UKKDD’05

Proceedings of the first UK Knowledge Discovery in Data Symposium

Wednesday 6 April 2005

Frans Coenen (Editor)

EPSRC
Engineering and Physical Sciences Research Council

THE UNIVERSITY of LIVERPOOL

BCS
THE BRITISH COMPUTER SOCIETY

KD-net
EUROPEAN KNOWLEDGE DISCOVERY NETWORK OF EXCELLENCE

SGAI

The first UK Knowledge Discovery in Data Symposium

Organising Committee:

Frans Coenen (Chair)	University of Liverpool
Alex Freitas	University of Kent
Beatriz de la Iglesia	University of East Anglia

Sponsors:

The Engineering and Physical Sciences Research Council (EPSRC)
The British Computer Society's Specialist Group on AI (BCS-SGAI)
The International Federation for Information Processing (IFIP) WG12.2 Machine Learning and Data Mining.

Supported by:

KD-net (The European Knowledge Discovery Network of Excelllence)
The University of Liverpool.

CONTENTS

Enclosures

Dept. Computer Science
The University of Liverpool
March 2005

FOREWARD

Welcome to the first UK symposium on Knowledge Discovery in Data (UKKDD’05). The objective of the symposium is to provide a forum for discussion, dissemination and exchange of ideas between practitioners and researchers working within the broad field of KDD in the UK. To this end a number of key people have been invited to present a "state of the art" review of much of the KDD research work currently in progress within UK institutions.

The motivation for the symposium goes back a few years (late 1990s) when myself and a colleague (Paul Leng) embarked on an industry funded research project on the application of data mining techniques to facilities management data bases. Over the following years my interest in KDD developed into a broader field, and I became increasingly aware of the range of work in KDD that was being carried out in the UK. By the end of 2003 I started to formulate the thought that it would be a good thing if there was some sort of forum at which the UK KDD community could meet. At the same time the British Computer Societies Specialist Group on AI (BCS-SGAI) was forming a policy to support a number of AI related workshops to be held across the UK (coordinated by Tony Allen). The first of these on natural language processing was held in Nottingham, the second on mobile agents was shortly to be held in Edinburgh. A SGAI KDD workshop seemed to be a clear possibility.

A canvass of many of the “key players” in the UK; Max Bramer, Peter Flach, Alex Freitas, David Hand, John Keane, Ross King, Victor Rayward-Smith, George Smith etc., revealed a remarkable amount of enthusiasm for the idea. Consequently, with the support of the SGAI it took a few weeks to set a date, book a venue and invite speakers!

Advertising was done via the SGAI magazine (Expert Update), KDnet (thanks to Michael May and Codrina Lauth), and a number of email lists. I also produced a set of flyers which were passed around the conference circuit.

The university of Liverpool have provided a discount for the venue and the Department of Computer Science has provided support with administration, printing and binding (thanks to Thelma Williams, Ken Chan and Claire Winterbottom).

The idea for getting support from IFIP WG12.2 (Machine Learning and Data Mining) came from Max Bramer (thanks also to Zhongzhi Shi, the current chair of IFIP WG12.2). The support currently consists purely of cross-advertising but may prove useful in the future.

The original idea for getting EPSRC support came from Peter Flach. As a result EPSRC have generously agreed to sponsor the event by providing £1,500 worth of student bursaries (thanks to Andrew Bebb and Rebecca Steliaros of the EPSRC for their support during the preparing and processing of the application).

As to the future it is difficult to say. I hope that some ideas will emerge at the symposium. I do believe that further KDD symposia may be appropriate but possibly as a biannual event (the SGAI may very well provide support for a second symposium). There is also the thought that we might attempt to bring PKDD/ECML to the UK.

Frans Coenen
Chair UKKDD'05

Short Presenter Biographies

Max Bramer is Professor of Information Technology at the University of Portsmouth. He is Chairman of the British Computer Society Specialist Group on Artificial Intelligence and of the Technical Committee on Artificial Intelligence of IFIP, the International Federation for Information Processing. He has a long-term involvement in the field of Knowledge Discovery and was Conference Chair for the fourth IEEE International Conference on Data Mining in Brighton in November 2004.

Peter Flach is Professor of Artificial Intelligence at the University of Bristol. He has published widely on inductive logic programming, multi-relational data mining, and machine learning. He was PC co-chair of ILP'99 and ECML'01, is on the steering committee of the recent ECML/PKDD conferences, and is a regular PC member of all major machine learning and data mining conferences including ICML, ECML/PKDD, ILP, ICDM, SDM, and PAKDD. Prof Flach is associate editor of Machine Learning, and serves on the editorial boards of Journal of Machine Learning Research, Journal of Artificial Intelligence Research, and Artificial Intelligence Communications.

David Hand, FBA, is Professor of Statistics at Imperial College. He was awarded the Royal Statistical Society's Guy Medal in Silver in 2002 partly for his work in areas at the interface with other disciplines, such as data mining. He has published over twenty books in statistics and related areas, including *Principles of Data Mining* and *Pattern Detection and Discovery*.

John Keane is Co-Director of the JISC/BBSRC/EPSRC funded National Centre in Text Mining. Professor Keane’s interests are in the design, development and engineering of data-intensive systems, most recently, working in the areas of location analysis, and data and text mining. John Keane is PI for the EPSRC projects IBHIS and HIPERSTAD, and Co-I for the BBSRC project Text Mining. He is a member of the EPSRC IT/CS Peer Review Panel, a member of the Editorial Board of Simulation Modelling.

Ross King is a Professor of Computer Science and head of the Computational Biology research group at Aberystwyth. His principal research area is in the application and development of machine learning/data mining to bioinformatic and chemoinformatic problems. He was on the programme committee for PKDD'03 (Croatia).

Paul Leng is Professor of e-Learning at the University of Liverpool, and Director of the e-Learning Unit, which is responsible for overseeing the University's online degree programmes, leading to degrees of MSc in I.T. and MBA. As well as e-Learning, his main research interests are in Data Mining, especially in methods of discovering Association Rules. In collaboration with Frans Coenen, he has developed efficient new algorithms for finding frequent sets and is exploring applications in text mining and classification.

George Smith is a senior lecturer within the Knowledge Extraction and Data Mining research group at the University of East Anglia --- the largest specialist Data Mining research group in the UK. He has considerable experience in the practical application of data mining techniques in collaboration with industrial and commercial partners.

The National Centre for Text Mining: aims and objectives[1]

Sophia Ananiadou[*], *Julia Chruszcz*[$], *John Keane*[^], *John McNaught*[^] *and Paul Watry*[+]

[*]School of Computing, Science and Engineering University of Salford

[$]MIMAS, Manchester Computing, University of Manchester

[^]School of Informatics, University of Manchester

[+]University Library, University of Liverpool

1. INTRODUCTION

In this article we describe the role of the National Centre for Text Mining (NaCTeM). NaCTeM is operated by a consortium of three Universities: the University of Manchester which leads the consortium, the University of Liverpool and the University of Salford. The intention is that the service activity will be run by the National Centre for Dataset Services (MIMAS), based within Manchester Computing (MC). As part of previous and on-going collaboration, NaCTeM involves, as self-funded partners, world-leading groups at San Diego Supercomputer Center (SDSC), the University of California at Berkeley (UCB), the University of Geneva, and the University of Tokyo. NaCTeM's initial focus is on bioscience and biomedical texts as there is an increasing need for biotext mining and automated methods to search, access, extract, integrate and manage textual information from large scale bioresources. NaCTeM was established in Summer 2004 with funding from the JISC, BBSRC and ESPRC, with the consortium itself investing almost the same amount as it received in funding.

2. NEED FOR TEXT MINING IN BIOLOGY

Dynamic development and new discoveries in the domains of bioscience and biomedicine have resulted in a huge volume of domain literature, which is constantly expanding both in size and thematic coverage. With the overwhelming amount of textual information presented in scientific literature, there is a need for effective automated processing that can help scientists to locate, gather and make use of knowledge encoded in electronically available literature [1] [11]. Although a great deal of crucial biomedical information is stored in factual databases, the most relevant and useful information is still represented in domain literature [2]. Medline [3] contains over 14 million records, extending its coverage with more than 40,000 abstracts each month. Open access publishers such as BioMed Central have growing collections of full text scientific articles. There is increasing activity and interest in linking factual biodatabases to the literature, in using the literature to check, complete or complement the contents of such databases, however currently such curation is laborious, being done largely manually with few sophisticated aids, thus the risks of introducing errors or leaving unsuspected lacunae are non-negligible. There is also great interest among biologists in exploiting the results of mining the literature in a tripartite discovery process involving factual biodatabases and their own experimental data.

Therefore, techniques for literature mining are no longer an option, but a prerequisite for effective knowledge discovery, management, maintenance and update in the long term.

To illustrate the growing scale of the task facing specialists trying to discover precise information of interest within the biobibliome, a query such as "*breast cancer treatment*" submitted to Medline's search engine in 2004 returned almost 70,000 references while it resulted in 20,000 abstracts in 2001. Effective management of biomedical information is, therefore, a critical issue, as researchers have to be able to process the information both rapidly and systematically. Traditionally, bioscientists search biomedical literature using the PUBMED interface to retrieve MEDLINE documents. PUBMED is an indexing and retrieval repository that manages several million documents. These documents are manually indexed, where index terms are selected and assigned to documents from a standard controlled vocabulary (the Medical Subject Headings, MESH). The retrieval is implemented as a Boolean keyword search, so documents that fully satisfy a query are

[1] This article originally appeared in Ariadne issue 42, January 2005 http://www.ariadne.ac.uk/issue42/. The authors wish to express their thanks to the editor Richard Waller.

retrieved. Another problem is the selection of the appropriate index terms which would retrieve the most relevant documents. Index terms do not necessarily characterise semantically documents, but are used to discriminate among documents. Classic Information Retrieval (IR) techniques do not use any linguistic techniques to cope with language variability such as synonymy and polysemy which may produce many false positives. Even controlled indexing approaches are inconsistent and limited since knowledge repositories are static and cannot cope with the dynamic nature of documents.

Using classic IR methods is not sufficient because the number of documents returned in response to a query is huge. Therefore, with the overwhelming amount of new terms being introduced in the literature on a daily basis, text mining tools such as automatic term management tools are indispensable for the systematic and efficient collection of biomedical data which go beyond keyword indexing and retrieval. Manually controlled vocabularies are error prone, subjective and limited in coverage. However, once a highly relevant set of documents is returned, through exploitation of term-based indexing and searching, this will typically still be large and, more importantly, will still not yield precise facts at this stage.

Processing biomedical literature faces many challenges, including both technical and linguistic. Technical challenges are posed by, for example, restricted availability and access, heterogeneous representation (storage) formats, extensive usage of non-textual contents, such as tables, graphs, figures, etc. and linguistic challenges are posed by the particularities of the biomedical sublanguage. One of the main challenges in bio-text mining is the identification of biological terminology, which is a key factor for accessing the information stored in literature, as information across scientific articles is conveyed through the terms and their relationships. Terms (which here are taken to include names of genes, proteins, gene products, organisms, drugs, chemical compounds, etc.) are the means of scientific communication as they are used to refer to domain concepts: in order to understand the meaning of an article and to extract appropriate information, precise identification and association of terms is required [4]. New terms are introduced in the domain vocabulary on a daily basis, and – given the number of names introduced around the world – it is practically impossible to have up-to-date terminologies that are produced and curated manually. There are almost 300 biomedical databases containing terminological information. Many of such resources contain *descriptors* rather than terms as used in documents, which makes matching controlled sets of terms in literature difficult. Terminological processing (i.e. identification, classification and association of terms) has been recognised as the main bottleneck in biomedical text mining [5] severely reducing the success rates of 'higher level' text mining processes which crucially depend on accurate depend on accurate identification and labeling of terms. Various approaches have been suggested for automatic recognition of terms in running text [6], [7] and [4]. Crucially, technical terms of the kind we consider here are to be distinguished from index terms used to characterize documents for retrieval: a good index term might not be a technical term; a technical term is of potential interest for text analysis even if it occurs infrequently in a collection; all technical terms in a document are of potential interest for text analysis.

Recognition of terms in text is not the ultimate aim: terms should be also related to existing knowledge and/or to each other; classes of terms and hierarchies of classes need to be established, as it is terms that provide the link between the world of text and the world of ontologies and other such classification schemes; the ontological elements terms map to serve further to drive ontology-based information extraction, discussed further below.

Several approaches have been suggested for the extraction of term relations from literature. The most common approach for discovering term associations is based on shallow parsing and Information Extraction (IE) techniques. These are based either on pattern matching or on IE-based semantic templates. While pattern matching approaches are typically effective, the cost for preparing domain oriented patterns is too high. Recall may be affected if there is not a broad coverage of patterns. Since the separate use of either statistical, knowledge intensive or machine learning approaches cannot capture all the semantic features needed by users, the combination of these approaches is more promising.

Given the dynamic nature of biomedicine, any method should be tunable and application independent. We believe that the usage of available knowledge sources has to be combined with the dynamic management of concepts (terms) encountered in texts. Most current systems address known relationships, and aim at the extraction of semantic or conceptual entities, properties of entities, and factual information involving identified entities. We propose to support not only the extraction of entities, properties and facts but also, through data mining, the discovery of associations and relationships not explicitly mentioned.

Or indeed totally unsuspected (discovery of new knowledge): this is the true power of text mining. Our view of text mining, thus, is that it involves advanced information retrieval yielding all precisely relevant texts, followed by information extraction processes that result in extraction of facts of interest to the user, followed by data mining to discover previously unsuspected associations.

3. ROLE OF THE NATIONAL CENTRE FOR TEXT MINING

The paramount responsibility of NaCTeM is to establish high quality service provision in text mining for the UK academic community, with particular focus on biological and biomedical science. Initial activity will establish the framework to enable a quality service, and to identify 'best of breed' tools. Evaluation and choice of appropriate tools is on-going, and tools will be customised in cooperation with partners and customers, bearing in mind existing competition and advantages to be gained from cooperation with technology providers.

The overall aims of NaCTeM are:

1. to provide a one-stop resource and focus primarily for the UK text mining community for user support and advice, service provision, dissemination, training, data access, expertise and knowledge in related emerging technologies, consultative services, tutorials, courses and materials, and demonstrator projects;
2. to drive the international and national research agenda in text mining informed by the collected experiences of the user community, allied to existing and developing knowledge and evaluation of the state-of-the-art;
3. to consolidate, evolve, and promulgate best practice from bio-related text mining into other domains;
4. to widen awareness of and participation in text mining to all science and engineering disciplines, and further to social sciences and humanities, including business and management;
5. to maintain and develop links with industry and tool suppliers, to establish best practice and provision.

The vision informing NaCTeM is to harness the synergy from service provision and user needs within varied domains, allied to development and research within text mining. The establishment of a virtuous feedback cycle of service provision based both on commercial software and on innovative tools and techniques, themselves in turn derived from user feedback, is intended to enable a quality service whilst ensuring advances within each associated domain. This paradigm is how advances within bio-text mining have occurred, not least within the NaCTeM consortium's recent activity. NaCTeM is working to consolidate existing successes, activity, and working relationships and models, and transfer them to related science and engineering activities and humanities. Importantly, the expectation is that NaCTeM will shortly (Summer 2005) housed in an interdisciplinary bio-centre co-locating life scientists, physicists, chemists, mathematicians, informaticians, computer scientists and language engineers with service providers and tool developers. Such co-location promises a step-change in awareness and utilisation of text mining such that very definite advances can be both realised and sustained.

The services offered by NaCTeM are expected to be available via a web-based portal. Three types of service are envisaged: those facilitating access to tools, resources and support; those offering on-line use of resources and tools, including tools to guide and instruct; and those offering a one-stop shop for complete, end-to-end processing by the centre with appropriate packaging of results. Services will thus include:

- Access to state of the art text mining tools developed from leading edge research
- Access to a selection of commercial text mining tools at a preferential rate
- Access to ontology libraries
- Access to large and varied data sources – guidance, and purchase of data sets at preferential rates
- Access to a library of data filtering tools
- On-line tutorials, briefings and white papers
- On-line advice on matching of specific requirements to text mining solutions
- On-line performance of text mining and packaging of results involving GRID-based flexible composition of tools, resources and data by users to carry out mining tasks via a portal
- Marketing and dissemination activities: e.g. training and course materials; conference and workshop organisation
- Collaborative development/enhancement of text mining tools, annotated corpora and ontologies
- Text mining tool trials and evaluations

Initially users of NaCTeM will be members of academic and research institutions, and later companies throughout the supply chain in the biotechnological and pharmaceutical industries. In addition, potential users will be public sector information organisations; SMEs in the life sciences sector and IT (knowledge management services) sector; regional development agencies; health service trusts and the NHS information authority; major corporates in the pharmaceutical, agropharma and life sciences industries including food and healthcare; government and the media.

NaCTeM integrates areas such as:

- Bioinformatics and genomics. Research involves predicting and extracting properties of biological entities through combining large-scale text analysis with experimental biological data and genomic information resources. The use of supervised learning over both text and biological data sources increases novelty detection. Recently, work has started on non-supervised learning approaches using sophisticated term and term relationship extraction. The overall goal is to discover strategies and methods that facilitate user comprehension of experimental data, genomic information and biomedical literature simultaneously.
- Ontologies, Lexica and Annotated Text Corpora.
  Ontologies describe domain specific knowledge and facilitate information exchange They store information in a structured way and are crucial resources for the Semantic Web. In an expanding domain such as biomedicine it is also necessary for ontologies to be easily extensible. Since ontologies are needed for automatic knowledge acquisition in biomedicine the challenge is their automatic update. Since manual expansion is almost impossible in a dynamic domain such as biomedicine, text mining solutions such as term clustering and term classification are beneficial for the automatisation of ontologies. Term clustering offers potential association between related terms, which can be used to verify or update instances of semantic relations in an ontology. Term classification results can be used to verify or update the taxonomic aspects of an ontology.
  Lexical resources (dictionaries, glossaries, taxonomies) and annotated text corpora are equally important for text mining. Electronic dictionaries give formal linguistic information on wordforms. Furthermore, as ontologies represent concepts and have no link with the surface world of words, a means is needed to link canonical text strings (words) with ontological concepts: dictionaries and taxonomies aid in establishing this mapping. Annotated text corpora [9] (GENIA) are essential for rule development, for training in machine learning techniques and for evaluation.

4. MEETING THE NEEDS OF USERS

We now elaborate on some of the above points where these concern the core text mining components provided by the consortium to underpin the national service.

Overall, the text mining process involves many steps, hence potentially many tools, and potentially large amounts of text and data to be analysed and stored at least temporarily, including all the intermediate results, and the need to access large resources, such as ontologies, terminologies, document collections, lexicons, training corpora and rule sets, potentially widely distributed. Much of the processing is compute-intensive and scalability of algorithms and processes is thus a challenge for the service to meet the requirements of users. Moreover, we expect that many users will want to process the same data again and again (e.g. Medline), with perhaps variation of need only at the higher levels of analysis (fact extraction) or during the data mining stage. It is thus inappropriate to, say, expensively analyse the entire contents of Medline for each text mining request. Essentially, parts of the analysis of some collection will remain unchanged, or will change only slowly with the advent of improved analysis techniques. Thus, once a collection has been processed once to annotate terms and named entities, and properties of these, there is no need to do so again in the general case: there is only a need to analyse new additions to a collection since the last analysis. We expect therefore that part of the service will be devoted to elaborating techniques to reuse previously analysed material, while also developing and exploiting caching techniques to cut down on the amount of processing and data transfer that may otherwise be required. For example, one need only think of the overhead involved in simple lookup of an ontology or dictionary for every concept or word in a collection of several million documents, where numerous requests are received to process that collection: lookup is one of the more straightforward mechanisms of text mining, but the scale of the task here is the point at issue, within a national service. Users may in fact prefer or have to use distributed high-speed processing facilities for large-scale processing rather than their desktop PC. In this case, it is essential to consider portals, GRID capabilities, and access to hosts capable of handling high-dimensional data for the academic community. We thus, in the national service, face a challenge typically irrelevant to or unaddressed by many current text mining systems: the need to provide scalable, robust, efficient and rapidly responsive services for very large collections that will be the target of many simultaneous requests, in a processing workflow where each process may have to massively consult large-scale resources, manage massive amounts of intermediate results and take advantage wherever possible of sophisticated optimisation mechanisms, distributed and parallel computing techniques. Then again, we must do all this while further recognizing the need of many users for security and confidentiality with respect to especially the high-level fact extraction and data mining results that they obtain, which leads us naturally into consideration of secure portals and management of levels of sharing of intermediate data and results.

Development work at NaCTeM will thus be emphasising scalability and efficiency issues, in an environment where different levels of access may need to be managed relative to certain types of intermediate data and results. Moreover, as we recognise that there are many different types of text mining, and that a user may not be interested at any one time in all stages, just in some sub-set (e.g. the user may want to stop after applying IR, term extraction and fact extraction processes, or may want simply to do sentence splitting, tokenisation and document zoning), we do not intend to offer a monolithic service consisting of a single workflow of mandatory tools. It is rather our intention to offer flexibility and the potential to reconfigure workflows as required. Hence, we shall be investing effort in elaborating a model that will allow flexible combination of components (tools and resources) to achieve some text mining task, in a GRID or otherwise distributed environment. This further implies a strong interest in standards: adopting appropriate standards or pushing the development of de facto standards where required. For the user, the advantage is that third party tools and resources, along with the user's own components, can be integrated in a distributed workflow, assuming interface standards are adhered to (e.g. web services, although with linguistic processing we must remember that standards are required at the linguistic level, not just at the transport protocol level, to ensure that linguistic data tags and attributes, for example, are consistently labeled and interpreted). We must also not forget that there are many types of user of a national service, and that we must support therefore the expert bioinformatician who is conversant with construction and deployment of components as much as the user who is a domain expert but has no knowledge of or interest in how things work: but a keen interest in getting appropriate results with modest effort in reasonable time. We shall thus also be working to develop environments that will guide users in the identification of appropriate components and resources, or indeed overall off-the-shelf workflows, to accomplish their text mining task. This may well involve an initial interaction with an environment to figure out what the scope of the text mining task is, what kind of facts are being sought, what kind of associations should be looked for, and so on. Our partners from the University of Geneva have long experience in designing and applying quality in use evaluation techniques to guide users in making appropriate choices of natural language processing tools to suit their needs and they will be working closely with us in this area.

As discussed above, term management is a crucial activity in text mining and one that is not at all well handled by the majority of text mining systems. We will be working to render scalable the highly successful ATRACT terminology management system of the University of Salford [10], which is based on a proven, language-independent hybrid statistical and linguistic approach, and to integrate it in text mining workflows.

Ontology-based information extraction is currently in its infancy, at least insofar as sophisticated use of ontologies of events is concerned. Our development work here will focus on developing the University of Manchester's CAFETIERE information extraction system to take full advantage of distributed and parallel computing, to render it scalable, and to augment its caching and data reuse capabilities. CAFETIERE is a rule-based analyser whose rules can access user ontologies of entities and events: facts are extracted from texts by looking for instances of entities that participate in ontological events. The onus of rule writing is much reduced by this approach, as the rule writer can write fewer and more generally applicable rules thanks to the efforts of ontology building by others: there is thus a direct, beneficial relationship between the world of ontology construction and the world of information extraction. CAFETIERE can, moreover, perform temporal information extraction, which is important not only for text mining of ephemera (newswires for competitive intelligence purposes) but also for any domain where there is volatility of terminology, of knowledge, as we see in the biobibliome: there is a need to anchor extracted facts temporally with respect to the terminology used and the state of knowledge at the time. This has a further bearing on curation of data over time and the relationship between a future user's state of knowledge and terminological vocabulary and those of the archived texts being analysed. CAFETIERE is in fact a complex package of individual components including tokenisers, part of speech taggers, named entity recognisers, etc. Each of these will be made available for separate use.

The University of Liverpool and UCB have jointly developed a 3rd generation online IR system, Cheshire, based on national and international standards and in use by a wide variety of national services and projects throughout the UK. The software addresses the need for developing and implementing advanced networking technologies required to support digital library services and online learning environments. We will use Cheshire to harvest and index data using an advanced clustering technique which will enable items to be interlinked automatically and retrieved quickly. This will include Cheshire support as a cross-protocol data harvester and as a transformation engine operating in a distributed, highly parallel environment. Development work on Cheshire will concentrate on meeting the IR needs of text mining, with particular work on advanced indexing and retrieval, focusing on metadata, on improved index term weighting, on search interfaces, and on ontology management. A key development will be the use of SDSC's SKIDL toolkit and Cheshire to enhance index term weighting approaches in an automatic text retrieval context, by combining Latent Semantic Analysis with probabilistic retrieval methods to yield salient text fragments as input for following information extraction components. The SKIDL data mining toolkit will be integrated not only to allow data mining over classic information extraction results, but also to associate biological entities, such as parsed genome data with bioscientific texts and bibliographic data. The primary advantage is that Cheshire will be able to support hybrid text mining (e.g. from a journal and from textual representations of DNA) in a transparent and efficient manner.

Data mining techniques have traditionally been used in domains that have structured data, such as customer relationship management in banking and retail. The focus of these techniques is the discovery of unknown but useful knowledge that is hidden within such data. Text mining extends this role to the semi-structured and unstructured world of textual documents. Text mining has been defined as 'the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources' [8]. Mining techniques are thus used to link together in a variety of ways the entities extracted from the IE activity. A number of approaches are possible for example, clustering is an unsupervised technique that produces groupings of related entities based on similarity criteria; classification is a supervised technique that learns from instances, for example, of user-classified documents of different types to auto-classify unseen documents; and association rules enumerate the frequency of occurrences of sets of entities, and in particular can derive the likelihood of a document containing specific entities given that the document is known to contain another entity.

5. CONCLUSIONS

The services provided by NaCTeM will not all be available instantly. It will be appreciated that the configuration and deployment of a range of scalable, efficient text mining services cannot happen overnight. Work is planned over 3 years, with increasing evolution towards full service capability. Initially, while development work is under way, we will be acting partially as a clearing house, catalogue and repository for 3rd party, open source or GNU licensed text mining tools, as a means of easily finding useful text mining tools and resources on the Web, and as an advice, consultancy and training centre. As our infrastructural text mining tools are developed, these will be released when appropriate for test purposes in order to gain feedback, before being fully deployed. At present, we are in the setting-up and requirements gathering phase.Throughout, close contacts will be established and maintained with the target user community, to ensure that needs and requirements are met, and that the range of possibilities for text mining is communicated and discussed in sufficient measure to inform the requirements gathering process. We also actively invite contact and discussion with potential users of text mining services from all other domains, as it is part of our remit to reach out to users in other areas in expectation and preparation of future evolution to serve their needs. Our events calendar testifies to the range of contacts we have had, presentations given and workshops attended thus far and we fully expect this activity to grow, given the high degree of interest that has been generated in the community in the centre's aims and activities.

References

[1] Blaschke, C., Hirschman, L. & Valencia, A. 2002. Information Extraction in Molecular Biology. *Briefings in Bioinformatics*, 3(2): 154-165.

[2] Hirschman, L., Park, J., Tsujii, J., Wong, L. & Wu, C. 2002. Accomplishments and Challenges in Literature Data Mining for Biology, in *Bioinformatics*, vol. 18, no 12, pp. 1553-1561

[3] MEDLINE. 2004. National Library of Medicine. Available at: http://www.ncbi.nlm.nih.gov/PubMed

[4] Krauthammer, M. & Nenadic, G. (2004) Term Identification in the Biomedical Literature, in Ananiadou, S., Friedman, C. & Tsujii, J. (eds) Special Issue on Named Entity Recognition in Biomedicine, *Journal of Biomedical Informatics*.

[5] Ananiadou, S., 2004: Challenges of term extraction in biomedical texts, available at: http://www.pdg.cnb.uam.es/BioLink/workshop_BioCreative_04/handout/

[6] Jacquemin, C., 2001: Spotting and Discovering Terms through NLP, MIT Press, Cambridge MA

[7] Ananiadou, S., Friedman, C. & Tsujii, J (eds) (2004) Named Entity Recognition in Biomedicine, Special Issue, *Journal of Biomedical Informatics*, vol. 37 (6)

[8] Hearst, M., 2003: What is Text Mining? http://www.sims.berkeley.edu/~hearst/text-mining.html, October 2003.

[9] GENIA, 2004: GENIA resources available at http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/

[10] Mima, H., Ananiadou, S. & Nenadic, G. (2001) The ATRACT Workbench: Automatic term recognition and clustering for terms. In Matousek, V., Mautner, P., Moucek, R. and Tauser, K. (eds.) *Text, Speech and Dialogue*. Lecture Notes in Artificial Intelligence 2166. Springer Verlag, Heidelberg, 126–133.

[11] Bretonnel Cohen, K. and Hunter, L. (in press) Natural Language Processing and Systems Biology. In Dubitzky and Pereira (eds) Artificial intelligence methods and tools for systems biology. Springer Verlag.

Spotting the difference: detecting local structures in large data sets

David J. Hand and Zhicheng Zhang
Imperial College London
{d.j.hand@imperial.ac.uk; zhzhang@imperial.ac.uk}

Abstract: Pattern discovery and detection is that part of data mining concerned with finding 'small, local' structures in large data sets. Although many algorithms have been developed for such exercises, the discipline lacks a solid theoretical base comparable to that developed over the twentieth century for the modelling of large scale structures. Some of the issues of pattern discovery and detection are described, and an outline of a general pattern discovery tool we are developing is given.

Keywords: data mining, pattern detection, pattern discovery, scan statistics, Peaker,

1. INTRODUCTION

Data mining has two faces (Hand *et al*, 2000, 2001). The first is the familiar one of building models which summarise large scale structures in the data. Examples of such models are segmentation analyses, regression analyses, neural networks, time series decompositions, graphical models, and so on. Such models have been well developed by the statistical community and, for certain classes of model, by the machine learning community. That being said, the common feature of data mining problems, that the data sets are very large, does give a new complexion to the problems. These issues have been discussed in depth in the above references.

The second face of data mining is what we call, for convenience, *pattern discovery and detection* (Hand and Bolton, 2004). Like statistical modelling, this covers a diversity of rather different exercises, but all with a common core. This common core is the detection of particular *local* features within a large data set. 'Local' here means that only a relatively small number of data points is involved, in constrast to modelling which generally seeks to summarise a large number of data points. In fact, of course, the words 'small' and 'large' are relative: a 'small local cluster' in astronomy could involve millions of stars, but their spatial distribution and, indeed, their number, will be small relative to the immensity of space.

The search for local features in data has always been an important issue, even before the advent of modern data mining. For example, observations which do not behave in the same way as the mass of observations, or which depart in some way from a theoretical explanation, suggest that the theory is not perfect and needs to be extended or replaced. Hargattai (2002, p71) describes how the chemist Herbert Brown was led to his discovery of hydroboration, and a Nobel Prize, when one in 57 substances behaved in an anomalous way. That being said, things have changed dramatically over the past decade or two. Developments in computer processing power, storage capacity, and automatic data acquisition, by electronic measuring instruments either directly measuring a feature of interest (e.g. telemetry from space shots, analysis of particle physics experiments, monitoring of weather) or capturing details of some human action (e.g. storing details of supermarket purchases, phone calls, or banking transactions) mean that the data simply stack up at vast rates without human intervention.

This deluge of data has led to increasing opportunities for detecting particular configurations of data and for finding anomalous behaviour or departures from the norm or from what was expected. It has also led to the requirement for new tools for searching the vast data sets, and to the need for theoretical developments to cope with some of the new issues which such problems generate. This paper looks at some of the tools and theoretical issues. Before doing so, however, to illustrate the breadth and context of such problems, here are a few examples from a wide range of different areas.

The SETI@home project uses idle time on a large number of computers, connected by the Internet, to search for narrow-bandwidth radio signals from space. The software aims to separate such signals from noise and man-made signals. Noise arises from other astronomical sources and also from random events internal to the receiver. Man-made signals include background radar, satellite, and TV and radio signals. The time-varying power spectrum of the data is calculated, pattern detection algorithms identify candidate patterns, and then these are compared with known patterns to see if there is a ready explanation.

The SETI project seeks anomalies - patterns which cannot be explained in terms of standard explanations. Other pattern detection problems involve locating data configurations of known kinds. For example, in real-time monitoring of credit card transactions, the data will be passed through a filter which looks for particular sequences of transactions known to be characteristic of fraud, and which are unusual for the customer. A card which was rarely used to obtain cash, but which suddenly showed a series of consecutive transactions for this purpose would arouse obvious suspicion, but so also would less obvious transaction patterns, such as the unusual purchase of multiple items of jewellery (portable and easily sold on) - see, for example, Bolton and Hand (2002).

Medical examples are increasingly common. With the advent of increasingly sophisticated scanning technologies, the aim is often to detect an anomalous structure against the background of what is normal. Detecting disease clusters (e.g. to discover the source of an outbreak, or perhaps an environmental cause) is another well-established medical application of pattern detection.

In banking, a recent application is in hedge funds, where the aim is to detect groups of stocks which move in predictable ways or for which the relationship behaves in a predictable way. The tools are also used in an effort to control money laundering. For example, in 1970 the US Bank Secrecy Act required that banks report all currency transactions of over $10,000 to the authorities. This led to the practice of dividing larger sums into multiple amounts of less than $10,000, and depositing these in different banks - a practice termed *smurfing* or *structuring*, and which is now illegal. Such a behaviour pattern can be detected (provided, of course, one has the data).

Another recent development is the application of pattern detection algorithms to monitor airline passengers in the USA. From the name, address, phone number, and date of birth, the airline reservation systems can automatically obtain a background check which includes a banking history and a criminal background check, leading to a risk assessment for each passenger. More generally, the DARPA's Total Information Awareness programme aims to combine data from a variety of sources which can then be searched for suspicious behaviour patterns.

2. THE VARIETY OF PATTERN SEARCH PROBLEMS

A useful distinction is between *pattern matching* and *pattern discovery*. In pattern matching the aim is to locate occurrences in the database of specified configurations of data values. For example, we might seek sequences of particular stock price values in technical analysis of stock markets (.e.g a 'head and shoulders' pattern), or search for particular nucleotide sequences in genomic analysis. In general, the pattern will not be completely determined, but will include wildcards and other ways in which the match may be generalised. For example, the match may permit a particular subsequence of symbols to be repeated an arbitrary number of times.

Some of the work on pattern matching seems to have recapitulated work on *syntactic pattern recognition* carried out in the 1960s and 1970s. A grammar of symbols and allowed relationships between symbols permitted transitions between symbols, according to certain probabilities given by a finite state automaton, and one could identify which automaton had the highest probability of having generated the observed sequence.

In pattern discovery, in contrast, the aim is to detect anomalies without any clear definition of what an anomaly is. Rather, one begins with a description of the background, relative to which the data configuration is anomalous. Outlier detection in statistics is a familiar illustration of this. Here the expected distribution of the data is given, and any data point which occurs with a sufficiently low probability is regarded as anomalous - an outlier. A less familiar example is the search for local subsets of data points which are unexpectedly similar. For example, pulsars were discovered when Jocelyn Bell Burnell recognised that a particular trace on records showing the apparent fluctuation in the intensity of radio emissions was repeated, in very similar form, whenever the radio telescope was pointed at a particular part of the sky. The signal was expected to have particular statistical properties, and these repeated similar observations departed from that, being sufficiently similar to each other to arouse suspicion.

It is obvious from the above that the notion of *distance* is fundamental in pattern detection and discovery problems. Only in certain (we would suggest rare) kinds of problems are *exact* matches the sole interest. Thus the aim is to find those data configurations which are similar (i) to specified configurations in the case of pattern matching, and (ii) to other data configurations in the data in the case of pattern discovery.

Sometimes it is also useful to consider a third type of problem, really a variant of pattern matching, in which the aim is not simply to locate occurrences or near occurrences of a given configuration of data values, but to identify the values of other variables which are associated with the occurrences of those data configurations. Thus, for example, we might want to know what sort of lifestyle characteristics lead to certain disease symptom patterns. Clearly this kind of pattern search is closely related to regression (what kind of predictor variables lead to a given response value) and to classical statistical pattern recognition (see, for example, McLachlan, 1992; Hand, 1997; Webb, 2002), although the aim is not to build a global predictive model, as it is in those cases.

From a technical, rather than an applications perspective, another important distinction is between pattern *search* algorithms and pattern *verification* algorithms. The former are concerned with finding configurations of data values which may be of interest, while the latter are concerned with establishing whether the detected configuration really represents some underlying reality, or is merely attributable to chance variation. This latter point, discussed in Section 4, is a pervasive issue in pattern detection and discovery: with a large enough data set with a substantial random aspect, any configuration you care to think of may have a high probability of occurring.

Even within the categories of pattern matching and pattern discovery there are different types of problems. Take pattern discovery as an example. Hand and Bolton (2004) attempted to provide a unified description of this aim as being to find local anomalous peaks of probability or probability density. Such peaks would be revealed by the occurrence of too many similar data configurations relative to the background probability (density) (as, for example, in the the pulsar example), but ‘too many’ can even be as few as one (e.g. in the case of outliers). In certain special problems, however, we are specifically searching for just a handful (typically two) data points which have strikingly similar vectors of values. We give an example, involving cheating students, in Section 5.

3. THE PEAKER ALGORITHM

Adams *et al* (2001) described an algorithm for detecting local peaks of probability density. The essential points of this algorithm (called ‘Peaker’) are:

- to estimate the underlying probability density at each data point;
- to find those data points at which the estimated probability density is higher than that at all neighbouring points.

These points with highest local estimated probability density function (pdf) are called peaks. They are approximations to local peaks of the underlying true pdf, and hence to local regions of anomalously high pdf. By restricting the calculations to the actual data points, the intractable problems of searching over high dimensional spaces (see, for example, the discussion of scan statistics below) are avoided.

Of course, this outline description does not define the algorithm completely, and it leaves open many choices. For example, how is the initial estimate of probability density obtained? All nonparametric probability density estimates represent smooths of the empirical distribution $\{x_1,...,x_n\}$, and if too much smoothing is used all peaks, except for a single global peak, will be smoothed away. Likewise, how is the 'neighbourhood' within which the probability density estimate is a maximum defined? Once again, a very large neighbourhood (all of the data, for example) will yield only a single peak, while a very small neighbourhood (at an extreme, just the single nearest neighbour, for example) will yield many peaks. Of course, underlying all of this is the choice of metric used in calculating the initial pdf estimates. When, as is very often the case, the variables are non-commensurate, this is not a trivial choice, but it is one which can affect the results.

The fact is that there is no uniquely best answer to these questions. Different answers will lead to variants of the algorithm which have different properties. The situation is analogous to cluster analysis, in which different choice of metric (Euclidean, city block, etc.), clustering criterion (trace of the within-cluster cross-product matrix, determinant of this matrix, etc.), and optimisation method (e.g. hierarchical or agglomerative) lead to cluster structures with different properties. This variety is in fact useful in an exploratory tool, where one will often be applying it to situations in which one cannot specify clearly at the start exactly what sort of structure one is seeking.

There has been considerable work on finding optimal degrees of smoothing for nonparametric pdf estimates, and the results of this could be used in choosing the pdf estimation method. In a real sense, this strikes an optimal choice between variance and bias in the pdf estimate.

As to the extent of the neighbourhood within which a point is a local maximum, we have explored various rules. An obvious first suggestion is to specify *m* beforehand (say *m* = 20 or 50, say, or perhaps as a proportion of the total data set size). This is all very well, but it is difficult to give guidance on an appropriate value for *m*, other than to say that it should not be too small (or random variation will detect many spurious peaks).

An alternative, in the spirit of exploratory data analysis and data mining, is to begin with a large value of *m* (equal to the entire data set, if one likes) and gradually decrease it. Initially, a single point will have the largest estimated pdf (there must be such a maximum, ignoring pathological cases, and difficulties arising from discrete data values). As *m* becomes smaller, so, at some value, another point will have a higher estimated pdf than its *m* nearest neighbours (but not its $(m+1)$ nearest neighbours). This is continued as far as one likes, though probably there is little point in continuing it beyond the point at which perhaps 10 or 20 data points corresponding to such local peaks have been detected. Note that a peak detected with $m = m_1$ would also be detected with $m = m_2$ whenever $m_2 < m_1$.

Figure 1(a) shows an example of this approach applied to a scatterplot of petal length by petal width in Fisher's iris data. The three solid points show the three 'largest' peaks found by starting *m* with the entire data set and decreasing it. In both this example and the simulated one below, a kernel density estimate was used with bandwidth selected by biased cross-validation (Sain *et al*, 1994).

However, neither of the above approaches would help us capture the essence of anomalous local concentrations. This is that they represent 'spikes' in the probability distribution - which by definition contains only a small probability mass. Both of the above methods would also detect the maximum of a mode in the underlying distribution which contained a large proportion of the probability mass. For example, the classic Fisher iris data has several such modes, each of which, one might hope, would lead to a detected peak. But none of those correspond to 'small local anomalously high regions of probability density' in any reasonable sense of 'local'. Somehow we need to measure the sharpness of a peak.

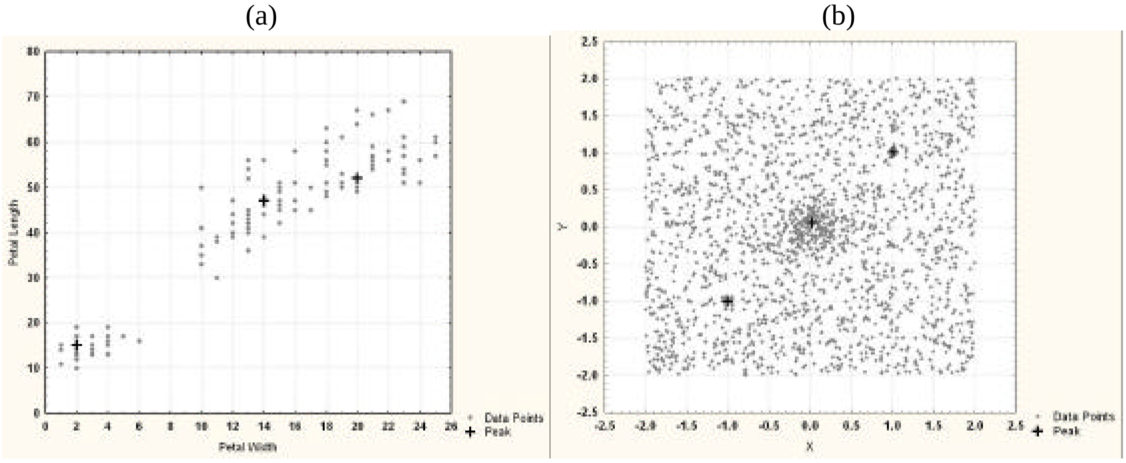


Figure 1: *(a) The Peaker algorithm applied to two variables from the Fisher iris data. (b) An example of the Peaker algorithm using the sharpness detection criterion on simulated data.*

One way of achieving this is as follows. Find potential peaks as above: a data point *x* is a potential peak if it has a higher estimated pdf $\hat{f}(x)$ than its *m* nearest neighbours, but not its $(m+1)$th nearest neighbour. Then a measure of sharpness of the peak is given by $\hat{f}(x) \Big/ \frac{2}{m}\sum_{i=1}^{m/2} \hat{f}(x_i)$, where points $\{x_1,...,x_{m/2}\}$ are the nearest $m/2$ points to *x*. The choice of $m/2$ here is arbitrary, and one might want to experiment with alternatives. Indeed, it might be wise to make the choice depend on the dimensionality, since in high dimensional spaces more points will lie further from *x*.

Figure 1(b) shows an example of this, on synthetic data. The background distribution is uniform on the square with opposite corners (-2,-2), (2,2). There are three bivariate normal distributions centred at (-1,-1), (1,1), and (0,0), with respective covariance matrices 0.025**I**, 0.025**I**, and 0.2**I**. The mixing probabilities of these four components are, respectively, 0.85, 0.025, 0.025, and 0.1. There are 2000 points in all. The crosses show the three peaks with the largest sharpness measures, as described above. Certainly, in this example, the method is highly effective: while the central peak is obvious to the naked eye, the other two are not.

We have begun to explore the properties of different variants of Peaker in simulations. This has thrown up various properties to which attention must be paid. For example, data points on the convex hull are unlikely to be flagged as peaks. This is easily seen in one dimension, where the extreme points must necessarily have lower pdf estimates than the points next to them. In higher dimensions things are not so extreme, but here things are further complicated by the fact that a larger proportion of the data are 'extreme' (lying on the convex hull). One implication of this is that the performance of Peaker might be affected by the shape of the region of support of the underlying pdf.

4. BUT IS IT REAL, IS IT INTERESTING, AND DOES IT MATTER?

With large data sets, there are many opportunities for chance coincidences to occur. Broadly speaking, the larger the data set, the greater the chance, and, of course, this probability is magnified when many different types of coincidences are sought. One needs to be aware of this when flagging local structures as possible patterns. For example, after three F-14 jets crashed in 25 days, the US Navy suspended all F-14 operations, but calculations suggest that the chance of such an event occurring in some month over a five year period is over a half.

Most work on pattern discovery has focussed on finding potential patterns in the first place - on finding local peaks of probability density, for example. This is reasonable - before one can examine them, one must find them. It is also a consequence of the history of development of data mining, with most early researchers being based in computing departments. However, once one has flagged a structure as of possible interest, one also needs to know if is likely to be real. Bolton *et al* (2004) described two possible tests, and illustrated them in action. More generally, though in a more restricted context, the ideas of scan statistics look at similar issues. In principle, a scan statistic moves a local window over the data space, looking for anomalous values of some function of the data within that window (e.g. an exceptional local density). Most of the work on this topic to date has concentrated on low dimensional spaces, and often assumes a relatively simple background model.

Multiplicity and chance occurrence of patterns is one deep problem with which pattern discovery searches must contend. Another is the simple fact that most of the patterns will be either already known or not be of any intrinsic interest. In the context of market basket analysis (a particular kind of pattern discovery problem, concerned with purchase data) Brin *et al* (1997) found 20,000 rules, but concluded 'the rules that came out at the top were things that were obvious'.

Things are further complicated by issues of data quality. This is something to which far too little attention has been devoted by the data mining community, and yet it is fundamental. Normally researchers are interested in discovering something about the mechanism underlying the data generating process, not the data capture process. If inadequacies in the latter induce anomalies in the data, then much effort has been wasted.

5. SOME EXAMPLES

Example 1: Cheating by a minority of dishonest students is a perennial problem, and it is one that has been aggravated by the shift towards coursework at the expense of examinations. It is all too easy for students to collude. Hand *et al* (2005) describe a pattern discovery system aimed at detecting when two (or more) students have produced statistics coursework assignments which are so similar that it is extremely unlikely that this occurred by chance. They do this by coding up the students' reports using mathematical syntactic markers such as +, -, and = (so that changing symbols, such as letters, will not prevent detection). This yields a high-dimensional data space so that similar descriptor vectors are unlikely to occur by chance.

Example 2: In unpublished work Richard Bolton and David Hand developed a tool they called *peer group analysis*, for detecting banking fraud. Traditional fraud detection methods use a variety of tools. One of them is based on modelling a customer's usual behaviour, so that sudden departures from this can be detected. In peer group analysis, retrospective data on each customer is used to identify a 'peer group' of other customers who tend to behave in a similar way. Then, by following each customer and his/her peer prospectively, anyone whose behaviour begins to deviate from that of their peer group can easily be detected. Figure 2 below shows an example of a peer group trace for a credit card customer who suddenly began to behave in an extreme manner. The very light traces are a random sample from the overall database of customers. The dark gray traces are the current customer's peer group. And the black line is the trace for the customer in question. He/she is clearly behaving very differently from most of those to whom he/she was previously very similar.

Example 3: When telecoms engineers repair faults they sometimes inadvertently cause other faults - especially if the infrastructure is beginning to age. This manifests itself in small local clusters of faults over time. In a study of this phenomenon, Dave Yearling and David Hand examined how faults cluster over time in a large number of distribution points in a large telecoms network. Figure 3 shows data from a sample of distribution points (columns). Time progresses down the figure. The light gray marks show when a fault has been repaired. The darker marks show when several faults, at a density greater than would have been expected by chance if their occurrences were independent, have been detected.

Example 4: Post marketing surveillance for pharmaceuticals involves monitoring reports of adverse reactions associated with taken drugs in an attempt to detect when the drugs were the causes of these reactions. These attempts are faced with some major difficulties; for example, a lack of information about the extent of prescription of drugs, and a need to wait until enough data have accumulated to give statistically valid conclusions. We have been developing a method based on dissimilarity measures between drugs, to tackle this last problem: drugs which are chemically similar, or which, based on the data, show similar adverse event patterns, are combined into a higher level 'drug class' which can then be used as a basis for a pattern search.

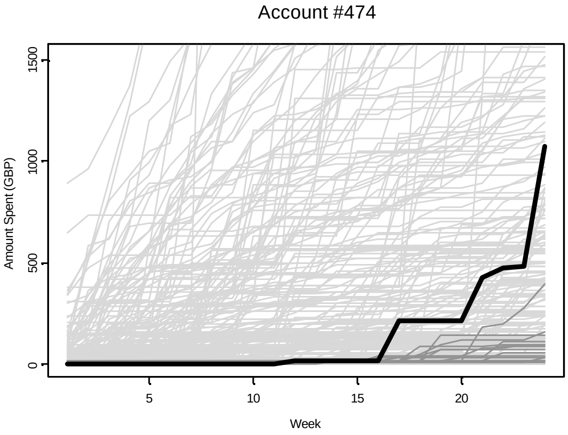


Figure 2: *A peer group analysis trace.*

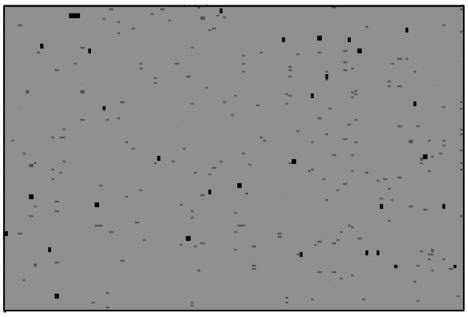


Figure 3: *Clusters of faults in distribution points of a telecoms network*

6. CONCLUSION

Pattern discovery and detection describes those aspects of data mining concerned with finding small, possibly anomalous, local structures in (typically large) data sets. It is an area which is the focus of increasing attention, based on the philosophy that there is almost certainly information of value concealed in large data sets, and that experience shows that this is likely to be where something unusual appears to occur.

Most work in this area has focussed on algorithms for detecting such local structures. The discipline lacks a solid theoretical basis analogous to that which has been developed for the modelling of large scale structures in datasets, over the course of the twentieth century (Hand and Bolton, 2004).

ACKNOWLEDGEMENTS: The work of the second author was supported by an EPSRC CASE award in conjunction with GlaxoSmithKline. We are particularly grateful for the support given by Dr Steven Barrett.

REFERENCES

Adams N.M., Hand D.J., and Till, R.J. (2001) Mining for classes and patterns in behavioural data. *Journal of the Operational Research Society*, **52**, 1017-1024.

Bolton R.J. and Hand D.J. (2002) Statistical fraud detection: a review. *Statistical Science*, **17**, 235-255.

Bolton R.J., Hand D.J. and Crowder M.J. (2004) Significance tests for unsupervised pattern discovery in large continuous multivariate data sets. *Computational Statistics and Data Analysis*, **46**, 57-79.

Brin S., Motwani R., Ullman J.D., and Tsur S. (1997) Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Tucson, Arizona*, ACM Press, 255-264.

Hand D.J. (1997) *Construction and Assessment of Classification Rules*. Chichester: Wiley.

Hand D.J., Adams N.M., and Heard N.A. (2005) Pattern discovery tools for detecting cheating in student coursework. Technical Report, Department of Mathematics, Imperial College London.

Hand D.J., Blunt G., Kelly M.G., and Adams N.M. (2000) Data mining for fun and profit. *Statistical Science*, **15**, 111-131.

Hand D.J. and Bolton R.J. (2004) Pattern discovery and detection: a unified statistical methodology. *Journal of Applied Statistics*, **31**, 885-924.

Hand D.J., Mannila H., and Smyth P. (2001) *Principles of Data Mining*. Cambridge, Mass.: MIT Press.

McLachlan G.J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley.

Sain S.R., Baggerly K.A., and Scott D.W. (1994) Cross-validation of multivariate densities. *Journal of the American Statistical Association*, **89**, 807-817.

Webb A. (2002) *Statistical Pattern Recognition*, 2nd ed. Chichester: Wiley.

Applied Inductive Logic Programming

Ross D. King, Sébastien Ferré, Amanda Clare
Department of Computer Science, University of Wales, Aberystwyth,
Penglais, Aberystwyth, Ceredigion, SY23 3DB, Wales, U.K.

Abstract We describe the application of Inductive Logic Programming (ILP) and Relational Data Mining (RDM) through application to a specific problem in bioinformatics. The problem comes from the field of functional genomics and is to predict the functional class of genes. This requires an ILP approach because the data is relational i.e. cannot naturally be represented as a single table. The data also has the complications of having a class hierarchical structure, and that the examples may have multiple class labels. The data is represented in Datalog. We used a hybrid ILP/propositional learning approach. We first used ILP to find frequent patterns in the Datalog data. We used two different ILP approaches. In the first we used our PolyFARM algorithm. This algorithm is based on WARMR and is a distributed first order association mining method that runs on Beowulf clusters. We also applied an approach based on dichotomic search and domain specific logics. The frequent patterns found by the ILP systems were converted into Boolean attributes, and the C4.5 algorithm used to generate prediction rules. We demonstrate the accuracy of our approach, and show several examples of predictions which have subsequently been demonstrated by biological experiment to be correct.

Key Words: Machine Learning, Bioinformatics, Proteins, Relational Data Mining.

1. INTRODUCTION

1.1. Inductive Logic Programming

Machine learning and data mining methods that employ first-order predicate logic (FOPL) to represent examples, background knowledge, and theories are generally described as coming from the field of Inductive Logic Programming (ILP) [1-3] or Relational Data Mining (RDM) [4] - depending on research emphasis. For simplicity in this article I will refer to all such algorithms as coming from ILP. ILP has shown its value in many scientific problems: in drug design and toxicology e.g. [5-10], and in molecular biology [11-16]. ILP has been particularly well suited to problems dealing with molecular structure. In such problems ILP has often found solutions not accessible to standard statistical methods, neural network, propositional machine learning, or genetic algorithms [6]. The theories produced by ILP have also been generally more comprehensible than those using propositional methods as they are more compact and closer to natural language [5,14].

1.2. Functional Genomics

For the first time in history we have access to the complete genomes of living organisms. These genomes provide the complete specification of the parts and programs to create living organisms. This knowledge is revolutionizing biology. Perhaps the most important discovery from the sequenced genomes is that the functions of only ~30-60% of the predicted genes are typically known with any confidence. For example in bakers yeast (*S. cerevisiae*), one of the most intensely studied of all organisms: of the ~6,000 predicted protein-encoding genes, the function of only ~70% can be assigned with any confidence. The science of *functional genomics* [17] is dedicated to determining the function of genes of unassigned function, and to further detailing the function of genes with purported function.

A key bioinformatic task in functional genomics is the prediction of protein function from sequence. Such predictions provide both important initial information about newly sequenced genomes and they aid "wet" experimental determination of function. Such predictions are usually done by using sequence similarity methods to find an evolutionary related (homologous) protein in the database which has a known function e.g. [18]. The function of the new sequence is then inferred to be the same as the homologous protein; as it is assumed to have been conserved over evolution. This is a kind of nearest-neighbour type inference in sequence space. Unfortunately, using this approach only around 50% of possible homologies are identified [19], and little biological insight obtained.

1.3. The Suitability of Using ILP in the Prediction of Gene Function

If a problem can be satisfactorily represented and solved using propositional methods then there is no need to apply ILP techniques. Propositional methods are generally better developed and more computationally and statistically efficient. ILP methods do not necessarily default to efficient propositional learners when given wholly propositional data. Use of ILP therefore needs to be justified. Our rationale is based on the following features of the problem and required solution:

Relational descriptors - functional genomics naturally involves many relationships in the data: phylogenic hierarchies (the tree of life), homologies (genes sharing a common ancestor), directed graphs relating functions, etc. Traditional propositional methods (statistical, neural network, machine learning, genetic algorithms, etc) cannot efficiently represent these relations in inductive inference.
Data heterogeneity - the relevant data comes from many different types of source and are necessarily stored in multiple tables of relational databases. To use a conventional data mining algorithm the tables would have to be joined to form a single prohibitively large and sparse table for analysis. This is impractical, and ILP/RDB allows the direct analysis of the multiple table formatted data.
Comprehensible results - it is important that the prediction rules are understandable. Biologists generally require that the rules are understandable so that they can suggest new biological ideas and so that they can have confidence in them. In some bioinformatic applications this is not necessary e.g. predicting protein secondary structure. However, given a choice, comprehensible results are always preferred.

In Aberystwyth we have developed a hybrid ILP/Propositional machine learning method to predict protein functional class directly from sequence [20-23].

2. METHODS

2.1. Data

Perhaps the most important recent advance in bioinformatics has been the development of good classification schemes (ontologies) to describe gene function, e.g. "GO" [24]. These schemes take the form of hierarchies or directed acyclic graphs. The creation of such schemes opened up the possibility of directly predicting gene functional class from sequence. Abstractly, what is required is a discrimination function that maps sequence to biological functional class. The existing sequence homology recognition methods can be viewed as examples of such functions: methods based on direct sequence similarity can be considered as nearest neighbour type functions (in sequence space), and the more complicated homology recognition methods based on motifs/profiles resemble case-based learning methods.

We describe our application of ILP to predicting the function of proteins in bakers yeast (*S. cerevisiae*). The methodology is shown in Figure 1. We first formed a Datalog database containing all the data we could collate from all the ~6,000 gene sequences in yeast. Datalog is the language of function free and negation free Horn clauses (Prolog without functions). As a database query language it has been extensively studied [25]. A wide variety of data sources are available for yeast. We chose to use 5 different types of data: data that can be directly calculated from sequence (e.g. amino acid ratios, molecular weight); phenotype data which describes the result of growth experiments using knockout mutants; microarray data (these measure the expression of each gene in the cell); homology data (the relationship between gene sequences and species type); and predicted secondary structure data. The homology and secondary structure datasets are relational in nature, whereas the other data sets are straightforward attribute-value data (http://www.aber.ac.uk/compsci/Research/bio/dss/ yeastpreds). When the database is represented as a a flat file of Datalog facts in plain uncompressed text, each gene has on average 150K of data associated with it (not including background knowledge). This is in total ~1G for the whole yeast genome when represented in this way.

2.2. PolyFARM

When data mining relational data we need to extend ordinary association mining to relational associations, expressed in the richer language of 1st order predicate logic. The associations are existentially quantified conjunctions of literals. Some examples of associations are:

? X, Y : buys(X, pizza) ? friend(X, Y) ? buys(Y, coke)

? X, Y : gene(X) ? similar(X; Y) ? classification(Y, virus) ? mol weight(Y, heavy)

The 1st order association rule learning algorithm was WARMR [26]. It works in a similar manner to APRIORI [27], extending associations in a levelwise fashion, but with other appropriate methods for candidate generation, to eliminate counting unnecessary, infrequent or duplicate associations. WARMR is a general purpose data mining algorithm that can discover knowledge in structured data. It can learn patterns reflecting one-to-many and many-to-many relationships over several tables. No propositional data mining program can do this, as they are restricted to simple associations in single tables. WARMR also introduced a language bias that allows the user to specify modes, types and constraints for the predicates that will be used to construct associations and hence restrict the search space. Language bias is used to restrict and direct the search.

To deal with the technical challenge of the relational data mining of ~1G of bioinformatic data, we (AJC), developed a WARMR-like algorithm, called PolyFARM which can deal with an arbitrarily large database [28]. The program counts associations in relational data, progressing in a levelwise fashion, and making use of the parallel capabilities of a Beowulf cluster of PCs by running in a distributed manner. PolyFARM is the first system to the best of our knowledge for distributed first order association mining. PolyFARM is written in Haskell.

2.3. Dichotomic Search

As an alternative to the WARMR data-mining approach, we (SF), developed a frequent pattern finding method based on dichotomic search [29]. This approach uses domain-specific logics as intermediates between propositional logic and predicate logic. The term “logic” is here used as a shorthand for a search space defined by a representation language ordered by a generalization relation (alias subsumption). A key issue with domain-specific logics is how generic the search algorithm is, as the search space may change from one application to another. This approach therefore requires generic programming techniques for separating the search algorithm from logic-specific operations (e.g., refinement operators). We further use the technology of logic functors [30] that enables to build logics from simpler parts. Each logic functor corresponds to a data structure (e.g., interval, sequence), and the definition of logics becomes similar to the definition of complex types from primitive types in programming languages.

Most existing algorithms traverse the search space in either a top-down or a bottom-up fashion. We propose a new search algorithm that focuses on the concepts to be discriminated, rather than hypotheses themselves (concept-based search), and which explores the search space in both direction with bigger leaps (dichotomic search). The first advantage of this is that the search is more data-driven, and can cope with infinite chains in the search space, allowing a wider range of logics to be used. Secondly, it combines completeness (w.r.t. concepts), non-redundancy, and flexibility at the same time, which has been recognized as a difficult problem [31]. This available flexibility allows more heuristics to be used to guide the search. We applied our dichotomic search approach only to the protein secondary structure data.

3. ATTRIBUTE BASED LEARNING

The frequently occurring patterns found using PolyFARM and Dichotomic search were converted into Boolean (indicator) attributes for propositional rule learning. An attribute has value 1 for a specific gene if the corresponding query succeeds for that gene, and 0 if the query fails. The propositional machine learning algorithm C4.5 [32] was then used to induce rules that predict function from these Boolean attributes. Good rules were selected on a validation set, and the unbiased accuracy of these rules estimated on a test set. Rules were selected to balance accuracy with coverage. The prediction rules were then applied to genes that have not been assigned a function to predict their functions. N.B. we did not aim for a general model of the relationship between sequence and function, we were satisfied with finding good rules to cover part of the space.

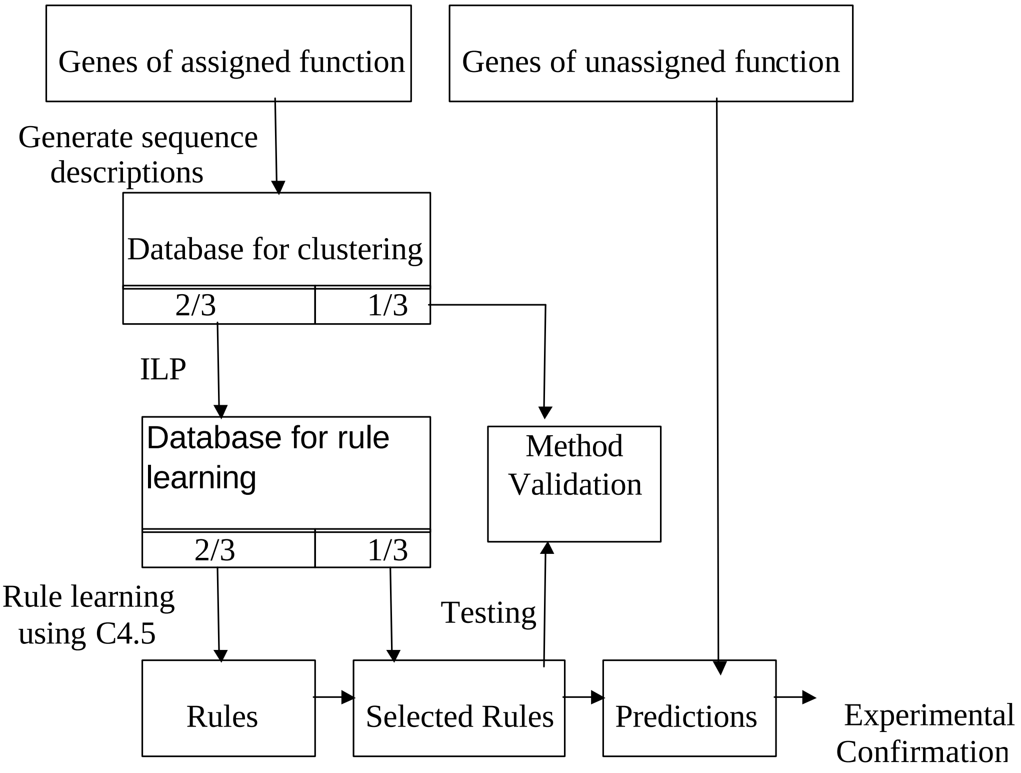

Genes of assigned function
Genes of unassigned function
Generate sequence descriptions
Database for clustering
2/3
1/3
ILP
Database for rule learning
2/3
1/3
Method Validation
Rule learning using C4.5
Testing
Rules
Selected Rules
Predictions
Experimental Confirmation

Figure 1 *Flow chart of the machine learning methodology. This ILP/Propositional hybrid approach has proved successful in the past on other scientific discovery tasks.*

4. RESULTS

All the results are given for the rule sets after validation has been applied (i.e. just the significant rules). The validation was applied by keeping only the rules which were shown to be statistically significant on the validation data set. Statistical significance was calculated by using the hypergeometric distribution with an a value of 0.05 and a Bonferroni correction.

Table 1 shows the test set average accuracy of each of the rule sets produced by PolyFARM from the different types of data. We list the accuracies produced from learning on each level of the classification hierarchy in turn with an especially developed multilabel version of C4.5 [33]. Level 1 is the most general and level 4 the most specific functional annotation. We also list for comparison the accuracy of the hierarchical version of C4.5, which makes use of multilabel classes at all levels. The accuracies range between 75% and 39% on level 1, dropping on lower levels where the data is sparse. These accuracies are highly significant when compared to the *a-priori* class probabilities.

Figure 2 shows a typical rule based on used of secondary structure data. The rule requires several predicted short coils followed by fairly long alphas, and this happens at least 3 times. There is no predicted pattern of coil-beta-coil, and there are no predicted fairly long coils followed by short alphas.

Table 2 shows the predicted functional class of four genes using dichotomic search and C4.5 and their actual function as subsequently uncovered by biological experimentation. The results for gene 1 and 3 are examples of correct predictions, that for gene 2 a "near miss" (as peroxisomes are closely related to mitochondria), and for gene 4 the prediction is plain wrong. Note that we make two predictions for YHL048w.

Datatype	Level				
	1	2	3	4	all
sequence	55	55	33	0	71
phenotype	75	40	7	0	68
structure	49	43	0	0	58
homology	65	38	69	20	55
microarray - ce	63	33	21	0	54
microarray - ch	75	43	0	0	53
microarray - de	64	51	0	0	61
microarray - ei	63	40	28	0	48
microarray - g1	39	46	44	75	38
microarray - g2	44	66	40	0	60
microarray - sp	43	63	0	0	46
microarray - ex	42	37	35	0	75

Table 1: *Average accuracy (percentages) on the test data of each ruleset. Only rules which were statistically significant on the validation set are included. Level ``all'' indicates the results of the hierarchical version of C4.5, which had classes from all levels in its training data. Significance was calculated by the hypergeometric distribution, with alpha=0.05 and Bonferroni correction. Note that we used 8 different types of microarray data.*

If no: coil followed by beta followed by coil (c-b-c)
and yes: coil (of length 3) followed by alpha (10 = length < 14)
and yes: coil (of length 1 or 2) followed by alpha (10 = length < 14)
and yes: coil (of length 3) followed by alpha (3 = length < 6)
and no: coil (6 $= length < 10) followed by alpha (of length 1 or 2)
then the function of this ORF is 8/4/0/0 "mitochondrial transport"

Figure 2: *Example of a secondary structure based rule produced using PolyFARM and C4.5*

5. DISCUSSION

- Was the use of ILP required? In many machine learning applications, perhaps most, it is not necessary to use ILP/Relational Data Mining as propositional methods are sufficient. This is because there has been orders of magnitude more work done on propositional methods, and ILP methods do not necessarily act as efficient propositional learners when given wholly propositional data. For example, in bioinformatics propositional methods would empirically seem sufficient to predict protein secondary structure, as neural network approaches have time after time the most successful in blind trials [34]. However, in the prediction of gene function it is very hard to see how the crucial relational aspects of the problem could be encoded efficiently.

- The functional classes for genes exist in hierarchies or directed acyclic graphs. This means that the classes are not independent of each other. Problems with this characteristic are relatively common in the real world (e.g. in text classification), but have been little considered by the statistical or machine learning community [33, 35].
- It is possible for genes to have more than one function, i.e. to have more than one class value. Such problems are also common in the real world and little studied e.g. [33, 36]. Of course, it is always possible to create disjoint classes, but this may distort the problem and create large numbers of artificial classes.

Gene	Predicted Class/ Actual function	Test Set Results
1 YHL048w	06.07 protein modification Nuclear membrane protein, member of a family of conserved, often subtelomerically encoded proteins; regulation suggests a potential role in the unfolded protein response	2/2 (100%)
2 YPR128c	08.04 mitochondrial transport Peroxisomal adenine nucleotide transporter; involved in beta-oxidation of medium-chain fatty acid; required for peroxisome proliferation	1/1 (100%)
3 YGL124c	06.13.01 cytoplasmic and nuclear degradation Protein required for fusion of cvt-vesicles and autophagosomes with the vacuole; associates, as a complex with Ccz1p, with a perivacuolar compartment; potential Cdc28p substrate	1/2 (50%)
4 YHL048w	01.05.07 C-compound, carbohydrate transport Nuclear membrane protein, member of a family of conserved, often subtelomerically encoded proteins; regulation suggests a potential role in the unfolded protein response	8/8 (100%)

Table 2: *Gene functional prediction results for dichotomic search and C4.5.*

ACKNOWLEDGEMENTS

The work was partly funded by the MRC grant G78/6609.

REFERENCES

1. Muggleton, S. H. 1990. Inductive Logic Programming. *New Generation Computing* 8: 295-318.
2. Muggleton, S. H. 1992. *Inductive Logic Programming*. Academic Press, London.
3. Lavrac, N. and Dzeroski, S. 1994. *Inductive logic programming: techniques and applications*. Ellis Horwood, Chichester.
4. Dzeroski, S. and Lavrac, N. 2001. *Relational Data Mining*. Springer, Berlin.
5. King, R. D., Muggleton, S., Lewis, R. A. and Sternberg, M. J. E. 1992. Drug design by machine learning - the use of inductive logic programming to model the structure-activity-relationships of trimethoprim analogs binding to dihydrofolate-reductase. *Proc. Natl. Acad. Sci*. 89: 11322-11326.
6. King, R. D., Muggleton, S. H., Srinivasan, A. and Sternberg, M. J. E. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to

predict mutagenicity by inductive logic programming. *Proc. Natl. Acad. Sci*. 93: 438-442.

7. Finn, P., Muggleton, S., Page, D., and Srinivasan, A. 1998. A. Pharmacophore discovery using the inductive logic programming system Progol. *Machine Learning*, 30: 241-271.
8. Dzeroski S., Blockeel H., Kompare B., Kramer S., Pfahringer B., Van Laer W. 1999. Experiments in Predicting Biodegradability. In*: Proceedings Ninth International Workshop on Inductive Logic Programming*, Vol. 1634 of Lecture Notes in Artificial Intelligence. Springer-Verlag, 80-91.
9. King, R. D., Srinivasan, A., and Dehaspe, L. 2001. Warmr: A Data Mining Tool for Chemical Data. *Journal of Computer-Aided Molecular Design*. 15: 173-181.
10. Srinivasan, A., Page, D., Camacho, R., and King, R.D. 2005 Quantitative pharmacophore models with inductive logic programming. *Machine Learning Journal* (in press)
11. Muggleton, S., King, R. D. and Sternberg, M. J. E. 1992. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering* 5: 647-657.
12. King, R. D., Clark, D. A., Shirazi, J. and Sternberg, M. J. E. 1994. On the use of machine learning to identify topological rules in the packing of beta-strands. *Protein Engineering* 7: 1295-1303.
13. Sternberg, M. J. E., King, R. D., Lewis, R. A. and Muggleton, S. (1994). Application of machine learning to structural molecular biology. *Philosophical Transactions of the Royal Society of London Series B- Biological Sciences* 344: 365-371.
14. Turcotte, M., Muggleton, S. H., and. Sternberg, M. J. E. 2001. Automated discovery of structural signatures of protein fold and function. *J. Mol. Biol*. 306: 591-605.
15. Donescu, A., Waissman, J., Richard, G., and Roux, G. 2002. Characterization of bio-chemical signals by inductive logic programming. Knowledge Based Systems 15: 129-137.
16. Badea, L. 2003. Functional discrimination of gene expression patterns in terms of the gene ontology. In Pac. Symp. Biocomput. 565-576.
17. Hieter, P. and Boguski, N. 1997. Functional genomics: it's all how you read it. *Science* 278: 601-602.
18. Altschul, S. F. *et al.* 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acid Res.*, 25, 3389-3402.
19. Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. and Chothia, C. 1998. Sequence Comparisons Using Multiple Sequences Detect Three Times as Many Remote Homologues as Pairwise Methods. *J. Mol. Biol.*, 284: 1201-1210.
20. King, R. D., Karwath, A., Clare, A., and Dehapse, L. 2000. Genome scale prediction of protein functional class from sequence using data mining. In: The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (eds. R. Ramakrishnan, S. Stolfo, R. Bayardo, and I Parsa) The Association for Computing Machinery, New York, USA. pp. 384-389.
21. King, R. D., Karwath, A., Clare, A., and Dehapse, L. 2000. Accurate prediction of protein class in the M. tuberculosis and E. coli genomes using data mining. *Yeast (Comparative and Functional Genomics)* 17: 283-293.
22. King, R. D., Karwath, A., Clare, A., and Dehapse, L. 2001. The utility of different representations of protein sequence for predicting functional class. *Bioinformatics*. 17: 445-454.
23. Clare, A.J., and King, R.D. 2003. Predicting gene function in Saccharomyces cerevisiae. *Bioinformatics* 19, ii42-ii49.
24. http://www.geneontology.org
25. Ullman, J. D. 1988. *Principles of databases and knowledge-base systems*, vol 1. Rockville, MD: Computer Science Press.
26. Dehaspe, L. and Toivonen, H. 1998. Discovery of frequent Datalog patterns. *Data Mining and Knowledge Discovery*. 3, 7-16.
27. Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. In: *20th International Conference on Very Large Databases* (VLDB).
28. Clare, A., and King, R.D. 2003 Data mining the yeast genome in a lazy functional language. In: *Practical Aspects of Declarative Languages* PADL' 03 19-37.
29. Ferre, S. & King, R.D. 2005 A dichotomic search algorithm for mining and learning domain-specific logics. *Fundamenta Informaticae* (in press)
30. Ferré, S. and Ridoux, O. 2002. A Framework for Developing Embeddable Customized Logics, *Int. Work. Logic-based Program Synthesis and Transformation* (A. Pettorossi, Ed.), LNCS 2372, Springer.

31. Badea, L. 2001. A refinement Operator for Theories, *Inductive Logic Programming*, LNCS 2157, 2001.
32. Quinlan, R. 1993. C*4.5: Programs for machine learning* (Morgan Kaufmann, San Mateo.
33. Clare, A. J. and King, R. D. 2001. Knowledge discovery in multi-label phenotype data. In: The 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01) (Eds. L. De Raedt, A. Siebes) Lecture Notes in A.I. 2168 Sringer-Verlag, Heidelberg.
34. http://predictioncenter.llnl.gov
35. Kohler, D. and Sahami, M. 1997. Hierarchically classifying documents using very few words. In: International Conference of Machine Learning (97) 170-176.
36. Schapire, R. and Singer, Y. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39: 135-168.

Computing Association Rules from Incomplete Support-Counts

Paul Leng
The Department of Computer Science
The University of Liverpool
phl@csc.liv.ac.uk

Abstract An important method of KDD involves the discovery of Association Rules in tabular data. The most computationally difficult part of this is the identification and counting support of the *frequent sets* of attribute-values from which potentially interesting rules can be derived. Algorithms for obtaining frequent sets have been a continuing topic for research, but as yet no wholly satisfactory methods exist to deal with very large databases with densely-populated records. This paper summarises a programme of work which is exploring ways of tackling the problem that begin with an incomplete summation of the necessary support-counts, placing these values in a set-enumeration tree structure. Further possible directions for research are discussed.

1. INTRODUCTION

The extraction of *Association Rules* from binary-valued data is one of the fundamental problems of Data Mining. For the purpose of Association Rule Mining (ARM), we assume that the data is, or can be transformed into, a set of transactions each of which records the values of a set of binary attributes for some instance of the data. We call each non-zero attribute-value an *item* that is *contained* in the transaction. An Association Rule has the form X => Y, where X and Y are disjoint subsets of the set {I} of items defined for the database under consideration. Two properties of the rule are of particular interest: its *support*, and its *confidence*. The support of the rule X => Y describes the number of transactions in the data which contain the *itemset* $X \cup Y$: this may be expressed either as a number, or as a proportion of the total number of transactions. The support, therefore, is a measure of the number of instances in the data for which the association can be observed. The confidence of the rule is the ratio: (Support of $X \cup Y$)/(Support of X), which expresses the frequency of the rule as a proportion of the instances of its antecedent. In general ARM involves the identification in a dataset of all association rules that satisfy some minimum threshold values for support and confidence.

ARM methods invariably proceed in two stages. The first stage identifies, and counts the support of, all subsets of {I} that meet the required threshold of support: we call these the *frequent* sets. Each frequent set of more than a single item defines two or more possible rules. In the second stage, those rules that meet the required confidence threshold are extracted. Because we have already counted the support of each candidate rule, and necessarily of its antecedent, this step is computationally straightforward. The principal problem in ARM, therefore, concerns the identification and support-counting of the frequent sets. The applications of most interest are those for which the number *n* of different items is large, perhaps 1000 or more, so there has been much research towards finding efficient algorithms for this.

The work described here starts from the observation that although computing complete support-counts for all 2^n possible frequent sets is, in general, infeasible, it is easy and fast to perform an incomplete summation of the necessary totals. We have attempted to use this incomplete summation as a first stage in algorithms that complete the ARM task efficiently. I shall here describe the methods we have developed, and discuss some future directions for the work.

2. BACKGROUND

Most methods for finding frequent sets are based to a greater or lesser extent on the "Apriori" algorithm [1]. Apriori performs repeated passes of the database, successively computing support-counts for sets of single items, pairs, triplets, and so on. At the end of each pass, sets that fail to reach the required support threshold are eliminated, and *candidates* for the next pass are constructed as supersets of the remaining (frequent) sets. Since no set can be frequent which has an infrequent subset, this procedure guarantees that all frequent sets will be found.

The performance problems of Apriori are twofold. First, the method requires us to perform multiple database passes: in general, one more than the size of the largest frequent set. Second, as each transaction is examined in pass k, the algorithm must identify all itemsets of size k present within the transaction that are candidates to be counted, and increment the count for each. The cost of this increases both with the number of candidates and, exponentially, with the size of the transaction under consideration. The number of candidates may become very large, especially in the early passes: for example, if all the n items in {I} are individually frequent, then all the $n(n-1)/2$ pairs will be candidates in pass 2, and so on. The problems are particularly acute if the data includes even one very large frequent set. For example, if a set of 50 items occurs often enough to meet the support threshold then not only will the algorithm require at least 50 passes, but all the 2^{50} subsets of this set will necessarily be candidates in some pass.

Approaches to mitigating these problems include methods of partitioning or sampling the data, and algorithms that try to find *maximal* frequent sets without examining all subsets. Partitioning [2] reduces the cost of multiple database passes by performing these on main-memory-resident partitions. The sampling strategy of [3] has a similar effect: here, a memory-resident sample of the data is used to identify likely frequent sets that are verified in (ideally) a single full database pass. Both methods, however, increase the number of candidates that must be considered. Strategies that look ahead to find maximal sets [4,5,6] can be more successful in dealing with long patterns although multiple database passes are still required. In this case, also, it remains necessary to compute support for all subsets of the maximal sets before rules can be determined. A satisfactory solution to these problems remains an open issue in ARM.

3. PARTIAL SUPPORT

In considering how we might address the problem, we begin with the observation that the itemsets under consideration can be visualised as a lattice: Figure 1 illustrates this, for {I} = {A,B,C,D}.

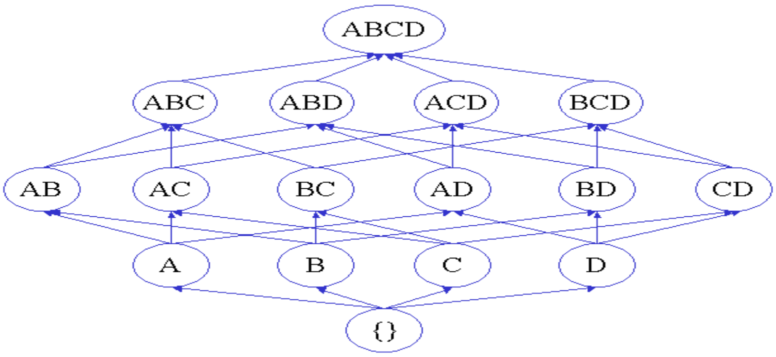


Figure 1: *Lattice of subsets of {A,B,C,D}*

For any itemset i, let $P(i)$ denote the number of transactions that are identical to i. We call this the *Partial support* of i, distinguishing this from the *Total support, T(i),* which is the number of transactions that contain i as a subset. If we imagine that every set in the lattice is annotated with its partial support value, then it is easy to see that the total support for any set i can be computed easily by summing the partial supports in the sublattice comprising i and its supersets. This suggests a way of computing support totals that would begin by organising partial supports in this way, and then complete the task by a cascading summation through the lattice. Unfortunately, there are two difficulties in implementing this: the lattice is exponential in size, and also there is no obvious efficient algorithm for computing support totals for all frequent sets in this way.

We compromise, instead, by representing the subsets in a tree structure. We order the tree lexicographically, so that each subtree contains only lexicographically-following supersets of its root (where the lexicographic labelling corresponds to some chosen ordering of {I}). When building a tree to represent a particular dataset, we will include, in general, only those sets *i* that occur as separate transactions, i.e. for which *P(i)* is non-zero. The tree is constructed in a single pass of the data. As each transaction is examined, the current tree is traversed to locate the position at which it is to be placed. If the set is not already present, a new node is inserted to represent it, and given a support-count of 1. If the corresponding node exists already, its support-count is incremented. If two nodes share a leading subset that is not already in the tree, then a node will be created to represent this. The latter provision is necessary to ensure that a tree, rather than a list, emerges in all cases. Notwithstanding these additional nodes, the final tree is of the same order of size as the original data, and may be smaller if there are many duplicated transactions.

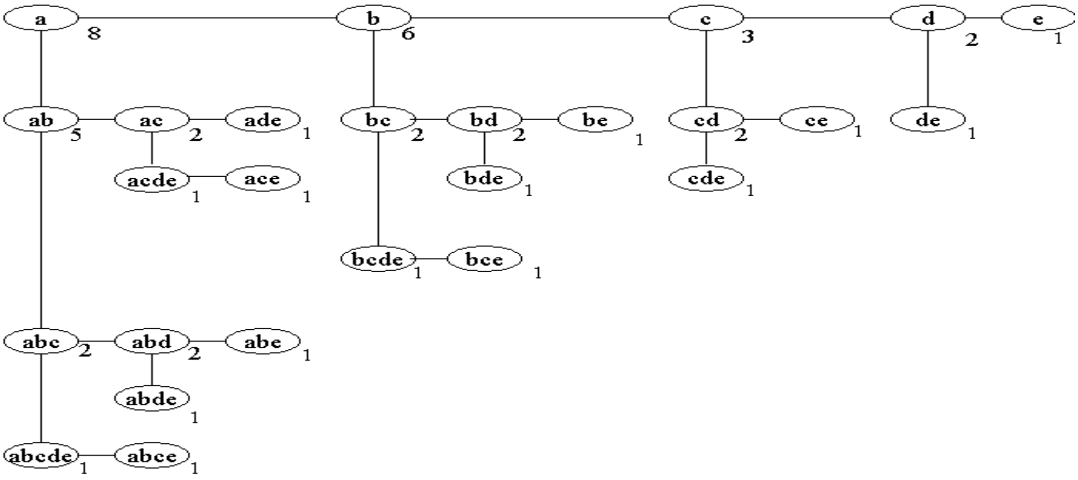


Figure 2: *Example of a P-tree*

Figure 2 illustrates the tree that would result from data comprising the transactions: {abcde,abce,abd,abde,abe,acde,ace,ade,b,bcde,bce,bd,bde,be,cd,cde,ce,d,de,e} (not necessarily in this order). As each transaction is added to the tree, the traversal to locate its position will pass through each node that is a lexicographically-preceding subset of this: for example, if the set cde were to be the last to be added, the traversal would locate this via nodes c and cd. During this traversal, the algorithm increments the count of all these nodes. The annotations of Figure 2 illustrate the counts that would result in this example. These counts represent *interim* support-counts for the corresponding sets; not the complete Total support *T(i),* but a value *Q(i)* that is the sum of the partial supports of lexicographically-following supersets. We use the name *P-tree* to describe this set-enumeration tree of incomplete support-counts constructed in this way [7].

4. COMPUTING TOTAL SUPPORT

The significance of the *P*-tree structure is that we have carried out a large part of the summation of support counts very efficiently in a single database pass. All the information necessary to complete the summation of support is present on the tree. For example, the interim support-count *Q(bd)* stored in the tree of Figure 2 has the value 2, derived from one instance of bd and one of its succeeding superset bde. To obtain the total support-count *T(bd)* we need to add in the support of its preceding supersets bcde, abd (incorporating abde) and abcde.

Various possibilities exist for this summation. We have chiefly experimented with an Apriori-type algorithm that we call Apriori-TFP (Total From Partial). This performs repeated traversals of the *P*-tree to count the total support for single items, then pairs, triples, as for Apriori. During this process a second set-enumeration tree (the *T-tree*) is constructed, level by level, to contain candidates for counting and, finally, the frequent sets. Each new level is built to contain sets all of whose subsets have been found to be frequent. After counting the support of these, infrequent sets are pruned from the tree before the next level is built. The algorithm is described in detail in [8].

The advantage obtained by using the *P*-tree is that, as each node in the tree is examined, we need only consider those subsets that are not covered by its parent. For example, consider the transaction represented by the set abce in our illustration. In pass 2 of the original Apriori, all the 6 pairs that are subsets of abce would be possible candidates whose support would be incremented. In Apriori-TFP, however, the contribution of the transaction abce to the subsets of abc has already been accumulated in the node representing the latter; so, when examining the node abce, we need only consider its subsets ae, be and ce. The advantage gained by this is proportionately greater, of course, for transactions with more items present. The greatest gain is realised when the data includes many transactions with common leading subsets (or, ideally, that are completely identical). This possibility can be maximised by ordering the items by frequency of occurrence.

A number of other researchers have also explored the possibilities of using set-enumeration trees to order candidates for counting. In particular, the *FP*-tree of Han et al [9], developed contemporaneously to our *P*-tree, has a quite similar structure and exploits similar advantages. The *FP*-tree, however, stores only a single item at each node, and includes extra pointers to facilitate the implementation of a specific algorithm, *FP-growth*. The *P*-tree structure, conversely, is more compact and is also generic in that it can be used, in principle, as a basis for many different summation algorithms.

5. OTHER POSSIBILITIES AND RESEARCH DIRECTIONS

5.1. Improved summation algorithms

Our experimental results have shown that Apriori-TFP offers similar or improved performance to the best current algorithms for finding frequent sets, including FP-growth. Nevertheless, it remains a naïve algorithm, sharing many of the drawbacks of the original Apriori. An open question is whether there is a more effective algorithm for exploiting the partial summation contained in the *P*-tree structure. We have examined two strategies: one that builds the *T*-tree in the manner we have described, by repeatedly traversing the *P*-tree first (PTF), and another that builds the *T*-tree iteratively, searching the *P*-tree to obtain support counts for each item added (TTF). Experimental work [10] shows that PTF is generally superior, but TTF is better in certain cases. A hybrid algorithm may be better than either. We are still, however, seeking an algorithm that would exploit the ordering of sets on the *P*-tree more effectively to avoid the multi-pass iteration of Apriori.

Another possible direction is towards the *lazy* evaluation of support totals. The structure of the *P*-tree offers various simple calculations of upper and lower bounds on support totals. For example, from Figure 2 we know the Total support *T(bd)* of set bd cannot be less than *Q(bd)*, and cannot be greater than *Q(bd) + Q(bc) + Q(a)*, because all the supersets of bd are found in the corresponding subtrees. If the latter is an insufficiently tight bound, we can replace *Q(a)* by *Q(ab)*, then by *Q(abc)+Q(abd)*, and so on. These observations suggest possible strategies for eliminating candidates once it is clear that their support is, or cannot be, above the required threshold.

5.2. Partitioned and parallel implementations

ARM is principally concerned with data of very large dimensions, and this often involves partitioning the data into manageable segments for separate processing. The problem with this is that results obtained separately from subsets of the data cannot simply be aggregated. For example, if a set is frequent over all the data, it is necessary to count its support in each partition, including those within which it is infrequent. It is this difficulty that causes problems for the methods [2] and [3].

We can use the *P*-tree and *T*-tree structures, however, to define a partitioning that allows independent processing of each partition. The method [11] requires us to build separate *P*-trees for successive overlapping subsets of {I} across the whole data, which may further be segmented. Each *P*-tree records the totals necessary to compute the support for the sets in some subset of {I}, and can then be used independently to build a *T*-tree for this. Our results show this scales more effectively than simpler partitioning strategies, especially when dealing with very densely-populated data, or very low support thresholds. These are the cases which cause most difficulty in ARM because of the large candidate sets involved.

The partitioning strategy also offers a basis for parallel and/or distributed implementation. Parallel ARM is usually carried out either by *count distribution*, which allocates segments of the data to separate processes for counting, or *candidate distribution,* which allocates subsets of the candidate set to each process [12]. Both methods require extensive inter-process communication, to exchange totals at the end of each Apriori-pass, so that the appropriate candidates can be determined for the next pass. Our tree-partitioning strategy, conversely, makes it possible for each process to complete without the need to communicate with others. The results for this approach [13] demonstrate significantly better performance than for either count or task distribution. Further research is still needed, however, to find the most effective degree of partitioning and/or parallelisation in different cases.

5.3. Constrained rule generation

A number of researchers (e.g. [14, 15]) have explored strategies that use additional constraints on rule generation to reduce the magnitude of the ARM task. One application which can exploit this is the use of ARM techniques for the generation of *classification* rules (CRs). Classification Association Rule Mining (CARM) generally begins by defining class-labels as attributes of the training set of data that is used to generate the CRs. The first stage of the process uses ARM to find a set of rules whose consequents are class-labels. Further processing is then required to reduce and order this set so as to obtain an effective classifier.

CARM differs from ARM in general in that we finally need, not all possible rules that satisfy our threshold requirements, but only a subset of these that is sufficient for a satisfactory classifier. This makes it possible to use the *confidence* threshold during the process of generating frequent sets. We have experimented with an algorithm, TFPC, derived from the general ARM algorithm outlined above [16]. In TFPC, we build a *T*-tree of frequent sets with separate branches for each class-label. As each level is constructed, we calculate the confidence of the rule corresponding to each set that meets the threshold of support. Whenever a rule is found which meets our confidence threshold, we cease generating supersets of that set: i.e., when a sufficiently-confident general rule is found, we stop looking for more specific rules. This greatly reduces the ARM task, and also results in a much smaller set of rules. Our preliminary results show that this approach is both efficient and effective provided the support and confidence thresholds are well-chosen.

6. CONCLUSIONS

Association Rule Mining continues to be a topic that attracts attention by KDD researchers: after more than ten years of work in the field, methods that deal effectively with the task of finding rules within large and densely-populated data are still elusive. We have tried to address this problem using an initial restructuring and partial counting of relevant sets in the form of set-enumeration tree structures. The methods described here fall short of a final solution to the ARM problem, but offer some advantages over existing methods, from which performance gains have been realised. I have here summarised the work carried out so far, as well as some possible avenues for further research.

ACKNOWLEDGEMENTS

All the work described here has been carried out in collaboration with Frans Coenen. We also acknowledge the contributions made to the research by a number of colleagues, research associates, and PhD students, especially Graham Goulbourne, Shakil Ahmed, and Aris Pagourtzis.

REFERENCES

1. Agrawal, R. and Srikant, R. Fast Algorithms for Mining Association Rules. In Proc. of the 20th VLDB Conference, Santiago, Santiago, Chile, pages 487-499, September 1994.
2. Savasere, A., Omiecinski, E. and Navathe, S. An Efficient Algorithm for Mining Association Rules in Large Databases. In Proc. of the 21th VLDB Conference, Zurich, Swizerland, pages 432-444, 1995.
3. Toivonen, H. Sampling Large Databases for Association Rules. In Proc. of the 22th VLDB Conference, Mumbai, India, pages 1-12, 1996.
4. Agarwal, R., Aggarwal, C. and Prasad, V. Depth First Generation of Long Patterns. In Proc. of the ACM KDD Conference on Management of Data, Boston, pages 108-118, 2000.
5. Bayardo, R.J. Efficiently Mining Long Patterns from Databases. In Proc. of the ACM SIGMOD Conference on Management of Data, pages 85-93, 1998.
6. Zaki, M.J. Parthasarathy, S. Ogihara, M. and Li, W. New Algorithms for fast discovery of association rules. Proc Third Int Conf on Knowledge Discovery in Databases and Data Mining, 283-286, 1997.
7. Goulbourne, G., Coenen, F. and Leng, P. Algorithms for Computing Association Rules Using a Partial-Support Tree. J. Knowledge-Based System 13 (2000), pages 141-149. (also Proc ES'99.)
8. Coenen, F, Goulbourne, G and Leng, P. Tree Structures for Mining Association Rules. Data Mining and Knowledge Discovery 8, 25-51, 2004
9. Han, J., Pei, J., Yin, Y. and Mao, R. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8, 53-87, 2004
10. Coenen, F., Leng, P., Pagourtzis, A., Rytter, W and Souliou, D. Techniques for Faster Generation of Frequent Itemsets Using Interim Support Trees. Submitted, 2005
11. Shakil Ahmed, Coenen, F. and Leng, P. A Tree Partitioning Method for Memory Management in Association Rule Mining. In Y Kambayashi, M Mohania, and W Woll (eds) "Data Warehousing and Knowledge Discovery", (Proc DAWAK 2004 conference, Zaragosa): LNCS 3181, Springer, 331-340, 2004
12. Agrawal, R. and Shafer, J.C. Parallel Mining of Association Rules: Design, Implementation and Experience. IEEE Trans. on Knowledge and Data Engineering 8, 1996.
13. Coenen, F, Leng, P. and Shakil Ahmed. T-trees, Vertical Partitioning and Distributed Association Rule Mining. Proc. IEEE International Conference on Data Mining (ICDM 2003), Florida, eds. X Wu, A Tuzhilin and J Shavlik: IEEE Press, 513-516, 2003
14. Bayardo, R, Agrawal, R and Gunopolos, D. Constraint-based Rule Mining in Large, Dense Databases. Proc 15th Int Conf on Data Engineering, 1999.
15. Richards, G. and Rayward-Smith, V.J. Discovery of Association Rules in Tabular Data. Proc IEEE Int Conf on Data Mining (ICDM 2001), IEEE press, 2001
16. Coenen, F, Leng, P and Zhang, L. Threshold Tuning for Improved Classification Association Rules Mining. Proc PAKDD 2005, to appear.

META-HEURISTICS IN THE KDD PROCESS

George D Smith
School of Computing Sciences, UEA Norwich, NR4 7TJ
gds@cmp.uea.ac.uk

Abstract The KDD process defines a sequence of tasks necessary to perform effective data mining. Two stages of the process, namely data pre-processing and mining, are associated with learning algorithms, the former to prepare an effective attribute vector for the latter. The learning algorithms are, by nature, seeking the best solution, whether it be a highly predictive attribute or a model representing a pattern in the data. We demonstrate the use of two meta-heuristic algorithms. The first is simulated annealing to discover rules that represent patterns in the data, the second is the use of genetic programming to discover highly predictive non-linear combinations of attributes for classification.

1. INTRODUCTION

Data mining is a term describing the process of constructing models that represent patterns in the data. These patterns represent relationships between attributes of the dataset, and are thus a form of high level knowledge. If the pattern is unknown, the knowledge can be exploited in a business context. Furthermore, the models induced, once validated, can be used to automate some of the decision-making processes in the organisation, see Figure 1.

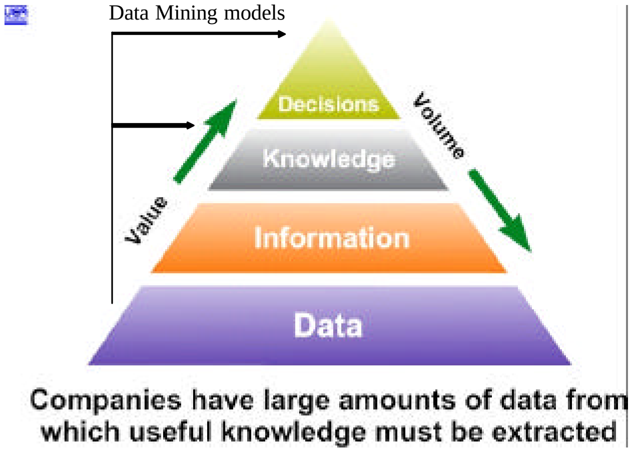

Data Mining models
Decisions
Knowledge
Information
Data
Value
Volume
Companies have large amounts of data from
which useful knowledge must be extracted

Figure 1: *From data to knowledge and decision-making.*

The extraction of patterns draws upon well established statistical methods and upon techniques that have emerged from the discipline of machine learning. These include decision trees models, rule induction models, artificial neural networks and, more recently, the use of meta-heuristic search techniques developed for the solution of combinatorial optimisation problems, see, for example [1].

This paper is an introduction to the application of meta-heuristics in data mining, not only in the extraction of the patterns themselves, but also in the preparation of the data for this task. In Section 2, data mining is described in the context of the entire Knowledge Discovery process, from the initial enquiry through to exploitation of the resulting pattern. In Section 3, we give a brief introduction to meta-heuristics, paying particular attention to those used in the Case Studies described in Section 4. Two case studies are described, namely the use of simulated annealing in rule induction (Section 4.1), and genetic programming in feature construction (Section 4.2).

2. THE KDD PROCESS

Knowledge Discovery in Databases (KDD) is a term describing the process of preparing and mining data extracted from large corporate databases. The objective of the mining is to discover patterns in the data that represent implicit relationships within the data.

The KDD process is made up of a sequence of stages, each comprising recognised tasks that are often necessary to extract useful patterns. These stages are presented in Figure 2, which shows the KDD Roadmap [2], a methodology for KDD.

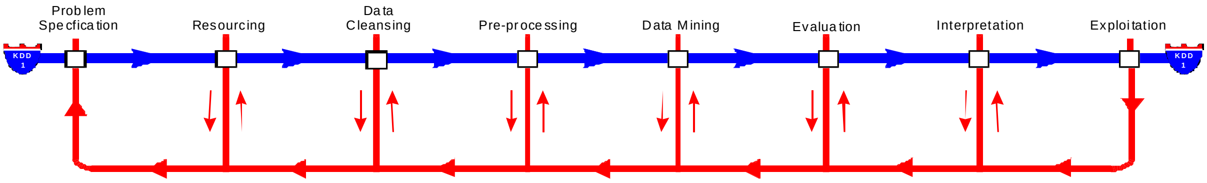


Figure 2: *The KDD Roadmap, a methodology for the KDD process.*

The first stage of the process is the **Problem Specification,** the main objective of which is to agree on a tightly defined specification of the problem. Tasks performed in this stage include preliminary database examination, determination of the required tasks in subsequent stages and a decision as to the feasibility of the project based on the resources required. If feasible, the **Resourcing** stage determines the schedule of activities and the resources, including people, software, hardware and, of course, data.

The next two stages prepare the data for the crucial mining stage. The first, **Cleansing**, describes a set of one-off tasks, such as dealing with errors or outliers, missing data and unbalanced datasets. **Data Pre-processing** is targeted at getting the best out of the attributes remaining in the dataset, whether it be through data reduction or the construction of features with greater predictive power. It comprises three key procedures:

- Feature selection: choosing a powerfully predictive subset of attributes.
- Feature construction: constructing one or more linear or non-linear combinations of the original attributes.
- Discretisation: restructuring an attribute which has a large number of possible values into a smaller number of discrete bins.

A small number of highly predictive fields, together with discretised versions of continuous attributes, facilitates the job of the algorithms used in the data mining stage.

The **Data Mining** stage deals with the actual application of the induction algorithms and the extraction of the patterns. Many different forms of mining exist; these include classification, regression, clustering, association rules, time series analysis, and visualisation. The form or forms chosen for any KDD project should relate to the objectives of the project. Furthermore, for each of these forms, there are a host of different algorithms that can be used. For instance, in classification, one can use tree induction, rule induction, neural networks, statistical models and many more. Finally, for each of these models, there is often a wide choice of algorithms available. For instance, there are 3 commonly used algorithms for tree induction, namely the C5 family, CART and CHAID.

Once the patterns have been extracted, each needs to be evaluated to assign a quality measure to its performance. This measure varies according to the actual model used. For decision trees, it is often just the overall accuracy of the tree when applied to a separate, hold-out set. For rules, we need more than just a measure of accuracy – this is described in more detail in Section 4.1. This is the role of the **Evaluation** stage.

If the patterns generated are of sufficiently good quality, they can then be interpreted in the context of the business to determine their interestingness. The user, or domain expert, plays the major role in this **Interpretation** stage. If the resulting pattern has identified some knowledge heretofore unknown, this pattern is then analysed for potential **Exploitation**.

In practice, however, the KDD process is very seldom as simple as just undertaking a series of tasks in the sequence shown in Figure 2. More often than not, stages have to be revisited and tasks redone with different decisions. Also, some stages are much more time-consuming than others. For instance, experience shows that around 60% of the project time can be taken up by the first four stages.

3. META-HEURISTICS

Meta-heuristics are a family of search algorithms that have been developed to seek optimal solutions to combinatorial optimisation (CO) problems, see [1]. In CO, one seeks for the solution vector $\underline{x}$ that delivers the optimal solution to an objective function $f(\underline{x})$, subject to constraints on $\underline{x}$ and where at least some of the elements of $\underline{x}$ are discrete-valued variables. For most CO problems of interest, the use of exact techniques to determine the optimal solution $\underline{x}^*$ is infeasible, hence approximate techniques are used. Meta-heuristics (MH) are approximate techniques that generally find optimal or near optimal solutions to a whole range of CO problems. Most MHs are variations on the simple local neighbourhood search (LNS) algorithm in which, initially, a random solution is generated. Subsequently, each iteration then considers a small adjustment of this solution (a neighbour). If this is a superior solution, it becomes the current solution and the algorithm proceeds. The problem with such an approach is that it is prone to get stuck in local optima, solutions which are better than all their neighbours but not the best global solution.

Simulated Annealing, one of the earliest to be developed, is a variation on LNS in which the search is allowed to move to inferior solutions with a probability that decreases as the algorithm proceeds. **Tabu search** is another variation on LNS in which the algorithm is forced to keep moving through the solution space, never returning directly to solutions it has recently visited by utilising a memory structure – a tabu list. **Variable Neighbourhood Search** is LNS but where different (increasing) neighbourhoods are utilised when the search gets stuck at a local minimum. The above are referred to as single trajectory methods, since they form a path through the solution space. There are also population-based MHs, such as **Genetic Algorithms**, in which a population of randomly generated solutions is evolved by the use of Darwinian 'selection of the fittest' and a simulation of sexual reproduction. When each solution $\underline{x}$ represents an expression, or program, this is referred to as **Genetic Programming**, see Figure 3. Other population-based MHs include **Ant Colony optimisation** and **Particle Swarm Optimisation** techniques. The reader is referred to [1] for more information.

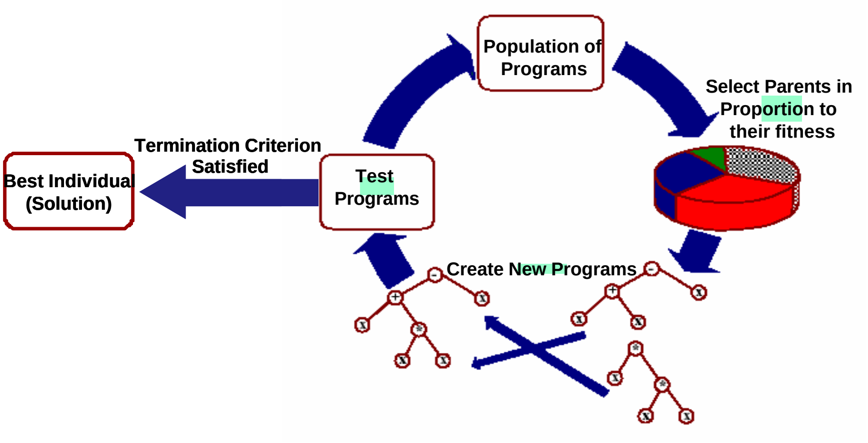

Population of Programs
Select Parents in Proportion to their fitness
Create New Programs
Test Programs
Termination Criterion Satisfied
Best Individual (Solution)

Figure 3: The evolution of programs using genetic programming.

In the following Section, we describe the use of two of the above MHs applied to activities within the KDD process, namely simulated annealing and genetic programming. The reader is referred to [3] for a treatment of genetic algorithms and genetic programming in data mining.

4. CASE STUDIES

4.1. Rule Induction with Simulated Annealing

Rule induction describes a range of techniques used to extract patterns in the form of rules. In data mining, rules induced can be classification rules, or association rules. Classification rules take the form:

IF (*condition*) **THEN** class or a ⇒ ß.

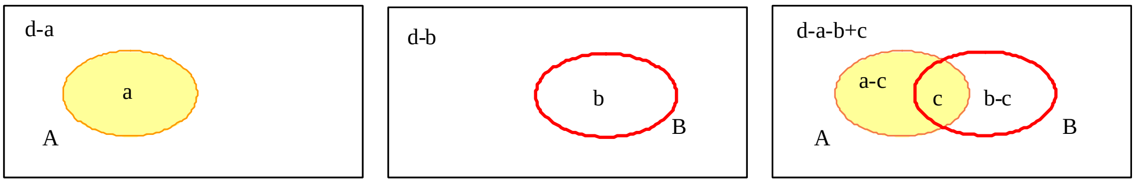
Here, *condition* is either a single condition on one of the predicting attributes (eg, AGE < 25), or a conjunction of single conditions (eg, AGE < 25 **AND** Gender = F). Any record in the dataset that satisfies the condition is then classified by the rule as belonging to *class*. Here, a is referred to as the **antecedent** of the rule, and ß as the **consequent** of the rule. Obviously, there are many rules that can be induced given a data set. The question we ask: what is the best rule to predict membership of a specific class? This in turn leads to the question: what do we mean by the best rule?

Assume we have a dataset, *D*, of d records. Assume that each record in *D* has n attributes (features), F_i, and that each F_i is defined over domain DM_i. For example, F_1 may be a binary-valued attribute so $DM_1 = \{0,1\}$. We are asked to discover a rule of the form a ⇒ ß that appears to hold for some records in *D*. Let r denote a record in *D*.

Associated with any rule a ⇒ ß are three sets of records:

- $A = \{r|\alpha(r)\}$ is the set of records matching the condition and hence classified as belonging to a particular class. Let $a = |A|$.
- $B = \{r|\beta(r)\}$ is the set of records that actually belong in this class. This is fixed for each class/data set. Let $b = |B|$.
- $C = \{r|\ \alpha(r)$ **AND** $\beta(r)\} = A \cap B$. Thus *C* is the set of records that are **accurately** classified by the rule. Let $c = |C|$.

These sets are better understood with reference to Figure 4.



(a) $A = \{r|\alpha(r)\}$ (b) $B = \{r|\beta(r)\}$ (c) $C = \{r|\ \alpha(r)$ Ù $\beta(r)\} = A \cap B$

Figure 4: *In rule induction, the three sets of records associated with a single rule a Þ ß.*

In Figure 4, the set *B* represents the target class. The set *A* denotes those records matching the condition part of the rule and hence predicted to belong to this class, whether correct or not. The set *C* represents those records matching the condition part of the rule and correctly predicted by the rule. We can now introduce 2 quality measures relating to any rule:

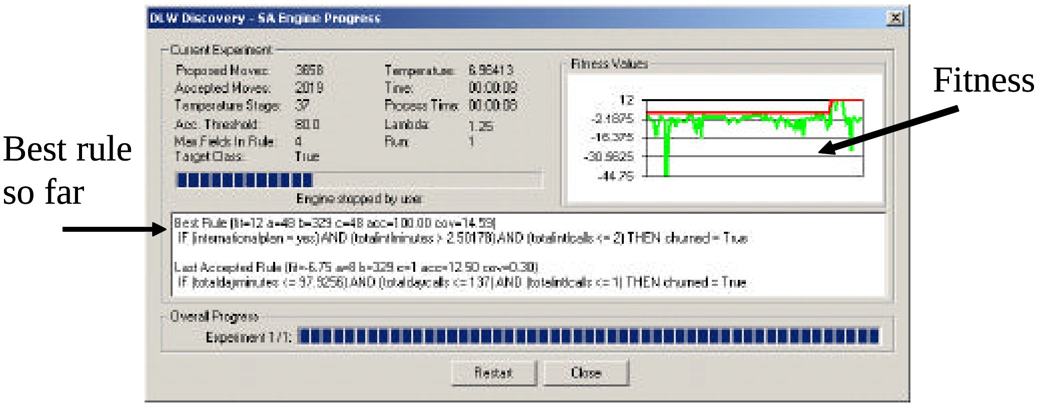
- **Accuracy** is a measure of how often the rule is correct. It is measured by the number of records correctly predicted, expressed as a proportion of the number of records matching the condition part, i.e. accuracy = c/a. (Also referred to as **confidence**.)
- **Coverage** is a measure of the proportion of the target set that is correctly predicted, i.e. coverage = c/b.

Ideally, we would like to discover a rule with 100% accuracy and 100% coverage, i.e. $A = B = C$.

However, in reality, accuracy and coverage are competing objectives. By introducing more conditions into a to increase the accuracy, the rule becomes more specific, covering fewer records. Hence coverage decreases. In the same vein, very general rules can have good coverage but less accuracy. Thus rule induction is, by nature, a multi-objective optimisation problem. Normally, the application domain determines which objective should hold sway, with fraud requiring high accuracy whilst less crucial applications (marketing) are satisfied with a minimum level of accuracy but good coverage.

Consider the following: we wish to increase c, whilst at the same time decrease a-c (number of records the rule misclassifies) and decrease b-c (the number of records belonging to the target class that are not picked up by the rule). With the objective function: maximise f = λ c – a, which we call the **fitness** of a rule, rule induction becomes an **optimisation** problem and we can use the wealth of meta-heuristic search algorithms to search for the best possible rule. Here, λ is a parameter that controls the emphasis of the search. If λ is high, then the emphasis is on rules with high coverage. If λ is small, then the search is for rules with high accuracy.

As part of a TCS Project with Lanner Group, members of the UEA KDD Research group developed a version of simulated annealing to search for optimal rules based on a fitness function f = λ c – a. This is the main discovery engine of the software Witness Miner, a product of the TCS Project. Figure 5 shows a screenshot of the Witness Miner discovery process at work.



Proposed Moves: 3858
Accepted Moves: 2019
Temperature Stage: 37
Acc. Threshold: 80.0
Max Fields In Rule: 4
Target Class: True
Temperature: 6.95413
Time: 00:00:09
Process Time: 00:00:08
Lambda: 1.25
Run: 1

Engine stopped by user

Best Rule (fit=12 a=48 b=329 c=48 acc=100.00 cov=14.59)
IF (internationalplan = yes) AND (totalintlminutes > 2.50178) AND (totalintlcalls <= 2) THEN churned = True

Last Accepted Rule (fit=-6.75 a=8 b=329 c=1 acc=12.50 cov=0.30)
IF (totaldayminutes <= 97.9256) AND (totaldaycalls <= 137) AND (totalintlcalls <= 1) THEN churned = True

Experiment 1/1:

Figure 5: *The Witness Miner rule discovery engine based on simulated annealing.*

Case Study - Derbyshire Police Force / Lanner Group

The project team consisted of Lanner consultants (in process management) and analysts from Derbyshire Police Force. The objectives of the project were to analyse data pertaining to the crime investigation process and to identify patterns (in data collection, decision-making, personnel, etc) that were adversely affecting this process. The project team utilised the KDD Roadmap [2], the software Witness Miner and were given training using their data by members of the UEA KDD research group.

By exploiting some of the findings of the exercise, the team were able to measure the benefits from one quarter of 2003/4 to the same quarter 2004/5. In brief,

- The time to detection was reduced by 68%.
- The detection rate was up by 58%.
- The volume of crime was reduced by 2260.
- The cost of crime was reduced by £2.47m.

The project team won the OR Society President's Gold Medal 2004 for this project and the findings are now in the hands of the Police Standards Unit for roll-out to other forces.

4.2. Feature Construction using Genetic Programming

Feature construction is the process of constructing new attributes which are linear or non-linear combinations of the original attributes. The objective is to discover combinations that have more predictive power. Feature construction is an activity associated with the data pre-processing stage of the KDD process. Tree & rule induction algorithms typically construct each condition on a field-by-field basis. Each test made to decide on the new condition is axis-parallel, i.e. the test (Age < 25) defines a hyperplane parallel to axes of the multidimensional attribute space. There are exceptions to this, however. Oblique classification (OC1) is a tree induction algorithm that considers linear combinations of attributes at each split. However, for any axis parallel test, any **combination** of attributes which presents a much stronger prediction will therefore be missed. The quality of the discovered pattern is therefore restricted by the quality of the raw data and any attributes derived manually from domain expertise.

We apply a GP to construct a single attribute which is a function of the original attribute set. We add this new attribute to the attribute set and compare the performances of the classifiers (three tree induction algorithms C5, CHAID, CART and also an ANN) with and without this constructed attribute. The parameters of the GP are:

v	Terminal set -	{original attribute set, 1}
v	Function set -	{+, -, x, / }
v	Population size	600
v	Tournament selection	(7)
v	Crossover rate 70%, mutation rate 50%	
v	Fitness – Either Information Gain or Gini Index.	

This is the value an attribute would be assigned at a splitting node in the tree induction algorithm C5 or CART respectively, the higher the value, the more predictive power of the (constructed) attribute.

Using 10 fold cross validation, we apply 4 different classifiers to the dataset using the original attribute set and ascertain the accuracies. Repeat this, but include a single attribute evolved using the GP whose fitness function was one of the above measures. This methodology is shown in Figure 6.

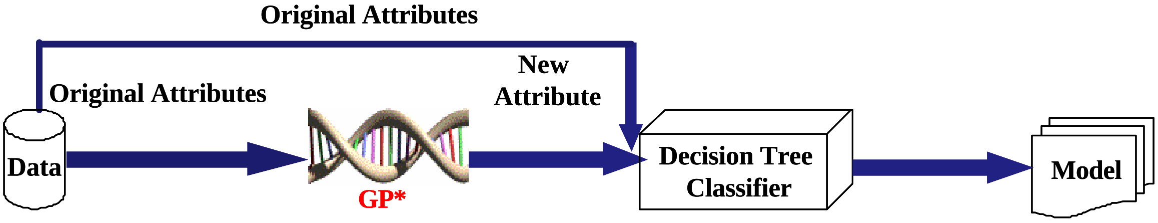
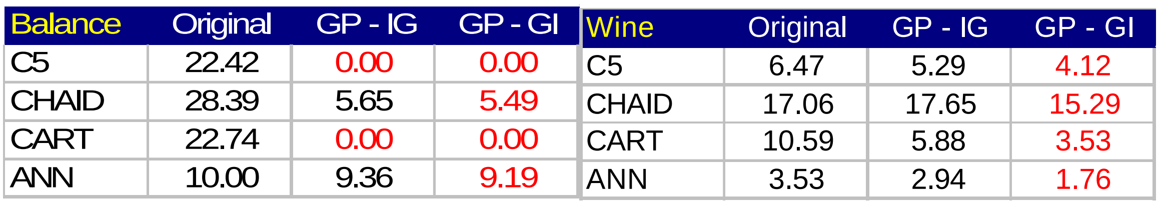


Figure 6: *The methodology of applying GP to feature construction.*

We show the results only for two data sets, see [4, 5] for more details of the full set of experiments. These are the Balance-Scale data set and the Wine data set [6]. The Balance-Scale dataset has 625 records and 3 classes, with 4 predicting attributes, and is an artificially generated data set. The Wine dataset is a real dataset containing 178 records, with 3 classes and 13 predicting attributes. It represents data about wine produced from a single harvest, with the class field representing the quality of the wine. The predicting attributes give the chemical composition and colour characteristics. Although small, we have also applied the GP to datasets of over 5000 records. The resulting error rates are shown in Figure 7.



Balance	Original	GP - IG	GP - GI
C5	22.42	0.00	0.00
CHAID	28.39	5.65	5.49
CART	22.74	0.00	0.00
ANN	10.00	9.36	9.19

Wine	Original	GP - IG	GP - GI
C5	6.47	5.29	4.12
CHAID	17.06	17.65	15.29
CART	10.59	5.88	3.53
ANN	3.53	2.94	1.76

Figure 7: *Error rates on test sets of 4 classifiers using original and augmented attribute sets.*

Firstly, for the Balance-Scale dataset, the augmented attribute set has yielded 100% accuracy on both training and test sets when C5 or CART is used to construct the tree. A significant improvement is also achieved with CHAID. In fact, the same attribute is constructed in all 10 trials by the GP, this is shown in Figure 8. This is precisely the function used to generate the data set.

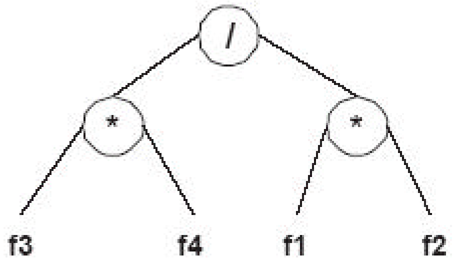


Figure 8: *The evolved attribute for the Balance-Scale dataset.*

Furthermore, the decision tree resulting from the addition of this attribute is reduced dramatically in size, from around 80 nodes to 5 nodes, yielding a much simpler model.

A similar result was obtained with the Wine dataset. Although not attaining 100% accuracy for all trials, C5 and CART did achieve 100% accuracy on the test sets in some trials. One evolved variable in particular accurately separated Class 2 (intermediate quality) from Classes 1 and 3. Subsequently, one of the original variables was able to distinguish between Classes 1 and 3 with 100% accuracy. This remarkable result needs further analysis: what does this evolved variable signify, if anything? Nevertheless, the outcome is accurate classification of the quality of the wine from a machine learning model. Put another way, analysis of the evolved variable may make it possible to adjust the chemical composition to move the wine from one class to a higher class.

REFERENCES

[1] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3): pages 268-308, 2003.

[2] J.C.W. Debuse, B. de la Iglesia, C.M. Howard, and V.J. Rayward-Smith. Building the KDD roadmap: A methodology for knowledge discovery. In R. Roy, editor, *Industrial Knowledge Management*, pages 179-196. Springer-Verlag, 2001.

[3] A.A. Freitas. Evolutionary Computation. In: J. Zytkow and W. Klosgen. (Eds.) Handbook of Data Mining and Knowledge Discovery. Oxford University Press, 2001.

[4] M. A. Muharram and G. D. Smith. Evolutionary feature construction using information gain and gini index. In M. Keijzer, et al., editors, *7th European Conf. on Genetic Programming, EuroGP 2004, Portugal*, LNCS, no 3003, pages 379-388. Springer Berlin, 2004.

[5] M. A. Muharram and G. D. Smith. A comparison of GP fitness functions in evolutionary feature construction. 2004. (Accepted for IEEE Trans KDE.)

[6] www.ics.uci.edu/~mlearn/MLRepository.html

Experiments in Hypertext Categorization

Houda Benbrahim and Max Bramer

Department of Computer Science and Software Engineering

University of Portsmouth, UK

{houda.benbrahim, max.bramer}@port.ac.uk

Abstract This paper looks at (i) which extra information hidden in HTML tags and linked neighbourhood pages to take into consideration to improve the task of automatic classification of web documents and (ii) how to deal with the high level of noise in linked pages. A hypertext dataset and four well-known learning algorithms (Naïve Bayes, K-Nearest Neighbour, Support Vector Machine and C4.5) are used to exploit the enriched text representation. The results show that the clever use of the information in linked neighbourhood and HTML tags improves the accuracy of the classification algorithms.

1. INTRODUCTION

It has been estimated that the World Wide Web comprises more than 3 billion pages and is growing at a rate of 1.5 million pages a day. Faced with such a huge volume of documents, search engines become limited: too much information to look at and too much information retrieved. The organization of web documents into categories will reduce the search space of search engines and improve their retrieval performance. A recent study showed that users prefer to navigate through directories of pre-classified content and that providing a categorised view of retrieved documents enables them to find more relevant information in a shorter time. The common use of the manually constructed category hierarchies for navigation support in Yahoo and other major web portals has also demonstrated the potential value of automating the process of hypertext categorization.

Text categorization is a relatively mature area where many algorithms have been developed and experiments conducted. Classification accuracy reached 87% for some algorithms applied to known text categorization corpora (Reuters, 20_newspaper…) where the vocabulary is coherent and the authorship is high. Those same classical classifiers perform badly on samples from Yahoo! pages. This is due to the extreme diversity of web pages (such as homepages, articles…) and authorship, and to limited consistency in vocabulary.

Automated hypertext categorization poses new research challenges because of the extra information in a hypertext document. Hyperlinks, HTML tags, metadata and linked neighbourhood all provide rich information for classifying hypertext that is not available in traditional text categorization. Researchers have only recently begun to explore the issues of exploiting rich hypertext information for automated categorization.

There is a growing volume of research in the area of learning over web text documents. Since most of the documents considered are in HTML format, researchers have taken advantage of the structure of those pages in the learning process. The systems generated differ in performance because of the quantity and nature of the additional information considered.

Benbrahim and Bramer used the BankSearch dataset to study the impact of the use of metadata (page keywords and description), page title and link anchors in a web page on classification. They concluded that the use of basic text content enhanced with weighted extra information (metadata + title + link anchors) improves the performance of three different classifiers.

Oh et al. reported some observations on a collection of online Korean encyclopaedia articles. They used system-predicted categories of the linked neighbours of a test document to reinforce the classification decision on that document and obtained a 13% improvement over the baseline performance when using local text alone.

Joachims et al. also reported a study using the WebKB university corpus, focusing on Support Vector Machines with different kernel functions. Using one kernel to represent one document based on its local words, and another kernel to represent hyperlinks, they give evidence that combining the two kernels leads to better performance in two out of three classification problems.

Yang, Slattery and Ghani have defined five hypertext regularities which may hold in a particular application domain, and whose presence may significantly influence the optimal design of a classifier. The experiments were carried out on 3 datasets and 3 learning algorithms. The results showed that the naïve use of the linked pages can be more harmful than helpful when the neighbourhood is noisy, and that the use of metadata when available improves the classification accuracy.

This paper deals with web document categorization. Two issues will be considered in depth: (i) the choice of representation for documents and the extra information hidden in HTML pages and its neighbourhood that should be taken into consideration to improve the classification task, and (ii) how to filter out the noisy neighbourhood. Finally, data collected from the web will be used to evaluate the performance of different classification methods with different choices of text representation.

Document representation is described in Section 2. Some classification algorithms used for hypertext are reviewed in Section 3. Section 4 presents experiments and results, comparing different classification algorithms with different webpage representation techniques.

2. TEXT REPRESENTATION

In order to apply machine-learning methods to document categorization, consideration first needs to be given to a representation for HTML pages. An indexing procedure that maps a text into a compact representation is applied to the dataset. The most frequently used method is a *bag-of-words* representation where all words from the set of documents under consideration are taken and no ordering of words or any structure of text is used. The words are selected to support classification under each category in turn, i.e. only those words that appear in documents in the specified category are used (the *local dictionary* approach). This means that the set of documents has a different feature representation (set of features) for each category. This approach for building the dictionary has been reported to lead to better performance. This leads to an attribute-value representation. Each distinct word corresponds to a feature, with the number of times the word occurs in the document as its value.

Based on our preliminary work, metadata (page keywords and description), page title and link anchors in a web page along with basic page content improved the accuracy of the classification task. This extra information is included in the data dictionary. Next, the words in the page's neighbours (documents pointing to, and pointed to by the target page) are included in the text representation. The blind use of the content in links may harm the classification task, this is due to the fact that many pages point to pages from different subjects, e.g., web pages of extremely diverse topics link to Yahoo! or BBC news web pages. To filter out the noisy link information, just the most similar adjacent pages to the target page are kept. The similarity of pages is determined by the cosine measure.

With the bag-of-words approach for text representation, it is possible to have tens of thousands of different words occurring in a fairly small set of documents. Using all these words is time consuming and represents a serious obstacle for a learning algorithm. Moreover many of them are not really important for the learning task and their usage can degrade the system's performance. There are many approaches to reducing the feature space dimension. The most common ones are: (i) the use of a stop list containing common English words, (ii) or the use of stemming, that is keeping the morphological root of words.

3. CLASSIFICATION ALGORITHMS

(a) Naïve Bayes (NB)

Naïve Bayes (NB) is a widely used model in machine learning and text classification. The basic idea is to use the joint probabilities of words and categories in the training set of documents to estimate the probabilities of categories for an unseen document. The term 'naïve' refers to the assumption that the conditional probability of a word is independent of the conditional probabilities of other words in the same category.

A document is modelled as a set of words from the same vocabulary, *V*. For each class, C_j, and word, w_k Î *V*, the probabilities, $P(C_j)$ and $P(w_k|C_j)$ are estimated from the training data. Then the posterior probability of each class given a document, D, is computed using Bayes' rule:

$$P(C_j \mid D) = \frac{P(C_j)}{P(D)} \prod_{i=1}^{|D|} P(a_i \mid C_j)$$

where a_i is the i^{th} word in the document, and |*D*| is the length of the document in words. Since for any given document, the prior probability *P(D)* is a constant, this factor can be ignored if all that is desired is ranking rather than a probability estimate. A ranking is produced by sorting documents by their odds ratios, $P(C_1|D) / P(C_0|D)$, where C_1 represents the positive class and C_0 represents the negative class. An example is classified as positive if the odds are greater than 1, and negative otherwise.

(b) K-Nearest Neighbour (KNN)

K-Nearest Neighbour (KNN) is a well-known statistical approach in pattern recognition. KNN assumes that similar documents are likely to have the same class label. Given a test document, the method finds the *K* nearest neighbours among the training documents, and uses the categories of the *K* neighbours to weight the category candidates. The similarity score of each neighbour document to the test document is used as the weight of the categories of the neighbour document. If several of the K nearest neighbours share a category, then the per-neighbour weights of that category are added together, and the resulting weighted sum is used as the likelihood score of that category with respect to the test document. By sorting the scores of candidate categories, a ranked list is obtained for the test document. By thresholding on these scores, binary category assignments are obtained. The decision rule in KNN can be written as:

$$y(\vec{x},c_j) = \sum_{d_i \in KNN} sim(\vec{x},\vec{d_i})\, y(\vec{d_i},c_j) - b_j$$

where $y(\vec{d_i},c_j) \in \{0,1\}$ is the classification for document $\vec{d_i}$ with respect to category cj (y = 1 for Yes, and y = 0 for No); $sim(\vec{x},\vec{d_i})$ is the similarity between the test document $\vec{x}$ and the training document $\vec{d_i}$; and b_j is the category specific threshold for the binary decisions.

(c) C4.5 (Decision Tree Classification)

C4.5 is a decision tree classifier developed by Quinlan. The training algorithm builds a decision tree by recursively splitting the data set using a test of maximum gain ratio. The tree is then pruned based on an estimate of error on unseen cases. During classification, a test vector starts at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the subtree rooted at the new node until a leaf is encountered, at which time the pattern is asserted to belong to the class named by that leaf.

(d) Support Vector Machine (SVM)

Support vector machines are based on the Structural Risk Minimization principle from computational learning theory. The idea is to find a hypothesis *h* for which we can guarantee the lowest true error. The true error of *h* is the probability that *h* will make an error on unseen and randomly selected test examples. An upper bound can be used to connect the true error of a hypothesis *h* with the error of *h* on the training set and the complexity of H (measured by VC-Dimension), the hypothesis space containing *h*. Support vector machines find the hypothesis *h*, which minimizes this bound on the true error by effectively and efficiently controlling the VC-Dimension of H. A remarkable property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space. This property makes them suitable for text categorization where the dimension of the input space is very high.

4. EXPERIMENTS

(a) Dataset

To test the proposed algorithms for hypertext classification, datasets were needed that reflected the properties of real world hypertext classification tasks.

The major practical problem in using web document datasets is that most of the URLs become unavailable. The well-known dataset WebKB project at CMU is outdated since most of its web pages are no longer available.

The BankSearch dataset used for the experiments comprises a set of HTML web documents. The Open Directory Project and Yahoo! categories were used to provide web pages that have already been categorized by people. The dataset considered consists of 11,000 pages distributed over 11 categories under 4 distinct themes. The dataset consists of some sets of categories that are quite distinct from each other, as well as other categories that are quite similar. Table I gives a summary of the dataset.

Dataset ID	Dataset Category	Associated Theme
A	Commercial Banks	Banking and Finance
B	Building Societies	Banking and Finance
C	Insurance Agencies	Banking and Finance
D	Java	Programming Languages
E	C/C++	Programming Languages
F	Visual Basic	Programming Languages
G	Astronomy	Science
H	Biology	Science
I	Soccer	Sport
J	Motor Sport	Sport
K	Sport	Sport

Table I. *Dataset Summary*

(b) Performance Measures

The evaluation of the different classifiers is measured using four different measures: recall (R), precision (P), accuracy (Acc), and F1 measure. These can all be defined using the 'confusion matrix' shown as Table II.

	Correct Class is C_k	*Correct Class is* $\overline{C_k}$
Assigned class is C_k	A	b
Assigned class is $\overline{C_k}$	C	d

Table II. *Confusion Matrix*

$$R = \frac{a}{(a+c)} \text{ if } (a+c) > 0 \text{ otherwise } R = 1$$

$$P = \frac{a}{(a+b)} \text{ if } (a+b) > 0 \text{ otherwise } P = 1$$

$$Acc = \frac{(a+d)}{n} \text{ where } n = a + b + c + d > 0$$

$$F1 = \frac{2PR}{(R+P)} = \frac{2a}{(2a+b+c)} \text{ if } (a+c) > 0 \text{ otherwise undefined.}$$

Recall (R) is the percentage of the documents for a given category that are classified correctly. Precision (P) is the percentage of the predicted documents for a given category that are classified correctly. Accuracy (Acc) is defined as the ratio of correct classification into a category C_k.

Neither recall nor precision makes sense in isolation from the other. In fact, a trivial algorithm that assigns class C_k to all documents will have a perfect recall (100%), but an unacceptably low precision. Conversely, if a system decides not to assign any document to C_k it will have a perfect precision but a low recall. The F1 measure has been introduced to balance recall and precision by giving them equal weights.

Classifying a document involves determining whether or not it should be classified in any or potentially all of the available categories. Since the four measures are defined with respect to a given category only, the results of all the binary classification tasks (one per category) need to be averaged to give a single performance figure for a multiple class problem.

In this paper, the 'micro-averaging' method will be used to estimate the four measures for the whole category set. Micro-averaging reflects the per-document performance of a system. It is obtained by globally summing over all individual decisions and uses the global contingency table.

(c) Design of experiments

The classification algorithms NB, KNN, SVM and C4.5 were applied to the BankSearch dataset to address the different binary classification problems. The dataset was randomly split into 70% training and 30% testing.

Two local dictionaries were then built for each category and for each text representation after stop word removal (using a stop list of 512 words provided by David Lewis), with the option of stemming turned either on or off. Documents were represented by a VSM where the weights were the term frequencies in documents.

Two series of experiments were conducted. The documents are represented by (i) the basic content of HTML documents or (ii) a combination of basic HTML content, metadata, title and link anchors of the target page, along with weighted content of the similar in-coming and out-going linked pages.

The local dictionaries and the document's VSM for the second option of text representation were constructed as follows. For each target page in the dataset, the set of neighbour pages was determined. The content of all the in-coming and out-going links along with the target page was used to build the dictionaries. Then, the similarity of each target page with its neighbours was calculated to filter out the noisy links. The term weights of the target pages were adjusted so that the target page was influenced by its similar neighbours.

(d) Results and interpretation

The pages considered in this specific dataset have on average 16.4 out-going links**,** with this number varying from a maximum of 189 to a minimum of 1. This number also varies depending on the category considered. Concerning the in-coming links, the average number of pages was 7, with a maximum of 456 and a minimum of 0. Many target pages in the dataset were not pointed to by any document in the web. An interesting remark was drawn while determining the similar pages to a given target page; the average number of similar pages (including both in-coming and out-going pages) was 5, with a minimum of 0 and a maximum of 36 (those numbers vary depending on the considered category). As a result, a large number of linked pages were thrown away. This explains clearly the fact that linked neighbourhood is noisy. This filtering step was helpful in this regard.

The different algorithms result in different performance depending on the features used to represent the documents.

The set of experiments evaluates SVM, C4.5, NB and KNN, for texts represented using either (i) the basic content enhanced by the meta data, title and link anchors with stemming option turned on or off, or (ii) a combination of basic content, metadata, title and link anchors of the target page with those of its similar neighbours where extra weight was assigned to common words between the target page and its neighbours, this is done with the stemming option turned on or off.

Figure 1 (Figure 2) reports the performance accuracy (F1 measure) on the test set of SVM, C4.5, NB and KNN for the different text representation options.

Figures 1 and 2 show that the use of stemming improves the performance of all the classifiers for all the options of text representation. Stemming is helpful since it decreases the size of the feature space. Moreover, it combines the weights of the different words that share the same stem.

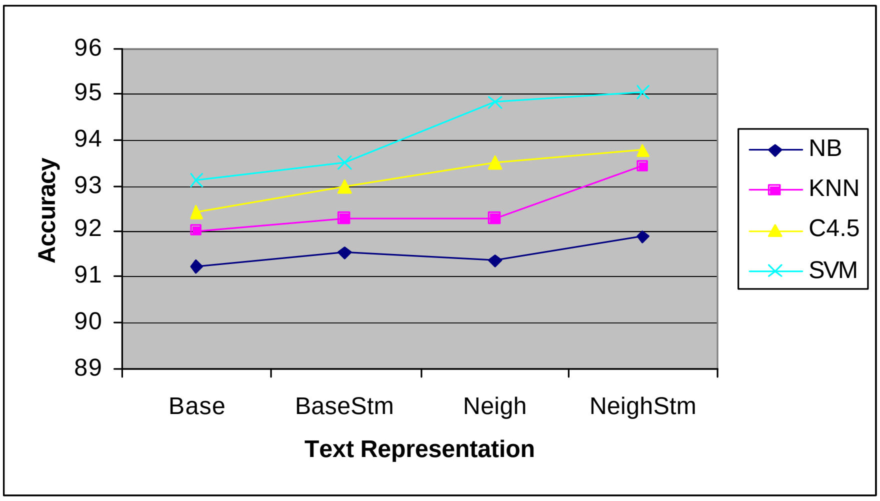


Figure 1: *N.B, KNN, C4.5 and SVM accuracy for different choices of text representation*

These figures also show that SVM outperforms all the other classifiers. This is not surprising since it has been reported that it works well in high dimensional feature space. This also explains its slight increase in performance when stemming was used. C4.5 also outperforms NB and KNN for the different text representations. The features selected by C4.5 to build the tree were meaningful in terms of class description.

Including the extra information in the filtered neighbourhood to the basic pages has improved the performance of the different classifiers. Note that if the threshold used to decide about the similarity of two pages is set too high no similar documents will be found and the performance of the classifiers, with the linked neighbourhood information taken into consideration for text representation, will be as good as that of the classifiers with the basic text content as text representation. The slight increase in each classifier's performance when the linked information is used means that all the noisy links that may harm the classification were filtered out. The threshold used in those experiments to decide about the similarity of two pages was set to 0.8. This threshold may seem too high since it may decline even useful links, but at the same time, somewhat secure since there is a low chance of including noisy links.

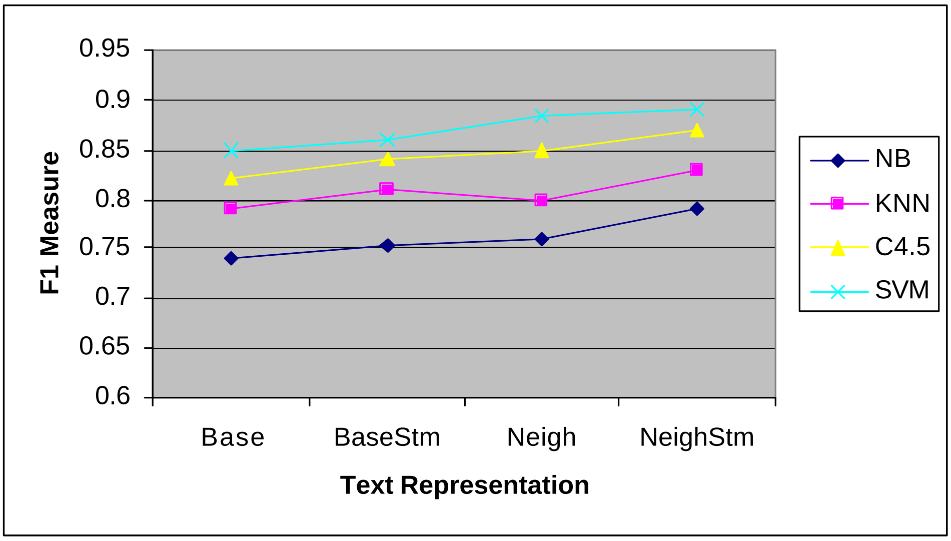


Figure 2: *N.B, KNN, C4.5 and SVM F1 measure for different choices of text representation*

5. CONCLUSIONS AND FUTURE WORK

In summary, a number of experiments were conducted to evaluate the performance of some well-known learning algorithms on hypertext data. Different text representations have been used and evaluated. It can be concluded that the careful use of the extra information available in the linked neighbourhood increases the performance of the classifiers. The improvement was smaller than expected since the filtering was too high, and useful links might have been filtered out.

The careful use of the extra information in the linked neighbourhood of HTML pages improved the performance of the different classifiers. In future work, this extra information will be extended by less severely selecting the useful neighbour links. The class of the linked neighbourhood instead of its similarity to the target page may be used to filter out the noisy links. Experiments with different datasets should also be conducted before final conclusions are drawn.

This paper is a slightly shortened version of [1], which includes a full set of references.

[1] Benbrahim, H. and Bramer, M.A. (2004). Neighbourhood Exploitation in Hypertext Categorization. *In* Research and Development in Intelligent Systems XXI, Springer-Verlag, 2005.

THE SPECIALIST GROUP ON AI (SGAI)

`http://www.bcs-sgai.org/sgai/sgai.htm`

What is SGAI?

SGAI is the British Computer Society (BCS) Specialist Group on Artificial Intelligence. The group was founded in June 1980 by Professor Donald Michie, an AI pioneer since the 1960s at the University of Edinburgh and a war time colleague of Alan Turing. Although the Group's original focus was on rule-based Expert Systems, it now covers all areas of Artificial Intelligence and has played a valuable role in the development of these fields in Britain over the last 20 years. The group has been a member of the European Co-ordinating Committee on Artificial Intelligence (ECCAI) since 1992. It has organised an annual international conference since 1981. A substantial proportion of the Group's membership is from industry and it thus provides a valuable forum in which the academic and industrial AI communities can meet.

Mission Statement

The Group's mission is: "*To foster achievement, capability and awareness in both business and research in Artificial Intelligence, and to promote the interests of the related community*".

Scope

The Group covers all areas of artificial intelligence (AI) and its applications. These include, but are not limited to:

- Expert and knowledge-based systems
- Knowledge representation, inference and reasoning
- Machine learning (including neural nets, genetic algorithms, case-based reasoning and data mining)
- Qualitative and temporal reasoning
- Knowledge management
- AI languages and environments
- Natural language processing
- Intelligent agents

Knowledge Engineering Review

SGAI was responsible for launching the Knowledge Engineering Review, which is now an independent journal published by Cambridge University Press and edited by Dr Simon Parsons (University of Liverpool). There is a reduced individual subscription rate for Group members.

List Server

SGAI has established a list server named AI-SGES on the Mailbase system at the University of Newcastle to facilitate communication on all aspects of Artificial Intelligence.

Subscribers can send messages for automatic transmission to all other subscribers in a simple way and have access to a range of other services. The service is open to all (whether members of BCS-SGAI or not) and is free of charge. Subscribers can unsubscribe at any time. To join the AI-SGES Mailbase, simply send an email message to:

mailbase@mailbase.ac.uk

containing two lines of text. If your name were Alan Turing, these lines would be:

JOIN AI-SGES ALAN TURING
STOP

This will subscribe you and automatically generate the transmission to you of the necessary documentation of the Mailbase service.

Web Site

The SGAI WWW site can be found at:

http://www.bcs-sgai.org

Monthly Mailings

SGAI members receive a monthly mailing of information concerning events such as conferences and workshops, research funding opportunities and other relevant material. To include items in the monthly mailings prices are as follows:

The above prices are in UK pounds and exclude VAT. For further information contact Rob Milne at Intelligent Applications Ltd. (Email: rmilne@cix.compulink.co.uk).

Activities

BCS-SGAI has organised an annual series of international conferences since 1981. The proceedings of the technical and applications streams are published in book form since 1984 initially by Springer Verlag as the "Research and Development in Intelligent Systems" and "Applications and Innovations in Intelligent Systems" series.

External Relations

The Group has links with many outside bodies. It is a specialist group of the British Computer Society and a member of ECCAI, the European Co-ordinating Committee for Artificial Intelligence.

Since its inception the group has enjoyed a good working relationship with government departments involved in the Artificial Intelligence field (beginning with the Alvey Programme in the 1980s). A succession of Department of Trade and Industry (DTI) representatives have been co-opted as committee members. The Group acted as co-organiser of the annual DTI Manufacturing Intelligence awards and has included sessions presenting the results of the DTI Intelligent Systems Integration Programme (ISIP) in its annual conferences.

The group also has a good relationship with the Institution of Electrical Engineers (IEE), with which it has co-sponsored colloquia over many years, and with NCAF, the Natural Computing Applications Forum. We also host the annual UK CBR (Case-Based Reasoning) workshops at our annual conferences.

Joining the SGAI

Both individual and Corporate Membership are available. Note that **you do not have to be a member of the BCS to join the SGAI**. For further information see the SGAI www site.

Subscription rates

Annual subscription rates for individual and corporate membership of SGAI are as follows:

<u>Individuals</u>	
Standard Members (UK addresses)	£31
Standard Members (Overseas addresses)	£41
BCS Members (UK addresses)	£22
BCS Members (Overseas addresses)	£32
Students* (UK addresses)	£11
Students*(Overseas addresses)	£21
Retired (UK addresses)	£11
Retired (Overseas addresses)	£21
<u>Corporate Members</u>	
(UK addresses)	£150
(Overseas addresses)	£190

*Students should supply evidence of status

Members' Discounts

The group has been able to use its extensive network of links to win substantial discounts for its members (see the SGAI web site for details (http://www.bcs-sgai.org).

BCS-SGAI

British Computer Society Specialist Group on Artificial Intelligence

Annual subscription rates for Individual and Corporate Members are:

INDIVIDUALS

Standard Members (UK addresses)	£31.00
Standard Members (Overseas addresses)	£41.00
BCS Members (UK addresses)	£22.00
BCS Members (Overseas addresses)	£31.00
Students[2] (UK addresses)	£11.00
(Overseas addresses)	£21.00
Retired (UK addresses)	£11.00
(Overseas addresses)	£21.00

CORPORATE

(UK addresses)	£150.00
(Overseas addresses)	£190.00

Add £5 to all these rates if not paying by standing order.

Please note that it is not necessary to be a member of the BCS to join SGAI

Tick here if you do not want your details in the ECCAI Membership Directory [ ]
Tick here if you do not want your email address added to the AI-SGES Mailbase [ ]

Your details will be held in computer readable form on the SGAI database to allow the membership secretary to: a) more easily manage the BCS SGAI mailings to members b) more easily analyse the type of members the SGAI has and c) more easily manage the membership subscriptions.

Please complete in BLOCK LETTERS and return with subscription to:

SueTueton
SGAI Membership Secretary, The British Computer Society,
1 Sanford Street, Swindon, SN1 1HJ
(Tel: +44 (0)1793 417416, Email: STueton@hq.bcs.org.uk)

[Cheques should be made payable to the BCS SGAI]

NAME: …………………………………… BCS NO: (if applicable) …………………..……………..

COMPANY NAME: ……………………………………………………………………………..

SUBSCRIPTION ADDRESS: ……………………………………………….…..……………...

…………………………………………………………………………………….……...……….

POSTCODE: ……………………… TEL. NO: ………………………. EMAIL: ……….……………..

AMOUNT ENCLOSED: £…………..…… MEMBERSHIP CATEGORY: …………………………..

SIGNATURE: …………………………..………………… DATE: …………………………………

[2] Proof of status is required.