A Case Study in Knowledge Acquisition for Insurance Risk Assessment using a KDD Methodology[*]

Graham J. Williams and Zhexue Huang
CSIRO Division of Information Technology
GPO Box 664 Canberra ACT 2601 Australia
Email: Graham.Williams@cbr.dit.csiro.au

Abstract

We describe some initial experiences in dealing with the task of acquiring knowledge where a very large collection of case histories is available. A Knowledge Discovery in Databases (KDD) approach is taken. KDD is the process of extracting novel information and knowledge from large databases, consisting of many interacting stages performing specific data manipulation and transformation operations with an information flow from one stage onto the next (and usually with feedback into previous stages). We characterise our experiences of this process for the task of acquiring knowledge for the domain of motor vehicle insurance premium setting for NRMA Insurance Limited.

Keywords: Knowledge acquisition, knowledge discovery in databases, data mining, insurance premiums, risk analysis, fraud.

1 Introduction

Knowledge Discovery in Databases (KDD) is commonly defined as the "non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad, Piatetsky-Shapiro and Smyth 1996). KDD technology combines techniques from a variety of related disciplines, notably Databases, Artificial Intelligence, Statistics, Scientific Discovery, and Visualisation. KDD is indeed identical to none of them but rather draws upon the research and technology from *all* of them. A particular focus of KDD is on extremely large real world training datasets often of sizes measured in giga-bytes and tera-bytes. The sheer size alone eliminates many existing techniques (from statistics, machine learning, and knowledge acquisition) for automatically, let alone manually, analysing the data.

With such a wealth of data available a knowledge engineering exercise in extracting interesting and useful knowledge needs to be carefully guided by the domain experts. We relate in this paper our experience in using a KDD process to assist in the task of knowledge acquisition when very large training datasets are available.

The KDD process is recognised as consisting of a number of interacting, iterative, stages involving various data manipulation and transformation operations. Information flows from one stage onto the next and also backwards to previous stages. Fayyad, Piatetsky-Shapiro and Smyth (1996), for instance, identify 9 steps in the KDD process whilst Williams and Huang (1996) identify a simpler, yet still sufficient model of the process involving 4 primary stages.

Most attention within the KDD community has focused on the Data Mining (algorithmic exploration of the data) stage of the process. It is critical, however, to recognise that Data Mining is only one part of the whole process (Brachman and Anand 1996). Anecdotal evidence (and our

[*] Presented at PKAW96, the Pacific Rim Knowledge Acquisition Workshop, Sydney, Australia, October 1996.

own experience) suggests that the other stages account for up to 95% of the effort—it is important then to understand the *whole* KDD process. Our project brings together a team of researchers in Databases, Machine Learning, Statistics, and Visualisation, to perform KDD research and development with industry partners who provide domain expertise and real world databases. A particular focus of the project is the use of high performance computers and parallel algorithms. Domains ranging from insurance fraud to astronomy have been studied.

In this paper we first present a model of the KDD process which identifies the necessary and sufficient elements of the process and supports the variety of operations required for knowledge acquisition using KDD. Section 2 identifies some of the issues that characterise KDD. We then present a view of the KDD process in Section 3. Sections 4–7 then illustrate the various aspects of the model in the context of an actual case study from insurance risk analysis performed with NRMA Insurance Limited, one of Australia's largest general insurers.

2 Characteristics of KDD Applications

What distinguishes KDD from related research areas such as Knowledge Acquisition, Machine Learning, and Statistics is the size of the training dataset available, the complexity of that data, and the expected results, rather than by the particular methods and algorithms used. But more than this, KDD is viewed as an all encompassing process for the discovery of knowledge in databases. KDD attempts to deal with real world problems and hence real world data. The source data for KDD is often extracted from databases which are generally not built with KDD in mind. The data must be cleansed and moulded into a form suitable for the particular KDD task. It must then be transformed into a format that the particular knowledge acquisition or machine learning tools can work with. Some of the issues which need to be addressed within the KDD context include noise, redundant information, missing values and attributes, large data sets and sparse data (Frawley, Piatetsky-Shapiro and Matheus 1992, Matheus, Chan and Piatetsky-Shapiro 1993).

Redundancy can result from the inclusion of attributes and records in the source data which are irrelevant or superfluous to the data mining. Determining which attributes and records are redundant is usually difficult. Removing apparently irrelevant attributes can lead to a reduction in the types of new knowledge that can be discovered. In insurance risk analysis, for example, the office at which an insurance application was lodged may seem irrelevant, yet could be a particularly interesting risk factor. Leaving truly redundant attributes in the data, though, can lead to aberrant results or could significantly impact upon the time taken to run the data mining algorithms.

Redundancy can also occur when multiple instances in the database represent the same object, but perhaps recording some different information. For many data mining algorithms this is not a desirable situation—often the algorithms assume independence between records. A typical example is when the source database simply records transactions performed by clients. While some data mining algorithms, such as the association algorithms of Agrawal and Srikant (1994), work directly with such data, many require the data to be entity (in our case, insurance policy) oriented.

The problem of missing attributes often arises because of the desire to use more information in data mining than the original designers of the database considered necessary. The missing data can only sometimes be obtained, and then at a cost.

The volume of data available for a KDD task is an obvious problem, but one which often leads to the related problem of sparse data. Data can be sparse, for example, when there are many missing values for various attributes. It can also be sparse when the concept of particular interest in the data does not occur very often. This is the case, for example, in using KDD to assist in fraud detection in large insurance databases, where instances of fraud are usually relatively rare.

Another issue which complicates KDD relates to its exploratory nature. Often in starting a KDD task it will not be clear exactly what is to be discovered. In general there is a broad idea of what is expected but as the process proceeds and results are generated the directions taken may change quite dramatically. Although the nature of the data may lead us to use particular techniques, it is not always clear a priori which tools will produce the best results. Indeed, the use of a variety of tools and techniques often leads to an interesting variety of results.

3 The KDD Process

We view the process of knowledge discovery from databases as consisting of the four stages identified in Figure 1. This covers the 9 steps identified by Fayyad, Piatetsky-Shapiro and Smyth (1996) into a smaller number of higher-level stages to more simply identify the whole process. It is important though to also appreciate the complexity of the process.

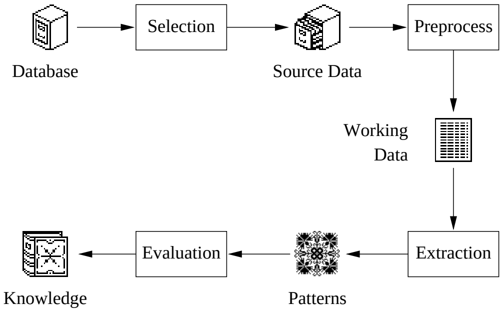

Figure 1: The four stage KDD process.

In most cases we begin with an original Database, usually developed for tasks other than KDD. After due consideration (which can require initial exploratory runs through the KDD process to gain better insights into what is required from the Database) an appropriate collection of Source Data is extracted from the Database. This Source Data forms the basis for the rest of the KDD process.

3.1 Data Pre Processing

The purpose of the Preprocessing stage is to cleanse the data as much as possible and to put it into a form that is suitable for use in later stages. The types of Preprocessing that a KDD environment needs to support includes traditional database projection and selection, and value mapping and classification functions. Data cleansing type operations, such as operations to resolve fuzzy matches between entities in the data that actually represent the same entity, are also useful.

The original Database is generally stored and maintained separately from the KDD process. Control and access to this original Database is usually restricted (by legislation and commercial concerns) due to the confidential nature of the material of interest. The Selection stage then is usually performed by the data owners, with input on data requirements from the KDD team.

The Source Data often requires cleansing as part of the KDD process and may be transformed in many ways to turn it into suitable Working Data. These operations are part of the arsenal represented in the set of operations $\mathcal{S}$.

Meta-knowledge about the Source Data is also important in KDD. This includes semantic information provided by the schema, domains of attributes including types and value ranges, distributions of attribute values, and relations between attributes. This meta-knowledge is usually obtained from domain experts or can be calculated directly from the data. It is often considered as part of the background knowledge.

3.2 Knowledge Extraction

Once a suitable Working Dataset has been produced the knowledge extraction phase begins. A variety of tools to explore different types of patterns in the Working Data can be explored. Some tools produce information that is used by other tools (e.g., statistical tools used to determine characteristics of the data required as input parameters for machine learning). The extraction stage itself is often cyclic. In Statistics it is often represented as a cycle between the three stages of: model identification; parameter estimation; and diagnostic checking of the model fitted.

The variety of techniques used in extraction include clustering, decision tree induction, neural networks, genetic algorithms, and a variety of statistical algorithms. These operations are usually oriented towards data description (as in clustering) or model fitting (as in classification). Classification rules are often extracted using these techniques and can be used for prediction on unseen objects. Various visualisation tools which include methods and algorithms for viewing the internal representation of the data in various human interpretable forms are also useful. These tools enable close involvement of the domain experts in *all* of the KDD stages—domain experts are often more sensitive to subtleties of the data than automated algorithms.

3.3 Pattern Evaluation

Using a variety of tools and even tuning a single tool in a variety of ways leads to the discovery of many different patterns. The fourth stage of KDD involves an evaluation of the patterns discovered in order to find those that give rise to useful and novel knowledge. Evaluation is a non-trivial task.

A Pattern Evaluation function is used to evaluate the interestingness of discovered knowledge from the user's viewpoint. KDD requires that the discovered knowledge be useful in some sense. The Pattern Evaluation function governs the selection of the patterns which are of interest to the user and has been identified by others as an integral part of KDD (Frawley et al. 1992, Matheus et al. 1993, Fayyad, Piatetsky-Shapiro and Smyth 1996). Pattern evaluation is also important in reducing the search space (Holsheimer, Kersten, Mannila and Toivonen 1995).

Formally, a Pattern Evaluation function $\mathcal{F}$ is a function that maps from a set of statements expressed in some language $\mathcal{L}$ (e.g., production rules) to a set of (usually) numeric values. The evaluation might consider each pattern in the context of all discovered patterns, or patterns might be evaluated for their usefulness, novelty, and validity in the context of the application domain. Evaluation functions are often quite complex and application specific.

Domain experts often provide the most effective form of evaluation of discovered patterns. Visualisation tools are a critical component of this Evaluation, providing effective presentations of the results of mining. In such cases there is little need nor desire to implement algorithmic evaluation functions. Where they can be automated it is advantageous to do so to assist in automatically processing the large collection of discoveries common in the KDD context.

Evaluation functions can be generic, as are those that are expressed in terms of the data mining tools being used (e.g., a good pattern is one that minimises the misclassification cost of the discovered rules). Alternatively they can be specific to an application, where the usefulness of a rule might be measured in terms of the associated claim costs in an insurance application.

3.4 The KDD Model

The KDD process as described above represents the typical scenario for those involved in knowledge discovery from databases, even though the details of the process may differ for different KDD applications. The key elements of the process are: the data; the background knowledge and the discovered patterns and knowledge; the method(s) for evaluating discovered patterns; and the collection of operations associated with the different stages of the process. The basic model is thus identified as a four tuple $< \mathcal{D}, \mathcal{L}, \mathcal{F}, \mathcal{S} >$ consisting of: database $\mathcal{D}$; knowledge representation language $\mathcal{L}$; evaluation functions $\mathcal{F}$; and operations $\mathcal{S}$.

It is important to emphasise that for the successful application of the KDD process both domain expertise and KDD expertise are crucial, and that the KDD process is very much an iterative process where earlier stages often need to be refined to address “discoveries” made in later stages.

4 Case Study: Insurance Risk Analysis

Assessing the performance of an insurance portfolio requires both overall and within-portfolio analyses. The overall analysis demonstrates whether the portfolio in the past was profitable or not, whereas a detailed within-portfolio analysis reveals which particular areas within the portfolio made significant profits and loses (Coutts 1984). The latter information is particularly important for a company in order to maintain both its competitiveness in the market and its profitability. Without knowing the areas of significant risk in its portfolio an insurer will be unable to sustain operation, even though the overall portfolio may perform well temporarily. A portfolio should balance its exposure to risk.

The overall analysis of an insurance portfolio can be performed using straight forward statistical techniques, based on the total premiums earned and the total of the claims paid. More sophisticated approaches are required for within-portfolio analysis (Brockman and Wright 1992, Coutts 1984).

A common approach to within-portfolio analysis is to partition the risk of the whole portfolio into small areas of risk. These are identified by a set of risk rating factors usually represented by a contingency table. Only a few variables are usually considered at a time (the claim frequency, the claim cost, and the exposure, for example) for each cell of the contingency table.

When a portfolio is partitioned a model can be fitted to each cell relating the level of the risk rating factors to the claim frequency and claim cost. Historical data is used to estimate parameters of the model. The model can then be used to predict the expected claim frequencies and claim costs for different risk rating factor levels.

With such an approach though the risk rating factors must be categorical. A continuous factor has to be categorised into a small number of levels. A trade-off is needed in the detail of the analysis and fit of the model. Also, the interaction between factors is often ignored because of the difficulty with handling them. With many variables and with categorical variables with many values, the number of possible interactions is very large.

Alternative approaches to the task of insurance risk analysis have primarily been explored within the statistical/actuarial domains. Siebes (1994), however, considered the problem of insurance risk analysis in the context of data mining by employing probability theory. Insurance claims on policies in any one year were viewed in terms of Bernoulli experiments. This work lead to the idea of developing equal probability, homogeneous descriptions of classes of the insurance portfolio.

An important goal in our exploration of the KDD process for real world knowledge acquisition tasks is to involve expert intuition as an integral aspect of the process, supported by formal analyses.

5 Insurance Portfolio Databases

A portfolio database in an insurance company contains a set of insurance policies purchased by customers. A policy insures up to a specified value a particular entity (motor vehicle, household contents, buildings) from loss. When damage or loss occurs to the insured entity a claim is made against the policy for compensation. A policy is valid for a certain time period and is usually renewed upon payment of the premium. The period during which a policy is active is referred to as the period of exposure (as in the insurer's exposure to risk) and is usually measured in days. At any particular time the exposures associated with active policies in the portfolio database will differ depending upon their dates of validity.

A key factor to the success of an insurance portfolio is balancing the trade-off between the setting of competitive premiums and covering the risk associated with the exposure. The insurance market is very competitive and setting premiums too high leads to a loss of market share. Yet setting premiums too low leads to loss of profit. Premiums are generally based on a small number of key factors (such as age of driver, type of vehicle, and address of owner) identified by various analyses and intuitions. Because of the size of many insurance portfolios, analysis is often restricted to straightforward techniques.

The performance of a portfolio is usually assessed in the context of the previous year's data. The analyses performed on the data are used by the underwriter to envisage the near future performance of the portfolio and to adjust the policy rating structure to reflect changes in the market and in the behaviour of the insured. Essentially the analyses are used each year to tune the rules which set premiums for the coming year.

There are two extremes in setting premiums: all policies attract the same premium; or each individual policy has an individual premium determined for it based on all of the details of the policy. Neither extreme is particularly useful nor practical. The former would likely penalise those who are less likely to lodge claims in favour of those who a more likely to do so, and would probably drive them to the competitors! The latter would generally be impractical because of the complexity of the rules that would be required. However, the goal of good insurance premium setting is likely to lean towards the latter than to the former. An insurer would prefer to fine tune their premium setting more so than is often currently performed, identifying more factors that affect the risk and taking more information into account in setting premiums.

6 Preprocessing

Starting with the data extracted from the source database maintained by NRMA Insurance Limited (consisting of many millions of policies), a number of transformations had to be performed before a suitable Working Dataset was built. This involved many iterations, some of which were performed after the initial knowledge extraction tasks were performed. (The initial runs indicated different directions to take, different data to obtain, and different transformations to be performed.)

The Source Dataset was extracted from the NRMA's motor vehicle insurance portfolio database. An initial trial dataset of some 125,000 records, covering a time period of only 6 months (1 July 1994 to 31 December 1994), was used prior to working with the full dataset. Each record contained 93 fields of data. Figure 2 summarises the preprocessing performed.

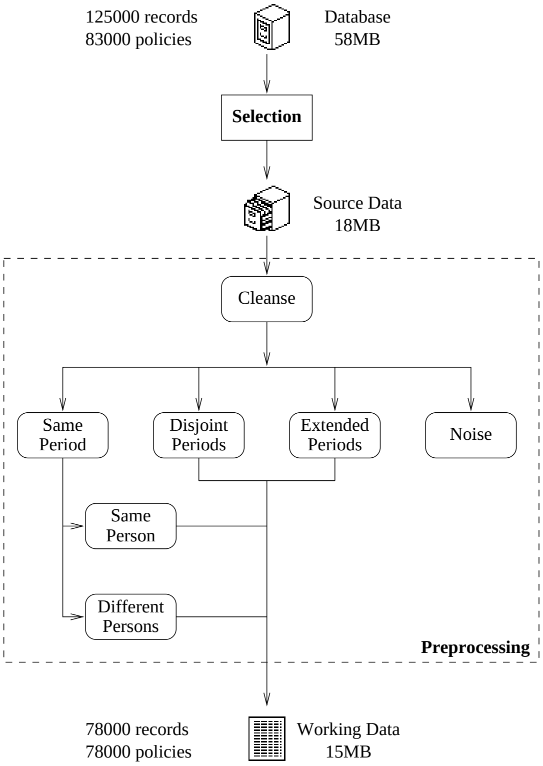

Figure 2: Preprocessing.

The first task was to remove from the database those fields/attributes/variables which were irrelevant to the task at hand. This process was complicated by the fact that some “obviously” irrelevant attributes (e.g., office at which application for insurance was lodged) could contain surprising information. On the other-hand, leaving irrelevant attributes in the data set can lead to aberrant results. A panel of KDD experts (from Machine Learning, Statistics, and Databases) and domain experts (from the NRMA) considered each attribute in turn, removing only those that were seen as clearly irrelevant by all.

The second task, labelled “Cleansing” in Figure 2, involved performing various transformations on the data (e.g., birth dates transformed into ages). Various computed fields were also added to the data at this stage. One, for example, calculated the period of exposure associated with the policy in the context of the period of interest (last six months of 1994).

The third task involved collapsing the transaction-oriented data that was supplied into a policy-oriented dataset which is required for the types of analyses intended to be performed. The original database was oriented towards managing the insurance portfolio–it was not designed with Data Mining in mind. A transaction in the original database records some change made to, or incident associated with, a policy. Hence, in the 125,000 records, there were only about 83,000 policies. Some policies had multiple owners (hence multiple records) while others had a variety of changes, such as renewals, cancellations, etc. Other policies appeared more than once when there was more than a single claim made on the policy in the period of interest.

A careful study of the dataset had to be made to understand all of its nuances. This finally resulted in the definition of a set of rules to merge the multi-record policies into single policy records. These rules were implemented as a variety of operations and were tuned iteratively as the data and problems became better understood.

An intermediate dataset was created with some 78,000 individual policies, each described by 43 attributes, some of which were generated in the merging process to record, for example, the number of owners of each individual policy and the total amount of all claims made against a policy. Policies that were cancelled or met other criteria (e.g., contained critical but missing data) were also removed from the dataset.

The intermediate dataset was intended to be used for multiple data mining objectives (e.g., fraud detection). Consequently, some of its attributes were still irrelevant to the risk analysis and thus further processing was performed to build the final risk analysis Working Dataset. This dataset contained some 75,000 unique policies, each being represented by 23 attributes, including exposure and total claims cost.

An initial observation that has some impact upon the type of analysis performed was that of the 75,000 policies, only about 4,000 of them actually had claims against them. While not particularly surprising, such a low relative occurrence (about 5%) of an important aspect of the data does require attention (e.g., appropriate use of prior probabilities), but is beyond the immediate scope of this paper.

A second observation of relevance is that different policies have different exposures, even though the majority have exposure for the whole period. This indicates that the exposure should be used to weight the examples in some way. This important aspect of the data is also left for future exposition.

7 Risk Analysis

We now describe the process of extracting knowledge for use in this domain.

7.1 Decision Trees

An implementation of the classification and regression tree software CART (Breiman, Friedman, Olshen and Stone 1984) was used in the initial experiments to extract interesting patterns. Due to the large number of training examples a parallel implementation by Thinking Machines Corporation, called StarTree and part of the Darwin toolkit (Thinking Machines Corporation 1995), was used. CART and StarTree are similar in many ways to the traditional machine learning decision tree induction algorithm C4.5 (Quinlan 1993).

StarTree consists of three operations: *grow tree*; *evaluate tree*; and *apply tree*. The *grow tree* stage actually implements the usual divide-and-conquer strategy in a parallel architecture for building a full decision tree based on a training set. The *evaluate tree* stage evaluates the generated tree in the context of a test dataset. It is here that pruning is performed and a variety of test datasets can be used to explore the performance of the tree in the context of pruning. The *apply tree* stage simply applies any resulting trees to new unseen data. StarTree provides a variety of output options, including a LaTeX formatted decision tree or a collection of rules.

7.2 Methodology

The claim cost attribute in the final database was selected as the target attribute (or dependent variable) and the remainder (excluding exposure) as the independent variables. All policies were classified into two classes: those with claims and those with no claims—the claim cost was effectively reclassified as 1 or 0. The dataset was processed with *grow tree* to produce a complete decision tree.

Using StarTree a rule base can be built from a generated decision tree. A rule base consists of a set of rules, each corresponding to a different path through the tree representing a conjunction of the conditions associated with the nodes of the tree. An example rule might be:

If age $\leq$ 20
and sex = Male
and insured_amount $\geq$ 5000
and insured_amount $<$ 10000
Then insurance_claimed = 1, cost = 0, (0, 15)

The rule indicates that under the given conditions an insurance policy is claimed against (and the cost of misclassification associated with this rule is 0). There are 15 true examples from which this rule was generated.

The primary rule base generated by StarTree is now extended with information derived from other sources, including the Source Dataset and original database. Significant areas of risk were explored by viewing those leaf nodes containing claims. The frequency of claims at any leaf together with the total of the claim costs at the leaf nodes provide important information. However, since claims cost had been factored out of the training set used by StarTree to build the decision tree, this information was incorporated as an extra step in the process. The rules are thus extended with information that records the total exposure and the total claim cost for each rule. Other information such as the total of the premiums associated with the "area of risk" could also have been determined.

The generation of this extra information is an important step, particularly in assessing the usefulness and appropriateness of the discovered knowledge. The additional information can be used to determine whether the information embodied as a rule is useful as knowledge. Such an exploration of the rules discovered turns this data mining operation, in a sense, back onto itself. This is particularly the case where we are dealing with very large datasets with many attributes from which very large trees (and hence very many rules) are generated without pruning (and hence over fitting). The generated rules themselves need to be data mined (in the context of this evaluation stage).

7.3 Initial Results

Some initial results from the analysis of the NRMA Insurance Limited sample database are presented here. Due to the commercial nature of the data the actual variables will be referred to as `F01`, `F02`, . . ., `F23`.

For illustration, three decision trees were generated using different selection measures for the partitions—we will refer to the three trees as the entropy, gini, and error trees. (Refer to Mingers

Tree name:	Entropy	Gini	Error
Decrease Function	entropy	gini	error
Number of Nodes	10215	10887	8197
Depth	43	38	39
Total leaves	5062	5391	3331
Positive leaves (Pl)	1967	2188	1192
Negative leaves (Nl)	3095	3203	2139
Policies by Pl	3815	3795	1682
Policies by Nl	71993	72013	74126

Table 1: Using alternative decrease functions.

(1989) for a discussion of selection measures.) An idea of the size and organisation of the trees produced from the sample data is gained from the statistics of the various trees grown (Table 1).

It is clear from Table 1 that very large trees are generated given the large number of attributes (and possible values) and the large number of examples. The row in the table giving the total number of leaves corresponds also to the number of rules generated from the tree. Similarly the positive leaves and the negative leaves. The Entropy tree, for example, generates 1967 rules for claims.

Although 21 independent variables were used in the input data, only 19 were selected by the Entropy and Gini trees and 20 by the Error tree. Table 2 lists the selected variables and the frequencies with which the variables appear in each tree. Thus, the variable F01 appears 111 times in the Error tree. Such information can be of use in gaining some insights into which variables appear to be important. (The location of the variable within the tree and the type of the variable—categorical versus continuous—are other important indicators.)

Variable	Entropy	Gini	Error
F02	0	0	0
F01	0	0	111
F14	465	840	60
F21	668	711	52
F19	1756	1367	119
F22	2223	2177	85
F18	2258	2034	374
F12	2667	2589	289
F16	3013	2461	82
F07	3051	2947	1407
F15	3543	3122	189
F13	4629	4928	224
F20	4843	5175	19
F17	6164	6197	1023
F04	6773	6743	4041
F10	6799	6431	3975
F06	8760	9122	10195
F08	10771	11772	20842
F09	11362	10200	7895
F03	12073	12349	11464
F11	12329	12989	1480

Table 2: Attribute frequencies.

Table 3 records the number of claims associated with the rules generated from each of the three trees, aggregated by the number of claims. There are 1090 rules from the Entropy tree, for example, which have just a single claim. On the other hand, there are 36 claims associated with one of the rules from the Gini tree. Those rules associated with a higher number of claims would tend to indicate areas of high risk.

Tables 4 summarises the rules from each tree with more than 14 claims against them or with a high average claim cost with respect to the exposure. For each such rule the tables record the number of claims, the total amount of exposure, and the sum of the claim costs. Each row represents a single rule. In the Gini sub-table, for example, there are 36 claims associated with a single rule. These claims cover 442 days of exposure and have a total claim cost of $159,472. Such information can again be used to target areas described by the rules for further explorations.

Armed with such "post learning" analyses, areas of significant insurance risk can be identified and further investigated. Such investigations could be expected to lead to a better understanding of insurance risk and to a finer tuning of insurance premium setting.

Claims	Number of Rules		
	Entropy	Gini	Error
1	1090	1493	941
2	494	404	150
3	192	135	48
4	82	51	27
5	38	31	7
6	15	16	7
7	18	17	3
8	9	7	3
9	5	6	2
10	5	5	1
11	4	5	1
12	2	4	
13	4	5	
14	4		2
15	2	1	
16	2	1	
17		2	
18		1	
22	1	2	
24		1	
36		1	

Table 3: Claims associated with each risk area.

Gini

Claims	Exposure	Cost
11	221	139673
15	1868	74367
16	2513	54750
17	2635	33152
17	2637	52721
22	2305	70558
22	3839	86988
24	3399	98918
36	442	159472

Entropy

Claims	Exposure	Cost
15	1745	43308
15	197	55921
16	2198	50213
16	2471	52201
22	265	85678

Error

Claims	Exposure	Cost
10	349	27827
14	2183	49883
14	2023	49889

Table 4: High claim risk areas.

8 Summary

In this paper we have followed the KDD process and highlighted its application to the domain of motor vehicle insurance risk assessment as an adjunct to the usual knowledge acquisition process. StarTree was used to analyse the large insurance dataset, generally building very large decision trees (and hence large rule sets). The rules were used only as an aid in the “discovery” of interesting areas of the data. By understanding such hot spots, better insights into policy premium setting can be gained.

We have identified a simple yet sufficient breakdown of the KDD process involving four interacting and iterative stages, and have presented an associated model of KDD identifying the key elements of the process. An example knowledge acquisition exercise using the KDD methodology for insurance risk analysis has been used to illustrate the process and the model.

Our ongoing work is exploring alternative approaches for dealing with the very large datasets that are generally available for applications such as insurance. An approach whereby large datasets are first clustered and where the resulting clusters are subjected to decision tree induction has been found to simplify the knowledge acquisition task. Even with high performance computers which can handle larger problems, such considerations remain necessary.

Acknowledgements

The authors acknowledge the support provided by the Cooperative Research Centre for Advanced Computational Systems (ACSys) established under the Australian Government's Cooperative Research Centres Program. This work has been performed within the Divisions of

Information Technology (DIT) and Mathematics and Statistics (DMS) of the Australian Governments' Commonwealth Scientific and Industrial Research Organisation (CSIRO). Contributions made by Peter Milne of DIT and Murray Cameron, Glenn Stone, Petra Kuhnert, and David Chan of DMS are also acknowledged. Industry collaborators from NRMA Insurance Limited include Keith Forster, Philip Woods, and Hong Ooi.

References

Agrawal, R. and Srikant, R.: 1994, Fast algorithms for mining association rules in large databases, *VLDB '94*.

Brachman, R. J. and Anand, T.: 1996, The process of knowledge discovery in databases: A human-centered approach, *in* Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy (1996), chapter 2.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J.: 1984, *Classification and regression trees*, Wadsworth, Belmont, CA.

Brockman, M. J. and Wright, T. S.: 1992, Statistical motor rating: Making effective use of your data, *Journal of Institute of Acturaries* **119**, 457–543.

Coutts, S. M.: 1984, Motor insurance rating: An acturarial approach, *Journal of Institute of Acturaries* **111**, 87–148.

Fayyad, U. M., Piatetsky-Shapiro, G. and Smyth, P.: 1996, From data mining to knowledge discovery: An overview, *in* Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy (1996), chapter 1.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (eds): 1996, *Advances in Knowledge Discovery and Data Mining*, AAAI Press.

Frawley, W. J., Piatetsky-Shapiro, G. and Matheus, C. J.: 1992, Knowledge discovery in databases: An overview, *AI Magazine* **13**(3), 57–70.

Holsheimer, M., Kersten, M., Mannila, H. and Toivonen, H.: 1995, A perspective on databases and data mining, *Proc. of the First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, pp. 150–155.

Matheus, C. J., Chan, P. K. and Piatetsky-Shapiro, G.: 1993, Systems for knowledge discovery in databases, *IEEE Transactions on Knowledge and Data Engineering* **5**(6), 903–913.

Mingers, J.: 1989, An empirical comparison of selection measures for decision-tree induction, *Machne Learning* **3**(4), 319–342.

Quinlan, J. R.: 1993, *C4.5: Programs for machine learning*, Morgan Kaufmann.

Siebes, A.: 1994, Homogeneous discoveries contain no surprises: Inferering risk-profiles from large databases, *Technical Report CS-R9430*, CWI.

Thinking Machines Corporation: 1995, The Darwin solution: A family of prediction and classification tools for large databases, *Technical report*, Thinking Machines Corporation.

Williams, G. J. and Huang, Z.: 1996, Modelling the KDD process: A four stage process and four element model, *Technical Report TR-DM-96013*, CSIRO Division of Information Technology, available from Graham.Williams@cbr.dit.csiro.au.