdout data. These figures provide a

[2] Actually, as $\emptyset \rightarrow C$ is a generalisation of $A \rightarrow C$, the latter condition subsumes the first.

dramatic illustration of the degree to which traditional association rule discovery results may be dominated by rules that are effectively noise.

To separate the issues of unproductiveness from spuriousness, I applied a filter to discard unproductive rules during the rule discovery phase. That is, during rule discovery a rule was discarded if it was unproductive as assessed using the observed strength on the exploratory data without application of a significance test.

With the same support and lift constraints, for covtype 433 rules were found of which 377 (87.1%) were rejected by holdout evaluation. Whereas only 40 rules passed the holdout evaluation when the support confidence framework was employed, when filtering of unproductive rules is added this is raised to 63 rules, as the number of multiple comparisons is reduced and hence the adjusted α value used in holdout evaluation is raised.

When a binomial test was applied during rule discovery to evaluate whether a rule was significantly productive (on the exploratory data), using $\alpha = 0.05$, the number of rules found was further reduced to 73 of which 45 (61.6%) were rejected by holdout evaluation. Note that the number of rules that have passed the holdout test (18) has decreased. This illustrates the problem of filtering so as to adequately balance the risks of type-1 and type-2 error. The filter applied during rule discovery has discarded 45 rules that were found with a weaker filter and then accepted after holdout evaluation.

Applying yet stronger filters, for example by adjusting for multiple comparisons the α used in the statistical test applied during rule discovery, can be expected to improve the proportion of rules that pass holdout evaluation, but to decrease the absolute number of rules that pass.

For census-income when unproductive rules were discarded during the rule discovery phase, 48 rules were discovered of which 8 were rejected by holdout evaluation. This resulted in the same 40 rules passing holdout evaluation as when unproductive rules were not discarded during the rule discovery phase. Tightening the filter applied during rule discovery by adding a significance test resulted in the discovery of 45 rules of which 5 were discarded by holdout evaluation, leaving the same 40 rules.

As a final test, I applied k-most-interesting rule discovery in place of the support-confidence framework. As a measure of interestingness I used *leverage*,

$$leverage(A \rightarrow C) = support/coverage(A) \times coverage(C) \ . \tag{3}$$

This represents the difference between the observed joint frequency and the joint frequency that would be expected if the antecedent and consequent were independent. I sought the 100 rules that maximised this value without any other constraints other than that all rules had to be significantly productive at the 0.05 level, that is that they had to pass a binomial test at the 0.05 level indicating that they had higher strength than any immediate generalisation. Note that this process did not require the time consuming and error prone business of identifying a suitable minimum support constraint.

For covtype all 100 rules passed the holdout evaluation. All rules found had extremely high support, the lowest being 0.436. The lowest lift was 2.19. It is interesting that the search for the 100 most interesting rules found quite a different trade-off between support and lift than I found during my manual attempt to find a set of constraints that provided sufficiently few rules for consideration, resulting in rules with higher support but lower lift. It is also notable that all rules so found passed holdout evaluation, as the search explicitly sought rules that were most exceptional on the exploratory data and hence the most valuable to evaluate on the holdout data.

For census-income, of the 100 rules found 13 were discarded by holdout evaluation. All rules found had high support, the lowest being 0.413. The lowest lift of a rule was 1.91. This illustrates the difficulty of finding appropriate constraints to apply within the traditional association rule framework, as it lay just outside the minimum lift that I had found after some experimentation in the attempt to return only a constrained number of rules.

For each data set, the k-most-interesting approach to rule discovery delivered higher numbers of statistically sound rules without need for manual determination of appropriate support and other constraints.

9 Conclusion

I have presented an approach to addressing the problems of spurious and unproductive rules in exploratory rule discovery. Two examples have demonstrated that over 99% of rules discovered using the support-lift framework can be spurious or unproductive. I have shown that the use of k-most-interesting rule discovery with holdout evaluation can overcome this problem, delivering for the first time statistically sound exploratory discovery of potentially interesting rules from data.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining associations between sets of items in massive databases. In: Proceedings of the 1993 ACM-SIGMOD International Conference on Management of Data, Washington, DC (1993) 207–216
2. Webb, G.I.: Efficient search for association rules. In: The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, The Association for Computing Machinery (2000) 99–107
3. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: ACM SIGKDD 1999 International Conference on Knowledge Discovery and Data Mining, ACM (1999) 15–18
4. Bay, S.D., Pazzani, M.J.: Detecting group differences: Mining contrast sets. Data Mining and Knowledge Discovery **5** (2001) 213–246
5. Zheng, Z., Kohavi, R., Mason, L.: Real world performance of association rule algorithms. In: KDD-2001: Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining, New York, NY, ACM (2001) 401–406
6. Webb, G.I., Butler, S., Douglas, N.: On detecting differences between groups. In: Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03). (2003) 256–265

7. Jensen, D.D., Cohen, P.R.: Multiple comparisons in induction algorithms. Machine Learning **38** (2000) 309–338
8. Webb, G.I.: Magnum Opus version 1.3. Computer software, Distributed by Rulequest Research, http://www.rulequest.com (2001)
9. Blake, C., Merz, C.J.: UCI repository of machine learning databases. [Machine-readable data repository]. University of California, Department of Information and Computer Science, Irvine, CA. (2001)

Two Phase Classification By Emerging Patterns[*]

Ming Fan[1], Weimei Zhi[1], Hongjian Fan[2], Yigui Sun[1],

[1] Department of Computer Science, Zhengzhou University
Zhengzhou Henan 450052, P.R.China
{mfan,iewmzhi}@zzu.edu.cn
[2] Department of CSSE, The University of Melbourne
Parkville, Vic 3052, Australia
hfan@cs.mu.oz.au

Abstract. Emerging Patterns (EPs) are itemsets whose supports change significantly from one data class to another. It has been shown that they are useful for constructing accurate classifiers. Existing EP-based classifiers try to use high support-ratio EPs, which may leads to poor generalization capability when applied to unseen instances. PNrule is a new two-phase framework for learning classifier models in data mining. The first phase detects the presence of the target class, while the second detects the absence of the target class. This work proposes a novel classification method, called Two-Phase Classification by Emerging Patterns (TPCEP), to combine the idea of two-phase induction and classification by emerging patterns. Our experiment study carried on benchmark datasets from the UCI Machine Learning Repository shows that TPCEP performs comparably with other state-of-the-art classification methods such as CBA, CMAR, C5.0, NB, and CAEP in terms of overall predictive accuracy.

Keyword： Data mining, classification, emerging pattern, two-phase classification

1 Introduction

Classification is an important data mining problem, and has also been studied substantially in statistics, machine learning, neural networks and expert systems over decades. In general, given a training dataset, the task of classification is to build a concise mode from the training dataset such that it can be used to predict class labels of unknown objects. Classification is also known as *supervised learning* as the learning of the model is "supervised" in that it is told which class each training example belongs to.

Classification has a wide range of applications in business, finance, DNA analysis, telecommunication, science research and so on. There are many classification models proposed by researchers in machine learning, expert systems, statistics, and neural networks. Most of these algorithms are memory-based, typically assuming a small

[*] This joint work is supported in part by the Nature Science Foundation of Henan Province of China under Grant No. 0211050100.

datasets. With the growth of data in volume and dimensionality, it has become a challenge to build efficient classifiers for large datasets. Recent data mining research focuses on developing scalable classification techniques capable of handing large disk-resident data [9].

1.1 Background

The traditional rule-based classifier models are popular in the domain of data mining, because humans can easily interpret the rules and the accuracy of the resulting classifiers is competitive to other state-of-the-arts. A general rule-based model includes a disjunction (union) of rules, where each rule is a conjunction of conditions imposed on different attributes. The goal of learning rule-based models directly from the training data is to discover a small number of rules to cover most of the positive examples of the target class (high coverage or recall) and very few of the negative examples (high accuracy or precision) [1].

Existing methods of general-to-specific learning techniques, such as C4.5 and Ripper, usually follow a sequential covering strategy. Their aim is to build a DNF (disjunct normal form) model. Initially, the model contains only the most general rule, an empty rule. Specific conditions are added to it progressively. In each iteration, a conjunctive rule that can predict the target class with a high accuracy is discovered. Then the instances covered by this rule are removed. Only the remaining instances will be used in the next iteration. The sequential covering technique works fine but may fail in the following two possible scenarios. The first one is when the target class signature is composed of the two components, presence of the target class and absence of the non-target-class, and the later component is not correctly or completely learned. It is referred to as the problem of splintered false positives. The second one is the problem of small disjuncts [11], in which rules that cover a small number of target class examples are more prone to generalization error than rules covering larger number of such examples.

PN-rule is a new two stage general-to-specific framework for learning classifier models in data mining [2]. It is based on both rules that predict presence of the target class (P-rules) and rules that predict absence of the target class (N-rules). In the first stage, a set of P-rules is learned. They together cover most of the positive training instances and each rule covers enough number of instances to maintain its statistical significance. The set of P-rules will also cover some negative training instances (called false positives) because of the relaxation of accuracy, e.g., accuracy is compromised in favour of support. In the second stage, the whole dataset is reduced to the union of all true positives and those false positives. On the reduced dataset, N-rules are learned to remove the false positives. The two-phase technique makes the resulting classifier less sensitive to the problem of small disjuncts. A case study on a real-life network intrusion-detection dataset shows that the two-phase method achieves comparable results with other state-of-the-art classification methods such as C4.5 and Ripper and it performs significantly better for rare classes.

1.2 Motivation

Emerging Pattern (EP) is a new kind of knowledge patterns [5], which represents the knowledge of sharp differences between data classes. Emerging Patterns are basically conjunctions of simple conditions imposed on different attributes. EPs are defined as multivariate features (i.e., itemsets) whose supports (or frequencies) change significantly from one class to another. The concept of EPs is very suitable for serving as a classification model. By aggregating the differentiating power of EPs, the constructed classification systems are usually more accurate than other existing state-of-the-art classifiers. EPs-based classifiers are effective for large datasets with high dimensionality because the learning phrase uses efficient algorithms such as border-based algorithms and tree-based algorithms to discover EPs. EPs-based classifiers differ from the rule-based classifiers in that they aggregate the power of many EPs, i.e., they consider many combinations of attributes for classification of a test, whereas the rule-based classifiers usually use only one rule for one test, i.e., they consider one group of attributes.

The number of EPs present in large datasets may be exponential in the worst case. It has been recognized that only a small fraction of the large number of EPs are very useful for classification purpose. Recently, a special kind of EPs, called essential emerging patterns (eEP), is suggested to be excellent candidates for building accurate classifiers [7]. Essential emerging patterns are the most general hypotheses that fit the training examples, that is, they are the most minimal itemsets satisfying the conditions. Any proper subset of an eEP does not satisfy the conditions. Super sets of eEPs are not regarded as essential because Ockham's razor states that the simplest hypothesis consistent with the data is preferred. The set of eEPs is not only high quality patterns for classification, but also orders of magnitude smaller than that of all EPs.

1.3 Our Work

In the paper we propose a novel classification method, called Two-Phase Classification by Emerging Patterns (TPCEP), which takes advantage of two-phase technique and the aggregation strength of EP-based classifiers.

TPCEP distinguishes itself from other EPs-based classifiers by two-phase induction of EPs and a new scoring mechanism. Existing EP-based classifiers such as JEP-C [12] usually use EPs with large growth rate because those EPs have very sharp discriminating power. However, high growth-rate EPs tend to have low supports and even they together cannot cover enough of the training data. This may leads to poor generalization capability when applied to unseen instances, i.e., the classifier cannot find any EP to classify some tests and it has to "guess" using the majority class, which is very unreliable. By two-phase induction of EPs, TPCEP has better generalization ability. In the first phase, it finds the set of EPs (called P-eEPs) that have high supports and high coverage on the training data. Initially large growth-rate EPs are selected, but later high support EPs are preferred to satisfy the coverage requirement. So we relax the strict requirement of large growth-rate EPs. The use of moderate growth-rate EPs makes the set of P-eEPs also cover some negative training instances (called false positives). The second phase will then try to mine another set of EPs (N-eEPs), which can remove false positives in the collection of the instances covered by the first phase EPs. Here we correct the errors due to the use of EPs whose discriminating power are not so sharp.

Our experiment study carried on 10 benchmark datasets from the UCI Machine Learning Repository shows that TPCEP performs comparably with other state-of-the-art classification methods such as CBA, CMAR, C5.0, NB and CAEP in terms of overall predictive accuracy,

Organization: An outline of the remainder of this paper is as follows. Section 2 defines the basic conceptions. Section 3 details our TPCEP to use eEPs for classification. Section 4 presents an extensive experimental evaluation of TPCEP on popular benchmark datasets from the UCI Machine Learning Repository and compares its performance with CBA, CMAR, C5.0, NB, CAEP and BCEP. Finally, in section 5 we provide a summary and discuss future research issues.

2. Preliminary

Suppose a dataset consists of a number of data objects (instances, or examples) of the form (a_1, a_2, …, a_n) following the schema (A_1, A_2,…, A_n), where A_1, A_2,…, A_n are called attributes. Attributes can be categorical or quantitative. Quantitative attributes are discretized by dividing the range of the attribute into intervals and the real data values are replaced by interval labels. Each data object in the dataset is also labelled by a class label $C \in \{C_1, C_2,\ldots, C_k\}$ to indicate which class the data object belongs to.

An item is a pair of the form (attribute-name, attribute-value). Let *I* be the set of all items appearing in the raw dataset. A set *X* of items is also called an itemset, which is defined as a subset of *I*. Each object in the raw dataset can be represented by an itemset. In the association rule context, such an itemset is called a transaction. Emerging patterns are defined on the discretized transaction database. We say any instance *S* contains an itemset *X*, if $X \subseteq S$.

Definition 1: The support of an itemset *X* in a dataset *D* of datasets, $sup_D(X)$, is $count_D(X)/|D|$, where $count_D\,(X)$ is the number of instances in *D* containing *X,* and |*D*| is the total number of instances in *D*.

Definition 2: Given two different datasets *D'* and *D,* the growth rate of an itemset *X* from *D'* to *D* is defined as

$$GR_{D'\to D}(X) = \begin{cases} 0 & \textit{if } sup_{D'}(X) = sup_D(X) = 0 \\ \infty & \textit{if } sup_{D'}(X) = 0, sup_D(X) \neq 0 \\ sup_D(X)/sup_{D'}(X) & \textit{otherwise} \end{cases}$$

Emerging patterns are itemsets whose supports change significantly from one data class to another.

Definition 3: Given a growth rate threshold $\rho > 1$, an itemset X is said to be an ρ-emerging pattern (ρ-EP or simply EP) from D' to D if $GR_{D' \to D}(X) \geq \rho$.

When D' and D are clear from context, an EP X from D' to D is simply called an EP of D. The support of X in D, denoted as $sup(X)$, is called the support of the EP X.

For example, table 1 shows two EPs between poisonous class and edible class of the Mushroom dataset, from the UCI Machine Learning Repository [3], where X = {(Bruises, no), (Gill-Spacing, close), (Veil-Colour, white)}, and Y = {(Odour, none), (Gill-Size, broad), (Ring-Number, one)}. Both EPs have very large growth rates. As an EP of poisonous class, the EP X, with a growth rate of 21.4, is a three-attribute feature contrasting the poisonous instances against the edible instances. It has very high predictive power: the odd that instances containing (or satisfying) X are poisonous is 95.5%. As an EP of edible class, the EP Y has even greater predictive power: the odd that instances containing Y are edible is 100%. In fact, Y is a *jumping emerging pattern* (JEP) with support 0 in poisonous class, and not-zero in edible class, and thus growth rate ∞.

Table 1. Examples of Emerging Patterns: two EPs between poisonous class and edible class of the Mushroom dataset.

EP	Poisonous	Edible	Growth-rate
X	81.4%	3.8%	21.4
Y	0%	63.9%	∞

EPs capture the difference between two data classes on multi-attributes, so they can be used as the basic means for classification. By aggregating the differentiating power of EPs/JEPs, classification methods such as JEP-Classifier [12], and CAEP (Classification by Aggregating Emerging Patterns) [5] usually achieve higher accuracy than other state-of the art classifiers such as C5.0.

2.1 Eessential Emerging Pattern (eEPs)

Emerging patterns can be described by borders. A collection of sets represented by the border <L, R> is

$$[L, R] = \{Y \mid \exists X \in L, \exists Z \in R, X \subseteq Y \subseteq Z\}.$$

For instance, the border < {{a}, {b, c}}, {{a, b, c, d}}> represents those sets which are either supersets of {a} and subsets of {a, b, c, d}, or supersets of {b, c} and subsets of {a, b, c, d}. Note that EPs in the left border are the most general or minimal, i.e., any training instance covered by EPs in the border will also be covered by EPs in the left border.

There can be a very large number (e.g., 10^9) of common EPs in the dense and high-dimensional datasets of typical classification problems. It has been shown that many of them are not so useful in classification [7]. Suppose that X_1 and X_2 are two EPs, and $X_1 \subset X_2$. X_2 is less useful than X_1 for classification because every example covered by X_2 must also be covered by X_1. Shorter EPs contains fewer attributes, and tend to have larger supports. If we can use a few attributes to distinguish two data classes, adding more attributes will not contribute to classification, and in some worse cases, bring noise.

Previous works show that essential emerging patterns are sufficient for building accurate classifiers.

Definition 4: An itemset *X* is called an essential emerging pattern (eEP) of the target class *C* if it satisfies the following conditions:

(1) *X* is an EP of class *C* with high growth rate ρ, and

(2) any proper subset of *X* is not an EP of class *C*, and

(3) the support of *X* in class *C* is not less than ξ, where ξ is a predefined min-support threshold.

eEPs are believed to be the most expressive patterns for classification because of the following reasons:

- Large or even infinite growth rates ensure that each eEP has significant level of discrimination.
- The minimum support threshold makes every eEP cover at least a certain number of training instances, because itemsets with too low supports are regarded as noise.
- Supersets of eEPs are not useful for classification because of the following reason. Suppose $E_1 \subset E_2$, where E_1 is an eEP. E_1 covers more (at least equal) training instances than E_2, because $sup(E_1) >= sup(E_2)$. By the definition of eEP, both E_1 and E_2 have large growth rate. So E_2 does not provide any more information for classification than E_1.

In fact, eEPs are the shortest EPs contained in the left bound of the border representing the EP collections, and have at least a certain coverage rate on the training dataset. The set of eEPs is much smaller than the set of all common EPs. The classifiers based on eEPs is not only more efficient, but also more effective.

2.2 Tree Based Algorithms for Efficiently Mining eEPs

Efficient tree based algorithms have been developed to mine essential emerging pattern [7]. The tree data structure is called P-tree. Like FP-tree [10], P-tree stores compressed all the item information of the training data. Unlike FP-tree, P-tree keeps the class attributes information in order to mine EPs. The algorithm adopts the pattern fragment growth mining method: it recursively partitions the database into sub-database according to the patterns found and search for local patterns to assemble longer global one. It searches the tree in the depth-first manner; it operates directly on the data contained in the tree, i.e., no new nodes are inserted into the original tree and no nodes are removed from it during the mining process. The major operations of mining are counting and link adjusting, which are usually less expensive than the previous Apriori level-wise, candidate generation-and-test approach.

The details of these algorithms are omitted here. They can be found in [6,7].

3. Two Phase Classification By Emerging Patterns (TPCEP)

3.1 Basic Ideas

We use two-class problem to show the basic idea of TPCEP classifier. Suppose the two classes are C_1 and C_2. TPCEP mines eEPs of C_1 and C_2 in two phases and uses these eEPs and their supports to construct classifier of each phase. In the first phase, we use all of the training instances to mine eEPs. In the second phase, we focus on those instances that are covered by eEPs of the wrong class. For example, eEPs of C_1 from the first phase also cover some instances of C_2. Our aim is to mine another set of eEPs to identify these "false positives" or "true negatives".

When the growth rate threshold ρ is high enough, eEPs of C will cover many training instances of C (cover-rate = ξ) and few training instances of non-C.

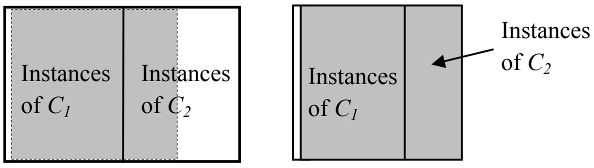
3.2 Mine P-eEPs and N-eEPs

In the first phase, we use all the training instances to generate eEPs of C_1 and C_2. By adjusting min-growth-rate threshold η and min-support threshold ξ, we make the eEPs of C_1 cover a certain percentage (e.g. 90% or more) of C_1. Here we don't care these eEPs' coverage on C_2. Figure 1 (a) expresses the process, where the two rectangles show the training instances of C_1 and C_2 respectively, the instances in the light-grey area are those instances that eEPs of C_1 covered. The eEPs cover most of the instances of C_1 and part of instances of C_2. Similarly, eEPs of C_2 cover many instances of C_2 while only a few instances of C_1.

The eEPs of C_1 (or C_2) mined in the first phase are called P-eEPs of C_1 (or C_2). Using the P-eEPs of C_1 and C_2 mined in the first phase, we can build a single-phase classifier. Suppose S is a test instance. Suppose S contains P-eEPs of C_1: $X_{11}, \ldots, X_{1m}$, with supports $s_{11},\ldots,s_{1m}$ respectively; S also contains P-eEPs of C_2: $X_{21}, \ldots, X_{2k}$, with supports $s_{21},\ldots,s_{2k}$ respectively. We use formula (1) (see section 3.3) to calculate the similarity rate of C_1 and C_2 for S, denoted as $SR_1(S, C_1)$ and $SR_1(S, C_2)$. The classifier of single-phase uses the following rules to classify S:

> If $SR_1(S,C_1) > SR_1(S,C_2)$ or ($SR_1(S,C_1) = SR_1(S,C_2)$ and $| C_1 | \geq | C_2 |$), the class label of S will be C_1, else be C_2.

Although the single-phase classifier is simple, it achieves good accuracy for classification. However, there is much room to improve it by adding the second phrase. (See experiment result and analysis)



(a) (b)

Fig. 1. The training instances used in mining eEPs in two phases. (a) In the first phase, we use all the training instances to generate P-eEPs of C_1; (b) in the second phase, we only use the instances that covered by P-eEPs of C_1 mined in the first phase to generate N-eEPs of C_1.

In the second phase, to generate eEPs of C_2 (or C_1), we only use the instances that covered by P-eEPs of C_1 (or C_2) mined in the first phase. Take mining eEPs of C_2 for example. Fig.1 (b) shows those instances used in the second phase. Here, we use all the instances of C_1 and the instances of C_2 covered by P-eEPs of C_1 to mine eEPs of C_2. The eEPs of C_2 mined here in the second phase are also called N-eEPs of C_1, because they are "negative" eEPs of C_1. Similarly, the eEPs of C_1 mined in the second phase are also called N-eEPs of C_2.

Now we have P-eEPs (the first phrase) and N-eEPs (the second phrase) for each class. In the second phase we pay more attention on the possible misclassification than the first phase. Since we have additional EPs mined in the second phrase, we need to design a new scoring method to use eEPs of both phases and their supports to make a better classification.

3.3 Scoring function (Calculation of Similarity Rate)

To classify an unseen instance *S*, TPCEP needs to calculate a score for each class. The class with the highest score is then returned as the classification. The idea behind the scoring function is that if the unknown instance *S* belongs to class C_i, *S* will be "similar" to the instances of C_i. The similarity can be measured using the eEPs of C_i that are contained in *S*. Suppose *S* contains an eEP of C_i, denoted as E_1. E_1 has enough support, which suggests *S* is consistent to a reasonable number of instances of C_i on a group of attributes. E_1 also has large growth rate, which means *S* is very unlike the classes other than C_i. Intuitively, the more percentage eEPs of *S* cover the instances of C_i, the more similar *S* is to C_i.

Definition 5: Given a test instance *S* and a set *E*(*C*) of EPs of a class *C* discovered from the training data, the similarity rate of *S* for *C*, denoted as *SR*(*S*,*C*), can be calculated using the following steps:

1. Find the EPs from *E*(*C*) which are contained by *S*, denoted as $X_1, \ldots, X_m$.
2. set *count* = 0;
3. for each instance in *C*, if it contains one EP from $X_1, \ldots, X_m$, then *count*++;

4. *SR*(*S,C*) = *count* / the total number of instances of *C*

Step 3 counts how many instances of *C* are covered by the set of EPs $X_1, ..., X_m$ collectively. Here we say an instance is covered by the EP set if the instance contains one EP from that set.

We can use the eEPs and their supports to calculate *SR*(*S,C*) approximately. Suppose *S* is a test instance and it contains eEPs $X_1, ..., X_m$, of C_i, where the supports are $s_1, ..., s_m$, respectively. Let A_i be the event "X_i appears in the instances of *C*". The calculation of *SR*(*S,C*) is equivalent to the calculation of probability $P(A_1 \cup \ldots \cup A_m)$. So, we have:

$$SR(S,C) = P(A_1 \cup ... \cup A_m)$$
$$= \sum_{i=1}^{m} P(A_i) - \sum_{1 \le i < j \le m} P(A_i A_j) + \sum_{1 \le i < j < k \le m} P(A_i A_j A_k) + ... + (-1)^{m-1} P(A_1 A_2 ... A_m)$$

where $P(A_i) = s_i$. Assuming that eEPs are independent, we have the following approximation:

$$SR(S,C) = \sum_{i=1}^{m} s_i - \sum_{1 \le i < j \le m} s_i s_j + \sum_{1 \le i < j < k \le m} s_i s_j s_k + ... + (-1)^{m-1} s_1 s_2 ... s_m \quad (1)$$

3.4 TPCEP Classification

In training phase, the task of TPCEP classification is to mine eEPs with their supports in two phases. eEPs and their supports construct the basis of TPCEP classification.

Definition 6: Suppose *S* is an unclassified instances, $SR_1(S, C_i)$ is the similarity rate of C_i for *S* using eEPs of C_i mined in the first phase (i.e., P-eEPs of C_i), and $SR_2(S, C_j)$ ($j \neq i$) are the similarity rate of C_j for *S* using eEPs mined in the second phase (i.e., N-eEPs of C_i). The score of *S* for C_i is defined as $score(S, C_i) = SR_1(S, C_i) - SR_2(S, C_j)$, where $i, j = 1$ or 2, $i \neq j$.

The scoring method considers to use the second phase to correct the errors made in the first phrase.

Given a test instance *S*, *S* will be classified as the class with the highest score. That is, after TPCEP calculates $score(S,C_1)$ and $score(S, C_2)$, it uses the following rules to decide the class label for *S*:

> If $score(S,C_1) > score(S,C_2)$ or $score(S,C_1) = score(S,C_2)$ and $|C_1| \ge |C_2|$,
> the class label of *S* will be C_1, else be C_2.

3.5 Multiple Classes Problem

TPCEP can be easily extended to *k* (>2) classes. For a *k*-class problem, we build *k* classifiers: $G_1, ..., G_k$. Firstly, we use all the training data and partition it into two classes: C_1 and non-C_1. We build classifier G_1 in the first step. Then the whole training data is regarded as another two classes: C_2 and non-C_2 (including C_1). In the second step we build the classifier G_2. Generally, to build G_i, we divide the whole training dataset into two classes: C_i and non-C_i.

To classify a test S, we use G_1 ,…, G_k to decide the class label of S. After computing all the scores of S for C_i, we compare the scores and assign S to the class with the highest score.

4. Experiment Result and Analysis

In order to investigate TPCEP's performance compared with that of other classifiers, we carry experiments on 10 datasets from UCI Machine Learning Repository. We compare TPCEP with other state-of-the-art classifiers: Naive Bayes (NB), the two important classifiers based on association rules CBA [14] and CMAR [13], the widely known decision tree induction C5.0; an EP-based classifier CAEP; and BCEP, a novel Bayes classifier based on emerging patterns.

Our experiments were performed on a 900Mhz Pentium III PC with 128Mb of memory. The programming environment is Microsoft Visual C++. The accuracy was obtained by using the methodology of ten-fold cross-validation. We use the Entropy method in [8] to discretize datasets containing continuous attributes. Experiment results of the competitive classifiers are taken from their original papers.

Table 2. Summary of the predictive accuracy of of classifiers.

Dataset	Accuracy (%)							
	CBA	CMAR	NB	C5.0	CAEP	BCEP	TPCEP	One-phase
Adult	---	---	84.12	85.54	83.09	85.00	80.30	78.60
Australia	84.90	86.10	85.65	84.93	85.51	86.40	89.30	86.97
Cleve	82.80	82.20	82.78	77.16	82.13	82.41	91.30	86.30
Diabete	74.50	75.80	75.13	73.03	67.30	76.80	77.30	71.78
German	73.40	74.90	74.10	71.90	74.50	74.50	73.30	72.08
Heart	81.90	82.20	88.22	76.30	82.22	81.85	88.93	76.70
Mushroom	---	---	99.68	100.00	93.91	100.00	99.20	98.70
Pima	72.90	75.10	75.90	75.39	77.60	75.66	77.70	71.82
Tic-tac	99.60	99.20	70.15	85.91	85.91	99.37	85.80	77.40
Vehicle	68.70	68.80	61.12	73.68	68.80	68.05	66.30	---
Average			79.69	80.38	80.10	83.00	82.94	

Table 2 summarizes the accuracy results. From the table, we can see that TPCEP achieves the best accuracy on 5 datasets and also performs well on the other datasets. The average accuracy of TPCEP is higher than that of NB, C5.0, and CAEP; and it is almost the same as BCEP. The advantage of TPCEP over BCEP is that TPCEP is much faster. BCEP is slow because it has to calculate probability approximation using many itemsets in a chain of product. TPCEP is fast due to a relatively simple scoring function. TPCEP dose not degrade accuracy because two-phrase mechanism can correct the errors made by simple scoring to some extend.

Comparing to the CBA and CMAR, two classifiers based on association rules, we can see that TPCEP wins on five datasets, Australia, Cleve, Diabete, Heart, Pima, but loses on the three datasets, namely, German, Tic-tac and Vehicle.

In the last column, we give the results obtained by EP-based single-phase classification that uses the same scoring function as TPCEP. We can see that single-phase classifier is fairly good. Further more, we can see that TPCEP wins its single-phase counterpart on most of the datasets. These experimental results confirm our belief that two-phase classification has advantages over one-phase.

5. Conclusion

In the paper, we have proposed a new novel classifier, i.e., Two-Phase Classification by Emerging Patterns (TPCEP). TPCEP combines the benefits of two-phase classification method and classification by emerging patterns. The first phase of TPCEP aims to find the EPs that have high supports and high coverage on the training data. Here we alleviate the strict requirement of high support-ratio EPs. The second phase will then tries to mine another set of EPs which can remove false positives in the collection of the instances covered by the first phase EPs. Here we correct the errors due to the use of moderate support-ratio EPs whose discriminating power are not so sharp. Our experiment study carried on 10 benchmark datasets from the UCI Machine Learning Repository shows that TPCEP performs comparably with other state-of-the-art classification methods such as CBA, CMAR, C5.0, NB, CAEP LB and BCEP in terms of overall predictive accuracy.

The factors that can affect the accuracy of TPCEP are the differentiation power of EPs and their coverage on data. These factors are represented by two interrelated parameters: min-support threshold and min-growth rate threshold. Generally, fixing min-support threshold, higher growth rate will result in more discriminating EPs, but can reduce coverage on the training data (does not generalize well). Fixing growth rate, higher min-support threshold may lead to lower coverage; but when min-support is too low, EPs may be not statistically significant and thus the classifier built upon them tends to overfit. To select the right values for these thresholds is the art of human, guided by trial and error. As the future work, we will go deeper on the problem of automatic optimization of the two parameters.

References

1. R. Agarwal, and M. V. Joshi. PNrule: A new Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection). In Proc. of the First SIAM Conference on Data Mining. Chicago, USA, April 2001.
2. R. Agarwal, M. V. Joshi, and V. Kumar. Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction. In Proc. of the 2001 ACM SIGMOD, pp 91-102, Santa Barbara, California, USA, May 2001.

3. C. Blake, and C. Merz. UCI repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.
4. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In Proc. of KDD'99, pp 15-18, San Diego, USA, Sep. 1999.
5. G. Dong, X. Zhang, L. Wong and J. Li. CAEP: Classification by Aggregating emerging patterns. In Proc. of the 2nd Int'l Conf. On Discovery Science (DS'99), pp 30-42, Tokyo, Japan, Dec 1999.
6. H. Fan and K. Ramamohanarao. An Efficient Single-Scan Algorithm for Mining Essential Jumping Emerging Patterns for Classification. In Proc. of 2002 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'02), pp455-562, Taipei, Taiwan, China, May 6-8, 2002.
7. H. Fan and K. Ramamohanarao. A Bayesian Approach to use Emerging Patterns for Classification. In Proc of the 14th Australasian Database Conference. pp 39-48 Feb 2003.
8. U. M. Fayyad and K. B. Irani. Multi-interval Discretization of Continuous-valued Attributes for Classification learning. In Proc. the 13 International Joint Conference on Artificial Intelligence (IJCAI), pp.1022-1029, San Francisco, USA.
9. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2000.
10. J. Han, J. Pei, J. and Y. Yin. Mining frequent patterns without candidate generation. In Proc. of the 2000 ACM-SIGMOD Intl. Conf. on Management of Data, pp 1-12, May 2000.
11. Holte, R. C., Acker, L. E. and Porter, B. W. (1989). Concept Learning and the Problem of Small Disjuncts. In Proc. of Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), 1989, pp 813-818.
12. J. Li, G. Dong and K. Ramamohanarao. Making Use of the Most expressive Jumping Emerging Patterns for Classification. In Proc. of 2000 Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'00), pp 220-232.
13. W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In ICDM'01, pp. 369-376, San Jose, CA, Nov. 2001.
14. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In KDD'98, pp. 80-86, New York, NY, Aug. 1998.

Feature Preparation in Text Categorization

Ciya Liao[1], Shamim Alpha[1], Paul Dixon[1]

[1] Oracle Corporation,
{david.liao, shamim.alpha, paul.dixon}@oracle.com

Abstract. Text categorization is an important application of machine learning to the field of document information retrieval. Most machine learning methods treat text documents as a feature vectors. We report text categorization accuracy for different types of features and different types of feature weights. We report the classification result for neural network classification and Support Vector Machine by using Reuters and Ohsumed collections. We found that SVM is superior to neural network classification in both effectiveness and efficiency. In our experiments, we did not see any significant improvement for classification accuracy by using noun-phrase features. The comparison of those classifiers shows surprisingly that the simply stemmed or un-stemmed single words as features give a better classifier compared to other type of features.

1 Introduction

Text categorization is a conventional classification problem applied to the textual domain. It solves the problem of assigning text content to predefined categories. As the volume of text content grows continuously on-line and in corporate domains, text categorization, acting as a way to organize the text content, becomes interesting not only from an academic but also from an industrial point of view. A growing number of statistical classification methods have been applied to text categorization, such as Naive Bayesian (Joachims,1997), Bayesian Network (Sahami,1996), Decision Tree (Quinlan,1993;Weiss,1999), Neural Network(Wiener,1995), Linear Regression(Yang,1992), k-NN (Yang,1999), Support Vector Machines (Dumais,1998; Joachims, 1998), and Boosting (Schapire,2000; Weiss,1999). A comprehensive comparative evaluation of a wide-range of text categorization methods is reported in ref.(Yang,1999; Dumais,1998) against the Reuters corpus.

Most of the statistical classification methods mentioned above are borrowed from the field of machine learning, where a classified item is treated as a feature vector. A simple way to transform a text document into a feature vector is using a “bag-of-words” representation, where each feature is a single token. There are two problems associated with this representation.

The first problem to be raised when using a feature vector representation is to answer the question, “what is a feature?”. In general, a feature can be either local or global. In text categorization, local features are always used but in different-length scales of locality. A feature can be as simple as a single token, or a linguistic phrase, or a much more complicated syntax template. A feature can be a characteristic quantity at different linguistic levels. To transform a document, which can be regarded as a string of tokens, into another set of tokens will lose some linguistic information such as word sequence. Word sequence is crucial for a human being to understand a document and should be also crucial for a computer. Using phrases as features is a partial solution for incorporating word sequence information into text categorization. This paper will investigate the effectiveness of different classifiers by using single tokens, phrases, stemmed tokens, etc. as features.

The second problem is how to quantify a feature. A feature weight should show the degree of information represented by local feature occurrences in a document, at a minimum. A slightly more complicated feature weight scheme may also represent statistical information of the feature's occurrence within the whole training set or in a pre-existing knowledge base (taxonomy or ontology). A yet more complicated feature weight may also include information about feature distribution among different classes. This paper will only investigate the first two types of feature weights.

2 From Text to Features

In order to transform a document into a feature vector, preprocessing is needed. This includes feature formation (tokenization, phrase formation, or higher level feature extraction), feature selection, and feature score calculations. Tokenization is a trivial problem for white-spaced languages like English.

Feature formation must be performed with reference to the definition of the features. Different linguistic components of a document can form different types of features. Features such as single tokens or single stemmed tokens are most frequently used in text categorization. In this bag-of-words representation, information about dependencies and the relative positions of different tokens are not used. Phrasal features consisting of more than one token are one possible way to make use of the dependencies and relative positions of component tokens. Previous experiments (Sahami,1996; Dumais,1998) show that introducing some degree of term dependence in the Bayesian network method will achieve undoubtably higher accuracy in text categorization compared to the independence assumption in the Naive Bayesian method. However, whether the introduction of phrases will improve the accuracy of text categorization has been debated for a long time. Lewis (Lewis,1992) was the first to study the effects of syntactic phrases in text categorization. In his study, a naive Bayesian classifier with only noun phrases yielded significantly lower effectiveness than a standard classifier using bag-of-single-words. More reports on inclusion of syntactic phrases show no significant improvement on rule-based classifiers (Scott,1999) and naive Bayesian and SVM classifiers (Dumais,1998). For statistical phrases like n-grams, one report (Caropreso,2001) shows that certain term selection methods such as document frequency, information gain and chi-square give high selection scores to a considerable number of statistical phrases, which indicates they have important predictive value. In the same report, directly using selected uni-grams or bigrams during text categorization with the Rocchio classifier yields a slightly higher effectiveness compared to only using uni-grams in the case that the classifier chooses an adequate but equal number of terms as features. A significant drop in effectiveness was observed when the classifier chose fewer terms. The report then commented that inclusion of some bigrams may only duplicate information of existing uni-grams but force other important uni-grams out. However, other reports on statistical phrases show that the addition of n-grams to the single words model can improve performance in the shorter-length n-grams case(Furnkranz, 1998; Mladenic,1998).

One type of a higher level feature has been studied in text categorization (Furnkranz,1998), where linguistic patterns were extracted automatically and input as features to naive Bayesian and rule-based classifiers. A consistent improvement in precision was observed in the naive Bayesian classifier and at low recall level in the rule-based classifier. Adding linguistic patterns to the single word representation yields consistent improvement of precision except at a very high recall level.

Feature selection has been studied by (Yang,1997), where information gain and chi-square methods are found most effective for k-NN and linear regression learning methods. Term selection based on document frequency in the training set as a whole is simple but has similar performance to information gain and chi-square methods.

Selected features must be associated with a numerical value to evaluate the impact of the feature to the classification problem. Most types of feature weighting schemes in text categorization are borrowed from the field of information retrieval. The most frequently used weight is TFIDF (Salton, 1988). The original TFIDF is:

$$\omega_{fd} = tf_{fd} \log \frac{D}{df_f} \tag{1}$$

where ω_{fd} is the weight of feature f in document d, tf_{fd} the occurrence frequency of feature f in document d, D the total number of documents in the training set, and df_f is the number of documents containing the feature f.

In this paper, we will compare text categorization using different types of features, and different types of feature weighting schemes. The feature types will include single tokens, single stemmed tokens, and phrases. Weighting schemes will include binary feature (BI), term frequency (TF), TFIDF(eq.1), logTFIDF(eq.2), etc.

$$\omega_{fd} = \log(tf_{fd} + 0.5) \log \frac{D}{df_f} \tag{2}$$

We note that the logarithm of the TF part is to amend unfavorable linearity. The machine learning algorithms we report in this paper include SVM (Joachims, 1998) and Neural Network. Feature selection in Neural Networks and Support Vector Machine classifiers is based on document frequency. Only features (single words or phrases) occurring in an adequate number of training documents will be selected. The corpus includes reuters-21578 and ohsumed.

3 Phrase Features

We only use training set documents to find valid phrases. We first scan the documents in the training set and detect phrases based on linguistic and statistical conditions. We only use noun phrases as valid phrases. Valid phrases are inserted into a phrase database which is specific to the training set. The phrase database is used to replace the phrases in the training documents and test documents with specific tokens. For example, the phrase "information retrieval" in the document will be changed to the token "information_retrieval". After phrases in the documents are marked, the documents can be input into tokenization program in training or classification processes for performance testing.

To detect valid noun phrase chunking, Brill's transformation-based part of speech tagger (Brill, 1995) was used to mark parts-of-speech in the training documents. Training documents with POS tags are input into Ramshaw&Marcus's noun phrase chunking detector (Ramshaw,1995) for noun phrase detection. The resultant noun phrase chunks are output to a file, which is input to a statistical chi-square test program. This program tests the statistical significance of co-occurrences of the component tokens in n-gram noun phrases. In particular, we choose the noun phrases (ngrams, up to 4-grams) such that the null hypothesis that its component tokens are independent of each other can be proved not true.

4 Machine Learning Algorithms

We test two different type of machine learning algorithms: Neural Networks and Support Vector Machines. We use the SVM_light package (Joachims, 1998) with default parameter settings, which results in a linear SVM classifier.

For Neural Network, we use a home-made program. The Neural Network has no hidden layer and therefore is equivalent to a linear classifier. Text document classification has high dimensional data characteristics because of the large size of natural language vocabulary. Documents in one class usually can be linearly separated from other classes due to high dimensionality (Joachims,1998;Schutze,1995). A prior experiment (Schutze, 1995) shows that linear neural networks can achieve the same accuracy as non-linear neural networks with hidden layers.

During the learning process, a sequential back propagation algorithm is used to minimize training error. We use cross-entropy error, thus making our learning method equivalent to logistic regression learning (Schutze, 1995). We tried to use weight regularization methods (Zhang, 2001) to deal with overfitting, but the accuracy was not improved and convergence is hard to achieve by using back propagation learning. The results we present in this paper do not use regularization.

5 Corpus

The evaluation experiments are done on two text collections. The first is Reuters-21578 with ModApte split. Many text categorization methods have been tested against this corpus (Yang,1999; Dumais,1998; Joachims,1998). This is a collection of newswire stories from 1987 compiled by David Lewis. The number of distinct tokens in the training set is 39189, of which 18586 tokens occur more than once, 12951 tokens occur more than twice, 10328 tokens occur more than three times, 8789 tokens occur more than four times, and 3262 tokens occur more than 20 times in the training set.

The second collection is taken from Ohsumed corpus used in the Filtering Track in TREC-9 (Robertson,2000). The Ohsumed collection consists of Medline documents from the years 1987-1991 and a set of topics and relevance judgments. In order to reduce the size of the problem, we chose MESH categories in which the number of Ohsumed documents in 1991 is larger than 300 (which results 98 categories). The training/testing split is across the document series number 91250000. Training documents have the document series number less than 91250000. This split results in 14655 training documents and 6698 test documents. The resultant training set and testing set have more homogenous distribution across different categories than the Reuters collection. The minimum (maximum) number of training documents in one category is 65 (465). The minimum (maximum) number of testing documents in one category is 29(214). In the training set, there are 52162 distinct tokens, of which 28857 tokens occur more than once, 22128 tokens occur more than twice, 18493 tokens occur more than three times, 9224 tokens occur more than 12 times, and 3458 tokens occur more than 60 times.

6 Experiment Results and Discussion

We first compare the impact of different feature types. The meaning of the following legends in the figure denote feature types: “single words” means only single tokens as features, “noun phrases” uses detected noun phrases as features without using component tokens, “stem words” uses Porter-stemmer-determined stems for each token as features, “noun phrases and words” uses detected noun phrases and their component tokens.

Fig.1 and Fig.2 show the micro-average breakeven points (BEP) with different numbers of features using the SVM classifier to classify the Reuters and ohsumed corpora, respectively. Breakeven accuracies increase with the number of features. There is no overfitting observed in the experimental range as the number of features increases. The maximum number of features in Fig.1 is 16000 for Reuters. This number of single tokens is roughly the number of tokens occurring twice or above in the training set. The maximum number of features in Fig.2 is 20000 for ohsumed. This number of single tokens is roughly the number of tokens occurring three times or above in the training set.

The maximum BEPs are achieved at the maximum number of features. For reuters, the best BEP is 0.88, which is slightly higher than the reported microAvg. BEP in (Joachims, 1998) (0.860). For ohsumed, the best BEP is 0.602, achieved by using stem words.

The effect of stemming can be easily seen in Fig. 1 and 2. When the number of features is small, the coverage of selected features is poor but stemming of words can increase the feature coverage, thus giving the best breakeven accuracy compared to other types of features. However, as the number of chosen features increases, and coverage of the chosen features becomes large enough, the accuracy of the features in conveying the information becomes more important. This can be seen in Fig.1 where the BEPs of other types of features are as good as stem words at large number of features.

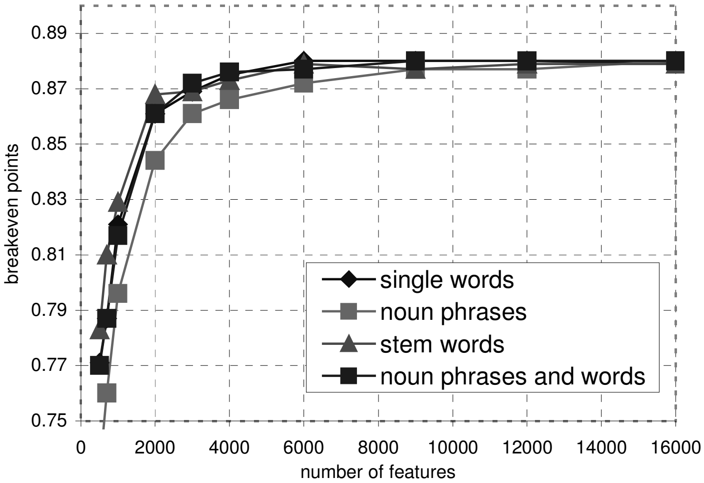


Fig. 1. Breakeven points for the Reuters collection using SVM. The feature vector is normalized to have unity sum. The feature weight is LOGTFIDF.

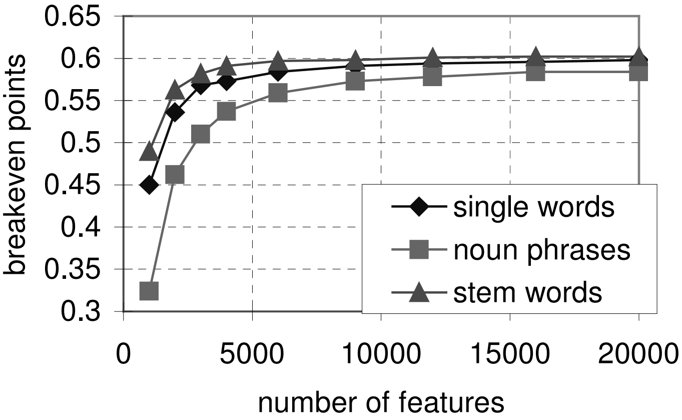


Fig. 2. Breakeven points for the ohsumed collection using SVM. The feature vector is normalized to have unity sum. The feature weight is LOGTFIDF.

It is interesting to see how well the automatically determined noun phrase features perform. The Fig.1 and Fig. 2 show that noun phrases classifiers give the worst BEP. This is disappointing. It was expected that good phrase features provide more accurate information by constraining the meaning of component words. One example is the phrase "consumer price index", where the combination of the three token means a specific index. However, this result is consistent with the findings of the previous literature (Lewis,1992; Scott,1999; Dumais,1998). This can be partially explained by decreased feature coverage due to replacement of original tokens by phrases. For example, if the two words "oracle database" are replaced by one phrase "oracle_database", this phrase feature will only match itself and can not convey any similarity with its individual word components "oracle" and "database". However, we know that an article discussing enterprise software may only mention "oracle" or "database" separately. Another example is replacing the two phrases "Oracle database" and "DB2 database" with two separate features makes it impossible to map the similarity existing between them. Thus each phrase will have narrower coverage. The coverage problem is caused by replacing a number of token features with a single feature which is more accurate but finer. This problem can be partially reduced by not eliminating the phrase's component words. In fig.1, we see a significant increase of BEP for noun phrases including component words.

The single words are natural linguistic units and are employed by many text classification systems. From Fig.1 and Fig.2, one can see that this natural and simple feature unit performs fairly well compared with other complicated types of features.

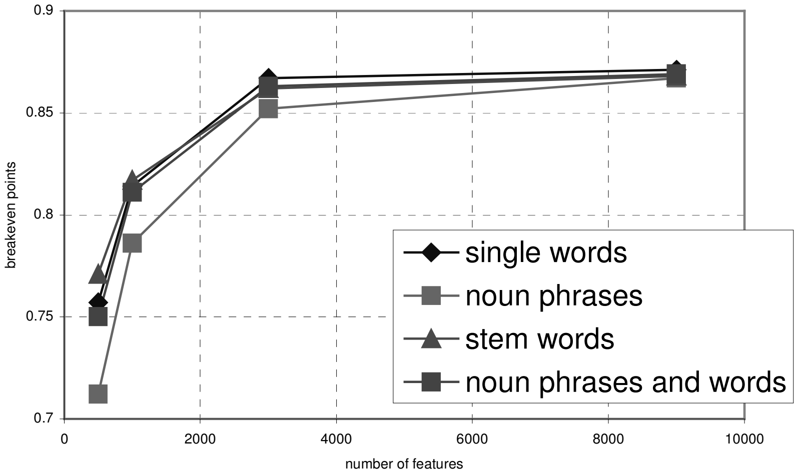


Fig. 3. Breakeven points for the Reuters collection using Neural Network. The feature vector is normalized to have unity sum. The feature weight is LOGTFIDF

The above results and discussion about types of features are not just applied to a single machine learning algorithm (SVM in Fig.1, Fig.2). We performed the same experiments with a Neural Network classifier. The results are shown in Fig.3 and Fig.4. The BEPs using Neural Network are not as good as those using SVM. The maximum BEP using Neural Network is 0.871 for Reuters using single words and 0.568 for ohsumed using stem words. It is worthwhile to mention that the training time for the neural network is much longer than SVM (>10 times in this experiment's problem scale).

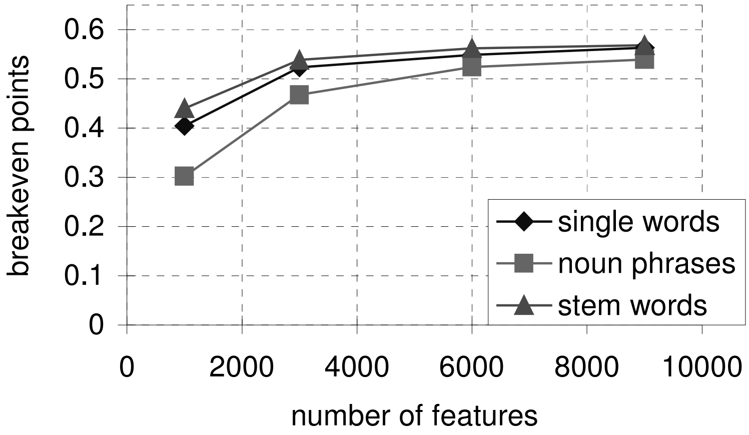


Fig. 4. Breakeven points for the ohsumed collection using Neural Network. The feature vector is normalized to have unity sum. The feature weight is LOGTFIDF

So far, the discussed results all use one feature weighting scheme: LOGTFIDF (eq. 2). In Fig.5, Fig.6, we employed different feature weighting schemes. They are:

– IDF:	$\log(D/df_f)$
– TF:	tf_{fd}
– TFIDF:	eq. 1
– LOGTFIDF:	eq. 2
– LOGTF:	$\log(tf_{fd})$
– BINARY:	1 or 0

Although the BEP ranking sequence for different weighting schemes is different for Fig.5 and Fig.6, we still can find some common characteristics for weighting methods based on the results vs. the Reuters and ohsumed collections. One can observe that the BEPs using LOGTF weight are always larger than those using TF weight. This shows that non-linear weighting of term frequency is better than conventional linear weighting. We think that this observation also holds in the field of information retrieval for relevance ranking.

Fig.5 and Fig.6 show that IDF weighting is better than BINARY weighting. Because IDF weight is only assigned to a feature occurring in the concerned document, IDF weight is actually BINARY weight multiplying an IDF score which contains the statistical information of the feature inside the whole corpus. Considering the fact that TFIDF is better than TF, LOGTFIDF better than LOGTF, one can than conclude that introducing the corpus information helps improve the accuracy of text categorization.

It is seen from Fig.5 and Fig.6 that LOGTFIDF, which is the multiplication of the LOGTF and IDF weights, performs the best in both collections.

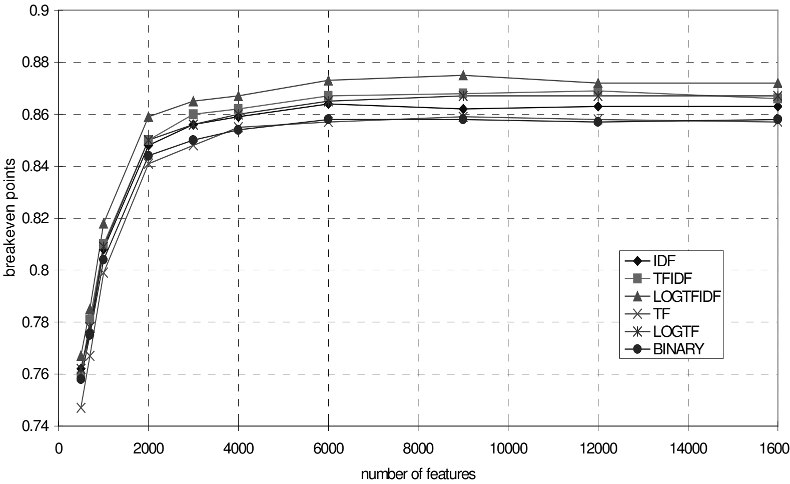


Fig. 5. Breakeven points for the Reuters collection using SVM. The feature vector is normalized to have unity length. The features are single words.

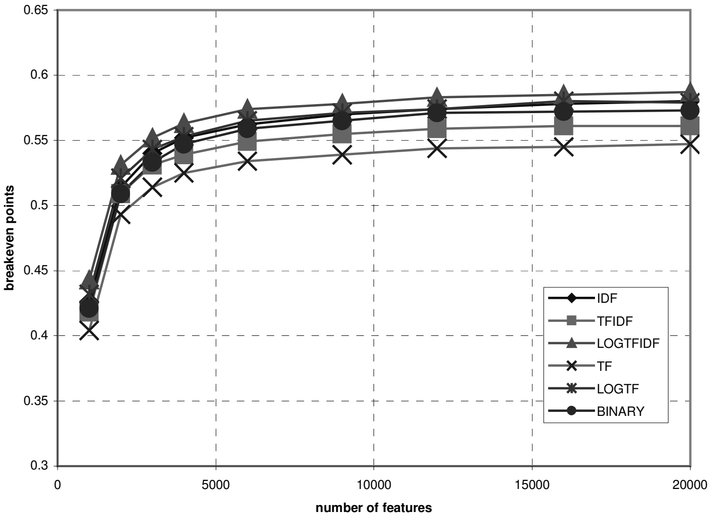


Fig. 6. Breakeven points for the ohsumed collection using SVM. The feature vector is normalized to have unity length. The features are single words.

7 Conclusions

We have compared text classifiers using different types of features and different weighting schemes. In order to achieve a generic conclusion regarding the feature preparation independent with machine learning algorithms, we employed Support Vector Machines and a Neural Network algorithm as two machine learning classifiers. We tested these classifiers on the Reuters and ohsumed collections. Based on this comparison, we find that SVM algorithm is superior to the linearized neural network both in accuracy and training speed. Stem words, which normalize different feature forms to one stem form, show significant advantages in the case where a small number of features are used because of the larger coverage of the stems. Replacing contiguous tokens with detected noun phrases as features gains accuracy but loses coverage due to the problem of normalization. One may surmise that if a similarity match between different features is introduced to replace the current binary match between two features, the feature normalization problem will be eliminated. Single words as features perform fairly well. The comparison of different weighting schemes shows LOGTFIDF as the preferred feature weighting method.

References

1. Joachims T.(1998), Text categorization with support vector machines: learning with many relevant features. Proceedings of ECML-98, 10th European Conference on Machine Learning.
2. Yang Y., Pederson J.O.(1997), A comparative study on feature selection in text categorization, International Conference on Machine Learning (ICML).
3. Yang Y.(1999), An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, Vol.1, No.1/2.
4. Weiss S.W.,Apte C.,Damerau F.J., Johnson D.E.,Oles F.J.,Goetz T.,Hampp T.(1999), Maximizing text-mining performance,IEEE Intelligent systems.
5. Quinlan J.R.(1993), C4.5: Programs for machine learning, Morgan Kaufmann.
6. Joachims T.(1997), A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, Proceedings of ICML-97, 14th International Conference on Machine Learning.
7. Wiener E.(1995), J.O.Pederson, A.S.Weigend, A neural network approach to topic spotting. Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval.
8. Dumais S., Platt J., Heckerman D., Sahami M.(1998), Inductive learning algorithms and representations for text categorization. Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM 98).
9. Sahami M. (1996),Learning limited dependence Bayesian classifier. In KDD-96: Proceedings of the second international Conference on Knowledge Discovery and Data Mining, 335-338. AAAI press.
10. Yang Y., Chute C.G.(1992), A linear least squares fit mapping method for information retrieval from natural language texts. Proceedings of the 14th International Conference on Computational Linguistics (COLING 92).
11. Schapire R.E., Singer Y.(2000),Boostexter: A boosting-based system for text categorization. Machine Learning, 39(2/3).
12. D.D.Lewis D.D.(1992), Feature selection and feature extraction for text categorization, Proceedings of Speech and Natural Language Workshop.
13. Scott S., Matwin S.(1999), Feature engineering for text classification, Proceedings of ICML-99, 16th International Conference on Machine Learning.
14. Caropreso M.F., Matwin S., Sebastiani F.(2001), A learner-independent evaluation of the usefulness of statistical phrases in automated text categorization, Text database and Document Management: Theory and Practice.
15. Furnkranz J.(1998), A study using n-gram features for text categorization. Technical Report OEFAITR-98-30.
16. Mladenic D., Grobelnik M.(1998), Word sequences as features in text-learning. Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conferences.
17. Furnkranz J., Mitchell T., Riloff E., A case study of using linguistic phrases for text categorization on the WWW, Proceedings of the 1st AAAI Workshop on Learning for Text Categorization.
18. Brill E.(1995),Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging, Computational Linguistics, 21(4).

19.Ramshaw L.A., Marcus M.P.(1995), Text chunking using transformation-based learning, Proceedings of third Workshop on Very Large Corpora.
20.Salton G., Buckley C.(1988), Term Weighting Approaches in Automatic Text Retrieval, Information Processing and Management, Vol. 24, No.5, P513.
21.Schutze H., Hull D.A. (1995), J.O.Pederson, A Comparison of Classifiers and Document Representations for the Routing Problem, in Proceedings of SIGIR95.
22.Zhang T., Oles F.J.(2001), Text Categorization Based on Regularized Linear Classification Methods, Information Retrieval, vol.4.
23.Robertson S., Hull D.A.(2000), The TREC-9 Filter Track Final report, Proceedings of the Ninth Text Retrieval Conference (TREC-9).

Learning Quantitative Gene Interactions from Microarray Data

Michael Bain[1,3] and Bruno Gaëta[1,2,3]

[1] School of Computer Science and Engineering, University of New South Wales, Sydney, Australia 2052
{mike,bgaeta}@cse.unsw.edu.au
[2] School of Biotechnology and Biomolecular Sciences, University of New South Wales, Sydney, Australia 2052
[3] The Clive & Vera Ramaciotti Centre for Gene Function Analysis

Abstract. The development of microarray technology has enabled the states of entire genomes comprising tens or hundreds of thousands of genes to be measured in controlled experiments. The computational inference of genetic regulatory networks is a rapidly emerging area of bioinformatics which requires the application of techniques from statistics and data-mining to microarray data. A key problem is that the underlying biological systems are believed to be quite complex, and may incorporate many genes at various times, although often there are relatively few samples in each experiment. This paper describes recent experiments in automatically constructing quantitative gene interaction models from microarray data. The approach is based on earlier work on behavioural cloning, where reactive agents are learned from the logged data of skilled human operators controlling a complex dynamic system. In behavioural cloning the task of controlling the system is decomposed into that of learning individual agents for the the control of a particular variable. Combining the agents results in a strategy for the successful control of the overall system. In this paper selected target genes are treated as agents and tree-structured models are applied to learn regulatory dependencies from a benchmark data set. We describe some interesting aspects of the approach and outline directions for extending the work, including the design of an interactive system for biologists to use these machine learning methods in exploratory data analysis.

1 Introduction

The increasing availability of technology to accomplish large scale assays is rapidly moving biological data analysis requirements into the area of experimental "systems" biology. This presents a challenge to devise methods suitable for discovery of biological knowledge from such data resources. It also makes available a role for the practice of data mining and machine learning alongside more traditional hypothesis testing methodologies in biology.

Genome-wide measurement of the activity of regulatory networks in cells is now possible using the technology of DNA microarrays [2]. Such snapshots can be combined to enable inference from data of the causal relationships between genes controlling the processes in living cells. Knowledge of such properties holds tremendous promise. For example, it may aid in understanding fundamental biological processes such as the cell life cycle. The effect on genes of drug treatments or environmental factors, or the role of genes in diseases such as cancer can also be investigated by these methods.

2 Biological background

This section provides a brief review of the general problem of the underlying biological systems of interest, how they might be modelled, and how such models might be learned.

2.1 Regulation in biological systems

Microarray technology allows experimental evaluation not of what constitutes the genome (the full complement of genetic information of the organism) or even the proteome (the protein-coding regions of the genome) but the transcriptome (the population of mRNA transcripts in the cell weighted by their expression levels) [7]. The complexity of biological systems arises from the differential expression of genes which results in the variation of gene products (such as proteins) available to participate in cell function. Typically only a small fraction of genes in a genome are expressed at any one time. Protein synthesis incurs an energy cost, so unnecessary gene expression is inefficient. The main stages in gene expression and their points of regulation are shown in Figure 1.

So far our description of gene expression has left out many important details. The key point, however, is that the process is regulated to maintain the required concentrations of proteins (and other biochemical compounds) in the cell. Regulation of the initiation of transcription in particular allows the synchronised expression of multiple gene products on which the control of cellular processes depends. The relationships involved in such regulatory dependencies are typically very complex. The interaction between protein and DNA involves a region called a promoter to which regulatory proteins bind. Once the appropriate proteins, called transcription factors, are bound as a transcription complex, the transcription of the gene can begin, shown in Figure 2

Biological activity in cells can thus be viewed in terms of systems of interrelated molecular compounds undergoing self-regulation. Control is achieved by pathways of regulatory dependencies. The effect of transcription regulation can be positive, to activate transcription, or negative, to suppress it. Through evolution these systems have developed to sustain the cell in an environment, so regulation must not only control the internal operation but also the movement of compounds in and out from the extracellular region. These metabolic and signalling processes also have an effect on the expression of genes.

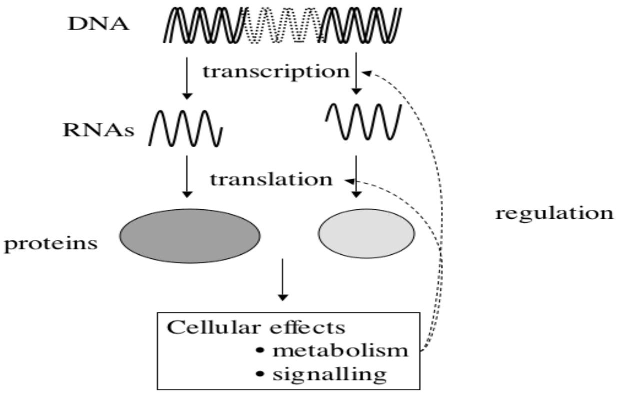


Fig. 1. Gene expression. In this example two genes (dark regions on the DNA) are transcribed into RNAs then translated into proteins. These gene products contribute to cellular effects, including possible regulation of transcription and translation

2.2 Microarray technology

The basis of microarray technology is in the ability of DNA and RNA to bind together in complementary strands (for example in the well-known double helix). Essentially, microarrays are constructed for a set of probes, complementary to a set of genes, such as the genome of an organism. These probes are bound to the surface of the array. When a sample containing genetic material (DNA or RNA) is introduced, the probes on the array will bind to the complementary nucleotide chain by a process called hybridization. With suitable marking, usually a fluorescent chemical, the intensity of the binding of the different probes can be measured.

The expression level as measured by this intensity (actually derived by processing an image of the entire array) is taken to indicate the amount of transcription of the corresponding genes in the biological system from which the sample was extracted. In turn, this is taken to indicate the protein concentration. A further complication is that expression levels can be estimated as ratios of two measurements, for example from two samples under different conditions.

2.3 Modelling cellular systems

As noted above, the advent of microarrays and other “high-throughput” forms of technology for biological analyses is leading to a shift of approach in experimentation. In the past a typical focus would be to test hypotheses applying to a single gene or gene product at a time. A picture of the underlying system would have to be built up step-by-step. Now it is possible to test all of the genes for their activity as part of entire biological systems. In terms of computational biology, this leads ultimately to the modelling *in silico* of cells or even complete "virtual organisms" [5, 10].

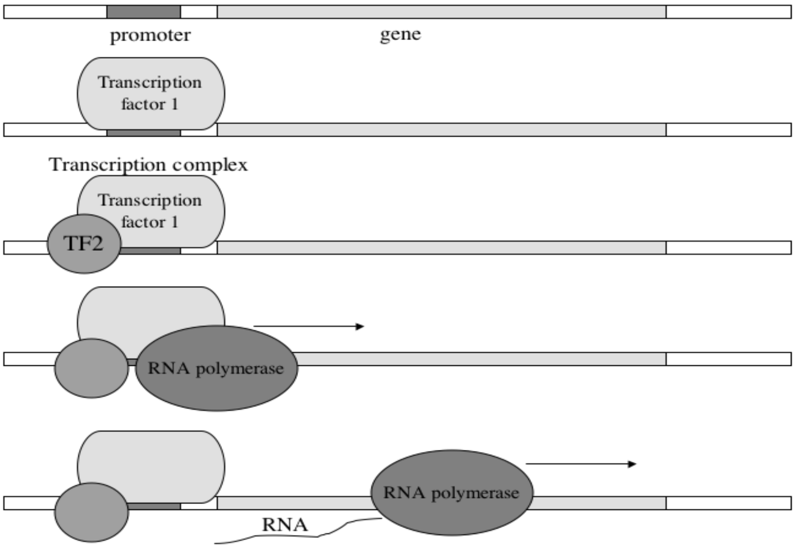


Fig. 2. Transcription factors bind to promoter regions singly or as a complex of several proteins to initiate transcription of a gene.

It is possible to separate the goals of simulation and identification in modelling. By simulation is meant running a computational model of some physical system, such as a cell or an oil refinery, with some given inputs to generate some behaviour of interest, such as a prediction of the chance of an explosion, or the response to a particular drug. In contrast identification is in some sense the inverse of simulation. It is less common and usually much harder; given some data about the behaviour of some physical system, e.g. inputs and outputs, generate a model to explain this behaviour and accurately predict other new behaviours of interest.

However, at the systems level such models become extremely complex. Both simulation and identification are required. Simulation models the entire system to generate predictions for empirical testing. Identification enables the automation of model construction given empirical data from the system. The domain specialist's understanding of the system is thus encoded in the model. In order to ensure confidence in the model, the formalism must be adequate with respect to both simulation and identification.

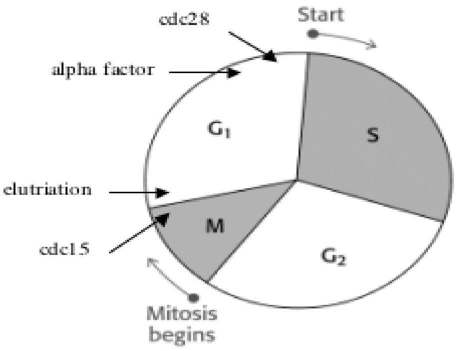


Fig. 3. Schematic diagram of the yeast cell cycle (modified from [3]). Actively dividing yeast cells must undergo DNA synthesis/replication (S phase) before mitosis/cell division (M phase). The S and M phases are separated by two "gap" phases (G1 and G2). Four different methods (cdc28, cdc15, alpha factor and elutriation) were used to arrest the cell cycle and synchronise yeast cell cultures at specific points of the cell cycle. The cells were then released from arrest and mRNA levels were measured at multiple time points for each group of synchronised cells [17]. The time-series data used in the present study was generated by sampling mRNA levels every 10 minutes for 300 minutes after cdc15 synchronisation.

Take an organism like E. coli where considerable knowledge has been accumulated on its molecular biology. This knowledge can be expressed in the form of a metabolic network, i.e. a labelled graph, where the vertices represent chemical compounds and the edges biochemical reactions. At a more abstract level, with a knowledge of the complete genome of E. coli, and data on proteins which have an effect on, say, the transcription of other proteins, we again construct a graphical representation of this biological system. In both cases, the dynamic nature of the underlying systems to be modelled leads to graphs being a natural formalism for the modelling task [9].

2.4 Learning system models

The task of inferring complete models of biological systems from data becomes overwhelming in the sample complexity due to the number of parameters. To deal with this problem a number of strategies have been adopted, such as restricting the class of models which can be learned [11], learning features common to sets of models [6] and learning with background knowledge containing known pathways [4]. In this paper we adopt an approach developed for learning to control dynamical systems, such as flight simulators, known as behavioural cloning.

Behavioural cloning is a machine learning technique which has been successfully used to construct control systems in a number of domains [12, 15, 18]. Clones are built by recording the performance of a skilled human operator and then running an induction algorithm over the traces of the behaviour. The most basic form of behavioural cloning results in a set of situation-action rules that map the current state of the process being controlled to a set of actions that achieve some desired goal.

In [1] this was generalised to include a method of learning goals, or reference values, which hold in a particular context, in addition to causal effects which result from applying certain control actions. This allows learning feedback control models from data. Recently this approach has been used to learn PID control rules directly from data [8].

Behavioural cloning is an appropriate framework for learning from microarray data since the cells from which the data are obtained can be viewed as dynamical systems under self-regulatory control. This is the same situation which holds when learning from behavioural traces in which one agent controls a plant or system. In the case of a human pilot controlling a flight simulator, the agent is external to the system being controlled. However, viewing the genome as in some sense the controller of the cellular system, clearly the controller is integral to the environment.

Experience with machine learning applications to behavioural cloning has demonstrated some advantages when dealing with data from complex systems. First, in behavioural cloning the task of controlling the system is broken down by selecting target variables to be controlled. Predictive models are learned for the behaviour of these variables. Second, the models are then embedded in agents dedicated to the control of a particular variable. Combining the agents results in a strategy for the successful control of the overall system. These points should enable modelling the combined activity of transcriptional regulation in the cell for a given set of experimental data. Third, using symbolic machine learning methods has the advantage that models mapping states to actions can be readable, allowing a degree of user insight into the dynamics of system control. This property is an advantage for biologists wishing to avoid "black-box" models.

3 Learning from microarray data

In this paper we adopt the behavioural cloning approach to learning causal dependencies within a complex system, namely the relationships in expression levels detected in a microarray experiment.

3.1 Learning models for individual genes

The approach taken is to decompose the problem of learning models of systems behaviour into learning models of individual "agent" behaviour given that of other relevant agents. In this paper we consider as relevant agents all genes in the genome, although in fact there will only be data on those in the transcriptome under any given set of experimental conditions.

```
YAL002W <= 0.015 :
|   YBR002C <= 0.015 : YDR054C = 0.368
|   YBR002C >  0.015 : YDR054C = 0.772
YAL002W >  0.015 :
|   YAL041W <= -0.035 : YDR054C = -0.392
|   YAL041W >  -0.035 : YDR054C = -1.04
```

YAL002W (VPS8): membrane-associated hydrophilic protein which contains a C-terminal cysteine-rich region that conforms to the H2 variant of the RING finger Zn2+ binding motif
YBR002C (RER2): cis-prenyltransferase
YAL041W (CDC24): guanine nucleotide exchange factor (a.k.a. GDP-release factor) for cdc42

Fig. 4. Regression tree for YDR054C (CDC34): E2 ubiquitin-conjugating enzyme.

Following from recent work on behavioural cloning [8] we select model trees as our representation. Tree-structured classifiers are efficient algorithms with a strong but effective bias, namely to prefer small trees. In the microarray application this means preferring to minimise the number of genes in a tree.

Regression or model trees are suitable for numerical prediction. This means there is no pre-discretisation required. They give a piecewise linear approximation to the unknown gene expression function. This means that non-linear regulation effects can be modelled. Since this approach is intended as a tool for exploratory data analysis, using tree-structured models is appropriate since they amenable to inspection and straightforward interpretation, i.e., not a "black-box".

3.2 An experiment on yeast expression data

The method adopted was to select a subset of genes of interest and try to predict their expression from the expression levels of the other genes.

A benchmark time series data set [17] and subset of genes of interest was selected [16]. The sample was taken from asynchronous cultures of the same cells growing exponentially at the same temperature in the same medium. A diagram of the cell-cycle is in Figure 3.

This data set contains measurements from four different experiments. Each is referred to by the name of the method used to synchronise the cycles of all cells in the experiment. We used the CDC15 dataset which has the largest number of data points. The remaining datasets will be used as part of future work.

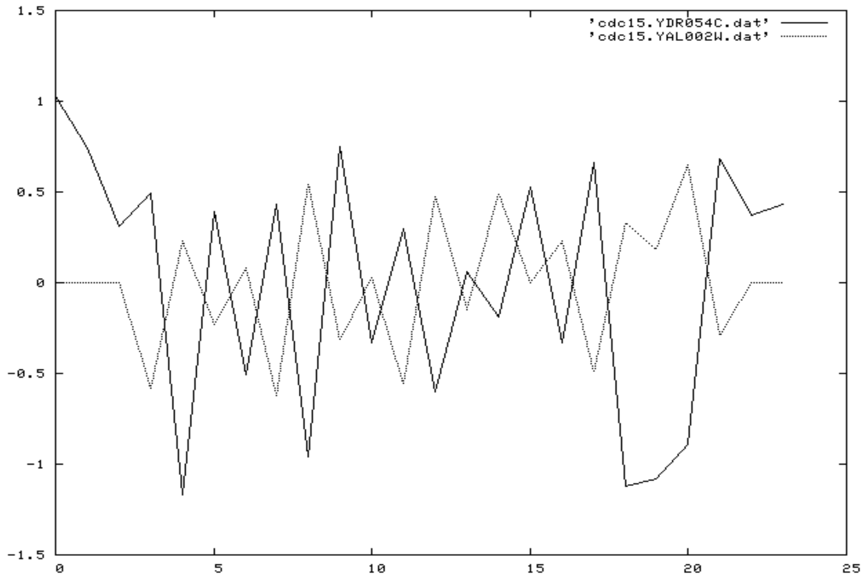


Fig. 5. The inverse relationship between target gene YDR054C (solid line) and YAL002W (dotted line). This pattern is captured by the top-level of the tree in Figure 4.

The microarray measurements were background corrected signal log ratios (see [17] for details), although this is not the only type of microarray measurement to which our methods are applicable. However, proper normalisation of the data should be undertaken to make measurements compatible over samples.

The approach taken requires selection of a number of genes as *target* genes for which trees are to be constructed.

For each gene [4] the following method was used to prepare the training data. Let X be a gene-expression matrix, and $x_{i,j}$ the expression level of gene i in sample j. For time series experiments, as in this case, each sample is taken at a separate time point. However, the approach is also applicable to non-time series data. A single example in the training set has the form

$$x_{i,j} \leftarrow \bigwedge_{k \neq i} x_{k,j}$$

[4] Standard ORF identifiers are used.

That is, the expression level of the target gene i in sample j depends on the conjunction of the expression levels of all other genes in sample j.

Examples in this format consist of an output variable (the expression level of the target gene) and a set of input variables (the expression levels of the remaining genes). An example in this format is generated for each sample in the experiment. A different set of examples if generated for each of the selected target genes.

Following the framework of [16] a set of twenty target genes selected known to be involved in cell cycle regulation was selected. It is worth noting that the whole genome could be taken as the set of target genes, although this would obviously generate a large number of trees.

For each of the target genes a tree-structured classifier was induced. We generated both regression trees and model trees [14] using the Weka machine learning toolkit [19]. Regression and model trees fit the data as decision trees but with a continuous “class” or output variable at the leaves of the tree. They differ as follows. Regression trees contain a single value at each leaf node, typically the mean of the output variable for all examples in that leaf. Model trees contain a linear regression model at each leaf node, fitted to the examples in that leaf.

Decision trees were also generated, using the binary classification ‘up-’ or ‘down-regulated’. Default parameter setting were used throughout. All runs were of 10-fold cross-validation on the entire training set.

3.3 Results and discussion

Correlation of actual and predicted values for each 10-fold cross-validation were obtained for model and regression trees. Predictive accuracy (1 - classification error) for decision trees was also obtained. These results are summarised in Table 1.

The results in terms of predictive accuracy are quite mixed. Some of the predictions appear to be quite accurate. These may also be showing interesting causal relationships, although this has yet to be evaluated. For other target genes, it is unlikely that any significant regulatory relationships have been uncovered.

Model trees seem to be the most accurate method. Decision trees are the most compact, often with only a single node. Regression trees are positioned in the middle; without the linear regression models in the leaves they are more compact than the model trees, and they contain more dependencies between the target and other genes than the decision trees. However, the data set is very small, particularly considering the number of genes. Further experiments with more data are necessary, since the methods may well be overfitting.

In terms of exploratory data analysis, the results are encouraging. Although we have not been able to verify any of the relationships appearing in the trees, some of the interactions are plausible. More work is needed for this.

As expected, the tree methods seem to be picking out a small number of genes for each target gene to predict its expression level. An example is the regression tree in Figure 4. This expresses a simple putative regulatory relationship which can be summarised as follows. If the gene YAL002W is down-regulated then

Table 1. Cross-validation results for target gene prediction ($N = 20$).

Decision Trees		Regression Trees		Model Trees	
Accuracy	No. of trees	Correlation	No. of trees	Correlation	No. of trees
$\geq$ 90%	3	$\geq$ 0.9	0	$\geq$ 0.9	1
$\geq$ 80%	4	$\geq$ 0.8	2	$\geq$ 0.8	2
$\geq$ 70%	12	$\geq$ 0.7	5	$\geq$ 0.7	8
$\geq$ 60%	15	$\geq$ 0.6	8	$\geq$ 0.6	11
$\geq$ 50%	20	$\geq$ 0.5	10	$\geq$ 0.5	13
		$\geq$ 0.4	16	$\geq$ 0.4	17
		$\geq$ 0.3	16	$\geq$ 0.3	17
		$\geq$ 0.2	16	$\geq$ 0.2	18
		$\geq$ 0.1	17	$\geq$ 0.1	19
		$\geq$ 0.0	18	$\geq$ 0.0	19
		...	...	...	...
		$\geq$ -0.3	20	$\geq$ -0.3	20
Mean = 72.29%		Mean = 0.46		Mean = 0.57	

(in general) the target gene YDR054C is up-regulated, and vice-versa. There is a further level of refinement, however, which is that gene YBR002C will up-regulate YDR054C if it is itself up-regulated; also gene YAL041W will down-regulate gene YDR054C if it is itself up-regulated. This example illustrates how positive and negative conditional dependencies can be encapsulated in a simple tree. The top level inverse regulatory relationship between genes YAL002W and YDR054C is shown by plotting the raw data in Figure 5.

A combination of using tree representations and simple plots provides a source of information on the behaviour of the underlying system. This would not be the case with simply clustering genes.

An additional feature of the approach is that model trees for each of the target genes can be combined to provide a simulation of the subsystem defined by those genes, as outlined in Section 2.4. Causal dependencies between target genes occur when one target gene appears as an attribute in the tree for another target gene. This situation is shown in Figure 6.

In our experiment this was observed for the gene SKP1 which was included as an attribute in regression and model trees for target gene SWI4. It is likely that other such dependencies would be uncovered with a larger set of target genes. This will be investigated as part of further work.

3.4 Related work

There has been a significant amount of work on applications of machine learning to microarray data (see [13] for an overview). Many applications are of a preliminary nature. It is not yet apparent that any single technique has a clear advantage on microarray data.

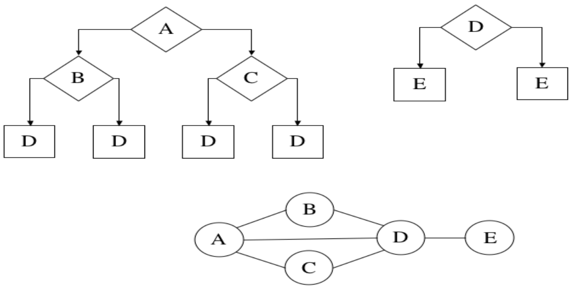


Fig. 6. The target gene D of the tree on the left appears as an attribute in the tree for target gene E on the right. Trees combined in a network showing only gene interactions.

The most common technique is still clustering, such as hierarchical clustering. With appropriate transformations of the data, such as a Fourier-type analysis to detect periodicities in time series data [17], clustering can be extremely useful. However, it is still restricted to grouping together similar examples in terms of a pre-defined similarity measure. For example, the disjunctive structure of context-dependent relationships found in tree-structured classifiers evident in Figure 4 may be a better fit for certain regulatory interactions. Although more complex distance functions may be defined for clustering, this places a burden on the user, and it may be difficult to know ahead of time what the “correct” distance function might be.

A well-known problem is dealing with the large number of genes in microarray data, particularly since the number of samples is usually small. As costs reduce over time, data sets may include many more samples. However, the main problem to be solved by machine learning algorithms on such data is to select the genes with important relationships and eliminate the redundant genes from the model.

In learning Bayes nets from expression data [6] this selection is on the basis of a statistical score which ranks genes for inclusion in the network. At each step in an iterative procedure the number of possible genes which can be added to the network is limited to a threshold $k \ll n$, where n is the number of genes.

Our work is more closely related to that of [16] in which decision-tree learning was applied to expression data for a number of selected target genes. Here the problem is similar to that of learning Bayes nets; for a given gene, how to select a subset of genes from the entire genome to be included in the tree. This is achieved by the tree-construction algorithm, which greedily selects a gene for inclusion in the tree by ranking each candidate in terms of its relationship to the target gene (by information gain, or standard deviation reduction, etc.) without the requirement of a threshold k.

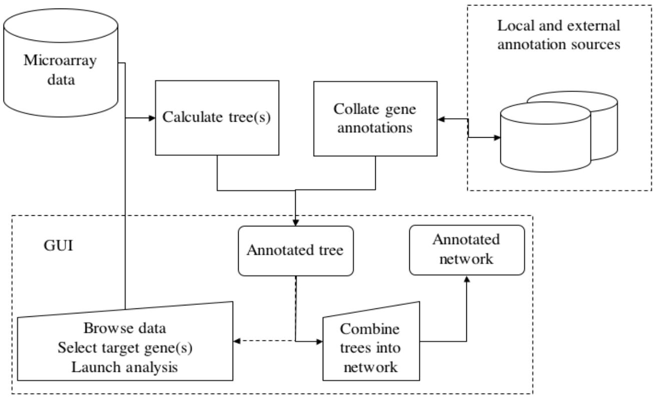


Fig. 7. Bioinformatics tool to support biologists in exploratory microarray data analysis. Annotation is used to extend the representation and formulate new learning tasks.

In contrast to [16] we avoid pre-discretisation of the expression levels of the genes by using regression and model trees. Thus the learned model includes quantitative values for the output variable, which can be important in showing the relation between expression levels at different nodes of a tree.

Lastly, it is interesting to note that with such tree-based methods the set of target genes could be *all* of the genes in the genome. Although this would lead to a large number of trees being learned, this could be done in parallel. Since tree construction is efficient, this may not require excessive computation. We may assume therefore that the approach may be quite scalable (given suitable parallel processing resources), although this has not been tested. However, the question of interpretability for such a large set of models is likely to become much more important.

4 Bioinformatics tools for biologists

A major advantage of our approach over currently used clustering methods is the extraction of explicit rules from the data, that can be further investigated and verified by domain experts. The techniques described in this paper can therefore be incorporated into a tool for biologists to explore the results from microarray experiments in an interactive fashion. We are currently developing such a tool in collaboration with a group of experimental biologists. The goal is to provide the user with an easy way to select target genes, launch analyses, and identify “interesting” relationships in the resulting trees.

In order to facilitate this exploration, the user interface should display not only the relationships in terms of gene names, but also annotate these results with relevant known attributes for the genes. These attributes include attributes of the gene itself (location on the chromosomes, relevant patterns in the surrounding DNA sequences) as well as attributes of its encoded protein (functional category, cellular localisation, known interaction partners etc). Combining the relationships predicted by the data mining techniques with an integrated view of these attributes should greatly facilitate the examination of the results and their evaluation and exploration by domain experts.

The exploration tool (see Figure 7) therefore incorporates a series of agents for retrieving the relevant attributes from various Internet data sources as required, and integrating these attributes with the data mining output. The exploration tool should also allow the user to combine multiple classifiers to construct a gene network, using both computational predictions and their own domain expertise. The exploration tool (see Figure 7) should include the following components:

- Database for storing microarray data
- Data mining tools
- Graphical user interface for
- Browsing through the microarray experiments
- Selecting target genes
- Launching analyses
- Displaying "annotated" trees
- Combining trees into networks
- Agents for querying local and external annotation sources and collating and formatting the retrieved annotations

The exploration tool should allow the domain experts to make immediate use domain experts to make immediate use of the classification methods for generating scientific hypotheses that can be validated experimentally. It will also allow a direct, subjective evaluation of the predictions by the domain experts. Further developments of this tool can include the automatic capture of the biologist's patterns of use, which could be used as the basis for the development of a more automated expert system for exploring microarray data.

5 Conclusions and further work

This is preliminary work and it is too soon to draw firm conclusions. Nevertheless there seem to be some promising aspects to our initial results. First, the approach of treating the problem as one of learning control rules for an unknown system is appealing. This allows a simple decomposition of the overall system, which makes learning models more efficient. These individual predictive models can be combined for system-level behaviour. The approach has been extensively investigated in the area of behavioural cloning, and this has resulted in many refinements such as learning feedback control rules, learning reference values for steady-state systems and a wide range of representations and learning methods.

Second, the use of symbolic machine learning methods which can fit quite complex, non-linear functions using simple structures has many advantages. The models are not black-boxes, which is important for biologists. They can be used for prediction and explanation. Overall, they have much more to offer than clustering, which has been the most commonly used method for applications to microarray data.

Third, as more information sources are available to include in training sets for model construction, both in the form of additional microarray data and other data such as sequence and interaction data, it should be straightforward to extend this approach. It is likely that more interactive stages in the learning will help to guide this process.

On the other side of the balance sheet there are a number of drawbacks. There is currently no way to obtain estimates of the statistical significance of the models learned by our methods. Although ultimately biological significance is more important than statistical significance, and it is well-known that statistical significance does not always imply biological significance, and vice versa, nonetheless it is important for data mining and machine learning practitioners to address this requirement. Possible directions include both theoretical developments based on the algorithms used, and empirical methods of significance estimation.

Much more work needs to be done to apply these approaches to more data. This includes: running these methods on many more target genes in a data set and comparing the output models; testing the predictive accuracy of the models on several data sets; applying the methods on microarray datasets other than the benchmark sets; validation of the predictions in terms of biological significance.

In summary, we fully expect the challenges of microarray data mining to lead to extensions to or new developments of machine learning methods, and to greatly improve our understanding of what is required for them to be successful on these large-scale biological data sets.

Acknowledgements. Thanks to Claude Sammut, Andrew Isaac, Ian Dawes, Geoff Kornfeld and Rohan Williams for helpful discussions on these topics.

References

1. M. Bain and C. Sammut. A Framework for Behavioural Cloning. In K. Furukawa, D. Michie, and S. Muggleton, editors, *Machine Intelligence 15*, pages 103–129. Clarendon Press, Oxford, 1999.
2. P. Baldi and W. Hatfield. *DNA Microarrays and Gene Expression.* Cambridge University Press, Cambridge, UK, 2002.
3. J.M. Berg, J.L. Tymoczo, and L. Stryer. *Biochemistry (5th Edition).* W. H. Freeman and Company, New York, NY, 2002.

4. C. Bryant, S. Muggleton, S. Oliver, D. Kell, P. Reiser, and R. King. Combining Inductive Logic Programming, Active Learning, and Robotics to Discover the Function of Genes. *Linkoping Electronic Articles in Computer and Information Science*, 6(12), 2001.
5. D. Endy and R. Brent. Modelling cellular behaviour. *Nature*, 409:391–395, 2001.
6. N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, 7:601 – 620, 2000.
7. D. Greenbaum, N. Luscombe, R. Jansen, J. Qian, and M. Gerstein. Interrelating Different Types of Genomic Data, from Proteome to Secretome: 'Oming in on Function. *Genome Research*, 11:1463–1468, 2001.
8. A. Isaac and C. Sammut. Goal-directed learning to fly. In *Proc. of the International Conference on Machine Learning*, pages 258–265, Los Altos, 2003. Morgan Kaufmann.
9. P. Karp. Pathway Databases: A Case Study in Computational Symbolic Theories. *Science*, 293:2040, 2001.
10. H. Kitano. Computational systems biology. *Nature*, 420:206–210, 2002.
11. S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures. In *Third Pacific Symposium on Biocomputing*, pages 18 – 29, 1998.
12. D. Michie, M. Bain, and J. Hayes-Michie. Cognitive models from subcognitive skills. In M. Grimble, S. McGhee, and P. Mowforth, editors, *Knowledge-based Systems in Industrial Control*. Peter Peregrinus, IEEE Press, 1990.
13. M. Molla, M. Waddell, D. Page, and J. Shavlik. Using Machine Learning to Design and Interpret Gene-Expression Microarrays. *AI Magazine (special issue on Bioinformatics)*, 2003.
14. J. R. Quinlan. Learning with Continuous Classes. In *Proc. 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, 1992.
15. C. Sammut, S. Hurst, D. Kedzier, and D. Michie. Learning to fly. In D. Sleeman and P. Edwards, editors, *Proceedings of the Ninth International Conference on Machine Learning*, San Mateo, CA, 1992. Morgan Kaufmann.
16. L. Soinov, M. Krestyaninova, and A. Brazma. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biology*, 4(1), 2003.
17. P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.
18. T. Urbančič and I. Bratko. Reconstructing human skill with machine learning. In *Proceedings of the 11th European Conference on Artificial Intelligence*. John Wiley & Sons, 1994.
19. I. Witten and E. Frank. *Data Mining*. Morgan Kaufmann, San Francisco, California, 2000.

An Mining Approach Using MOUCLAS Patterns

Yalei Hao[1] **Markus Stumptner**[1] **Gerald Quirchmayr**[1] [2]

[1] *Advanced Computing Research Centre, University of South Australia, SA5095, Australia*
Yalei.Hao@postgrads.unisa.edu.au, mst@cs.unisa.edu.au, Gerald.Quirchmayr@unisa.edu.au

[2] *Institut für Informatik und Wirtschaftsinformatik, Universität Wien, Liebiggasse 4, A-1010 Wien, Austria*

Abstract. The integration of data mining techniques with classification problems relies on the success in the mining of interesting knowledge patterns during the learning phase of the classifiers. This paper proposes a new kind of patterns for the classification over quantitative data in high dimensional database, which is called *MOUCLAS* (MOUntain function based CLASsification) Patterns, based on the concept of the fuzzy set membership function which gives the new approach a solid mathematical foundation and compact mathematical description of classifiers, and integrating classification and clustering and association rules mining to identify interesting knowledge in the databases. The framework of the paper is formed by concentrating on three major issues of classification: the definition of the new *MOUCLAS* patterns, the algorithm for the discovery of the *MOUCLAS* patterns, and the construction of new classifiers.

1 Introduction and Motivation

An increasing part of research in the data mining communities focuses on the task of classification, in addition to three other tasks: generation of association rules, clustering, and concept description. Data mining based classification aims to build accurate and efficient classifiers not only on small data sets but more importantly also on large and high dimensional data sets, while the widely used traditional statistical data analysis techniques are not sufficiently powerful for this task [12], [13].

The use of association rules to discover dependencies among data [2] has been extensively studied in both the database and data mining communities for a long time, with major studies in [3], [4], [5], [15], [17], [21], [23], [25], [27], [29], [30], and [33]. With the development of new data mining techniques on association rules, new classification approaches based on concepts from association rule mining are emerging. These include such classifiers as ARCS [20], CBA [18], LB [22], CAEP [10], etc., which are different from the classic decision tree based classifier C4.5 [24] and k-nearest neighbor [6] in both the learning and testing phases.

ARCS [20] demonstrated the successful application of concepts of clustering for the purpose of classification. However, ARCS is limited to 2D-rules based classifiers of the format $A \wedge B \Rightarrow \text{Class}_i$, where A and B are two predicates. It uses the method of “Binning” to discretize the value of quantitative attributes. Consequently, the accuracy of ARCS is strongly related to the degree of discretization used. A non-grid-based technique [26] has been proposed to find quantitative association rules that can have more than two predicates in the antecedent. The authors noticed that the information loss caused by partitioning could not be ignored and have tried to employ a measure of partial completeness to quantify the information lost, but the measure is still constrained by the framework of binning. Though there are several excellent discretization algorithms [11], [8], a standard approach to discretization has not yet been developed. Different approaches could lead to different collections of large itemsets even with respect to the same support threshold in a given data set. ARCS and the non-grid-based technique lead to research question 1 being addressed: “Is it possible that an association rule based classifier can be developed for quantitative attributes by the concepts of clustering which can overcome the limitation caused by the discretization method? ” CBA [18] gives us an interesting indication that the idea of apriori property can be applied to a set of predicates (itemsets) for classification. Suppose an association rule based classifier in the form of $A_1 \wedge A_2 \wedge \ldots \wedge A_l \Rightarrow C_i$, where A_j (j=1, …, l) are predicate variables, C_i is the class label, the antecedent of the rule is a frequent itemset. This raises question 2: “If an association rule based classifier can be built based on the concept of clustering, is it possible that a link between CBA and ARCS can be found so that an association rule based classifier with any number of predicates in the antecedent can be setup by clustering? ”

The above research issues establish a challenge that comes within our research focus. In this paper, we present a new approach to the classification over quantitative data in high dimensional databases, called *MOUCLAS* (MOUntain function based CLASsification), based on the concept of the fuzzy set membership function. It aims at integrating the advantages of classification, clustering and association rules mining to identify interesting patterns in selected sample data sets.

2 Problem Statement

We now give a formal statement of the problem of *MOUCLAS* Patterns (called *MPs*) and introduce some definitions. The *MOUCLAS* algorithm, similar to ARCS, assumes that the initial association rules can be agglomerated into clustering regions, while obeying the anti-monotone rule constraint. Our proposed framework assumes that the training dataset *D* is a normal relational set, where transaction $d \in D$. Each transaction *d* is described by attributes A_j, $j = 1$ to *l*. The dimension of *D* is *l*, the number of attributes used in *D*. This allows us to describe a database in terms of volume and dimension. *D* can be classified into a set of known classes Y, $y \in Y$. The value of an attribute must be quantitative. In this work, we treat all the attributes uniformly. We can treat a transaction as a set of (attributes, value) pairs and a class label. We call each (attribute, value) pair an item. A set of items is simply called an itemset.

Since CBA indicates the feasibility of setting up a link between association rule and classification and ARCS proves that the idea of designing a classification pattern based on clustering can work effectively and efficiently, we design a *MOUCLAS* Pattern (so called *MP*) as an implication of the form:

$$Cluster(D)_t \rightarrow y,$$

where $Cluster(D)_t$ *is a cluster of* *D*, $t = 1$ to *m*, and *y* is a class label. The definitions of *frequency* and *accuracy* of *MOUCLAS* Patterns are defined as following: The *MP* satisfying minimum support is ***frequent***, where *MP* has support s if s% of the transactions in *D* belong to $Cluster(D)_t$ and are labeled with class *y*. The *MP* that satisfies a pre-specified minimum confidence is called ***accurate***, where *MP* has confidence c if c% of the transactions belonging to $Cluster(D)_t$ are labeled with class *y*.

Though framework of support – confidence is used in most of the applications of association rule mining, it may be misleading by identifying a rule A ⟹ B as interesting, even though the occurrence of A may not imply the occurrence of B. This requires a complementary framework for finding interesting relations. Correlation [15] is one of the most efficient interestingness measures other than support and confidence. Here we adopt the concept of reliability [34] to describe the correlation. The measure of reliability of the association rule A ⟹ B can be defined as:

$$\text{reliability} \quad \mathrm{R(A \Longrightarrow B)} = \left| \frac{P(A \wedge B)}{P(A)} - P(B) \right|$$

Since R is the difference between the conditional probability of B given A and the unconditional of B, it measures the effect of available information of A on the probability of the association rule. Correspondingly, the greater R is, the stronger *MOUCLAS* patterns are, which means the occurrence of $Cluster(D)_t$ more strongly implies the occurrence of *y*. Therefore, we can utilize reliability to further prune the selected *frequent and accurate and reliable MOUCLAS* patterns (*MPs*) to identify the truly interesting *MPs* and make the discovered *MPs* more understandable. The *MP* satisfying minimum reliability is ***reliable***, where *MP* has reliability defined by the above formula.

Given a set of transactions, *D*, the problems of *MOUCLAS* are to discover *MPs* that have support and confidence greater than the user-specified minimum support threshold (called *minsup*) [4], and minimum confidence threshold (call *minconf*) [4] and minimum reliability threshold (call *minR*) respectively, and to construct a classifier based upon *MPs*.

3 The *MOUCLAS* Algorithm

The classification technique, *MOUCLAS*, consists of two steps:

1. Discovery of *frequent*, *accurate* and *reliable MPs*.
2. Construction of a classifier, called *De-MP*, based on *MPs*.

The core of the first step in the *MOUCLAS* algorithm is to find all *cluster_rules* that have support above *minsup*. Let *C* denote the dataset *D* after dimensionality reduction processing. A *cluster_rule* represents a *MP*, namely a rule:

$$cluset \rightarrow y,$$

where *cluset* is a set of itemsets from a cluster $Cluster(C)_t$, *y* is a class label, $y \in Y$. The support count of the *cluset* (called *clusupCount*) is the number of transactions in *C* that belong to the *cluset*. The support count of the *cluster_rule* (called *cisupCount*) is the number of transactions in *D* that belong to the *cluset* and are labeled with class *y*. The *confidence* of a *cluster_rule* is (*cisupCount* / *clusupCount*) × 100%. The support count of the *class y* (called *clasupCount*) is the number of transactions in *C* that belong to the class *y*. The *support* of a *class* (called *clasup*) is (*clasupCount* / |C|) × 100%, where | *C* | is the size of the dataset *C*.

Given an *MP*, the *reliability* R can be defined as:

$$R(cluset \to y) = \left| \ (cisupCount \ / \ clusupCount) - (clasupCount \ / \ |C|) \ \right| \times 100\%$$

The traditional association rule mining only uses a single *minsup* in rule generation, which is inadequate for many practical datasets with uneven class frequency distributions. As a result, it may happen that the rules found for infrequent classes are insufficient and too many may be found for frequent classes, inducing useless or over-fitting rules, if the single *minsup* value is too high or too low. To overcome this drawback, we apply the theory of mining with multiple minimum supports [32] in the step of discovering the frequent MPs as following.

Suppose the total support is *t-minsup*, the different minimum class support for each class *y*, denoted as $minsup_i$ can be defined by the formula:

$$minsup_i = t\text{-}minsup \times \text{freqDistr}(y)$$

where, freqDistr(*y*) is the function of class distributions. *Cluster_rules* that satisfy $minsup_i$ are called *frequent cluster_rules*, while the rest are called *infrequent cluster_rules*. If the *confidence* is greater than *minconf*, we say the *MP* is *accurate*.

The first step of the *MOUCLAS* algorithm works in three sub-steps, by which the problem of discovering a set of *MPs* is solved:

Algorithm: Mining *frequent* and *accurate* and *reliable MOUCLAS* patterns (*MPs*)

Input: A training transaction database, *D*; minimum support threshold ($minsup_i$); minimum confidence threshold (*minconf*)

Output: A set of *frequent*, *accurate* and *reliable MOUCLAS* patterns (*MPs*)

Methods:

(1) Reduce the dimensionality of transactions *d*, which efficiently reduces the data size by removing irrelevant or redundant attributes (or dimensions) from the training data, and

(2) Identify the clusters of database *C* for all transactions *d* after dimensionality reduction on attributes A_j in database *C*, based on the Mountain function, which is a fuzzy set membership function, and specially capable of transforming quantitative values of attributes in transactions into linguistic terms, and

(3) Generate a set of *MPs* that are both *frequent*, *accurate* and *reliable*, namely, which satisfy the user-specified minimum support (called $minsup_i$), minimum confidence (called *minconf*) and minimum reliability (called *minR*) constraints.

In the first sub-step, we reduce the dimensionality of transactions in order to enhance the quality of data mining and decrease the computational cost of the *MOUCLAS* algorithm. Since, for attributes A_j, $j = 1$ to l in database, *D*, an exhaustive search for the optimal subset of attributes within 2^l possible subsets can be prohibitively expensive, especially in high dimensional databases, we use heuristic methods to reduce the search space. Such greedy methods are effective in practice, and include such techniques as stepwise forward selection, stepwise backward elimination, combination of forwards selection and backward elimination, etc. The first sub-step is particularly important when dealing with raw data sets. Detailed methods concerning dimensionality reduction can be found in [9], [19], [28], [16].

Fuzzy based clustering is performed in the second sub-step to find the clusters of quantitative data. The Mountain-climb technique proposed by R. R. Yager and D. P. Filev [31] employed the concept of a mountain function, a fuzzy set membership function, in determining cluster centers used to initialize a Neuro-Fuzzy system. The substractive clustering technique [7] was defined as an improvement of Mountain-climb clustering. A similar approach is provided by the DENCLUE algorithm [14], which is especially efficient for clustering on high dimensional databases with noise. The techniques of Mountain-climb clustering, Substractive clustering and Denclue provide an effective way of dealing with quantitative attributes by mountain functions (or influence functions), which has a solid mathematical foundation and compact mathematical description and is totally different from the traditional processing method of binning. It offers us an opportunity of mining the patterns of data from an innovative angle. As a result, question 1 presented in the introduction can now be favorably answered.

The observation that, a region which is dense in a particular subspace must create dense regions when projected onto lower dimensional subspaces, has been proved by R. Agrawal and his research cooperators in CLIQUE [1]. In other words, the observation follows the concepts of the apriori property. Hence, we may employ prior knowledge of items in the search space based on the property so that portions of the space can be pruned. The successful performance of CLIQUE has again proved the feasibility of applying the concept of apriori property to clustering. It brings us a step further towards the solution of problem 2, that is, if the initial association rules can be agglomerated into clustering regions, just like the condition in ARCS, we may be able to design a new classifier for the purpose of classification, which confines its search for the classifier to the cluster of dense units of high dimensional space. The answer to question 2 can contribute to the third sub-step of the *MOUCLAS* algorithm, i.e., the forming of the antecedents of *cluster_rules*, with any number of predicates in the antecedent. In the third sub-step, we identify the candidate *cluster_rules* which are actually *frequent* and *accurate* and *reliable*. From this set of *frequent* and *accurate* and *reliable cluster_rules*, we produce a set of *MPs*.

Let I be the set of all items in D, C be the dataset D after dimensionality reduction, where transaction $d \in C$ contains $X \subseteq I$, a k-itemset. Let E denote the set of candidates of cluster_rules, where $e \in E$, *and* F denote the set of frequent cluster_rules. The first step of the *MOUCLAS* algorithm is given in Figure 1 as follows.

```
$X$ = reduceDim ($I$); // reduce the dimensionality on the set of all items $I$ of in $D$
$Cluster(C)_t$ = genCluster ($C$); // identify the complete clusters of $C$
for each $Cluster(C)_t$ do
    $E$ = genClusterrules(cluset, class); // generate a set of candidate cluster_rules
    for each transaction $d \in C$ do
        $E_d$ = genSubClusterrules ($E, d$); // find all the cluster_rules in $E$ whose cluset are supported by $d$
        for each $e \in E_d$ do
            e. clusupCount++; // accumulate the clusupCount of the cluset of cluster_rule e
            if $d$.class = $e$.class then e.cisupCount++ // accumulate the cisupCount of cluster_rule e supported by $d$
        end
    end
    $F = \{e \in E \mid e.cisupCount \geq minsup_i\}$; // construct the set of frequent cluster_rules
    $MP$ = genRules ($F$); //generate $MP$ using the genRules function by minconf and minR
end
$MPs = \cup MP$; // discover the final set of $MPs$
```

Figure 1: The First Step of the *MOUCLAS* Algorithm

The task of the second step in *MOUCLAS* algorithm is to use a heuristic method to generate a classifier, named *De-MP*, where the discovered *MPs* can cover D and are organized according to a decreasing precedence based on their confidence and support. Suppose R be the set of *frequent*, *accurate* and *reliable MPs* which are generated in the past step, and $MP_{default\_class}$ *denotes* the default class, which has the lowest precedence. We can then present the *De-MP* classifier in the form of

$$<MP_1, MP_2, \ldots, MP_n, MP_{default\_class}>,$$

where $MP_i \in R$, $i = 1$ to n, $MP_a \succ MP_b$ if $n \geq b > a \geq 1$ *and* $a, b \in i$, $C \subseteq \cup$ *cluset of* MP_i,.

The second step of the *MOUCLAS* algorithm also consists of three sub-steps, by which the *De-MP* classifier is formed:

Algorithm: Constructing *De-MP* Classifier

Input: A training database after dimensionality reduction, C; The set of *frequent and accurate and reliable MOUCLAS* patterns (*MPs*)

Output: *De-MP* Classifier

Methods:

(1) Identify the order of all discovered *MPs* based on the definition of precedence and sequence them according to decreasing precedence order.

(2) Determine possible *MPs* for *De-MP* classifier from R following the descending sequence of *MPs*.

(3) Discard the *MPs* which cannot contribute to the improvement of the accuracy of the *De-MP* classifier and keep the final set of *MPs* to construct the *De-MP* classifier.

In the first sub-step, the *MPs* are sorted in descending order, which has the training transactions surely covered by the *MPs* with the highest precedence when possible in the next sub-step. The sort of the whole set of *MPs* is performed following the definition of *precedence* as in CBA:

Given two *MPs*, we say that MP_a has a higher precedence than MP_b, denoted as $MP_a \succ MP_b$, if $\forall$ MP_a, $MP_b \in MPs$, it holds that: the confidence of MP_a is greater than that of MP_b, or if their confidences are the same, but the support of MP_a is greater than that of MP_b, or if both the confidences and supports of MP_a and MP_b are the same, but MP_a is generated earlier than MP_b.

In the second sub-step, we test the *MPs* following decreasing precedence and stop the sub-step when there is no rule or no training transaction. For each *MP*, we scan *C* to find those transactions satisfying the cluset of the *MP*. If the *MP* can correctly classify one transaction, we store it in a set denoted as *L*. Those transactions satisfying the cluset of the *MP* will be removed from *C* at each pass. Each transaction can be identified by a unique ID. The next pass will be performed on the remaining data. A default class is defined at each scan, which is the majority class in the remaining data. At the end of each pass, the total number of errors that are made by the current *L* and the default class are also stored. When there is no rule or no training transaction left, we terminate this sub-step. After this sub-step, every *MP* in *L can* correctly classify at least one training transaction in *C*.

In the third sub-step, though we would like to find as many *MPs* as possible to give good coverage of the training transactions in the second sub-step, we prefer strong *MPs* which have relatively high support and confidence, due to their characteristics of corresponding to larger coverage and stronger differentiating power. Meanwhile, we hope that the *De-MP* classifier, consisting of a combination of strong *MPs*, has a relatively smaller number of classification errors, because of greedy strategy. In addition, the reduction of *MPs* can increase the understandability of the classifier. Therefore, in this sub-step, we identify the first *MP* with the least number of errors in *L* and discard all the MPs after it because these *MPs* produce more errors. The undiscarded *MPs* and the default class corresponding to the first *MP* with the least number of errors in *L* form our *De-MP* classifier.

The second step of the *MOUCLAS* algorithm is shown in Figure 2.

```
$R$ = sort($R$); // sort MPs based on their precedence
for each $MP \in R$ in sequence do
    $temp = \emptyset$ ;
    for each transaction $d \in C$ do
        if $d$ satisfies the cluset of $MP$ then
            store $d$.ID in $temp$;
            if $MP$ correctly classifies $d$ then
                insert $MP$ at the end of $L$;
        delete the transaction who has ID in $temp$ from $C$;
        selecting a default class for the current $L$; // determine the default class based on majority class of remaining transactions in $C$
    end
    compute the total number of errors of $L$; // compute the total number of errors that are made by the current $L$ and the default class
end
Find the first MP in $L$ with the lowest total number of errors and discard all the $MPs$ after the $MP$ in $L$;
Add the default class associated with the above mentioned first MP to end of $L$;
$De\text{-}MP$ classifier = $L$
```

Figure 2: The Second Step of the *MOUCLAS* Algorithm

In the testing phase, when we classify a new transaction, the first *MP* in *De-MP* satisfying the transaction is used to classify it. In *De-MP* classifier, *default_class*, having the lowest precedence, is used to specify a default class for any new sample that is not satisfied by any other *MPs* as in C4.5[24], CBA[18].

4 Example of *MOUCLAS* Application

Oil/gas formation identification is a vital task in the petroleum industry, where the petroleum database contains such records (or attributes) as seismic data, various types of well logging data (e.g. GR, DEN, CNL, Resistivity, BCSL, etc.), and physical propertiy data (e.g. porosity, permeability, etc.), whose values are all quantitative. An illustration of using well logging date for purpose of oil/gas formation identification is illustrated in figure 3. One transaction of the database can be treated as a set of the items corresponding to the same depth and a class label (oil/gas formation or not). A hypothetically useful *MP* may suggest a relation between petroleum data at a certain depth and the class label of oil/gas formation. In this sense, a selected set of such *MPs* can be a useful guide to petroleum engineers to identify possible drilling targets and their depth and thickness at the stage of exploration and exploitation.

The notable advantage of *MOUCLAS* over more traditional processing techniques such as seismic inversion is that a physical model to describe the relationship between the seismic data and the property of interest is not needed; nor is an very precise understanding of the phases of the seismic data. From this point of view, *MOUCLAS* provides a complementary and useful technical approach towards the interpretation of petroleum data and benefits petroleum discovery.

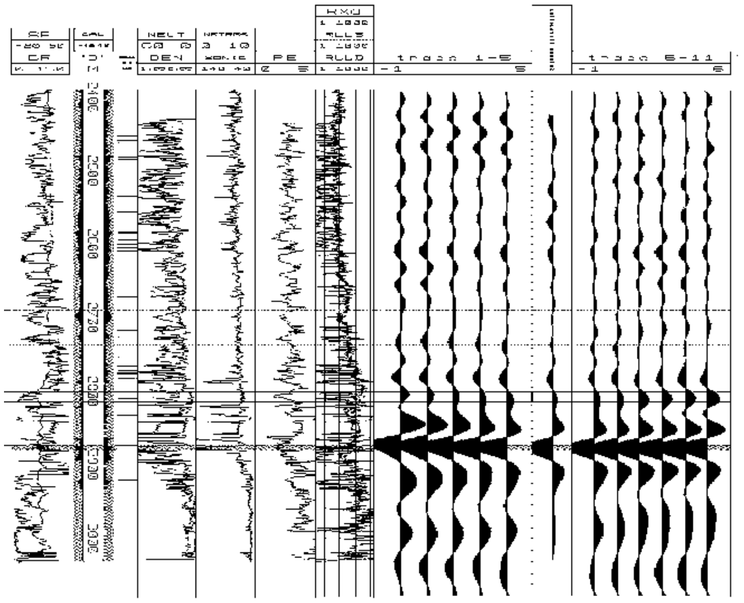


Figure 3: Quantitative Petroleum Data Suitable for *MOUCLAS* Mining
(note: the dashed lines indicate the location of oil formations)

5 Conclusion

We have introduced a new type of classification patterns, the *MOUCLAS* Pattern (*MP*), for quantitative data in high dimensional databases. We have also proposed an algorithm for discovering the interesting *MPs* and construct a new classifier called *De-MP*. As a hybrid of classification, clustering, and association rules mining, our approach may have several advantages which are that (1) it has a solid mathematical foundation and compact mathematical description of classifiers, (2) it does not require discretization, as opposed to other, otherwise quite similar methods such as ARCS, (3) it is robust when handling noisy or incomplete data in high dimensional data space, regardless of the database size, due to its grid-based characteristics, (4) it is not sensitive to the order of input items and it scales linearly with the size of the input.

6 Acknowledgement

This work was partially supported by the Australia-China Special Fund for Scientific and Technological Cooperation under grant CH030086. We would like to thank Dr. Dan Corbett for many fruitful discussions that helped to further refine our approach.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98. (1998)
2. Agrawal, R., Imielinski, T., & Swami, A. Mining association rules between sets of items in large databases. Proc. of the 1993 ACM-SIGMOD ACM Press. (1993) 207--216
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. Fast discovery of association rules. Advances in knowledge discovery and data mining. AAAI/MIT Press. (1996) 1-34
4. Agrawal, R., Srikant, R. Fast algorithms for mining association rules. Proc. of the 20th VLDB (1994) 487- 499
5. Bayardo, R. J. Efficiently mining long patterns from databases. Proc. of the 1998 ACM-SIGMOD. ACM Press. (1998) 85-93
6. Cover, T. M., & Hart, P. E. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13. (1967) 21-27
7. Chiu, S. L. Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy System, 2(3), (1994)

8. Dougherty, J., Kohavi, R., & Sahami, M. Supervised and unsupervised discretization of continuous features. Proc. of the Twelfth Int'l Conf. on Machine Learning pp. 94--202. Morgan Kaufmann. (1995)

9. Dong, G., & Li, J. Feature selection methods for classification. Intelligent Data Analysis: An International Journal, 1, (1997)

10. Dong, G., & Li, J. Efficient mining of emerging patterns: Discovering trends and differences. Proc. of the Fifth ACM SIGKDD. (1999)

11. Fayyad, U., & Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. Proc. of the 13th Int'l Conf. on Artificial Intelligence. Morgan Kaufmann. (1993) 1022--1029

12. Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. From data mining to knowledge discovery: An overview. Advances in knowledge discovery and data mining. AAAI/MIT Press. (1996) 1-34

13. Han, J., & M. Kamber. Data mining: concepts and techniques. Morgan Kaufmann Publishers. (2000)

14. A. Hinneburg and D. Keim. An efficient approach to clustering in large Multimedia dataset with noise. KDD'98, (1998) 58-65

15. Han, J., Pei, J., & Yin, Y. Mining frequent patterns without candidates generation. Proc. of the 2000 ACM-SIGMOD. ACM Press. (2000) 1-12

16. R. Kohavi and G. John. Wrappers for feature subset selection. Artificial Intelligence, (1997) 273-324

17. Lee, S. D., Cheung, D. W., & Kao, B. Is sampling useful in data mining? a case in the maintenance of discovered association rules. Data Mining and Knowledge Discovery, 2. (1998) 233-262.

18. B. Liu, W.Hsu, and Y.Ma. Integrating classification and association rule mining. KDD'98. (1998) 80-86

19. H. Liu and H. Motoda, editors. Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers, (1998)

20. B. Lent, A. Swami, and J. Widom. Clustering association rules. ICDE'97, (1997) 220-231

21. Mannila, H., Toivonen, H., & Verkamo, A. I. Efficient algorithms for discovering association rules. Proc. of AAAI'94 Workshop KDD'94. AAAI Press. (1994) 181-192.

22. Meretakis, D., & Wuthrich, B. Extending naive Bayes classifiers using long itemsets. Proc. of the Fifth ACM SIGKDD. ACM Press. (1999) 165-174

23. R.J. Miller and Y. Yang. Association rules over interval data. SIGMOD'97, (1997) 452-461

24. Quinlan, J. R. C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann. (1993)

25. Srikant, R., & Agrawal, R. Mining generalized association rules. Proc. of the 21st Int'l. Conf. on VLDB. Morgan Kaufmann. (1995) 407-419

26. R. Skikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIG-MOD'96, (1996) 1-12.

27. Savasere, A., Omiecinski, E., & Navathe, S. An efficient algorithm for mining association rules in large databases. Proc. of the 21st VLDB. Morgan Kaufmann. (1995) 432-443

28. W.Sarawagi and M. Stonebraker. On automatic feature selection. Int'l J. of Pattern Recognition and Artificial Intelligence, 2, (1988) 197-220.

29. Toivonen, H. Sampling large databases for association rules. Proc. of the 22rd VLDB. Bombay, India: Morgan Kaufmann. (1996) 134-145

30. Wijsen, J., & Meersman, R. On the complexity of mining quantitative association rules. Data Mining and Knowledge Discovery, 2, 1998 263-282

31. Yager, R. and D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, (1994) 209-219

32. Bing Liu, Wynne Hsu, Yiming Ma, "Mining Association Rules with Multiple Minimum Supports" Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99), August 15-18, San Diego, CA, USA (1999)

33. Zhang, T. Association rules. Proc. of the Fourth Pacific-Asia Conf. on Knowledge Discovery and Data Mining (2000) 245-256

34. Khalil M. Ahmed, Nagwa M. El-Makky, Yousry Taha: A note on "Beyond Market Baskets: Generalizing Association Rules to Correlations". In The Proceedings of SIGKDD Explorations Volume1, Issue 2, (2000) 46-48

From Rule Visualisation to Guided Knowledge Discovery

Aaron Ceglar, John F. Roddick, Carl H. Mooney and Paul Calder

School of Informatics and Engineering
Flinders University
PO Box 2100, Adelaide 5001, South Australia.
Email: {aaron.ceglar, john.roddick, carl.mooney,
paul.calder}@infoeng.flinders.edu.au

Abstract. As a result of the inability of computer systems to understand abstract concepts, data mining algorithms do not generally constrain the generation of rules to those that are of interest to the user adequately. Interactive knowledge discovery techniques aim to alleviate this problem by involving the user in the mining process, so that the user's broader understanding of abstract semantic concepts and domain knowledge can guide the discovery process, resulting in accelerated mining with improved results. At present this is done largely through data visualisation techniques and to a lesser degree through rule visualisation. This paper discusses some of the current research in interactive data mining research and argues that the next stage after rule visualisation is the interactive user manipulation of the mining process - Guided Knowledge Discovery. The paper also introduces two unique visualisation tools – the CARV hierarchical association rule mining visualisation tool and the INTEM sequential pattern rule visualisation tool.

Additional Keywords: Interactive data mining.

1 Introduction

Although computer based, the knowledge discovery process is human-centric because of its reliance upon the user's involvement in both mechanistic aspects such as data selection and preparation, and in aspects involving quantitative judgement such as analysis and interpretation of the results. Using current techniques, the user is involved within all stages of the discovery process, with the exception of the analysis[1] stage, which remains a 'black box'. This analysis stage uses data mining algorithms to explore a dataset and discover patterns or structures, which are influenced by user specified constraints and objective measures of interestingness. To date, data mining research has focused mainly upon heuristic correctness and efficiency. In interactive data mining, the process is extended to

[1] For convenience we use the CRISP nomenclature (Chapman, Kerber, Clinton, Khabaza, Reinartz & Wirth 1999)

investigate ways in which the user can become an integral part of the mining process. The need for user inclusion is based on the premise that the concept of interestingness is subjective, and cannot therefore be fully defined in heuristic terms. This suggests that by extending data mining algorithms to incorporate subjective measures of interestingness (through user participation), more useful results will be produced. The collaboration between computer and human will result in a symbiosis; the computer will provide processing power and storage facilities, and the user will contribute such capabilities as understanding and perception.

This paper investigates interaction techniques that allow the user to actively guide the knowledge discovery process, in effect overcoming the computer's inability to incorporate knowledge about intangible subjective measures such as domain knowledge and data semantics. In addition to producing more interesting results, guidance of the mining process implies that the algorithm can be dynamically constrained during processing (to reduce the breadth or depth of analysis), hence reducing both mining time and result set size.

There are two classes of data mining tasks: directed and undirected. Directed mining, also known as supervised learning or predictive analysis, refers to a group of methods that build a model based upon a set of data and make predictions about new items based on this model. Undirected mining, also known as unsupervised learning or explorative analysis, employs techniques that are used to discover patterns, unknown or theorised by the user. Interactive mining can only be applied to the explorative tasks such as clustering and association mining; as directed mining tasks such as classification and characterisation are guided through training sets of data.

This paper explores the techniques available for visualisation of and interaction with undirected knowledge discovery systems. Section 2 builds toward a discussion on interactive data mining by outlining the need for human participation within the knowledge discovery process. This is followed by a discussion of presentation paradigms in Section 3, highlighting both the strengths and weaknesses of textual and graphical methods. Section 4 contains a comprehensive taxonomy of undirected mining presentations, including the presentation of hierarchical, temporal and spatial semantics. Section 5 looks at interaction and, more specifically, direct manipulation techniques and the creation of interaction mappings. The section also provides a discussion about interactive views and distortion, which are two common interaction-based methods used to alleviate some of the problems incurred through the presentation of large complex datasets in coordinate space. Finally Section 6 discusses the current state of interactive data mining and the few relevant tools that are available, and Section 7 provides a brief look at the future directions of interactive data mining research.

2 User Participation

Computers process data at a syntactic level only. For example, a computer has little understanding of the semantics behind the string *book* and therefore which is the correct interpretation for a particular instance. For the correct semantic interpretation the computer needs to understand the context in which the term is presented. An accurate comprehension of complex context is beyond the ability of computers at present. The inability of the computer to understand is significant to data mining and the knowledge discovery process, as the objective is to find patterns of interest.

Identifying what is of interest is non-trivial and much research has been done within this field (Hilderman & Hamilton 1999, Piatetsky-Shapiro 1991, Silberschatz & Tuzhilin 1996). There are two classes of measures of interestingness: objective and subjective. Objective measures are based upon heuristics, where the interestingness of the pattern is defined objectively based upon a function of the discovered pattern and its associated data. Piatetsky-Shapiro (1991) formally describes this function as follows.

Definition 1 *The objective interestingness of a rule* $X \rightarrow Y$ *is defined as a function of* $f(X)$, $f(Y)$ *and* $f(X_Y)$, *where* $f(k)$ *is the probability that* k *is true.*

However, objective measures fail to capture all the characteristics of pattern interestingness as heuristic measures are logically constrained (Silberschatz & Tuzhilin 1996). An item of interest is one that incorporates characteristics of novelty, complexity, focus and usefulness. From this definition it is apparent that patterns cannot be classed as interesting through an analysis of a pattern's structure alone, but must also incorporate subjective measures.

Subjective measures of interestingness depend not only upon the structure of the rule and the underlying data but also upon the user's interpretation of the pattern's representation. For example, one characteristic of an interesting rule is that it must be goal-oriented; satisfaction of this characteristic is based upon an understanding of the mining task goals. For example, if a user is trying to justify additional department funding, a pattern indicating a trend in increasing employee height would not be useful. Subjective interpretation provides semantic understanding of patterns because users have the ability to comprehend data semantics and relate them to the problem domain. This builds upon the concept of knowledge-based architectures for Human Computer Interaction (HCI), which have explored the possibility of an implicit communication channel that, in an abstract sense, provides the computer with knowledge of the problem domain and objectives (see Figure 1) (Dix, Finlay, Abowd & Beale 1998).

Data mining algorithms can generate a large number of patterns, most of which are of no interest to the user. It is therefore essential to incorporate both subjective and objective measures of interestingness into the mining process, constraining the algorithm to an extent where only the most interesting patterns are generated. The inclusion of subjective measures requires the user to actively participate in the data mining process, creating synergy through an understanding of the data, that will result in the discovery of a more concise set of interesting rules and probably decrease mining time. Participation may also promote better work ethics due to what is known as the *Hawthorne Effect*, which states that *people tend to work harder when they sense that they are participating in something new or in something in which they have more control*

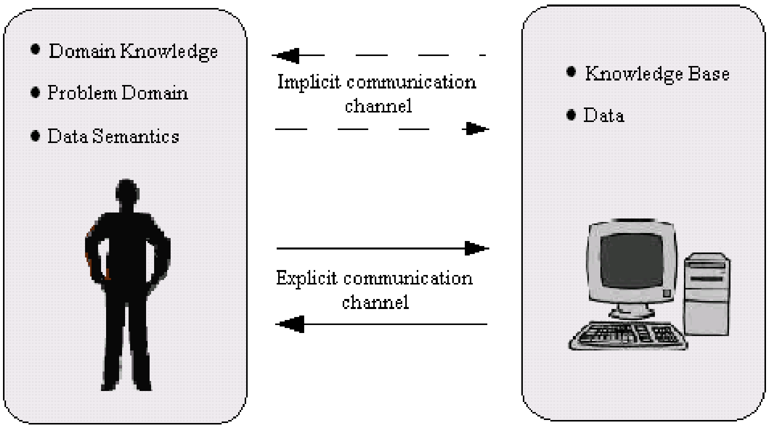


Fig. 1: Knowledge Based HCI

(Mayo 1945). In order for the user to participate in the mining process there must be mechanisms in place to provide for this functionality. Such mechanisms include:

- One or more interfaces between the user and the mining process.
- A cause and effect mapping between interaction primitives and mining process manipulation.
- Mining algorithm extensions allowing for guidance of the processing through human interaction.

3 Presentation

The acquisition of knowledge derived from a set of data requires a presentation of the data's underlying structure to the user. The most common method is visual presentation, which is the focus of this section. However there is current research into the auditory presentation of data and the benefits of combining auditory with visual presentations (Barass 1995). There are two classes of visual presentation: textual and graphical. Textual presentation methods are simpler but more constrained. Graphical techniques are more difficult to implement but show more promise as they facilitate discovery through incorporating human perception in a less constrained manner.

Textual presentations are constrained to a set of well defined primitives (characters, symbols and mathematical operators), which are interpreted by the user in a sequential manner at a fine-grained level of detail, with each primitive examined in turn. For example, reading is a sequential low level interpretation of the symbols on a page. The benefit of this presentation style is that it is recognised and perceived in the same way by different users and is relatively quick and easy to produce. A drawback for textual presentations is that they are not conducive to the analysis of patterns, complex data or large data sets, all of which are key characteristics of useful data mining results.

Graphical methods or visualisations of mining results provide more powerful forms of presentation as they are not constrained to a pre-specified set of primitives. Graphical presentations take many different forms, as the underlying data can be mapped to many different types of graphical primitives such as position, shape, colour and size. Such diversity leads to individual visualisations being able to present many dimensions of data in a concise manner by mapping data dimensions to varied graphical primitives. By contrast, in textual presentations the data dimensions are mapped to the same textual primitive type.

Human Perception and Information Theory (Miller 1956) indicates that graphical presentation facilitates the search for patterns by harnessing the capabilities of the human visual system to elicit information, through visualisation, multi-dimensional perception, recoding, and relative judgement. Many experiments within the field of cognitive psychology have identified that regardless of sensory type (eg. sight, taste, and smell), humans can accurately perceive differences in the stimuli to a greater extent when many parameters of that stimuli are presented. For example, in experiments by Garner et al. (Garner, Hake & Erikson 1956), participants were presented with a series of single dimension stimuli in the form of images each showing a point at a different position on a line. Participants were asked to label each image either from a list of possibilities or with a number from 0 to 100 indicating where to the best of their judgement the point lay on the line. Results showed that on average humans could accurately perceive approximately 10 different placements. However in experiments where the visual stimulus was increased to two dimensions (Klemmer & Frick 1953) by the presentation of a point within a square, the level of perception rose to approximately 25 different placements. Multi-dimension perception thus suggests that graphical presentations will improve user perception due to their multi-dimensional nature. However, the relationship between dimensionality and perception has been found to be asymptotic. Above ten or so dimensions, addition of further dimensions does not improve perception (Miller 1956).

Recoding is the process of reorganising information into fewer chunks with more information within each chunk. This process is the means by which humans extend short-term memory. The concept of recoding suggests that it is more difficult to perceive patterns within textual presentations because of the fine-grained sequential interpretation required. This is not conducive to pattern perception as the logical units remain small, resulting in the inability to understand the underlying structure of the result set. Visual presentations present a more contiguous representation of the data that can often be interpreted as a single logical unit, providing a conducive means by which the overall structure of the data set may be examined.

Weber's Law states that the *likelihood of detection* [of a change] is proportional to the relative change, not the absolute change of a graphical attribute. This law indicates that a user's perception will be superior when relative judgement instead of absolute measurement is made. For example, it is easier to perceive the change in a graphical object if its original form is displayed with the newly modified representation because we can compare the difference or relative change between the two objects, whereas it is more difficult to perceive changes when the original object is replaced by the new because no comparison is available and reliance is instead placed upon the knowledge of the object's absolute measures.

Relative judgement is a graphical capability and is a major strength of graphical presentations as it allows the users to obtain a holistic qualitative view of the result set where relative differences between items can be recognised. This qualitative view is then used to focus attention, with subsequently more focused and quantitative analysis (absolute measurement) following. This process was dubbed the *Visual Information Seeking Mantra* by Shneiderman (1996) and is conducive to pattern discovery as it allows the user to analyse a picture at different levels.

Although more powerful and flexible than textual presentations, graphical presentations are more difficult to create and are open to subjective interpretation, whereas textual primitives have in general a more stable interpretation. Subjective interpretation is due to the abstraction of the underlying results into graphical primitives through defined mappings. This allows the results to be presented in ways that facilitate perception of patterns and structure within the result set, but if non-intuitive mappings are used then the perception of patterns will be less predictable.

4 Presentation of Mining Results

The variance in subjective interpretations of a presentation can be reduced through good design. This includes ensuring that presentation styles reflect the data-mining task, and that the mapping between mining and graphical primitives is intuitive, takes into consideration the user's objectives and facilitates interaction techniques. Clearly there is no single best presentation technique for a data-mining task as there are too many factors that depend upon both the user and the problem domain. The solution is therefore to create a flexible set of presentation formats for each mining task that can satisfactorily be applied to a wide range of problems.

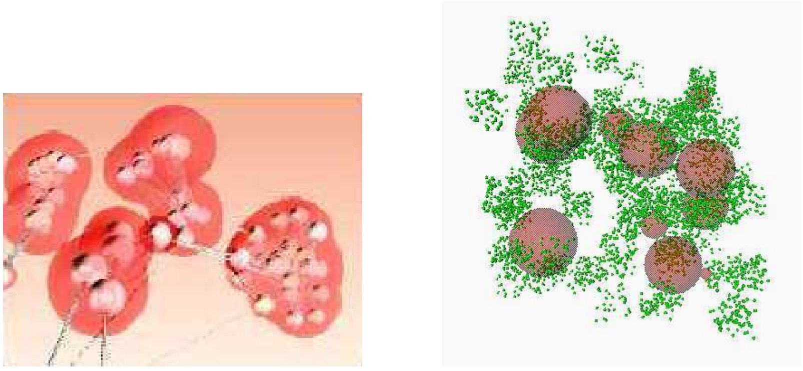
4.1 Clustering

Clustering refers to a group of automated techniques that objectively group items into classes based upon selected item attributes, maximising intra-class similarity and inter-class dissimilarity. These techniques are used to discover attribute correlations and overall distribution patterns within sets of data, helping the user understand the natural grouping structure of the data (see (Fasulo 1999, Hartigan 1975, Jain & Dubes 1988, Rasmussen 1992) for detailed discussions on clustering algorithms).

Clustering algorithms are based upon calculating the similarity measure of selected attributes between items. This requires the participating attributes to be in numeric form. Hence clustering visualisations are usually presented in coordinate space within which the items coordinates are directly mapped to the environment's coordinates. The base criterion for a clustering visualisation is an understandable representation of both the participating items and the clusters to which they belong. However the visualisation of these criteria are subjective. The choice of mapping depends upon the type of clustering algorithm used, the objectives of the clustering task and the users perception of how this information is best represented.

We present here an overview of clustering visualisations that facilitate perception. The visualisations are divided into partition-based and density-based presentations. Grid-based techniques are incorporated within the density-based section due to their common grounding. This is followed by discussions on volume rendering and projection, which deals with clustering visualisation issues. The section concludes with discussions on the inclusion of hierarchical, spatial and temporal semantics into clustering presentations.

Partition-based Presentations Partition-based methods analytically subdivide items into a number of clusters such that items in a cluster are more similar to each other than they are to items in other clusters.



(a) H-Blob: 3D clustering presentation

(b) 3D centroid clustering with membership mapping

Fig. 2: Centroid based 3D clustering presentations

Figure 2 presents two types of 3D partitioning visualisation. Figure 2a is a snapshot of the tool H-Blob created by Sprenger, Brunella and Gross (2000). The figure indicates the item-points and represents their cluster membership as a translucent encasing. Figure 2b presents the same elements and although less informative than H-Blob with respect to item membership, provides more information regarding cluster membership by correlating centroid size with membership.

A general problem with partition-based clustering methods is that the shape of all discovered clusters are convex because a partition is equivalent to a Voronoi Diagram and each cluster is contained within one of the Voronoi polygons (Sander, Ester, Kriegel & Xu 1998). To overcome this limitation, density-based algorithms were devised.

Density-based Presentations Density-based algorithms regard clusters as dense regions of items which are separated by regions of low density. This group of algorithms relies upon point-density functions and a density threshold parameter to discover clusters of arbitrary shape. The first presentations of these algorithms were planar, as illustrated by Figure 3. The figure shows a DBSCAN (Ester, Kriegel, Sander & Xu 1996) presentation that illustrates the discovery of arbitrarily shaped clusters, each represented by a different colour.

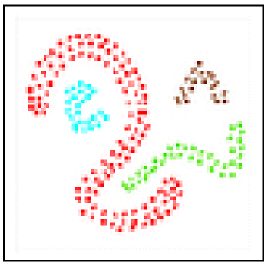


Fig. 3: Density based clustering presentation

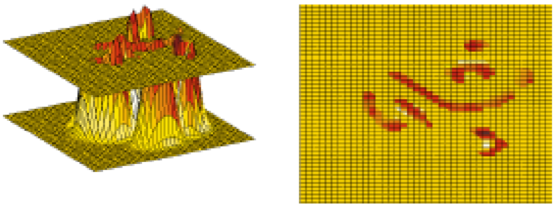


Fig. 4: Partition based density presentation

Figure 4 represents a more informative visualisation from the University of Halle's DENCLUE system (Hinneburg & Keim 1999), which illustrates clustering over two attributes. The left image extends into 3D space indicating item-point density within the planar coordinate space. The selected density threshold is represented as a slice parallel to the image base. The right image reflects the identified clusters as though looking down upon the threshold slice as it cuts through the 3D space.

Grid based clustering is a density-based method that optimises processing through the summarisation of point data. This is accomplished by mapping the data points to a grid of like dimensionality. Where the number of points within a particular cell exceed a density threshold, the cell is classed as dense and included as part of the cluster. This type of clustering differs from other density-based presentations, as the visualisation is of the cells not the data-points. As indicated in Figure 5 the clustering can be performed at different levels of resolution by varying the grid cell size. This variance results in a tradeoff between clustering accuracy and processing time (Hinneburg, Keim & Wawryniuk 1999).

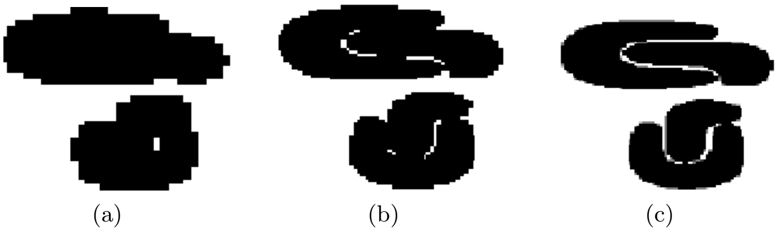


Fig. 5: Grid based clustering presentation with resolution variance

Projection techniques To this point we have assumed that clustering has involved only two or three item attributes, the results of which can be mapped directly onto coordinate-space presentations[2]. Presentation of clustering involving more than three item attributes is more difficult because projections of higher dimensional space are unfamiliar. There are two solutions to this problem: the reduction of attributes through non-linear dimension reduction techniques prior to the clustering process, and the viewing of high dimensional results through projecting the data onto lower dimension sub-spaces.

Central to non-linear dimension reduction techniques is the concept that, regardless of the dimensionality, a relative distance can be calculated between all item pairs (Li, Vel & Coomans 1995). Calculation of this distance provides a basis for defining topology-preserving transformations that project each item from n-dimensional space to two or three dimensional space whilst preserving the relative distances between each item. Several different techniques exist for performing these transformations including, multidimensional scaling (Young 1987) and spring-embedding systems (Bruss & Frick 1996, Gross, Sprenger & Finger 1997). Once the transformation has occurred the clustering is performed on two or three attributes which can then be directly mapped to viewing space.

Subspace projection techniques can be presented either statically or through animation. Static presentations appear as scatterplot matrices (Carr, LittleField, Nicholson & Littlefield 1987, Ward 1994), each of which contains a different paired combination of the clustered attributes. Figure 6 illustrates this method with colours used to differentiate clusters. Animated subspace presentation is based upon the grand-tour concept devised by Asimov (1985). This concept is an extension of data-rotation for multidimensional datasets, whereby a tour of the high dimensional space is created by iterating through subspace projections via interpolation along a geodesic path, creating the illusion of smooth motion.

Brushing techniques (Ward 1994) can be used within both the static and animated techniques to track items in separate subspace visualisations. For example, within scatterplot matrices the brushing of an item will result in it being highlighted in each matrix. Within animated projections such as GrandTour, brushing of an item will allow the user to track its movement from one subspace

[2] Single attribute clustering is presented using common techniques like pie-charts and histograms, which are not discussed within this paper.

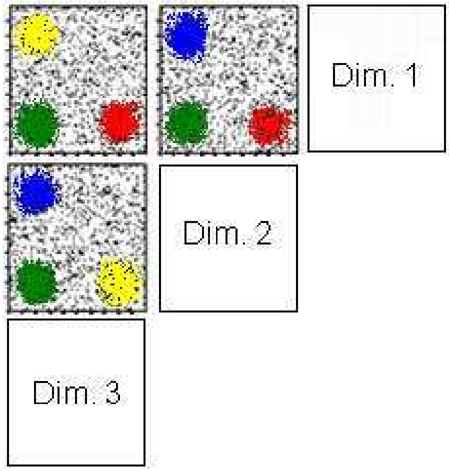


Fig. 6: Scatterplot matrix projection

presentation to another. The way in which the point moves may elicit further information about the item and its relationships.

Tour paths can be selected in different ways including random, statistical and cluster-separation techniques. Random tours arbitrarily iterate through every set of different attributes. Statistical tours use explorative statistical techniques such as principal component analysis (Faloutsos, & Lin 1995) and projection pursuit indices (Asimov & Buja 1995) to examine the statistical significance of variables and to decide which should be included in further analysis. Cluster-separation techniques (Yang 2000) use cluster-centroid positions to facilitate projection selection, each one based upon the mapping of the item to a three dimensional subspace determined by the centroids of four clusters whereby the distance between these centroids is maximised.

Volume rendering It is often the case in large datasets that multiple items map to the same visualisation coordinate, which ultimately leads to misinterpretation of the visualisation as many items appears as one. The presentation of large numbers of items can result in a cluttered environment, making it difficult to comprehend. To alleviate these issues the results can be presented as an item-density map instead of a item-point map using volume rendering techniques. This technique promotes the perception of density at each location in the respective environment. Although not as accurate as item-point representations due to the use of binning techniques to calculate the voxelised data to be rendered, it is a useful overview technique that can be used as the starting point for exploration of the results. A detail threshold can be defined within this type of presentation, at which point the representation could change to a item-point map to give a more accurate, detailed representation.

Hierarchical clustering presentations Cluster algorithms that result in the identification of hierarchical clusters or clusters of differing strengths are based upon extensions to either partition or density-based algorithms. The associated presentations reflect the underlying nature of the algorithm and incorporate hierarchical semantics or structure as illustrated in Figure 7.

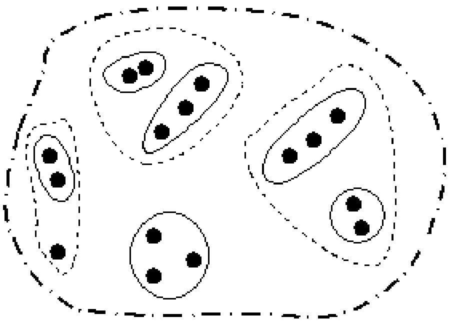
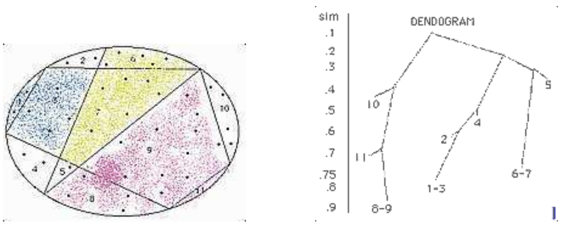


Fig. 7: Partition-based hierarchical clustering

Partition-hierarchy presentations are based upon the result of a sequence of partitioning operations, whereby different levels of clusters are discovered by splitting or merging currently discovered clusters. Figure 8 presents two different representations of the same clustered dataset. Figure 8a presents a planar circle bounded set of text item-points positioned in accordance with selected feature values and represented as black points. The dissecting lines represent the hierarchical clustering of the dataset. Each area (numerically labelled) relates to a leaf-item in an associated dendogram shown in Figure 8b, which clearly indicates the hierarchical nature of the clustering (Fox 2001).

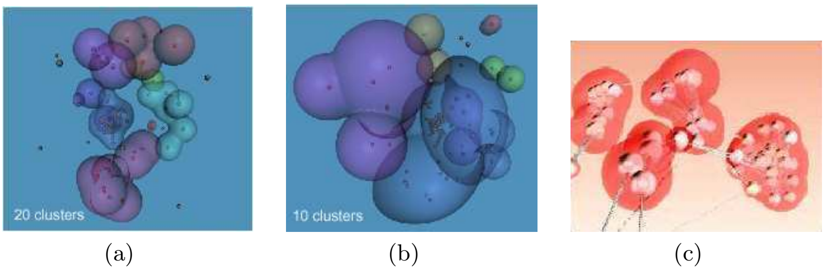


(a)

(b)

Fig. 8: Hierarchical clustering planar presentation, comprised of a hierarchically partitioned space and associated dendogram (Fox,2001)

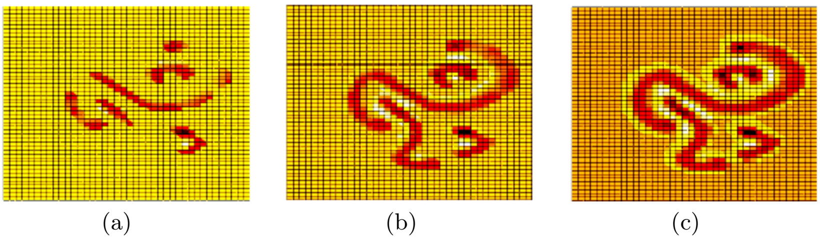
Three dimensional presentations of partition-based hierarchical clustering are typified in Figure 9, which shows a visualisation of the H-Blob system that uses smooth translucent shapes (blobs) to indicate cluster boundaries (Sprenger, Brunella & Gross 2000). In sequence, these separate visualisations represent instances of an agglomerative H-blob session where the clusters are built up from individual data points. The different levels of clustering are represented separately to help perception, as the superimposition of all blobs upon a single image would result in a cluttered and occluded environment.



(a) (b) (c)

Fig. 9: H-BLOB: Hierarchical clustering presentation

Density-hierarchy clustering methods are based upon the selection of a sequence of density threshold levels as illustrated in Figure 10, Figure 10a is the same as that presented in Figure 4. By varying the density-threshold level, different clusters are apparent. In this instance the thresholds have been arbitrarily selected. However more intelligent techniques such as OPTICS's reachability-plots have been developed (Ankherst, Breunig, Kriegel & Sander 1999). Figure 11 illustrates an OPTICS visualisation of both the reachability plot and the resultant hierarchical clustering with arrows super imposed to illustrate the mapping between the two images.



(a) (b) (c)

Fig. 10: DENCLUE - Density presentation of same dataset with different thresholds indicating possible hierarchical extension

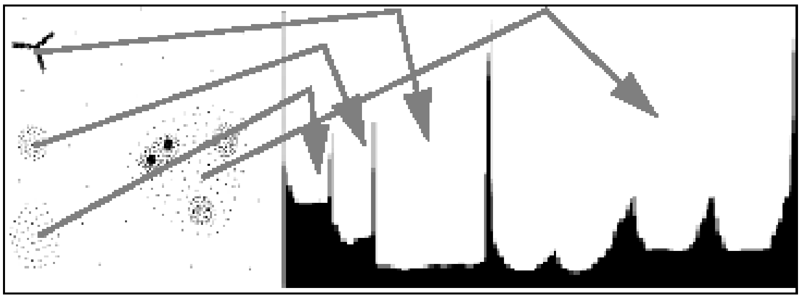


Fig. 11: OPTICS - hierarchical clustering using reachability plot

Conglomerative hierarchical visualisations exist within planar space (Figures 10, 11, 13, and 14). 3D space presentations commonly avoid conglomerative visualisations (Figure 9) and instead represent hierarchical clustering as a sequence of separate visualisations. This techniques reduces occlusion, whereby the lower level clusters will be difficult to see because the higher level clusters will in effect hide them from view. For example, H-Blob (Figure 9) does not use conglomerative presentation because although the membranes representing the clusters are translucent, lower level cluster membranes would remain unclear, especially in scenario's involving many levels.

Spatial clustering presentations In general the same techniques can be used to present spatial and non-spatial clustering results because of the common grounding of all clustering algorithms in distance metrics. The only difference is whether the coordinate-space mapping is direct or abstract, where a direct mapping refers to spatial semantics. However there are two techniques related to spatial clustering that are of interest from a presentation perspective: spatially extended objects, and spatial obstacles.

Within many different application areas items occupy an area instead of a single point. The clustering of these spatially extended objects or polygons requires specialisation of the density-based method whereby each object is considered to have a bounding area instead of occupying a particular point. The results are then presented in a typical coordinate-space as illustrated by Figure 12.

The incorporation of real-world physical constraints such as the presence of lakes and highways can effect clustering results. For example, Figure 13 shows how COD (Tung, Hou & Han 2001) tackles this problem. The underlying algorithm incorporates knowledge of the spatial location of obstacles that constrain the clustering, such that a cluster cannot cross an obstacle.

Temporal clustering presentations Clustering can incorporate temporal semantics through either incremental or instance-based techniques. Incremental techniques involve the real-time incorporation of new or modified data into the

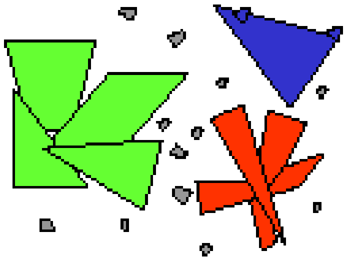
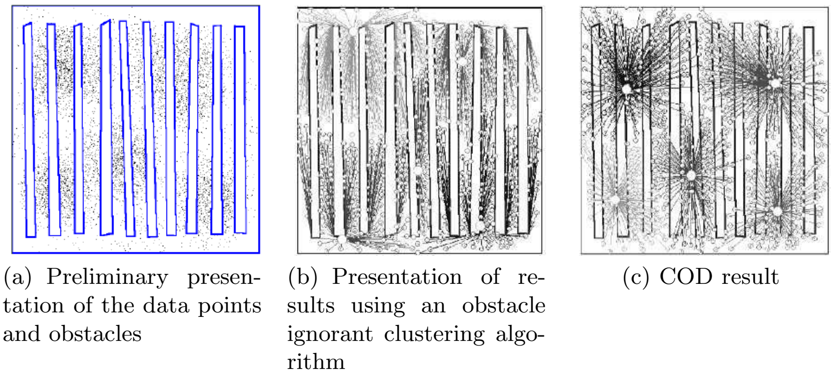


Fig. 12: Clustering of spatially extended objects



(a) Preliminary presentation of the data points and obstacles

(b) Presentation of results using an obstacle ignorant clustering algorithm

(c) COD result

Fig. 13: COD - Clustering with obstructed distance

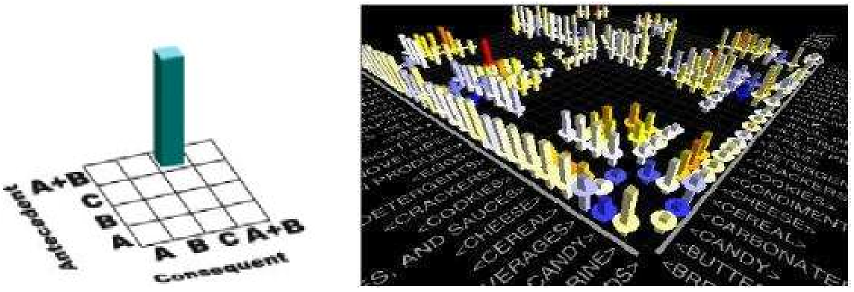
clustered result set. Instance-based techniques require the data set to be clustered in its entirety at specific instances in time. The presentation of temporal clustering may be either static as a sequence of static images, each of which represents the clustering at a particular time, or the temporal semantics can be incorporated through the use of animation. However, the actual presentation forms do not differ from those already presented.

Presentation difficulties may arise when trying to incorporate both hierarchical and temporal semantics in a clustering presentation, especially in 3D space. Hierarchical inclusion requires sequencing of images to avoid occlusion. If temporal semantics were also to be included as a sequence of images, a matrix of images would be required, each one representing a level of the hierarchy at a particular point in time. The most efficient way of presenting these semantics together would be in a single planar partition-based hierarchical presentation, using interpolation along an intuitive path to incorporate time.

4.2 Association Presentations

Association mining, or group affiliation, refers to a group of techniques that discover relationships between items based on frequency metrics (see (Agrawal, Imielinski & Swami 1993, Han, Kamber & Tung 2001, Srikant, Vu & Agrawal 1997) for further discussion). Associated presentations incorporate representations of the items and their relationships. These presentations fall into two classes: matrix-based and graph-based methods.

Matrix-based Presentations Two types of matrix structure used in the presentation of association rules are 2D matrices and mosaic plots. 2D matrices map the antecedent and consequent to separate axes with the third axis indicating the relationship strength, as illustrated by Figure 14a. Figure 14b is a visualisation from SGI's *MINESET* tool, which follows the same matrix design. Matrix-based visualisations are useful when small numbers of itemsets are to be presented. However they degenerate as the underlying result set increases in size and complexity as every new combination of valid antecedents or consequents is appended to each axis in the presentation resulting in an order of n2 rate of matrix growth. This may lead to large presentations that are cumbersome, occluded and hence difficult to understand.



(a) 2D matrix illustrating the rule $A + B \rightarrow C$ with support indicated by the third dimension

(b) SGI Mineset Visualisation of Association Rules

Fig. 14: Matrix based Association Presentations

Wong et al. (Wong, Whitney & Thomas 1999) have tried to minimise some of these matrix problems through the implementation of a rule-to-item based matrix shown in Figure 15, which is based upon the premise that an item can only occur once in a rule. The technique improves upon previous matrix presentations in that the matrix growth is linear when new rules are appended and the matrix is less sparse. Occlusion is also improved by displaying the associated support and confidence data as wall plates.

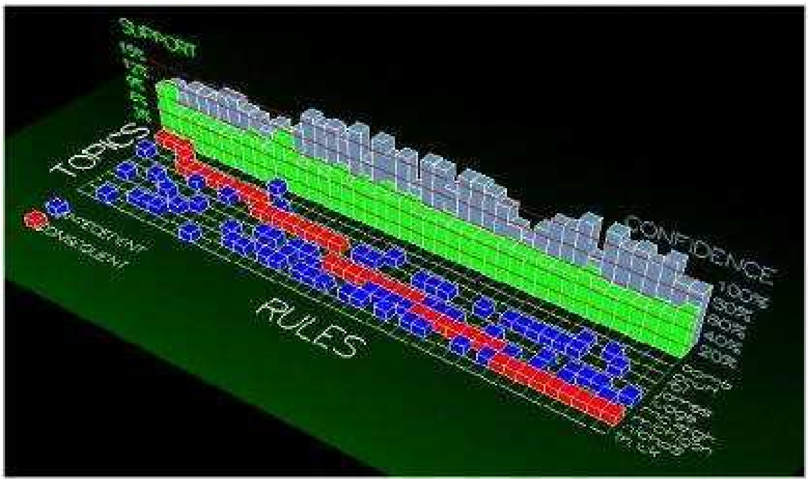


Fig. 15: Rule vs item association matrix

Hofman et al. (2000) created an alternative form of matrix visualisation - Interactive Mosaic Plots. This visualisation technique (shown in Figure 16) allows the investigation of the associations between a set of antecedents (and all permutations thereof) and a consequent. Within mosaic plots individual antecedents are represented as horizontal bars along the x-axis and the strength of an association is represented by the height of the vertical column above the specified antecedent permutation (inclusion denoted by black bar). Figure 16 illustrates the associations between the antecedent set *heineken*&*coke*&*chicken* and the consequent *sardines*, the vertical columns indicate both the strength of the positive-association (dark grey) and its negation *notsardines* (light grey).

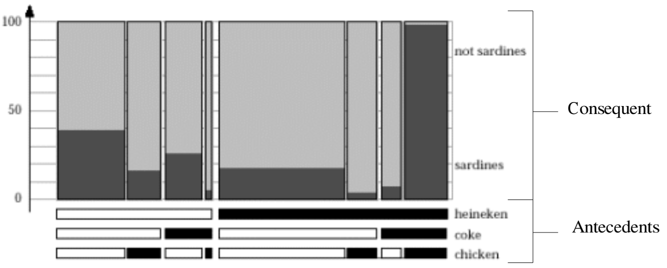


Fig. 16: Interactive Mosaic Plot (Hofman, Siebes & Wilhelm 2000)

Interactive mosaic plots allow the user to arbitrarily specify sets of antecedents and consequents. However the technique is designed for focused discovery where the set of attributes under consideration is small and becomes increasingly difficult to interpret as the number of items increase.

Figure 16 indicates that a potential rule of interest may exist in the form of *heineken&coke&chicken* ⇒ *sardines* as there is a significant difference between its confidence and that of all other permutations. The example highlights the contribution of the mosaic plots technique visualisation technique as although it is constrained in terms of volume of information presented, it allows a detailed analysis of the participants.

Graph-based presentations Graph based techniques present items as nodes and associations as the linking of nodes. Presentations vary in the placement of the nodes and the representation of metadata, including direction, confidence and support. This presentation type displays association rules in a more concise manner than that of matrix-based techniques. However as the number of items increase, graph based visualisations become cluttered and hence also hard to interpret.

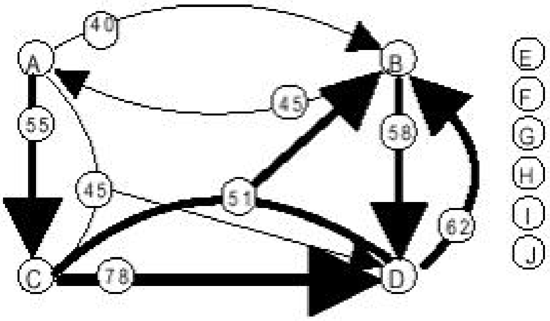


Fig. 17: Rule Graph

Rule Graph (Klemettinen, Mannila, Ronkainen & Verkano 1994) illustrated in Figure 17, is a comprehensive directed graph presentation. In this presentation instance items are represented by alpha-labelled nodes, with the arc thickness and label representing the association's confidence and support. Rule Graph uses rule templates to reduce the complexity of the presentation by allowing users to create display filters through template manipulation. This is indicated in the figure by items E through J, which have been removed from the display and appear to the right of the image. This allows the user to focus on rule subsets aiding in presentation comprehension.

Rainsford and Roddick (2000) (Figure 18) developed a circular visualisation in which the items are evenly spaced around the circumference and L_2 associations are represented as chords coloured with respect to association direction. This type of visualisation is effective in relating holistic information regarding the mining session, concisely representing results and indicating areas of interest.

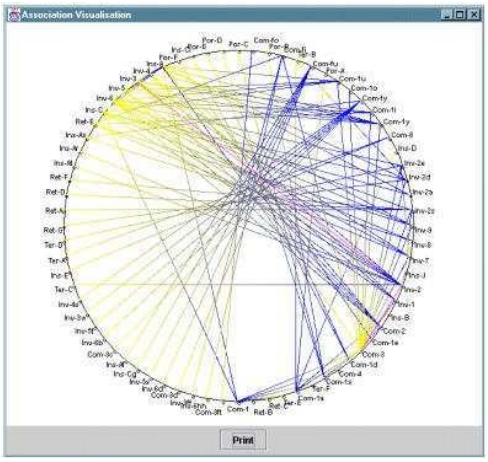
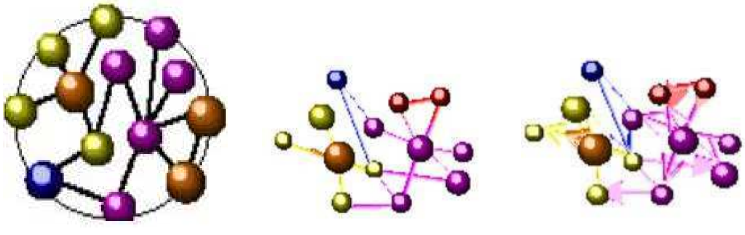


Fig. 18: Circular association rule visualisation (Rainsford & Roddick 2000)

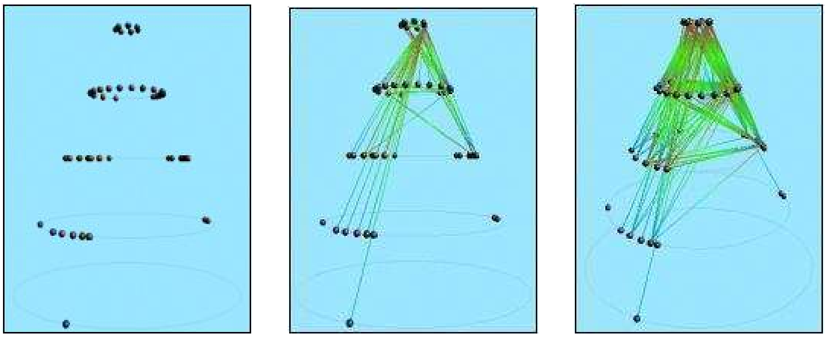
Directed Associated Visualisation (DAV) (Hao, Dayal, Hsu, Sprenger & Gross 2001), is a 3D visualisation technique that maps the items and relationships to positions and vertices on a sphere, using weighted edges to indicate confidence and arrows for direction. DAV distributes items equally on a spherical surface (Figure 19(a)). Based on physics principles of masses and springs, a support matrix is then created that relates the strength of the association between items in terms of spring tension. The spherical structure is then relaxed and a state of low local minimum energy is reached (Figure 19(b)), resulting in each item's relative position reflecting its associations. The direction and confidence of each vertex is then calculated (Figure 19(c)), and finally presented to the user.

Hierarchical association presentations Association mining over different levels of abstraction implies that items belong to a taxonomy and that interesting rules may be discovered by mining associations at not only the item level but also at higher levels in the taxonomy. Related presentations incorporate this hierarchical structure with associations that may be discovered amongst the differing levels of the taxonomy. Our research at Flinders University has created a visualisation technique incorporating hierarchical semantics known as CARV (Concentric Association Rule Visualisation) (Ceglar, Roddick, Calder & Rainsford 2003). This technique is capable of displaying both single-level and hierarchical association mining results as illustrated in Figure 20.



(a) Initialisation (b) Relaxation (c) Direction

Fig. 19: DAV Process (Hao, Dayal, Hsu, Sprenger & Gross 2000)



(a) Initial presentation of itemsets (b) Intermediate presentation stage (c) Final presentation

Fig. 20: Concentric Association Rule Visualisation

Spatial association presentation Spatial association mining is the discovery of relationships among sets of spatially oriented items, possibly influenced by nonspatial predicates. For example, the spatial association rule $is(caravanpark)$ and $closeto(waterbody) \rightarrow has(boathire)$ consists of spatial antecedents and a non-spatial consequent. Spatial items use spatial predicates to represent topological relationships between spatial objects, as illustrated in the above example with the predicates is and $closeto$, other predicates include $intersects$, $contains$, $adjacentto$, $leftof$ and $covers$ (Koperski & Han 1995).

Koperski and Han have undertaken the extensive research into this field. Although the algorithmic development is advanced, in general presentation is in textual form. Visual representations need to incorporate spatial predicates as well as regular association presentation elements. At present there is no intuitive means of doing so. A current technique by Koperski and Han uses a regular association graph annotated with textual spatial predicates (Figure 21).

Temporal association presentation The inclusion of temporal semantics within association mining algorithms involves the discovery of discrete events

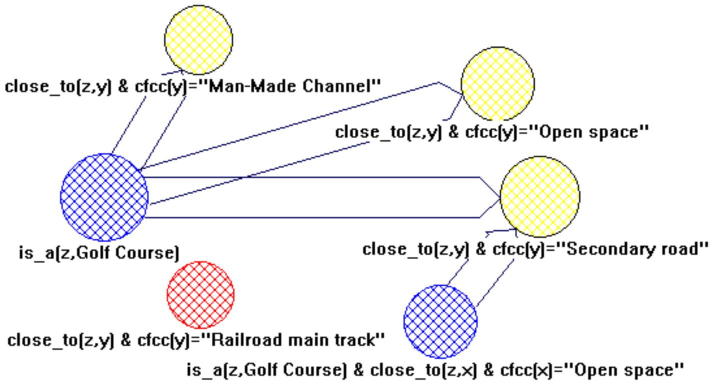

close_to(z,y) & cfcc(y)="Man-Made Channel"
close_to(z,y) & cfcc(y)="Open space"
is_a(z,Golf Course)
close_to(z,y) & cfcc(y)="Secondary road"
close_to(z,y) & cfcc(y)="Railroad main track"
is_a(z,Golf Course) & close_to(z,x) & cfcc(x)="Open space"

Fig. 21: Geominer - Spatial association rule visualisation

that frequently occur in the same arrangement along a time line. Therefore temporal mining is the study of the temporal element arrangement, whereas association mining is the study of item relationships. Like spatial mining, temporal mining requires the incorporation of a set of defined temporal predicates. For example, *tea* $\overset{after}{\rightarrow}$ *cinema* consists of two events *tea* and *cinema* with the predicate $\overset{after}{\rightarrow}$ indicating their temporal arrangement.

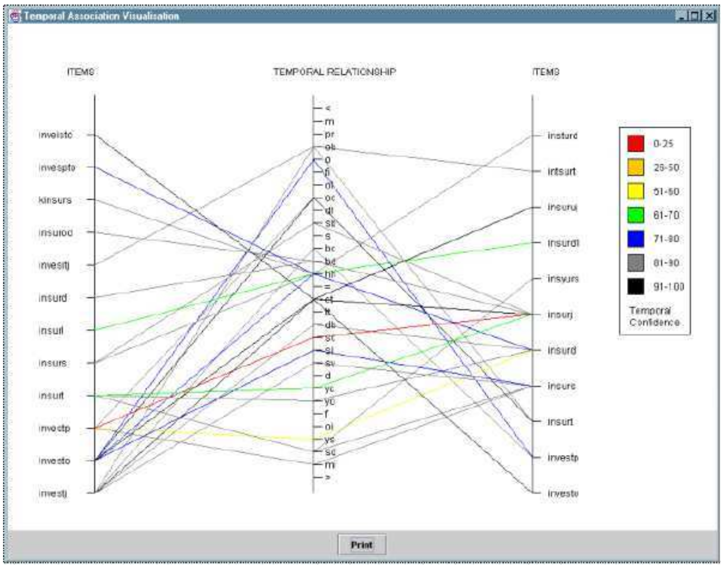


Fig. 22: Temporal Association Mining

Figure 22 shows a temporal presentation designed by Rainsford and Roddick (1999), that incorporates temporality, by representing the temporal predicates (centre column) through which each rule (line) passes. In this way the temporal relationships between the antecedent (left) and consequent (right) can be seen, with colour representing each rule's confidence. Although restricted to simple rules (single antecedent and consequent), this visualisation technique efficiently presents rules with diverse temporal predicates.

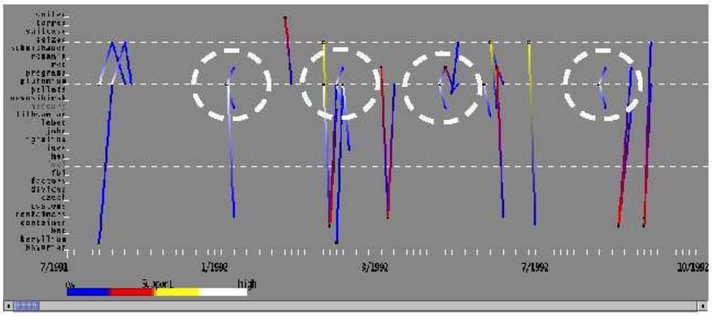
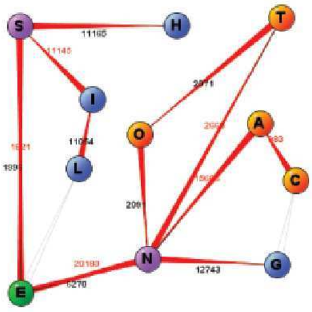


Fig. 23: Sequence Mining Presentation

Closely related to temporal mining is sequence mining, in which the patterns being discovered are constrained to sequentially discrete events. This focus on a particular temporal aspect facilitates visualisation design as only a single temporal predicate is to be represented. Research by Wong et al. (Wong, Cowley, Foote, Jurrus & Thomas 2000) focus on sequential mining and to this end they have developed a visualisation for the presentation of temporal patterns discovered from newspaper article topics over a period of time. This is illustrated in Figure 23, where the topics are listed on the y-axis and the timeline along the x-axis. The patterns found at various times are displayed in a colour representing level of support. The four dashed circles highlight the presence of the same two patterns within the time period.

The technique by Wong et al. provides a mixture of qualitative and quantitative information, showing the patterns while providing information regarding the times at which they occurred. A more qualitative and concise presentation of the interesting information (the two re-occurring patterns) might have been achieved through the use of Rainsford's presentation method. However this would be dependent upon the underlying algorithm.

Advances in sequence mining has resulted in a need to incorporate more meaning within the generated rules, resulting in the inclusion of temporal logic semantics (Allen 1983) within sequential mining algorithms (Höppner 2002, Padmanabhan & Tuzhilin 1996). This enables rules such as the following to be detected and reported.



Legend	
Node Colour	**Description**
Green	Root Node for the interaction (E)
Blue	Enclosing sub-episode (G, L, I, H)
Orange	Enclosed sub-episode (C, A, T, O)
Purple	Shared node of both enclosing and enclosed sub-episodes (N, S)
Edge Colour	
Red	Relationship between two Nodes and their supports
Gray	The point(s) at which the enclosed sub-episode begins/ends within the enclosing sub-episode

Fig. 24: INTEM interaction of *CANTONESE* **during** *ENGLISH*

$$((\text{Alarm A} \overset{before}{\longrightarrow} \text{Alarm B}) \overset{during}{\longleftrightarrow} \text{Alarm C}) \overset{starts}{\longleftrightarrow} \text{Alarm D}$$

These types of rules can become quite complex and to the authors' knowledge have to date only been presented textually. In an attempt to present the rules in a more meaningful manner, we at Flinders have developed the INTEM (**INT**eracting **E**pisode **M**iner) visualization technique. The technique is similar to mosaic plots 16 in that it allows the detailed investigation of particular sequence interactions that are of interest to the user, as illustrated in Figure 24.

5 Human Computer Interaction

Visualisation facilitates the perception of patterns and structure within data mining results. However, a static presentation in itself is often inadequate and interactive capabilities are required to allow effective exploration of the visualisation. This relates to Shneiderman's Visual Information Seeking Mantra, which stated that the initial view is qualitative and of an overview nature and that through interaction the user can proceed to focus upon interesting sub-areas for more quantitative analysis.

Interaction occurs at many different levels, as illustrated by the Layered Interaction Model, shown in Figure 25. This model identifies a sequence of interaction levels that build upon each other, illustrating that a interaction requirement can be broken into different levels of abstraction, ranging from the subjective goals of the interaction through to the physical I/O of the interaction. There is however a definite shift between the concept-based upper levels and the activity-based lower levels, indicating a transition between what is required and how it is done. The mapping or transformation between the concept and activity levels of interaction is known as direct manipulation and occurs at the junction of the syntactic and semantic levels. This section discusses the exploration of presentations through direct manipulation and the use of views to facilitate understanding in large complex visualisations.

Level	*Name*	*Exchanged Information*	*Example*
7	Goal	Real World Concepts	Remove letter section
6	Task	Computer oriented actions	Delete 6 lines of edited text
5	Semantics	Specific operations	Delete selected lines
4	Syntax	Sentences of tokens	Click at left of first char., whilst holding down left mouse button, click to the right of the last character
3	Lexical	Tokens (smallest info carrying units)	Click at left of first char.
2	Alphabetic	Lexemes (primitive symbols)	Click at (200,150)
1	Physical	Hard I/O (movement,click)	Click

Fig. 25: Layered Interaction Model

5.1 Direct manipulation

Direct manipulation can be defined as the mapping between the semantic and syntactic levels of the layered interaction model (refer to Figure 25). The objective in constructing this mapping is to create as close a match as possible between the structure of how users think about a task and the activity used in solving it, while attempting to maximise problem domain compatibility (John, Rosenbloom & Newell 1985). Direct manipulation design effects the interactive quality of the system, including error frequency, speed of task performance, and user skill retention (Buxton 1986).

There is not always a single best set of interaction capabilities for a particular visualisation. Not all users have the same mental model, and even for a single user the mental model may differ depending upon the user's current goals. Therefore the optimal solution is an intuitive set of mappings that mimic real world activities. For example, the task of moving a file in a paper-based office involves going to the relevant filing cabinet, removing the required file and carrying it to its new location. Intuitively the same task on a computer system should follow the same steps. Graphical interfaces provide this capability by representing data as graphical icons and encapsulating activities within interaction mappings. So the movement of a piece of data within the computer system will generally involve selecting the relevant icon and dragging it to its new location. Importantly, the syntax of operations in an interaction command should closely correspond to the data's required semantic changes, and the screen representation should reflect these changes.

Graphical level interaction is based upon selection and navigation activities that are specified to the computer by the user through pointing devices. Selection is the designation of a point of interest within the graphical interface, signified through an action such as clicking a mouse button or pressing a key. Navigation is the movement of the interest focus, which is generally accomplished through a continuous activity such as moving the mouse or holding down specific keys.

These primitives work together within different environments to provide the means by which any presentation may be explored in a detailed manner.

As there are an arbitrary number of semantic tasks that can be undertaken within a presentation, the overloading of an activity primitive such as selection is achieved by varying the graphical primitive specification. For example, the functionality required within a presentation environment might include the ability to delete items and to display item details in a pop-up dialog, both of which involve item selection. Overloading of the selection primitive is achieved by either varying the selection action for each semantic task or requiring a combination action either in sequence or parallel. Another technique used to combat overloading, (especially in navigation) is indirect manipulation, whereby the presentation is manipulated through interaction with associated graphic artefacts. A common example is the use of scroll-bars within presentation environments to provide navigation at both the screen and document levels. However, indirect manipulation techniques require that the user's focus be drawn away from the actual presentation and lessen the interactive experience (Koedinger 1992). Mice with scroll-wheels overcome this by providing an additional form of vertical navigation, hence providing a means for direct navigation at both the screen and document levels.

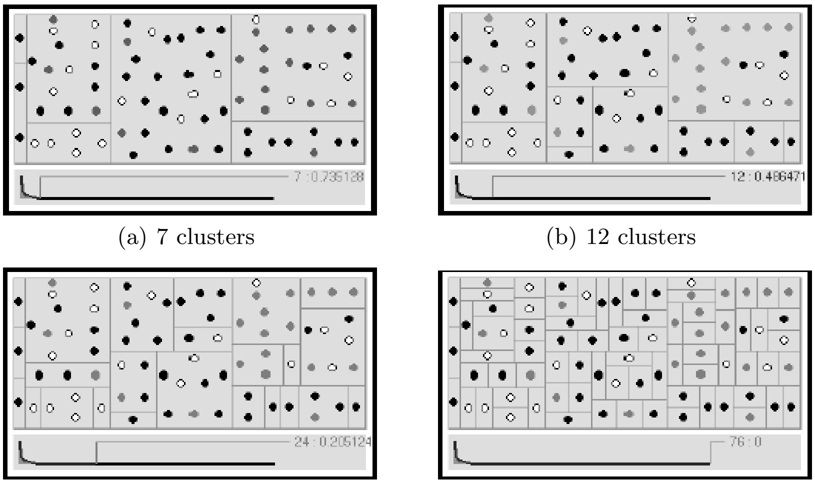
3D presentations commonly provide more freedom but increase input overloading and hence the variation of actions required to effectively explore the environment. Alternatively the use of immersive presentation devices such as headsets, and pointing devices that allow further degrees of freedom such as flying mice and gloves can increase the user's experience of direct interaction and reduce the mapping overload.

The provision of a succinct set of direct manipulation mappings is critical for effective presentation exploration. The level of interactive functionality provided is important as too little will constrain the exploration process and too much will result in interactive quality degradation. This degradation reflects the provision of too many functional alternatives, resulting in non-intuitive mappings and hence longer task times and less skill retention. Therefore good direct manipulation design involves the specification of a succinct intuitive set of mappings based upon selection and navigation primitives to facilitate the comprehensive exploration of a presentation.

5.2 Interactive Views

Mining result sets are often large and complex, typically tens of thousands of items. Their size makes them difficult to entirely represent in a form conducive to understanding by the user. Large result sets produce cluttered presentations because of the larger number of graphical objects required to represent the underlying data and their discovered relationships. Additional problems occur in the presentation of hierarchical clustering semantics where separate presentations are required to display different hierarchical instances of the clustering.

Interactive views use direct and indirect manipulation methods to enhance user perception of a result through user-specified filtering of the result set so that only a subset are presented (Klemettinen, Mannila & Toivonen 1997, Ribarsky, Katz, Jiang & Holland 1999, Wills 1998). Filter parameters are generally specified through indirect manipulation via interface controls and can involve threshold, parameter, and template specification and also control the participation of individual items and discovered relations. Threshold specification can be used in association rules to constrain the confidence and support parameters that provide heuristics as to the strength of the association and the importance of the item within the data set. By raising the required confidence threshold, the weaker rules will no longer be displayed. Templates can constrain the rule form. For example, the presentation can be limited to those rules that have only a single antecedent.



(a) 7 clusters

(b) 12 clusters

(c) 24 clusters

(d) 76 clusters

Fig. 26: Interactive Views of hierarchical clustering

Clustering based presentations are different in that the discovered relationships are global in nature unlike association mining where each relationship effects only a few items, therefore constraint of the presentation to a subset of clustered items is generally unwarranted. However Figure 26 shows how interactive views can be useful in presenting hierarchical clustering results. The user can change the number and hence levels of clustering viewed through the use of an associated slide bar.

6 Guided Knowledge Discovery and Interactive Data Mining

Interactive data mining is a process whereby the user can guide the knowledge discovery. It is achieved by dynamically presenting the results of the data mining process and incorporating interactive capabilities within the presentation environment. This allows the user to interact with the underlying mining process. Through such interaction the user is able to guide the mining process and steer the discovery process to areas of user interest.

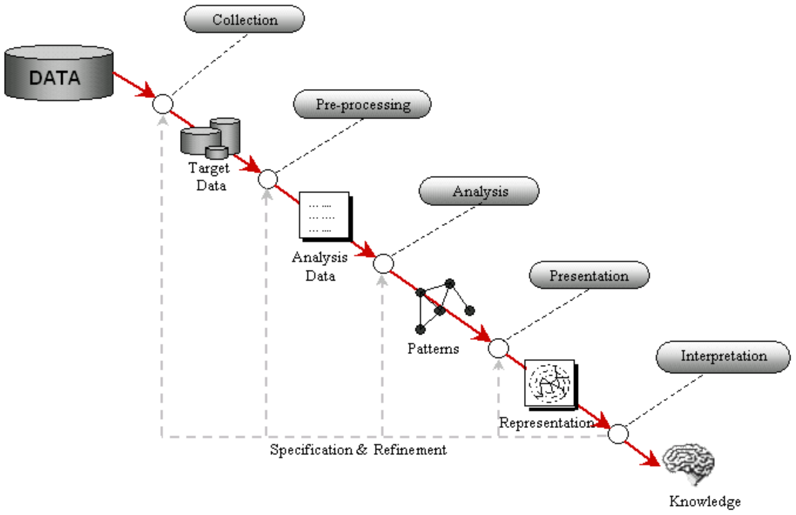


Fig. 27: Knowledge Discovery Process

The iterative knowledge discovery process is illustrated in Figure 27. This process is controlled by user specification of data sources, tools, and associated parameters. The process includes data collection (possibly from multiple heterogenous sources), pre-processing (which massages the data into a form required for analysis), analysis (mining), presentation, and finally interpretation by the user. This widely accepted model of the knowledge discovery process shows each stage as being independent, with the interaction between stages constrained to the piping of the output of one stage to the input of the next stage. Depending upon the extent to which the results satisfy the user's goals, the user may refine the specification of tools and parameters and *re-cook* the results. This *re-cooking* or iteration of the knowledge discovery process or portions thereof can be a time expensive exercise, with the stages of collection, preprocessing and analysis requiring many hours of work for large or complex data sets.

Some researchers report that the initial specification of the collection and preprocessing stages can take up to 60% of the processing time. However because the user has a high degree of control over these processes and can specify exactly what is required, refinement is typically unnecessary. The analysis or mining stage, however, is generally batch oriented, so the user has no control over the process. User refinement and re-cooking generally occur at the analysis stage, with the user tweaking the mining algorithm's parameters in an attempt to produce results of greater interest. It is this lack of user control within the analysis process that is rectified by incorporating interactive capabilities within the mining process.

Involving the user in the mining process requires an interface whereby the user can see and guide the mining. In this way, the coupling between the analysis and presentation stages is strengthened by providing real-time streaming of analysis results to the presentation tool for user interpretation and by allowing the subsequent guidance of the mining algorithm through user interaction with the presentation.

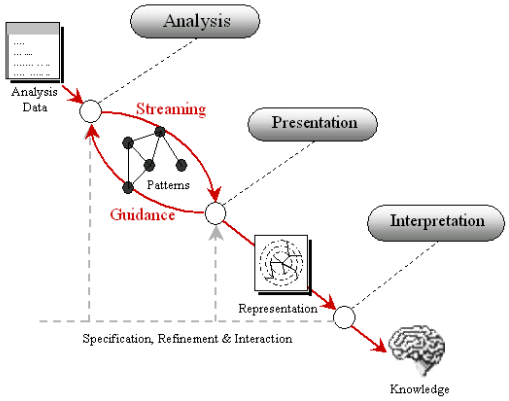


Fig. 28: Guided Knowledge Discovery Process

Figure 28 shows the changes to the process. The change results in the merging of the analysis and presentation stages, although current research is focused on maintaining stage independence and increasing the coupling through the specification of a generic set of interface methods that capture the required interactive functionality. This technology will provide flexibility by allowing the user to select the presentation technique that best suits their current needs. Examples of this technology are the FIDO project currently being undertaken at Flinders University (Roddick & Ceglar 2001) and the KESO project at the University of Helsinki (Wrobel, Wettschereck, Verkamo, Siebes, Mannila, Kwakkel & Klosgen 1996).

Increased coupling between the analysis and presentation stages is illustrated in Figure 29 and Figure 30, which differentiate between a batch and interactive mining run. The four images comprising these figures represent data mining runs as a tree structure with each level implying a point in time. The time interval is not constant but indicates a point at which the mining algorithm reaches a stage where intermediate results may be produced. The production of intermediate results is algorithm dependant, however most exploratory mining algorithms, for example *Apriori* and k-means, involve multiple iterations over the dataset, which provide natural processing stages. Where the algorithm requires only a single pass, techniques such as sampling can be used to provide intermediate results. Within Figures 32 and 33, the nodes within each tree level indicate the discovered information at that point in time; the blue nodes represent intermediate results and the red nodes represent terminal results.

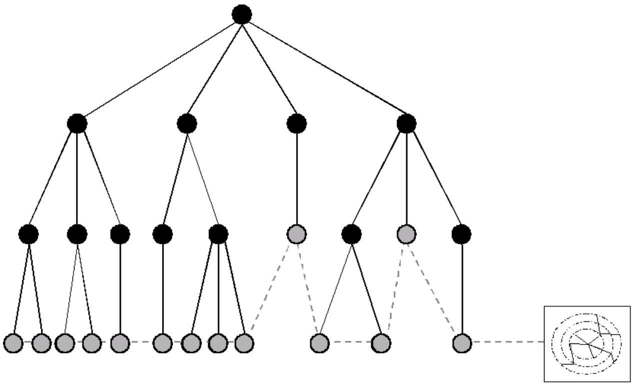
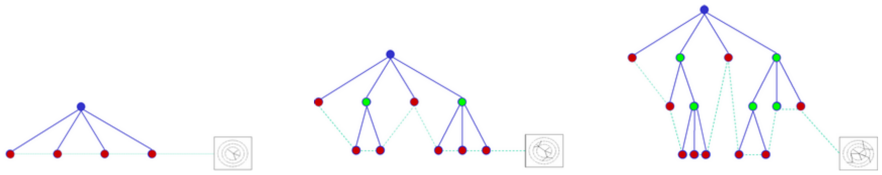


Fig. 29: Batch Data Mining Run

Figure 29 presents a regular batch mining run where the terminal results are presented only at completion. If refinement is required, the entire mining run must be repeated. Figure 30 illustrates the guidance of the knowledge discovery process through interactive mining. These three diagrams show the production or update of the presentation at different stages during the analysis process, thus providing the user with insight into the process.

The initial presentation (Figure 30a) is generated based upon the available intermediate results. The user is then able to interact with this presentation to guide further processing, represented in Figure 30b by the green coloured nodes that indicate the users interest in these intermediate results. Further processing is then constrained to the intermediate results indicated by these nodes. Figure 30c illustrates the completed state of the exploration process, within which can be seen the inclusion of further guidance.

Comparing Figure 29 and Figure 30c highlights the advantages of interactive data mining. In particular the terminal result set is smaller, with less inter-



(a) Initial presentation

(b) Intermediate Presentation

(c) Completed presentation

Fig. 30: Interactive data mining

mediate results being processed. This reduces mining time and produces less cluttered presentations. In addition the incorporation of subjective as well as objective measures of interest within guided knowledge discovery ensures that this smaller, more quickly produced result set will be of greater interest to the user. It is therefore also likely that less refinement and reprocessing will be required, further reducing mining time.

Other techniques used to reduce mining time, particularly in the case of association mining, are cache-based and constraint-based mining. Cache-based mining or dynamic mining involves the storage of the results of frequently requested queries for fast retrieval of mining over common sets (Nag, Deshpande & DeWitt 1999, Raghavan & Hafez 2000). Constraint-based mining includes techniques such as subset mining and the prior specification of constraints. Subset mining involves the mining of a subset of the data, the rules of which are then verified against the whole set (Toivonen 1996). However, this method is probabilistic and may not discover some rules of interest. Prior constraint techniques involve the specification of boolean expressions to constrain the mining process (Srikant et al. 1997). However it is often the case in explorative tasks the user does not know beforehand what is of interest; only through the presentation of options can the user focus on what is of greater interest. Although these methods may reduce mining time, they do not pre-empt guided discovery through interaction, and are not associated with the benefits it offers.

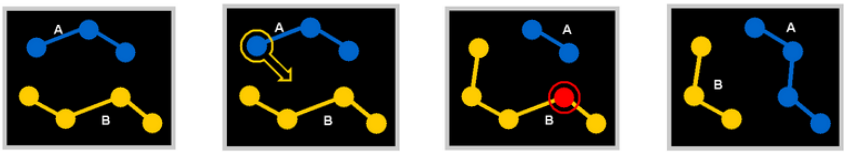
The granularity and types of interaction possible are dependent upon both the presentation type and mining task. The user may manipulate the mining process either in terms of its focus, its parameters or its constituent dataset by interacting with the associated real-time presentation. Figure 30 is indicative of an exclusion form of interaction whereby the user eliminates a node from further processing. A more advanced technique is that of priority interaction whereby the user selects the nodes of interest, which are processed first and hence presented first. This doesn't exclude the further processing of the other nodes; it simply puts off this processing until the prioritised work has been completed. This ensures that eventually all elements will be processed, so all possibly interesting results are discovered, while arranging for those of most interest (as specified by the user) to be processed first (Wrobel et al. 1996).

Although some researchers have investigated interaction in the context of data mining, most such systems actually incorporate forms of interactive views (Chu & Wong 1998, Klemettinen et al. 1997, Ribarsky et al. 1999, Wills 1998, Xiao & Dunham 2001) or are iterative but allow refinement through graphical interaction (Kim, Kwon & Cook 2000). The following subsections discuss those published works providing interactive capabilities in the fields of clustering and association mining.

6.1 Interactive Clustering

The Mitsubishi Electronic Research Laboratory (MERL) at Cambridge University have investigated the use of interaction in explorative analysis to solve optimisation problems in the fields of vehicle routing (Anderson, Anderson, Lesh, Marks, Perlin, Ratajczak & Ryall 2000) and network partitioning (Lesh, Marks & Patrignani 2000). This study centres on the Human Guided Simple Search (HuGSS) paradigm, which improves the effectiveness of a relatively simple search algorithm by allowing users to steer the search process interactively. The interactive capabilities provided by this system include the ability to escape local minima via manual editing and to focus searches into areas of promise.

An application to which the HuGSS paradigm has been applied is that of vehicle routing with time windows. This study involves the optimisation of goods delivery to a group of customers with the fewest trucks, while minimising the distance travelled by each truck. The system uses either a greedy or steep-descent clustering algorithm to determine the number of trucks and the routes they should take. The user specifies the number of steps in a search invocation. This effectively controls the number of automated allocation moves the computer can make before presenting a set of intermediate results to the user. Users have the opportunity to insert guidance primitives for the next invocation of automated allocation.



(a) Initial Solution

(b) User movement of customer from route A to B

(c) Moving that customer result in sub optimal solution

(d) Algorithm re-optimises based upon new constraints

Fig. 31: Repercussions of user movement of customer from route A to B

In this example, the user guides the route allocation process and escapes local minima by manually assigning customers to routes, in effect changing the cluster to which the item belongs. This automatically invokes a route optimisation algorithm on the effected routes only, ensuring that under these new constraints the system still provides the best possible solution, as illustrated in Figure 31. The user can additionally change a customer's priority, which effects when the customer is considered for movement by the algorithm, and changes the algorithm's associated objective measures and mode before invocating another sequence of automated route allocations.

The MERL group has also applied the HuGSS paradigm to the area of k-way network partitioning, a NP-hard problem arising from VLSI design and elsewhere. This application has required the development of a different set of presentation techniques that visualise the required relevant aspects. The work consequently led to the development of a new set of interaction mappings that effectively provide the same types of interactive guidance as the route discovery application (Lesh et al. 2000). Other work (Nascimento & Eades 2001) allow the user to interactively change clusters number and size however the crucial element in MERL's research is the ability to guide the cluster membership of particular items. This research by MERL is the most significant published work in the field to date.

6.2 Interactive Association Mining

Research undertaken by Brin and Page at the Stanford University (Brin & Page 1998) appears to be the only piece of research that satisfies our criteria for guided association mining. Known as Dynamic Data Mining (DDM), their work attempts to produce more interesting rules by foregoing traditional support-based algorithms (that use a single deterministic run) and instead use a method that continuously explores more of the search space. This is accomplished through the incorporation of a user-defined measure of interest, $Weight$, which can be redefined dynamically and a heuristic, $HeavyEdgeProperty$, which guides the exploration process.

Rather than the mining process being a single deterministic run (producing a well-defined set of itemsets and rules), DDM invokes a process that continually generates improving itemsets, based upon the Dynamic Itemset Counting algorithm (DIC) (Brin, Motwani, Ullman & Tsur 1997). This allows the DDM to take advantage of intermediate counts to form an estimate of an itemset's occurrence or weight, which results in the presentation of intermediate results to the user. The user is then able to dynamically adjust individual item weights, in effect prioritising them. User interaction is indirect and textual, with refinement occurring through an associated text box within which the user can adjust global mining parameters and individual item weights via a simple language.

This research provides a means of prioritising the mining of particular items and in effect allows the user to insert a subjective measure of interestingness into the algorithm. The technique will produce a more interesting and smaller set of results than traditional batch processing methods.

7 Conclusion

Guided knowledge discovery through interactive data mining as a discrete field of research is still in its infancy and as such there are few published works of relevance. This leaves open broad and diverse areas of further research in the areas of algorithmic development, interaction, and presentations and in associated areas such as collaborative guidance.

Future algorithmic research in the field will focus upon methods by which guidance can be incorporated into new or existing explorative algorithms at different levels of granularity. Preliminary areas that show promise include priority based algorithms (Brin & Page 1998), incremental computation (Sundaresh & Hudak 1991) and state based processing, which uses the concept of *rollback* to return to a previous intermediate state instead of re-instigating a new analysis. An associated area is the investigation of supporting frameworks that provide flexible interactive knowledge discovery environments.

It seems likely that the majority of these techniques will remain domain-specific (if not task- specific), because of associated subjective interpretation. The challenge lies in the creation of generic sets of interaction mappings between the graphical interface and the underlying mining process. The development of such mappings was indicated by the research of the MERL team (Anderson et al. 2000, Lesh et al. 2000) where a single set of interaction functions were effectively incorporated within two different problems domains, using different domain specific presentations.

Existing knowledge discovery tools do not adequately provide the capabilities to incorporate subjective measures of interestingness into the analysis process. Current analysis results in ineffective discovery processes, as heuristic measures cannot accurately portray what is potentially of interest to the user. As shown by the MERL team (Anderson et al. 2000, Lesh et al. 2000), and Brin and Page (1998), subjective judgement can be incorporated by actively engaging the user in the mining process. Benefits include an accelerated knowledge discovery process and improved results. User participation in the mining process results in greater confidence in the correctness of the discovered patterns, due to the sense of control that guidance capabilities provide.

References

Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, *in* '1993 ACM SIGMOD Int. Conf. Management of Data', Washington D.C. U.S.A, pp. 207–216.

Allen, J. F. (1983), 'Maintaining knowledge about temporal intervals', *Communications of the ACM* **26**(11), 832–843.

Anderson, D., Anderson, E., Lesh, N., Marks, J., Perlin, K., Ratajczak, D. & Ryall, K. (2000), Human guided simple search: combining information visualization and heuristic search, *in* 'Proc. of the workshop on new paradigms in information visualization and manipulation. In conjunction with the eighth ACM international conference on Information and Knowledge Management', ACM Press, Kansas City, MO, pp. 21–25.

Ankherst, M., Breunig, M. M., Kriegel, H.-P. & Sander, J. (1999), Optics: Ordering points to identify cluster structure, *in* 'ACM SIGMOD Int. Conf. on Management of Data', Philadephia PA.

Asimov, D. (1985), 'The grandtour: A tool for viewing multidimensional data', *SIAM journal of Science and Stat. Comp.* **6**, 128–143.

Asimov, D. & Buja, A. (1995), 'Grand tour and projection pursuit', *Journal of Computational and Graphical Statistics* **4**(3), 155–172.

Barass, S. (1995), Personify: a toolkit for perceptually meaningful sonification, *in* 'Australian Computer Music Conference ACMA'95'.

Brin, S., Motwani, R., Ullman, J. D. & Tsur, S. (1997), 'Dynamic itemset counting and implication rules for market basket data', *SIGMOD Record (ACM Special interest group on the Management of Data)* **26**(2), 255–276.

Brin, S. & Page, L. (1998), 'Dynamic data mining: Exploring large rule spaces by sampling'.

Bruss, I. & Frick, A. (1996), Fast interactive 3-d graph visualization, *in* 'Proceeding of Graph Drawing', Springer Verlag, pp. 324–349.

Buxton, W. (1986), Chunking and phrasing and the design of human computer dialogues, *in* 'Proc. of the IFIP 10th World Computer Congress', pp. 475–480.

Carr, D., LittleField, R., Nicholson, W. & Littlefield, J. (1987), 'Scatterplot matrices for large n', *JASA* **82**(398), 424–436.

Ceglar, A., Roddick, J. F., Calder, P. & Rainsford, C. P. (2003), 'Visualising hierarchical associations', *Knowledge and Information Systems (to appear).*

Chapman, P., Kerber, R., Clinton, J., Khabaza, T., Reinartz, T. & Wirth, R. (1999), The crisp-dm process model, Discussion paper, CRISP-DM Consortium.

Chu, H. K. & Wong, M. H. (1998), Interactive data analysis on numeric-data, *in* '1999 Int. Symp. Database Engineering and Applications', Montreal Canada.

Dix, A., Finlay, J., Abowd, G. & Beale, R. (1998), *Human Computer Interaction*, Prentice Hall Europe, Hemel Hempstead U.K.

Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, *in* '2nd Int. Conf. on Knowledge Discovery and Data Mining', Portland, Oregon, pp. 226–231.

Faloutsos, C., & Lin, K.-I. (1995), Fastmap: A fast algorithm for indexing data-mining and visualization of traditional and multimedia datasets, *in* '1995 ACM SIGMOD', Vol. 24, ACM Press, San Jose, CA, USA, pp. 163–174.

Fasulo, D. (1999), An analysis of recent work on clustering algorithms, Technical Report 01-03-02, Department of Computer Science & Engineering, University of Washington.

Fox, E. A. (2001), 'Information storage and retrieval lecture notes'.

Garner, W. R., Hake, H. & Erikson, C. W. (1956), 'Operationism and the concept of perception', *Psychological Review* **63**, 149–159.

Gross, M. H., Sprenger, T. C. & Finger, J. (1997), Visualizing information on a sphere, *in* 'Information Visualization 97", Phoenix, Arizona.

Han, J., Kamber, M. & Tung, A. K. H. (2001), Spatial clustering methods in data mining: A survey, *in* H. Miller & J. Han, eds, 'Geographic Data Mining and Knowledge discovery', Taylor and Francis.

Hao, M. C., Dayal, U., Hsu, M., Sprenger, T. & Gross, M. H. (2001), Visualization of directed associations in e-commerce transaction data, *in* 'Proceedings of VisSym'01, Joint Eurographics - IEEE TCVG Symposium on Visualization', IEEE Press, Ascona, Switzerland, pp. 185–192.

Hartigan, J. (1975), *Clustering Algorithms*, John Wiley & Sons, New York U.S.A.

Hilderman, R. J. & Hamilton, H. J. (1999), Knowledge discovery and interestingness measures: A survey, Technical Report CS 99-04, Department of Computer Science, University of Regina.

Hinneburg, A. & Keim, D. A. (1999), A:(tutorial) clustering techniques for large data sets: From the past to the future, *in* 'SIGMOD Conference', Philadelphia.

Hinneburg, A., Keim, D. A. & Wawryniuk, M. (1999), 'Hd_eye: Visual mining of high dimensional data', *IEEE Computer Graphics and Applications* **19**(5), 22–31.

Hofman, H., Siebes, A. P. & Wilhelm, A. F. (2000), Visualizing association rules with interactive mosaic plots, *in* 'KDD 2000', ACM, Boston, MA USA, pp. 227–235.

Höppner, F. (2002), Discovery of core episodes from sequences – using generalization for defragmentation of rule sets, *in* 'Pattern Detection and Discovery in Data Mining, LNAI', Vol. 2447, Springer, London, England, pp. 199–213.

Jain, A. K. & Dubes, R. C. (1988), *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs New Jersey U.S.A.

John, Rosenbloom, P. & Newell, A. (1985), A theory of stimulus-response compatibility applied to human computer interaction, *in* 'Proceedings of CHI'85', pp. 213–230.

Kim, S.-S., Kwon, S. & Cook, D. (2000), 'Interactive visualization of hierarchical clusters using mds and mst', *Metrika* **51**(1), 39–51.

Klemettinen, M., Mannila, H., Ronkainen, T. & Verkano, A. (1994), Finding interesting rules from large sets of discovered association rules, *in* N. R. Adam, B. K. Bhargava & Y. Yesha, eds, 'Third International Conference on Information and Knowledge Management (CIKM'94)', ACM Press, Gaitherburg Maryland USA, pp. 401–407.

Klemettinen, M., Mannila, H. & Toivonen, H. (1997), A data mining methodology and its application to semi-automatic knowledge acquisition, *in* 'Proceedings of the 8th International Workshop on Database and Expert Systems Applications', IEEE Press, pp. 67–677.

Klemmer, E. T. & Frick, F. C. (1953), 'Assimilation of information from dot and matrix patterns', *Experimental Psychology* **45**, 15–19.

Koedinger, K. R. (1992), Emergent properties and structural constraints: Advantages of diagrammatic representations for reasoning and learning, *in* 'AAAI Spring Symposia on Reasoning with Diagrammatic Representations', Stanford University.

Koperski, K. & Han, J. (1995), Discovery of spatial association rules in geographic information databases, *in* '4th Int'l Symp. on Large Spatial Databases (SSD'95)', Portland, Maine, pp. 47–66.

Lesh, N., Marks, J. & Patrignani, M. (2000), Interactive partitioning, Technical report, Mitsubishi Electronic Research Laboratory.

Li, S., Vel, O. d. & Coomans, D. (1995), Comparative performance analysis of non-linear dimensionality reduction methods, Technical report, James Cook University.

Mayo, E. (1945), *The Social Problems of an Industrialized Society*, Harvard University Press, Boston.

Miller, G. A. (1956), 'The magic number seven,plus or minus two: Some limits on our capacity for processing information', *Psychological Review* **63**, 81–97.

Nag, B., Deshpande, P. M. & DeWitt, D. J. (1999), Using a knowledge cache for interactive discovery of association rules, *in* 'KDD-99', ACM Press, San Deigo Ca USA, pp. 244–253.

Nascimento, H. A. & Eades, P. (2001), Interactive graph clustering based upon user hints, *in* 'Proc. of the Second Int. Workshop on Soft Computing Applied to Software Engineering', Enschede, The Netherlands, p. 7.

Padmanabhan, B. & Tuzhilin, A. (1996), Pattern discovery in temporal databases: a temporal logic approach, *in* E. Simoudis, J. Han & U. Fayyad, eds, 'Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining', AAAI Press, Portland, Oregon, pp. 351–354.

Piatetsky-Shapiro, G. (1991), Discovery, analysis and presentation of strong rules, *in* W. J. Frawley & G. Piatetsky-Shapiro, eds, 'Knowledge Discovery in Databases', AAAI - MIT Press.

Raghavan, V. & Hafez, A. (2000), Dynamic data mining, *in* 'Industrial and Engineering Applications of Artificial Intelligence and Expert Systems', Lecture Notes in Computer Science, Springer-Verlag, pp. 220–229.

Rainsford, C. P. & Roddick, J. F. (1999), Adding temporal semantics to association rules, *in* J. Zytkow & J. Rauch, eds, 'Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD'99', Springer Verlag, Prague, Czech Republic, pp. 504–509.

Rainsford, C. & Roddick, J. (2000), Visualisation of temporal interval association rules, *in* 'Proc 2nd International Conference on Intelligent Data Engineering and Automated Learning', Shatin, N.T. Hong Kong.

Rasmussen, E. (1992), Clustering algorithms, *in* W. Frakes & R. Baeze-Yates, eds, 'Information Retrieval: Data Structures and Algorithms', Prentice-Hall, Englewood Cliffs, New Jersey U.S.A.

Ribarsky, W., Katz, J., Jiang, F. & Holland, A. (1999), 'Discovery visualization using fast clustering', *IEEE Computer Graphics and Applications* **19**(5), 32–39.

Roddick, J. & Ceglar, A. (2001), 'Fido'.

Sander, J., Ester, M., Kriegel, H.-P. & Xu, X. (1998), 'Density-based clustering in spatial databases: The algorithm gdbscan and its applications', *Data Mining and Knowledge Discovery* **2**(2), 169–194.

Shneiderman, B. (1996), The eyes have it: A task by data type taxonomy for information visualization, *in* '1996 IEEE Symp. on Visual Languages', Boulder Colorado, pp. 336–343.

Silberschatz, A. & Tuzhilin, A. (1996), 'What makes patterns interesting in knowledge discovery systems?', *IEEE Transactions on Knowledge and Data Engineering* **8**(6), 970–974.

Sprenger, T. C., Brunella, R. & Gross, M. H. (2000), H-blob: A hierarchical visual clustering method using implicit surfaces, *in* 'IEEE Visualisation 2000', Salt Lake City, UTAH, USA.

Srikant, R., Vu, Q. & Agrawal, R. (1997), Mining association rules with item constraints, *in* D. Eckerman, H. Mannila, D. Pregibon & R. Uthursamy, eds, '3rd Int. Conf. on Knowledge Discovery and Data Mining', AAAI Press, Newport Beach, C.A., U.S.A., pp. 67–73.

Sundaresh, R. S. & Hudak, P. (1991), Incremental computation via partial evaluation, *in* '18th Annual ACM Symp. on POPL', ACM Press, New York, pp. 1–13.

Toivonen, H. (1996), Sampling large databases for association rules, *in* T. Vijayaraman, Alejandro, P.Buchmann, C. Mohan & N. Sarda, eds, 'Proceedings of the 22nd International Conference on Very Large Data Bases', Morgan Kaufman, Mumbia(Bombay), India, pp. 134–145.

Tung, A. K. H., Hou, J. & Han, J. (2001), Spatial clustering in the presence of obstacles, *in* 'Int. Conf. on Data Engineering', Heidelberg, Germany.

Ward, M. O. (1994), Xmdvtool : Integrating multiple methods for visualising multivariate data, *in* R. D. Bergeron & A. E. Kaufman, eds, 'IEEE Visualization '94', pp. 326–333.

Wills, G. (1998), An interactive view for hierarchical clustering, *in* 'Information Visualization '98', Raleigh, North Carolina, USA.

Wong, P. C., Cowley, W., Foote, H., Jurrus, E. & Thomas, J. (2000), Visualizing sequential patterns for text mining, *in* 'IEEE Sym. on Information Visualization 2000', IEEE Press, Salt Lake City, Utah.

Wong, P. C., Whitney, P. & Thomas, J. (1999), Visualizing association rules for text mining, *in* 'Proceedings of IEEE Symposium on Information Visualization'99', IEEE Computer Society Press, Los Alamitos, California,USA, pp. 120–124.

Wrobel, S., Wettschereck, D., Verkamo, I., Siebes, A., Mannila, H., Kwakkel, F. & Klosgen, W. (1996), User interactivity in very large scale data mining, *in* W. Dilger, M. Schlosser, J. Zeidler & A. Ittner, eds, 'FGML-96 Annual Workshop of the GI Special Interest Group Machine Learning', TU Chemnitz-Zwickau, pp. 125–130.

Xiao, Y. & Dunham, M. H. (2001), Interactive clustering for transaction data, *in* Y. Kambayashi, W. Winiwarter & M. Arikawa, eds, 'Third Int. Conf. Data Warehousing and Knowledge Discovery', Springer Verlag, Munich, Germany, pp. 121–130.

Yang, L. (2000), n23tool: A tool for exploring large relational datasets trough dynamic projections, *in* '9th Int. Conf. on Information and Knowledge Management CIKM 2000', ACM Press, pp. 322–327.

Young, F. (1987), *Multidimensional Scaling: history, theory and applications*, Lawrence Erlbaum Associates, Hillsdale New Jersey U.S.A.

Texture Analysis via Data Mining

Martin Heckel

Brno University of Technology, Faculty of Information Technology, Bozetechova 2,
612 66 Brno, Czech Republic
heckel@fit.vutbr.cz

Abstract. This paper deals with an idea of the use of data mining approach in texture analysis. A new method based on association rules is proposed. This approach extracts texture primitives from an image texture and describes mutual relationships between these primitives. Within this method, a technique for feature vector construction without a priori knowledge of textures different from the analyzed one is presented. The Fisher criterion was used to measure an ability of the proposed method for successful discrimination of pairs of textures. This work provides also comparison of the proposed technique with a wide-used wavelet texture features.

1 Introduction

Texture is one of the most important property of image data used in many application domains, for instance medical image processing, industrial inspection, remote sensing, document processing etc. Texture description usually as a part of feature vector has also appeared in the area of content based image retrieval for image browsing, searching and retrieval [1], [2], [3].

In the last thirty years, many texture analysis techniques have been developed. They can be divided into four main classes [4]: statistical methods, signal processing methods, model based approaches and geometrical approaches. Currently, signal processing methods, mainly wavelet decomposition (e.g. [5], [3]) and Gabor filters (e.g. [1], [2]) belong to the most popular techniques.

This work focuses on the other class, on geometrical properties of image texture. Herein, a texture is considered as being composed of texture primitives and an arrangement of these primitives is described by certain rules.

We attempt to capture texture arrangement properties by data mining approach, particularly by association rules. The best-known application domain, where association rules have been successfully used, is market basket analysis. Here, the goal is to identify items frequently purchased together. We utilize association rules to determine texture primitives occurring together in a texture.

The idea of using association rules during texture analysis was published by J. A. Rushing et al. in [6]. We extended this idea by the processing of image texture at the primitive level. We also developed a technique of the feature vector construction without a priori knowledge of textures different from the analyzed one.

The paper is organized as follows. The next section describes a proposed texture analysis method. Section 3 deals with separability measurements on pairs formed from some Brodatz textures [7]. Section 4 summarizes the results of the proposed approach.

2 Texture Description Method

We developed a new method of texture image description based on association rules for analyzing grey level texture images. This approach consists of three following steps:

1. *Extraction and description of texture primitives* – The goal is to extract and describe texture primitives in an image texture (Section 2.1).
2. *Mining of association rules* – In this step, the transaction databases are constructed and association rules are mined (Section 2.2).
3. *Feature vector construction* – The goal is to compute a feature vector without a priori knowledge of a set of textures, different from the analyzed one (Section 2.3).

In this method, the image texture is not analyzed at the pixel level (as in [6]), but at higher one, at the image primitive level. Obtained association rules describe texture primitive arrangement in a non-deterministic way. This data mining approach finds frequently occurring local structures at the primitive level in an analyzed image.

2.1 Extraction and Description of Texture Primitives

In this stage, we extract texture primitives from the analyzed image texture. As a texture primitive, we understand a connected region composed of pixels at the same grey level. Thus, the number and the size of extracted primitives depend on a texture type. For instance, we obtain a big number of small primitives in case of a microtexture.

Once the primitives are identified, we describe them by their own grey levels and the geometrical properties. Then, the set of primitive descriptions is the output of this stage.

In order to extract texture primitives, it is necessary to quantize every image to a certain number of grey levels. We chose the well-known statistical clustering algorithm k-means [8] to perform an adaptive quantization to a certain small number of levels.

It remains texture primitive identification. For this task, we use connected component analysis. The connected component is a set of image pixels, which share a certain property. In this component, there exists a path between every pair of pixels. Every connected component represents one texture primitive, whereby a grey level of pixels is used as a shared property. We use a row-by-row labeling algorithm proposed by A. Rosenfeld and J. L. Phaltz [9].

Every identified primitive is described by its properties:

1. The grey level.
2. The area of primitive (number of pixels)

$$m_{00} = \sum_x \sum_y f(x,y). \tag{1}$$

3. The centroid (x_t, y_t):

$$x_t = \frac{\sum_x \sum_y x f(x,y)}{m_{00}}, \tag{2}$$

$$y_t = \frac{\sum_x \sum_y y f(x,y)}{m_{00}}, \tag{3}$$

where $f(x,y)$ denotes the analyzed image texture. This function returns 1, if a pixel (x,y) belongs to the primitive, otherwise 0.

2.2 Mining of Association Rule

This part of a proposed method focuses on the mining of association rules, which characterize mutual relationships between texture primitives. The set of primitive descriptions together with the primitive adjacency information are the input of this stage. For the set of primitive descriptions, a transaction database is constructed based on primitive adjacency (Subsection *Transaction Database Construction*). In order to mine the association rules from a transaction database, we used the well-known Apriori algorithm (SubSection *Association Rule Discovery*). Thus, the set of association rules form the output of this stage. SubSection *Association Rules* provides brief introduction to association rules.

Association Rules. Association Rules was introduced by R. Agrawal et al. [10] as one knowledge type, which can be discovered in databases. Association rules are non-deterministic rules, which capture mutual relationship between data stored in a database. Initially, these rules are designed for transaction databases. Such a database consists of transactions, where a *transaction* is a set of *items* related together.

An association rule is an implication of the form $A \Rightarrow B$, where A, B are disjoint sets of items. This rule has two probability values: support s and confidence c.

Support s of the rule expresses the percentage of transactions in the whole database, which contain both the sets A and B. Confidence c means the percentage of transactions containing both A and B from all transaction containing A. This can be written as follows:

$$s(A \Rightarrow B) = P(A \cup B), \tag{4}$$

$$c(A \Rightarrow B) = P(B|A) = \frac{P(A \cup B)}{P(A)}. \quad \textbf{(5)}$$

A typical example of the use of association rules is market basket analysis, where a transaction database contains information about costumer purchases. Each transaction consists of items within one purchase. Then, the interpretation of association rule $A \Rightarrow B$ can be: "If a costumer buys items A (for instance bread and butter), then he also buys items B (for example milk)."

Transaction Database Construction. A transaction database is constructed based on a primitive adjacency. In this process, the central primitives play an important role. *The central primitive* is each non-border texture primitive completely surrounded by its immediate neighbours.

For every central primitive, *a transaction* is created, which contains this primitive with all immediately neighbouring ones. Thus, the cardinality of database is the number of central primitives. Fig. 2 depicts one example of such a transaction.

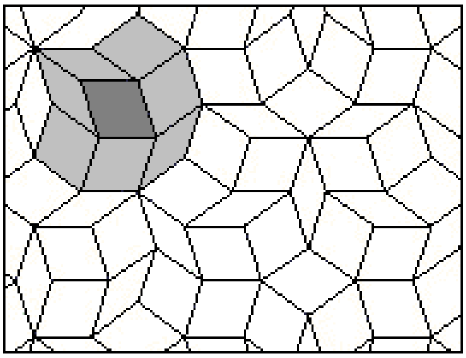


Fig. 2. The example of a transaction – the central primitive is dark grey and its immediate neighbouring primitives are grey

Each primitive in a transaction is represented by one vector containing its grey level, its area and the following two values describing relationship with its central primitive:

1. *The centroid distance*,
2. *The angle* between the line traversing the both centroids and the axis of the coordinate system.

In Fig. 3, a draft of the both values representing inter-primitive relationship can be seen. The centroid distance and the angle serve to describe local image structures at the primitive level.

Because of the numeral nature of information stored in the transaction, it is necessary to quantize these values in the whole transaction database. For instance, the angle can be quantized to 4, 8 or 16 levels etc.

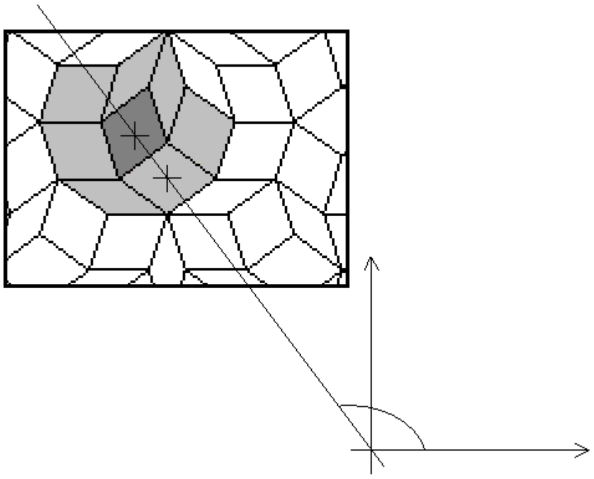


Fig. 3. A relationship between central (dark grey) and neighbouring primitive (grey)

Association Rule Discovery. We selected the well-known Apriori algorithm [10] to process a transaction database (for further details of this algorithm see e.g. [11]). The output of this algorithm is the set of frequent itemsets, which are sets of items satisfying the condition of minimum support value.

From this set of frequent itemsets, association rules, which fulfill the minimum confidence condition, are generated according to the following widely used algorithm:

1. For each frequent itemset *l*, all its non-empty subsets are generated.
2. For every generated subset *A*, create a rule *A* ⇒ *(l - A)*, if its confidence $\frac{s(l)}{s(A)}$ is equal or greater than the minimum confidence value.

Note, that the itemset *l* of cardinality *k* generates 2^k candidate association rules. Based on the number of generated rules for tested texture images, we limited the right and the left side cardinalities of rules to the following rule cardinality types:

1. $(\,) \Rightarrow (\,)$,
2. $(\,)(\,) \Rightarrow (\,)$,
3. $(\,)(\,) \Rightarrow (\,)(\,)$,
4. $(\,)(\,)(\,) \Rightarrow (\,)$,

where the symbol ( ) denotes one item.

2.3 Feature Vector Construction

In this stage, a feature vector representing an analyzed image texture is computed. The set of association rules is the input and one feature vector is the output. The feature vector is constructed in the following steps:

1. Input set of association rules is divided into four groups of the same cardinality type (see Section 2.2).
2. The rules in every group are sorted by the support and by the confidence.
3. In every sorted group, only a certain number of first rules is kept. We used only the first two rules from every group.
4. The length of the feature vector is divided into four partitions for every association rule groups.
5. Each association rule is stored to the feature vector in this way: First, the values of the items on the right side and second, the values of the items of the left side (items of both sides of the rule are presorted in lexicographic order).
6. If some rule group has less than two rules, the space for this rule in the feature vector is completed by zeros.

Using this algorithm, a 104-dimensional feature vector is composed (4 values per one item). The feature vector is constructed without a priori knowledge of textures different from analyzed one (in contrast with [6]) and it has a fixed size and form. Such a form of the feature vector is suitable e.g. for content based image retrieval or for the data mining tasks performed over the entire image database.

3 Discriminatory Ability of the Method

In order to analyze properties of the proposed method, the Fisher criterion (Section 3.1) which determine a discriminatory ability for a pair of textures is used. We performed a number of experiments over textures from the Brodatz album [7] and provide a comparison with wide-used wavelet texture features (Section 3.2).

3.1 Fisher criterion

In this work, the Fisher criterion is utilized to measure of proposed method ability to discriminate one texture from the other. The Fisher criterion expresses a separability of two clusters containing feature vectors, which characterize an analyzed pair of textures. In the area of texture analysis, this method was first proposed by P. Kruizinga and N. Petkov [12]. More details of the Fisher criterion can be found e.g. in [15], [16].

The Fisher linear discriminant function can be written in the following form:

$$y = (\vec{\mu}_1 - \vec{\mu}_2)^T S^{-1} \vec{x} , \qquad \textbf{(6)}$$

where $\vec{x}$ is feature vector obtained from an analyzed texture, y is its projection, $\vec{\mu}_1$, $\vec{\mu}_2$ are the means of the two clusters and S is a pooled covariance matrix

$$S = \frac{1}{N_1 + N_2 - 2}\left(\sum_{j=1}^{N_1}\left(\vec{x}_{1;j} - \vec{\mu}_1\right)^T\left(\vec{x}_{1;j} - \vec{\mu}_1\right) + \sum_{j=1}^{N_2}\left(\vec{x}_{2;j} - \vec{\mu}_2\right)^T\left(\vec{x}_{2;j} - \vec{\mu}_2\right)\right). \quad \textbf{(7)}$$

N_1 is the number of selected (e.g. randomly) feature vectors $\vec{x}_{1;j}$ from the first cluster and N_2 is the number of selected vectors $\vec{x}_{2;j}$ from the second one.

The Fisher linear discriminant function (6) realizes the projection of the n-dimensional feature space on one-dimensional space (on a line). The dimensionality of a feature space is given by the number of features, contained in feature vectors. This projection maximizes the Fisher criterion

$$f = \frac{|\eta_1 - \eta_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}}, \quad \textbf{(8)}$$

where η_1, η_2 and σ_1^2, σ_2^2 are the means and the variances characterizing the distribution of the projected feature vectors of the two clusters. The means η_1, η_2 are yielded by the projections:

$$\begin{aligned} \eta_1 &= (\vec{\mu}_1 - \vec{\mu}_2)^T S^{-1}\vec{\mu}_1 , \\ \eta_2 &= (\vec{\mu}_1 - \vec{\mu}_2)^T S^{-1}\vec{\mu}_2 . \end{aligned} \quad \textbf{(9)}$$

The Fisher criterion measures the distance between the means of the both clusters relative to the sum of their variances, thus to their compactness. That is why, it is suitable to measure separability of two clusters and also to express the capability of texture description methods for successful discrimination of pairs of different textures.

3.2 Experimental schema and results

We performed a number of experiments with the proposed method over textures from the Brodatz album [7] under following conditions:

Six textures D34, D49, D75, D84, D95, D104 that were used are shown in Fig. 4. From every texture, one thousand of feature vectors were extracted by analysing texture in a 128 × 128 pixel window moving through the analysed image texture. All positions of this moving window cover the whole analysed texture uniformly. Thus, these thousand vectors for every given texture form a respective cluster.

The texture analysis method was configured as follows: An input image (128 × 128) – content of the moving window is quantized into 4 levels. Transaction database is quantized: 1. the grey level into 16 levels, 2. the area into 5 levels, 3. the distance into 5 levels and 4. the angle into 8 levels. For association rule mining, the support is set to 0.02 and the confidence to 0.2.

We measured the Fisher criterion for every possible pair of the selected textures. The obtained results can be seen in Table 1. There are also the results of a wavelet features there to provide a comparison between this new approach and one of the wide-used texture features.

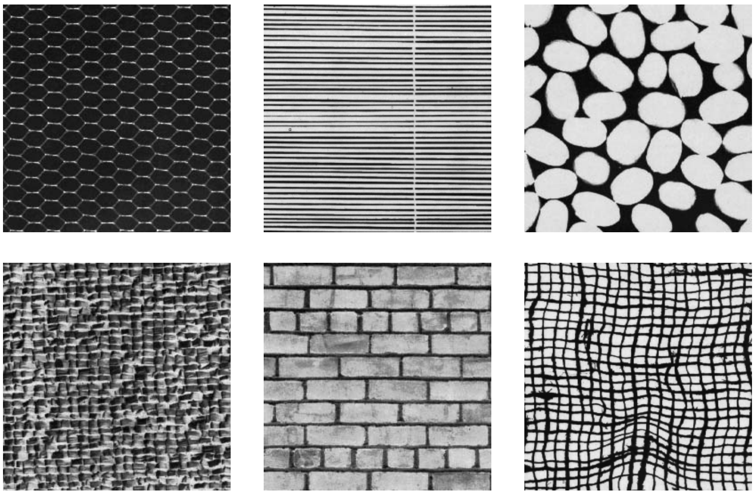


Fig. 4. Textures from the Brodatz album [7] used in experiments: D34, D49, D75, D84, D95, D104 (left to right, top to bottom)

For this task, wavelet features similar to one mentioned in [5] were utilized. The manner of obtaining the cluster of feature vectors stays the same as for the association rule features, only the moving window size is changed to 32 × 32 pixel. Over input image (content of the moving window), three-level tree-structured wavelet decomposition using Haar filters is constructed (for further details see e.g. [13], [14]). From every of the ten obtained subbands, mean and standard deviation is computed. These values form a 20-dimensional feature vector.

Table 1. The Fisher criterion values computed from wavelet features (WT) and association rule features (AR)

Texture	D34	D49	D75	D84	D95	D104
D34	-	AR: 17.53	AR: 7.25	AR: 115.09	AR: 27.08	AR: 4.20
	-	WT: 27.16	WT: 12.30	WT: 9.08	WT: 11.10	WT: 26.45
D49		-	AR: 3.38	AR: 45.78	AR: 15.07	AR: 50.01
		-	WT: 13.17	WT: 18.19	WT: 10.51	WT: 16.85
D75			-	AR: 1.26	AR: 2.09	AR: 2.31
			-	WT: 8.29	WT: 4.55	WT: 8.71
D84				-	AR: 1.37	AR: 2.16
				-	WT: 5.69	WT: 10.51
D95					-	AR: 2.08
					-	WT: 10.81
D104						-
						-

Many results of the proposed method are a little worse than wavelet feature, but some results overcome them significantly. It is important to notice that this new method is still at the beginning of its development and after some modification (e.g. better extraction of primitives) it could yield much better results.

4 Conclusion

In this paper, a new method of a texture analysis based on the association rules has been proposed. This approach processes an image texture at primitive level. Within this new method, a technique for the feature vector construction without a priori knowledge of textures different from analyzed one (in contrast with [6]) was presented. Such a feature vector is suitable e.g. for content based image retrieval or for the data mining tasks performed over the entire image database. The discriminatory capability of the method was tested on some textures selected from the Brodatz album [7].

Currently, we try to extend this approach by the multiresolution analysis, thus an image texture processing in several resolution levels. Our future work will focus on the research on the influence of an adjacency level extension upon the rule discovery process. The adjacency level extension means that a transaction will not only contain immediate neighbours of a central primitive, but even neighbours of these neighbours etc. We would also like to define the texture primitive description with more precision by using its shape properties.

Acknowledgements

This research has been supported by the Ministry of Education, Youth and Sports, grant No. FR830/2003/G1 and by the Czech Grant Agency grant No. GA102/02/0507.

References

1. Manjunath, B. S., Ma, W. Y.: Texture Features for Browsing and Retrieval of Image Data. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, 1996, pp. 837-842
2. Wu, P., Manjunath, B. S., Newsam, S., Shin, H. D.: A Texture Descriptor for Browsing and Similarity Retrieval. Journal of Signal Processing: Image Communication, Vol. 16, Issue 1-2, 2000, pp. 33-43
3. Wang, J. Z., Wiederhold, G., Firschein, O., Wei, S. X.: Wavelet-based image indexing techniques with partial sketch retrieval capability. Proceedings IEEE Forum on Research and Technology Advances in Digital Libraries (ADL'97), Washington D.C., IEEE, 1997, pp. 13-24
4. Tuceryan, M., Jain, A. K.: Texture Analysis, In: Handbook of Pattern Recognition and Computer Vision (2nd edition), World Scientific Publishing Co., 1998, pp. 207-248

5. Smith, J. R., Chang, S.: Transform Features For Texture Classification and Discrimination in Large Image Databases. Proceedings of the IEEE International Conference on Image Processing, 1994, pp. 407-411
6. Rushing, J. A., Ranganath, H. S., Hinke, T. H., Graves, S. J.: Using Association Rules as Texture Features. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No 8, 2001, pp. 845-858
7. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. New York, Dover Publication, 1966
8. Hartigan, J. A., Wong, M. A.: A k-means Clustering Algorithm. Applied Statistics, 28, 1979, pp. 100-108
9. Rosenfeld, A., Pfaltz, J. L.: Sequential Operations in Digital Image Processing. Journal of the Association for Computing Machinery, Vol. 13, 1966, pp. 471-494
10. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, USA, 1993
11. Han, J., Kamber, M.: Data Mining Techniques And Concepts. Morgan Kaufmann Publishers, 2000
12. Kruizinga, P., Petkov, N.: Nonlinear Operator for Oriented Texture. IEEE Transactions on Image Processing, Vol. 8, No 10, 1999, pp. 1395-1407
13. Fournier, A.: Wavelets and their Applications in Computer Graphics. SIGGRAPH'95 Course Notes, 1995
14. Heckel, M.: Texture Analysis for Content Based Image Retrieval, In: Proceedings of 5th International Conference ISM, Ostrava, CZ, MARQ, 2002, pp. 37-44
15. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification (second edition). John Wiley and Sons, 2001
16. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Academic Press, 1999

Clustering Time Series from Mixture Polynomial Models with Discretised Data

A. J. Bagnall, G. Janacek, B. de la Iglesia and M. Zhang

University of East Anglia
Norwich
England NR4 7TJ
Contact email: ajb@cmp.uea.ac.uk

Abstract. Clustering time series is an active research area with applications in many fields. One common feature of time series is the likely presence of outliers. These uncharacteristic data can significantly effect the quality of clusters formed. This paper evaluates a method of overcoming the detrimental effects of outliers. We describe some of the alternative approaches to clustering time series, then specify a particular class of model for experimentation with k-means clustering and a correlation based distance metric. For data derived from this class of model we demonstrate that discretising the data into a binary series of above and below the median improves the clustering when the data has outliers. More specifically, we show that firstly, discretisation does not significantly effect the accuracy of the clusters when there are no outliers and secondly it significantly increases the accuracy in the presence of outliers, even when the probability of outliers is very low.

1 Introduction

The clustering of time series has attracted the interest of researchers from a wide range of fields, particularly from statistics [37], signal processing [17] and data mining [12]. This has resulted in the development of a wide variety of techniques designed to detect common underlying structural similarities in time dependent data. A review of some of the work in the field is given in Section 2. These techniques have been applied to data arising from many areas, for example: web mining [13, 5]; finance and economics [43, 20]; medicine [23]; meterology [8]; speech recognition [17, 28]; gene expression analysis [18, 6] and robotics [45]. Our interest is primarily motivated by the desire to be able to detect common patterns of behaviour in bidding strategies of agents competing in markets in order to quantify adaptive agent performance [4].

Clustering is an unsupervised learning task, in that the learning algorithm is not informed whether the assignment of a data to a cluster is correct or not. For background into clustering see [25]. There are two main ways clustering has been used with time series. Firstly, clustering can be applied to a single time series, frequently using a windowing system, to form different generating models of the single series [15]. Note that there is some controversy over the usefulness of this approach (see [27]). The second problem involves forming k clusters for m time series rather than from a single series. We are interested in the latter problem, which can be described as follows:

Given m time series, $S = \{s_1, \ldots, s_m\}$ of length $n_1, \ldots n_m$, the problem is to form k clusters or sets of time series, $C = \{C_1, \ldots, C_k\}$, so that the most "similar" time series are in the same cluster. k may or may not be known *a priori.* Also, cluster membership may be deterministic or probabilistic. To encompass both we can generalise the clustering task to assigning a probability distribution $p_s(j)$, to each time series which defines the probability that series s is in cluster C_j.

The obvious crucial question is what is meant by most "similar" time series. This is usually model and problem dependent, but can be generalised as follows. Suppose a distance function $d(a, b)$ is defined on the space of all possible series, D. A distance function $d(s_i, s_j) : D \times D \rightarrow \Re$ is a **metric** if it satisfies the four conditions

$$\begin{array}{ll} d(a,b) > 0 & \text{if } a \neq b \\ d(a,a) = 0 & \\ d(a,b) = d(b,a) & \\ & \\ d(a,c) \leq d(a,b) + d(b,c) \; \forall \; a,b,c \in A. & \end{array}$$

The distance function may have a domain that is the space of all time series or it may be embedded in a lower dimensional space formed through, for example, fitting a parameterised model to each series. Given a distance metric, the clustering task is to find the clusters that minimize the distance between the elements within each cluster and maximize the distance between clusters. This can be described by the introduction of a cost function. Suppose the cost for a cluster C_j is defined as

$$c_j = \sum_{a,b \in S} p_a(j) \cdot p_b(j) \cdot d(a,b).$$

The clustering problem for a given k is to find the partition that minimizes the total cost, $c = \sum_{j=1}^{m} c_j$. If k is not given, then some weighting function has to be included to encourage parsimonious clustering.

Clustering algorithms can be classified as two types, hierarchical methods and partitioning methods. Hierarchical methods involve calculating all distances then forming a dendrogram by using a linkage method such as nearest or furthest neighbour. The second approach involves partitioning using an iterative algorithm that attempts to optimize cluster assignation based on minimizing a cost function.

The most commonly used partitioning method is the k-means algorithm [35]. This is an iterative local search method that attempts to minimize the distance within the clusters. The EM (Expectation Maximizing) algorithm is a generalisation of k-means [16]. Instead of assigning a data to a particular cluster, a probability of membership of all clusters is maintained. In addition to a centroid recording the means of the cluster, the EM algorithm also records a covariance matrix. Both k-means and EM have been used for clustering time series. A comprehensive description of clustering algorithms can be found in [25].

The aim of this research is to demonstrate that discretizing the data can make clustering time series more robust to the presence of outliers without significantly decreasing accuracy when outliers are not present or highly unlikely. Our initial approach to this is to define a simple class of underlying model similar to that used by other researchers, then measure the effect on performance of the introduction of outliers. Section 2 provides background into some of the research into clustering time series. Section 3 describes experimentation on simulated data with a standard clustering technique (k-means) and a simple distance metric based on correlation and provides evidence of a scenario under which clipping allows the optimal clusters to be found. Section 4 summarises the results and describes the next stages of this research.

2 Related Research

Most research assumes some underlying form of the model and performs the clustering based on this assumption. [12] makes the case for a model based, or *generative* approach, which can be classified into three broad categories, discussed in Section 2.1: AutoRegressive Moving Average (ARIMA) models, Markov Chain (MC) and Hidden Markov models (HMM) and polynomial mixture models. Approaches that do not assume a model form, often called similarity based approaches, are summarised in Section 2.2. Focardi [20] provides good background material on clustering time series.

2.1 Model Based Approaches

ARIMA Models: The main approach of statistics researchers to the problem of clustering time series is to assume the underlying models are generated by an ARIMA process [10]. The clustering procedure usually involves:

1. fitting a model to each time series;
2. measuring distance between fitted models;
3. clustering based on these distances.

This approach is adopted by Piccolo [44], Maharaj [36, 37] and Baragona [7]. Tong and Dabas [48] cluster different ARIMA models that have been fitted to the same data set, but the techniques used are also relevant to clustering models from different data sets.

Fitting the model requires the estimation of the structure and parameters of an ARIMA model. Structure is either assumed to be given or estimated using, for example, Akaike's Information Criterion or Schwartz's Bayesian Imformation Criterion [10]. Parameters are commonly fitted using the generalised least squares estimators. An order m ARIMA model can be fully specified by a set of parameters

$$\pi = \{\pi_1, \pi_2, \ldots \pi_m\}.$$

Some of the research based on assuming an ARIMA model derives the distance function from the differences between the estimates of these parameters. Piccolo [44] uses the Euclidean distance between the parameters,

$$d(\pi_a, \pi_b) = \left(\sum_{i=1}^{\infty} (\pi_{i,a} - \pi_{i,b})^2 \right)^{\frac{1}{2}}.$$

Maharaj [36, 37] adjusts her measure of distance between parameter sets by the correlation matrix estimated by the least squares to allow for dependent time series. She uses the resulting statistic as a test for
H_0: $\pi_a = \pi_b$ vs
H_1: $\pi_a \neq \pi_b$
A function of the p-value of this test is used as a similarity mesasure (low p-values making common cluster membership unlikely). An alternative approach to forming a distance function, used in [48, 7], is to base distance on the residuals of the model. Let e_a be the residuals for model π_a and $\rho_{a,b}(i)$ be the correlation between the residuals e_a and e_b. Tong [48] uses the sample correlation coefficient with lag 0, denoted $\rho(0)$,

$$d(a, b) = 1 - |\rho(a, b)(0)|$$

Baragona [7] uses a distance function that scales the zero lag correlation by the sum of the lagged correlations,

$$d(a, b) = \sqrt{\frac{(1 - \rho^2_{a,b}(0))}{\sum_{i=1}^{m}(\rho^2_{a,b}(i))}}$$

This function was proposed in [9], although in this case it was used on the time series rather than the residuals of the models. A variety of clustering techniques have been employed. For example, principle coordinates and multidimensional scaling were used in [48, 44], hierarchical clustering with average linkage was used in [36] and with single linkage, complete linkage and Ward's method in [48] and heuristic search techniques (genetic algorithms, simulated annealing and tabu search) were employed in [7].

Hidden Markov Models (HMM): An alternative approach to the problem has been adopted by researchers in speech recognition and machine learning. Instead of an ARMA model, it is common to assume that the underlying generating models for each cluster can be accurately described as a markov chain (MC) or hidden markov model (HMM). A HMM is a set of unobserved states, each of which has an associated probability distribution for the random variable being observed, and a transition matrix that specifies the probability of moving from one state to another on any time step. A first-order HMM is an HMM where T is dependent only on the previous state. A MC also involves a set of states, except that the states correspond to the set of observable values of the random variable (and hence are not hidden).

For both approaches, the clustering algorithm generally involves the following steps:

1. form an initial estimate of cluster membership;
2. form HMM models based on membership;
3. while there is some improvement in models
   (a) adjust cluster membership;
   (b) reform models;

The clustering may be hierarchical or partitional. One key difference in technique between the ARIMA and the MC/HMM methods is that the ARIMA approach is to fit a model to each data before clustering, whereas most research into HMMs involves forming the cluster models on each iteration of the clustering algorithm.

MC models have been adopted by Ramoni *et al* [45] to model and cluster discrete series. Each state is associated with each value a data can take, and the problem becomes one of finding k transition matrices and identifying which series originates from which matrix. Their algorithm, called Bayesian Clustering by Dynamics (BCD), is a bottom up hierarchical agglomerative method, where distance between models is measured using the Kullback-Leiber distance.

Cadez *et al* [12] also use a MC model in the context of a generalised probabilistic EM-based framework for clustering. In [13] they apply the technique to web mining. Ridgeway [46] compares using EM against Gibbs resampling when clustering Markov processes.

Smyth [47] clusters using HMM by fitting a model to each series, then uses the log-likelihood as a distance for a hierarchical furthest neighbour technique. Parameters for a given model structure are estimated with the Baum-Welch procedure.

Oates *et al* [40, 39, 42, 41] fit k HMMs using the Viterbi algorithm to train HMM on greedily selected subsets of series. In [41] they set the initial clustering using Dynamic Time Warping. HMM are fitted to each cluster, a Monte Carlo simulation is conducted on each model and series that are empirically unlikely to have been observed from a model are removed from the cluster. The model is then retrained and the process repeated until no more series can be removed. It is then tested whether unassigned series can be placed into other clusters. If not, they form their own new clustering. They find that the hybridization of DTW and HMM forms better clusters than either approach alone on simulated data (which is also discretised) from models used in [47].

Zhong and Ghosh [53, 50–52, 49] use a model-based k-means clustering algorithm and a version of the EM algorithm. The also use a hierarchical model similar to that of [45], using HMM instead of MC models. Li and Biswas [32, 33, 30, 31, 34] propose a Bayesian HMM clustering methodology that includes determining the number of clusters and the structure of the HMM. Cadez, Gaffney and Smyth [12, 13] use HMM within the context of a generalised probabilistic framework. Alon *et al* [2] use the EM algorithm in HMM based clustering and assess the performance of EM in relation to k-means.

Polynomial Models Another approach is to assume the underlying model is a mixture of Polynomial functions. Gaffney and Smyth [21, 22] assume a mixture regression model. The EM algorithm with Maximum A Posteriori (MAP) estimates is used to estimate the cluster membership probabilities and weighted least squares used to fit the models. The technique is applied to simulated data, environmental data and video streaming data.

Bar-Joseph *et al* [6] adopt a mixture spline model for gene expression data, again using the EM algorithm in conjunction with least squares.

2.2 Model Free Approaches

Rather than assume a model form and base similarity on fitted parameter estimates, an alternative approach is to measure distance with the original or transformed data.

The simplest approach is to treat the time series as an N-dimensional vector and use the L_q Minkowsky distances, most commonly the Euclidean distance metric, L_2,

$$L_2 = (\sum_{i=1}^{N} |a_i - b_i|^2)^{\frac{1}{2}} \tag{1}$$

This measure is used by [1] in conjunction with fast fourier transforms. The main problem with using an L_q measure for time series similarity is that they are effected by the scale of the two time series, thus shape characteristics can be lost, (a further problem is that it is required that data be available for the same time steps, and this may not always be the case). [29] use a distance metric based on the Euclidean distance but introduces an extra set of shape parameters. An alternative is to use a metric that does capture the similarity in shape, for example one based on the correlation between the series. If we let $C(a, b)$ be the correlation between the series a and b, then Equation 2 is a metric, as demonstrated by Ormerod and Mounfield [43]. Similar metrics were used in [9].

$$d(a, b) = \sqrt{2(1 - C(a, b))} \tag{2}$$

Other researchers look for commonality measures based on common subsequences. For example [14] and [19] define measures based on common subsequences.

An alternative approach is to transform the data then use an associated metric. Approaches used include: time warping [41]; fast fourier transforms [1]; wavelet transforms [38] and piecewise constant approximation [26].

3 Experimentation

The results presented in this paper demonstrate that, for a certain class of underlying clustering model (described in Section 3.1), and with a particular experimental set up and clustering algorithm (outlined in Section 3.2), transforming the continuous time series into a discrete binary series

– does not significantly degrade clustering performance when there are no outliers; and
– significantly improves the quality of the final clusters found when there are outliers, even when the probability of an outlier is very low.

3.1 Experimental Model

We generate time series data from polynomial models of the form

$$m(t) = p(t) + \epsilon \tag{3}$$

where ϵ is $N(0, \sigma)$ and σ is constant. We assume the polynomial is order 1, i.e.

$$p(t) = a + b \cdot t$$

The purpose of these experiments is to demonstrate the robustness in the presence of outliers of using a discretised time series rather than the the continuous data for clustering. Hence, we add a further term to Equation 3 to model the effect of outliers. A continuous time series is assumed to be generated by a sequence of observations from the model

$$m(t) = a + b \cdot t + \epsilon + r \tag{4}$$

where

$$r = s \cdot x \cdot y.$$

s is a constant, $x \in \{0, 1\}$ and $y \in \{-1, 1\}$ are observations of independent random variables, X and Y, where X has density

$$f(x) = p^x(1 - p)^{1-x}$$

and Y has density

$$f(y) = \frac{1}{2}.$$

r is a *random shock* effect that can occur with probability p, and if it occurs it has the effect of either adding or subtracting a constant s to the data (with equal probability). A continuous time series is a sequence of observations from a model, now defined as

$$y(t) = p(t) + \epsilon + r \quad t = 1 \ldots n \tag{5}$$

A binary data series is generated by transforming a continuous series into series of above and below the median. If ϕ_y is the sample median of the data series $y(t), t = 1, \ldots, n$, then the associated discretised time series, z, is defined as

$$z(t) = \begin{cases} 1 \text{ if} & y(t) > \phi_y \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

A data set is parameterised as follows: there are k models of the form given in Equation 5, each of which generates l time series; each of the $l \cdot k$ time series is of length n and is sampled at the same points $t = 1, 2, \ldots, n$; σ defines the variability of static noise, s the level of random shocks and p the probability of shocks. From a data mining perspective, the clustering problem we are attempting to solve has the following properties:

- learning is unsupervised since cluster membership is not known *a priori*;
- cluster sizes are equal (l is the same for all clusters k);
- there is no missing data (each series sampled at the same points);
- the number of clusters, k, is known *a priori*; and
- the distribution of ϵ is constant for all observations and all series.

3.2 Experimental Procedure

We use the k-means algorithm with the correlation based distance metric given in Equation 2 for experimentation. We choose k-means as it is one of the most popular and simple clustering algorithms. Further experimentation will involve assessment of the clustering using alterative algorithms and distance metrics.

We initialise the centroids for k-means to a random data series. It is well known that k-means is sensitive to initial conditions [11], hence we repeat the classification algorithm with random initial conditions and then average over the runs. For any data set D of $l \cdot k$ time series derived from a particular set of k models, the clustering algorithm is run u times. For any particular parameter values, v different sets of k models are generated.

Clustering performance is measured by the classification accuracy, i.e. the ratio of the percentage of the data in the final clustering that is in the correct cluster. Note we are measuring accuracy on the training data rather than applying the data to a separate testing data set. We do this because we wish to measure the effects of outliers in the training data rather than assess the algorithm's ability to solve the clustering problem. We use this measure rather than some of the alternatives (see [24]) since we know the correct clustering.

For a given clustering we measure the accuracy by forming a $k \times k$ contingency matrix. Since the clustering label may not coincide with the actual labelling (e.g. all those series in cluster 1 may be labelled cluster 2 by the clustering algorithm) we evaluate the accuracy (number correctly classified divided by the total number of series) for all possible $k!$ permutations of the columns of the contingency table. The achieved accuracy is the maximum accuracy over all permutations.

We average the accuracy over the u repetitions to find the average accuracy for a set of particular models, and average this data over the v different model sets to find the average performance for a particular set of parameter values. This average of averages we term *the average correct classification.*

All the parameters are given in Figure 1. Unless otherwise stated, the parameter values used in all experimentation is given in brackets.

Parameters	Meaning	Default value
experiment parameters		
k	Number of clusters	$k = 2$
n	Time series length	$n = 100$
l	Series per cluster	$l = 10$
u	Clusterings per model	$u = 20$
v	Number of models	$v = 20$
D	Data set consisting of $l \cdot k$ series	
m_i	A generating model	
model parameters		
$a_i, b_i, \ \ i = 1 \ldots k$	Linear parameters	
σ	Model noise	$\sigma = 10$
p	Outlier probability	
s	Random shock value	$s = 100$

Fig. 1. List of experimental parameters

3.3 Experimental Sequence

We demonstrate that the discretised data results in significantly better clusters when there are outliers in the data by conducting two experiments.

- Experiment 1 shows that discretising the data does not significantly reduce the clustering accuracy when there are no outliers or outliers are very unlikely (Section 3.4).
- Experiment 2 shows that discretising the data does significantly increase the clustering accuracy when outliers are more likely (Section 3.5).

3.4 Experiment 1: Showing that using z does not significantly decrease accuracy

The objective of this experiment is to determine whether discretising the data significantly reduces the accuracy of the classification of the k-means algorithm using a correlation based distance metric. We perform this experiment with a sample from a wider class of models than used in Experiments 1 and 2. The format of the models is as given in Equations 5 and 6 and the default parameters are used (Figure 1). For each cluster, b_i is selected randomly on the interval $[-0.5, 0.5]$ and a_1 and a_2 are uniformly sampled in the range $[100, 200]$ for each time series. The justification for these parameters is given in [3].

Let M be the set of all models considered in the experiment, with an instance denoted m_i. M contains all linear models with constants in the range $[100, 200]$ and gradient in the range $[-0.5, 0.5]$. Let C be the set $M \times M$ of generators of the two cluster model. ϕ_y is the population median of the average classification accuracy of the k-means algorithm (k known, random initial centroids) over the space of underlying models C and ϕ_z denotes the population median when using the discretised data. μ_y and μ_z are the associated population means. Given a random sample of size v from C we wish to test $H_0 : \phi_z = \phi_y$ against the alternative $H_1 : \phi_z < \phi_y$.

To test this hypothesis we conduct both paired and unpaired tests. With the paired tests, the k-means algorithm is run ($u = 20$ times) on continuous series and discrete series derived from the same continuous data. Table 1 shows that there is little difference in the accuracy of the resulting clusterings.

Table 1. Clustering accuracy summary for paired samples. The difference series is the continuous data minus the discrete data

	Mean	Median	Min	Max	StDev
Continuous	80.13%	91.15%	56.60%	100%	17.80
Discrete	79.84%	86.30%	57.80%	100%	17.64
Difference	2.67%	0%	-5%	8.7%	2.67

Of the 50 trials, there were 24 trials with a positive difference (i.e. continuous data resulted in a higher accuracy than the discrete data), 4 had no difference and 21 had negative difference. There is a positive mean difference but the median difference is zero and we cannot reject the null hypothesis that $H_0 : \phi_z = \phi_y$ for the alternative $H_1 : \phi_z < \phi_y$ using the Wilcoxon's test for matched pairs. It is worth noting that we cannot rejected the hyptothesis $H_0 : \mu_z = \mu_y$ in favour of the alternative $H_1 : \mu_z < \mu_y$ using a t-test. Despite this result, we use non-parametric tests due to the decidely non normal nature of the data.

To verify there is in fact no significant difference in the median clustering accuracy for the models considered, we ran the experiment with unmatched pairs and 100 models in each sample ($v = 100$).

Table 2. Accuracy summary for unmatched models

	Mean	Median	Min	Max	StDev
Continuous	76.801%	73.00%	56.20%	100%	16.96
Discrete	76.798%	72.85%	55.50%	100%	15.83

Table 2 summarises the results. The difference in the mean is neglible, and using the Mann-Whittley test we cannot reject the null hypothesis $H_0 : \phi_z = \phi_y$ in favour of the alternative $H_1 : \phi_z \neq \phi_y$.

These results clearly demonstrate that discretising the data does not decrease the accuracy of the k-means clustering algorithm used to cluster data derived from two models of the form given in Section 3.1 when there are no outliers in the data. The next experiment shows that the accuracy of the clustering significantly improves even when the probability of an outlier is very small.

3.5 Experiment 2: Using a discretised series increases accuracy

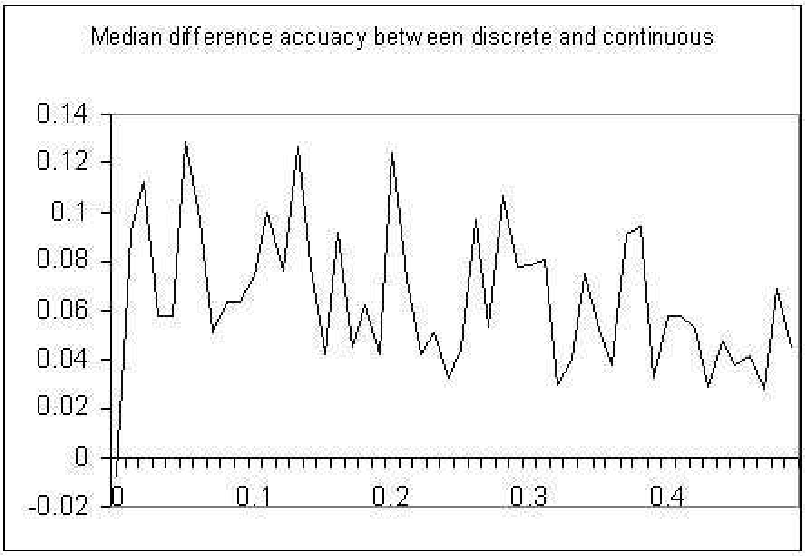


Fig. 2. Difference in clustering accuracy for probability of an outlier between 0 and 0.5 with paired samples

To demonstrate the desirability of discretising, we repeat experiment 3 for various values of p with both paired and unpaired samples. All other parameters are identical to those used in results presented in Section 3.4 ($v = 50, x = 0.5$). Figure 3.5 shows how the accuracy difference changes as the probability of an outlier increases. Each data represents the median of 50 evaluations, where each evaluation consists of 20 runs of the k-means algorithm. There is an initial dramatic decrease in accuracy of clustering using the continuous data. As the probability of an outlier increases the accuracy difference between using discretised and continuous data decreases. This is because the noise eventually overwhelms the algorithms ability to cluster correctly. To illustrate the effect of outliers more clearly, Figure 3 shows the results for the same experiment using a smaller range of p. Clearly the clustering algorithm is performing much better with the discretised data even when the probability of outlier is very low.

Figure 4 shows a repeat of the experiment described by Figure 3 with unpaired samples. For contrast with Figure 3, the mean values rather than the

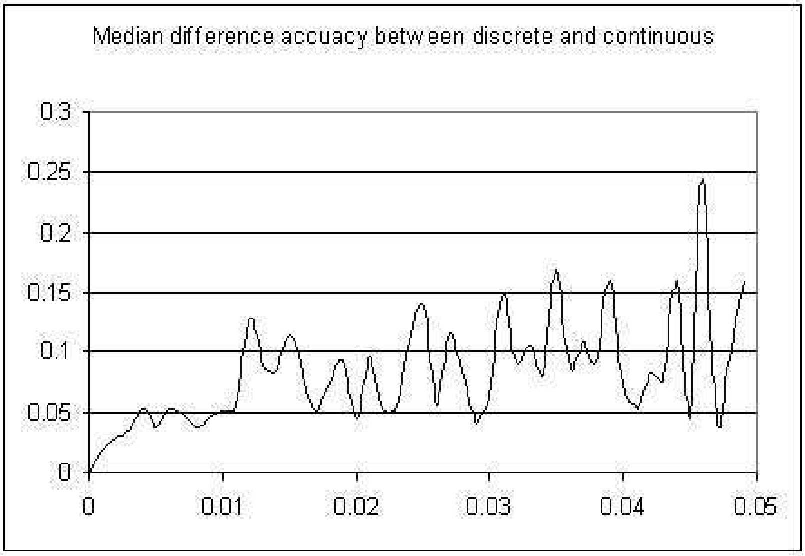


Fig. 3. Difference in clustering accuracy for probability of an outlier between 0 and 0.05 with paired data

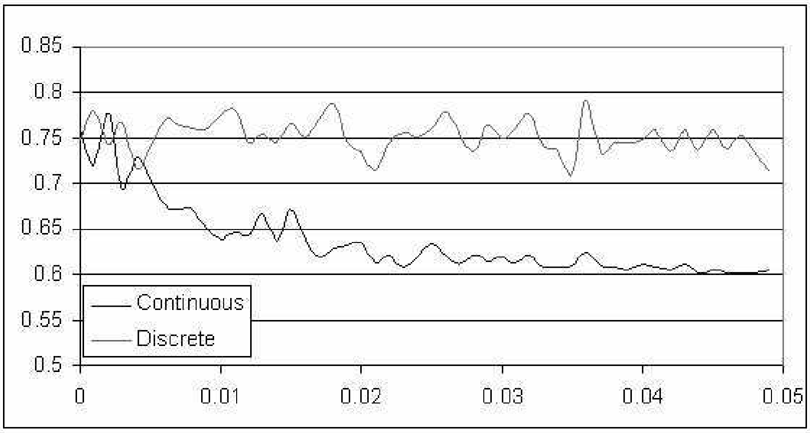


Fig. 4. Difference in clustering accuracy for probability of an outlier between 0 and 0.05 with unpaired data

medians are shown, but the pattern in both averages is the same. A very small probability of outliers results in a much improved performance when the discretised data is used. Finally, to emphasise the point further, we fixed the number of outliers so that each series of 100 data had exactly one outlier and repeated the paired experiment. Of the 50 trials, 9 resulted in the continuous data having higher accuracy, in 1 trial the accuracy was the same and in the remaining 40 the accuracy was greater when the discrete data was used.

Using Wilcoxon's signed-rank test for matched pairs, the null hypothesis $H_0 : \phi_d = 0$ can be rejected in favour of the alternative $H_1 : \phi_d < 0$ at the 1% level

4 Conclusions and Future Direction

The clustering of time series is a field that has attracted the interest of researchers from a wide range of disciplines. This report has provided a brief review of the techniques used, including a description of the types of models assumed, the distance metrics employed and the clustering techniques used. Many real world time series have the unfortunate property that they contain outliers, and the aim of this research is to demonstrate that if discretised series are used instead of the continuous data then the effect of outliers can be lessened significantly.

We have demonstrated how, for a certain class of model, distance metric and clustering algorithm, discretising the series into binary series of above and below the median can improve the clustering accuracy when there are outliers in the data, even when the probability of an outlier is very small.

Although there are benefits from using the binary series of above and below the median when there are outliers, it obviously means some of the information in the original data is discarded. It is worthwhile discovering how much this effects the quality of clusters formed.

The obvious way of extending this work would be to assess the effect of discretisation when the data arises from other models and when alternative distance metrics and/or clustering algorithms are employed. It is also a logical extension to apply the technique to real world data.

Working with binary series can often allow for significant speed improvements with model fitting and clustering techniques, and this could be another benefit of discretisation.

References

1. Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient Similarity Search In Sequence Databases. In D. Lomet, editor, *Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)*, pages 69–84, Chicago, Illinois, 1993. Springer Verlag.
2. Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. Discovering clusters in motion time-series data. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 2003.
3. A. J. Bagnall, G. Janakec, and M. Zhang. Clustering time series from mixture polynomial models with discretised data. Technical Report CMP-C03-17, School of Computing Sciences, University of East Anglia, 2003.

4. A. J. Bagnall and I. Toft. An agent model for first price and second price private value auctions. In *Proceedings of the 6th International Conference on Artificial Evolution*, 2003.
5. A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago, April 2001.*, 2001.
6. Z. Bar-Joseph, G. Gerber, D. Gifford, T. Jaakkola, and I. Simon. A new approach to analyzing gene expression time series data. In *Proceedings of The Sixth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 39–48, 2002.
7. Roberto Baragona. A simulation study on clustering time series with metaheuristic methods. *Quaderni di Statistica*, 3, 2001.
8. R. Blender, K. Fraedrich, and F. Lunkeit. Identification of cyclone-track regimes in the north atlantic. *Quart J. Royal Meteor. Soc.*, (123):727–741, 1997.
9. Z. Bohte, D. Cepar, and K. Kosmelj. Clustering of time series. In *Proceedings in Computational Statistics*. Physica-Verlag, 1980.
10. G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control, 3rd Edition.* Prentice Hall, 1994.
11. Paul S. Bradley and Usama M. Fayyad. Refining initial points for K-Means clustering. In *Proc. 15th International Conf. on Machine Learning*, pages 91–99. Morgan Kaufmann, San Francisco, CA, 1998.
12. Igor V. Cadez, Scott Gaffney, and Padhraic Smyth. A general probabilistic framework for clustering individuals and objects. In *Knowledge Discovery and Data Mining*, pages 140–149, 2000.
13. Igor V. Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *Knowledge Discovery and Data Mining*, pages 280–284, 2000.
14. Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.
15. Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *Knowledge Discovery and Data Mining*, pages 16–22, 1998.
16. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data. *J. R. Stat. Soc. B*, 39:1–38, 1972.
17. Evangelos Dermatas and George Kokkinakis. Algorithm for clustering continuous density HMM by recognition error. *IEEE Tr. On Speech and Audio Processing*, 4(3):231–234, 1996.
18. Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.
19. Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings 1994 ACM SIGMOD Conference, Mineapolis, MN*, pages 419–429, 1994.
20. Sergio M. Focardi. Clustering economic and financial time series: exploring the existence of stable correlation conditions. Technical Report 2001-04, The Intertek Group, 2001.
21. Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. Technical Report 99-15, Department of Information and Computer Science, University of California, 1999.

22. Scott Gaffney and Padhraic Smyth. Curve clustering with random effects regression mixtures. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.
23. Amir B. Geva and Dan H. Kerem. *Fuzzy and Neuro-Fuzzy Systems in Medicine*, chapter 3. Brain state identification and forecasting of acute pathology using unsupervised fuzzy clustering of EEG temporal patterns. CRC Press, 1998.
24. Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.
25. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, New Jersey, 1988.
26. E. J. Keogh and M. J. Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. In T. Terano, H. Liu, and A. Chen, editors, *Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference, PAKDD 2000*, volume 1805, pages 122–133, Kyoto, Japan, 2000. Springer.
27. Eamonn Keogh, Jessica Lin, and Wagner Truppel. Clustering of streaming time series is meaningless. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003.
28. Filipp Korkmazskiy, Biing-Hwang Juang, and Frank Soong. Generalized mixture of HMMs for continuous speech recognition. In *Proceedings IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 2, pages 1443–1446, 1997.
29. K. Kosmelj and V. Batagelj. Cross-sectional approach for clustering time varying data. *Journal of Classification*, 7:99–109, 1990.
30. Cen Li and Gautam Biswas. Clustering sequence data using hidden markov model representation. In *SPIE'99 Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology*, pages 14–21, 1999.
31. Cin Li. *A Bayesian approach to temporal data clustering using the hidden Markov model methodology.* PhD thesis, Vanderbilt University, Nashville, 2000.
32. Cin Li and Gautam Biswas. Profiling of dynamic system behaviors using hidden markov model representation. In *Proceedings of the ICSC'99 Advances in Intelligent Data Analysis(AIDA'99)*, 1999.
33. Cin Li and Gautam Biswas. Temporal pattern generation using hidden markov model based unsupervised classification. In D. Hand, K. Kok, , and M. Berthold, editors, *Advances in Intelligent Data Analysis, Lecture Notes in Computer Science vol. 1642.* Springer, 1999.
34. Cin Li and Gautam Biswas. Bayesian clustering for temporal data using hidden markov model representation. In *proceedings of the Seventeenth International Conference on Machine Learning*, pages 543–550, 2000.
35. J. MacQueen. Some methods for classification and analysis of multivariate observations. In Lucien M. Le Cam and Jerzy Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I,Statistics.* University of California Press, 1967.
36. Elizabeth Ann Maharaj. A significance test for classifying arma models. *Journal of Statistical Computation and Simulation*, 54:305–331, 1996.
37. Elizabeth Ann Maharaj. Clusters of time series. *Journal of Classification*, 17:297–314, 2000.
38. Eamonn Keogh Michail Vlachos, Jessica Lin and Dimitrios Gunopulos. A wavelet-based anytime algorithm for k-means clustering of time series, 2003.

39. Tim Oates. Identifying distinctive subsequences in multivariate time series by clustering. In S. Chaudhuri and D. Madigan, editors, *Fifth International Conference on Knowledge Discovery and Data Mining*, pages 322–326, San Diego, CA, USA, 1999. ACM Press.
40. Tim Oates, Laura Firoiu, and Paul Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21, 1999.
41. Tim Oates, Laura Firoiu, and Paul R. Cohen. Using dynamic time warping to bootstrap HMM-based clustering of time series. *Lecture Notes in Computer Science*, 1828:35–52, 2001.
42. Tim Oates, Matthew D. Schmill, and Paul R. Cohen. Identifying qualitatively different outcomes of actions: Gaining autonomy through learning. In Carles Sierra, Maria Gini, and Jeffrey S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 110–111, Barcelona, Catalonia, Spain, 2000. ACM Press.
43. Paul Ormerod and Craig Mounfield. Localised structures in the temporal evolution of asset prices. In *New Approaches to Financial Economics*. Santa Fe Conference, 2000.
44. Domenico Piccolo. A distance measure for classifying ARIMA models. *Journal of Time Series Analysis*, 11(2):153–164, 1990.
45. Marco Ramoni, Paola Sebastiani, and Paul Cohen. Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121, 2002.
46. Greg Ridgeway. Finite discrete Markov processes. Technical Report MSR-TR-97-24, Microsoft Research, 1997.
47. Padhraic Smyth. Clustering sequences with hidden markov models. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 648. The MIT Press, 1997.
48. P. Tong and H. Dabas. Cluster of time series models: An example. *Journal of Applied Statistics*, 17:187–198, 1990.
49. Shi Zhong. *Probabilistic model-based clustering of complex data*. PhD thesis, University of Texas at Austin, 2002.
50. Shi Zhong and Joydeep Ghosh. HMMs and coupled HMMs for multi-channel EEG classification. In *Proc. IEEE Int. Joint Conf. on Neural Networks*, 2002.
51. Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering. In *Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE)*, 2002.
52. Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering and its application to clustering time sequences. Technical report, Department of Electrical and Computer Engineering, University of Texas, 2002.
53. Shi Zhong and Joydeep Ghosh. Scalable, balanced model-based clustering. In *Proceedings of SIAM Int. Conf. on Data Mining*, 2003.

Relative Temporal Association Rule Mining

Edi Winarko and John F. Roddick

School of Informatics and Engineering
Flinders University,
PO Box 2100, Adelaide,
South Australia 5001,
Email: {edi.winarko, john.roddick}@infoeng.flinders.edu.au

Abstract. A temporal association rule is one that has a specified temporal constraint on its validity, for example, it may hold only during a specific time interval. A *relative* temporal rule is one that is further qualified according to a temporal relationship with other events or intervals as opposed to an absolute point in time. Due to their potential application, the problem of finding temporal association rules has recently become an important research topic and is receiving great interest from researchers. Several models of temporal association rules and algorithms for discovering such rules have been proposed, although the discovery of relative temporal rules has received less attention. In this paper, we survey the work to date in temporal association rule mining and present a taxonomy of research to date. We also present a new mechanism for discovering relative temporal association rules.

Keywords: Temporal Data Mining, Relative Temporal Association Rules.

1 Introduction

The problem of finding association rules was originally proposed in 1993 (Agrawal, Imielinski & Swami 1993) and has been investigated widely for a number of years. In general, however, this work does not consider any temporal aspect which might be inherent in the data and thus important information may remain undiscovered. While association rule mining can only ever reliably report on useful knowledge in the dataset under examination, and therefore for the time period for which the dataset corresponds, there is often an implicit assumption that the discovered rules are valid universally in time. Moreover, the dataset itself may contain useful time-dependant observations and thus temporal correlations, trends and changes may be contained in the data.

As a result, the information that can be conveyed by non-temporal rules can often be less valuable as their applicability is generalised – there is no way of knowing when the rules are strongest, or the manner in which they occur in time. As stated by Chen and Petrounias (2000), the association rule - *customers who buy bread and butter also buy milk during the summer* - may be more useful than the non-temporal equivalent - *customers who buy bread and butter also buy milk*.

Because of this richer rule semantics, the problem of finding temporal association rules has recently become an important research topic and is receiving a great deal of research interest. A number of temporal association rule models and algorithms to discover such rules have been proposed (Ale & Rossi 2000, Chen & Petrounias 2000, Lee, Lin & Chen 2001, Li, Ning, Wang & Jajodia 2001, Ozden, Ramaswamy & Silberschatz 1998, Ramaswamy, Mahajan & Silberschatz 1998, Rainsford & Roddick 1999, Wang, Yang & Muntz 1999, Wang, Yang & Muntz 2001, Zimbrão, de Souza, de Almeida & da Silva 2002) and a survey of temporal data mining methods and paradigms in general is given in (Roddick & Spiliopoulou 2002).

The paper is structured in two major parts. The first part presents a survey of work in temporal association rule mining and discusses a taxonomy for this research. The second part presents a new method for discovering relative temporal association rules including a discussion of the problems of applying the *Apriori* principle to temporally augmented itemsets.

2 Current Research in Temporal Association Rule Mining

Ozden et al. (1998) discuss the problem of mining cyclic association rules, i.e., the association rules that occur periodically over time. Two algorithms and optimization techniques to discover cyclic association rules are proposed, although, this work is limited because it cannot describe real-life concepts such as *the first business day of every month* in which the distance between two consecutive such business days are not always the same. Ramasmamy et al. (1998) extend this work by considering the discovery of association rules that hold during the time intervals described by a user-defined calendar algebra expression. The calendar algebra adopted is considered more powerful in defining temporal patterns, but to give such expressions the users need to know what temporal patterns they are interested in.

To provide more flexibility to the user Li et al. (2001) propose a temporal association rule model that uses calendar schemas as a framework for temporal patterns, instead of using user-defined calendar algebra expressions. Their work considers all possible temporal patterns in the calendar schema as opposed to simply complying with user-supplied temporal expressions.

Ale and Rossi (2000) studied the discovery of association rules that hold for transactions during the lifetime (lifespan) of items involved in the rules. The lifetime is intrinsic to the data, so that the users are not required to define it. The lifetime of items is a period between the first and the last time the items appears in transactions in the dataset. Similar to this work is the work proposed by Lee et al. (2001) who studied the problem of mining general temporal association rules in publication database. A publication database refers to a set of transactions where each transaction contains an individual exhibition period. Zimbrão et al. (2002) extend this work by proposing a new approach to discover calendar-based association rules with item's lifespan restriction.

The discovery of the longest interval and the longest periodicity of association rules was presented in (Chen & Petrounias 2000) and in (Rainsford & Roddick 1999), it was proposed to add temporal features to association rules by associating a conjunction of binary temporal predicates that specify the relationships between the timestamps of transactions. Visualisation techniques for viewing these temporal predicate association rules were provided in (Rainsford & Roddick 2000).

All the above work assumed that each transaction is associated with a temporal attribute that records the time for which the attributes of the transaction are valid in the modeled domain. They also assumed that all non-temporal attributes have binary (boolean) domain. Unlike these work, (Wang et al. 1999, Wang et al. 2001) has proposed another model by introducing the mining of temporal association rules for evolving numerical attributes.

Despite this work, to the best of our knowledge, there has not been specific survey available on the topic of temporal association rule discovery although (Antunes & Oliveira 2001) provide a survey on the most significant techniques to deal with temporal sequences while (Roddick & Spiliopoulou 2002) provide a broad survey of temporal data mining research. In order to fill this gap, in this paper we first survey the research area.

2.1 Taxonomy of Temporal Association Rules

A number of temporal association models has been proposed, each with different reasons and goals. Each model has its own characteristics, differences and similarities. In this section, we classify the models by looking at them from four different aspects: the domain in which the models applied, measures of interestingness used, temporal feature associated with the rules, and algorithms used to discover the rules.

Domain of Attributes It can be seen in Table 1 that most of temporal association rule models discussed here has assumed that all non-temporal attributes have binary (boolean) domain. It means that the generated temporal association rules only deal with the association between the presence or absence of items or attributes. The transaction dataset of these models is normally defined as follows:
Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. D is a dataset of transactions, where each transaction $\mathbf{s}$ consists of a set of items such that $\mathbf{s} \subseteq I$. Associated with each transaction $\mathbf{s}$ is a timestamp t_s which represents valid time of transaction $\mathbf{s}$.

Wang et al. (1999, 2001) studied the mining of temporal association rules over evolving numerical attributes, instead of binary attributes. This work is different from the model of quantitative association rules introduced in (Srikant & Agrawal 1996). In this model, the dataset consists of a set of objects, each of which has a unique ID and a set of time varying numerical attributes.

Measures In association rules mining, two measures of rule interestingness that are commonly used are *support* and *confidence*. Both reflect the usefulness and certainty of discovered rules (Han & Kamber 2001). However, for some of temporal association rule models, these two measures are considered insufficient and an additional measure is considered necessary in their proposed model, for example temporal support (Ale & Rossi 2000), frequency (Chen & Petrounias 2000), temporal confidence (Rainsford & Roddick 1999), and strength and density (Wang et al. 1999, Wang et al. 2001), as shown in Table 1.

Ale and Rossi (2000) introduce the notion of temporal support to filter the items with high support but short life. The combination of support and temporal support is used to determine if an itemset is large. Ie., an itemset is large if it satisfies the user-specified minimum support and minimum temporal support.

Chen and Petrounias (2000) propose using frequency to measure the proportion of the intervals, during which the rules satisfy minimum support and minimum confidence, to the intervals in $\phi(TF)$. $\phi(TF)$ represents a set of time intervals, i.e., $\phi(TF) = \{P_1, P_2, \ldots, P_n\}$, where P_i is a time interval. The association rules should satisfy the minimum frequency in addition to minimum support and minimum confidence.

In (Rainsford & Roddick 1999), temporal confidence is used to determine how strong the temporal relationship between temporal items in the rule. Since the temporal relationship between items is represented as a binary predicate, each predicate in the rule must satisfy the minimum temporal confidence.

Wang et al. (2001) propose two new measures, strength and density, to qualify the validity of rules on evolving numerical attributes. The support, strength, and density of the rules indicates the frequency of occurrences, concentration of population, and the degree of non-independence represented by the rules, respectively.

Temporal Association Rule Types According to (Chen & Petrounias 2000), a temporal association rule can be represented as a pair $< AR, TF >$, where AR is an association rule and TF is temporal feature belongs to AR. Depending on the interpretation of the temporal feature TF, a temporal association rules $< AR, TF >$ can be classified as:

- a *universal* association rule if $\phi(TF) = \{T\}$, where T represents the time domain;
- an *interval* association rule if $\phi(TF) = \{itvl\}$, where $itvl \subset T$ is a specific time interval;
- a *periodic* association rule if $\phi(TF) = \{p_1, p_2, \ldots, p_n\}$, where $p_i \subset T$ is a periodic interval in cycles;
- a *calendric* association rule if $\phi(TF) = \{cal_1, cal_2, \ldots, cal_m\}$, where $cal_j \subset T$ is a calendric interval in a specific calendar.

This classification method is used to classify the temporal association rule models presented in this paper. However, since certain models cannot be included in these classes we have to create two new classes – binary predicate and numerical attribute evolution. The universal association rule class is excluded from our classification because it represents a class of classical association rules (non-temporal association rules). Table 1 shows five classes in our classification, namely interval, cyclic, calendric, binary predicate, and numerical attribute evolution. Using this classification, a model can be classified into more than one class, depending on the temporal feature associated with association rules. Each temporal association model with its temporal feature is presented in Table 2.

Author	Measures			Type of Association Rule					Attribute Domain
	Supp.	Conf.	Extra Measure	Interval	Cyclic	Calendric	Binary Pred.	Attr. Evol.	
Ale & Rossi (200)	✓	✓	Temporal support	✓					Binary
Ozden et al. (1998)	✓	✓			✓				Binary
Chen & Petrounias (2000)	✓	✓	Frequency	✓	✓				Binary
Ramaswamy et al. (1998)	✓	✓				✓			Binary
Li et al. (2001)	✓	✓				✓			Binary
Rainsford & Roddick (1999)	✓	✓	Temporal confidence				✓		Binary
Lee et al. (2001)	✓	✓		✓					Binary
Zimbrao et al. (2002)	✓	✓		✓		✓			Binary
Wang et al. (1999,2001)	✓		Strength & Density					✓	Numeric

Table 1. Temporal association rule classification

Author	Algorithms		Temporal AR Models
	Apriori-based	Other	
Ale & Rossi (2000)	✓		$(X \Rightarrow Y, [t_1, t_2])$ $[t_1, t_2]$ is a lifespan of $X \cup Y$
Ozden et al. (1998)	✓		$(X \Rightarrow Y, c = (l, o))$ c is a cycle with length l and offset o
Chen & Petrounias (2000)	✓		$(X \Rightarrow Y, TF)$ $\phi(TF) = \{P_1, P_2, \ldots, P_n\}$
Ramaswamy et al. (1998)	✓		$(X \Rightarrow Y, C)$ C is a calendar expression
Li et al. (2001)	✓		$(X \Rightarrow Y, e)$ e is a calendar pattern
Rainsford & Roddick (1999)	✓		$X \Rightarrow Y \wedge P_1 \wedge P_2 \ldots P_n$ P_i is a binary temporal predicate
Lee et al. (2001)	✓	PPM	$(X \Rightarrow Y, (t, n))$ (t, n) is the max. common exhibition period
Zimbrao et al. (2002)	✓		$(X \Rightarrow Y, e, [t_1, t_2])$ e is a calendar pattern $[t_1, t_2]$ is a lifespan of $X \cup Y$
Wang et al. (1999,2001)	✓	TAR	$X \Longleftrightarrow Y$ X and Y are conjunctions of att. evolution

Table 2. Temporal association rule models and algorithms

From the four models in the interval association rule class, each model provides different meaning to the time interval. In (Ale & Rossi 2000, Zimbrão et al. 2002), a time interval represents the lifespan of items in the rules, in (Lee et al. 2001), it represents the maximum common exhibition period of items in the rules, while in (Chen & Petrounias 2000), it represents any specific time interval.

We classify the works of Ramaswamy et al. (1998) and Li et al. (2001) into the calendric association class. (Ramaswamy et al. 1998) introduced the notion of calendar algebra to describe phenomena of interest in association rules. It is based on the work reported in (Allen 1983, Leban, McDonald & Forster 1986) and the implementation reported in (Chandra, Segev & Stonebraker 1994). The calendar algebra expression defines a set of time intervals that the algorithm considers in discovering the association rules. The association rules are called calendric if they have the minimum support and confidence in every time unit contained in the calendar. The algorithms proposed to discover such rules are similar to the ones discussed in (Ozden et al. 1998), but they were modified to deal with calendars, instead of cycles. Li et al. (2001) use calendar schemas instead of calendar algebra expressions.

Algorithms The algorithms so far proposed to discover temporal association rules can be divided into two categories, i.e., *Apriori*-based and non *Apriori*-based algorithms. Table 2 shows that all models discussed here can be categorised as using the *Apriori*-based algorithms. Two models also proposed new algorithms which are not based on *Apriori* (Lee et al. 2001, Wang et al. 1999, Wang et al. 2001). In (Lee et al. 2001) the algorithm Progressive-Partition-Miner (PPM) is proposed to discover general temporal association rules in a publication database, while Wang et al. (1999,2001) discuss the TAR algorithm to discover the temporal association rules on evolving numerical attributes.

In the following sections, we discuss in more detail the *Apriori*-based algorithms to discover interval association rules, cyclic association rules, calendric association rules, and temporal predicate association rules. In addition, we also discuss the optimized algorithms to discover cyclic and calendric association rules.

2.2 Interval Association Rules

The interval association rule model discovers association rules that hold during the lifetime of items involved in the rules (Ale & Rossi 2000). The model is motivated by the observation that it is possible to have association rules with a high confidence but with little support. During the mining process such rules may not be discovered as their support is less than the minimum support. as the denominator in the support calculation is the total number of transactions in dataset. If the denominator is limited to the total number of transactions belonging only to the lifetime of items, these rules could be discovered. Therefore, in this model the search for large itemsets is limited to the lifetime of the itemset's members and each generated rule has an associated time frame, corresponding to the lifetime of the items participating in the rule.

We will use the following transaction dataset, found in (Ale & Rossi 2000), to describe how the model works.

Example 3.1: Let $\mathbf{I} = \{A, B, C, D, E, F, G, H, I\}$ and D contain six transactions:

$s_1 = \{A, C, F, H, I\}$, t: 1
$s_2 = \{A, B, C, G\}$, t: 2
$s_3 = \{B, C, D, G, I\}$, t: 3
$s_4 = \{A, C, I\}$, t: 4
$s_5 = \{C, D, E, H, I\}$, t: 5
$s_6 = \{A, D, F, G\}$, t: 6

The lifespan of an item A is represented by a closed interval $[t_i, t_j]$, where $t_i <_T t_j$, and is denoted by l_A. If D is the transaction dataset, then D_{l_A} is the subset of D whose timestamps $t_i \in l_A$, and $|D_{l_A}|$ is the number of transactions whose timestamps $t_i \in l_A$. As an example, from the transaction dataset above, the lifespan of A, B and C is l_A = [1, 6], l_B = [2, 3] and l_C = [1, 5], respectively. The lifespan of an itemset X is calculated as an intersection of the items' lifespan in the itemset. If $X = \{I_1, I_2, \ldots, I_n\}$ then its lifespan is $l_X = l_{I_1} \cap l_{I_2} \cap \ldots \cap l_{I_n}$. In the case $X = \{A, C\}$ then its lifespan is $l_X = l_A \cap l_C = [1, 6] \cap [1, 5] = [1, 5]$.

The calculation of support of an itemset X is modified by taking into consideration the itemset's lifespan. The denominator is not the number of transactions of the entire dataset $|D|$, but $|D_{l_X}|$, that is the number of transactions whose timestamp $t_i \in l_X$. The support of X in D over its lifespan l_X is denoted as $sup(X, l_X, D)$. Thus, the support of $X = \{A\}$ and $Y = \{H\}$ is $sup(X, l_X, D) = 4/6 = 0.67$ and $sup(Y, l_Y, D) = 2/5 = 0.40$, respectively.

It is possible for the itemset to have high support but short life. As an example, an item E has support of 100% but its lifespan is short, i.e., $|l_E| = 1$. The temporal support can be used to filter such items. Therefore, to determine if an itemset X is large or not the combination of support and temporal support is used. Given the minimum support $\sigma \in [0, 1]$ and the minimum temporal support τ, an itemset X is large in its lifespan l_X if $sup(X, l_X, D) \geq \sigma$ and $|l_X| \geq \tau$.

This model proposed the concept of item's obsolescence. It is used to filter out the items or itemsets that are considered obsolete. An item whose lifespan is [t_i, t_j] is obsolete at a specified time instant t_o if $t_2 < t_o$. It is not necessary to check for obsolete k-itemsets, for $k > 1$, because a k-itemset is obsolete if it contains an obsolete item.

The confidence of a rule $X \Rightarrow Y$ in $[t_1, t_2]$, where $[t_1, t_2]$ is a time frame corresponding to the lifespan of $X \cup Y$, is denoted by $conf(X \Rightarrow Y, l_{X \cup Y}, D)$ and defined as

$$conf(X \Rightarrow Y, l_{X \cup Y}, D) = sup(X \cup Y, l_{X \cup Y}, D)/sup(X, l_{X \cup Y}, D)$$

Let T = $\{\ldots, t_o, t_1, t_2, \ldots\}$ be a set of time instants, countably infinite, over which a linier order $<_T$ is defined, where $t_1 <_T t_2$ means that t_1 occurs before

t_2. Given the transaction dataset D (as defined in section 2.1), the minimum support σ, the minimum temporal support τ, and the minimum confidence γ, the interval association rule $X \Rightarrow Y[t_1, t_2]$ holds in D if $sup(X \cup Y, l_{X \cup Y}, D) \geq \sigma$, $|l_{X \cup Y}| \geq \tau$ and $conf(X \Rightarrow Y, l_{X \cup Y}, D) \geq \gamma$, where $l_{X \cup Y} = [t_1, t_2]$.

The paper asserts that any existing algorithm for association rule discovery, for example (Agrawal & Srikant 1994, Brin, Motwani, Ullman & Tsur 1997, Park, Chen & Yu 1995), can be modified to discover these interval association rules. Algorithm 2.1 is a modified version of the *Apriori* algorithm (Agrawal & Srikant 1994) to find such rules. The algorithm consists of two phases. First, the generation of every itemset X such that X is large in its lifespan l_X, and second, finding the rules from every large itemset X. The first phase consists of several passes. In the first pass, to obtain the large 1-itemsets L_1, the algorithm not only counts the item occurrences and records its lifespan but also counts the number of transactions in this lifespan so that the support of the item in its lifespan can be calculated. In subsequent passes, for each pass $k > 1$, the candidate itemsets C_k are generated from the large itemset L_{k-1}, using *Apriori*'s method for candidate generation. The lifespan of a k-itemset is determined as follows: if the k-itemset U is obtained by joining $(k-1)$-itemsets V and W, the lifespan of U is the intersection of the lifespan of V and W.

The second phase can be done by using *Apriori*'s method of rules generation. For every large itemset Z it is required to find the rules $X \Rightarrow (Z - X)[t_1, t_2]$ such that $sup(Z, l_Z, D)/sup(X, l_Z, D) \geq \gamma$, for each $X \subset Z$. In computing the rule confidence, the value of $sup(X, l_Z, D)$ is estimated by using the value of $sup(X, l_X, D)$. The reason for doing this is to avoid recalculating the support for $2^k - 2$ itemsets X in l_Z.

Algorithm 2.1 Find association rules in items' lifespan

```
// Phase 1: Find all large itemsets $X$ in $l_X$
$L_1$ = {large 1-itemsets}
for ($k = 2; L_{k-1} \neq \emptyset; k++$) do
    $C_k$ = AprioriGen($L_{k-1}$)
    for all transaction $s \in D$ do
        Count the support of all candidates in $C_k$
    end for
    $L_k = \{X \in C_k | sup(X, l_X, D) \geq \sigma$ and $|l_X| \geq \tau\}$
end for
// Phase 2: Generate the rules
for all large itemset $Z \in L_k$, $k \geq 2$ do
    genrules($Z$)
end for
```

2.3 Cyclic Association Rules

The concept of cyclic (periodic) association rules and the mining tasks for discovering such rules was described in (Ozden et al. 1998). An association rule is called *cyclic* if the rule has the minimum confidence and support at regular time intervals. Such a rule is not required to hold for the entire transactional dataset, but rather only for transactional data in a particular periodic time interval.

In order to discover cyclic association rules, the model assumes that the unit of time (hour, day, week, month, etc.) is given by the user. The i^{th} time unit is denoted by t_i, $i \geq 0$, and corresponds to the time interval $[i.t, (i+1).t]$. Given a transaction dataset D, a set of transactions executed in t_i is denoted by $D[i]$.

A cycle c is a tuple (l, o) consisting of length l (multiples of the time unit) and an offset o (the first time unit in which the cycle occurs), $0 \leq o \leq l$. A time unit t_i is part of a cycle $c = (l, o)$ if $o = i \bmod l$ holds. For example, if the unit of time is an hour, every fourth hour starting from the 3^{rd} hour (3^{rd}, 7^{th},...) is part of cycle $c = (4, 3)$.

The support of an itemset X in $D[i]$ is the fraction of transactions in $D[i]$ that contain X. The confidence of a rule $X \Rightarrow Y$ in $D[i]$ is the fraction of transactions in $D[i]$ containing X that also contain Y. Given the minimum support σ and the minimum confidence γ, an association rule $X \Rightarrow Y$ holds in time unit t_i if the support of $X \cup Y$ in $D[i]$ exceeds σ and the confidence of $X \Rightarrow Y$ in $D[i]$ exceeds γ.

An association rule has a cycle $c = (l, o)$ if the association rule holds in every i^{th} time unit starting with time unit t_o. Thus, if the unit of time is an hour and a rule $X \Rightarrow Y$ holds during the interval 8am-9am every day (i.e., every 24 hours), then $X \Rightarrow Y$ has a cycle $c = (24, 8)$. An association rule can have more than one cycle. For example, if $X \Rightarrow Y$ holds during the interval 8am-9am and 4pm-5pm every day, then $X \Rightarrow Y$ has two cycles $c_1 = (24, 8)$ and $c_2 = (24, 16)$.

A cycle can be a multiple of other cycles. A cycle (l_i, o_i) is a multiple of another cycle (l_j, o_j) if l_j divide l_i and $(o_j = o_i \bmod l_j)$. From this definition, a cycle $(24, 15)$ is a multiple of a cycle $(12, 3)$. Therefore, it is only interesting to discover cyclic association rules with 'large' cycles, i.e., the cycle that is not multiple of any other cycle.

Two algorithms are proposed to discover cyclic association rules. The first algorithm, called *sequential algorithm*, is the extension of non temporal association rule mining techniques which treats association rules and cycles independently. The second algorithm, which is called *interleaved algorithm*, uses optimization techniques for discovering cyclic association rules. This section discusses the sequential algorithm. The interleaved algorithm will be discussed in Section 2.6.

The sequential algorithm consists of two phases. First, the generation of the association rules that hold in each time unit. Second, the discovery of cyclic association rules using the cycle detection procedure as shown in Algorithm 2.2.

The first phase can be done using one of the existing methods (Agrawal & Srikant 1994, Savasere, Omiecinski & Navathe 1995). In (Ozden et al. 1998), the implementation is based on the *Apriori* algorithm from (Agrawal & Srikant 1994). In the second phase, it is required to represent each association rule as

Algorithm 2.2 Find cyclic association rules

```
// Phase 1: Generate rules for each time unit
for all time unit $t_i$ do
    Generate large itemsets in $t_i$
    Generate the rules from the large itemsets in $t_i$
end for
// Phase 2: Detect large cycles
Convert each rule into binary sequence
for all binary sequence do
    Generate the set of cycles for each rule
    Discover the large cycles from the set of cycles
end for
```

a binary sequence. In this representation, ones correspond to the time units in which the rule holds and zeros correspond to the time units in which the rule does not hold. For example, the binary sequence 001100010101 represents the association rule $X \Rightarrow Y$ holds in $D[2]$, $D[3]$, $D[7]$, $D[9]$, and $D[11]$. After each association rule is represented in a binary sequence, the cycle detection procedure can be started to discover cyclic association rules. From the above example the cycle detection procedure will discover a rule $X \Rightarrow Y$ has a cycle $c = (4, 3)$ because the rule holds in every fourth time unit starting from time unit t_3, followed by t_7 and t_{11}.

The cycle detection procedure is composed of two steps. In the first step, the sequence is scanned. Each time a zero is encountered at a sequence position i, candidate cycles $(j, i \bmod j)$, $1 \leq j \leq m$, where m is the maximum cycle length of interest, are eliminated from the set of candidate cycles. Initially, the set of candidate cycles contains all possible cycles. This step completes whenever the last bit of the sequence is scanned or the set of candidate cycles becomes empty, whichever is first. In the second step, large cycles (cycles that are not multiples of any existing cycles) are detected. This can be done by eliminating cycles that are not large as follows: starting from the shortest cycle, for each cycle $c_i = (l_i, o_i)$ eliminate each other cycle $c_j = (l_j, o_j)$ from the set of cycles if l_j is a multiple of l_i and $o_i = o_j \bmod l_i)$ holds.

2.4 Calendric Association Rules

In this section, we discuss the calendric association rules as described in (Li et al. 2001). In order to discover calendric association rules, the user needs only to give a simple calendar patterns belonging to a calendar schema. A calendar schema is a relational schema $R = (f_n : D_n, f_{n-1} : D_{n-1}, \ldots, f_1 : D_1)$. Each attribute f_i is a calendar unit name such as year, month, and week etc. Each domain D_i is a finite subset of the positive integers. Given a calendar schema $R = (f_n : D_n, f_{n-1} : D_{n-1}, \ldots, f_1 : D_1)$, a calendar pattern on the calendar schema R is a tuple on R of the form $\langle d_n, d_{n-1}, \ldots, d_1 \rangle$, where d_i is in D_i or the wild-card symbol "*". Each calendar pattern represents the time interval given by a set of tuples in D_n x D_{n-1} x ... x D_1. As an example, found in (Li et al. 2001), a calendar schema may be (year:{1995, 1996, ..., 1999}), month:{1, 2, ..., 12}, day:{1, 2, ..., 31}). The calendar pattern (1999, 12, *) represents *every day in December, 1999.*

A calendric association rule over calendar schema R is a pair (r, e), where r is an association rule and e is a calendar pattern on R. The paper defines two classes of calendric association rules: *precise-match* association rules and *fuzzy-match* association rules. Precise-match association rules require the rules to hold during every interval while fuzzy-match association rules require the rules to hold during a significant fraction of these intervals. More formal definition of precise-match and fuzzy-match association rules can be stated as follows.

1. (Precise match) Given a calendar schema R and a set of timestamped transactions $\mathcal{T}$, a precise-match association rule (r, e) holds in $\mathcal{T}$ if and only if the association rule r holds in $\mathcal{T}[t]$ for each basic time interval t covered by e.
2. (Fuzzy match) Given a calendar schema R, a set of timestamped transactions $\mathcal{T}$, and a real number m (*match ratio*), where $0 < m < 1$, a fuzzy-match association rule (r, e) holds in $\mathcal{T}$ if and only if the association rule r holds in $\mathcal{T}[t]$ for at least $100m\%$ of the basic time intervals t covered by e.

Similar to the problem of finding non-temporal association rules, the problem of finding calendric association rules can also be divided into two subproblems. First, finding all large itemsets for all calendar patterns, and second, generating calendric association rules using the large itemsets and their calendar patterns. The work in (Li et al. 2001) only focuses on finding calendric large itemsets. Two types of algorithms for finding calendric large itemsets are proposed and both are based on *Apriori* algorithm (Agrawal & Srikant 1994): *direct-Apriori* algorithms that treat each basic time interval individually and directly apply *Apriori* method (Agrawal & Srikant 1994) for candidate generation and *temporal-Apriori* algorithms that are optimized by exploiting the relationship among calendar patterns. In this section we only discuss the direct-Apriori algorithms to find both classes of calendric association rules. The temporal-Apriori algorithms will be discussed in section 2.6.

Algorithm 2.3 shows the direct-Apriori algorithm to generate precise-match and fuzzy-match large itemsets. In pass k of the algorithm, where $k > 1$, the processing of each basic time interval e_o consists of three phases. The first phase generates candidate k-itemsets for each basic time interval from large $(k-1)$-itemsets. This can be done by using *Apriori*'s method for candidate generation as follows: $C_k(e_o) = AprioriGen(L_{k-1}(e_o))$. The second phase scans the transactions whose timestamps are covered by the basic time interval, and discovers large k-itemsets for this basic time interval. These two phases are the same for both precise-match and fuzzy-match. In the third phase, the discovered large k-itemsets for the basic interval, $L_k(e_o)$, are used to update the large k-itemsets for each star calendar pattern e that covers the basic time interval. The update procedure for precise-match and fuzzy-match is different. For precise-match, the update is performed by intersecting the set of large k-itemsets for the basic interval e_o with the set of large k-itemsets for the calendar pattern e, i.e., $L_k(e) = L_k(e) \cap L_k(e_o)$. In the first update, $L_k(e) = L_k(e_o)$.

Algorithm 2.3 Find precise-match and fuzzy-match large itemsets

```
for all basic time interval $e_o$ do
  $L_1(e_o) = \{$large 1-itemsets in $\mathcal{T}[e_o]\}$
  for all star patterns $e$ that covers $e_o$ do
    update $L_1(e)$ using $L_1(e_o)$;
  end for
end for
for ($k = 2$; $\exists$ a star calendar pattern $e$ such that $L_{k-1}(e) \neq \emptyset$;$k++$) do
  for all basic time interval $e_o$ do
    // Phase 1: Generate candidates
    generate candidate $C_k(e_o)$;
    // Phase 2: Scan the transactions
    for all transactions $T \in \mathcal{T}[e_o]$ do
      // increment the count of $c \in C_k(e_o)$ contained in $T$
      subset $(C_k(e_o), T)$
    end for
    $L_k(e_o) = \{c \in C_k(e_o) | sup(c) \geq \sigma\}$
    // Phase 3: Update star calendar patterns
    for all star patterns e that covers $e_o$ do
      update $L_k(e)$ using $L_k(e_o)$;
    end for
  end for
  return $\langle L_k(e), e \rangle$ for all star calendar pattern $e$
end for
```

The update procedure for fuzzy-match is more complex than that for precise-match. The procedure is presented in Algorithm 2.4. Each large itemset $l \in L_k(e)$ is associated with the counter *updatecounter* which is initially set to 1.

Algorithm 2.4 Fuzzy-match Update

```
$N$ = the number of basic time interval covered by $e$
$n$ = the $n$-th update to $L_k(e)$
$m$ = match ratio
if $L_k(e)$ has never been updated then
  $L_k(e) = L_k(eo)$
  $l.updatecount = 1$ for each $l \in L_k(e)$
else
  $l.updatecount = 1$ for each $l \in L_k(e_o) - L_k(e)$
  $l.updatecount++$ for each $l \in L_k(e_o) \cap L_k(e)$
  $L_k(e) = \{l \in L_k(e) \cup L_k(e_o) | l.updatecount + (N - n) \geq m.N\}$
end if
```

2.5 Temporal Predicate Association Rules

The temporal predicate association rule was first proposed by (Rainsford & Roddick 1999). It extended the association rules by adding to the rules a conjunction of binary temporal predicates that specify the relationships between the timestamps of transactions. Binary temporal predicates are defined using thirteen interval based relations proposed by Allen (1983) and the neighborhood relationships defined by Freksa that allow generalisation of relationships (Freksa 1992).

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D be a dataset of transactions, where each transaction s consists of a set of items such that $s \subseteq I$. Given an *itemset* $X \subseteq I$, a transaction s *contains* X if and only if $X \subseteq s$. Associated with each transaction is temporal attributes that record the time for which the item is valid. Let $P_1 \wedge P_2 \ldots \wedge P_n$ be a conjunction of binary temporal predicates defined on attributes contained in either X or Y, where $n \geq 0$. The temporal predicate association rule is a rule of the form $X \Rightarrow Y \wedge P_1 \wedge P_2 \ldots \wedge P_n$, where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$.

The temporal confidence is used to determine how strong the temporal relationship between temporal items in the rule. The rule $X \Rightarrow Y \wedge P_1 \wedge P_2 \ldots \wedge P_n$ holds in a dataset D with the confidence c if and only if at least c% of transactions in D that contain X also contain Y. Likewise, each predicate P_i holds with a temporal confidence tc_{pi} if and only if at least tc% of transactions in D that contain X and Y also satisfy P_i. Following is an example of temporal predicate association rule found in (Rainsford & Roddick 1999):

$$\begin{gathered}\text{policyC} \Rightarrow \text{investA, productB} \mid 0.87 \wedge \\ \textit{during}(\text{investA,policyC}) \mid 0.79 \wedge \\ \textit{before}(\text{productB,investA})|0.91\end{gathered}$$

An interpretation for this rule is:

> The purchase of investment A and product B are associated with insurance policy C with a confidence factor 0.87. The investment A occurs during the period of policy C with a temporal confidence factor 0.79 and the purchase of product B occurs before investment A with a temporal confidence factor of 0.91.

The algorithm to discover temporal predicate association rules consists of four phases. The first phase of the algorithm can be performed using any association rule algorithm. During this phase the temporal attributes associated with the items are not considered. In the second phase all of possible pairings of temporal items in each generated rule are generated. For example, if the association rule is $AB \Rightarrow C$ then there are three possible pairings, AB, AC, and BC. In the third phase, the dataset is scanned and each tuple is checked to see if it supports a given rule. Finally, the aggregation of temporal relationships found in this phase is then concatenated with the original rule to generate temporal predicate association rules.

2.6 Improving Efficiency of the Algorithms

In mining non-temporal association rules, many variations of the *Apriori* algorithm that focus on improving the efficiency of the original algorithm have been proposed. One of the goals is to reduce the search space for large itemsets. In temporal mining the search space will be even larger than in non-temporal mining and thus optimization is required. This section discusses optimization techniques employed in discovering cyclic association rules and calendric association rules.

Cycle Pruning, Cycle Skipping and Cycle Elimination In the sequential algorithm (as discussed in Section 2.3), the cycle detection procedure has to scan and process every bit position of binary sequence sequentially. By exploiting the relationship between cycles and large itemsets this procedure can be optimized so that the sequence position that cannot be part of any of the candidate cycles is skipped. In order to do this the interleaved algorithm employs three optimization techniques: *cycle-pruning*, *cycle-skipping*, and *cycle-elimination*.

An itemset, in the same way an association rule, can be represented as a binary sequence where ones correspond to the time units in which the itemset is large and zeros correspond to the time units in which the itemset is not large. A cycle can also be represented as a binary sequence. A cycle $c = (l, o)$ is represented as a binary sequence with length l and the value of bit position o one. For example, a cycle $c = (4, 3)$ is represented as a binary sequence 0001.

Cycle-pruning is based on the property that if an itemset X has a cycle (l, o), then any subset of X has the cycle (l, o). It implies that, any cycle of itemset X must be a multiple of a cycle of an itemset that is subset of X and the number of cycles of an itemset X is less than or equal to the number of cycles of any subset of X. For example, if 010 is the only large cycle of A, and 010 is also the only large cycle of item B, then cycle-pruning implies that the itemset AB can have only the cycle 010 or its multiples.

Cycle-elimination is based on the property that if an itemset X is not large in time $D[i]$, then X cannot have the cycle $(j, i \bmod j)$, $l_{min} \leq j \leq l_{max}$. Cycle-elimination can be used to discard cycles that an itemset X cannot have as soon as possible. So if l_{max} is the maximum cycle length and the support for an itemset A is not large for the first l_{max} time units, then cycle-elimination implies that A cannot have any cycles.

Cycle-skipping is based on the property that if time unit t_i is not part of a cycle of an itemset X then there is no need to calculate the support of X in time segment $D[i]$.

The interleaved algorithm consists of two phases. In the first phase, the algorithm generates the cyclic large itemsets. In the second phase, cyclic association rules are generated from cyclic large itemsets. The first phase of the algorithm is shown in Algorithm 2.5. IHT_k contains candidate k-itemsets and their potential cycles, and THT contains the itemsets that are active in time unit t. An itemset is active in time unit t if it has a cycle that t participates in. Cycle-skipping determines, from the set of candidate cycles for k-itemsets, the set of k-itemsets for which support will be calculated in time segment $D[i]$. For a candidate itemset

X, if X is not large at time segment t_i then by applying cycle-elimination each cycle $c = (l, o)$, where ($o = i \bmod l$) holds, is removed from the set of potential cycles of X.

Algorithm 2.5 Find cyclic large itemsets

```
k = 1
while there are still candidate in $IHT_k$ with potential cycles do
    for all time unit $t_i$ do
        // apply cycle-skipping
        insert active itemsets from $IHT_k$ into $THT$
        count support of each itemset in $THT$ in $t_i$
        for all $X \in THT$ do
            if $sup(X) < \sigma$ then
                // apply cycle-elimination
                delete corresponding cycles of itemset $X$
            end if
        end for
        empty $THT$
    end for
    verify actual cycles of each member of $IHT_k$
    // apply cycle-pruning
    $IHT_{k+1}$ = generate new candidate of size $k + 1$
    $k = k + 1$
end while
```

Temporal AprioriGen and Horizontal Pruning Two optimization techniques discussed in this section focus on improving the candidate generation phase (phase 1) of the direct-Apriori algorithm shown in Algorithm 2.3. It can be seen that during the processing of each basic time interval e_o, the algorithm has to count the support of all the potentially large k-itemsets generated by $C_k(e_o) = AprioriGen(L_{k-1}(e_o))$. It is considered as inefficient because even if a candidate k-itemset could be large for e_o, it can be ignored if it cannot be large for any of the star calendar patterns that cover e_o.

Therefore, in order to optimize the candidate generation, the computation of $C_k(e_o)$ is based on $L_{k-1}(e_1))$, where e_1 is 1-star pattern that covers e_o, instead of $L_{k-1}(e_o))$. The optimized candidate generation procedure, called *temporal AprioriGen*, is shown in Algorithm 2.6.

As an example, let the calendar schema $R = (week : \{1, \ldots, 5\}, day : \{1, \ldots, 7\})$. Let large 2-itemsets $L_2(\langle 3, 2\rangle) = \{AB, AC, AD, AE, BC, BD, CD, CE\}$, $L_2(\langle *, 2\rangle) = \{AB, AC, AD, BC, BD, CE\}$, and $L_2(\langle 3, *\rangle) = \{AB, AC, AD, BD, CD\}$. The direct-Apriori algorithm generates the set of candidate large 3-itemsets $C_3(\langle 3, 2\rangle)$ from $L_2(\langle 3, 2\rangle)$. The result is $C_3(\langle 3, 2\rangle) = \{ABC, ABD, ACD, ACE, BCD\}$. On the other hand, if the temporal AprioriGen is used, it first generates $C_3(\langle *, 2\rangle) =$

Algorithm 2.6 Temporal AprioriGen

```
$C_k(e_o) = \emptyset$
**for all** 1-star patterns $e_1$ that covers $e_o$ **do**
    $C_k(e_o) = C_k(e_o) \cup AprioriGen(L_{k-1}(e_1))$
**end for**
```

$\{ABC, ABD\}$ and $C_3(\langle 3, *\rangle) = \{ABD, ACD\}$, then followed by $C_3(\langle 3, 2\rangle) = C_3(\langle *, 2\rangle) \cup C_3(\langle 3, *\rangle) = \{ABC, ABD, ACD\}$.

Even though the temporal AprioriGen procedure has been able to reduce the number of candidate k-itemsets generated, this number can be further reduced by using the second optimization technique, *horizontal pruning*, shown in Algorithm 2.7.

Algorithm 2.7 Horizontal Pruning

```
**if** $\exists$ 1-star pattern $e_1$ that covers $e_o$ such that $L_k(e_1)$ has not been updated even once **then**
    **return** $C_k(e_o)$
**end if**
$P = \emptyset$
**for all** 1-star patterns $e_1$ that covers $e_o$ **do**
    $P = P \cup L_k(e_1)$
**end for**
**return** $(C_k(e_o) \cap P)$
```

Continuing from an example above, suppose when the basic interval $\langle 3, 2\rangle$ is being processed, we already have $L_3(\langle *, 2\rangle) = \{ABD\}$ and $L_3(\langle 3, *\rangle) = \{ABD, ACD\}$. Given the generated set of candidate large 3-itemsets $C_3(\langle 3, 2\rangle) = \{ABC, ABD, ACD\}$, it can be further pruned by $C_3(\langle 3, 2\rangle) = C_3(\langle 3, 2\rangle) \cap (L_3(\langle 3, *\rangle) \cup L_3(\langle 3, *\rangle)) = \{ABD, ACD\}$.

3 A New Method for Relative Temporal Association Rule Mining

We now discuss a new method for mining relative temporal association rules and in particular some issues that pose problems in the process.

A relative temporal association rule, for the purposes of this work, is one structured as follows:

$$antecedents \overset{temprel}{\Longrightarrow} consequents, \qquad quals \tag{1}$$

where *temprel* is a temporal relationship taken from, for example, those suggested by Allen, Freksa and others, and *quals* is some rule quality qualifications (such as support).

In our current work, we propose here a new relative temporal association rule mining model. The model is proposed to discover an important form of temporal association rules which are useful but cannot be discovered with the existing temporal association rule mining framework. Taking medical data as an example, our rule can have the following form:

$$A, B, C \overset{<}{\Longrightarrow} D, E \tag{2}$$

which is equivalent to an assertion that *patients who have attributes (such as symptoms) A, B and C are also likely to later have recorded attributes D and E*. Ie. in this rule, the $<$ annotation means that A, B and C occur *before* D and E.

Unlike temporal association rule models discussed above, the proposed model takes into account transaction-id and transaction timestamp. For example, consider a scenario in which a patient is admitted to hospital at time t_i with symptom A. Later, at time t_j, the patient is readmitted with symptoms B and C. Other models consider this scenario as two different transactions. However, the proposed model offers the user the ability to view it as one (temporal) transaction. It means that each transaction-id can have more than one timestamp. Even though our transaction dataset is similar to the one used for mining sequential patterns discussed in (Agrawal & Srikant 1995), the rules generated by the proposed model are different.

Note that the temporal nature of the rule means that, unlike static rules, the same attribute can occur on both sides of the rule. Ie., in this model, it is possible to have the following rules:

$$A, B \overset{<}{\Longrightarrow} A \tag{3}$$

which means that attribute A and B implies the continuation or reoccurrence of A in later periods, while

$$D \overset{>}{\Longrightarrow} CD \tag{4}$$

means that attribute C and D were probably preceded by D in earlier time periods.

The algorithm consists of four phases: Sort phase, Litemset phase, Transformation phase, and Relative litemset phase.

1. Sort Phase. In this phase, the original transaction dataset is sorted, with customer-id as the primary key and transaction-time as the secondary key, as shown in Table 3.

2. Litemset Phase. In this phase, we find the set of all large itemsets (Litemsets) by using *Apriori* algorithm. In doing so, the definition of support is modified. Originally, the support of an itemset is defined as the fraction of transactions in which an itemset is present. Here, the support an itemset is defined as the fraction of transactions in which an item appears (at any time). For example,

CustID	Timestamp	Items							
1	25-Jun-03								H
1	30-Jun-03			C					
2	10-Jun-03	A	B		D				
2	15-Jun-03			C					
2	20-Jun-03				D		F	G	
2	25-Jun-03		B						
3	25-Jun-03			C		E		G	
4	25-Jun-03			C					
4	30-Jun-03				D			G	
4	25-Jul-03				D				
4	30-Jul-03		B						

Table 3. Sorted dataset

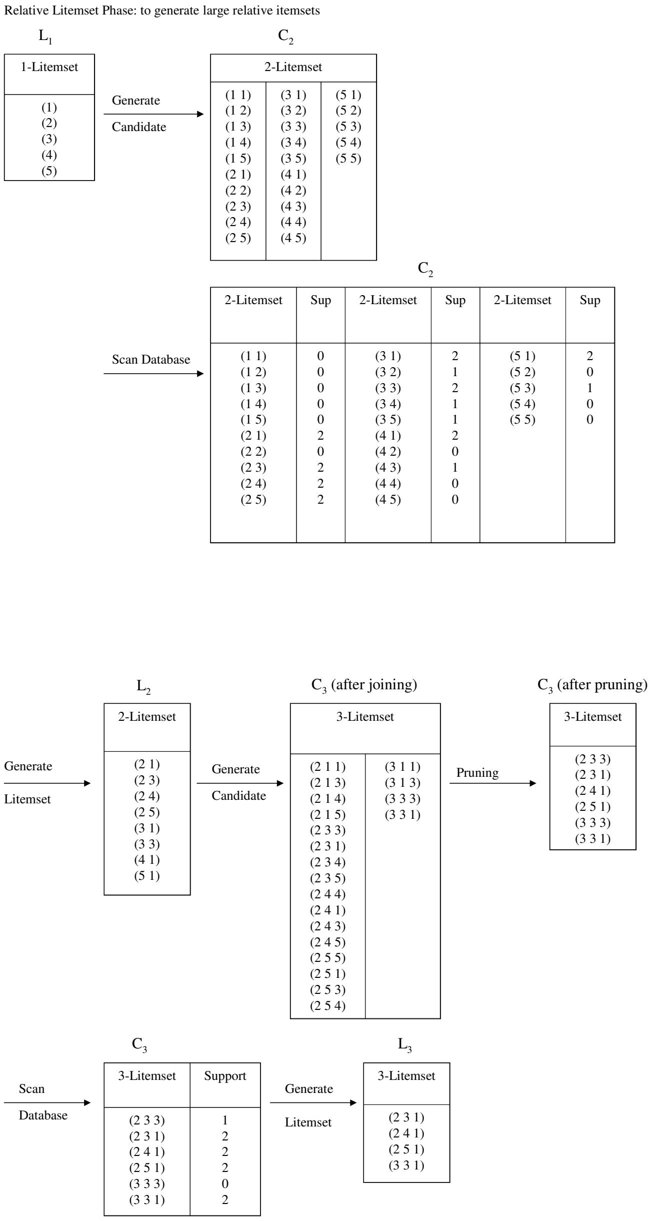
even though the item B appears three times in the transaction dataset, its support is 2 as it appears for only the second and fourth transaction. By taking a minimum support of 2, the large itemsets that can be generated from transaction dataset above is shown in table 4.

Large Itemsets	Support	Mapped To
{B}	2	1
{C}	4	2
{D}	2	3
{G}	3	4
{DG}	2	5

Table 4. Large Itemsets

3. Transformation Phase. Before the transformation phase takes place, the set of large itemsets generated in previous phase is first mapped into a set of integers, as shown in Table 4, column three. The result of transformation phase is shown in table 5. Each transaction is replaced by the set of all litemsets contained in that transaction. As an example, the first transaction of the second customer (A B D H) is transformed into a set of litemset {(D), (H)}. If a transaction does not contain any litemset, it is not retained in the transformed sequence. The first transaction of the first customer does not contain any litemset, thus it is drop from the transformed customer sequence, as shown in table 5. If a transformed customer sequence only contains one set of litemsets, this sequence is drop from the transformed sequence. The customer sequence that has only one set of litemset will not produce relative itemsets. Therefore, customer sequence of the first and the third customer are drop from the transformed sequence. However, they are still being used to the count of total number of customers.

4. Relative Itemset Phase. This phase is used to generate large relative itemsets. The algorithm is based on the *Apriori* algorithm and is shown in Algorithm 3.1.



Relative Litemset Phase: to generate large relative itemsets

L_1

1-Litemset
(1)
(2)
(3)
(4)
(5)

Generate Candidate

C_2

2-Litemset		
(1 1)	(3 1)	(5 1)
(1 2)	(3 2)	(5 2)
(1 3)	(3 3)	(5 3)
(1 4)	(3 4)	(5 4)
(1 5)	(3 5)	(5 5)
(2 1)	(4 1)	
(2 2)	(4 2)	
(2 3)	(4 3)	
(2 4)	(4 4)	
(2 5)	(4 5)	

Scan Database

C_2

2-Litemset	Sup	2-Litemset	Sup	2-Litemset	Sup
(1 1)	0	(3 1)	2	(5 1)	2
(1 2)	0	(3 2)	1	(5 2)	0
(1 3)	0	(3 3)	2	(5 3)	1
(1 4)	0	(3 4)	1	(5 4)	0
(1 5)	0	(3 5)	1	(5 5)	0
(2 1)	2	(4 1)	2		
(2 2)	0	(4 2)	0		
(2 3)	2	(4 3)	1		
(2 4)	2	(4 4)	0		
(2 5)	2	(4 5)	0		

Generate Litemset

L_2

2-Litemset
(2 1)
(2 3)
(2 4)
(2 5)
(3 1)
(3 3)
(4 1)
(5 1)

Generate Candidate

C_3 (after joining)

3-Litemset	
(2 1 1)	(3 1 1)
(2 1 3)	(3 1 3)
(2 1 4)	(3 3 3)
(2 1 5)	(3 3 1)
(2 3 3)	
(2 3 1)	
(2 3 4)	
(2 3 5)	
(2 4 4)	
(2 4 1)	
(2 4 3)	
(2 4 5)	
(2 5 5)	
(2 5 1)	
(2 5 3)	
(2 5 4)	

Pruning

C_3 (after pruning)

3-Litemset
(2 3 3)
(2 3 1)
(2 4 1)
(2 5 1)
(3 3 3)
(3 3 1)

Scan Database

C_3

3-Litemset	Support
(2 3 3)	1
(2 3 1)	2
(2 4 1)	2
(2 5 1)	2
(3 3 3)	0
(3 3 1)	2

Generate Litemset

L_3

3-Litemset
(2 3 1)
(2 4 1)
(2 5 1)
(3 3 1)

Fig. 1. Generation of candidate and large relative itemsets

Cust ID	Original Customer Sequence	Transformed Customer Sequence	After Mapping
1	(H) (C)	{(C)}	
2	(A B D H) (C) (D F G)	{(B), (D)} {(C)} {(D), (G), (DG)} {B}	{1, 3} {2} {3, 4, 5} {1}
3	(C E G)	{(C), (G)}	
4	(C) (D G) (D H)	{(C)} {D), (G), (DG)} {(D)} {B}	{2} {3, 4, 5} {3} {1}

Table 5. Transformed dataset

Algorithm 3.1 Generate Large Relative Itemset

```
$L_1$ = {Large relative 1-itemset};
for ($k = 2; L_{k-1} \neq \emptyset; k++$) do
    $C_k$ = Candidate generation from $L_{k-1}$;
    for each customer sequence $s$ in the dataset do
        Increment the count of all candidate in $C_k$ that are contained in $s$;
    end for
    $L_k$ = Candidate in $C_k$ with minimum support;
end for
```

In the algorithm, L_k is a set of large relative k-itemsets. The set of large relative 1-itemset L_1 is initialized by using the the set of large itemsets generated in the litemset phase. The candidate generation to obtain C_k is generated by joining L_{k-1}. Let p and q be large relative $k-1$-itemsets in L_{k-1}. In *Apriori*, p and q are joinable if their first $(k-2)$ items are in common, and the resulting itemset is $p[1]p[2]\ldots p[k-1]q[k-1]$, where $p[k-1] < q[k-1]$. In this algorithm, the requirement that $p[k-1] < q[k-1]$ is removed, thus the itemset p can be joined to itself. The join procedure is shown below:

```
insert into $C_k$
        select $p[1], p[2], \ldots, p[k-1], q[k-1]$
        from $L_{k-1}\ p, L_{k-1}\ q$
        where $p[1] = q[1], \ldots, p[k-2] = q[k-2]$
```

The pruning procedure will delete all candidate $c \in C_k$ if some $k-1$-itemset of c is not in L_{k-1}.

Using the transformed dataset in Table 5, the generation of candidate and large relative itemsets is shown in Figure 1. In this figure, the k-itemset $(I_1 I_2 \ldots I_k)$ represents the relationship $I_1 < I_2 < \ldots < I_k$. Therefore, the large relative 3-itemset (2 5 1) represents the relationship $2 < 5 < 1$, which has the interpretation that C occurs *before* DG and DG occurs *before* B.

4 Conclusions and Future Work

In this paper, we have surveyed temporal association rule models. In order to do that, we have classified the models based on four different aspects of the models: attribute domains, measures, temporal features, and algorithms. This is followed by the discussion of the algorithms to discover interval association rules, cyclic association rules, calendric association rules, and binary predicate association rules. We also discuss how the algorithms to discover cyclic and calendric association rules can be optimized.

We also discuss a new method to generate relative temporal association rules. At this stage the basic algorithm is complete but substantial further work is required in order to optimise and operationalise the ideas. Nevertheless, the algorithm looks promising and all indications are that this will be a useful new method.

References

Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, *in* 'Proceedings of ACM Conference on Management of Data', pp. 207–216.

Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules, *in* J. B. Bocca, M. Jarke & C. Zaniolo, eds, 'Proceedings of the 20th International Conference on Very Large Data Bases', Morgan Kaufmann, pp. 487–499.

Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, *in* P. S. Yu & A. S. P. Chen, eds, 'Proceedings of 11th International Conference on Data Engineering', pp. 3–14.

Ale, J. M. & Rossi, G. H. (2000), An approach to discovering temporal association rules, *in* 'Proceedings of the 2000 ACM Symposium on Applied Computing', pp. 294–300.

Allen, J. F. (1983), 'Maintaining knowledge about temporal intervals', *Communications of the ACM* **26**(11), 832–843.

Antunes, C. M. & Oliveira, A. L. (2001), Temporal data mining: An overview, *in* 'Proceedings of the Knowledge Discovery and Data Mining (KDD2001) Workshop on Temporal Data Mining', San Francisco, EUA.

Brin, S., Motwani, R., Ullman, J. D. & Tsur, S. (1997), Dynamic itemset counting and implication rules for market basket data, *in* J. Peckham, ed., 'Proceedings ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, USA', ACM Press, pp. 255–264.

Chandra, R., Segev, A. & Stonebraker, M. (1994), Implementing calendars and temporal rules in next generation databases, *in* 'Proceedings of the 10th International Conference on Data Engineering (ICDE)', Houston, TX, pp. 264–273.

Chen, X. & Petrounias, I. (2000), An integrated query and mining system for temporal association rules, *in* Y. Kambayashi, M. K. Mohania & A. M. Tjoa, eds, 'Proceedings of the Second International Conference, DaWaK 2000', London, UK, pp. 327–336.

Freksa, C. (1992), 'Temporal reasoning based on semi-intervals', *Artificial Intelligence* **54**(1), 199–227.

Han, J. & Kamber, M. (2001), *Data Mining : Concepts and Techniques*, Academic Press, San Diego.

Leban, B., McDonald, D. & Forster, D. (1986), A representation for collections of temporal intervals, *in* 'Proceedings of the AAAI-1986 5th Int. Conf. on Artificial Intelligence', pp. 367–371.

Lee, C.-H., Lin, C.-R. & Chen, M.-S. (2001), On mining general temporal association rules in a publication database, *in* 'Proceedings of the First IEEE International Conference on Data Mining'.

Li, Y., Ning, P., Wang, X. S. & Jajodia, S. (2001), Discovering calendar-based temporal association rules, *in* 'Proceedings of the 8th International Symposium on Temporal Representation and Reasoning', pp. 111–118.

Ozden, B., Ramaswamy, S. & Silberschatz, A. (1998), Cyclic association rules, *in* 'Proceedings of the 14th International Conference on Data Engineering', pp. 412–421.

Park, J. S., Chen, M.-S. & Yu, P. S. (1995), An effective hash based algorithm for mining association rules, *in* M. J. Carey & D. A. Schneider, eds, 'Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data', San Jose, California, pp. 175–186.

Rainsford, C. P. & Roddick, J. F. (1999), Adding temporal semantics to association rules, *in* J. M. Zytkow & J. Rauch, eds, 'Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'99)', pp. 504–509.

Rainsford, C. P. & Roddick, J. F. (2000), Visualisation of temporal interval association rules, *in* 'Proceedings of 2nd International Conference on Intelligent Data Engineering and Automated Learning, (IDEAL 2000), Shatin, N.T., Hong Kong', Vol. 1983 of *Lecture Notes in Computer Science*, Springer, pp. 91–96.

Ramaswamy, S., Mahajan, S. & Silberschatz, A. (1998), On the discovery of interesting patterns in association rules, *in* 'Proceedings of the 24th International Conference on Very Large Data Bases', pp. 368–379.

Roddick, J. F. & Spiliopoulou, M. (2002), 'A survey of temporal knowledge discovery paradigms and methods', *IEEE Transactions on Knowledge and Data Engineering* **14**(4), 750–767.

Savasere, A., Omiecinski, E. & Navathe, S. B. (1995), An efficient algorithm for mining association rules in large databases, *in* 'Proceedings of the 21th International Conference on Very Large Data Bases', pp. 432–444.

Srikant, R. & Agrawal, R. (1996), Mining quantitative association rules in large relational tables, *in* 'Proceedings of ACM SIGMOD International Conference on Management of Data', pp. 1–12.

Wang, W., Yang, J. & Muntz, R. (1999), Temporal association rules with numerical attributes, Technical Report 980031, UCLA CSD.

Wang, W., Yang, J. & Muntz, R. R. (2001), TAR: Temporal association rules on evolving numerical attributes, *in* 'Proceedings of the Seventeenth International Conference on Data Engineering, ICDE 2001', IEEE Computer Society, Heidelberg, Germany, pp. 283–292.

Zimbrão, G., de Souza, J. M., de Almeida, V. T. & da Silva, W. A. (2002), An algorithm to discover calendar-based temporal association rules with item's lifespan restriction, *in* 'Proceedings of The Second Workshop on Temporal Data Mining The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining'.

A Study of Drug-Reaction Relations in Australian Drug Safety Data

M. A. Mamedov G. W. Saunders and E. Dekker

School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria, 3353, Australia

Contact: G. W Saunders, PhD Student ITMS University of Ballarat
University Drive Mount Helen 3353 Victoria Australia
phone: 613 5327 9376
email: g.saunders@ballarat.edu.au
fax: 613 5327 9289
M. A. Mammadov
Postdoctoral Fellow
phone: 613 5327 9336
email: m.mammadov@ballarat.edu.au
E. Dekker
Research Assistant
phone: 613 5327 9336
email: e.dekker@ballarat.edu.au

Keywords: knowledge representation, adverse drug reaction, text categorization, multi-label classes, suspected drugs.

Abstract

This paper studies drug-reaction relationships using the system organ class grouping of reactions from Australian drug safety data. For each drug we define a vector of weights which indicates the "probability" of occurrence of reactions. Such a representation of drug-reaction associations and the accuracy of established representations are evaluated applying two algorithms: the well-known text categorization algorithm BoosTexter and a new algorithm introduced in this paper. We use different evaluation measures. The ways of developing reasonable distance measures is investigated and discussed. This novel use of text categorization type algorithms provides a broader perspective for the development of new algorithms to study drug-reaction relationships.

1 Introduction

An Adverse Drug Reaction (ADR) is defined by the World Health Organization (WHO) as: “a response to a drug that is noxious and unintended and occurs at *doses normally used* in man for the prophylaxis, diagnosis or therapy of disease, or for modification of physiological function” [34]. ADRs are estimated to be the fourth leading cause of death in the USA [26], and the amount of published literature on the subject is vast [1]. Some of the problems concerning ADRs are discussed in our research report [19]. Many approaches have been tried for the analysis of adverse reaction data, such as: Fisher’s Exact Test and matched pair designs (McNemar’s test) [30], Reporting Odds Ratio (ROR), and Yule’s Q [32]. One approach that has had some success is the Proportional Reporting Ratios (PRR) for generating signals from data in the United Kingdom. One problem with this method is that very striking signals for a particular drug will reduce the magnitude of the PRR for other signals with that drug due to inflation of the denominator [12]. The Norwood-Sampson Model has been applied to data in the United States of America and approved by the Food and Drug Administration (FDA), despite some bias inherent in the model – Hillson *et al.* propose a modification of the Norwood-Sampson method to adjust for this [17]. A common approach to the assessment of ADRs uses the Bayesian method [8]. For example, the Bayesian confidence propagation neural network (BCPNN) [2], and an empirical Bayesian statistical data mining program called a Gamma Poisson Shrinker (GPS) [10], and the Multi-item Gamma Poisson Shrinker (MGPS) [28], which have been applied to the United Sates Food and Drug Administration (FDA) Spontaneous Reporting System database. The Bayesian method has met with success, but is very exacting regarding the quantification of expectations [18]. Each method has its own advantages and disadvantages in respect of applicability in different situations and possibilities for implementation [32]. However, these methods are still prone to problems of interpretation such as Simpson’s paradox, when two variables appear to be related because of their mutual association with a third variable [10]. The relative merits of these methods is difficult to assess due to the lack of a ‘gold standard’.

One of the main problems of ADR is the following: given a patient (the set of drugs taken and reactions reported) to identify drug(s) which are responsible for these reactions. Such drugs are termed, in the ADRAC database, “suspected” drugs. The accurate definition of suspected drugs for each report has a very important effect on the quality of the database for future study of drug-reaction relationships.

The authors of this paper are developing an alternative approach to the ADR problem [19], [20], [21]. We formulate the main goal of our ADR study as follows: Given a patient having taken some drug(s), to be able to predict what kind of reaction(s) can occur. This problem relys on the availability of suitable data from a sufficient number of reports of ADR cases using well developed methods [1], [3], [4], [15], [23], [26], [29].

In general, the necessary information can be divided into two groups: individual patient information including “reason for use”, “history” and so on, and information about drug(s) including dosage, duration and so on.

We simplify the main ADR problem assuming that the main goal can be achieved by solving the following problems *separately*:

P1. To study drug-reaction relationships not involving any other patient information;

that is, for each drug to define all the possible reactions that can occur (with corresponding weights).

P2. To predict the possible reaction(s) for a particular patient using both the patient information and the drug-reaction relationships, which requires data mining techniques.

The study of useful information about patients and the collection of such kinds of information are important problems [15], [22] and there is an initiative to add consumer ADR reporting [7], [9], [13], [31]. In ADRAC data that we consider in this paper, some data fields are very poorly presented for easy analysis. This is particularly the case with drug dosage information as well as other important fields having many records with missing values (see [19]), which highlights the advantage of studying the problems **P1** and **P2** separately.

In this paper we will consider the first problem **P1**. There are some issues which complicate the study of this problem: dosages, frequency and duration of drugs taken. The use of dosage information is difficult because in the ADRAC data different units (liters, grams, and so on), which makes standardization and scaling a significant problem. At this stage of investigation we decided to neglect the dosage information and to apply two different values: "Yes" – if drug was taken and "No" if not. This is, a common assumption used by many researchers (because of the assumption of normal dose in the definition of an ADR – see above), although not explicitly stated (for example see – [11], [14], [25], [33], [36]).

By understanding the drug-reaction relationship in the absence of information about other factors influencing this relationship we expect to be able to establish a clearer relationship between drugs and reactions. When these relationships become characterized more clearly, this knowledge can be applied to the more general study of the total dataset. Another reason for focussing primarily on drugs and reactions relates to the inconsistent quality and quantity of relevant data on factors which also play a role in the drug-reaction association.

1.1 ADRAC Data

The Australian Adverse Drug Reaction Advisory Committee (ADRAC) database has been developed and maintained by the Therapeutic Goods Administration (TGA) with the aim to detect signals from adverse drug reactions as early as possible. The ADRAC data contains 137,297 voluntarily reported adverse drug reaction records involving 5057 different drugs, based on the 'drug dictionary' used by ADRAC of 7416 different drug terms, and 1224 different reactions, based on 1392 different reaction terms. A more detailed account of the ADRAC database is given in [19]. Much of this data on ADRs is derived from voluntary reporting, some of the problems and advantages of such a reporting system are discussed in [3], [23], [29], [33], [5], [16], [24].

The biggest challenge in summarizing safety data is the need to consolidate the massive amount of data into a manageable format. One way is to group the safety data into K classes characterized by body systems and determined in conjunction with underlying disease and treatments involved. Such pooling of data through coding is especially helpful for rare events [6], [23]. ADRAC uses 18 systems organ class (SOC) reaction term classes, which we use to group the 1224 different reaction terms in the ADRAC data collected from 1971–2001 and called this dataset **Mallreac**. The number of reaction terms and number of occurrences for each class is shown in Table 1, where the occurrence is a cumulative count for each reaction class.

Table 1: System Organ Classes in ADRAC **Number of reaction terms in class †Total occurrence of reaction class – see text*

Code	System Organ Class Name	Terms*	Occurrence †
0100	Skin and appendages disorders	101	57269
0200	Musculo-skeletal system disorders	55	8881
0300	Collagen disorders	21	827
0400	Nervous system and special senses	276	79149
0500	Psychiatric disorders	57	24406
0600	Gastro-intestinal system disorders	144	41093
0700	Liver and biliary system disorders	54	11026
0800	Metabolic and nutritional disorders	84	11631
0900	Endocrine disorders	48	2352
1000	Cardiovascular system	164	27488
1100	Respiratory system disorders	66	14658
1200	Haemic and lymphatic systems	192	18703
1300	Urinary system disorders	56	8893
1400	Reproductive system	121	3998
1500	Foetal disorders	89	893
1600	Neonatal and infancy disorders	63	202
1700	Neoplasm	70	397
1800	Body as a whole	194	59273

2 Statement of the problem

We formulate the problem **P1**. It has become clear that the approach we are using to analyze ADRs bears a resemblance to the approach of others to the problem of text categorization. For a review of some of the issues in text categorization see [35].

Let $\mathcal{X}$ denote the set of all patients and $\mathcal{D}$ denote the set of all drugs used by these patients. Let c be a finite number of possible reactions (classes). Given patient $x \in \mathcal{X}$ we denote by $\mathcal{D}(x)$ the set of drugs taken by this patient. In ADRAC data the number of drugs reported for a patient is restricted to 10. We also denote by $\mathcal{Y}(x) = (\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_c)$ an c-dimensional vector of reactions observed for the patient x; where $\mathcal{Y}_i = 1$ if the reaction i has occurred, and $\mathcal{Y}_i = 0$ if it has not.

The goal of the study of drug-reaction relationships is to find a function $h : \mathcal{D} \rightarrow R^c_+$, where given drug $d \in \mathcal{D}$ the components h_i of the vector $h(d) = (h_1, h_2, \ldots, h_c)$ associate the weights ("probabilities") of the occurrence of the reactions $i = 1, 2, \ldots, c$.

Here R^c_+ is the set of all c-dimensional vectors with non-negative coordinates.

In the next step, given a set of drugs $\Delta \subset \mathcal{D}$, we need to define a vector $H = (H_1, H_2, \ldots, H_c)$, where the component H_i indicates the probability of occurrence of the reaction i after taking the drugs Δ. In other words, we need to define a function $H : S(\mathcal{D}) \rightarrow R^c_+$, where $S(\mathcal{D})$ is the set of all subsets of $\mathcal{D}$. The function H can be defined in different ways and it is an interesting problem in terms of ADR(s). We will briefly discuss this problem below.

Given patient $x \in \mathcal{X}$ and a set of drugs $\mathcal{D}(x)$, we will use the notation $H(x) = H(\mathcal{D}(x))$.

Therefore, we will denote a classifier as the couple (h, H). To evaluate the performance of different classifiers we need to measure the closeness of the two vectors $H(x)$ and $\mathcal{Y}(x)$. In this work we will use different evaluation measures presented in the Section on *Evaluation Measures.*

In this statement, the problem **P1** is a multi-class, multi-label text categorization problem, but there are some interesting points that should be mentioned in relation to **P1.** One of the main characteristics of ADRs is that the number of drugs (that is, words in the context of text categorization) for each patient is restricted to 10 in the ADRAC data, and for majority of patients just one drug was used. This complicates learning and classification, but on the other hand, this allows us to introduce simple and fast algorithms.

It also should be noted that, at this stage of our investigation, the classification of reactions is not the major aim. Here we aim to establish drug-reaction relations $h(d)$ such that representations $H(x)$, $x \in \mathcal{X}$, are close to $\mathcal{Y}(x)$. This makes the problem **P1** more than just a classification problem. Thus, we mainly concentrate on drug-reaction relations. Some other characteristics of the problem **P1** which are of interest in terms of ADRs will be discussed below.

2.1 Potential Reactions

The vectors $h(d)$ show what kind of reactions are caused by the drugs $d \in \mathcal{D}(x)$. Therefore the vector $H(x)$ can be considered as potential reactions which could occur with patient x. But what kind of reactions will occur? This will depend upon the individual characteristics of the patient as well as external factors [6]. Different patients can have different predispositions for different reactions. Some reactions which have potentially high degrees may not be observed because of the strong resistance of the patient to developing these reactions. But the existence of these potential reactions could have an influence on the patient somehow. The results obtained in [19] show that the information about the existence of potential reactions (but which were not reported to ADRAC) helps to make prediction of reaction outcomes (*bad* and *good*) more precise.

The function H can be defined in different ways. The study of more sensible definitions of the function H is an interesting problem for future investigations. This problem is also related to the study of Interaction of Drugs [19]. In the calculation below we will use the following linear function H, which provided more accurate classification in [19]: $H = (H_1, \dots, H_c)$; where for each subset $\Delta \subset \mathcal{D}$ the components H_i are defined as follows: $H_i = \sum_{d \in \Delta} h_i(d)$, $i = 1, \dots, c$. In this case, for each patient $x \in \mathcal{X}$, we have $H(x) = (H_1(x), \dots, H_c(x))$, where

$$H_i(x) = \sum_{d \in \mathcal{D}(x)} h_i(d), \quad i = 1, \dots, c. \tag{2.1}$$

The use of this function means that, we accumulate the effects from different drugs. For example, if $h_i(d_n) = 0.2$ (n=1,2) for some reaction i, then there exists a potential of 0.4 for this reaction; that is, the two small effects (i.e. 0.2) become a greater effect (i.e. 0.4). This method seems a more natural one, because physically both drugs are taken by the patient, and the outcome could even be worse if there were drug-drug interaction(s).

The other important issue that is related to the definition of H, is the time factor; that is, the time when the administration of each drug $d \in \mathcal{D}(x)$ ceases and the decay function of the drug. In a simple case we can describe this decay by the function $exp\,(t_d - t)$, where $t \geq t_d$ and t_d is the time of cessation of drug d. Then we can use the following formula for the definition of H :

$$H_i(x) \; = \sum_{d \in \mathcal{D}(x)} f(t_d - t^*)\, h_i(d), \quad i = 1, \ldots, c; \tag{2.2}$$

where t^* is the time of onset of reaction(s) and $f(t_d - t) = exp\,(t_d - t)$ if $t \geq t_d$, and $f(t_d - t) = 0$ if $t < t_d$. The application of the formula (2.2) is a very interesting problem for future investigations which also needs to take into account many other factors such as the metabolism and elimination of drugs. Drug exposure time is another important factor to take into consideration [6].

3 Evaluation measures

To evaluate the accuracy of established drug-reaction relations by a given classifier (h, H), we will use different measures.

3.1 Average Distance

This measure evaluates the closeness of the two vectors $H(x)$ (predicted reactions) and $\mathcal{Y}(x)$ (observed reactions). In this case, we define a distance $dist(H(x), \mathcal{Y}(x))$ between these vectors. The better classifier should provide the minimal sum of all distances. Therefore, we are looking for a classifier (h, H) which minimizes the total sum of distances.

A common evaluation measure used in multi-label problems is the Hamming distance. But it is not reasonable to use this distance here, because we deal with real valued weights $H(x)$.

In this paper we will examine the following distance functions:

$$dist_p\,(H(x), \mathcal{Y}(x)) = \sum_{i=1}^{c} (\|\mathcal{Y}(x)\|)^{-p}\; (\bar{H}_i - \mathcal{Y}_i)^2, \;\; p = 0, 1, 2; \tag{3.3}$$

where $H(x) = (H_1(x), \ldots, H_c(x))$, $\mathcal{Y}(x) = (\mathcal{Y}_1(x), \ldots, \mathcal{Y}_c(x))$, $\|\mathcal{Y}(x)\| = \sum_{j=1,\ldots,c} \mathcal{Y}_j(x)$ is the number of reactions for the patient x, and the sign “bar” indicates a normalization:

$$\bar{H}_i(x) = \frac{\|\mathcal{Y}(x)\|}{\sum_{j=1,\ldots,c} H_j(x)}\, H_i(x).$$

The role of number p can be explained as follows. Clearly

$$dist_p\left(H(x),\mathcal{Y}(x)\right)=\sum_{i=1}^{c}\Big((\|\mathcal{Y}(x)\|)^{-\frac{p}{2}}\,\bar{H}_i(x)-\|\mathcal{Y}(x)\|)^{-\frac{p}{2}}\,\mathcal{Y}_i(x)\Big)^2,$$

and therefore, potential reactions are normalized such that the sum of these normalized potential reactions can be represented as

$$\sum_{i=1}^{c}(\|\mathcal{Y}(x)\|)^{-\frac{p}{2}}\,\bar{H}_i(x)=(\|\mathcal{Y}(x)\|)^{1-\frac{p}{2}}.$$

In the distance $dist_0$ this sum is equal to the number of reactions $\|\mathcal{Y}(x)\|$. In $dist_2$ we get the corresponding sum is equal to 1. $dist_1$ can be considered as a middle version where this sum is $\sqrt{\|\mathcal{Y}(x)\|}$, and $1\leq\sqrt{\|\mathcal{Y}(x)\|}\leq\|\mathcal{Y}(x)\|$.

It would be interesting to consider the Euclidian distance. But some preliminary analysis showed that this distance does not provide us a reasonable evaluation.

Therefore, we will examine only the measures 3.3. Given a classifier (h,H), the average distance error will be calculated as

$$E^p_{av}=\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}dist_p\left(H(x),\mathcal{Y}(x)\right)\tag{3.4}$$

Here $|\mathcal{X}|$ stands for the cardinality of the set $\mathcal{X}$.

Now we formulate the problem **P1** as the following optimization problem:

$$E^p_{av}\quad\rightarrow\quad\min;\tag{3.5}$$

$$\text{subject to :}\qquad h_i(d)\geq 0,\quad i=1,...,c,\;\; d\in\mathcal{D}.\tag{3.6}$$

In this paper we will describe an algorithm which aims to minimize the average distance error E^p_{av}. This aim changes by taking different numbers $p=0,1,2$. A discussion of the most reasonable choice of distance measures is one of the main goals of this paper.

3.2 Other evaluation measures

We will also consider the following measures used in [27].

1. One-error. This measure evaluates how many times the reaction, (say i) having the maximal weight in the vector $H(x)$, has not occurred (that is, $\mathcal{Y}_i(x)=0$). In the case where there is more than one reaction, having the same maximal weight in $H(x)$, we need to precisely define this measure.

Denote $H^*(x) = \{i \in \{1,\dots,c\} : H_i(x) = \max\{H_1(x),\dots,H_c(x)\}\}$, and $Y^*(x) = \{i \in \{1,\dots,c\} : i \in H^*(x) \text{ and } \mathcal{Y}_i(x) = 1\}$. Then we define the one-error as

$$E_{one-error} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left(1 - \frac{|Y^*(x)|}{|H^*(x)|}\right).$$

The meaning of this measure can be clarified using a simple example. Assume that there are two reactions having the maximal weight. If both of them have occurred then the error is zero, if just one of them has occurred then error is 0.5, if none have occurred then the error is 1.

2. Coverage. This measure evaluates the performance of a classifier for all the reactions that have been observed.

Given $x \in \mathcal{X}$, we denote by $\mathcal{T}(x)$ the set of all ordered reactions $\tau = \{i_1,\dots,i_c\} = \{1,\dots,c\}$ satisfying $H_{i_1}(x) \geq \ldots \geq H_{i_c}(x)$. Then according to a reaction vector $(\mathcal{Y}_1(x),\dots,\mathcal{Y}_c(x))$, we define the rank and the error as:

$$rank_\tau(x) = \max\{n : \mathcal{Y}_{i_n}(x) = 1,\, n = 1,\dots,c\}; \quad error_\tau(x) = \frac{rank_\tau(x)}{\|\mathcal{Y}(x)\|} - 1.$$

Obviously, the number $rank_\tau(x)$ and $error_\tau(x)$ depend on the order τ. One way to avoid the dependence on ordering is to take the middle value of maximal and minimal ranks. In this paper we will use this approach. We define the rank as

$$rank(x) = \frac{1}{2}(rank_{max}(x) + rank_{min}(x));$$

where

$$rank_{max}(x) = \max_{\tau \in \mathcal{T}(x)} rank_\tau(x), \quad \text{and} \quad rank_{min}(x) = \min_{\tau \in \mathcal{T}(x)} rank_\tau(x).$$

The numbers $rank_{max}(x)$ and $rank_{min}(x)$ associated to the “worst” and “best” ordering, respectively.

To define the average error - coverage, we will use the formula:

$$E_{cov} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left(\frac{rank(x)}{\|\mathcal{Y}(x)\|} - 1\right).$$

Note that, $E_{cov} = 0$ if a classifier makes a prediction such that for all $x \in \mathcal{X}$ the observed reactions are placed at the top of the ordering list of weights $H_i(x)$.

3. Average Precision.

One-error and coverage do not completely describe multi-label classification problems. In [27] the average precision was used to achieve evaluation more completely. We also will use this measure. Similar to the average error, the average precision depends on a given order $\tau = \{\tau_1, \ldots, \tau_c\} \in \mathcal{T}(x)$. So we define the average precision as a middle value of average precisions obtained by the "worst" and "best" ordering.

Let $Y(x) = \{l \in \{1, \ldots, c\} : \mathcal{Y}_l(x) = 1\}$ be a set of reactions that have been observed for the patient x. Given order $\tau = \{\tau_1, \ldots, \tau_c\} \in \mathcal{T}(x)$, (that is, $H_{\tau_1}(x) \geq \ldots \geq H_{\tau_c}(x)$), we define the rank for each reaction $l \in Y(x)$ as $rank_\tau(x; l) = k$, where $\tau_k = l$. Then, Average Precision will be defined as:

$$P_{av} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \frac{1}{2|Y(x)|} \left(P_{worst}(x) + P_{best}(x) \right);$$

where

$$P_{worst}(x) = \min_{\tau \in \mathcal{T}(x)} \sum_{l \in Y(x)} \frac{|\{k \in Y(x) : \ rank_\tau(x; k) \leq rank_\tau(x; l)\}|}{rank_\tau(x; l)};$$

$$P_{best}(x) = \max_{\tau \in \mathcal{T}(x)} \sum_{l \in Y(x)} \frac{|\{k \in Y(x) : \ rank_\tau(x; k) \leq rank_\tau(x; l)\}|}{rank_\tau(x; l)}.$$

4 A solution to the optimization problem (3.5),(3.6)

The function in (3.4) is non-convex and non-linear, and therefore may have many local minimum points. We need to find the global optimum point. The number of variables is $|\mathcal{D}| \cdot c$. For the data **Mallreac**, that we will consider, $|\mathcal{D}| = 5057$ and $c = 18$. Thus we have a global optimization problem with 91026 variables, which is very hard to handle using existing global optimization methods. Note that, we also tried to use local minimization methods which were unsuccessful. This means that there is a clear need to develop new optimization algorithms for solving problem (3.5),(3.6), taking into account some peculiarities of the problem.

In this paper we suggest one heuristic method for finding a "good" solution to the problem (3.5),(3.6). This method is based on the proposition given below.

We denote by S the unit simplex in R^c; that is,

$$S = \{h = (h_1, \dots, h_c) : \quad h_i \geq 0,\ h_1 + \dots h_c = 1\}.$$

In this case for each $h(d) \in S$ the component $h_i(d)$ indicates simply the probability of the occurrence of the reaction i.

Given drug d we denote by $X(d)$ the set of all records in $\mathcal{X}$, which used just one drug – d. Simply, the set $X(d)$ combines all records where the drug d was used alone.

Consider the problem:

$$\sum_{x \in X(d)} \sum_{j=1}^{c} \|\mathcal{Y}(x)\|^{-p} \cdot (\mathcal{Y}_j(x) - \|\mathcal{Y}(x)\|\, h_j(d))^2 \ \rightarrow \ \min, \tag{4.7}$$

$$h(d) = (h_1(d), \dots, h_c(d)) \in S. \tag{4.8}$$

Proposition 4.1 *A point* $h^*(d) = (h_1^*(d), \dots, h_c^*(d))$, *where*

$$h_j^*(d) = \left(\sum_{x \in X(d)} \|\mathcal{Y}(x)\|^{2-p} \right)^{-1} \cdot \sum_{x \in X(d)} \|\mathcal{Y}(x)\|^{1-p}\, \mathcal{Y}_j(x), \quad j = 1, \dots, c, \tag{4.9}$$

is the global minimum point for the problem (4.7),(4.8).

Now, given drug d, we consider the set $X_{all}(d)$ which combines all records that used the drug d. Clearly $X(d) \subset X_{all}(d)$. The involvement of other drugs makes it impossible to solve the corresponding optimization problem similar to (4.7), (4.8). In this case, we will use the following heuristic approach to find a "good" solution.

We denote by $N_{drug}(x)$ the number of drugs taken by the patient x. Then, we set:

$$h_j^{**}(d) = \left(\sum_{x \in X_{all}(d)} \|\mathcal{Y}(x)\|^{2-p} \right)^{-1} \cdot \sum_{x \in X_{all}(d)} \|\mathcal{Y}(x)\|^{1-p}\, \frac{\mathcal{Y}_j(x)}{N_{drug}(x)}, \quad j = 1, \dots, c. \tag{4.10}$$

This formula has the following meaning. If $N_{drug}(x) = 1$ for all $x \in X_{all}(d)$, then (4.10) provides global minimum solution. Let $N_{drug}(x) > 1$ for some record $x \in X_{all}(d)$. In this case, we assume that all drugs are responsible to the same degree; so we associate only the part $1/N_{drug}(x)$ of the reactions $\mathcal{Y}_j(x)$ to this drug.

4.1 The calculation of weights for each drug

For each drug d we define the sets $X(d)$ – the set of all cases where drug d was used alone and $X_{all}(d)$ – the set of all cases where drug d was used. The set $X(d)$ carries very important information, because here the drug d and reactions are observed in a pure relationship. Therefore, if the set $X(d)$ contains a "sufficiently large" number of records, then it will be reasonable to define the weights $h_j(d)$, $(j = 1, \dots, c)$ by this set.

We consider two numbers: $|X(d)|$ – the number of cases where the drug is used alone, and $P(d) = 100|X(d)|/|X_{all}(d)|$ – the percentage of these cases. To determine whether the set $X(d)$ contains enough records we need to use the both numbers. We will consider a function $\phi(d) = a|X(d)| + bP(d)$ to describe how large the set $X(d)$ is.

Therefore, we define $h(d) = (h_1(d), \dots h_c(d))$ as follows:

$$h(d) = \begin{cases} h^*(d) & \text{if } \ \phi(d) \geq \phi^*; \\ h^{**}(d) & \text{otherwise;} \end{cases} \tag{4.11}$$

where $h^*(d)$ and $h^{**}(d)$ are defined by (4.9) and (4.10), respectively.

Remark 4.1 *We note that the weight $h_i(d)$ is not exactly a probability of the occurrence of the reaction i; that is, the sum $\sum_{i=1}^{c} h_i(d)$ does not need to be equal to 1.*

Remark 4.2 *We have the situation where, for some new (test) examples, new drugs were involved. For each such new drug d we set $h_i(d) = 0, i = 1, \dots, c$.*

5 The algorithms

For our analysis we use two algorithms. The first algorithm $A(p)$ which is introduced in this paper is described below. The second algorithm that we use is BoosTexter (version AdaBoost.MH with real-valued predictions, [27]) which has a high performance in text categorization problems and seems to be suitable for representing drug-reaction associations.

These two algorithms produce the weighted vector $H(x)$ for each patient x which makes it suitable for the applying distance evaluation measures. But the methods of calculating the vectors $H(x)$ are quite different: the algorithm $A(p)$ uses only drugs that have been taken by the patient x, in contrast, BoosTexter uses all drugs in the list of attributes defined (that is, even drugs that have not been taken by patient x, are used for the calculation of the vector $H(x)$). Our hope is that, we can make more accurate conclusions by applying both quite different methods.

The algorithm $A(p)$ determines a classifier (h, H), where the weights $h(d)$, $d \in \mathcal{D}$, are defined by (4.11), and the function $H(x)$ is defined by (2.1). We will consider three versions: $A(0)$, $A(1)$, $A(2)$, corresponding to the distance functions $dist_p$, $p = 0, 1, 2$, respectively. Each of these algorithms tends to minimize the average distance calculated by its own measure.

The second algorithm BoosTexter [27] produces predictions in the form $\mathcal{H}(x) = (\mathcal{H}_1(x), \ldots, \mathcal{H}_c(x))$, where the numbers $\mathcal{H}_i(x)$ are real values which can be positive or negative. In other words, this algorithm defines potential reactions that we are interested in.

To apply the distance measures described above, we need to make all weights calculated by BoosTexter non-negative. Let $\mathcal{H}_{min}(x) = \min_{i=1,\ldots,c} \mathcal{H}_i(x)$. Then we set $H(x) = \mathcal{H}(x)$, if $\mathcal{H}_{min}(x) \geq 0$; and

$$H(x) = (\mathcal{H}_1(x) - \mathcal{H}_{min}(x), \ldots, \mathcal{H}_c(x) - \mathcal{H}_{min}(x), \quad \text{if} \quad \mathcal{H}_{min}(x) < 0.$$

Therefore, we will apply two quite different algorithms – $A(p)$ and BoosTexter. It would be interesting to compare the drug-reaction relations (that is, $h(d)$), produced by these algorithms. For this aim we consider one example.

Example 5.1 Assume that, the drug d, which was used alone in 4 cases: in one case the first and second reactions and in the all other cases just the first reaction have been observed. We have the following representations $h(d)$. The results for BoosTexter are normalized.

$A(0)$**:**	$h(d) = (0.714, 0.286, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$;
$A(1)$**:**	$h(d) = (0.800, 0.200, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$;
$A(2)$**:**	$h(d) = (0.875, 0.125, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$;
BoosTexter (round=500):	$h(d) = (0.673, 0.327, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$;
BoosTexter (round=2000):	$h(d) = (0.675, 0.325, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

In this example the first reaction occurred 4 times and the second once. We see that, BoosTexter produces weights such that the weight for the first reaction is just 2 times greater than for the second reaction, whereas, this difference is greatest for algorithm $A(2)$. The interesting question is: which drug-reaction representation is "better" – one which offers a big difference in weights or one which offers a slight difference in weights in spite of big differences in distribution.

For this purpose we use different evaluation measures. Note that, in terms of *One-error*, *Coverage* and *Average Precision* all the representations in Example 5.1 work similarly; that is, the rank for the first reaction is 1 and for the second is 2. The effect of different drug-reaction representations can be observed considering the cases where more than one drug was used.

New events

One of the main difficulties that arises in the study of drug-reaction relationships is the low level of repeating cases (events) in spite of the large number of records. We consider eighteen groups of reactions (out of 1224 reactions) in order to have a sufficient number of repeating cases. However, in the dataset **Mallreac** this problem still exists (for the drugs). One way to avoid this problem is to combine similar drugs as one meta-drug (as it has been done for reactions), but to achieve this requires a more complete classification of drugs by ADRAC, which has partially implemented the Nordic anatomical therapeutic chemical (ATC) classification [23].

We define *a new event* as a case where, for some test example, all drugs used are not presented in the training set; in other words, all drugs have been taken by this patient are "new". This situation relates to the fact that, new drugs are constantly appearing on the market. Obviously, to make prediction for such examples does not make sense. Therefore, in the calculations below, we will remove all new events from test sets.

6 The results of numerical experiments

In the calculations below we take as a test set records sequentially from each year, starting from 1996 until 2001. For example, if records from 1999 are taken as a test set, then all records from years 1972–1998 form a training set. In the Table 2 we summarized the number of records in test and training sets, and, also, the number of new events removed.

Table 2: The Training and Test Sets **"Removed" means how many records were removed from test set.*

Year	Number of Records		
	Training	Test	Removed*
1996	79660	7734	410
1997	87804	8090	715
1998	96609	9361	774
1999	106744	11216	840
2000	118880	11244	671
2001	130795	6186	191

First we made calculations by the algorithm BoosTexter. We ran this algorithm choosing different numbers of training rounds. The results obtained for 6000 rounds are presented in Tables 3 and 4. In Table 4 we present the results obtained for *One-error*, *Coverage* and *Average Precision.*

Next we applied the algorithms $A(0)$, $A(1)$ and $A(2)$, corresponding to the distance functions $dist_p$, $p = 0, 1, 2$, respectively. We used a function $\phi(d) = |X(d)| + P(d)$ to describe the informativeness of the set $X(d)$. We also need to set a number ϕ^*. The calculations show that the results are not essentially changed for different values of ϕ^* in the region $\phi^* \geq 50$. We set a large number $\phi^* = 1000$ in the calculations which means that for the majority of drugs weights are caculated by formula (4.10). The results are presented in Tables 5 and 6.

In Table 3 we present the results for the average distance errors comparing algorithm $A(p)$ to BoosTexter for average error. On the basis of this measure algorithm $A(p)$ performs consitently better than BoosTexter. This means that the formulae (4.11) for definition of weights $h(d)$ provides a "good" approximation for the minimal solution to the Problem (3.5), (3.6).

The results for the other measures (one-error, coverage and average precision) for BoosTexter are given in Table 4, and for $A(p)$ in Tables 5 and 6. It can be seen that, for all drugs, comparing Tables 4 and 5, the performance of $A(p)$ and BoosTexter are very similar. When the suspected drugs are compared (Tables 4 and 6), algorithm $A(1)$ performs better than BoosTexter. The comparison of results obtained for other evaluation measures (Tables 5 and 6) reveals differences in the performance of these three versions. We observe that by *One-error* measure $A(1)$ performs better than the others, by *Coverage* $A(2)$ performs better and by *Average Precision* both $A(1)$ and $A(2)$ perform better. In all cases $A(0)$ has a worse performance. Therefore, we can conclude that, the definition of distance measure, taking p =1 or 2, is preferable than p=0.

Table 3: The Results Obtained by A(0), A(1), A(2) and BoosTexter for Average Error for Distance Measure $p = 1$ *The algorithm BoosTexter2_1 [27] was set to run for 6000 training rounds. 'all drugs" means that the drug-reaction weights are calculated assuming all drug(s), suspected "sus. drugs" means that we use only suspected drug(s) reported in ADRAC data*

Year	Drugs	A(0)		A(1)		A(2)		BoosTexter	
		Training	Test	Training	Test	Training	Test	Training	Test
1996	all drugs	0.695	0.740	0.678	0.731	0.693	0.751	0.817	0.820
	sus. drugs	0.679	0.740	0.661	0.732	0.681	0.759	0.816	0.820
1997	all drugs	0.696	0.740	0.679	0.734	0.694	0.755	0.817	0.824
	sus. drugs	0.680	0.743	0.663	0.737	0.683	0.762	0.817	0.825
1998	all drugs	0.697	0.737	0.680	0.731	0.694	0.751	0.818	0.818
	sus. drugs	0.681	0.748	0.664	0.742	0.684	0.767	0.817	0.818
1999	all drugs	0.697	0.726	0.680	0.722	0.694	0.751	0.817	0.820
	sus. drugs	0.681	0.722	0.665	0.718	0.683	0.749	0.816	0.820
2000	all drugs	0.696	0.756	0.680	0.752	0.693	0.783	0.817	0.815
	sus. drugs	0.681	0.755	0.665	0.753	0.683	0.787	0.817	0.815
2001	all drugs	0.697	0.749	0.682	0.747	0.695	0.781	0.817	0.809
	sus. drugs	0.684	0.749	0.668	0.748	0.687	0.787	0.817	0.809

As we have shown in Example 5.1, the algorithm $A(2)$, in comparison with $A(1)$, generates drug-reaction representations with greater differences in weights. The results presented in Table 5, for *One-error* and *Coverage*, obtained by the algorithms $A(1)$ and $A(2)$, allow us to make very interesting conclusions:

– if we are interested in *One-error* (that is, simply saying, the probability of occurrence of the reaction which has the greatest weight) then it is better to use drug-reaction representations with small differences in weights;

– if we are interested in *Coverage* then it is better to use drug-reaction representations with big differences in weights. We note that, in all cases, the results are improved if only suspected drugs are used.

Below we will only concentrate on the results obtained by $A(1)$ and BoosTexter.

The comparison of the training errors in Tables 3, 4, 5 and 6 shows that $A(1)$ performs generally better than BoosTexter. We can decrease training errors for BoosTexter by increasing the number of training rounds (results not presented), but in this case test errors increase. This should be expected, because the algorithm BoosTexter is not designed to minimize some distance measures, instead it tends to achieve good performance for the other evaluation measures.

There are two important points that make using the algorithm $A(1)$ preferable for the study drug-reaction associations.

1. BoosTexter does not calculate weights for each drug. For example, until the year 2000, where 5057 drugs were used. BoosTexter running for 6000 rounds defines weights only for 4521

Table 4: The Results Obtained by BoosTexter for One-Error, Coverage and Average Precision *The algorithm BoosTexter2_1 [27] was set to run for 6000 training rounds* *Average Precision is presented in percent

Year	Drugs	$E_{one-error}$		E_{cov}		P_{av}*	
		Training	Test	Training	Test	Training	Test
1996	all drugs	0.463	0.533	1.580	2.134	63.93	57.06
	sus. drugs	0.485	0.533	1.686	2.208	62.50	55.87
1997	all drugs	0.466	0.541	1.601	2.185	63.67	56.76
	sus. drugs	0.486	0.560	1.708	2.243	62.27	55.64
1998	all drugs	0.469	0.539	1.630	2.147	63.29	56.84
	sus. drugs	0.489	0.558	1.730	2.197	62.00	56.25
1999	all drugs	0.471	0.507	1.651	2.038	63.07	59.26
	sus. drugs	0.490	0.503	1.747	2.015	61.86	59.72
2000	all drugs	0.471	0.553	1.665	2.059	62.98	56.94
	sus. drugs	0.488	0.549	1.747	2.036	61.95	57.35
2001	all drugs	0.475	0.557	1.683	2.112	62.69	55.70
	sus. drugs	0.490	0.565	1.754	2.117	61.79	55.37

drugs to this year. In contrast, $A(1)$ calculates weights for each drug encountered, which is very important (in this case we establish drug-reaction relations for all drugs).

2. The algorithm BoosTexter classifies examples so that drugs that are not used are still assigned weights to the function $H(x)$. In the other words, reactions are predicted not only by drugs actually used, but also, drugs which were not taken.

In spite of these two points, the application of the algorithm BoosTexter is useful, because this algorithm, based on quite different method, provides us very important information about the possible accuracy of the prediction that could be achieved.

One more important fact should also be noted. In all cases above the results obtained are much better than the default values (we define default values assuming that for each record all reactions can occur with the same weight). This emphasizes that it possible to study drug-reaction relations, not involving other information about patients. The drug-reaction relationships could then be used, together with the patient information, to enhance the prediction of reactions that may occur.

Table 5: The Results Obtained by $A(0)$, $A(1)$ and $A(2)$ for Different Evaluation Measures – using all drugs **Average Precision is presented in percent*

Year	Algorithms	$E_{one-error}$		E_{cov}		P_{av}*	
		Train	Test	Train	Test	Train	Test
1996	$A(0)$	0.494	0.538	1.930	2.304	60.03	55.70
	$A(1)$	0.471	0.526	1.655	2.112	63.10	57.72
	$A(2)$	0.497	0.538	1.512	2.013	63.46	58.30
1997	$A(0)$	0.496	0.542	1.949	2.355	59.73	55.41
	$A(1)$	0.473	0.534	1.676	2.193	62.88	57.22
	$A(2)$	0.498	0.551	1.532	2.102	63.26	57.24
1998	$A(0)$	0.497	0.536	1.971	2.288	59.54	56.45
	$A(1)$	0.475	0.525	1.701	2.146	62.60	57.81
	$A(2)$	0.501	0.562	1.556	2.068	62.99	56.96
1999	$A(0)$	0.497	0.515	1.980	2.185	59.52	58.42
	$A(1)$	0.474	0.504	1.720	2.043	62.49	59.81
	$A(2)$	0.501	0.532	1.578	1.979	62.84	59.28
2000	$A(0)$	0.498	0.568	1.982	2.365	59.45	53.74
	$A(1)$	0.474	0.559	1.724	2.180	62.47	55.49
	$A(2)$	0.499	0.575	1.584	2.061	62.87	56.67
2001	$A(0)$	0.502	0.563	1.998	2.288	59.15	54.04
	$A(1)$	0.478	0.550	1.735	2.125	62.22	56.11
	$A(2)$	0.502	0.571	1.595	2.043	62.64	56.68

7 Conclusion

In this paper we have studied drug-reaction relations using the system organ class grouping of reactions from ADRAC data. These relations are presented in the form of a vector of weights. To determine these vectors we applied two algorithms based on quite different methods. The results show the possibility of studying drug-reaction relations, not involving other information about patients. In all cases above the results obtained are much better than the default values. For instance, the results for test sets (that is, for new patients) in terms of *One-error* were around 0.270 which should be considered sufficiently low error rate. This error rate means that for 73 percent of new patients, the reaction having the greatest weight has occurred.

To develop new algorithms taking into account the peculiarities of ADRs is an important problem. The development of these algorithms should help us to extract more useful information from ADRAC data. In particular, the study of drug-reaction associations, drug-drug interactions and the influence of other data fields contained in the ADRAC data are interesting problems for future investigation.

Table 6: The Results Obtained by $A(0)$, $A(1)$ and $A(2)$ for Different Evaluation Measures – using only suspected drugs **Average Precision is presented in percent*

Year	Algorithms	$E_{one-error}$		E_{cov}		P_{av}*	
		Train	Test	Train	Test	Train	Test
1996	$A(0)$	0.472	0.520	1.806	2.236	61.98	57.28
	$A(1)$	0.453	0.510	1.547	2.057	64.76	59.14
	$A(2)$	0.469	0.518	1.410	1.984	65.13	59.38
1997	$A(0)$	0.474	0.531	1.827	2.303	61.70	56.75
	$A(1)$	0.454	0.519	1.566	2.156	64.56	58.29
	$A(2)$	0.470	0.532	1.429	2.071	64.93	58.38
1998	$A(0)$	0.474	0.520	1.840	2.238	61.61	57.80
	$A(1)$	0.456	0.516	1.586	2.106	64.33	58.86
	$A(2)$	0.471	0.549	1.451	2.038	64.71	57.96
1999	$A(0)$	0.473	0.503	1.847	2.141	61.61	59.37
	$A(1)$	0.456	0.492	1.599	2.010	64.26	60.66
	$A(2)$	0.472	0.518	1.469	1.955	64.57	60.08
2000	$A(0)$	0.476	0.566	1.855	2.343	61.43	54.15
	$A(1)$	0.456	0.558	1.611	2.135	64.16	56.38
	$A(2)$	0.473	0.564	1.482	2.041	64.48	57.24
2001	$A(0)$	0.481	0.554	1.878	2.271	60.99	54.43
	$A(1)$	0.462	0.542	1.631	2.090	63.80	56.91
	$A(2)$	0.478	0.562	1.502	2.022	64.11	57.36

8 Acknowledgements

The work reported in this paper has been supported by The Australian Research Council, the TGA (Therapeutic Goods Administration), the Ballarat Division of General Practice, the Medical Software Industry Association, the Therapeutic Guidelines Pty. Ltd. We are indebted to Patrick Purcell from the TGA for providing expert comments. We also thank Rob Schapire for his helpful advice in using BoosTexter.

References

[1] Aronson J.K., Derry S., Loke Y.K. Adverse drug reactions: keeping up to date. Fundam Clin Pharmacol 2002;16:49–56.

[2] Bate A, Lindquist M, Edwards IR, Olsson S, Orre O, Lansner A, De Freitas R. A Bayesian neural network method for adverse drug reaction signal generation. Eur J Clin Pharmacol 1998;54(4):315–321.

[3] Bates D.W., Evans R.S., Murff H., Stetson P.D., Pizziferri L., Hripcsak G. Detecting Adverse Events Using Information Technology. J Am Med Inform Assoc 2003;10:115–128.

[4] Bates D.W., Gawande A.A. Improving Safety with Information Technology. N Engl J Med 2003;348:2526–34.

[5] Brown Jr Stephen D., Landry Frank J. Recognizing, Reporting, and Reducing Adverse Drug Reactions. South Med J 2001;94(4):454–462.

[6] Chuang-Stein C. Statistics for Safety Data. In: Stephens M.D.B., Talbot J.C.C., Routledge P.A., editors. Detection of New Adverse Drug Reactions. 4th ed. London: Macmillan Reference Ltd 1998. p. 271–79.

[7] Consumer Reports on Medicines. Consensus Document adopted at the First Conference on CRM. 2000 Sep 29–Oct 1; Sigtuna, Sweden. Available at: http://www.kilen.org/kilen/htm/crm.htm

[8] Coulter David M., Bate Andrew, Meyboom Ronald H. B., Lindquist Marie, Edwards I. Ralph. Antipsychotic drugs and heart muscle disorder in international pharmacovigilance: data mining. BMJ 2001;322(7296):1207–1209.

[9] Dormann H., Criegee-Rieck M., Neubert A., Egger T., Geise A., Krebs S., et al. Lack of Awareness of Community-Aquired Adverse Drug Reactions Upon Hospital Admission – Dimensions and Consequences of a Dilemma. Drug Saf 2003;26(5):353–362.

[10] DuMouchel W. Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system. Am Statistician 1999;53(3):177–190.

[11] Egberts A.C.G., Meyboom R.H.B., von Puijenbrock E.P. Use of Measures of Disproportionality in Pharmacovigilance. Drug Saf 2002;25(6):453–458.

[12] Evans S.J.W., Waller P.C., Davis S. Use of proportional reporting ratios (PRRs) for signal generation from spontaneous adverse drug reaction reports. Pharmacoepidemiol Drug Saf 2001;10:1–4.

[13] Fernanopulle R.B.M., Weerasuriya K. What Can Consumer Adverse Drug Reaction Reporting Add to Health Professional-Based Systems? Drug Saf 2003;26(4):219–225.

[14] Harvey J.T., Turville C., Barty S.M. Bayesian Data Mining of the Australian Adverse Drug Reactions Database. International Trans Operational Research 2002. *In press*

[15] Hartmann K., Doser A.K., Kuhn M. Postmarketing Safety Information: How Useful are Spontaneous Reports? Pharmacoepidemiol Drug Saf 1999;8:S65–S71.

[16] Heely Emma, Riley Jane, Layton Deborah, Wilton Lynda V., Shakir, Saad A. W. Prescription-event monitoring and reporting of adverse drug reactions. Lancet 2001;358(9296):182–184.

[17] Hillson Eric M., Reeves Jaxk H., McMillan Charlotte A. A statistical signalling model for use in surveillance of adverse drug reaction data. J Appl Statistics 1998;25(1):23–41.

[18] Hutchinson Tom A. Bayesian assessment of adverse drug reactions. Can Med Assoc J 2000;163(11):1463–1466.

[19] Mamedov M.A., Saunders G.W. An Analysis of Adverse Drug Reactions from the ADRAC Database. Part1: Cardiovascular group. University of Ballarat School of Information Technology and Mathematical Sciences, Research Report 02/01, Ballarat, Australia, February 2002;1–48. Available at: http://www.ballarat.edu.au/itms/research-papers/paper2002.shtml

[20] Mamedov M.A., Saunders G.W. A Fuzzy Derivative Approach to Classification of outcomes from the ADRAC database. International Trans Operational Research 2002. *In press*

[21] Mamedov M.A., Saunders G.W. Analysis of Cardiovascular Adverse Drug Reactions from the ADRAC Database. Proceedings of the APAC Conference and Exibition on Advanced Computing, Grid Applications and eResearch 2003. Royal Pines Resort, Gold Coast, Queensland, 29 September – 2 October 2003 http://www.apac.edu.au/apac03/

[22] Orsini M., Funk P.A. An ADR Surveillance Program: Increasing Quality, Number of Incidence Reports. Formulary 1995;30(8):454–461.

[23] Pinkston V., Swain E.J. Management of Adverse Drug Reaction and Adverse Event Data through Collection, Storage and Retrieval. In: Stephens MDB, Talbot JCC, Routledge PA, editors. Detection of New Adverse Drug Reactions. 4th ed. London: Macmillan Reference Ltd 1998. p. 281–96.

[24] Pirmohamed Munir, Breckenridge Alasdair M., Kitteringham Neil R., Park B. Kevin. Adverse drug reactions. BMJ 1998;316(7140):1294–1299.

[25] Purcell P., Barty S. Statistical Techniques for Signal Generation – The Australian Experience. Drug Saf 2002;25(6):415–421.

[26] Redfern W.S., Wakefield I.D., Prior H., Pollard C.E., Hammond T.G., Valentin J-P. Safety pharmacology – a progressive approach. Fundam Clin Pharmacol 2002;16:161–173.

[27] Schapire Robert E., Singer Yoran. Boostexter: A boosting-based system for text categorization. Machine Learning 2000;39: 135–168.
BoosTexter software available at:
http://www.cs.princeton.edu/ ∼ schapire/boostexter.html

[28] Szarfmann Ana, Machado Stella G., O'Neill Robert T. Use of Screening Algorithms and Computer Systems to Efficiently Signal Higher-Than-Expected Combinations of Drugs and Events in the US FDA's Spontaneous Reports Database. Drug Saf 2002;25(6):381–392.

[29] Troutman W.G., Doherty K.M. Comparison of voluntary adverse drug reaction reports and corresponding medical records. Am. J Health-Syst Pharm 2003;60(6):572–575.

[30] Tubert-Bitter P., Begaud B. Comparing Safety of Drugs. Post Marketing Surveillance 1993;7:119–137.

[31] van Groothees K., de Graaf L., de Jong-van den Berg T.W. Consumer Adverse Drug Reaction Reporting – A New Step in Pharmacovigilance? Drug Saf 2003;26(4):211–217.

[32] van Puijenbroek Eugne P., Bate Andrew, Leufkens Hubert G. M., Lindquist Marie, Orre Roland, Egberts Antoine C. G. A comparison of measures of disproportionality for signal detection in spontaneous reporting systems for adverse drug reactions. Pharmacoepidemiol Drug Saf 2002;11(1):3–10.

[33] van Puijenbroek Eugne P., Diemount Willem L., van Grootheest Kees. Application of Quanitative Signal Detection in the Dutch Spontaneous Reporting System for Adverse Drug Reactions. Drug Saf 2003;26(5):293–301.

[34] World Health Organization. WHO Technical Report No 498, and Note for Guidance on Clinical Safety Data Management: Definitions and Standards for Expedited Reporting (CPMP/ICH/377/95) 1972.

[35] Yang Yimming, Liu Xin. A re-examination of text categorization methods. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval 1999;39:42–49.

[36] Zapater Pedro, Such Jos, Prez-Mateo Miguel, Horga Jos Francisco. A New Poisson and Bayesian-Based Method to Assign Risk and Causuality in Patiens with Suspected Hepatic Adverse Drug Reactions. Drug Saf 2002;25(10):735–750.

Mining Geographical Data with Sparse Grids

Markus Hegland[1], and Shawn W. Laffan[2,1]

[1] Mathematical Sciences Institute, Australian National University, Canberra, Australia, 0200.
Markus.Hegland@anu.edu.au
[2] School of Biological, Earth and Environmental Sciences, University of New South Wales, Sydney, Australia, 2052.
Shawn.Laffan@unsw.edu.au

Abstract. Predictive modelling tools combined with spatial data mining techniques can be used to discover relationships among geographic phenomena. Mining geographical data requires very flexible and simple function classes for data fitting, highly efficient methods to be able to fit a large number of models locally to vast amounts of data. The models need to have structure which allows collections of models to be mined. It appears that sparse grids do satisfy these requirements of geographical data mining.
After a discussion of requirements of geographical data, mining sparse grids will be reviewed and their suitability for mining geographical data will be discussed. Some results are presented and some further ideas for mining of large collections of local sparse grid models are discussed.

Keywords: Spatial analysis, geographic data mining, predictive modelling, sparse grids

1 What do geographers want from data mining?

One can describe geography as being the study of where things are, where things are not, and how these things interact with one another through space and time. "Things" in this sense are any phenomena that have some spatial component, and might be discrete objects or continuous surfaces (fields) [25]. Examples include individual trees and animals, or populations of such trees and animals. Alternately, one might investigate spatial processes such as the flow of water through a catchment and the associated flux of sediment, nutrients and pollutants [30]. From this, one can see that geographers are interested in a wide variety of problems from many different fields of research.

The sparse grids technique one means of analysing geographic phenomena, and we describe its potential for geographic data mining in this paper.

1.1 The problems of geographic data

There is now an enormous amount of information collected that has spatial attributes, including from remote sensing satellites, national censuses, mobile telephone records, crime statistics, land use change in agriculture, health records, epidemiological data, and consumer purchasing records. The challenge for geographers is to make sense of these data to derive an understanding of the underlying spatial phenomena. The problem we face is that there is an overwhelming volume of data.

As an example of the volume of data, the 2001 census of the Australian population has over 17,000 sample areas for the state of New South Wales at the finest aggregate unit (census collection district). For each of these collection districts there are thirty-three categories containing over 4000 attribute fields into which the population is divided. Each person in the collection district at the time of the census will fall into at least one field within each category, and sometimes more. When one extends the analysis to include temporal relationships among the population over several censuses then the amount of data increases considerably. This is despite each earlier census being of slightly lower detail.

As another example, consider the amount of data generated by remote sensing satellites. A commonly used data source is the Landsat Enhanced Thematic Mapper, which collects data in six visible and infra-red bands (30 m resolution), one thermal band (with high and low spatial resolution versions (120 m and 60 m), and one panchromatic band (across visible and near infra-red spectra, 15 m resolution). The data are divided into scenes for distribution, with each scene approximately 185 km along each side. This translates to approximately 6000 by 6000 cells for the visible and infra-red bands. This is normally a reasonable data set to deal with. However, when one includes the temporal component then, once again, the amount of data increases rapidly. The Landsat satellite repeats its polar orbit cycle every sixteen days, so a location at the equator is sampled every sixteen days, and more frequently near the poles due to spatial overlap of the sample swaths. The result is that, if there is no cloud to obscure the ground, one would have twenty-two scenes to analyse for each year of a study. While this is very simple example, as many locations are obscured by cloud for at least some portion of the scenes each year, there are other satellites carrying radar sensors that are not affected by cloud, hyperspectral sensors which record spectral response in tens to hundreds of bands, and sensors with very high spatial resolutions (eg. 0.6 m). This is a major issue given that each of these sensors provides complementary information and so they could all be used in a single analysis.

A second consideration is spatial and temporal non-stationarity, which can have serious effects on analysis results. This is related to Tobler's First Law of Geography "that everything is related to everything else; but that near things are more related than those far apart" [29].

If a model is applied to a data set and is assumed to apply equally over the entire study area, study period, or both, then one is assuming that the processes are spatially and temporally stationary. This means that the landscape studied is in some form of equilibrium between the phenomenon of interest (response variable) and the phenomena it is being related to (correlate variables). This is often not the case [18–20]. Spatial phenomena are often the result of a series of superimposed processes, all operating at different spatial and temporal scales [3].

Some of these processes may be in equilibrium with the phenomenon of interest, but many may not be. The end result is that the response variable will have a good relationship with some of the correlate variables, and little relationship with others. If the analysis is conducted at a geographically local scale (by using a geographically local sample) then many of these problems may be reduced because the effects of non-stationarity will be less over smaller regions.

A third major problem that arises from these geographic data are that they are all highly correlated, both through space and time. Any analysis applied to the data must be able to operate in the presence of such correlation without violating the assumptions of the method.

Finally, there are issues with the treatment of geographic and temporal spaces. Geographic processes are often complex and non-Euclidean [19], meaning that the treatment of the geographic relationships will impact on the validity of the results of the mining exercise. In addition to this, time cannot be treated as merely an extension of the spatial domain [27]. The nature of time is very different in that it is unidirectional in the flow of effects, whereas spatial processes operate in three dimensions.

The above issues are just a sample of those that affect geographic analyses (see [24] and [23]), and there is a clear need for data mining tools that can approach geographical problems involving such complexity and sheer volumes of data. Sparse grids provide one approach that may be able to cope with some of these issues.

2 Predictive modelling with sparse grids

2.1 Sparse grid functions

The sparse grid idea was known for some time, it appears in the modern literature in [28] and sparse grids were applied to the solution of engineering problems by Zenger [31]. The first data mining applications of sparse grids appeared in [16] and [9] and geographical applications were first considered in [21]. Sparse grids can also be viewed as a special variant of multivariate regression splines (MARS) [8]. The theory of sparse grids has been developed considerably, especially for the case of the classical sparse grids, starting with [31] and the “Munich school”.

In the following we provide a short introduction into sparse grid functions and contrast them with regular grid functions.

Sparse grid functions are multivariate real functions $f(x_1, \ldots, x_d)$. They combine many features of linear models, regression trees [2], wavelets [4], MARS [8], finite elements [1], additive models [13], and splines [5].

In the univariate case, $d = 1$ the sparse grids are piecewise linear functions [5], defined on a regular grid, i.e. a grid with equidistant grid points. The number of gridpoints is n_k where $n_0 = 1$ corresponds to constant functions, $n_1 = 1$ to linear functions. For $k > 1$ one has $n_k = 2^{k-1} + 1$. Piecewise linear functions provide a good compromise between computational efficiency and continuity and form the basis for MARS [8, 13, 5]. Adaptivity is obtained through the choice of the level parameter k.

In the case of of bivariate functions, $d = 2$ regular grids define piecewise bilinear functions. In each cell defined by four grid lines the functions are bilinear, i.e., have the form $a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$. Such function spaces are commonly used in finite element analysis, for example, in finite element fitting and for the approximation of smoothing splines [26]. The number of grid points is now the product $n_{k_1}n_{k_2}$ of the numbers of gridpoints in each dimension. The simplest spaces of functions defined in this way contain the constant functions with one grid point, the functions which are linear in one variable with two grid points, and the bilinear functions with 4 grid points.

These are the simplest four spaces defined by two-dimensional regular grids. Note that in particular the space of linear functions, while a subspace of bilinear functions, does not correspond to a regular grid. As the functions are represented by the values at the grid points it is not totally obvious unless one computes a difference of the type $y_1 - y_2 - y_3 + y_4$ of the values of the four grid points if the bilinear function is actually linear or not. This is a slight computational disadvantage for data mining if one would like to generate many models and extract the linear ones.

This situation is even worse in the case of more than 2 variables. The simplest regular grid which involves all d variables has 2^d grid points and defines a multilinear function which is defined by the values in the grid points and thus requires 2^d coefficients for their representation. Linear functions, which can be represented with $d+1$ parameters, are thus not economically described by regular grid functions which use 2^d parameters. Moreover, the determination of the linearity of a multilinear function is much more complicated than in the previous case. Note that the simplest function defined by a regular grid and involving all d variables is multilinear and needs 2^d coefficients. This is an aspect of the curse of dimensionality. Regular grid function approximations are thus not suitable for most data mining applications.

Sparse grid functions address these shortcomings. The grids are unions of regular (and typically small) grids and a sparse grid function is the sum of the corresponding regular grid functions. For example, a sparse grid function $f(x_1, \ldots, x_d)$ could have the form

$$f(x_1, \ldots, x_d) = f_1(x_1) + f_2(x_4) + f_3(x_2, x_4, x_5) + f_4(x_2, x_4, x_5) + \cdots$$

where each component f_i is defined by a regular grid. Note that the same variables, e.g., x_2, x_4, x_5, can occur in different components which relate to different grid sizes, e.g., a grid with $5 \times 5 \times 3$ grid points and a grid with $2 \times 3 \times 9$ grid points. Note that there is some redundancy in this representation, for example one could add a constant to any component and subtract it from any other without changing the function.

Any regular grid is also a sparse grid. For example, the space of constant functions is a sparse grid function space. More generally, if one takes the union of the linear functions in one variable (all of which are regular grids) one obtains the space of linear functions which is thus a sparse grid space. For the representation one requires two coefficients for each component function and so, in the case of d variables, one requires $2d$ coefficients for the representation of linear functions as a sparse grid as the constant part is represented redundantly in each component. While slightly redundant, this compares very favourably with the case where one represents a linear function by a regular grid, i.e., as a multilinear function with 2^d coefficients. With sparse grids one can now directly search for linear functions and in collections of large numbers of models recognise a linear function immediately. More generally, if one considers the union of regular grids with functions which only depend on one variable one obtains additive functions. Like in the case of the linear functions one avoids the curse of dimensionality (the exponential dependence of the number of coefficients) and has the same small redundancy as the constants are represented in all components. If one now considers the union of regular grids where the functions only depend on 2 variables, i.e., where the grids are planar, one obtains a sparse grid which include interactions between variables. Higher order interactions can also be obtained when one uses regular grid functions which depend on more than 2 variables. A simple example of a construction of a sparse grid from a couple of regular grids is given in figure 1.

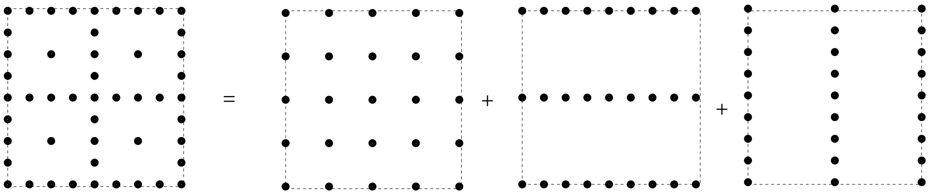


Fig. 1. Construction of a sparse grid from regular components

Sparse grids provide an efficient tool to investigate multiscale effects in addition to the interaction effects discussed previously. In the case of one dimension the width of the grid cells defines the scale. Very fine grids allow the representation of highly oscillating functions and coarser grids are used for more smooth variations. In higher dimensions one may have different scales for the different variables. Moreover, the scales of the interactions, in particular, high-order interactions are typically lower than the ones for additive components as they introduce a higher degree of singularity. For a grid with one variable and n_k grid points one defines the level or scale as k. This can be generalised for two dimensions where the grid with n_k by n_j grid points has now an overall level of $k + j$. The so defined level is approximately the logarithm of the total number of grid points and thus a reasonable complexity measure. It is used to define the "classical" sparse grids [31] which contain components which all have the same level. It was demonstrated that these classical sparse grids have very similar approximation performance as the regular grid which contains the sparse grid.

The classical sparse grids are computationally feasible up to around ten dimensions. This is substantially higher than the regular grids which are predominantly used in two and three dimensions and are maybe feasible for four dimensions. In data mining applications, however, one does encounter regularly between 10 and 100 variables and in many applications in bioinformatics, banking, spectroscopy and others thousands of variables are common. Sparse grids are able to represent functions with close to arbitrary numbers of variables.

2.2 Why sparse grids are suitable for mining geographical data

While sparse grid functions have not been widely used so far in geographical data analysis, it appears that they provide an efficient tool for predictive modelling-based data mining and, in particular, compare favourably with other methods used like artificial neural nets and regression networks.

1. Sparse grids define a collection of function spaces which include the space of linear functions, spaces of additive models, ANOVA decomposition spaces and multilevel decomposition spaces. A function space selection procedure is an important part of the sparse grid mining routines and the type of space fitted provides valuable information about the data.
2. Sparse grids are flexible enough to approximate arbitrary functions. In particular, sparse grid functions provide good approximations of functions from reproducing kernel Hilbert spaces, radial basis functions or regularisation networks. There is some literature on sparse grid approximation properties but no systematic review, see, e.g., `http://bibliothek.iam.uni-bonn.de/duennbib.html` for some earlier references.
3. Sparse grids are additive and sparse grid fitting amounts to solving a penalised least squares problem. The regularisation parameter can be chosen using crossvaliation or test data sets, see [13]
4. There are powerful computational techniques akin to backfitting and using preconditioned conjugate gradient methods which reduce the solution of the sparse grid fitting problem to the iterated solution of multiple regular grid fitting problems for which efficient iterative and direct solvers are available [11, 17].
5. Sparse grid functions can be readily compared in terms of their component spaces and the actual functions and thus lend themselves to mining the collection of models obtained (see last section).
6. Sparse grid functions are similar to ANOVA decompositions and have the same interpretative power in terms of additive effects and interactions. This feature is shared with methods like MARS [8], additive models [14] and ANOVA splines [22].
7. Sparse grids allow the identification of linear trends, curvatures, and higher order fluctuations and permit multiresolution analysis.
8. Sparse grid fitting and evaluation can be done efficiently using parallel and high performance computers and exhibit both coarse and fine grain parallelism. Sparse grid algorithms use finite element technology developed for

engineering problems and tested in many applications and commercial packages. The algorithms scale linearly in the data size and allow scalable parallel execution over the data records. Consequently, the sparse grid algorithms have the capacity required to analyse very large data sets and fit complex models [10, 16, 17].

9. Open source software is available, see `http://datamining.anu.edu.au`.
10. The sparse grid model can be interpreted as a linear model using features which are defined by the regular grid basis functions. Thus any methods based on linear models can be applied. In particular, one may take into account the autocorrelation postulated in Tobler's law using *sparse grid autoregressive models* which are models of the form

$$y(p) = f(x(p), y(p_N), y(p_S), y(p_E), y(p_W))$$

where p denotes the position of the prediction and p_N etc etc are the positions of neighbouring points. After such a model has been learnt the prediction at any new point requires an iterative process which will predict the values in a neighbourhood of points as well. The sparse grids are again just a flexible generalisation of ordinary autoregressive models. Alternatively, sparse grids allow the introduction of models using differential equations of the form

$$y(p) = f(x(p), \varDelta_p y(p))$$

where $\varDelta_p$ is a differential operator in the position.

In summary, sparse grids provide a computationally efficient tool to model functions of very many variables, fitting vast amounts of data, and they can be used to extract and identify additive components, interactions and effects at multiple scales. In geographical applications all these properties are seen to be useful, one often has many variables, and would like to find components, and in particular, compare various models. This is further expanded in the following.

2.3 Fitting sparse grids to data

Predictive modelling addresses the problem of learning from data, i.e. given n data points $x_1, \ldots, x_n$ corresponding to n objects or observations which all have labels or features $y_1, \ldots, y_n$ one would like to find a function f such that $y_i \approx f(x_i)$ so that the function f can be used to predict the labels or features for future observations. Typically this prediction will not be ideal as the model will be limited and the data is usually not sufficient to explain every variation of the labels or features. In order to identify a "best possible" function f one introduces a loss function which depends on both the observed data and the function and f is chosen as the minimum of the loss function. Much has been written about this both in the statistical and machine learning literature and shall not be discussed any further here. Instead the focus of this discussion is on computational issues relating to sparse grid learning. For a good recent reference on the topic of statistical learning in general, see [13].

First consider learning a function from a given sparse grid function space by a least squares fit such that f is determined as minimiser of the functional:

$$J(f) = \sum_{i=1} (y_i - f(x_i))^2.$$

The sparse grid is represented as a sum of regular grid functions f_j such that:

$$f(x) = \sum_{j=1}^{m} f_j(x).$$

If one expands the functions f_j in terms of any basis functions a linear system of equations for the coefficient vectors c_j of the functions f_j is obtained of the form:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

It turns out that this system of equations does pose two computational challenges. First, it can be seen that in most cases the off-diagonal blocks A_{ij} $(i \neq j)$ are fairly dense and only the diagonal blocks are sparse and second the matrix is typically large and singular. The singularity of the matrix can be addressed by regularisation.

The size and density of the matrix is dealt with by a backfitting approach where a sequence of approximations is constructed by

$$f^{(k)} = \sum_{j=1}^{m} f_j^{(k)}$$

and

$$f_j^{(k+1)} = \text{argmin}_{f_j}\, J(\sum_{s=1}^{j-1} f_s^{(k+1)} + f_j + \sum_{s=j+1}^{m} f_s^{(k)}).$$

In this way, at each step a sequence of regular grids have to be determined and thus a collection of small sparse linear systems of equations are solved instead of the one big dense system.

While this algorithm, and, in particular, an acceleration using conjugate gradients, converges reasonably quickly, the small sparse linear systems cannot be solved concurrently. However, there is an alternative algorithm which allows the concurrent determination of components where:

$$f_j^{(k+1)} = \text{argmin}_{f_j}\, J(\sum_{s=1}^{j-1} f_s^{(k)} + f_j + \sum_{s=j+1}^{m} f_s^{(k)}).$$

This approach is basically a block Jacobi method and converges considerably slower than the original backfitting which is a block Gauss-Seidel method.

In [17] a method has been introduced which combines advantages of the additive approach with fast convergence. The idea of this algorithm is based on the *combination technique* [12, 15]. In order to be able to apply the combination technique one requires that

1. The regular spaces which define the sparse grid also contain all the intersections of any two generating grids (which is regular as well).
2. If one first fits any of the component grids to the data, then evaluates on the data and then fits these values to a different component grid one gets exactly the fit onto the intersection of the two components.

In this case it can be seen that there are coefficients γ_j which do not depend on the data such that sparse grid f which fits the data is given by

$$f = \sum_{j=1}^{m} \gamma_j f_j$$

where the f_j are the fits of the components onto the data.

In general, this formula would not hold but one can define an iterative technique where $f_j^{(0)} = 0$ and

$$f_j^{(k+1)} = f_j^{(k)} + \operatorname{argmin}_{g_j} J(f^{(k)} + g_j)$$

where g_j is in the j-th component space, where

$$f^{(k)} = \sum_{j=1}^{m} \gamma_j^{(k)} f_j^{(k)}$$

and where the coefficients $\gamma_j^{(k)}$ are chosen such that for the given $f_j^{(k)}$ the functional $J(f^{(k)})$ is minimised. This algorithm allows the parallel determination of all the components f_j. It can be seen that the algorithm has similar performance to the backfitting approach, see [17].

The most difficult part of learning with sparse grids is the determination of the sparse grid space from the data. Only a heuristic, greedy approach is available here. The algorithm uses the natural ordering of the sparse grids given by the subgrid relation. It starts with a grid with one point and considers all next larger grids, the so-called covering grids. The covering sparse grid with the best performance is selected as the next candidate space in the iteration. This greedy algorithm is similar to the one used in the MARS method [8]. After a relatively fine grid is obtained crossvalidation or test data is used to prune the grid. Note that the procedure used is closely related to subspace or variable selection.

2.4 A sparse grid web service

For ease of use, the sparse grid predictive modelling capacity is delivered by a web server using forms to select the type of analysis and data sets required. A first prototype demonstrator is available, see `http://datamining.anu.edu.au/software/geographic`. As the fitting procedures are computationally intensive, the computations are done on a remote high performance computing centre. The data resides in a data repository which is typically neither collocated with the computing centre nor the web server. This requires a distributed application which has to be able to access remote data, do remote computing and deliver the results over the web. The components of a first demonstrator are displayed in figure 2. An example of a web page delivered by the demonstrator is given in figure 3. At this stage the actual sparse grid space can be chosen by hand. The demonstrator will be further developed, in particular general data sets, grid technology, and further parallel processing will be implemented. The demonstrator requires at this stage that a server process is continuously running on the high performance compute server. This limitation will also be addressed in the future.

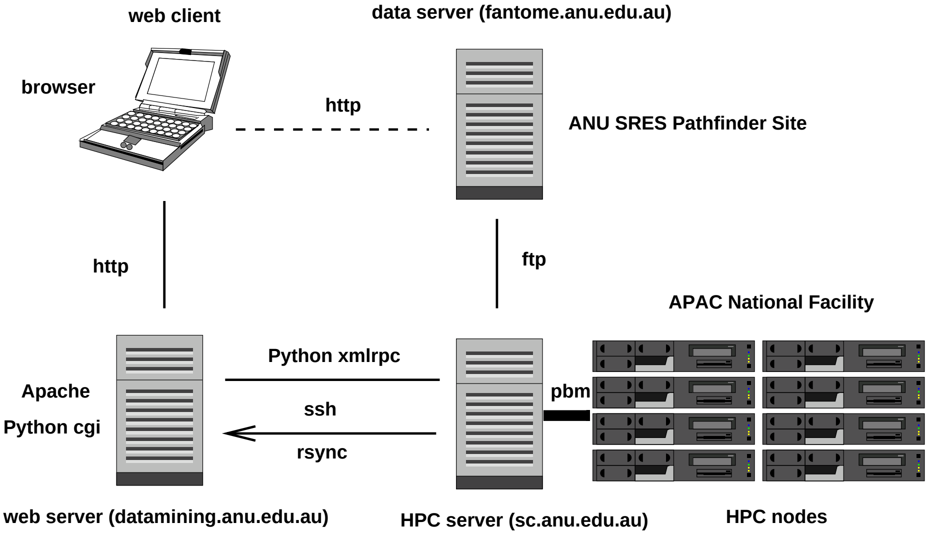


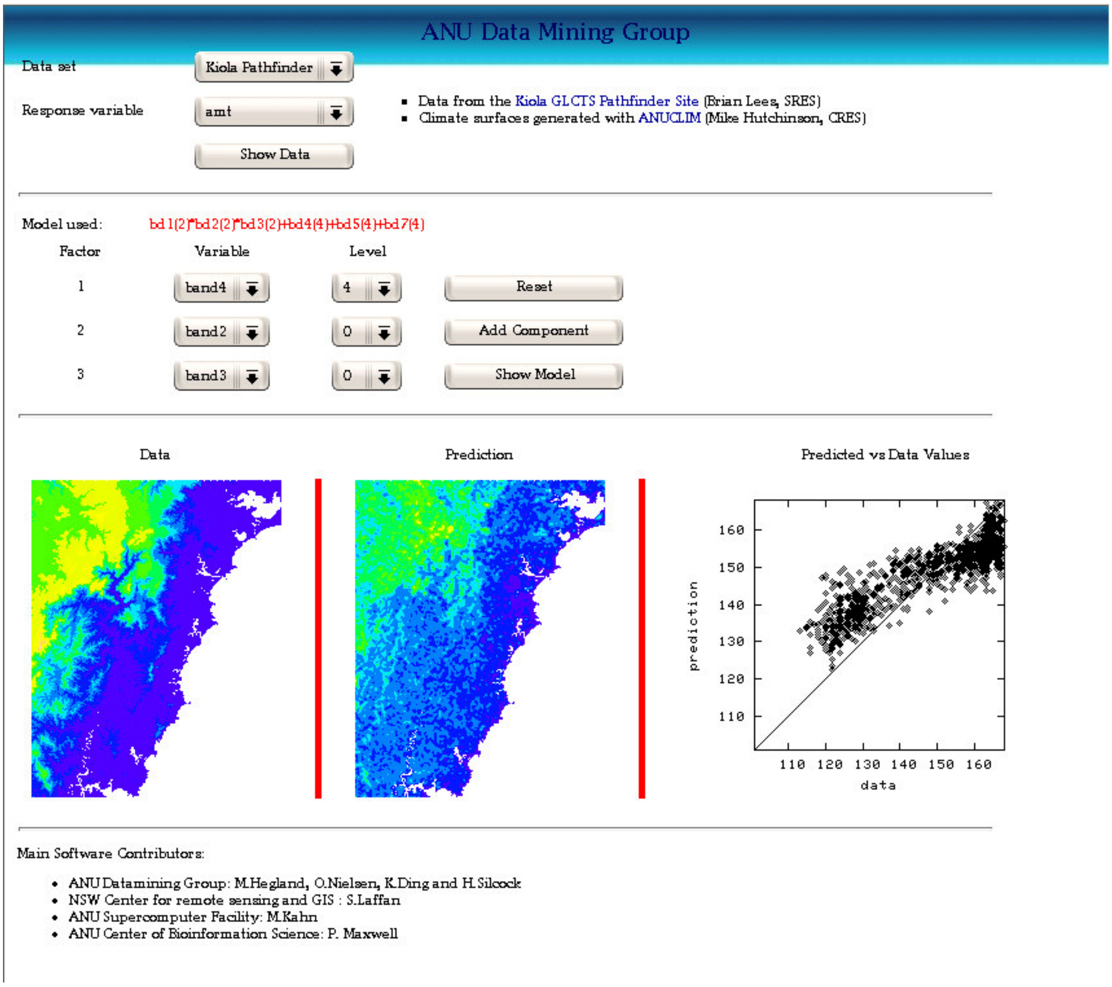
Fig. 2. Components of the Service

3 Sparse grid mining

In this section some first results mining geographical data are presented and then some ideas about further work are developed.

3.1 Sparse grids and geographic processes

A sample of the results from a spatially global sparse grids analysis are given in figure 4. These results were calibrated using the data set of [18–20], which



ANU Data Mining Group

Data set	Kiola Pathfinder
Response variable	amt

Show Data

- Data from the Kiola GLCTS Pathfinder Site (Brian Lees, SRES)
- Climate surfaces generated with ANUCLIM (Mike Hutchinson, CRES)

Model used: bd1(2)*bd2(2)*bd3(2)+bd4(4)+bd5(4)+bd7(4)

Factor	Variable	Level
1	band4	4
2	band2	0
3	band3	0

Main Software Contributors:

- ANU Datamining Group: M.Hegland, O.Nielsen, K.Ding and H.Silcock
- NSW Center for remote sensing and GIS : S.Laffan
- ANU Supercomputer Facility: M.Kahn
- ANU Center of Bioinformation Science: P. Maxwell

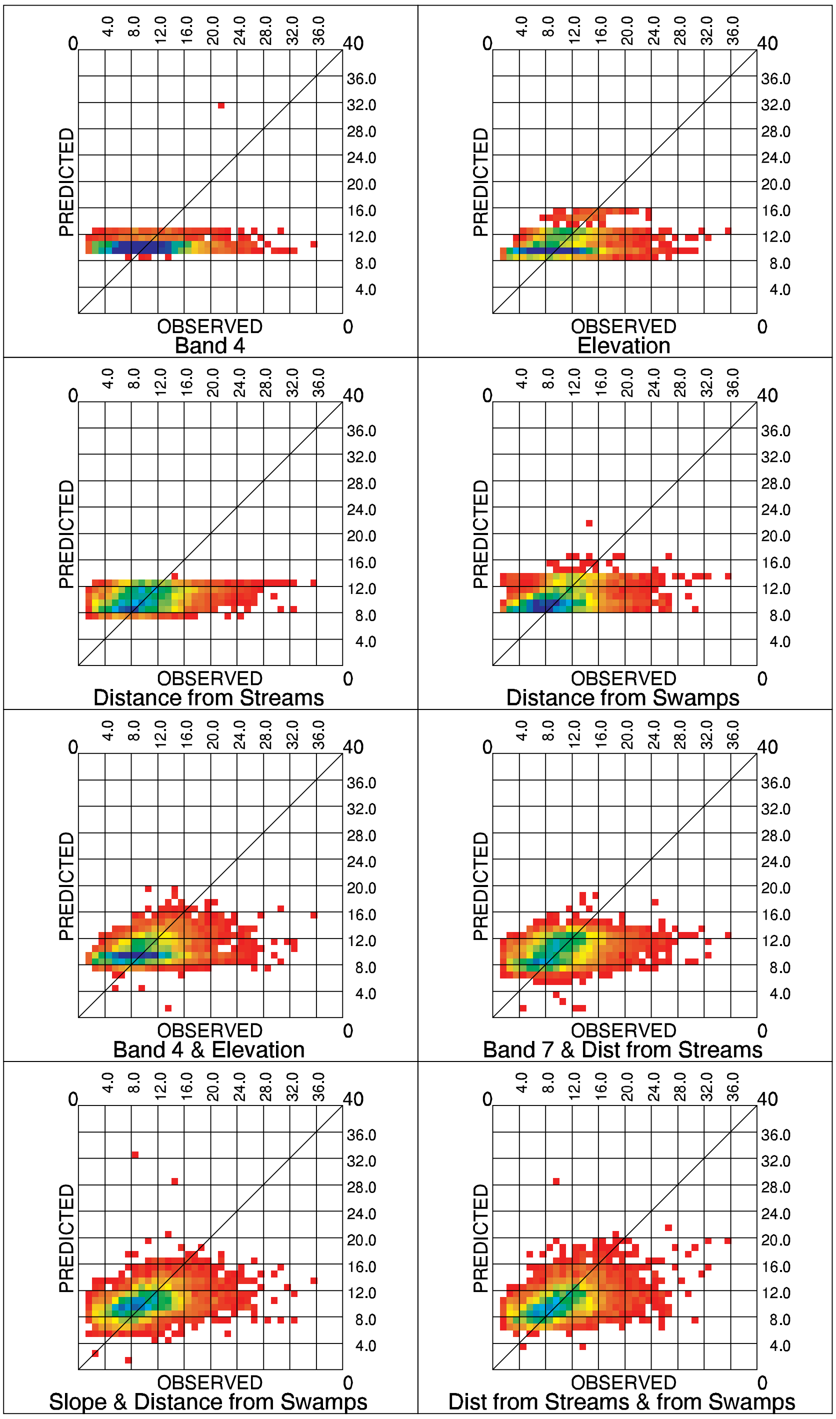
Fig. 3. Web page of the Sparse Grid Web service

contains sample records for regolith properties as well as continuous surfaces of topographic and hydrological indices and Landsat spectral response. They used sparse grids with seventeen grid points in each dimension, with eight variables and 28 first order interaction grids. From these results it is clear that not all of the variables are good predictors of silica abundance, and one can easily determine which is useful.

These results are just a sample of what can be generated, and there are many potential avenues for data mining of sparse grids. We consider here only three of the possible approaches that can be used to gain a greater understanding of geographic patterns and processes.

1. The first consideration is an understanding of where there is a relationship between the predictor variable and the correlates. As noted above, this is likely to be in geographically local areas. The approach one can take to analyse this is to use geographically local sample regions and fit a sparse grids model within them. This is analogous to kernel methods, but the weightings are calculated using only the geographical coordinates. In the case of sparse grids, we have devised such a system of analysing geographically local samples, where a sparse grids model is generated for each of these local samples. The results are a general improvement over the global model, and are better than the results of [19, 20] who used an artificial neural network and moving window regression analyses applied to the same data.
2. However, the above approach can be limiting in that each sample location uses a different model, ignoring those used elsewhere in the study area. Following from this, one might be interested in where similar processes occur across a landscape. For example, similar processes controlling a relationship may occur in several regions across a landscape, but these regions may be separated by some distance. When a model is generated for a local sample it can readily be applied to any other location in the study area where there are sufficient data. By mapping the locations where errors for such a prediction are low, one might be able to generate interesting hypotheses concerning the relationships between the response and correlate variables. This is described further in the next section.
3. The local analyses are very useful approaches to understanding geographic phenomena. However, one still needs to interpret the underlying relationships represented by the sparse grid models to understand what relationships there are. We now consider a sample of these.

   One method of interpreting the sparse grids system is an analysis of the accuracies of each of the input grids. While this can be implemented as a standard step-wise procedure for model selection (§2.3), the interpretation of the relationship between each of the different correlates is very valuable information (fig 4). When it is mapped as a geographic surface then one can gain even greater insight into the strength of the relationships between variables throughout a study area. With this information one can begin to better understand the system of relationships as they vary through space.



Band 4

Elevation

Distance from Streams

Distance from Swamps

Band 4 & Elevation

Band 7 & Dist from Streams

Slope & Distance from Swamps

Dist from Streams & from Swamps

Fig. 4. Residual error density plots for a sample of sparse grids used in a global analysis of silica in relation to topographic, hydrological and Landsat spectral indices. Blue represents a higher density of prediction/observation combinations, red is lowest. Predictions from a perfect model will plot along the diagonal.

The above approach is not particular to sparse grids, as it can be applied to any predictive model where one can derive the relationships between variables. What sparse grids also allow one to do is analyse the number of grid points (and thus model complexity) used for each grid (see §2.1). If the analysis is conducted such that the number of grid points is optimised to the data, then some understanding of the nature of the relationship may be obtained. For example, if the relationship between the correlate and one predictor variable is consistently represented using few grid points, and has a reasonably good accuracy, then there is a broad relationship. If many grid points are used, then the relationship is one of fine detail. This should, of course, be corrected to allow for the variability of the original data. These indices can be readily extracted from the sparse grids system.

3.2 Mining sparse grid collections

In addition to the analysis using local predictive models sparse grids allow further analysis due to the fact that different sparse grids can be readily compared at various levels. How this can be done is discussed in the following. To focus the discussion one can consider the fitting problem. As above, one determines a family of predictive models $f(x;z)$ where z represents the local area. In our example, the models are determined as the minimisers of a family of functionals

$$J(f;p) = \sum_{i=1}^{n} w(\|p_i - p\|)(y_i - f(x_i))^2$$

where p_i are the spatial locations of the observations and p the location of the local model found. Here the weight functions are given, for example

$$w(s) = \exp(-\lambda s^2)$$

where λ is a given coefficient. Possible centres z used could include a large subset of the data points x_i. Models are then fitted in a greedy way for each of the z. Of course, the computational requirements are considerably increased in this way. The difference to the approach considered so far and discussed above is, however, what is done after all the models have been determined.

After this first step a large number of local sparse grid functions have been determined which model local data and they all have a spatial attribute. This collection of spatial models can now be mined using spatial mining tools. Aspects of spatial mining include the detection of interesting regions, i.e., regions where certain models are prevalent, e.g., where particular types of correlations can be found. Another aspect is the detection of spatial trends, i.e., the analysis of change of the models in space. The underlying ideas are very similar to traditional spatial data mining, see, e.g. [6] where tools are provided for the detection of interesting regions which either have concentrations of high values of certain features [7] or have trends of feature values. The application of this analysis benefits from the fact that the sparse grid spaces are fairly simple linear spaces.

Basically, the functions are grouped into classes of similar functions and the support in the data (similar to association rule mining) and geographical distribution for each of these classes is determined. Classes can be either given a priori, or they may be found by clustering. The a priori given classes are obtained by comparing various features of the locally found models:

- A simple analysis could consider only the set of variables selected for the sparse grids at different locations. These selected variables are the ones which jointly provide a best explanation of the variation of the response. The coincidence of gradients of the response with gradients in other variables and possibly higher level variations may be detected. In this case two models are "the same" if they use the same variables.
- The location and feasibility of linear models can be investigated.
- A slightly more sophisticated analysis would consider the types of interactions used in any one model
- Finally, one can discretise model space and investigate support and location of specific models.

A challenge of some these approaches is that one needs to introduce a metric on the sparse grid space in order to find similar models. However, once the metric has been selected and once the collections of models have been found one may further mine these collections using methods related to frequent itemset mining. In addition the geographical distribution of certain classes of models may provide information about underlying processes.

References

1. Dietrich Braess. *Finite elements.* Cambridge University Press, Cambridge, second edition, 2001.
2. Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.
3. P. A. Burrough. Soil variability: A late 20th century view. *Soils and Fertilizers*, 56:529–562, 1993.
4. Ingrid Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
5. Carl de Boor. *A Practical Guide to Splines.* Springer, 1978.
6. M. Ester, H.-P. Kriegel, and J. Sander. Knowledge discovery in spatial databases. In *23rd German Conf. on Artificial Intelligence (KI '99)*, volume 1701 of *Lecture Notes in Computer Science*, pages 61–74. Springer Verlag, 1999.
7. V. Estivill-Castro and I. Lee. Data mining techniques for autonomous exploration of large volumes of geo-referenced crime data. In D. Pullar, editor, *Proceedings of the 6th International Conference on GeoComputation*, University of Queensland, Brisbane, September 2001.
8. Jerome H. Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19(1):1–141, 1991. With Discussion and A Rejoinder By The Author.

9. J. Garcke and M. Griebel. Classification with sparse grids using simplicial basis functions. *Intelligent Data Analysis*, 6(6):483–502, 2002.
10. J. Garcke, M. Griebel, and M. Thess. Data mining with sparse grids. *Computing*, 67(3):225–253, 2001.
11. M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In P. de Groen and R. Beauwens, editors, *Iterative Methods in Linear Algebra*, pages 263–281. IMACS, Elsevier, North Holland, 1992.
12. Michael Griebel, Michael Schneider, and Christoph Zenger. A combination technique for the solution of sparse grid problems. In *Iterative methods in linear algebra (Brussels, 1991)*, pages 263–281. North-Holland, Amsterdam, 1992.
13. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, 2001.
14. T. J. Hastie and R. J. Tibshirani. *Generalized additive models*, volume 43 of *Monographs on Statistics and Applied Probability.* Chapman and Hall Ltd., London, 1990.
15. M. Hegland. Adaptive sparse grids. In K. Burrage and Roger B. Sidje, editors, *Proc. of 10th Computational Techniques and Applications Conference CTAC-2001*, volume 44, pages C335–C353, April 2003. [Online] `http://anziamj.austms.org.au/V44/CTAC2001/Hegl` [April 1, 2003].
16. M. Hegland, O. Nielsen, and Z. Shen. High dimensional smoothing based on multilevel analysis, 2000.
17. Markus Hegland. Additive sparse grid fitting. In *Curve and Surface Fitting: St Malo 2002*, pages 209–218, 2002.
18. S. W. Laffan. *Inferring the Spatial Distribution of Regolith Properties Using Surface Measurable Features.* PhD thesis, Australian National University, 2001.
19. S. W. Laffan. Using process models to improve spatial analysis. *International Journal of Geographical Information Science*, 16:245–257, 2002.
20. S. W. Laffan and B. G. Lees. Predicting regolith properties using environmental correlation: a comparison of spatially global and spatially local approaches. *Geoderma*, in press.
21. S. W. Laffan, H. Silcock, O. Nielsen, and M. Hegland. A new approach to analysing spatial data using sparse grids. In *MODSIM 2003, Integrative Modelling of Biophysical, Social and Economic Systems for Resource Management Solutions*, 2003.
22. Z. Luo, G. Wahba, and D. R. Johnson. Spatial-temporal analysis of temperature using smoothing spline ANOVA. *Journal of Climate*, 11:18–28, 1998.
23. H. J. Miller and J. Han. *Geographic data mining and knowledge discovery.* Taylor and Francis, London, UK, 2001.
24. S Openshaw. Geographical data mining: key design issues. In *Proceedings of the 4th International Conference on GeoComputation, Geocomputation99*, Mary Washington College, Virginia, USA, July 1999.
25. D. J. Peuquet. *Representations of space and time.* Guilford, 2002.
26. Stephen Roberts, Markus Hegland, and Irfan Altas. Approximation of a thin plate spline smoother using continuous piecewise polynomial functions. *SIAM J. Numer. Anal.*, 41(1):208–234 (electronic), 2003.
27. J. F. Roddick and B. G. Lees. Paradigms for spoatial and spatio-temporal data mining. In H. J. Miller and J. Han, editors, *Geographic data mining and knowledge discovery*, chapter 2, pages 33–49. Taylor and Francis, 2001.
28. S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 148:1042–1043, 1963. Russian, Engl. Transl.: Soviet Math. Dokl. 4:240–243, 1963.

29. W. R. Tobler. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46 (supplement):234–240, 1970.
30. T. G. Whiteway, S. W. Laffan, and R. J. Wasson. Using sediment budgets to investigate the pathogen flux through catchments. *Environmental Management*, submitted.
31. C. Zenger. Sparse grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, 1990*, volume 31 of *Notes on Num. Fluid Mech.*, pages 241–251. Vieweg, 1991.

CONGO: Clustering on the Gene Ontology

Paul J. Kennedy[*] and Simeon J. Simoff[**]

Faculty of Information Technology, University of Technology, Sydney, PO Box 123,
Broadway, NSW 2007, AUSTRALIA

Abstract. Rapid development of technologies for the collection of biological data have led to large increases in the amount of information available for understanding diseases and biological mechanisms. However, progress has not been as fast in comprehending the data. Developments in understanding diseases and biological mechanisms governing them may come from combining data from different sources. We describe a method of clustering lists of genes identified as important to the understanding of a childhood cancer using functional information about the genes from the Gene Ontology. The measure of distance used in the clustering algorithm is notable for considering the relationship between terms in the ontology. Meaningful descriptions of clusters are automatically generated from the Gene Ontology terms.

1 Introduction

Rapid developments in bio–technology, measurement and collection of diverse biological and clinical data have led to revolutionary changes in bio–medicine and biomedical research. The data collected in bio–medical experiments or as a result of medical examination ranges from gene expression levels measured using microarray technologies to data collected in therapy research. Researchers are looking at discovering relations between patterns of genes (sequences, interactions between specific genes, dependencies between changes in gene expressions and patient's responses to treatment). The confluence of bio–technology and statistical analysis is known as bioinformatics. The "classical" statistical techniques used in bioinformatics — a broad range of cluster, classification and multivariate analysis methods, have been challenged by the large number of genes that are analysed simultaneously and the curse of dimensionality of gene expression measurements. As a rule, the gene–to–data points ratio is high (i.e. the so–called "wide" data table, i.e. if we are looking at N genes and our sample is of size m, then usually $N \gg m$). When there are more attributes than data records (cases), problems may arise (for example, there can be strong correlations between some of the attributes, or the covariance matrix may become singular, the curse of dimensionality may begin to bite). This challenge has attracted the attention of researchers in the two very closely related fields of "data mining" (initiated by

[*] paulk@it.uts.edu.au

[**] simeon@it.uts.edu.au

researchers in databases (see [1])) and "intelligent data analysis" (initiated by researchers working in the area of mathematical statistics and machine learning (see Chap. 1 in [2])). Bearing in mind that researchers and research communities often disagree about the precise boundaries of their dedicated field of investigation, further in this paper we refer only to data mining [3] as the "analysis of large observational data sets to find unsuspected relationships and to summarise the data in novel ways that are both understandable and useful to the data owner". There is a number of ways in which data mining is expected to be able to assist the bio–data analysis (see [4] for brief overview). One important area are the tasks of similarity search, comparison and grouping of gene patterns and assisting in understanding these patterns in medical bio–data, as many diseases are triggered by a combination of genes acting together. The work presented in this paper is in this area.

Addressing the "Wide" Data Table Problem

Having many more genes than data points offers a number of strategies for the analysis of such data [5], that can be grouped in three broader categories: "summarise then analyse" (STA), "analyse then summarise" (ATS) and "summarise while analysing" (SWA). STA scenario uses an unsupervised learning technique (e.g. cluster analysis) to reduce the large number of genes to gene clusters (or gene profiles). The cluster representations then are used for predictive modelling (see [6]). In ATS scenario, modelling is conducted initially for each gene, producing some statistics, and then one can apply some threshold (for example, select all genes with value of that statistic above the threshold). SWA approach addresses the issues of possible existence of some relations between the genes, hence, suggests to proceed with summarisation and classification in a single step. For example, regression tree model [7] can be used to identify a small subset of predictive genes.

The above presented scenarios do not consider the utilisation of already existing knowledge about relations between genes to assist the outcome of the data mining step. The approach proposed in this paper extends the STA scenario, by imposing the results of the initial clustering of the genes with further clustering over an ontology that relates the genes in the input clusters. This approach can be labelled as "summarise, impose, then analyse" (SITA).

Cluster Analysis and Visualisation

As we have mentioned earlier, clustering algorithms divide the set of genes into groups so that gene expression patterns within a group are more similar than the patterns across groups. Most clustering techniques include a "magic" set of parameters, that one needs to adjust to get "good" clusters. However, in the case of gene expression data sets, the selection and "tuning" of these parameters may not be that intuitive and obvious, due to the high dimensionality of the space. Hence, clustering relies substantially on visualisation. An efficient visualisation schema allows to expose problems with the clusters, prompting towards some intervention, for example, selection of different similarity and inter–cluster distance measures, or forcing some of the clusters into one group. The paper presents a visualisation method that supports the proposed SITA scenario.

In this paper, we use information from one source (the Gene Ontology [8]) to gain an understanding of a list of genes that were generated as the result of another data mining step. The list of genes is clustered into groups of genes with similar biological functionality. Descriptions of the clusters are automatically determined using the Gene Ontology (GO) data.

The broad goals of our bioinformatics project are to improve the understanding of genes related to a specific form of childhood cancer. Data regarding the relative expression levels of genes (in tumour cells compared with normal cells) is combined with clinical data (concerning the tumours and patients) to form a list of "interesting" genes. Details of this step are not relevant to the techniques explored in this paper.

The Gene Ontology is a controlled vocabulary of terms that describe gene products in terms of their effect in the cell and their known place in the cell. Terms in the ontology are interrelated. For example, a "glucose metabolism" *is a* "hexose metabolism" (see Fig. 1). In this example, "hexose metabolism" is a more general concept (or term) than "glucose metabolism". There are currently around 16,000 terms in the Gene Ontology and each gene is associated with between two and ten terms. The relationships between terms in the ontology allow us to measure the similarity between genes in a functional way. For example, one gene may be associated with the term "carbohydrate metabolism" and another gene associated with "alcohol metabolism". As can be seen in Fig. 1 both of these terms are child terms (or more specific concepts) of "metabolism". Hence, they are related quite closely.

The list of genes, then, is clustered according to the associated Gene Ontology terms. The clustering considers the interrelationships of terms in the ontology. Once clusters are created, the terms in the Gene Ontology permit the automatic construction of cluster descriptions (in terms of the Gene Ontology concepts).

The method of clustering over an ontology is general and may be applied to (non–biological) data associated with other ontologies.

Applying information from the Gene Ontology to cluster genes allows for an understanding of the genes and their interrelationships in functional terms. Currently biologists search through such lists gene–by–gene analysing each one individually and trying to piece together the many strands of information. Automating the process, at least to some extent, would allow biologists to concentrate more on the important relationships rather than the minutiae of searching as well as give savings in time and effort.

Related Work

Other workers use the Gene Ontology. There are a variety of browsers for the Gene Ontology linked from their web site [9]. In general, such browsers have facilities such as: (i) traversing the large Gene Ontology and viewing their interrelationships; (ii) finding Gene Ontology terms associated with ensembles of

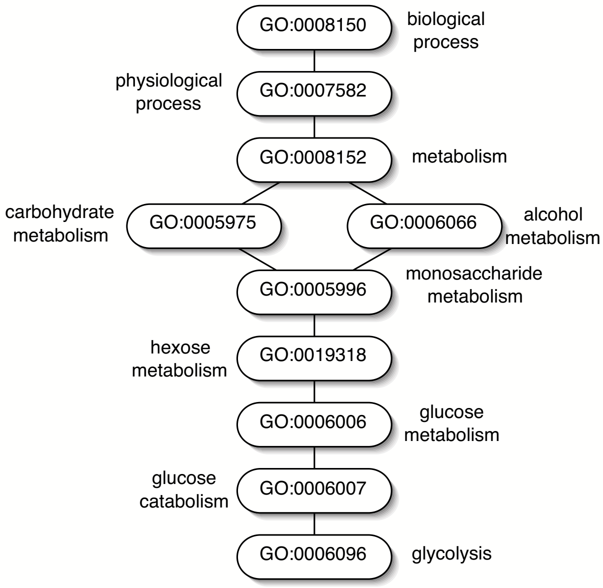


Fig. 1. A small section of the GO hierarchy from the "biological processes" ontology. Each node is a term in the ontology. Inside each node is the identifier for the term and beside is the term itself. More general terms are towards the top of the diagram. All links shown are is–a relationships that are directed upwards

genes; or (iii) finding known genes associated with particular Gene Ontology terms, to name a few.

Many tools (for example, eGOn [10] or FatiGO [11]) take as input a list of genes (often resulting from microarray experiments) and map the genes to GO categories. Most of these tools additionally allow comparison of GO mappings between different gene lists usually with some statistical measure of the similarity of distributions. The tools GOMiner [12] [13] and EASE [14] [15] additionally look for "biological themes" in lists of genes. That is, they identify the predominant set of GO terms that describe the entire gene list. They have a similar goal to the method we propose, except that we first cluster the data into subsets of related genes.

Hierarchical information is also used with other data mining techniques (possibly unrelated to biology). For example, [16] and [17] use ontological information to mine "generalized" association rules. The "basic" algorithm in [17] takes an approach that is reminiscent of ours (ie. simply including information from higher in the tree). Both the generalized association rules and the ontological clustering in this paper use the idea of combining specialised concepts but have different goals. The generalized association rules combine them to produce stronger rules, whereas we combine concepts to build looser forms of equivalence to make the clustering more flexible.

Method Overview

The cluster analysis and visualisation described in this paper takes as input (i) a list of genes highlighted from a previous data mining step and (ii) data from the Gene Ontology. The previous data mining step used gene expression data (from cDNA microarray experiments) and clinical data describing the tumour cells in detail, effect of drug protocols and (human) classifications of patients into high or low risk categories. cDNA microarray experiments are a recent technology available to cellular biologists that measure the relative expression levels of thousands of genes in cells at one instant. Expression levels of genes in a test sample (i.e. tumour cells) compared to genes in a control sample (i.e. "normal" cells) are measured.

Gene Ontology terms are associated with each gene in the list by searching in the SOURCE database [18]. The list of genes is clustered into groups with similar functionality using a distance measure that explicitly considers the relationship between terms in the ontology. Finally, descriptions of each cluster are found by examining Gene Ontology terms that are representative of the cluster. Graphs of Gene Ontology terms for each cluster together with cluster descriptions give a visualisation of each cluster in functional terms.

2 The Gene Ontology

The Gene Ontology [8] is a large collaborative public database constructed by researchers world–wide. It provides a set of controlled vocabularies (i.e. ontologies) of terms that describe gene products in terms of their effect in the cell. That is, their functionality. The goal of the Gene Ontology is "to produce a dynamic, controlled vocabulary that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing" [8].

As described in Sect. 1 the Gene Ontology contains terms and their interrelationships (parent/child, general/specific, etc). Three ontologies are defined in the Gene Ontology: (i) biological processes, (ii) cellular components, and (iii) molecular functions. The ontologies are directed acyclic graphs (DAGs) where the terms form nodes and two kinds of relationships form edges: "is–a" relationships such as "glycolysis" is–a "glucose catabolism" and "part–of" relationships such as "nuclear chromosome" is part–of "nucleus". Apart from the specific individual terms, the Gene Ontology is unremarkable in this regard. All ontologies are DAGs of terms.

Each term in the ontology has a number of attributes: the term itself (eg. glycolysis), a unique accession number (eg. GO:0006096), and a definition (eg. the breakdown of a monosaccharide (generally glucose) into simpler components, including pyruvate). There may also be technical references to the definition (eg. links to PubMed articles), cross references into other biological databases, synonyms and comments.

There are a number of benefits of using the Gene Ontology as part of the data mining process. It is large (7045 terms in the Molecular Function ontology, 7763 terms in the Biological Process ontology and 1335 terms in the Cellular Component ontology as of 16 September 2003 [9]) and well worked on by researchers (16 member organisations of the Gene Ontology Consortium as of August 2003 [9]). Entries are curated before being added to the ontology. The ontology may be accessed in the RDF XML file format. In this computer legible form it is easier to apply the information to data mining methods and immediately richer than by determining similar information with text mining methods.

GO terms may be associated with genes using databases like SOURCE [18] as long as accession numbers of genes or gene names are known. See Table 1 for an example.

Table 1. GO terms associated with an example gene (named CLK1) for each of the three ontologies.

CLK1 (CDC–like kinase 1)
Molecular Function
GO:0004715 non–membrane spanning protein tyrosine kinase activity
GO:0005524 ATP binding activity
GO:0004674 protein serine/threonine kinase activity
GO:0016740 transferase activity
Biological Process
GO:0006468 protein amino acid phosphorylation
GO:0008283 cell proliferation
GO:0000074 regulation of cell cycle
Cellular Component
GO:0005634 nucleus

3 Clustering over Ontologies

Many algorithms exist for clustering data (see for example [19] or [20]). The data we wish to cluster is slightly different to normal, however, and this advises our choice of algorithm and distance measure.

There are two main differences between our clustering and “normal” cluster analysis. The first difference is that there are a different number of attributes (GO terms) for each gene to be clustered whereas usually the number of attributes in a dataset is the same for all records. Secondly, we are interested in complex relationships between terms (as a result of the structure of the ontology) so simply comparing values of terms with one another will not be sufficient.

Both difficulties stem from the fact that there is an ontology associated with the data. Once we solve the data mining problem of clustering over an ontology, the special case of clustering over the Gene Ontology will follow easily.

Similarities might be drawn with clustering text documents (for example into spam and non–spam), as there are different numbers of words in each document and complex relationships among the words (ontological ones too). One approach to clustering documents is to use a fixed length vector of word counts in each document, with each vector position representing a different word (drawn from an a priori prescribed list). In this way each document to be classified with potentially many different words and counts of words is reduced to a fixed number of attributes with all documents having the same number of attributes.

A similar approach could be applied to cluster the genes and GO terms. A fixed length binary vector of the union of all GO terms in the genes could be set up as the attributes for each gene. A bit would be set if the term was associated with the gene or unset if there was no association. Such an approach, however, suffers from two defects. Firstly, the vast majority of GO terms are only associated with one gene in the dataset. This would mean the binary vectors for genes would be very sparse and few similarities could be found with the vectors for other genes in the dataset. The other, more serious, problem with this approach is that it does not take into account the ontological relationships at all.

Our method solves the problem of different numbers of attributes by treating all the terms for a gene as essentially one attribute. The second problem of considering the ontological relationships is accomplished by using a more specialised distance function that compares a set of terms based on their relative positions in the ontologies, rather than just the value of the term, which is, essentially, meaningless.

The distance function, then, is the crucial element and the particular clustering algorithm used is a secondary consideration. We use a simple clustering algorithm named the Modified Basic Sequential Algorithmic Scheme (MBSAS). This particular algorithm was chosen because of its simplicity and because it is not necessary to specify a priori the number of clusters. One of many other algorithms (eg. k–means) could have been used instead.

In the following two subsections we will describe in more detail the distance measure and the MBSAS clustering algorithm.

Distance Measure

The elements to be clustered have different numbers of attributes and this means that a special distance measure must be used. The distance measure is special in that it measures distances across the ontology. The distance measure is in some ways more important than the actual clustering algorithm as any of many different clustering algorithms may be used, but a distance measure similar to this must be used to traverse the ontology.

We use a function adapted from the Tanimoto Measure [19] [20]. The Tanimoto measure provides a measure of similarity between sets:

$$\frac{n_{X\cap Y}}{n_X + n_Y - n_{X\cap Y}} = \frac{n_{X\cap Y}}{n_{X\cup Y}} \tag{1}$$

where X and Y are the two sets being compared and n_X, n_Y and $n_{X \cap Y}$ are the number of elements in the sets X, Y and $X \cap Y$ respectively.

In our situation, the "sets" being compared are the GO terms for two genes. However, for reasons which will become clear, "bags" (where elements may be repeated) are used rather than sets.

An important characteristic of our distance measure is that it considers terms higher in the ontology. This is because the GO terms themselves are simply constant values with no implicit relationship to other terms. As in any ontology, the relationship between terms arises from their relative positions in the hierarchy. So, for each gene we wish to compare, we add to the gene's associated GO terms all terms higher in the ontology. These terms form a "background" or context to the terms explicitly associated with the gene. However, as the ontologies are tree–like, two terms in a gene often have the same ancestors. We include the parent terms each time they are encountered, so we require bags rather than sets.

Terms higher in the ontology represent terms that are more general. Although general terms are a factor in the comparison, the more specialised terms (i.e. lower in the hierarchy) are more important. For this reason, when counting the number of terms in a bag, terms are weighted by their distance from their descendent GO term explicitly associated with the gene. In effect, we calculate a "weighted" cardinality of the bag of GO terms.

The final distance function used, then, is

$$D_{X,Y} = \frac{n'_{X \cap Y}}{n'_X + n'_Y - n'_{X \cap Y}} = \frac{n'_{X \cap Y}}{n'_{X \cup Y}} \tag{2}$$

where X and Y are the two bags of terms being compared and n'_X, n'_Y and $n'_{X \cap Y}$ are the weighted cardinalities of the bags X, Y and $X \cap Y$ respectively given by

$$n'_X = \sum_{i \in X} c^{d_i} \tag{3}$$

where X is the bag of GO terms, d_i is the distance of element of X with index i from its associated descendent in the original set of GO terms for the gene, and c is the weight constant. The weighted cardinality of the other bags is similarly defined.

The more general terms provide a context for the lower level terms directly associated with genes. The c parameter allows variation of the importance of the "context" to the comparison. A value of $c = 0$ means that ancestral terms are not considered. A value of 1 would mean that all terms are considered equally as part of the context. Plainly, in this case though, the very general terms would be regarded as overly important. The c parameter, then, may be viewed as a sort of "constant of gravity" for the clusters. The higher the value of c, the easier it is that distantly related genes gather into a cluster. We arbitrarily chose $c = 0.9$ for our experiments.

Other distance measures apart from a gene–to–gene distance are also required for use in the clustering algorithm. A measure of the distance between a gene and a cluster of genes is determined by taking the average distance from the gene to each gene in the cluster. Similarly when calculating the distance between two clusters of genes we use the average of the distances for each gene of one cluster to the genes in the other cluster. An alternative to using the mean distances would be to use minimum (or maximum) distances. We plan to explore these possibilities in the future.

Cluster Algorithm

With the intention of attacking the clustering problem as simply as possible, we use a standard simple clustering algorithm called Modified Basic Sequential Algorithmic Scheme (MBSAS) as described by [19]. MBSAS has two advantages compared with other algorithms such as the ubiquitous k–means algorithm. It is (i) not necessary to specify a priori the number of clusters; and (ii) the data is presented to the algorithm only a few times (depending on the particular variation of MBSAS chosen).

The variation of MBSAS we use is dependent on three parameters (and one other parameter is necessary for the distance measure). These parameters are shown in Table 2. Whilst MBSAS does not require an explicit parameter for the number of clusters, the parameters (Θ, q and M_1) have the same effect.

Table 2. Parameters used in the Modified Basic Sequential Algorithmic Scheme clustering algorithm. The last parameter is used only in the distance measure and is not formally part of MBSAS. See text for a detailed description of c.

Parameter	Meaning
Θ	Minimum distance for points to be considered to be in the same cluster. (Theodoridis and Koutroumbas [19] call this the "threshold of dissimilarity").
q	Maximum allowable number of clusters.
M_1	Minimum distance for clusters to be deemed separate before they are merged.
c	Discount weight applied to GO nodes in the ontology.

The MBSAS algorithm has four main steps as described below. The first two steps are mandatory, whilst the latter two are optional.

```
1. determine_clusters
2. classify_patterns
3. merge_nearby_clusters (optional)
4. reassign_points (optional)
```

The `determine_clusters` step determine the initial clusters. It chooses up to q data points that are sufficiently distant from one another (using the Θ parameter) as point representatives.

After finding the initial clusters the next step (`classify_patterns`) classifies the rest of the patterns into the cluster that is closest using $D_{X,Y}$ as defined in (2).

Theodoridis and Koutroumbas [19] describe two general drawbacks of sequential clustering algorithms. They are (i) that clusters may arise that are very close together and (ii) that they are sensitive to the order of presentation of the data. The third and fourth steps address these problems respectively. Although optional, we always perform them.

The `merge_nearby_clusters` step identifies clusters having a distance less than the value of parameter M_1 and merges them together.

Finally, in the `reassign_points` step, all points are reassigned to their closest cluster so as to minimise the effects of the presentation order of the data and any changes due to the `merge_nearby_clusters` step.

4 Experiments

As described in Sect. 1 the data used for this paper was a list of genes highlighted as the result of a previous data mining procedure. Information from the Gene Ontology was matched to the genes using the SOURCE database.

There are, at this stage, two goals for our experiments: (i) discovery of parameter values that produce acceptable clusters and (ii) determination of ways to visualise the clusters.

The parameter values Θ and M_1 are dependent on the range of values returned by the distance measure $D_{X,Y}$ and have been determined largely by trial and error. In the experiments described in this paper, Θ is set to 0.001 and M_1 to 0.1. The maximum number of clusters (q) is set at 5 and, as described above, c, the discounting constant for more general terms is set at 0.9.

Visualisation of clusters is made difficult by the fact that there is no clear way to transform genes into coordinates to plot on a single graph because each gene is identified by different numbers of GO terms. So we plot the terms for all the genes on a graph with their relationships shown in different shades for each cluster. We also automatically build cluster descriptions from the terms in each cluster.

5 Results

With the parameters values given above (i.e. $\Theta = 0.001$, $M_1 = 0.1$, $q = 5$ and $c = 0.9$) five clusters are found as shown in Table 3. Half of the genes have been allocated to one cluster. The rest of the genes have been split into four smaller clusters with one cluster containing only two genes. Such a tabular representation does not increase our understanding of the clusters as the gene accession codes are not descriptive.

With this in mind, we plotted the subset of terms associated with the clustered genes as nodes on a graph with relationships represented by edges and the

Table 3. Clusters found with the MBSAS clustering algorithm. The codes AA*nnnn* are GenBank accession codes.

Cluster Number	Gene Count	Genes
0	6	AA040427 AA406485 AA434408 AA487466 AA609609 AA609759
1	2	AA046690 AA644679
2	6	AA055946 AA398011 AA458965 AA487426 AA490846 AA504272
3	9	AA112660 AA397823 AA443547 AA447618 AA455300 AA478436 AA608514 AA669758 AA683085
4	20	AA126911 AA133577 AA400973 AA464034 AA464743 AA486531 AA488346 AA488626 AA497029 AA629641 AA629719 AA629808 AA664241 AA664284 AA668301 AA669359 AA683050 AA700005 AA700688 AA775874

GO nodes of a cluster localised to one part of the graph as much as possible (Fig. 2). The clusters are represented by the five large boxes with the cluster numbers (as listed in Table 3) given inside each box. Nodes inside the clusters are the GO terms associated with genes in that cluster. More general terms are on the right hand side of the diagram. Edges between nodes represent the links in the ontology. Some terms, particularly the more general ones at the right hand side of the diagram, have links from terms in a different cluster. Each node is shown in only one cluster box, but links between the boxes show where GO terms are shared by genes in the different clusters. The grey scale of the link represents the cluster that link is in. Also, a darker grey scale is used for links in the original dataset whilst a lighter shade is used for relationships inferred from traversing the ontology. Inside some cluster boxes may be seen links from a different cluster (if both child and parent terms are drawn in one cluster box, but the link is also in another cluster). For example, inside the large middle cluster (representing cluster 4) may be seen some links associated with the second top cluster (representing cluster 3) although this is difficult to see on the diagram. It is likely that these are either outliers or indicators of poor clustering.

Figure 2 is reminiscent of the dendrograms that are used in hierarchical clustering. This is hardly surprising since both methods are dealing with hierarchies. However, in Fig. 2 the length of edges is not correlated to the distance between nodes (as in dendrograms). We will apply a hierarchical clustering algorithm in the future.

Figure 3 shows essentially the same information as Fig. 2 except that the more general terms are at the bottom of the diagram. To improve the readability of the diagram, the cluster boxes are in a different order than in Fig. 2. Again, cluster numbers are given inside each box. The GO terms lying along the bottom edges of the cluster boxes are clearer in this diagram, particularly those on the left– and rightmost cluster boxes (clusters 3 and 4). These terms are part of

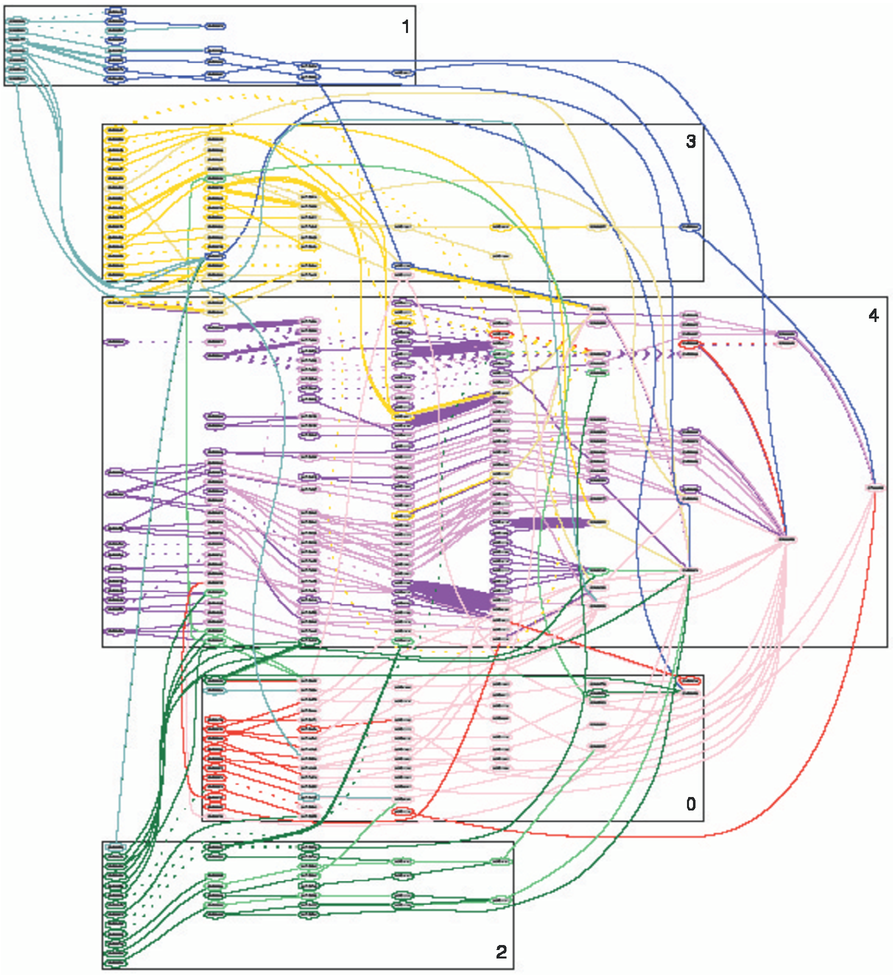


Fig. 2. Parts of the GO hierarchy associated with genes being clustered. More general terms are at the right of the diagram. See text for description of graph

the most general descriptions for a cluster that *do not also* describe another cluster. Figure 4 shows a closer view of the terms at the bottom edge of the large rightmost cluster (number 4). These terms are used to automatically determine cluster descriptions. Another feature visible in Fig. 3 are the links that fly from one cluster to another. These are important because they show where cluster meanings overlap or blur together. The rope of links at the bottom right of the diagram is unimportant as these links are to the most general terms and therefore, the least descriptive for our purposes.

A good visualisation of clusters should make evident the properties that genes in a cluster share. Essentially this entails a functional description of a cluster. A good description might also state how the cluster differs from other clusters.

The ontology is able to describe how genes are similar. Cluster descriptions are inferred in the following way. Starting with all the GO terms directly associated with genes in a particular cluster, we climb the hierarchy replacing GO terms with their parent terms. Terms are replaced only if the parent node is *not* associated with genes in another cluster (or is one of any of the ancestor terms in another cluster). This results in a list of terms for a cluster that describe in the most general way possible the union of functionalities of all genes in that cluster (but not so general that it describes another cluster).

Cluster descriptions derived in this way are shown in Table 4. Only the *is–a* relationships were followed to build this table. We expect to trace the *part–of* relationships in future work. There are far fewer *part–of* relationships in the hierarchies so we do not believe that omitting them affects the results. The cluster descriptions give some insight to the genes in the cluster and also give feedback on the quality of the clustering. The terms listed in the table are associated only with genes in each cluster and not in any other cluster.

Cluster 0 in Table 4 has no terms that are associated with more than one gene. This suggests that the genes in the cluster are either unrelated or related only in ways that are sufficiently high level that the terms exist in other clusters. This suggests that the quality of the cluster is not good.

The other clusters, however, have genes that are more strongly interrelated. Cluster 1 contains at least two genes that are related to the cell cytoskeleton and to microtubules (microtubules are components of the cytoskeleton). Cluster 2 contains three or four genes associated with signal transduction and cell signalling. Cluster 3 contains three or four genes related to transcription of genes and cluster 4 seems to contain genes associated with RNA binding.

The question, however, may be asked: what about the other genes in the clusters? What is their relationship? Are these genes unrelated to the "core" description of the cluster and just bundled into the cluster because the maximum number of clusters q has been reached, or are there more subtle relationships? The simple statistic of the number of genes associated with each GO term in the cluster is insufficient to answer the question. The names of the individual genes are required. This will be investigated further in future work. Also, we plan to cluster the data into more clusters, perhaps with an hierarchical clustering algorithm to determine whether better descriptions and "tighter" clusters result.

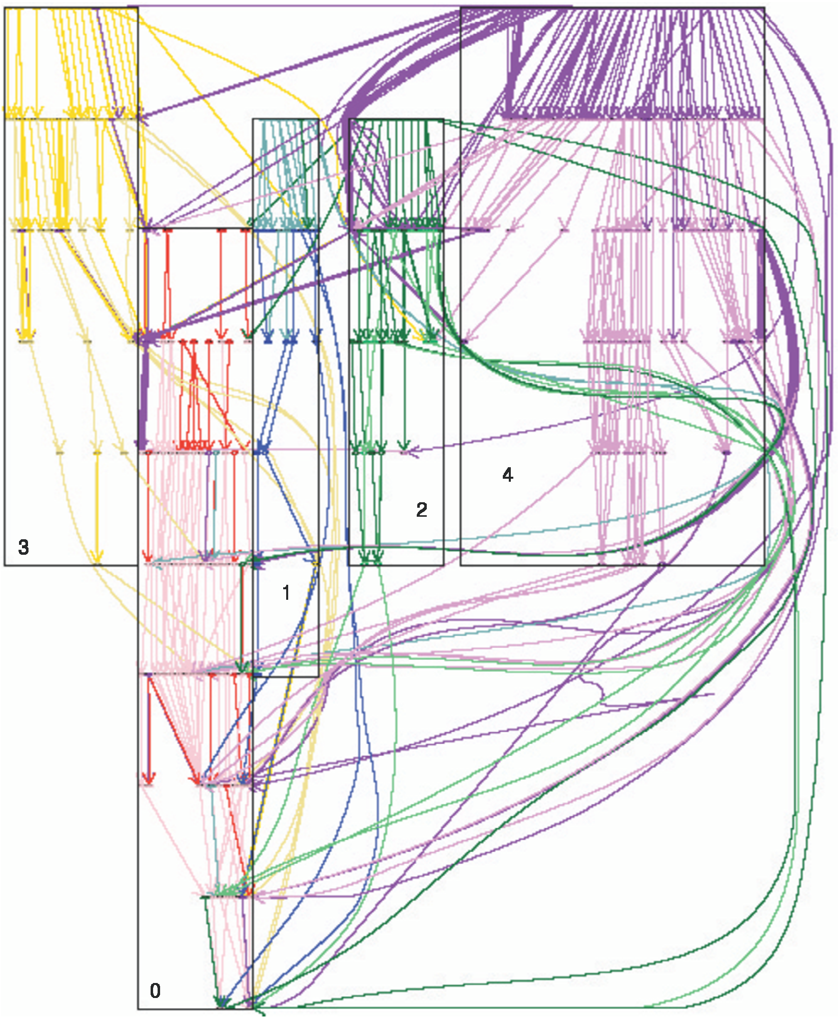


Fig. 3. Diagram showing essentially the same information as Fig. 2 except that important descriptive GO terms are more visible. See text for description of graph

Table 4. Principal cluster descriptions for the genes clustered with the MBSAS algorithm derived as stated in the text. The last column gives the number of genes in the cluster associated with the term.

GO ID	GO Term	Number of Genes
Cluster 0 — 6 genes		
	20 GO terms but each associated with only one gene	1
Cluster 1 — 2 genes		
GO:0008092	cytoskeletal protein binding activity	2
GO:0007028	cytoplasm organization and biogenesis	2
GO:0003774	motor activity	2
GO:0005875	microtubule associated complex	2
	5 GO terms but each associated with only one gene	1
Cluster 2 — 6 genes		
GO:0004871	signal transducer activity	4
GO:0007154	cell communication	4
GO:0005887	integral to plasma membrane	3
GO:0005886	plasma membrane	3
GO:0005194	cell adhesion molecule activity	2
	11 GO terms but each associated with only one gene	1
Cluster 3 — 9 genes		
GO:0030528	transcription regulator activity	4
GO:0008134	transcription factor binding activity	3
GO:0006366	transcription from Pol II promoter	3
GO:0003700	transcription factor activity	3
GO:0006357	regulation of transcription from Pol II promoter	3
	5 GO terms but each associated with only two genes each	2
	13 GO terms but each associated with only one gene	1
Cluster 4 — 20 genes		
GO:0003723	RNA binding activity	10
GO:0030529	ribonucleoprotein complex	9
GO:0009059	macromolecule biosynthesis	9
GO:0006412	protein biosynthesis	9
GO:0005829	cytosol	9
GO:0003735	structural constituent of ribosome	8
	2 GO terms but each associated with only four genes each	4
	5 GO terms but each associated with only three genes each	3
	1 GO term associated with only two genes	2
	33 GO terms but each associated with only one gene	1

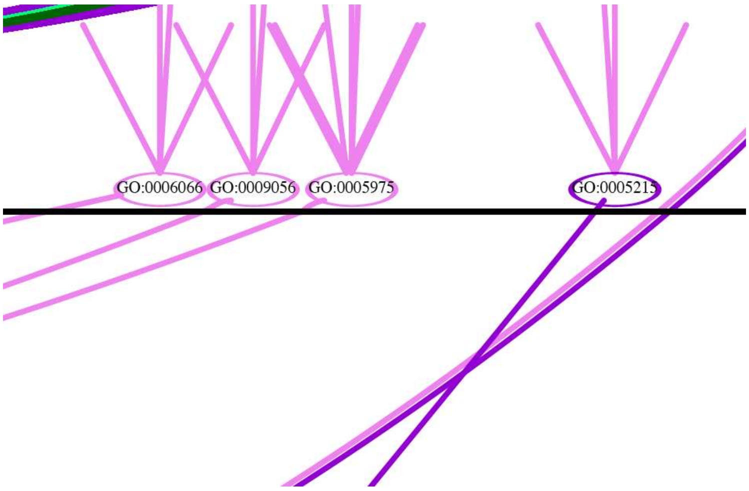


Fig. 4. Diagram showing a close up of the most general GO terms in the large cluster. See text for further description

Another consideration with the possibility of clusters being overly large is that the value of c, the "constant of gravity", might be too large for this dataset. We plan to examine the consequences of lower values of this parameter.

It is also instructive to understand how clusters are different. In a similar way to that described for finding descriptions of clusters, we can build a list of terms that are shared by one other cluster (at their most general level possible). This tells us how two clusters are similar, but different to other clusters. It is essentially an ontological measure of the distance between clusters. The same sort of algorithm could be used for different groupings of clusters. However, an explosion of computational complexity soon occurs.

6 Future Work

Future work may be categorised into four areas: cluster validation, cluster refinement, experimentation with other algorithms and integration of feedback from domain experts.

Validation of the clustering algorithm and the resultant clusters is required to ensure that the clusters describe anything worthwhile. We plan to validate the clustering in three ways: (i) hand choose a set of genes for known GO relationships and then determine whether the clustering algorithm infers at least those relationships; (ii) examine the effect of different sets of q and Θ parameters (as well as the other two parameters) with the aim of seeing whether clusters break up and combine smoothly; and (iii) compare the results of our clustering algorithm with other similar systems.

The clustering algorithm will be refined in the following two ways: (i) the stability of clusters needs to be analysed with respect to the order of presentation of data; and (ii) choice of parameter values requires more understanding.

Different clustering algorithms will be tried. MBSAS was simply a starting point. At least k–means and hierarchical clustering algorithms will be attempted.

The clustering behaviour must be refined based on feedback from medical experts who understand the different genes and will be able to determine whether the clustering increases their understanding of the genes. Cluster analysis like this project is, in some ways, an exercise in prototyping. Once the domain experts gain some knowledge they are able to ask other questions.

7 Conclusions

This paper describes a technique for clustering genes according to their functionality as defined by associated terms in the Gene Ontology. The clustering algorithm is notable for considering the relationships between terms by traversing the ontology.

The Gene Ontology is used to visualise the clusters by automatically building cluster descriptions. Preliminary results clustering genes give insights into the clusters and the efficacy of the clustering algorithm.

Acknowledgements

We would like to thank the other members of our group, Daniel Catchpoole and David Skillicorn for their assistance developing the ideas in this paper and for their analysis and gathering of the data and Andre Skusa and Jacob Koehler for their comments and discussion on drafts of the paper.

References

1. Fayyad, U.M., Piatetsky-Shapiro, G., et al.: From data mining to knowledge discovery in databases. AI Magazine **17** (1996) 37–54
2. Berthold, M., Hand, D.J., eds.: Intelligent Data Analysis. Springer, Heidelberg (2003)
3. Hand, D., Mannila, H., et al.: Principles of Data Mining. The MIT Press, Cambridge, MA (2001)
4. Han, J.: How can data mining help bio–data analysis. In: Proceedings 2nd Workshop on Data Mining in Bioinformatics BIOKDD02, in conjunction with ACM SIGKDD 8th International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, ACM Press (2002)
5. Parmigiani, G., Garrett, E.S., et al.: The analysis of gene expression data: An overview of methods and software. In Parmigiani, G., Garrett, E.S., Irizarry, R.A., Zeger, S.L., eds.: The analysis of gene expression data, Heidelberg, Springer–Verlag (2003) 1–45
6. Rosenwald, A., Wright, G., et al.: The use of molecular profiling to predict survival after chemotherapy for diffuse large–B–cell lymphoma. New England Journal of Medicine **346** (2002) 1937–1947

7. Hastie, T., Tibshirani, R., et al.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer–Verlag, Heidelberg (2001)
8. The Gene Ontology Consortium: Gene Ontology: tool for the unification of biology. Nature Genetics **25** (2000) 25–29 PubMed ID:10802651.
9. Gene Ontology Consortium: Gene Ontology Consortium. Available on: `http://www.geneontology.org` (2003) Viewed at 15 October 2003.
10. Norwegian University of Science and Technology: eGOn (explore Gene Ontology). Available on: `http://nova2.idi.ntnu.no/egon/` (2003) Viewed at 23 October 2003.
11. Al-Shahrour, F., Díaz-Uriarte, R., Dopazo, J.: FatiGO. Available on: `http://fatigo.bioinfo.cnio.es/` (2003) Viewed at 23 October 2003.
12. Zeeberg, B.R., Feng, W., Wang, G., Wang, M.D., Fojo, A.T., Sunshine, M., Narasimhan, S., Kane, D.W., Reinhold, W.C., Lababidi, S., Bussey, K.J., Riss, J., Barrett, J., Weinstein, J.N.: GOMiner: A resource for biological interpretation of genomic and proteomic data. Genome Biology **4** (2003)
13. Zeeberg, B.R., Feng, W., Wang, G., Wang, M.D., Fojo, A.T., Sunshine, M., Narasimhan, S., Kane, D.W., Reinhold, W.C., Lababidi, S., Bussey, K.J., Riss, J., Barrett, J., Weinstein, J.N.: GOMiner. Available on: `http://discover.nci.nih.gov/gominer/` (2003) Viewed at 23 October 2003.
14. Hosack, D.A., Dennis Jr., G., Sherman, B.T., Lane, H., Lempicki, R.A.: EASE. Available on: `http://david.niaid.nih.gov/david/ease.htm` (2003) Viewed at 23 October 2003.
15. Hosack, D.A., Dennis Jr., G., Sherman, B.T., Lane, H., Lempicki, R.A.: Identifying biological themes within lists of genes with EASE. Genome Biology **4** (2003)
16. Han, J., Fu, Y.: Discovery of multiple–level association rules from large databases. In: Proceedings 1995 International Conference on Very Large Data Bases. (1995) 420–431
17. Srikant, R., Agrawal, R.: Mining generalized association rules. In: Proceedings 1995 International Conference on Very Large Data Bases. (1995) 407–419
18. Diehn, M., Sherlock, G., Binkley, G., Jin, H., Matese, J.C., Hernandez-Boussard, T., Rees, C.A., Cherry, J., Botstein, D., Brown, P.O., Alizadeh, A.A.: SOURCE: a unified genomic resource of functional annotations, ontologies, and gene expression data. Nucleic Acids Research **31** (2003) 219–223
19. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press, San Diego, USA (1999)
20. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Second edn. John Wiley and Sons, New York (2001)

Adaptive Mining Techniques for Data Streams using Algorithm Output Granularity

Mohamed Medhat Gaber[1], Shonali Krishnaswamy [1], Arkady Zaslavsky [1]

[1] School of Computer Science and Software Engineering, Monash University,
900 Dandenong Rd, Caulfield East, VIC3145, Australia
{Mohamed.Medhat.Gaber, Shonali.Krishnaswamy,
Arkady.Zaslavsky}@infotech.monash.edu.au

Abstract. Mining data streams is an emerging area of research given the potentially large number of business and scientific applications. A significant challenge in analyzing/mining data streams is the high data rate of the stream. In this paper, we propose a novel approach to cope with the high data rate of incoming data streams. We termed our approach "algorithm output granularity". It is a resource-aware approach that is adaptable to available memory, time constraints, and data stream rate. The approach is generic and applicable to clustering, classification and counting frequent items mining techniques. We have developed a data stream clustering algorithm based on the algorithm output granularity approach. We present this algorithm and discuss its implementation and empirical evaluation. The experiments show acceptable accuracy accompanied with run-time efficiency. They show that the proposed algorithm outperforms the K-means in terms of running time while preserving the accuracy that our algorithm can achieve.

1 Introduction

A data stream is a sequence of unbounded, real time data items with a very high data rate that can only read once by an application [2], [16], [17], [24], [25]. Data stream analysis has recently attracted attention in the research community. Algorithms for mining data streams and ongoing projects in business and scientific applications have been developed and discussed in [2], [13], [19]. Most of these algorithms focus on developing approximate one-pass techniques.

Two recent advancements motivate the need for data stream processing systems [16],[24]:

- The automatic generation of a highly detailed, high data rate sequence of data items in different scientific and business applications. For example: satellite, radar, and astronomical data streams for scientific applications, and stock market and transaction web log data streams for business applications.
- The need for complex analyses of these high-speed data streams such as clustering and outlier detection, classification, frequent itemsets and counting frequent items.

There are recent projects that stimulate the need for developing techniques that analyze high speed data streams in real time. These include:

- JPL/NASA are developing a project called Diamond Eye [5]. They aim to enable remote systems as well as scientists to analyze spatial objects in real time image stream. The project focuses on enabling "a new era of exploration using highly autonomous spacecraft, rovers, and sensors" [5].
- Kargupta et al. [19], [21] have developed MobiMine. It is a client/server PDA-based distributed data mining application for financial data streams.
- Kargupta et al. [20] have developed The Vehicle Data Stream Mining System (VEDAS) which is a ubiquitous data mining system that allows continuous monitoring and pattern extraction from data streams generated on-board a moving vehicle .
- Tanner et al. [30] are developing EnVironment for On-Board Processing (EVE). This system analyzes data streams continuously generated from measurements of different satellite on-board sensors using data mining, feature extraction, event detection and prediction techniques. Only interesting patterns are sent to the ground processing centre saving the limited bandwidth.
- Srivastava and Stroeve [29] are developing a NASA project for onboard detection of geophysical processes such as snow, ice and clouds using kernel clustering methods for data compression conserving the limited bandwidth needed to send streaming images to the ground centers.

These projects and others demonstrate the need for data stream analysis techniques and strategies that can cope with the high data rate and deliver the analysis results in real time in resource constrained environments.

There are two strategies for addressing the problem of the high speed nature of data streams. Input and output rate adaptation of the mining algorithm is the first strategy. The rate adaptation means controlling the input and output rate of the mining algorithm according to the available resources. The algorithm approximation by developing new light-weight techniques that have only one look at each data item is the second strategy. The main focus of mining data stream techniques proposed so far is the design of approximate mining algorithms that have only one-pass or less over the data stream. In this paper, we propose a novel approach that is able to mine data streams in one pass. Moreover, it is adaptable to memory, time constraints and data stream rate. We termed our approach as algorithm output granularity (AOG). This approach has the advantage of simplicity, generality and is an enhancement of the approximate algorithms research by being resource-aware. That means that the algorithm can adapt the output rate according to available resources.

The paper is organized as follows. Section 2 is a discussion on issues related to mining data streams and proposes our algorithm output granularity approach. One-pass mining techniques using our approach are proposed in section 3. The empirical studies for clustering data streams using algorithm output granularity are shown and discussed in section 4. Section 5 presents related work in mining data streams algorithms. Finally, we conclude the paper and present our future work in section 6.

2 Issues in Mining Data Streams

In this section, we present issues and challenges that arise in mining data streams and solutions that address these challenges. Fig. 1 shows the general processing model of mining data streams.

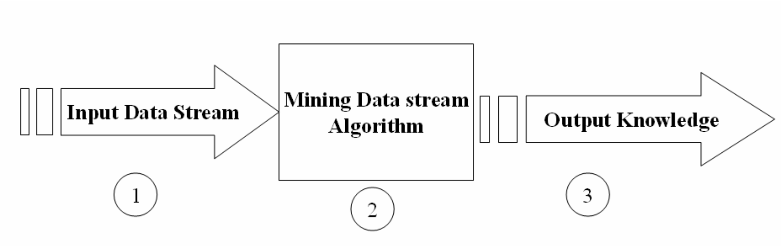


Fig. 1. Mining Data Stream Process

Issues and challenges with mining data streams:

1) Unbounded memory requirements due to the continuous feature of the incoming data elements.
2) Mining algorithms require several passes over data streams and this is not applicable because of the high data rate feature of the data stream.
3) Data streams generated from sensors and other wireless data sources create a real challenge to transfer these huge amounts of data elements to a central server to be analyzed.

There are several strategies that address these challenges. These include:

1) **Input data rate adaptation:** this approach uses sampling, filtering, aggregation, and load shedding on the incoming data elements. Sampling is the process of statistically selecting the elements of the incoming stream that would be analyzed. Filtering is the semantics sampling in which the data element is checked for its importance for example to be analyzed or not. Aggregation is the representation of number of elements in one aggregated elements using some statistical measure such as the average. While load shedding, which has been proposed in the context of querying data streams [3], [31], [32], [33] rather than mining data streams, is the process of eliminating a batch of subsequent elements from being analyzed rather than checking each element that is used in the sampling technique. Fig. 2 illustrates the idea of data rate adaptation from the input side using sampling.

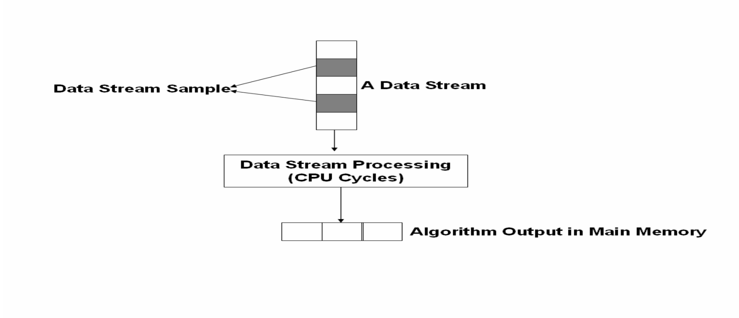


Fig. 2. Data Rate Adaptation using Sampling

2) **Output concept level:** using the higher concept level in applying data mining in order to cope with the data rate, that is to categorize the incoming elements into a limited number of categories and replacing each incoming element with the matching category according to a specified measure or a look-up table. This would produce fewer results conserving the limited memory. Moreover, it would require fewer number of processing CPU cycles.
3) **Approximate algorithms:** design one pass mining algorithms to approximate the mining results according to some acceptable error margin.
4) **On-board analysis**: To avoid transferring huge amounts of data, the data mining would be done at the data source location. For example, (VEDAS) project [20], (EVE) project [30] and Diamond Eye project [5]. This however assumes the availability of significant computational resources at the site of data stream generation.
5) **Algorithm output granularity:** This is our proposed solution approach. It uses a control parameter as a part of the algorithm logic to control the output rate of the algorithm according to the available memory, the remaining time to fill the available memory before incremental knowledge integration takes place and the data rate of the incoming stream. Fig. 3 shows the idea of our proposed approach.

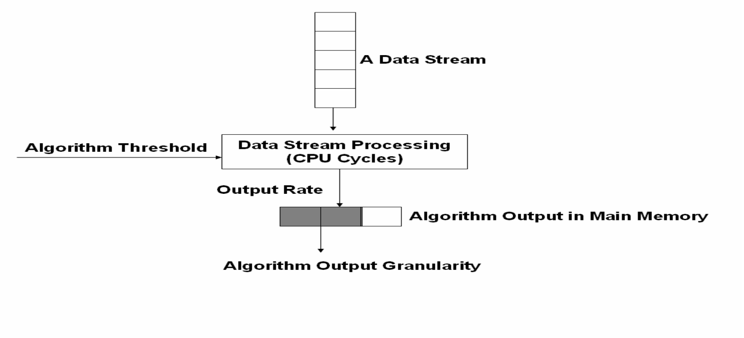


Fig. 3. Algorithm Output Granularity Approach

Algorithm output granularity:

To demonstrate our approach in mining data streams, we first define the following terms:

Algorithm threshold: is a controlling parameter built in the algorithm logic that encourages or discourages the creation of new outputs according to three factors that vary over temporal scale:

a) Available memory.
b) Remaining time to fill the available memory.
c) Data stream rate.

Output granularity: is the amount of generated results that are acceptable according to specified accuracy measure. This amount should be resident in memory before doing any incremental integration.

Time threshold: is the required time to generate the results before any incremental integration according to some accuracy measure. This time might be specified by the user or calculated adaptively based on the history of running the algorithm.

The main steps for mining data streams using our proposed approach:

1) Determine the time threshold and the algorithm output granularity.
2) According to the data rate, calculate the algorithm output rate and the algorithm threshold.
3) Mine the incoming stream using the calculated algorithm threshold.
4) Adjust the threshold after a time frame to adapt with the change in the data rate using linear regression.
5) Repeat the last two steps till the algorithm lasts the time interval threshold.
6) Perform knowledge integration of the results.

The following section will show the use of algorithm output granularity in clustering, classification and frequent items mining algorithms.

3 Algorithm Granularity based Mining Techniques

In the following subsections, we show the application of the algorithm output granularity to clustering, classification and frequent items.

3.1 LWC

In this section, our one-look clustering algorithm (LWC) is explained and discussed. The algorithm has two main components. The first one is the resource-aware RA component that uses the data adaptation techniques to catch up with the high-speed data stream and at the same time to achieve the optimum accuracy according to the available resources. The process starts by checking the minimum data rate that could be achieved using data adaptation techniques with an acceptable accuracy. If the algorithm can catch up with the minimum data rate, the RA component tries to find a solution that maximizes the accuracy by increasing the data rate. Otherwise the algorithm should send a data mining request to a data mining server that can achieve the minimum acceptable accuracy.

The other component is the LWC algorithm. The algorithm follows the following steps:

1- Data items arrive in sequence with a data rate.
2- The algorithm starts by considering the first point as a center.
3- Compare any new data item with the centers to find the distance.
4- If the distance for all the centers is greater than a threshold, the new item is considered as a new center; else increase the weight for the center that has the shortest distance between the data item and the center by 1 and let the new center equals the weighted average.
5- Repeat 3 and 4.
6- If the number of centers = k (according to the available memory) then create a new centers vector.
7- Repeat 3, 4, 5, and 6.
8- If memory is full then re -cluster (integrate clusters) and send to the server if needed.

The algorithm output granularity (k) is represented here by the number of cluster centers' kept in memory before doing any incremental re-clustering. The higher the algorithm granularity the higher is the algorithm accuracy. The threshold value here represents the minimum distance between any point and the cluster center. The lower the threshold the more the clusters is created.

Fig. 4 shows the pseudo code for this algorithm. The following is the notation used in the algorithm pseudo code.

Let *D* be the data rate in items/second.

Let *Max(D)* be unfiltered data rate in items/second.

Let *Min(D)* be filtered and aggregated data rate in items/second.

Let *AR* be algorithm rate: number of centers generated by the algorithm in centers/second.

Let *Dist* be the minimum distance between any point and the cluster center.

Let *M* be number of memory blocks, each block can store one center.

Let *T* be the time needed for generating a number of Centers that can fit all the memory blocks in seconds.

Let *TT* be the time threshold that is required for the algorithm accuracy in seconds.

```
1. x = 1, c=1, M = number of memory blocks available
2. Receive data item DI[x].
3. Center[c] = DI[x].
4. M = M -1
5. Repeat
     a. x = x+1
     b. Receive DI[x]
     c. For i = 1 to c
          Measure the distance between Center[i]
        and DI[x]
      d. If distance > dist (The threshold)
        Then
              c=c+1
              if (M <> 0)
              Then
                    Center[c] = DI[x]
              Else
                    Recluster DI[]
          Else
          For j=1 to c
        Compare between Center[j] and DI[x] to find
        the shortest distance.
        Increase the weight for the Center[j] with
        the shortest distance.
        Center[j] = (Center[j] * weight + DI[x]) /
```

Fig. 4. Light-Weight Clustering Algorithm

The algorithm according to the given threshold and the data set domain generates the maximum number of subsequent data items , each of which represents a center; that will be given using the following formula:

Maximum number of subsequent data points that could be centers = [(Maximum item value in the data set - Minimum item value in the data set) / threshold].

Since these points in the worst case might be the first points in the data stream in order for them to be centers, the following formula gives the number of data elements that would do the comparison over the generated centers:

Cluster Members = Data Set Size - [(Maximum item value in the data set - Minimum item value in the data set) / threshold].

Thus the algorithm complexity is *O(nm)*, where “n” is the data set size, and “m” is maximum number of subsequent data points that could be centers.

We have performed experimental evaluation and compared our algorithm with k-means. The results presented in Section 4 shows that our algorithm outperforms k-means in running time with an acceptable accuracy.

3.2 LWClass

In this section, we present the application of the algorithm output granularity to light weight K-Nearest-Neighbors classification LWClass. The algorithm starts with detemining the number of instances according to the available space in the main memory. When a new classified data element arrives, the algorithm searches for the nearest instance already in the main memory according to a pre-specified distance threshold. The threshold here represents the similarity measure acceptable by the algorithm to consider two or more elements as one element according to the element attributes' values. If the algorithm finds this element, it checks the class label. If the class label is the same, it increases the weight for this instance by one, otherwise it decrements the weight by one. If the weight becomes zero, this element will be released from the memory. The algorithm granularity here could be controlled by the distance threshold value and could be changing over time to cope with the high speed of the incoming data elements. The algorithm procedure could be described as follows:

1) Data streams arrive item by item. Each item contains attribute values for a1, a2, …,an attributes and the class category.
2) According to the data rate and the available memory, we apply the algorithm output granularity as follows:
    a) Measure the distance between the new item and the stored ones.
    b) If the distance is less than a threshold, store the average of these two items and increase the weight for this average as an item by 1. (The threshold value determines the algorithm accuracy and should be chosen according to the available memory and data rate that determines the algorithm rate).
    This is in case that both items have the same class category. If they have different class categories, we delete both).
    c) After a time threshold for the training, we come up with a sample result like the one in table 1.

Table 1. Sample LWClass Training Results

A1	A2	…	An	Class	Weight
Value(a1)	Value(a2)	…	Value(an)	Class category	X (represents that X items contribute in the values of this tuple)
Value(a1)	Value(a2)	…	Value(an)	Class category	Y
Value(a1)	Value(a2)	…	Value(an)	Class category	Z

3) Using the above table, we have some items that we need to classify them. According to the available time for the classification process, we choose nearest K-items and these items will be variable according to the time needed by the process.
4) Find the majority class category taking into account the calculated weights from the K items and this will be the answer for this classification task.

3.3 LWF

In this section, we present light-weight frequent items LWF algorithm. The algorithm starts by setting the number of frequent items that will be calculated according to the available memory. This number changes over time to cope with the high data rate. The main idea behind the algorithm is the algorithm output granularity. The AG is represented here by the number of frequent items that the algorithm can calculate as well as the number of counters that will be re-set after some time threshold to be able to cope with the continuous nature of the data stream. The algorithm receives the data elements one by one and tries to find a counter for any new item and increase the item for the registered items. If all the counters are occupied, any new item will be ignored and the counters will be decreased by one till the algorithm reaches some time threshold a number of the least frequent items will be ignored and their counters will be re-set to zero. If the new item is similar to one of the items in memory according to a similarity threshold, the average of both items will be allocated and the counter will be increased by one. The main parameters that can affect the algorithm accuracy are time threshold, number of calculated frequent items and number of items that will be ignored and their counter will be re-set after some time threshold. Fig. 5 shows the algorithm outline for the LWF algorithm.

```
1- Set the number of the top frequent items to k.
2- Set a counter for each k.
3- Repeat
     a. Receive the item.
     b. If the item is new and one of the k counters
        are 0
        Then
        Put this item and increase the counter by 1.
        Else
        If the item is already in one of the k
        counters.
        Then
        Increase the counter by 1.
        Else
        If the item is new and all the counters are
        full
        Then
        Check the time
        If time > Threshold Time
        Then
        Re-set number of least n of k counters to 0
        Put the new item and increase the counter by
        1
        Else
        Ignore the item.
```

Fig. 5. LWF Algorithm

4 Empirical studies for LWC

In this section, we discuss our empirical results for the LWC algorithm. The experiments were conducted using Matlab 6.0 in which the LWC is developed and the k-means algorithm included in the Matlab package is used as a guide to measure the algorithm accuracy. The experiments were conducted using a machine with Pentium 4 CPU 2.41 GHz, 480 MB of RAM, and running Windows XP Professional operation system.

There are three main parameters that we measure in our experiments; algorithm threshold, running time and accuracy. We have conducted a number of experiments to evaluate the algorithm.

Experiment 1: (Fig. 6)

Aim: Measure the algorithm running time with different threshold values.

Experiment Setup: Running LWC several times using different threshold values with a synthesized data set.

Results: The higher the threshold the lower the running time.

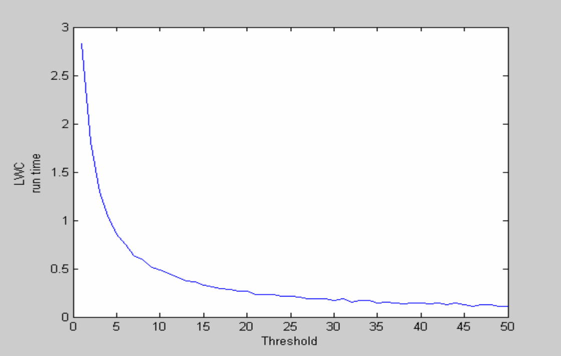


Fig. 6. LWC Running Time.

Analysis: We have to minimize the threshold according to the available resources of memory and CPU utilization. The threshold is an rate output adaptation technique. That is because the threshold value controls the algorithm rate (The higher the threshold the lower the algorithm rate). On the other hand, we can use the threshold as an application-oriented parameter that does not affect the accuracy; however it might increase it according to some domain knowledge about the clustering problem that might be known in advance.

Experiment 2: (Fig. 7)

Aim: Measuring the algorithm accuracy with different threshold values.

Experiment Setup: Running LWC and K-means several times with different threshold values. The experiment is repeated three times with different data set sizes.

Results: The lower the threshold the higher the accuracy of the algorithm which is measured as follows: *Accuracy (LWC) = average (|sorted LWC centers – sorted K-means centers|). The lower this measure will be, the higher the accuracy.*

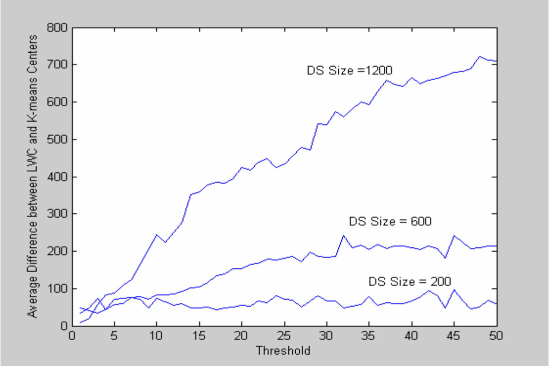


Fig. 7. LWC Accuracy (DS Size measured in number of data items).

Analysis: Choosing the threshold value is an important issue to achieve the required accuracy. It should be pointed out that from this experiment and the previous one the higher the accuracy the higher the running time. And that both factors are affected by the threshold value.

Experiment 3: (Fig. 8)

Aim: Comparison of K-means and LWC centers.

Experiment setup: Running LWC and K-means several times with the same threshold but different data set sizes.

Results: Assuming that the accuracy of K-means algorithm is high because it mines static data sets with any number of passes. The experiment shows that LWC generates similar centers that K-means algorithm generates .

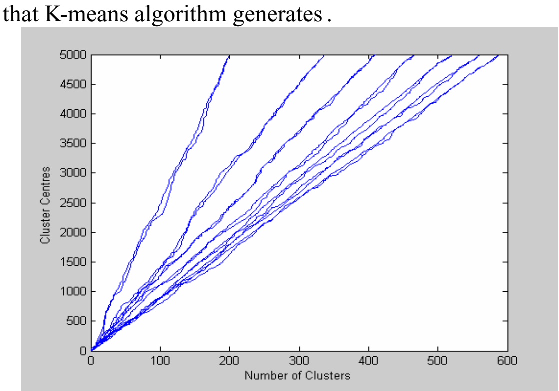


Fig. 8. LWC compared to K-means

Analysis: The accuracy of LWC is acceptable because it is very similar to k-means results that process the data set as static stored data set and not streaming data. That means that k-means algorithm performs several passes over the data set to result in the final cluster centers. As shown in the figure, the seven experiments show very similar cluster centers for our one-pass algorithm compared to k-means.

Experiment 4: (Fig. 9)

Aim: Measure the LWC algorithm running time against the data set sizes.

Experiment setup: Running the LWC algorithm with different large data sets.

Results: The algorithm has a linear relation with the data set size.

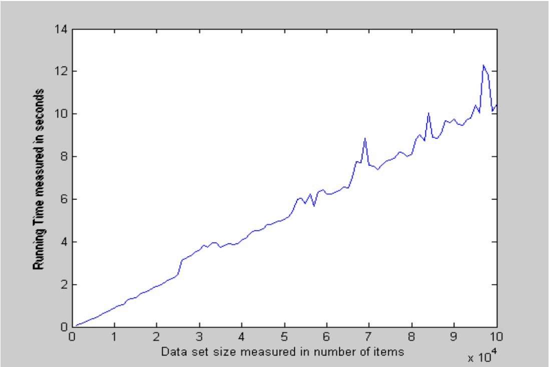


Fig. 9. LWC running time with different data set sizes

Analysis: the LWC algorithm is efficient for large data sets due to the linearity of the running time with data set size. This linearity results from performing only one-pass over the data stream. It is worth to point out here that the data stream rate is the major factor that control the behavior of LWC since the higher the rate the larger the size of the data set.

Experiment 5: (Fig. 10)

Aim: Measuring the effect of the threshold on the above experiment.

Experiment setup: Running LWC algorithm with the same data set sizes as the above experiment, but with decreasing threshold value with each run.

Results: The threshold value affects the running time of the algorithm since the maximum running time in the above experiment is approximately 12 seconds. The maximum running time in this experiment is about 47 seconds.

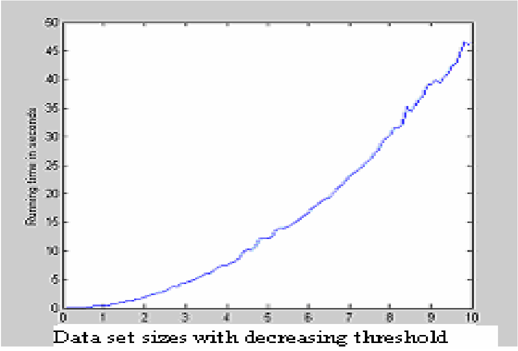


Fig. 10. LWC running time with different data set sizes and threshold values

Analysis: According to the application and/or the required accuracy, we have to maximize the threshold value to have more efficient algorithm in terms of running time.

The algorithm threshold would be controlled according to the available memory and a time threshold constraint that represents the algorithm accuracy.

Experiment 6: (Fig. 11)

Aim: Comparison between K-means and LWC efficiency.

Experiment setup: Running LWC (with a small threshold value which results in a high accuracy) and K-means several times on the same data sets with different sizes and measuring the running time.

Results: The running time of LWC is low compared to K-means with small data set sizes.

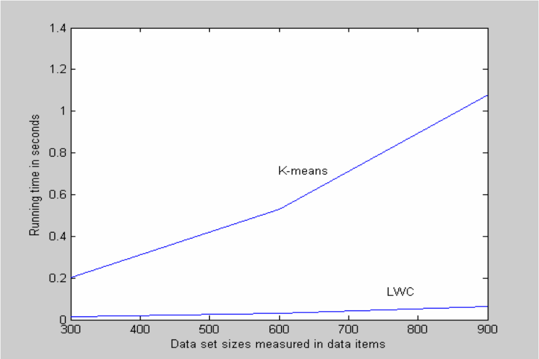


Fig. 11. K-means and LWC comparison in terms of running time

Analysis: LWC is efficient compared to K-means for small data sets, when we try to run both on large data sets; we found that LWC outperforms the K-means. The LWC runs with highest possible accuracy (the least threshold value) and outperforms k-means with different data set sizes.

The above experiments show an efficient one-look clustering algorithm that is adaptable to the available resources using our algorithmoutput granularity approach. The LWC outperforms k-means in terms of running time and has the advantage of linearity of running time with the increase in the data set sizes. The algorithm threshold is the controlling parameter of algorithm accuracy, efficiency, and algorithm output rate.

5 Related Work

There are different algorithms proposed to deal with the high speed nature for mining data streams using different techniques. Clustering data streams has been studied in [1], [4], [6], [7], [9], [10], [15], [22], [26]. Data stream classification has been studied in [11], [12], [18], [28], [34]. Extracting frequent items and frequent itemsets have been studied in [8], [14], [23].

The above algorithms deal with the problem of mining data streams using different methodologies. These algorithms basically focus on the design of approximate algorithms for mining data streams. However these approaches are not resource-aware and do not focus on adaptation strategies to cope with high data rates, our approach for output rate adaptation is resource-aware approach that can adapt to the available resources.

6 Conclusions and Future Work

In this paper, we discussed the problems of mining data streams and proposed possible solutions. Our algorithm output granularity approach in mining data streams has been presented and discussed. The proposed approach is distinguished from previous work in mining data streams by being resource-aware. We have developed a one-pass mining data streams algorithm. The application of the proposed approach to clustering, classification and counting frequent items has been presented. The implementation and empirical studies of our LWC algorithm have been demonstrated. The experiments showed an acceptable accuracy accompanied with efficiency in running time that outperforms k-means algorithm. Having implemented and tested LWC, we are developing LWClass and LWF. The application of these algorithms in a ubiquitous environment is planned for future work. The simplicity, generality, and efficiency of our proposed approach in mining data streams facilitate the application of the algorithms in various scientific and business applications that require data stream analysis.

References

1. Aggarwal C., Han J., Wang J., Yu P. S.: A Framework for Clustering Evolving Data Streams. Proc. 2003 Int. Conf. on Very Large Data Bases (VLDB'03), Berlin, Germany (2003).
2. Babcock B., Babu S., Datar M., Motwani R., and Widom J.: Models and issues in data stream systems. In Proceedings of PODS (2002).
3. Babcock B., Datar M., and Motwani R.: Load Shedding Techniques for Data Stream Systems (short paper). In Proc. of the 2003 Workshop on Management and Processing of Data Streams (MPDS 2003) (2003).
4. Babcock B., Datar M., Motwani R., O'Callaghan L.: Maintaining Variance and k-Medians over Data Stream Windows. To appear in Proceedings of the 22nd Symposium on Principles of Database Systems (PODS 2003) (2003).
5. Burl M., Fowlkes C., Roden J., Stechert A., and Mukhtar S. Diamond Eye: A distributed architecture for image data mining. In SPIE DMKD, Orlando, April (1999).
6. Charikar M., O'Callaghan L., and Panigrahy R.: Better streaming algorithms for clustering problems. In Proc. of 35th ACM Symposium on Theory of Computing (STOC) (2003).

7. O'Callaghan L., Mishra N., Meyerson A., Guha S., and Motwani R.: Streaming-data algorithms for high-quality clustering. Proceedings of IEEE International Conference on Data Engineering, March (2002).
8. Cormode G., Muthukrishnan S.: What's hot and what's not: tracking most frequent items dynamically. PODS 2003. (2003) 296-306
9. Datar M., Gionis A., Indyk P., Motwani R.: Maintaining Stream Statistics over Sliding Windows (Extended Abstract). In Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002) (2002).
10. Domingos P. and Hulten G., A General Method for Scaling Up Machine Learning Algorithms and its Application to Clustering. Proceedings of the Eighteenth International Conference on Machine Learning, 106--113, Williamstown, MA, Morgan Kaufmann. (2001)
11. Domingos P. and Hulten G. Mining High-Speed Data Streams. In Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining, (2000) 71—80.
12. Ganti V., Gehrke J., Ramakrishnan R.: Mining Data Streams under Block Evolution. SIGKDD Explorations 3(2): (2002) 1-10.
13. Garofalakis M., Gehrke J., Rastogi R.: Querying and mining data streams: you only get one look a tutorial. SIGMOD Conference 2002: 635. (2002).
14. Giannella C., Han J., Pei J., Yan X., and Yu P.S.: Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In Kargupta H., Joshi A., Sivakumar K., and Yesha Y. (eds.), Next Generation Data Mining, AAAI/MIT (2003).
15. Guha S., Mishra N., Motwani R., and O'Callaghan L.: Clustering data streams. In Proceedings of the Annual Symposium on Foundations of Computer Science. IEEE, November (2000).
16. Golab L. and Ozsu M. T. : Issues in Data Stream Management. In SIGMOD Record, Volume 32, Number 2, June (2003) 5-14.
17. Henzinger M., Raghavan P, and Rajagopalan S.: Computing on data streams. Technical Note 1998-011, Digital Systems Research Center, Palo Alto, CA, May (1998).
18. Hulten G., Spencer L., and Domingos P.: Mining Time-Changing Data Streams. ACM SIGKDD (2001).
19. Kargupta H.: CAREER: Ubiquitous Distributed Knowledge Discovery from Heterogeneous Data. NSF Information and Data Management (IDM) Workshop (2001).
20. Kargupta. H.: VEhicle DAta Stream Mining (VEDAS) Project. http://www.cs.umbc.edu/%7Ehillol/vedas.html. (2003).
21. Kargupta, H., Park, B., Pittie, S., Liu, L., Kushraj, D. and Sarkar, K. (2002). MobiMine: Monitoring the Stock Market from a PDA. ACM SIGKDD Explorations. January 2002. Volume 3, Issue 2. Pages 37--46. ACM Press.
22. Keogh E., Lin J., and Truppel W.: Clustering of Time Series Subsequences is Meaningless: Implications for Past and Future Research. In proceedings of the 3rd IEEE International Conference on Data Mining. Melbourne, FL. November (2003) 19-22.

23. Manku G. S. and Motwani R.: Approximate frequency counts over data streams. In Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, August (2002).
24. Muthukrishnan S.: Data streams: algorithms and applications. Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms (2003).
25. Muthukrishnan S: Seminar on Processing Massive Data Sets. Available Online: http://athos.rutgers.edu/%7Emuthu/stream-seminar.html (2003).
26. Ordonez C.: Clustering Binary Data Streams with K-means .ACM DMKD (2003).
27. Park B. and Kargupta H.. Distributed Data Mining: Algorithms, Systems, and Applications. Data Mining Handbook. Editor: Nong Ye (2002).
28. Papadimitriou S., Faloutsos C., and Brockwell A.: Adaptive, Hands-Off Stream Mining. 29th International Conference on Very Large Data Bases VLDB (2003).
29. Srivastava A. and Stroeve J.: Onboard Detection of Snow, Ice, Clouds and Other Geophysical Processes Using Kernel Methods. Proceedings of the ICML'03 workshop on Machine Learning Technologies for Autonomous Space Applications (2003).
30. Tanner S., Alshayeb M., Criswell E., Iyer M., McDowell A., McEniry M., Regner K., EVE: On-Board Process Planning and Execution, Earth Science Technology Conference, Pasadena, CA, Jun. 11 - 14, (2002).
31. Tatbul N., Cetintemel U., Zdonik S., Cherniack M. and Stonebraker M.: Load Shedding in a Data Stream Manager. Proceedings of the 29th International Conference on Very Large Data Bases (VLDB), September (2003).
32. Tatbul N., Cetintemel U., Zdonik S., Cherniack M. and Stonebraker M.: Load Shedding on Data Streams. In Proceedings of the Workshop on Management and Processing of Data Streams (MPDS 03), San Diego, CA, USA, June (2003).
33. Viglas S. D. and Naughton J.: Rate based query optimization for streaming information sources. In Proc. of SIGMOD (2002).
34. Wang H., Fan W., Yu P. and Han J.: Mining Concept-Drifting Data Streams using Ensemble Classifiers. In the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Aug., Washington DC, USA (2003).

An Analytical Approach for Handling Association Rule Mining Results

Rodrigo Salvador Monteiro[1], Geraldo Zimbrão[1,2], Jano Moreira de Souza[1,2]

[1] Computer Science Department, Graduate School of Engineering,
Federal University of Rio de Janeiro, PO Box 68511, ZIP code: 21945-970,
Rio de Janeiro, Brazil
[2] Computer Science Department, Institute of Mathematics,
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
{salvador, zimbrao, jano}@cos.ufrj.br

Abstract. Association rule mining has played an important role in data mining research since its introduction by Agrawal in [2]. An important question that has come up with this new technique is how to analyze the commonly huge number of rules produced by the algorithms. Many proposals have arisen, ranging from simple ideas, like grouping correlated rules up to more ambitious ones, such as the Inductive Databases [19]. Other works attempt to avoid this problem by imposing constraints and interestingness measures during the discovery of association rules in order to output only the significant rules. We present a new proposal that defines a framework for a Data Warehouse of Association Rules capable of providing a full environment for analyzing association rules. This new approach can benefit naturally from many data warehouse features and, besides, it comprises an interesting framework for proposing new knowledge discovery models.

1 Introduction

Advances in data gathering mechanisms, the use of bar codes in most commercial products, and information on many business and governmental transactions have been flooding us with data, creating an urgent need for new techniques and tools to intelligently and automatically aid in the transformation of this data into useful knowledge [8]. Data Mining, known as Knowledge Discovery in Databases – KDD, appeared as a group of methods, techniques and tools capable of meeting this demand.

One of the techniques most widely studied and explored for pattern discovery in large databases is association rule mining. It was presented and formalized by [2], and quickly gained importance for its large applicability. Since then, many works proposing new algorithms for association rule mining, such as [3,25,5,4,16,15,23,6,1,11,20,12], beside several others, affirm the importance and attention that has been dedicated to this data mining area.

Paradoxically, data mining itself can produce such great amounts of data that there has arisen a new knowledge management problem [17]. Association rule mining fits in this category and, therefore, it needs solutions for analysis and management of discovered knowledge. All solutions proposed so far to help the analysis of a huge volume of association rules have at least one of the following problems or restrictions: 1 – the analysis is restricted to valid rules at one specific moment in time, so it is not possible to confront valid rules in different spots of time; 2 – limited analysis capabilities, such as grouping criteria specification and specific interestingness measures; 3 – serious performance problems, and no feasible implementation with the currently-available technology.

As an alternative to avoid the problem, several works have been proposed to push constraints and interestingness measures into the kernel of association rule mining algorithms. However, the great problem concerning this solution is defining which constraints should be pushed and which rule characteristics are interesting. In other words, according to [22], were we to know what we were looking for, the discovery process would be trivial. Besides, by adopting this alternative, the analyst never gets an overview of the rules in the whole database.

The integration of OLAP operations with data mining tasks, proposed by [14], is a great advance in methods and analysis tools to support the analysis/evaluation of patterns step in the knowledge discovery process. However, when it comes to association rules, this integration does not take place naturally, according to discussions presented in the same work [14].

The proposal of this ongoing work consists of defining a data warehouse of association rules through structure, operation and model specifications in order to provide a complete environment for analyzing association rules. The present article defines the desired functionalities of a data warehouse of association rules. We argue that providing sophisticated tools to assist the analyst in the pattern evaluation/analysis step turns out to be as important as mining association rules. There is no use to mine rules efficiently if we are not able to understand and find them. This new approach can naturally benefit from many data warehouse features, and besides, it constitutes an interesting framework for proposing new knowledge discovery models.

The remainder of this work is organized as follows: section 2 presents the problem definition of mining association rules; section 3 presents the existing approaches for handling association rule mining results; section 4 presents our proposal, the Data Warehouse of Association Rules Framework (DWARF), and; finally, section 5 draws a few conclusions.

2 Association Rule Mining - ARM

Association rule mining is one of the most widely studied and explored techniques for Knowledge Discovery in Databases. Mining association rules mean searching for correlation patterns among facts recorded in one transactional database. Traditionally, the problem is presented through one classic application: the supermarket basket. A client’s supermarket basket consists of a set of items, such as rice, beans, meat, etc. Each purchase corresponds to one transaction. Discovering which supermarket items are frequently sold together is valuable information. As an example, we may discover that 70% of clients who buy sugar-free candy buy diet beverages too. The supermarket manager, owning this information, can redistribute the items on the shelves; he may also plan promotions, besides getting a better understanding of the business and of his clients' behavior. One actual example is Wal-Mart. By discovering an association between purchases of diapers and beers, Wal-Mart optimized their store layout in sales points, putting these items side by side. As a result, there was an increase of 30% in these product sales.

Agrawal in [2] was the first to present and formalize the problem of association rule mining. The formal definition from [2] follows below.

2.1 Formal Definition of Association Rule Mining Problem

Let $\Gamma = I_1, I_2, ..., I_m$ be a set of binary attributes, called items. Let T be a database of transactions. Each transaction t is represented as a binary vector, with $t[k] = 1$ if t bought item I_k, and $t[k] = 0$ otherwise. There is one tuple in the database for each transaction. Let X be a set of some items in Γ. We say that a transaction t satisfies X if, for all items I_k in X, $t[k] = 1$.

Association rules mean an implication of the form $X \rightarrow I_j$, where X is a set of some items in Γ, and I_j is a single item in Γ that is not present in X. The rule $X \rightarrow I_j$ is satisfied in the set of transactions T with the confidence factor $0 \leq c \leq 1$ iff at least c% of transactions in T that satisfy X also satisfy I_j. Besides, rule $X \rightarrow I_j$ has the support factor $0 \leq s \leq 1$, where s is the fraction of transactions in T that satisfy $X \cup I_j$.

Support should not be confused with confidence. While confidence is a measure of the rule's strength, support corresponds to statistical significance.

In this formulation, the problem of rule mining can be decomposed into two subproblems:

1. Generate all combinations of items that have fractional transaction support above a certain threshold, called *minsupport*. Call these combinations *large itemsets*, and all other combinations that do not meet the threshold *small itemsets*.

2. For a given *large itemset* $Y = I_1, I_2, ..., I_k$, $k \geq 2$, generate all rules (at most k rules) that use items from set $I_1, I_2, ..., I_k$. Only rules with confidence above a certain threshold are reported.

Computational complexity is concentrated on the first subproblem. Once we have all large itemsets available, the solution of the second subproblem is extremely simple. That is why it is common to see the association rule mining problem reduced to a problem of finding large itemsets.

3 Existing Approaches for Handling ARM Results

Different approaches were proposed for handling results of data mining algorithms. In this section we present the most relevant ones emphasizing their applicability to association rules.

3.1 OLAP Mining

Han in [14] defined the concept of OLAP Mining as a mechanism that integrates on-line analytical processing (OLAP) with data mining so that mining can be performed at different levels of abstraction at the user's fingertips.

The concept of OLAP Mining predicts a total integration, it thus being possible to mix OLAP operation with data mining tasks. This is an extremely interesting solution for analyzing data mining tasks results. We mean by OLAP operations the many possible ways of interaction with a data cube in multidimensional data analysis, such as drill-down, roll-up, pivot, slice, dice, etc. In order for such integration to take place, we should be able to express a data mining task execution result as a data cube. Considering the classification and clustering mining tasks, we achieve perfect integration. In such tasks we can associate the pre-existing classes or the newly discovered categories to a new dimension. The same does not happen when considering association rules. A new data cube cannot easily or naturally represent the result of an association rule mining algorithm execution on a data cube. In [14] this integration problem is underlined, some alternatives are discussed but however, no solution is presented. This fact prevents the execution of OLAP operations to analyze the rules discovered. In other words, the mechanism of OLAP Mining does not provide operation for association rule analysis.

3.2 Inductive Databases

Inductive Databases, defined by Mannila in [19], are databases which, in addition to data, also contain inductive generalizations about the data. This approach has a basic principle: many data mining tasks can be described as problems on how to find interesting sentences given a specific logic. Considering the association rule, we can translate it as the problem of finding valid rules on a database given support and confidence constraints. Once this concept is adopted, the analyst's task can be accomplished by simply querying the database theory using a conventional query language. [10] present a discussion on two alternatives: 1) mining association rules on demand as part of a query execution plan and; 2) discovery of all valid and interesting rules according to some criteria and storing them in the database. A hybrid approach is also considered.

Citing Goethals and Bussche [10], the *framework* of the Inductive Databases is an elegant formalization of the interactive mining process. However, we can identify some topics not satisfactorily solved by this framework. They are as follows:

1. Inexistence of sophisticated analysis operations (e.g.: OLAP operations) essential for evaluating huge amounts of data. This deficiency also prevents the analyst from obtaining an overview of the rules in the whole database;
2. Past association rules are not considered as there is no history on past association rules. The database theory corresponds only to the current status of the database.

Considering the mining rule on demand approach there is an additional problem: how to accomplish the cleaning step, which is essential to ensure data quality? Without a good treatment of input data, it is impossible to obtain good results with any data mining algorithms. Translated into the commonly-used sentence: garbage in, garbage out.

3.3 Interestingness Measures

Data mining algorithms, in special those of association rule mining, typically create a great number of patterns. However, many patterns are irrelevant or obvious, and bring no new knowledge [21]. Many measures have been proposed in the attempt to evaluate the utility and relevance of discovered patterns, which are referenced in the literature as interestingness measures. The use of an interestingness measure provides a ranking ordering the discovered patterns.

The great challenge for interestingness measures is to quantify what is interesting to the analyst. The problem becomes worse if we remember that analysts usually do not know for sure what they are looking for and, therefore, they do not know what is interesting. Thus, the analyst can be losing valuable information if we constrain the discovered patterns by some interestingness measure threshold. However, the use of interestingness measures is extremely useful to eliminate obvious patterns and to provide a ranking. It can be very useful when combined with other approaches.

Some examples of interestingness measures that can be used on association rule are: Itemset Measures [3], Rule Templates [17], Interestingness [24,13,7], Surprisingness [9], Reliable Exceptions [18] and Peculiarity [26].

3.4 Visualization Techniques

Some proposals on association rule visualization attempt to provide the analyst with an overview of the entire rule set. The visualization of a rule set corresponds to the problem of visualizing a hipergraph, with even the visualization of one single rule comprising a complex problem. [17].

A set of association rules can be represented through, for example, a dependence graph (Figure 1) or a bar chart (Figure 2), showing interestingness measures (e.g., support or confidence). In a bar chart, we have limits on the number of dimensions, restricting the visualization to rules with a few items (2 items, the third dimension represents an interestingness measure). In a dependence graph, we do not have limits on the number of rule items. However, we can perceive, observing Figure 1, that even sets of few rules can make the graph unreadable. Besides, the node disposition problem represents a great challenge.

4 Our Proposal

Even considering reasonable thresholds for support and confidence, one can obtain hundreds or thousands of association rules [17]. We argue that forcing other constraints to reduce the number of rules produced may not be a good solution, for the analyst must reduce the scope of his search artificially. There is no doubt that allowing the introduction of constraints is an interesting and useful feature to be used by the analyst.

However, we argue that this mechanism must not be used to constrain the rule's search in order to make the result tractable.

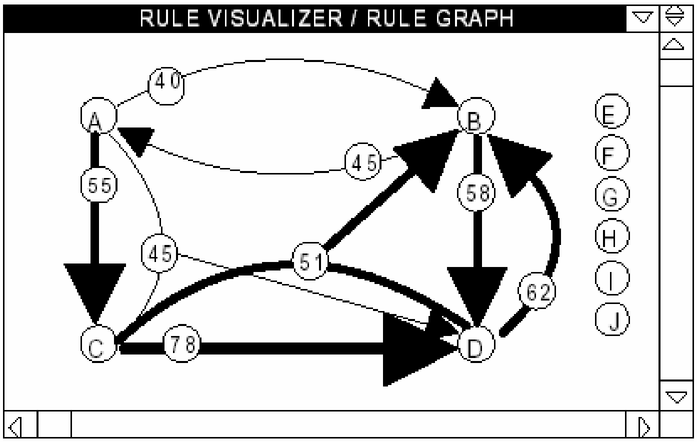
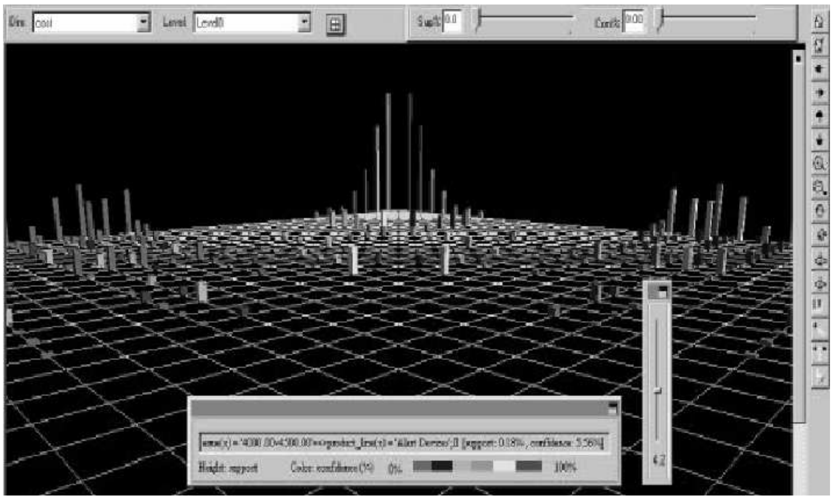


Fig. 1. Dependence graph in Rule Visualizer / Rule Graph [17]



area(x) = '4000.00-4300.00'=>product_line(x) = 'Alert Devices',[] [support: 0.18%, confidence: 5.56%]

Height: support Color: confidence (%) 0% 100%

Fig. 2. Bar Chart with interestingness measure in DBMiner Associator [14]

Facing the desirable functionalities and the limitation of the existing approaches, we developed the proposal for a new framework to supply the current deficiencies in the evaluation/analysis of association rules. In this new framework we attempt to maintain the advantages of the existing ideas and solve the problems with new thoughts, providing new analysis features. We call this new framework the Data Warehouse of Association Rules Framework – DWARF.

4.1 Data Warehouse of Association Rules Framework (DWARF)

Our proposal defines a framework in which the features provided by the OLAP Mining mechanism, beside others, can be applied to great numbers of association rules produced by mining algorithms. The definition of a Data Warehouse of Association Rules has the goal of providing a complete and friendly environment in which the analyst will have such powerful tools that he will not have to worry about the number of rules produced.

A Data Warehouse of Association Rules must have the following features:

1. A cube cell being analyzed corresponds to a set of association rules. It should be possible to visualize different aggregate values representing the set, from simple counting to more sophisticated interestingness measures. The rule set visualization must also be possible, varying from simple listing to more elaborate visualization techniques;
2. Allow for the association of interestingness measures, such as support and confidence, to DW dimensions. As an example, it must be possible to visualize the whole set of rules by support ranges or to constrain the rules to be considered to any specific range;
3. Have a Time dimension following the traditional DW approach. Any restriction in this dimension must have the effect of considering only valid rules on the constrained period being analyzed;
4. Have two Item dimensions, one associated to rule antecedent and the other associated to rule consequent. These dimensions can have hierarchies allowing for several organizations and grouping of items;
5. Allow the execution of traditional OLAP operations, such as drill-down, roll-up, pivot, slice, dice, etc, on the rule cube being analyzed;
6. Other dimensions, such as spatial dimension, allowing for the restriction of rules valid in a specific region.

The data necessary to the Data Warehouse of Association Rules operations must be available in some previously-loaded structure just as the traditional Data Warehouse.

Besides the analysis power provided by this new approach, we believe that new knowledge-discovering models can be proposed on the rules stored in the DW. A preliminary idea is presented on the next section.

4.2 New Knowledge-Discovering Model Proposal

We can think of identifying changes in a business once we have available, at DWARF, sets of association rules of different points of time. Using, for example, measures of rule similarity, we can identify similarities and differences between two rule sets. Imagine the following scenario: in one month, an analyst recognizes some management decision that should be taken based on the analysis of rules. In the next month it can be interesting to check whether the desired changes have occurred and, mainly, whether unexpected changes have taken place as a side effect of the decision taken.

5 Conclusions

Analyzing great number of association rules is a problem that came up immediately after the definition of the first association rule mining algorithm. Since then, several alternative proposals and solutions have been developed. However, all current approaches have weak points. In this work, we present the Data Warehouse of Association Rules Framework (DWARF) proposal, which aims at maintaining the advantages of the existing ideas and solving the problems with new ones, providing new analysis features. We defined the DWARF by listing the set of features it must provide.

A great advantage of the DWARF approach is to provide analysis tools that are familiar to the analysts who are used to OLAP tools. Thus, it is extremely easy to be used by managers and decision-makers in a company.

It is important to note that the definition of DWARF provided in this work is part of an ongoing project in which structures capable of providing the desired features in an efficient way will be researched and investigated. We strongly believe that we are in the right path, since other works, such as [22], point to the adaptation algorithms and development of structures for mining data in secondary and even tertiary storage.

We strongly believe that the consolidation of the DWARF proposal will provide an environment for association rule analysis capable of relieving the analyst from worrying about the number of rules produced, there by leaving him free to deal only with his investigation.

References

1. Agarwal, R. C., Aggarwal, C. C., and Prasad, V. V. V.: “A Tree Projection Algorithm for Generation of Frequent Item Sets”. Journal of Parallel and Distributed Computing 61(3): 350-371, 2001.
2. Agrawal, R., Imielinski, T., and Swami, A.: “Mining Association Rules between Sets of Items in Large Databases”. Proceedings of the 1993 ACM SIGMOD Conf., pages 207-216, Washington, DC, May 1993.
3. Agrawal, R., and Srikant, R.: “Fast Algorithms for Mining Association Rules in Large Databases”. Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487-499, Santiago de Chile, Chile, 1994.
4. Bayardo Jr, R. J.: “Efficiently Mining Long Patterns from Databases”. Proceedings of the 1998 SIGMOD Conference, pp. 85-93, 1998.
5. Brin, S., Motwani, R., Ullman, J. D., and Tsur, S.: “Dynamic Itemset Counting and Implication Rules for Market Basket Data”. Proceeding of the 1997 SIGMOD Conference, pp. 255-264, 1997.
6. Burdick, D., Calimlim, M., and Gehrke, J. E.: “MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases”. Proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, April 2001.
7. Dong, G., and Li, J.: “Interestingness of Discovered Association Rules in Terms of Neighborhood-Based Unexpectedness”. In Proc. of Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining, pp. 72-86, Melbourne, Australia, 1998.
8. Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R.: “Advances in Knowledge Discovery and Data Mining”. AAAI Press, 1998.

9. Freitas, A. A.: “On Objective Measures of Rule Surprisingness”. In Proc. of the Second European Symposium on Principles of Data Mining and Knowledge Discovery, pp. 1-9, Nantes, France, 1998.

10. Goethals, B., and Bussche, J.: “A priori versus a posteriori filtering of association rules.” 1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 1999.

11. Gouda, K., and Zaki, M. J.: “Efficiently Mining Maximal Frequent Itemsets”. Proceeding of the 2001 ICDM Conference, pp. 163-170, 2001.

12. Grahne, G., and Zhu, J.: “High Performance Mining of Maximal Frequent Itemsets”. Proceedings of the 6th International Workshop on High Performance Data Mining, 2003.

13. Gray, B., and Orlowska, M. E.: “CCAIIA: Clustering Categorial Attributed into Interseting Accociation Rules”. In Proc. of Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining, pp. 132-143, Melbourne, Australia, 1998.

14. Han, J.: “OLAP Mining: An Integration of OLAP with Data Mining”. Proceedings of the 1997 IFIP Conference on Data Semantics (DS-7), pp. 1-11, Leysin, Switzerland, Oct 1997.

15. Han, J., Pei, J., and Yin, Y.: “Mining frequent patterns without candidate generation”. In Proceeding of the 2000 SIGMOD Conference, pp. 1-12, Dallas, Texas, May 2000.

16. Hidber, C.: “Online Association Rule Mining”. Proceedings of the ACM SIGMOD International Conference on Management of Data, 1999.

17. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I.: “Finding Interesting Rules from Large Sets of Discovered Association Rules”. Proceeding of the Third International Conference on Information and Knowledge Management, pp. 401-407, 1994.

18. Liu, H., Lu, H., Feng, L., and Hussain, F.: “Efficient Search of Reliable Exceptions”. In Proc. of Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, pp. 194-203, Beijing, China, 1999.

19. Mannila, H.: “Inductive Databases and Condensed Representations for Data Mining”. Proceedings of the 1997 International Logic Programming Symposium, pp. 21-30, 1997.

20. Orlando, S., Palmerini, P., Perego, R., and Silvestri, F.: “Adaptive and Resource-Aware Mining of Frequent Sets”. Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 338-345. Maebashi City, Japan, December 2002.

21. Padmanabhan, B., and Tuzhilin, A.: “A Belief-Driven Method for Discovering Unexpected Patterns”. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 94-100, New York, 1998.

22. Ramakrishnan, N., and Grama, A. Y.: “Data Mining: From Serendipity to Science”. IEEE Computer, 32(8): 34-37, 1999.

23. Seno, M., and Karypis, G.: “LPMiner: An Algorithm for Finding Frequent Itemsets Using Length-Decreasing Support Constraint”. In Proceeding of the 2001 IEEE International Conference on Data Mining, pp. 505-512, San Jose, California, 2001.

24. Silberschatz, A., and Tuzhilin, A.: “On Subjective Measures of Interestingness in Knowledge Discovery”. Proceedings of the First International Conference on Knowledge Discovery and Data Mining, pp. 275-281, Montreal, Canada, 1995.

25. Srikant, R., and Agrawal, R.: “Mining Quantitative Association Rules in Large Relational Tables”. Proc. of 1996 SIGMOD Conference on Management of Data, pages 1-12, Montreal, Quebec, June 1996.

26. Zhong, N., Yao, Y. Y., and Ohsuga, S.: “Peculiarity Oriented Multi-database Mining”. In Proc. of Third European Conference on Principles of Data Mining and Knowledge Discovery, pp. 136-146, Prague, Czech Republic, 1999.

Modelling Insurance Risk: A Comparison of Data Mining and Logistic Regression Approaches

Inna Kolyshkina[1], Peter Petocz[2], Ingrid Rylander[3]

Abstract

Interest in Data Mining techniques has been increasing recently amongst the statistical profession. This paper presents and discusses a case study showing the application of one of the better-known Data Mining techniques, Classification and Regression Trees (CART®), to a business problem that required modelling risk in insurance, based on a project performed for an insurance company by PWC Actuarial. The nonlinear and nonparametric approach on which CART methodology is based provides good insights into the hidden patterns in such large data sets, with maybe a few million cases and several hundreds of possible predictor variables. Such data sets are common in many areas of insurance, healthcare, telecommunications, credit risk, banking, etc. The paper discusses the use of CART methodology and introduces some innovative model performance measures often used in data mining, such as gain or lift. The results of CART modelling are compared to the results achievable by using more traditional, linear and generalised linear modelling techniques such as logistic regression. Comparisons are made in terms of time taken, predictive power, selection of most important predictors out of large number of possible predictors, handling predictors with many categories (such as postcode or occupation code), interpretability of the model, dealing with missing values, etc. The more practical and non-statistical issues of implementation and client feedback are also discussed.

1 Introduction

[1] PricewaterhouseCoopers, GPO Box 2650, SYDNEY NSW 1171, Australia
inna.kolyshkina@au.pwc.com

[2] Department of Mathematical Sciences, University of Technology, Sydney
peter.petocz@uts.edu.au

[3] PricewaterhouseCoopers, GPO Box 2650, SYDNEY NSW 1171, Australia
Ingrid.rylander@au.pwc.com

Interest in Data Mining techniques such as decision trees in the actuarial community has been increasing.

The main reasons for the increasing attractiveness of decision trees are as follows:

- It overcomes the shortcomings of linear methods that operate under the assumption that data is distributed either normally (as is the case in linear regression), or according to another distribution in the exponential family, such as binomial, Poisson, Gamma, inverse Gaussian etc (as it is required for a generalised linear model), which is often not quite the case. Decision trees are less affected by the distributional assumptions.
- It relies more heavily on “brutal force” of computing power than traditional models do, and because of that is less time-consuming and more precise than classical methods. When analyzing a large data set, say with 1 million cases and several hundred potential predictors, traditional approach would require significantly more time and will have difficulties with selecting the important predictors
- Classical methods often have a hard time dealing with categorical variables with a large number of categories (for example, claimant’s occupation code, industry code or postcode), which means that such variables are either left out from the model, or have to be grouped by hand prior to including them in the model).

These problems make it difficult to use linear methods when analysing large sets of data with a mix of all kinds of categorical and numeric variables.

This article gives an example of the application of CART® (Classification and Regression Trees; the name is a registered trade mark, but this has not been indicated throughout) to modelling risk in insurance. This nonlinear, nonparametric approach may provide greater insight into the hidden patterns in the data.

2 Problem and background

In workers’ compensation insurance, serious claims (for example, claims that were litigated or claims where the claimant stayed off work for a very long period of time etc), comprise only about 15% of all claims by number, but create some 90% of the incurred cost. This means that in order to reduce the cost in a maximally effective way, an insurer would need to concentrate its attention on such claims. From a practical point of view, the insurer must ensure that the management of claimant injuries is carried out in such a way that the injured person receives the most effective medical treatment at appropriate points in time to prevent his or her injury from becoming chronic and to enable the claimant to return to work in an optimal manner.

To do so, the insurer ideally would need to know, at the time of a claim being received, whether the claim is likely to become serious. But in most cases this is not obvious as there are many factors contributing to the result. Therefore, it would be useful to have a model that would account for all such factors and would be able to predict at the outset of a claim the likelihood of this claim becoming serious.

3 The Data and the Analysis

The data available for modelling was represented by several years' worth of information about a large number of claims from the NSW workers' compensation scheme.

The data available contained information:

– about the claim itself (such as date the claim was registered, policy renewal year of the claim, date when the claim was closed, date of the claim reopening, whether the claim was litigated, various liability estimates, payments made on the claim, reporting delay etc.
– about the claimant. Data on claimant included demographic characteristics of the client such as sex, age at the time of injury, family situation and whether the claimant had dependants, claimant's cultural background). Also there were variables related to the claimant's occupation, type of employment and work duties such as code for industry and occupation, nature of employment (permanent, casual, part or fulltime), wages etc.
– about the injury or disease such as time and place of injury, injury type, body location of the injury, cause or mechanism of injury, nature of injury etc.

Overall there were about 100 variables that might have been considered as potential predictors, most of them categorical with many categories. For example, the variable "occupation code" had more than 250 categories, "injury location code" had more than 80 categories.

The two major purposes of the analysis were:

– to identify the most important predictors from a large number of available variables containing information about a claim at the time when the claim is registered; and
– to build a model based on such predictors, which would classify a claim as "likely to become serious" or "not likely to become serious".

4 Traditional Statistical Modelling

Our first step was to attempt building the model using logistic regression, the traditional statistical modelling approach for analysis of data with binary response. Logistic regression is a well-known classical technique and is easily implemented in SAS, the software package that is mainly used for statistical analysis within our practice as well as by the client, and is a familiar and reliable data analysis tool. We built logistic regression models by using SAS v8 PROC LOGISTIC and PROC GENMOD.

The major difficulty that we encountered in using logistic regression was the fact that most variables in the data (such as location of injury, claimant's occupation code, industry code and other variables) were categorical with large numbers of categories. This caused a considerable increase in time required for computation, and even more importantly, a high level of sparseness, potentially leading to instability in the model estimates. Although PROC LOGISTIC in version 8 of SAS does have a feature that can handle sparseness (see SAS, 2002), we found that using this feature was time-consuming (it took us 1hr 56 min to run logistic regression with this feature on a sample of 10,000, so we decided that, considering that the model will need to be refined and rerun several times, it was too time-consuming to use this feature on a larger sample).

To overcome this problem and to be able to use in the model the information contained in such potentially important high-level categorical predictors as, for example, location of injury or occupation code, we had to transform these predictors. We grouped the categories “by hand”, according to recommendations of health management specialists as well as actuaries who have a lot of experience with insurance data, into a smaller set of broader categories.

Selection of the most important predictors and especially predictor interactions was another difficult task. Logistic regression theoretically can select the set of best predictors by using the stepwise method, but this process might take too long if the number of potential predictors is high (above 100 in our example). After several attempts to select the predictors this way ended in the computer crashing, we decided to take the top 30 predictors chosen by CART and allow stepwise logistic regression to further refine the selection. The use of PROC LOGISTIC with a main effects model further identified about 20 significant predictors.

Looking for predictor interactions using logistic regression proved to be time-consuming and again caused sparseness problem because the interaction of two categorical predictors even after we manually reduced the number of categories, has many categories (product of numbers of categories for the both predictors involved)

In Table 1 probability level =*p* means that the predicted value for a case is 1 if probability predicted by the logistic regression for the case is greater than *p* and is 0 otherwise, sensitivity is the proportion of true positives correctly identified by the model for a specified probability level and specificity is the proportion of true negatives correctly identified by the model for a specified probability level. Table 1 presents classification results for a few probability levels.

Table 1. Classification results from the logistic regression modeling

Classification Table									
	Correct		Incorrect		Percentages				
Prob Level	Event	Non-Event	Event	Non-Event	Correct	Sensitivity	Specificity	False POS	False NEG
0.06	6,199	19,514	22,111	454	53.3	93.2	46.9	78.1	2.3
0.10	5,354	29,529	12,096	1,299	72.3	80.5	70.9	69.3	4.2
0.12	4,943	32,585	9,040	1,710	77.7	74.3	78.3	64.6	5.0
0.14	4,642	34,473	7,152	2,011	81.0	69.8	82.8	60.6	5.5
0.16	4,419	35,644	5,981	2,234	83.0	66.4	85.6	57.5	5.9
0.18	4,242	36,410	5,215	2,411	84.2	63.8	87.5	55.1	6.2
0.20	4,128	36,913	4,712	2,525	85.0	62.0	88.7	53.3	6.4
0.22	4,028	37,275	4,350	2,625	85.6	60.5	89.5	51.9	6.6
0.24	3,905	37,595	4,030	2,748	86.0	58.7	90.3	50.8	6.8
0.26	3,775	37,893	3,732	2,878	86.3	56.7	91.0	49.7	7.1
0.28	3,644	38,185	3,440	3,009	86.6	54.8	91.7	48.6	7.3
0.30	3,508	38,470	3,155	3,145	87.0	52.7	92.4	47.4	7.6
0.40	2,700	39,751	1,874	3,953	87.9	40.6	95.5	41.0	9.0
0.50	2,002	40,608	1,017	4,651	88.3	30.1	97.6	33.7	10.3
0.60	1,485	41,015	610	5,168	88.0	22.3	98.5	29.1	11.2
0.70	955	41,321	304	5,698	87.6	14.4	99.3	24.1	12.1
0.80	629	41,462	163	6,024	87.2	9.5	99.6	20.6	12.7

Table 1 suggests that probability level that is providing the best balance of sensitivity and specificity for logistic regression is 0.18 and this is the probability level that is used for comparison of the confusion matrices for logistic regression and CART.

5 Using CART for Modelling

We then tried a Classification and Regression Trees approach. The CART methodology is technically known as binary recursive partitioning (see Hastie, Tibshirani and Friedman, 2001).

CART offers 2 main tools of model evaluation: gains chart and two classification tables also called confusion matrices: one for the learn sample and one for the test sample.

These tools allow us to appreciate three aspects of the model: conduct specificity and sensitivity analysis (from the classification tables), model stability (by comparing classification tables for the test and learn samples, Table 2), and how well model performs the ranking of the cases (by examining the gains chart, Figure 1). Both methods are easy to interpret and to explain to a client.

The model was built using both classification methods (Gini and Twoing) that CART offers as suitable for a binary response. Models resulting from using these two methods separately gave very similar results, additionally confirming the stability of the model.

Table 2. Misclassification tables for the learn and test data

Misclassification For Learn Data				
Class	N Cases	N Misclassed	Percentage Error	Cost
Serious	16,922	3,891	22.99	0.23
Non-Serious	105,358	25,744	24.43	0.24

Misclassification For Test Data				
Class	N Cases	N Misclassed	Percentage Error	Cost
Non-Serious	52,866	12,923	24.44	0.24
Serious	8,558	2,275	26.58	0.27

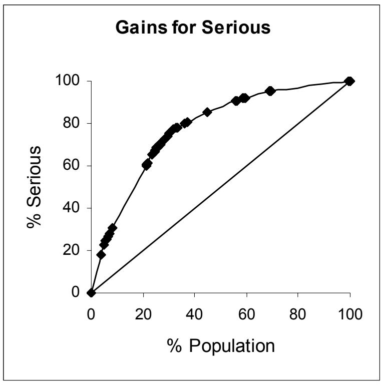


Figure 1. Gains chart for the CART model

6 Comparison of Traditional and Data mining Approaches

We identified the following differences between the two approaches:

6.1 Computational speed and time requirements

CART models were quicker to build (and therefore easier to refine). For example, running a CART model on data with some 190,000 cases and 25 predictors, some of which had more than 100 categories, took 9 minutes while running a stepwise logistic regression on a 50,000 random sample of the data with reduced numbers of categories for all categorical predictors (maximum 40 categories for a predictor) required reducing the number of categories by hand which involved consultation with health-management and actuarial experts, of writing a SAS program that performs the suggested recoding and running the program, taking a random sample of the data and then running the regression on the sample. This took two to three days and involved the time of the data analyst, actuarial consultant and health management experts.

6.2 Significant predictor selection

CART quickly selected around 20 significant predictors out of over 100. Logistic regression implemented using PROC LOGISTIC in SAS can select “best” predictors, but the process can take a long time. We decided to take the top 30 predictors chosen by CART and allow stepwise logistic regression to further refine the selection.

Theoretically, PROC LOGISTIC in stepwise regression will not necessarily find the right model, even given enough time. The trouble with stepwise methods implemented in statistical packages is that they do not allow for functional form modification or interactions, both of which are crucial to getting the right model. It is fairer to say that stepwise statistical methods cannot be used to learn the correct functional form and cannot identify the needed interactions. The only question is: can they find the best linear model? Even there, the search parameters (forwards, backwards) will affect whether this is true Only ‘all subsets’ can find the best linear approximation – but that approximation may be a terrible model (Steinberg, 2002).

It is worth mentioning here, that if our response variable had been continuous rather than binary (if, for example we needed to predict the cost of treatment), the process of predictor selection would have been even more time-consuming, because we would have needed to build a generalized linear model and use PROC GENMOD rather than PROC LOGISTIC, and PROC GENMOD does not perform predictor selection. And if our target variable had had more than two non-ordinal categories (it was suggested at one stage to use three categories: “litigated claim”, “non-litigated claim with claimant staying on benefits longer that a month” and “other”), we would not have been able to use either PROC GENMOD or PROC LOGISTIC to fit the model. Both of these limitations would be irrelevant for CART. This demonstrates that CART has better and more usable built-in variable selection methods than are offered by widely used stepwise methods.

6.3 Handling categorical predictors with many categories

Logistic regression could not handle categorical predictors with many categories (as it caused sparsity problem) so we needed to spend time on reducing the number of categories as was discussed above. CART® did not require such time investment as it can deal with categorical predictors with many categories more effectively.

6.4 Picking up interactions of predictors

It was not easy to check the significance of interactions of predictors using logistic regression. Such selection would have to either be “by hand” and therefore would include checking the significance of some interactions chosen based on advice of the experts who are familiar with the data (which could leave out some potentially interesting interactions) or would involve checking all possible interactions which can be very time-consuming.

Another issue with interaction selection is that if the predictors in whose interactions we are interested (for example the claimant’s occupation code and mechanism of injury) have relatively many categories even after reduction of the number of categories, it might again cause the sparsity problem. For example if after reduction of the number of categories the predictors have respectively 9 and 10 categories, their interaction will have 90 categories. This issue will then need to be solved using time-consuming methods as described above.

It is well known that in traditional approach higher-order (even three-way) interactions are usually hard to interpret. CART, on the other hand, because of the nature of its modelling approach, easily picks high-order interactions and the structure of the tree model makes it easy to interpret them.

There are however other ways than stepwise methods to look for interactions such as treating the various levels of interaction as a hierarchy, and bring whole levels in at a time. These ways have not been investigated in this paper.

6.5 Missing values

PROC LOGISTIC automatically excludes from the analysis all observations with any missing values for the explanatory variables. As we were working with a “real-life” data, it was reasonable to expect (as it was the case) that at least in 5% of cases there would be a missing value for one of the 30 predictors. This meant excluding from the analysis about 5% of all observations which was not desirable. CART did not have a problem with handling the missing values and did not require excluding any observations from the analysis.

6.6 Interpretability of the model

CART model is represented in the form of a diagram and so was very visual and easy to interpret and to explain to the client. Logistic regression model is expressed as a formula involving mathematical notation and can be easier interpreted by a client with some previous experience with logistic regression (or even simple regression) analysis.

6.7 Individual scoring vs segmenting

CART segmented all cases into several groups that were homogeneous in terms of likelihood of a serious claim (all cases within such group were assigned the same estimate of serious claim probability), while logistic regression gave each case an individual estimate of the probability of a serious claim.

6.8 Model evaluation

CART offers 2 main tools of model evaluation: gains chart and two classification tables: one for the learn sample and one for the test sample.

These tools allow us to appreciate three aspects of the model: conduct specificity and sensitivity analysis (from the classification tables), model stability (by comparing classification tables for the test and learn samples),

and how well model performs the ranking of the cases (by examining the gains chart). Both methods are easy to interpret and to explain to a client.

Logistic regression offers a classification table, as well as traditional, standard linear model diagnostics such as model fit indicators, residual diagnostics etc. These diagnostics tools are familiar to most data analysts and allow them to make a judgment about overall model fit as well as analyse outliers, influential observations and other aspects of the model.

6.9 Significance issues for predictors in a parametric linear model built on a data set with large number of cases

Parametric linear models built on data sets with large number of cases, usually have problems with identifying significant predictors. This happens because the number of the degrees of freedom for error are large, which results in practically all model terms being declared by the tests as significant. In other words, hypothesis testing using chi-squared test (which is the usual method used in all parametric linear models), does not work that well for large data sets. To some degree this problem can be overcome by building models on smaller random samples of the data and comparing them. This does mean that time needs to be spent on creating the random samples and rerunning models on them.

6.10 Precision

Comparison of classification tables will help to appreciate the accuracy of prediction level. We also created a gains chart based on the logistic regression prediction results and compared it with the gains chart for the CART model (Figure 2).

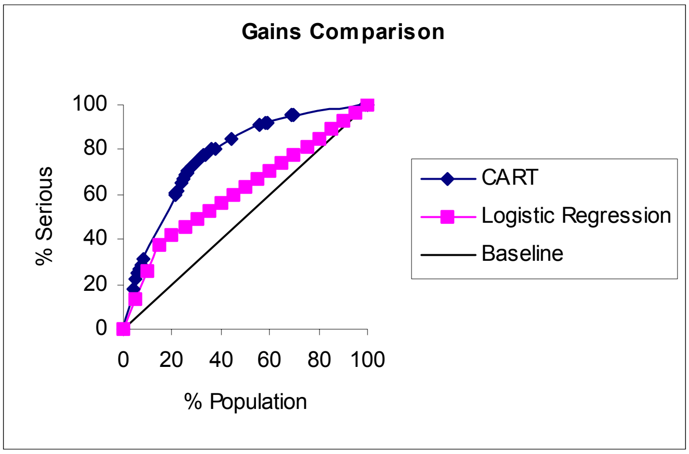


Figure 2. Comparison of gains with CART and Logistic Regression models

7 Findings and Results

The two results of the modelling that were most important for us were:

7.1 Selection of important predictors

The CART model quickly selected around 20 out of some 100 variables that could be regarded as potential predictors. It is interesting to note that some of the variables that turned out to be significant predictors were expected to be so on the basis of previous experience and analysis, for example, injury details (nature, location and mechanism of injury), while some others such as language skills of the claimant, were unexpected.

7.2 Prediction of serious claims

The CART model classifies correctly about 80% of all serious claims. The model targets 30% of all claims as “likely to be serious”. Of these about half turn out to be serious.

8 Implementation Issues and Client Feedback

The model has been incorporated into the insurer's computer system and automatically generates a "priority score" for each claim. This score is used to determine the level of staff who will be allocated the claim. Claims staff are grouped into three categories based on their experience in managing claims.

The scoring of the claim is however not only done initially, but reviewed at set points in time, and when important information comes to hand. Since all the data fields required for generating a score are captured as part of the claims database, it is easy for staff to get information about the change of score. Workflow processes have been designed to deal with those claims which change in likeliness to be serious.

We are implementing a continuous evaluation and feedback process to improve the model. A report on the results of this monitoring will be developed in due course.

9 Summary and Future Directions

In the insurance application discussed in this paper, for the analysis of large data sets, the CART methodology proved superior to the 'classical' methodology of logistic regression. Moreover, there is reason to believe that this is not an isolated example (see Salford Systems, 2002). There are situations where the logistic regression methods might be preferred (for instance when the number of predictors is relatively small, most of them are numeric rather than categorical, and the assumptions for logistic regression are clearly valid), but hybrid models using the strengths of both approaches are already being developed and applied (Steinberg and Cardell, 1998a and b). For future improvement of our model, we plan to investigate and apply such hybrid models.

References

1. Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Pacific Grove, CA.
2. Fisher, N. I. (2001). Crucial issues for statistics in the next two decades. *International Statistical Review*, 69(1), 3–4.
3. Friedman, J. H. (2001). The role of statistics in the data revolution. *International Statistical Review*, 69(1), 5–10.
4. Francis, L. (2001) .Neural Networks Demystified. In Casualty Actuarial Society Forum Winter 2001, 252-319.
5. Haberman, S. and Renshaw, A. E. (1998). Actuarial applications of generalized linear models. In Hand, D. J. and Jacka, S. D. (eds). *Statistics in Finance*. Arnold, London.
6. Hastie, T., Tibshirani R. and Friedman J.(2001). *The Elements of Statistical learning. Data Mining, Inference, and Precision*. Springer Series in Statistics
7. McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models* (2nd edition). Chapman and Hall, London.
8. Salford Systems (2002). Classification and Regression Trees (CART®). Details on-line, http://www.salford-systems.com, (accessed 11/04/2002).
9. SAS (2002). [On-line] PROC LOGISTIC in EXACT, http://www.sas.com/service/techsup/intro.html, (accessed 17/04/2002).

10.Smyth, G. (2002). [On-line] Generalised linear modelling, http://www.statsci.org/glm/index.html, (accessed 17/04/2002).

11.Steinberg, D. (2002). Personal communication.

12.Steinberg, D. and Cardell, N. S. (1998a). Improving data mining with new hybrid methods. Presented at DCI's Database and Client Server World, Boston, MA.

13.Steinberg, D. and Cardell, N. S. (1998b). The hybrid CART-Logit model in classification and data mining. Eighth Annual Advanced Research Techniques Forum, American Marketing Association, Keystone, CO.

14.Steinberg, D. and Colla, P. L., (1995). *CART: Tree-Structured Nonparametric Data Analysis*. Salford Systems, San Diego, CA.

Author Index