Modelling the KDD Process*

A Four Stage Process and Four Element Model

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

February 1996

Abstract

Knowledge Discovery in Databases (KDD) is the *process* of extracting novel information and knowledge from large databases. This process consists of many interacting stages performing specific data manipulation and transformation operations with an information flow from one stage onto the next (and often back into previous stages). The process can be very complex and may exhibit much variety in the context of the variety tasks undertaken within KDD. In this paper we characterise our experiences of the KDD process and formalise its key elements in a model. A case study of insurance risk analysis for policy premium setting is used to illustrate the process and the model. The model provides a framework for comparing and differentiating various approaches to KDD.

Keywords: Knowledge discovery in databases, data mining, process, model, insurance premiums, risk analysis, fraud.

1 Introduction

Knowledge Discovery in Databases (KDD) is 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 databases often of sizes measured in giga-bytes and tera-bytes. The sheer size alone eliminates many existing techniques for analysing data.

KDD applications in the real world can be as diverse as the real world databases that exist today. The requirements for the KDD process in the variety of application domains, and even within the one domain of application, vary greatly. In insurance applications, for instance, discovery of rules for premium setting and fraud detection require quite different KDD processes even though the data is often very similar. Such different requirements entail a variety of techniques to be employed. In current practice the diversity of databases, application domains, and requirements has resulted in an artificial barrier to the promotion of KDD technology—trial-and-error has become a dominant approach with little, but increasing, formal guidance available. A theory or methodology will guide the KDD practitioner in the effective use of the technology.

Recent efforts recognise the KDD process as consisting of a number of interacting, iterative, stages involving various data manipulation and transformation operations. Information flows from one stage onto the

*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.

next, as well as backwards to previous stages. Fayyad, Piatetsky-Shapiro and Smyth (1996), for instance, identify 9 steps in the KDD process.

Most attention within the KDD community has focused on the Data Mining 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 own experience within the ACSys Data Mining Project) suggests that the other stages account for up to 95% of the effort—it is important then to understand the *whole* KDD process. Brachman and Anand observe that "no one has begun to identify all of the building blocks in a realistic KDD process," and they begin to address the issues and to describe the complexities of the process.

The ACSys Data Mining Project supports the whole-process view of KDD, developing the ACSys Data Mining environment to support all stages of the process. The 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. Specifically, the project uses the Darwin data mining toolkit (Thinking Machines Corporation 1995) amongst others on a variety of high performance computing platforms. A number of domains have been considered, ranging from insurance to astronomy.

In this paper we 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 each stage. Section 2 identifies some of the issues that characterise KDD. We then present a view of the KDD process in Section 3. A model which captures the key elements of the KDD process is described in Section 4. Section 5 illustrates 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 Machine Learning and Statistics is often characterised in terms of the size of the dataset, the complexity of the 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 concerned with issues ranging from relating to databases and data warehouses (Inmon and Hackathorn 1994) to issues relating to the usefulness of the knowledge discovered. We explore here some of the characteristics that distinguish KDD.

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 data mining 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 this case client) 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 data mining proceeds and results are generated the directions taken in the mining may change quite dramatically. Although the nature of the data may lead us to use clustering techniques, or perhaps a classification algorithm or contingency tables, it is not always clear a priori which tools will produce the best results. Indeed, the use of a variety of tools often leads to a more interesting variety of results.

3 The KDD Process

In the context of such issues we consider the process of knowledge discovery from databases as consisting of the four stages identified in Figure 1. This encapsulates 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.

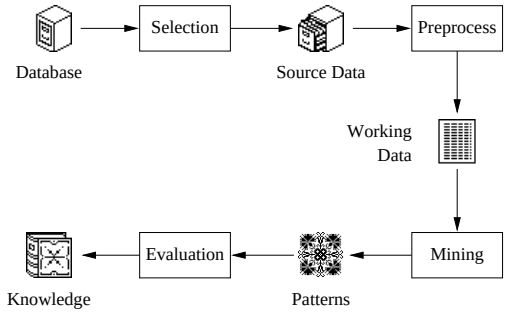

Figure 1: The four stage KDD process.

In most cases we begin with an original Database, usually developed for tasks other than KDD. After due consideration (and this can require some 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.

The Source Data will usually exhibit many of the characteristics listed in the previous section. 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 processing with the Mining tools. The ACSys Data Mining Environment uses the "standard" Machine Learning format for storing the resulting Working Dataset (a simple format with a single record per line with comma separated fields). The types of Preprocessing that the environment supports include traditional database projection and selection, and value mapping and classification functions. Data cleansing type operations are also supported, such as operations to resolve fuzzy matches between entities in the data that actually represent the same entity.

Once a suitable Working Dataset has been produced the Mining process can begin. Mining employs a variety of tools to explore different types of patterns in the Working Data. 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 Mining 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.

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 process 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.

Two further important points that should be emphasised about the KDD process are:

- both domain expertise and KDD expertise are crucial for every stage of the KDD process;

- 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 Conceptual 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 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 $\mathcal{S}$. We discuss each of these elements of the model here and illustrate the concepts with an actual KDD application in the following section.

4.1 Database $\mathcal{D}$

The Database component begins with the original data from which knowledge is to be discovered and continues through the Source Data (generally derived from the Database using traditional database technology) to the Working Data which is used directly in the Data Mining stage.

The original Database (commonly a Data Warehouse) is generally stored and maintained separately from the KDD process and exhibits the characteristics identified in Section 2. Control and access to the 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.

For KDD purposes (and particularly in the context of the Source Data) it is useful to view $\mathcal{D}$ as a universal relation (Ullman 1988) effectively mapping multiple relational tables into a single table. The universal relation then provides a unified interface for the variety of operations that are to be performed during the KDD process. A particular advantage of this view of the Source Data is that the universal relation can easily be materialised as a flat file (a common and convenient format for machine learning and statistical tools, although as our datasets grow database support will be required).

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.

4.2 Knowledge Representation Language $\mathcal{L}$

A language is required for representing the patterns discovered in the database and any background knowledge available. Knowledge representation has been well studied in AI—a number of formal and informal representation methods have been developed and applied in various systems. Production rules are a common (and simple) language often used (particularly popularised with expert systems). A typical example of a rule is:

> **If** insured is male **and** age is less than 21 **and** item insured is expensive
> **Then** an insurance claim is rarely lodged.

(Young male drivers with expensive cars tend to take care of their expensive investment.)

Within the ACSys Data Mining Environment production rules are used to store and to communicate discovered patterns. This basic unit of representation is encompassed within a more structured representation which can record meta-information about the rules themselves. Such meta-information can record details of the origin of the rules (which data mining operations were employed, what pruning might have been performed, misclassification costs associated with the rule etc.). It can also record information about the

rule derived from the Source Data directly (such as the total cost of claims associated with the rule in the context of insurance risk analysis).

One of the major advantages of using rules as the representation language is the ease with which they are interpretable, both by human readers of the rules and by traditional rule-based systems. However, the possible complexities of discoveries requires flexibility in the representation: we might discover that the probability of some event has a Poisson distribution whose mean is a function of other factors. The language used to represent discoveries must cope.

4.3 Pattern Evaluation $\mathcal{F}$

The 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 $\mathcal{L}$ (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 can often be the most effective form of evaluation of discovered patterns. Visualisation tools become a critical component then of the 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 discoveries.

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.

4.4 Set of Operations $\mathcal{S}$

A variety of operations are used throughout the whole KDD process, from the collection of raw data to the production of useful knowledge. The choice of operations depends on how the problems are formulated and how the solutions are perceived by the user. The following classes of operations are identified for the various stages of KDD. This classification is only partially systematic, and various operations are often not easily identifiable as belonging to one class over another. However, identification of the different classes can help in the specification of the functionality of a generic KDD system.

Source data collection operations: This group of operations is used to extract the source data to be mined from the Database, usually extracting it as a universal relation. The common operations here include the relational operators such as projection, selection, and join. Others that are useful include aggregation and partition operations like summation, average, and group-by. These operations are usually supported by the source database management system, in languages such as SQL. Generally such standard languages are adequate for the selection of the initial Source Data but are inadequate for the types of preprocessing operations that need to be performed on the Source Data.

Data preprocessing operations: A cleansed source dataset provides better opportunities for successful data mining. The types of operations here include detection and correction of erroneous data (often referred to as *data scrubbing*), deletion of redundant data, and compensation for missing data. Such operations are usually specific to the data being processed and general purpose data manipulation languages (like awk) are useful for implementation.

Meta-data operations: Meta-data is the data about the source data to be mined and therefore represents part of the domain knowledge. Meta-data plays an important role in the success of data mining. Operations for meta-data include determining the minimum and maximum values of attributes in ordered domains, the sets of categorical values of attributes in the categorical domains, value distributions of attributes, correlations of attributes, etc.

Data mining operations: Operations in this group represent the kernel stage in the KDD process, i.e., extracting knowledge from the Working Dataset. A large number of algorithms can be classified into this group, including clustering, decision tree induction, neural networks, genetic algorithms, and a variety of statistical algorithms. These operations might be oriented towards data description (as in clustering) or model fitting (as in classification).

Prediction operations: Classification rules extracted by the rule mining operations are often used to predict classes for new objects. Rule based systems are a common example of the type of interpreters that can be used, although more expressive languages (such as Prolog) may be more appropriate.

Visualisation operations: Operations in this group include methods and algorithms which can visualise the internal representation of the data in various human interpretable forms. These operations enable close involvement of the end users in *all* of the KDD stages. Since data mining is an exploratory and uncertain process in nature, in many circumstances human users are more sensitive to subtleties of the data than automated algorithms.

Operations such as those described in this section have been implemented in the ACSys Data Mining Environment as filters that work together to process the data. In the longer term a programming language for manipulating data and performing analyses is needed.

5 Case Study: Insurance Risk Analysis

To be useful, a model should provide a framework in which to perform KDD. We now describe the KDD process for a particular application from the insurance domain. The discussion will remain at a generic level to avoid disclosing commercially sensitive details. The aim of the KDD project was to discover knowledge that might assist our collaborators, NRMA Insurance Limited, in better determining policy premiums (cost of insurance to a customer) from an analysis of the risk associated with policies (Williams and Huang 1996).

NRMA Insurance Limited maintain large databases of millions of records. The data that was made available to us was transaction oriented, consisting of transactions performed on individual policies. A transaction might be new business, a renewal, a cancellation of a policy, a change to some details of the policy, a claim on a policy, etc. The original data was stored as a number of relational tables and various relational operations were performed by the data owners to build the Source Data.

A suite of operations were then performed on the Source Data to transform it into the Working Data. The major transformation was from a transaction oriented view of the data to a policy oriented view. This involved an elaborate analysis of the data, leading to the implementation of a collection of automated operations to perform the task. (The automation was important so that different transformations could easily be performed on the data as the data became better understood.) The actual process is summarised in Figure 2 (with indications of size for a small trial database).

The primary task is encapsulated in the Preprocessing stage. This commenced with the cleansing of the Source Data:

- records with missing (critical) values were removed;
- certain field values were transformed to forms more appropriate for analysis.

The transformations ranged from simple calculations, such as the determination of an age rather than a birth date, to mappings of large range categorical values to a smaller set of categorical values (required in the context of particular data mining tools).

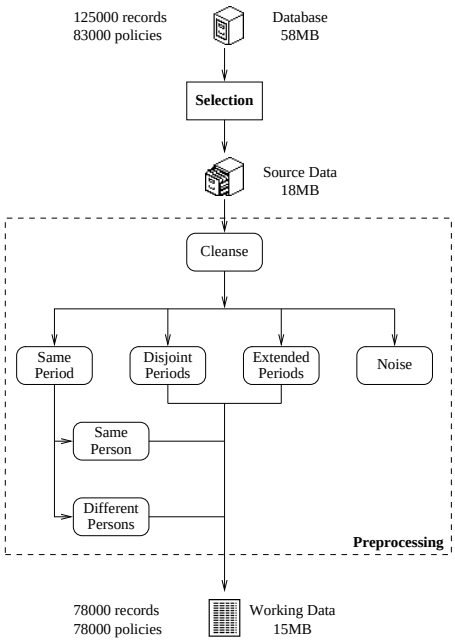

Figure 2: Preprocessing.

The major effort expended in the Preprocessing stage was the merging of multiple transactions into single policies. This task required:

- the identification of individual policies;
- the merging of multiple transaction policies into single records;
- the creation of new fields to record aggregate information.

A collection of rules was developed to automate the process of merging. This facilitated the critical process of revising earlier stages as we iterated through the KDD process. The process was implemented as a collection of filters that could be linked together manually or via a user interface.

Having produced a clean Working Dataset, a task that required considerable effort, the task of most interest to the customer—exploring the data with a variety of tools—could be addressed. This exploration required revisions to be made to earlier decisions, including the extraction of further attributes from the database and various tuning of the cleansing and merging tasks.

A variety of Data Mining explorations of the data were performed—many not proving to be particularly insightful, but some leading to interesting snippets of knowledge. StarTree from the Darwin suite of Data Mining tools (Thinking Machines Corporation 1995), for example, was used to build a decision tree to predict if a claim might be made on a policy. This analysis identified a number of hot spots in the data which, when combined with further information derived from the data (relating to periods of exposure and size of claim costs), could be used to pinpoint previously unrecognised high risk areas.

An important element of this KDD exercise was the determination of whether the discovered patterns were useful. Subjective opinion lead to the development of some objective criteria for the evaluation of the patterns discovered. For example, a discovered rule was deemed to be interesting if it was derived from multiple policies where the total claim cost was significant (above a certain threshold). In determining the worthiness of a rule, extra data not used in the actual Data Mining stage was used, again sometimes requiring modifications to be made to earlier stages of the KDD process.

6 Summary

Practical KDD applications exhibit a variety that requires considerable flexibility on the part of those performing the task and on the support environments used. It is difficult to provide a single KDD environment

suitable for all kinds of applications. An ideal KDD environment would implement a set of basic operations (possibly as a special purpose programming language), covering all stages of the KDD process as described in this paper and by others (Brachman and Anand 1996).

We have identified a simple yet sufficient breakdown of the KDD process involving four interacting and iterative stages. We have presented an associated model of KDD which identifies the key elements of the process: a database; a knowledge representation language for expressing background knowledge and discovered patterns; a means for evaluating the patterns discovered; and a collection of operations associated with the various stages of the KDD process.

An example KDD application in insurance risk analysis has been used to illustrate the process and the model. The ACSys Data Mining Environment continues to be developed to provide an environment which facilitates many of the tasks of KDD in addition to the traditional Data Mining tasks. Having a model to formulate the KDD process provides the framework in which to implement the environment. Such a model is gradually evolving within the KDD community and this paper provides further refinements of and support for such a model.

Acknowledgements

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). The KDD team also includes Peter Milne of DIT and Murray Cameron, Glenn Stone, Petra Kuhnert, and David Chan of DMS. Industry collaborators from the NRMA include Keith Forster, Philip Woods, and Hong Ooi. Thanks to all for their comments on this paper.

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.

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.

Inmon, W. H. and Hackathorn, R. D.: 1994, *Using the data warehouse*, John Wiley, New York.

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.

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

Ullman, J. D.: 1988, *Principles of Database and Knowledge-Base Systems*, Vol. I, Computer Science Press, Rockville, Maryland.

Williams, G. J. and Huang, Z.: 1996, KDD for insurance risk analysis: A case study, *Technical Report TR-DM-96-06b*, CSIRO Division of Information Technology, available from Graham.Williams@cbr.dit.csiro.au.