DATABASE ISSUES IN KNOWLEDGE DISCOVERY AND DATA MINING

Chris P. Rainsford
Defence Science and Technology Organisation
Information Technology Division
DSTO C3 Research Centre
Fernhill Park, Canberra 2600
chris.rainsford@dsto.defence.gov.au
John F. Roddick
School of Computer and Information Science
University of South Australia
Mawson Lakes, Adelaide.
South Australia 5095
Roddick@cis.unisa.edu.au

ABSTRACT

In recent years both the number and the size of organisational databases have increased rapidly. However, although available processing power has also grown, the increase in stored data has not necessarily led to a corresponding increase in useful information and knowledge. This has led to a growing interest in the development of tools capable of harnessing the increased processing power available to better utilise the potential of stored data. The terms "Knowledge Discovery in Databases" and "Data Mining" have been adopted for a field of research dealing with the automatic discovery of knowledge implicit within databases. Data mining is useful in situations where the volume of data is either too large or too complicated for manual processing or, to a lesser extent, where human experts are unavailable to provide knowledge. The success already attained by a wide range of data mining applications has continued to prompt further investigation into alternative data mining techniques and the extension of data mining to new domains. This paper surveys, from the standpoint of the database systems community, current issues in data mining research by examining the architectural and process models adopted by knowledge discovery systems, the different types of discovered knowledge, the way knowledge discovery systems operate on different data types, various techniques for knowledge discovery and the ways in which discovered knowledge is used.

INTRODUCTION

In recent years the continued growth in the size of databases has led to an increased interest in the automatic extraction of knowledge from data. It is therefore not surprising that many leading database researchers have identified this as an area worthy of significant investigation (Silberschatz, Stonebraker and Ullman 1996; Stonebraker, *et al.* 1993). The term Data Mining, or Knowledge Discovery in Databases (KDD), has been adopted for a field of research dealing with the discovery of information or knowledge from data held in more or less structured databases (Fayyad, *et al.* 1996; Piatetsky-Shapiro and Frawley 1991). Although these two terms have been used interchangeably in the past, leading researchers in the field have only recently distinguished between them (Fayyad, Piatetsky-Shapiro and Smyth 1996). Following their distinction, knowledge discovery in databases can be seen as the overall process of extracting useful and interesting information from databases. This process includes the selection and preparation of data and the manipulation and analysis of results. By comparison data mining can be viewed as the application of knowledge discovery algorithms without the other stages of the knowledge discovery process, and is therefore a subset of KDD. KDD is typically most useful in situations where the volume of data is either very large or too complicated for traditional methods, or where human experts are unavailable to extract knowledge. As would have been expected, KDD has borrowed heavily from traditional machine learning and database theory.

Learning can be defined as *knowledge acquisition in the absence of explicit programming* (Valiant 1984). Machine learning aims to automate the learning process, so that knowledge can be acquired with minimal dependency upon human input (Michalski, Carbonell and Mitchell 1984). Machine learning has traditionally focussed on learning from sets of specifically and, in many cases, artificially generated data. Data mining aims to adapt these machine learning paradigms to learn from databases containing real world data. Learning from within databases has some advantages (Roddick and Rice 1998):

- The data is stored in a more or less structured manner. For example, in a relational database data is typically normalised into relations that eliminate redundancy and can be joined in various ways to retrieve required data sets from the database. In other database paradigms, either more or less structure is available. Nevertheless any apriori known structure can be utilised.
- Some domain knowledge is already encoded implicitly within the database. For example the existence of a participation constraint may be flagged by a not null constraint. Similarly the cardinality of relationships is also often explicit within database structure and constraints.

- High performance query, data manipulation and transaction tools are already available. This would include the database management system, associated query language, specialised hardware and other database tools. It therefore makes some sense to use these tools to interrogate the database where appropriate.
- The number of databases with data applicable to mining techniques is large and growing. The effort expended in developing such tools is thus economically viable.

Similarly, the use of data from databases imposes a number of characteristics and constraints:

- The volume of data is typically very large. For example, the SKICAT system has been developed to process three terabytes of graphic images resulting from a sky survey (Fayyad, Weir and Djorgovski 1993). Therefore, any data mining tool must perform satisfactorily on large volumes of data.
- The data may contain noise. Data mining tools must provide adequate mechanisms for finding sufficiently accurate results from noisy data.
- The database may contain incomplete information. Not all information useful for the discovery of knowledge may actually be stored within the database. Likewise much redundant and useless data may also be present. Therefore data mining tools must facilitate both the selection of relevant data, and learning with incomplete knowledge.
- The data has not generally been collected for the purpose of knowledge discovery. As well as leading to some of the above problems this means that data may be formatted inappropriately. Knowledge discovery tools must therefore be able to access data stored in various formats.

One overall goal of knowledge discovery and data mining research is to utilise the advantages of learning from databases, while accommodating the constraints imposed.

This paper provides a survey of current areas of research within data mining. The next section describes a process model of a data mining system. The major types of discovered knowledge are then discussed and following that, the data types that are targeted for knowledge discovery. Some of the major applications of discovered knowledge are then described and areas for future research are identified and discussed in the final section.

A DATA MINING MODEL

Figure 1 represents one possible model of the data mining process (adapted from (Rainsford and Roddick 1996)). For any database, the number of possible rules that can be extracted is far greater than the number of tuples in the database. Knowledge discovery can thus be viewed as the multi-stage process of selecting interesting rules from the total rule-space that exists within a database. There is therefore a process of progressively reducing the initial infinite rule space down to a small subset of useful rules.

The model outlined is based on the nature of the refinement as illustrated in many current research tools – that of a reduction process performed using a selection of filters that reduce the target rule space on the basis of source data, rule pattern, statistics and semantics. There are many special cases where a phase of the filtering process does not exist or is not used within a given research tool, and in these cases the filter effectively allows the rule space to pass unreduced. For example, semantic filtering may not be used if rules are to be used for query optimisation.

Each of the filtering stages may consist of zero or more filters specified by the user or discovery system. The target rule set may be passed back and forth between the filters for reprocessing. A central controller coordinates the operation of the filters. As noted on the diagram the final rule set can be integrated into the existing knowledge base. As indicated, both the knowledge base and the user may interact with each step of the rule-space reduction process. Note that the filtering processes may have a profound effect on the outcome of the data mining process. Thus, the outcome of the data mining process is quasi-nondeterministic.

Data Filtering

The initial stage involves the selection of the data of interest for knowledge discovery. The user may direct the KDD system to areas of interest using templates, visualisation tools and by specifying sampling strategies. The result of this phase is a reduced set of test data and a correspondingly reduced rule-space.

Pattern Filtering

The second phase is the pattern filter, where a particular rule type to be searched for is specified. This may be fixed by the limitations of the system or specified to a greater or lesser extent via the use of templates or rule type selection. The type of pattern discovered is typically restricted by the KDD system itself because most systems can only discover a limited number of rule types. The pattern can be further restricted by the user requiring the presence of particular attributes on one or other side of a discovered rule, or by restricting the number of possible conjunctions it may contain. The effect of the pattern filter is therefore to reduce the relevant rule space to only rules of a particular type, eg. association rules with a given term in the consequent. The specification of a pattern to be searched for can be made intuitively in the form of a template-based request such as:

> Find all *Association* rules with *Butter* in the *Consequent.*

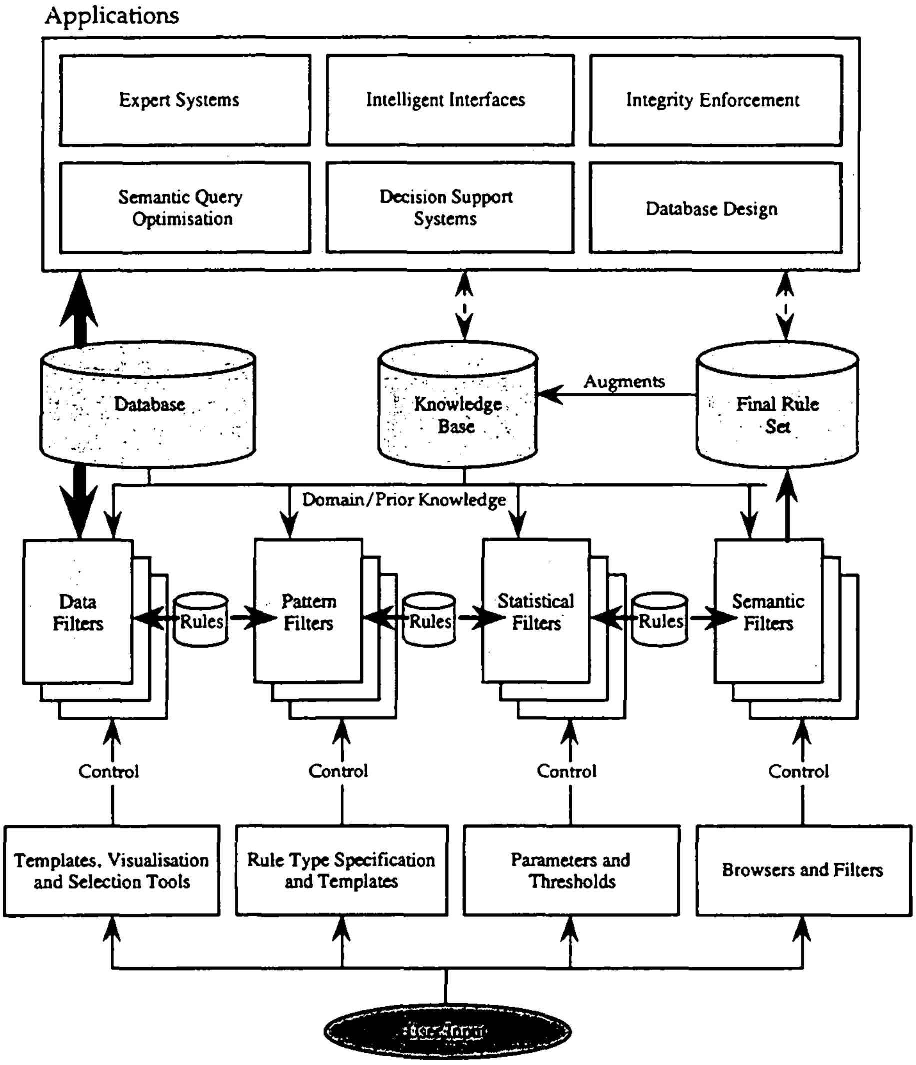

Figure 1. A Model of the Data Mining Process.

In this simplistic example of a template, the key words in italics may be replaced to form different requests. This can be implemented in an interface using tools such as pick lists, which simplify the users interaction with the system. Most KDD systems implement some form of template for pattern specification. One Example is the *Nielsen Opportunity Explorer*™, a knowledge discovery tool utilising discovery templates as described by Anand and Kahn (1993). Within this system, knowledge discovery templates allow the user to also specify both the analytic techniques to be applied (statistical filtering), and the data that they are to be applied to (data filtering). Likewise the *Explora* system utilises statement types that act as templates, corresponding to specific types of rules to be discovered (Hoschka and Klösgen 1991).

Statistical Filtering

The rule space is further refined through a third phase of statistical filtering. At this stage the space of rules that satisfy the pattern is further reduced by the application of statistical tests. This process eliminates rules that fit the specified pattern, but are deemed statistically unsatisfactory or insignificant. The user may interact with this phase by setting statistical parameters or turning statistical tests on or off. As an example consider the following simplistic template:

> Find all Association rules with Butter in the Consequent having a minimum support of 0.002 and a minimum confidence of 0.85

In this case the template specifies two statistical measures support and confidence, with a required minimum value. Although the statistical values a tool associates with a rule vary, they typically describe attributes such as the confidence of the rule or the amount of supporting evidence for the rule in the database. Whilst traditional statistical techniques form a foundation for statistical evaluation, in many cases customised measures have been developed to meet the specific requirements of KDD applications.

As an example of a statistical measure developed for KDD, Agrawal *et al.*(1993) describe two statistical functions that can be used to describe an association rule, *support* and *confidence*. *Support* is a measure of the probability that a transaction in the database will contain the given pattern. Given the presence of the antecedent in a tuple, *confidence* is the probability that the consequent will also be present.

Another approach is adopted by *Explora* Version 1.1 as described by Klösgen (1995b), which uses measures of *evidence* and *affinity* to help eliminate redundant findings. *Evidence* is a statistical measure of the quality of an individual rule, which may be calculated in several ways. *Affinity* is an asymmetric measure of the similarity between two findings. These two measures can be combined to filter out a finding that is similar, but weaker than another finding and hence reduce redundancy.

Anand *et al.* (1995b) advocate evidential theory as a more general-purpose statistical technique for use within knowledge discovery (Guan and Bell 1991, 1992). Evidential theory, which employs a generalisation of the Bayesian Model for Uncertainty, has two major advantages of over the widely used Bayesian model. Firstly, it allows a belief value to be associated with ignorance that can be used to handle missing database values. Secondly, it allows evidence at various levels of coarseness to be combined. This system is inherently parallel and therefore leads to parallel systems for knowledge discovery. The application of evidential theory to parallel knowledge discovery is described by Anand *et al.* (1995a); likewise the application of evidential theory to knowledge discovery from images in a spatial context is described by Bell *et al.* (1994).

Semantic Filtering

The final, and arguably the most difficult, phase of rule space reduction is semantic filtering. At this stage the interestingness and utility of discovered rules is assessed. Some rules that satisfy all other requirements may simply be uninteresting or redundant, and hence are removed from the rule space. This phase typically involves heavy interaction with the user via browsers and rule visualisation tools. In addition the knowledge base may be consulted to place the discovered rule set in the context of what is already known. The output of this process is the final set of discovered rules that satisfy the user requirements. For the purposes of semantic evaluation it would be desirable to allow users to browse a set of discovered rules. Interesting rules could then be identified for further investigation or presentation.

A good example of support for interactive semantic selection can be seen in the *Explora* system. The *Explora* system which finds rules and examples corresponding to user selected statement types facilitates interactive browsing of its discovered results (Hoschka and Klösgen 1991). The results of analysis are presented as messages linked by appropriate relationships and the user may browse the resulting message space. A report of the analysis can be interactively composed with the assistance of three available activity spaces. The analysis space contains a window for each statement type underlying the search and related findings such as supporting and contradictory examples can be browsed. An outlining space facilitates the creation of the final report from selected analysis results. An argumentation space can then be used to organise the selected findings into argumentative structures. This set of browsing tools thereby provides a user interactive process to create valuable reports from raw analytical results.

The utility of KDD tools can be improved if the semantic evaluation of rules is automated, removing the requirement for human interaction at this stage. However the definition of what makes a rule *interesting* remains subjective and composed of too many aspects to be easily definable and the problem has become widely acknowledged in the KDD field *qv.* Piatetsky-Shapiro (1994). Klösgen however identifies several aspects that capture the nature of interestingness in discovered knowledge (Klösgen 1995b). The properties suggested by Klösgen are Evidence, Redundancy, Usefulness, Novelty, Simplicity and Generality. A similar set of adjectives describing interestingness is provided by Asa and Mangano: Performance, Simplicity, Novelty and Significance (Asa and Mangano 1995).

TYPES OF DISCOVERED KNOWLEDGE

The type of knowledge that is discovered from databases and its corresponding representational form varies widely depending on both the application area and database type. The specification of the type of knowledge to be discovered directs the pattern filtering process. Knowledge learned from large sets of data can take many forms including classification knowledge, characteristic rules, association rules, functional relationships, functional dependencies and causal rules. This section will describe each of these categories of knowledge and discuss example systems that learn each type.

In Table 1 the types of knowledge which are explicitly supported by a selection of current data mining tools are indicated. Many of these tools are subject to ongoing development and therefore this represents a summary at the present time. Moreover, the purpose of this survey is to demonstrate the broad diversity of a cross section of data mining tools and not to form the basis of any tool comparison or evaluation.

Classification Knowledge

Classification knowledge can be used to categorise new examples into classes on the basis of known properties. Such information can, for example, be used by lending institutions to classify the credit risk of prospective borrowers and could be constructed from records of past loans. Following the formalism of Agrawal *et al.* (1992) inferring classification functions from examples can be described as follows: Let G be a set of *m* group labels $\{G_1, G_2,..., G_m\}$. Let A be a set of *n* attributes (features) $\{A_1, A_2,..., A_n\}$. Let dom(A_i) refer to the set of possible values for attribute A_i. We are given a large database of objects D in which each object is an *n*-tuple of the form $< v_1, v_2,..., v_n >$ where $v_i \in$ dom(A_i) and G is not one of A_i. In other words, the group labels of objects in D are not known. We are also given a set of example objects E in which each object is a (n+1)-tuple of the form $< v_1, v_2,..., v_n, g_k>$ where $v_i \in$ dom(A_i) and $g_k \in G$. In other words, the objects in E have the same attributes as the objects in D, and additionally have group labels associated with them. The problem is to obtain m classification functions, one for each group G_i, using the information in E, with the classification function f_j for group G_j being f_j: $A_1 \times A_2 \times ... A_n - G_j$ for $j = 1,..., m$. We also refer to the examples set E as the training set and the database D as the test data set.

System / System Extension	Classification	Characteristic	Association	Functional Relationships	Functional Dependencies	Causal	Temporal	Clustering
Clementine	✓	✓					✓	✓
DBMiner (Han, *et al.* 1996)	✓	✓	✓				✓	✓
Emerald	✓	✓		✓			✓	✓
Explora (Klösgen 1993; Klösgen 1995b)	✓	✓					✓	
Mine Rule (Meo, Psaila and Ceri 1996)			✓					✓
MineSet 1.1	✓	✓	✓				✓	✓
Posch (Long, Irani and Slagle 1991)						✓		
Quest (Agrawal, *et al.* 1996)	✓		✓				✓	✓
RX Project (Blum 1982)						✓		
Savnik & Flach (Savnik and Flach 1993)					✓			

Table 1 – Knowledge types supported by selected research tools

As noted by Rubinstein and Hastie (1997) classification can be approached in two different ways. The first approach is informative learning where the focus is upon finding a good description for each class. Classification then proceeds by comparing the instance to be classified with each class to find the best match. This approach is also useful when the emphasis is on determining the characteristics that are associated with a particular class of instances. The second approach is discriminative learning where the emphasis is on defining the boundaries between classes. A model for categorising examples into each class is developed rather than a model describing the characteristics of each class. This approach is applicable to problems where the emphasis is on classifying instances of unknown class.

One of the most popular models for representing discriminative classification knowledge is the decision tree. The nodes of the tree represent attributes to be tested, and the branches correspond to the possible values of those attributes. The leaves of the tree represent classes into which examples are classified. Therefore, starting at the root node on the tree, an example can be tested at each node it encounters and follow the resulting branches to be classified in its appropriate leaf.

The induction of decision trees has been widely investigated and an important algorithm for the induction of decision trees, *ID3,* is described by Quinlan (1986). This algorithm has been built on extensively and proposals to accommodate extensions, such as the accommodation of approximate data, have been numerous. Decision trees can be equally represented as rules. One rule can be constructed for each leaf node, where the leafs classification class is the consequent of the rule. The antecedent is a conjunction of the attribute value pairs encountered at each branch of the tree. Agrawal *et al.* (1992) describe an interval classifier *IC* that produces *k*-ary classification trees. Importantly, non-categorical attributes are nót divided into binary sub-trees as in ID3, but are instead split into *k*-ary sub-trees. In comparison with ID3 the resulting trees can be created with greater efficiency and display favourable classification accuracy. The improved speed of classification represents a significant advantage within knowledge discovery systems where the data is both dynamic and of large volume. In addition, speed of classification is important for the support of ad-hoc queries.

One informative approach to classification is to identify target classes and then find a description that uniquely identifies members of each class. Following this approach Cai *et al.* describe a classification rule learning algorithm *LCLR* (Cai, Cercone and Han 1990). Their approach utilises conceptual hierarchies as a tool for induction. Classes to be classified are firstly specified. All tuples in the learning data set that describe a specific class are then generalised stepwise until a description consisting of a specified number of generalised tuples is reached. Generalisation takes place via concept ascension of attribute level conceptual hierarchies. As descriptions describing more than one class are eliminated, areas of overlapping classes in the decision space remain undescribed by the resulting clauses. This approach produces a classification rule in the form of a generalised relation.

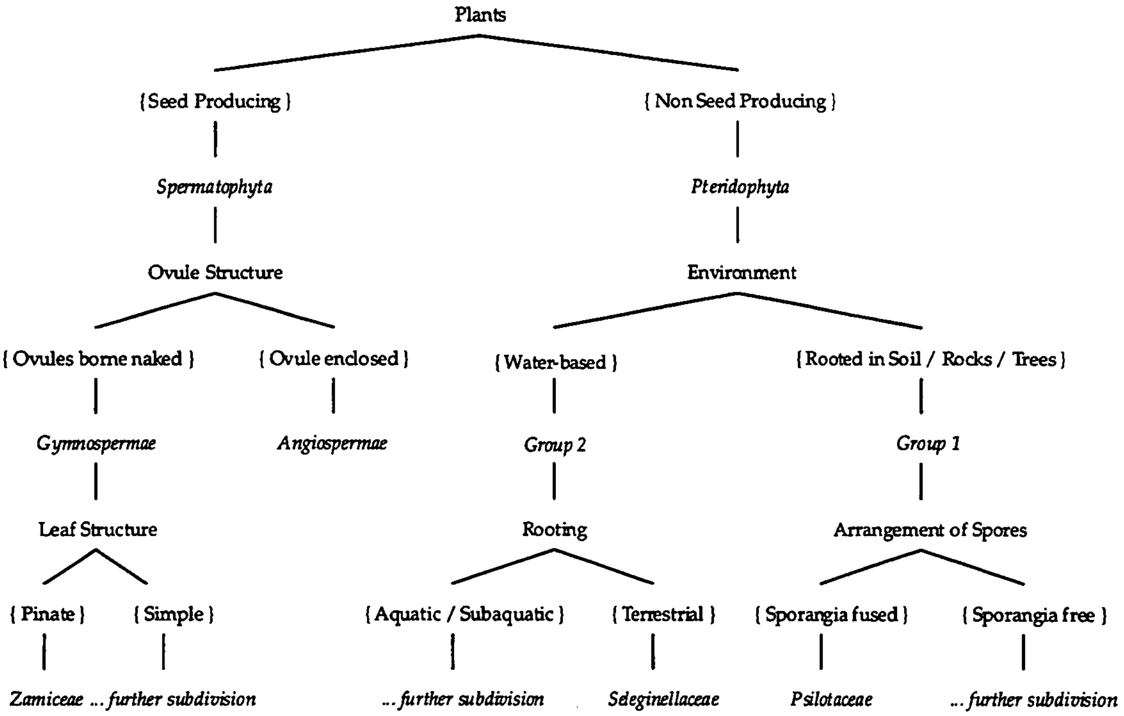

Figure 2. An example decision tree, plant classification (after (Beadle, Evans and Carolin 1986))

Another informative approach to classification is to classify attribute values individually and assign appropriate classification(s) that can then be combined to classify an object displaying these values. This approach is adopted by Chan and Wong (1991) who describe a statistically based technique for the classification of objects from existing databases. Their approach assigns a weight of evidence to attribute values indicating how the value of an attribute effects an object's potential membership of a given class. The total evidence for and against membership of a particular class can then be determined for each object to be classified, based upon its attribute values, and the best classification determined. This approach can be applied to noisy data with missing values, which may be encountered in real world databases. In addition, objects to be classified are tested for possible classification against all classes. Because weightings are used, it can be determined how good an example of a class a particular object is. Membership of more than one class is also possible, where the membership weighting is within acceptable limits. This feature is particularly useful in domains where membership of more than one class are valid.

Characteristic Rules

A characteristic rule can be defined as *an assertion that characterises the concept satisfied by all of the relevant data in the database* (Han, Cai and Cercone 1993). Characteristic rules are essentially rules that describe the characteristics of a concept, via an abstraction based upon the data in the database. The need to provide generalisations to describe concepts means that a path of generalisation, such as a conceptual hierarchy, is typically required as domain knowledge. Characteristic rules make no explicit reference to relationships between entities, or means of classifying them. However, characteristic knowledge is useful for providing a summary or abstraction of the data it describes, which may be used for applications such as query optimisation, integrity enforcement or the automatic discovery of dependencies.

Han *et al.* (1993) describe an approach to learning characterisation rules that is based on a process of concept ascension. Within this model generalisation occurs at the attribute level, following a generalisation hierarchy provided as domain knowledge. This approach assumes that such a hierarchy is available. Both a qualitative and quantitative algorithm has been developed. The quantitative algorithm facilitates statistical methods, whereas the qualitative method can be used for purposes such as complete induction of the entire data set. Lu *et al*, describe similar techniques to induce characteristic rules from spatial data (Lu, Han and Ooi 1993).

Association Rules

The discovery of association rules in large databases was first described by Agrawal *et al.* (1993). The initial motivation for association rules was to aid in the analysis of large transaction databases, such as those collected by supermarkets. The discovery of associations between the purchase of various line items can potentially aid decision making within retail organisations. Transaction databases are therefore the primary domain targeted for association rule discovery.

Using the formalism provided by Agrawal *et al*, association rules can be defined as follows. Let $I = I_1, I_2, \dots, I_m$ be a set of binary attributes or items and T be a database of tuples. Association rules were first proposed for use within transaction databases, where each transaction t is recorded with a corresponding tuple. Hence attributes represented items and were limited to a binary domain where $t(k) = 1$ indicated that the item I_k had been purchased as part of the transaction, and $t(k) = 0$ indicated that it had not. Association rules in this form can however be generalised to any attribute domain in which one set of domain values corresponds to 1 and another set to 0. For example within a university database domain values corresponding to science subjects {Chemistry, Maths, Physics} may be assigned the value 1, and humanities subjects {Literature, Modern Art, Environmental Design} assigned the value 0. Therefore t may be any tuple with binary domain attributes, which need not represent a transaction. Let X be a set of some attributes in I. We say that a transaction t satisfies X if for all attributes I_k in X, $t(k) = 1$. By an association rule, we mean an implication of the form $X \Rightarrow Y_,$ where X, the antecedent, is a set of attributes in I and Y, the consequent, is a set of attributes in I that is not present in X. The rule $X \Rightarrow Y$ is satisfied in the set of transactions T with the confidence factor $0 \leq c \leq 1$ iff at least c% of transactions in T that satisfy X also satisfy Y. We will use the notation $X \Rightarrow Y | c$ to specify that the rule $X \Rightarrow Y$ has a confidence factor of c.

Association rules are particularly applicable to commercial data mining applications. For instance, in a database containing details of product sales they allow the user to request knowledge such as:

> find any rules that have *Diet Lemonade* as the consequent

Such a request enables market analysts to find the factors affecting the sale of diet lemonade. Conversely learning requests regarding the antecedent can be asked, such as:

> find any rules that have *Taco Shells* in the antecedent

Requests such as this that involve finding rules with an item in the antecedent can help analysts in determining how the particular item may be associated with the purchase of other items or more generally, indicate to researchers possible correlations that may be worth further investigation. As a simplistic example, it may be induced that the decision to stop selling *Taco Shells* may lead to a significant fall in the sale of related products such as *Taco Sauce*. The strength of any such associations could be tested using an association rule such as the one above. This approach may help identify previously unknown or overlooked associations, and test the strength of associations believed to exist. Meo *et al.* describe an SQL-like operator for mining association rules - *Mine Rule* (Meo, Psaila and Ceri 1996). This operator may be utilised by systems users and analysts to direct searches such as those described above.

An itemset can be defined as a set of line items purchased as part of a single transaction. The initial algorithm, *AIS*, proposed by Agrawal *et al*, operates by looking for itemsets with the required support within the database, called large itemsets. Trying to find all of the large itemsets in a single pass over the data is inefficient because most of the itemsets that would be measured are not large. The *AIS* technique reduces this inefficiency by making several passes over the data and using the results of the current pass to produce candidate large itemsets to evaluate in the next pass. This process of discovering large itemsets continues until no more large itemsets are found. This approach however has proven to produce too many small itemsets as candidates for large itemsets. This is because it produces candidate itemsets by extending the existing large itemsets using itemsets that are not large. Mannila *et al.* (1993, 1994) describe an algorithm that is faster than the Agrawal *et al.* algorithm, and has demonstrated a performance improvement of a factor of five in testing examples. The efficiency gain in their algorithm is based upon the observation that subsets of large sets must also be large. The number of candidate sets can therefore be dramatically reduced by using only large itemsets in the construction of other candidate large itemsets. They also point out the viability of sampling as an efficient method for finding large itemsets. However, as expected, sampling techniques result in a trade-off in accuracy.

As a result of the same observation as Mannila *et al*, Agrawal and Srikant propose two alternative algorithms, *Apriori* and *AprioriTid*, both of which are more efficient than *AIS* (Agrawal and Srikant 1994). The efficiency gain is obtained by only considering previously discovered large itemsets in the construction of candidate large itemsets. The underlying principle is the same as that observed by Mannila *et al*; namely that subsets of large itemsets must also be large. The improvement in performance increases with the size of the database being input, ranging from a threefold improvement of performance to more than an order of magnitude improvement on larger sets of data. The *AprioriTid* algorithm differs from the *Apriori* algorithm, in that it does not need to test the candidate itemsets for support against the main database, but can check against the existing large itemsets themselves. This approach can offer significant performance improvement particularly when a large number of attributes are present in the candidate itemset. However this performance improvement is contingent upon *AprioriTid* being able to store the large itemsets in main memory and where this is not the case, *Apriori* starts outperforming *AprioriTid*. With this trade-off in mind, Agrawal and Srikant describe *AprioriHybrid*, a hybrid of *Apriori* and *AprioriTid*. *AprioriHybrid* begins like the *Apriori* algorithm scanning the database to generate each generation of candidate large itemsets, until it estimates the large itemsets will fit into main memory. At this point, the algorithm then generates candidates from existing large itemsets like *AprioriTid*.

A further improvement in performance is achieved by Park *et al.* (1995) who describe *DHP*, a hash-based algorithm for the discovery of association rules. Their approach is based upon the observation that the most costly process in the discovery of association rules is the determination of large itemsets in the early iterations. Importantly, large performance improvements can be gained by reducing the cost of finding large itemsets with two items. This reduction is achieved by using a hash table constructed in initial passes over the data to select fewer redundant candidate large itemsets. Because fewer candidate large itemsets are generated, the processing costs are significantly reduced. Subsequent performance improvements can also be achieved by reducing the size of the transaction database by removing redundant items and itemsets. Testing has shown that after the initial overhead required to produce the hash table, the performance of this algorithm is significantly better than algorithms such as *Apriori*.

Another approach to reducing the cost of learning association rules is described by Savasere *et al.* (1995) who describe an algorithm called *Partition* for the efficient discovery of association rules. This technique reduces disk I/O by limiting the number of passes over the database to two. This is achieved by firstly finding all candidate large itemsets in the first pass. The actual support for these itemsets can then be determined in the second pass. In the initial pass the database is divided into multiple partitions small enough to be placed in main memory and all large itemsets within each partition are found. These large itemsets are then combined to produce a set of all itemsets that are candidates to be large within the entire database. This set contains all large itemsets because any itemset that is large within the database must be large in at least one partition.

Han and Fu, describe a set of algorithms for learning association rules with the use of conceptual hierarchies (Han and Fu 1995). This approach is powerful because it widens the scope of rules that can be learnt. In addition, if the hierarchies represent real-world relationships, then it allows users to exploit an intuitive and powerful notation for selecting sets. Itemsets of different size and attributes at different levels of generalisation are both considered in the search for multiple level association rules. The order in which the search considers different sized itemsets and different levels of generalisation effects the efficiency of the algorithm. The different algorithms proposed by Han and Fu vary in their approach to searching efficiently, which largely results from reducing the search space as quickly as possible. However the utility of individual algorithms is dependent upon the data being used. A method for selecting an appropriate method based upon the nature of the data would therefore be useful. Srikant and Agrawal also address the issue of mining association rules at different levels of generalisation (Srikant and Agrawal 1995). Similar to Han and Fu, they describe multiple algorithms for mining association rules. Their optimisation techniques include a sampling technique using a subset of the database to help determine candidate large itemsets. In addition, they describe a measure of interestingness specific to the discovery of multiple level association rules. This measure is based upon the assertion that if the support for an association is consistent with the support for its more generalised form, then this more specific association is of little interest and therefore is a candidate for pruning.

The potential exists to extend association rules to encompass both different types of associations, and different types of data. Whilst the initial focus has been upon transaction databases, association rules can also be extracted from other organisational databases. Relational databases within many organisations store quantitative and categorical attributes. Quantitative association rules learnt from quantitative and categorical data are described by Srikant and Agrawal (1996). Likewise, temporal or spatial relationships can, for example, form the basis of an association. Spatial association rules are discussed by Koperski and Han (1995), and an algorithm for their derivation is provided.

Functional Relationships

Functional relationships within data describe the value of one or more attributes as a function of other attributes. For example, an attribute y may be described as a function of two other attributes x, and z as $y = \sqrt{2x * 7z}$. Such relationships are important within the scientific domain where the functional relationship between two attributes within the data may reflect relationships in the underlying domain. However whilst it may be expected that such a relationship exists within a data set, the exact relationship may be unknown. Because the data sets in question may be very large and complex, manual extraction of relationships from the data may be impractical. Therefore the automatic discovery of functional relationships, using AI techniques is a useful application domain for data mining.

A major limitation amongst function finding tools is highlighted by Schaffer (1991) and, in earlier work, referred to in the survey by Angluin and Smith (1983); for any given data of example values, an infinite number of describing functions can be derived. The selection of the most *appropriate* function is therefore integral to the operation of function finding tools. The characteristics of the most appropriate function depend upon both domain knowledge, and the proposed use of the discovered function. As these parameters will vary between applications, no single universal solution can be provided. Because of this, in many cases the utility of function finding tools will be determined in part by the ability of domain experts to specify the characteristics of desirable functions. Typically, a set of heuristics must be used to moderate the search including the expected accuracy and simplicity of the candidate functions. For example, if a quadratic and a cubic equation give similar accuracy on the data, then it is likely that the quadratic would be preferred for simplicity.

Functional Dependencies

Adopting the notation of Elmasri and Navathe (1989), a functional dependency can be described as follows. Given two sets of attributes X and Y existing in the database, a functional dependency (denoted as $X \rightarrow Y$) states that for any two given tuples t_1 and t_2, if $t_1(X) = t_2(X)$, then $t_1(Y) = t_2(Y)$ must also be true. An example of a functional dependency may be:

Course_code → Course_name

That is, if we know a subject's course code, we can determine its name. The implication of this within a relational database is that we need only store each Course_name once with its corresponding Course_code. Because of this property, functional dependencies are used to design the structure of a relational database, helping to eliminate redundant data storage, via normalisation. If previously undetected induced dependencies are found to exist within a database, restructuring of the database may take place, via a process of schema evolution, see (Roddick 1995) for a survey of publications on schema evolution.

Given the importance of functional relationships within the relational model, it is not surprising that the induction of functional dependencies within databases has been widely investigated. Roddick *et al*, loosely defines induced functional dependencies as follows: ***functional dependencies that are consistent with the data currently held in the database but which have not been defined within the database schema***, (Roddick, Craske and Richards 1996). Induced dependencies only hold true for the current data in the database, ie. there is no guarantee that valid data, which contradicts them, may later be entered into the database. Restructuring the database on the basis of induced dependencies may therefore be unwise, however they may be used for purposes such as semantic query optimisation, or integrity enforcement and error detection.

Top-down approaches to the discovery of induced dependencies begin by suggesting general induced dependencies, and then refining these to more specific dependencies as they are made invalid by contradicting examples within the data. However such an approach is impractical, as it is slow within the realm of very large databases. This is a major limitation, given that the discovery of induced dependencies is most promising within larger databases. Savnik and Flach propose an improvement by initially adopting a bottom-up inductive approach (Savnik and Flach 1993). Their technique begins by defining the cover for invalid dependencies, via bottom-up induction. This is stored in a tree-like data structure, and can be quickly accessed to assess the validity of any proposed functional dependencies. Therefore the efficiency for the top-down assessment of proposed dependencies can be dramatically improved. A totally bottom-up approach to the induction of functional dependencies may also be adopted, for example using the algorithms of Han *et al.* (1993).

The utility of functional dependencies is not restricted to traditional relational databases. Where other data models are used, functional dependencies, which incorporate the associated semantics, can potentially be found. Within temporal databases, some functional dependencies may be dependent upon temporal relationships, for example, the time and value of a change in one set of attributes may be inferable from a change in another set. In addition, functional dependencies can be constrained to be valid only at specific times. Roddick *et al*, (1996) address this issue, extending functional dependencies to the temporal database domain by defining temporal induced dependencies. Within spatial databases, spatial semantics may likewise be incorporated into functional dependencies.

Causal Rules

Causal rules describe relationships where changes in one part of the modelled reality cause subsequent changes in other parts of the domain. Blum provides the following operational definition of causality: *A is said to cause B if over repeated observations (1) A generally precedes B, (2) the intensity of A is correlated with the intensity of B, and (3) there is no known third variable C, responsible for the correlation,* (Blum 1982). The discovery of causal relationships is important within many areas of scientific investigation and especially medicine. The search for causal rules within databases also offers potential for uncovering knowledge useful in the understanding of organisational operation. The work of Roddick *et al*, mentioned in the previous section also has relevance to causal relationships, as temporal dependencies can be the result of underlying causal relationships. However causal relationships are not implied by temporal dependencies and their existence is not investigated.

Causal relationships are common targets of scientific investigation within the medical domain, where the search for factors that may cause particular medical conditions is a fundamental objective. Therefore, it is not surprising that much of the investigation into causal rule discovery to date has been within the medical domain. Most notably, the *RX* project (Blum 1982) is well known as a tool for the discovery of causal relationships within patient databases. Another example of a tool for the discovery of causal relationships within medical data is the Program on the Surgical Control of the Hyperlipidemias (*POSCH*) AI project described by Long *et al.* (1991). Unlike the *RX* project, which operates upon data not collected for the purposes of knowledge discovery, the *POSCH* AI project operates upon controlled data derived from a clinical test. Therefore whilst the *RX* project is capable of operating upon existing data, it must accommodate greater amounts of noise and errors than the *POSCH* AI project. As a result, less significance can be assigned to the findings of the *RX* Project because the objectivity and accuracy of the data is unknown.

Causal relationships typically require a significant statistical proof, and therefore, once detected via knowledge discovery, may require additional investigation. Despite this, KDD tools are useful for uncovering potential causal relationships in the first instance. Expert guidance may also be utilised at a high level in suggesting possible relationships to be investigated, and scrutinising the results. Once detected, detailed experiments can be set up to undertake a more thorough investigation of suspected causal relationships.

Temporal Knowledge

A key characteristic of KDD and data mining is the presence of a dynamic domain where data is typically updated on a regular basis. Therefore it is often useful to examine the way that data and the knowledge derived from it are changing over time. Trends, cycles and patterns may occur and their detection can be useful in analysing historic data and predicting future behaviour. Importantly, these patterns can exist in both discovered knowledge and the underlying data. While temporal knowledge can describe a wide range of different types of rules derived from different types of data, the common component is the consideration of the temporal dimension and its influence on the behaviour of entities within the modelled domain.

A common form of temporal knowledge is the existence of changes in derived rule sets over time. The detection of patterns within time series data has received significant attention. The types of data considered are typically numeric, continuous and use complex algorithms to detect patterns within a time series. As noted by Keogh and Smyth most approaches to solving this kind of problem require three fundamental components: (1) a technique to represent abstract shapes, (2) a distance measure for comparing two sequences, and (3) a mechanism for finding matching sequences within large time series databases (Keogh and Smyth 1997). A similar problem is the discovery of patterns of sequences in categorical data. Because the data is discrete and with a typically limited number of values this problem can be somewhat less computationally expensive to solve. Shapes can be represented as a sequence of domain values, the distance between sequences can be determined by comparing categorical values and the quality of the match between two sequences is determined accordingly. This is particularly interesting given that knowledge discovery has extended the application of machine learning techniques to everyday organisational databases in which categorical data is often found. Padmanabhan and Tuzhilin describe the use of temporal logic to discover pattern occurring in categorical data (Padmanabhan and Tuzhilin 1996). Agrawal *et al.* propose a technique for comparing sequences and describe fast techniques for finding matching sequences.

Clustering knowledge

Clustering is a technique concerned with identifying clusters of instances within the domain space. Clustering *is a form of unsupervised learning that partitions observations into classes or clusters (collectively called a clustering)* (Fisher 1995). As noted by Fisher, Clustering approaches can be defined in terms of the way they evaluate clustering quality (objective function) and the way that they search the space of clusterings (control strategy). The unsupervised nature of clustering makes it applicable to applications where the user has limited domain knowledge. An example application is the clustering of web-search results. Although several approaches to clustering exist and several different distance measures can be employed it has been demonstrated by Bouguettaya *et al.* that in some circumstances many of these methods have similar performance (Bouguettaya, Viet and Colea 1997). For the purposes of data mining, where large volumes of data are present it may therefore be more appropriate to choose clustering techniques based upon their efficiency.

In addition finding an appropriate control strategy and objective function there is a need to find methods to determine the optimal number of clusters within the data. Objects may be merged with nearest neighbours to form clusters, finding a stopping point for this process requires a technique that can estimate the optimal number of clusters in the data. Smyth addresses this issue and introduces a new technique based upon Monte Carlo Cross-Validation for determining the optimal number of clusters (Smyth 1996). Smyth compares his proposed algorithm with several existing approaches, concluding that the Monte Carlo Cross-Validation method offers an alternative to other methods.

Zamir *et al.* (1997) describe the application of clustering has been applied to web document retrieval. Ketterlin (1997) describes a bottom-up approach to clustering sequences of complex objects. This approach uses a hierarchical control strategy to find the least generalised covering cluster of the component objects in a sequence. Clustering is widely applied to spatial databases where clustering can be used to group items of close proximity in physical space. *Some examples of spatial clustering are described in Section 4.5.*

TARGET DATA TYPES

Most research into KDD has focused on the discovery of knowledge within the context of traditional database paradigms such as the relational model. However techniques designed for relational databases are likely to fully exploit only the relational types of knowledge implicit within other database models. Likewise the type of data being mined may be multi-dimensional, textual or graphical. This section describes knowledge discovery from eight different data sources that are distinguished by the nature of the data itself and the characteristics of its storage. Whilst we have limited our examination of data and its storage to eight major areas, KDD techniques are also applicable to other forms of data, for example Czyzewski (1996) describes the application of data mining techniques to the removal of noise from audio data. In this example the audio signal is broken down into bands which are then masked out or left unchanged based upon their noise content. A rough set technique is used to derive rules determining which segments of the signal should be masked from a set of sample data. These rules are then applied to the entire audio signal. This approach has yielded positive results and indicates the potential utility of applying KDD techniques to audio processing applications.

Relational Data

Relational databases are in widespread use throughout many organisations. Within the relational model data are normalised into relations to eliminate redundancy. As a result, data required for the purpose of data mining may be stored in several relations. To retrieve this data, relations must be joined, and required attributes projected out. For the purpose of data mining it is typical to assume that the data being processed is stored in a single relation created through the appropriate operations. The data is therefore presented in a simple tabular form. Most KDD tools developed to date are designed to operate on such tabular data sets and are therefore highly applicable to relational databases. Importantly however the task of generating a single relation for knowledge discovery may involve a large amount of data cleaning and manipulation.

Object Oriented Data

Object oriented databases (OODB), have the structure to model complex objects found in application areas such as computer aided design (CAD), software engineering and geographic information systems. This data typically contains hierarchies of objects and include concepts such as classes, inheritance, encapsulation and polymorphism, *qv.* Nahouraii and Petry (1991). In addition, objects have associated methods and communicate via messages. Whilst the OODB model allows for flexibility in modelling entities, the resulting lack of uniformity may hinder the KDD process. Importantly, OODB's such as *GemStone*, associate types with individual values, not the attribute fields they are stored in, (Maier and Stein 1990). Therefore a field or slot in an object may potentially contain data of any type and structure. Whilst stronger restrictions may be refined in the design process, the trade-off between uniformity and flexibility remains at the heart of the OODB paradigm. In general, where the data being stored is comparable to the data stored in relational databases, existing relational database techniques are applicable. However when more complex data exists, potential approaches to KDD become unclear. The complexity of object-oriented databases makes the application of KDD techniques a challenging problem. As part of the data filtering process existing knowledge discovery systems normally begin by selecting data from the target database in the form of one or more tables. Therefore the major focus is upon learning from tabular data. Whilst tabular data can be extracted from object-oriented databases for KDD purposes, methods for exploiting the more complex semantics associated with them remains a major challenge. To date little investigation has been undertaken in this area. However Nishio *et al.* (1993) describe the 'first step' towards knowledge discovery in object-oriented databases. Their work focuses on the extension of attribute-oriented induction techniques to object-oriented databases. By utilising an attribute-oriented induction technique, the primary challenge to be overcome is the creation of generalisation hierarchies for the complex data types typically found within object oriented databases. The existence of set based attributes poses a problem for the construction of conceptual hierarchies as several approaches to generalisation are available. Each item in the set can be generalised, and redundancy in the resulting set removed. Alternatively a generalisation may be a description of the overall set, such as the number of items, or the mean value of the items. In addition, an object may inherit attributes from one or more parent objects, provided that these attributes can be retrieved. It should be noted that data mining is likely to be of most use in large datasets and while OODBs have high structural complexity, their data volumes have, at least to date, been comparatively low.

Transaction Data

Much of the research associated with the generation of association rules assumes the availability of transaction data structured to list items association with a single commercial or other form of transaction. In the commonly used example, that of market basket analysis, each data record consists of those items that are purchased at the same time, together with optional additional information which may be of use. This is transformed into the data required.

Textual Data

The application of data mining techniques to textual documents to extract knowledge has become an increasingly interesting field with the advent of large textual databases. In recent years many models for the management of textual data have been proposed and developed, *qv.* Loeffen (1994). The lack of any standard model can be attributed to the complex issue of storing and retrieving textual data with its associated semantics. Different applications will inevitably require different semantics to be associated with textual data. For example, books may be structured into chapters, while a dictionary or thesaurus may be divided according to keywords. This lack of any uniform approach to the storage of textual data means that the development of *general-purpose* textual knowledge discovery tools is currently problematic and may be unrealistic. Whilst tools that can learn from raw, unstructured text are the most versatile, the utility of tools that cannot exploit the full semantics available is reduced. This is a serious limitation given the importance of structure in organising text by flagging keywords or similar attributes.

Most documents are not created in a format where structure is explicit. However structures exist in many types of documents and structural components can be identified by consistent types of formatting. Therefore these components can be identified within stored documents. Ahonen *et al.* describe a technique for finding a small description of a documents structure, in the form of grammars once its components have been identified (Ahonen, Mannila and Nikunen 1993). Their approach begins by constructing weighted finite-state automata, which are then generalised. The automata are generalised in relation to each other and finally transformed into regular expressions. Therefore if the components of a document can be identified, a grammar describing its structure can be derived. This may then be useful in indicating the type of knowledge associated with a document or conducting queries upon it.

A major application of knowledge discovery within textual databases is the development of classification rules to automatically classify documents. The primary objective behind this is to allow the automatic selection of documents of interest within a specific domain. Apté *et al.* (1993) describe an approach to document classification, which begins by extracting relevant keywords within documents as attributes. These attributes, along with an appropriate document classification, are then used to induce a set of classification rules with an associated measure of classification accuracy. The extraction of classification knowledge is greatly aided by the availability of large numbers of examples. However large numbers of example documents can pose a processing problem that may necessitate the use of a random sampling technique. In addition, the number of classifications to be learnt is also typically high. The most promising approach where pre-classified examples exist is to develop a classification model for a single class of documents at one time, with all other documents being used as negative examples. A third problem is created by the size of the dictionary of attributes or keywords. Apté *et al.* advocate a simple elimination of less significant attributes based upon frequency of occurrence, where the most commonly occurring attributes are used. This rule-based approach has been applied to several large document databases with promising results.

Hébrail and Marsais (1992) describe experiments on the analysis of research project descriptions. The experiments have been conducted within Electricité de France, where more than 1,500 research projects are undertaken each year. This large number of projects has made summarising the overall research being undertaken within the organisation a complex task. The application described analyses textual project reports to provide management with an overview of its research activities. Their approach utilises a custom thesaurus as a source of domain knowledge. The thesaurus contains over 13,000 keywords that are classified into almost 300 separate subject areas. Keywords are also linked via a semantic net, which models *synonymy*, *genericity*, *specificity* and *related topic* relations. This custom thesaurus is an integral part of the data analysis system, and it is likely that the creation of such a resource would be a major undertaking, however once completed it would be of general use in other areas such as document retrieval.

Where keywords are associated with documents they can be treated as data and associations between the keywords that commonly occur together can be investigated. Feldman and Dagan (1995) describe the KDT system for Knowledge Discovery in Text. This system operates upon texts marked with associated keywords. Keywords are organised into conceptual hierarchies describing the associated domain. By examining concept distributions knowledge about the texts can be derived. For example, in economic newswire data it may be found that crop producing regions of the world feature disproportionately in articles discussing agricultural concepts. By measuring deviations between expected distributions of concepts and actual observations interesting relationships can be found and changes in distributions can be tracked over time. Similarly Feldman and Hirsh (1996) describe the FACT system for discovering associations from text documents in the presence of background knowledge. Given a collection of documents with associated keywords, relevant background knowledge and a user-specified query the FACT system finds all of the appropriate associations between the keywords. The learning process is similar to the association rule learning algorithm of Agrawal *et al.* (1993). The FACT system employs a user interface that allows queries for associations to be easily defined. Importantly this system utilises background knowledge to restrict the search space and hence the incorporation of background knowledge improves the learning efficiency of the tool. The FACT system has been successfully applied to newswire data.

Temporal Data

Traditional databases store only the current state of the data, so that when new values become valid, old values are overwritten. This approach cannot model the way the entities represented in the database change over time. Temporal databases overcome this limitation by not overwriting attribute values, but instead storing valid time ranges with them, which can be used to determine their validity at particular times, including the present[23]. This functionality is further extended by the fact that future values can be entered proactively in preparation for their impending validity, ie. before they become current. In traditional databases, reasoning about temporal information is restricted to comparisons between temporal valued attributes. Temporal information in such systems is treated in the same way as other attributes. For example, Lee *et al.* (1985) describe the application of temporal inference to administrative databases. For a survey of temporal semantics in information systems see Roddick and Patrick (1992).

Whilst much progress has been made in the development of temporal databases (Tansel, *et al.* 1993), little progress, has been made towards the development of general-purpose temporal data mining systems. A discussion of some issues involved with temporal knowledge discovery is provided by Rainsford and Roddick (1996). Likewise a theoretical framework for temporal knowledge discovery described by Al-Naemi (1994). Within the context of bitemporal databases Roddick describes a formal method of defining induced temporal relationships termed temporal induced dependencies in (Roddick 1994), see also (Roddick, Craske and Richards 1996). Temporal induced dependencies are induced functional dependencies that are weakened by temporal conditions. These dependencies may only be valid at particular times, or may specify a temporal relationship between part of the functional dependency. Temporal dependencies can be induced using techniques such as the characteristic rule learning algorithm described by Han *et al*, (Han, Cai and Cercone 1993). However this is only possible if a framework for generalising temporal intervals is provided. Rainsford and Roddick present a simple framework for facilitating the generalisation of temporal intervals in the context of attribute-oriented algorithms (Rainsford and Roddick 1997).

A temporal database is not essential for temporal knowledge discovery. For example Hoschka and Klösgen (1991), describe the potential for limited temporal reasoning within the *Explora* system. The suggested temporal reasoning is added by storing separate snapshots of the rule set over time. These rule sets can then be compared to draw conclusions regarding the change in data over time. This technique could be applied to any non-temporal database to allow some temporal reasoning. However because data is not stored within a temporal database, rules describing the change in the data over time can only be derived indirectly from changes in the stored rule set. Because the snapshots are derived without any knowledge of temporal patterns existing within the data, many interesting temporal patterns may be lost. Moreover the fact that only a limited number of views of the rule set are available restricts the reasoning capability of such a system. For example temporal behaviour such as cycles may not be detected if the time between rule sets is too great. In addition, the ability to find rules describing the change in data over time is dependent upon, and restricted by, the information held in the stored rule sets. For these reasons, the use of temporal databases in situations where temporal semantics are meaningful is arguably a better approach.

In order to describe relationships between temporal events a taxonomy of temporal relationships is required. A widely used taxonomy of relationships between intervals is described by Allen (1983). As an example of the role temporal relationships play consider the rule:

> The blackouts occurred after the peak period

The above rule uses the temporal relationship *after* as an expression of the relationship between two intervals. The taxonomy of Allen is generalised by Freksa (1992), who describes an algebra based upon semi-intervals. This approach supports reasoning with partial knowledge, where only one endpoint is known. In addition, Freksa's taxonomy allows the coarseness of temporal reasoning to be adjusted via neighbourhood relations to accommodate the information available.

In addition to relationships between temporal events, there has been significant investigation into tools for finding pre-specified patterns within temporal data. For example, Wade *et al.* (1994) describe a set-based approach to detecting temporal patterns in patient drug usage data. Drug misuse can occur unwittingly, when a patient is prescribed two or more interacting drugs independently for usage within temporal proximity to each other. Drugs that interact undesirably are recorded along with the time frame in the form a pattern that can be looked for within patient records. Rules that describe such instances of drug misuse are then successfully induced based on medical administrative records.

Whilst Wade *et al.* focus on the detection of patterns within tabular data, the detection of patterns within continuous data is more applicable to many domains. Berndt and Clifford describe the detection of patterns in time series data (Berndt and Clifford 1995). They adopt a dynamic time warping technique utilised in natural language processing. As stated by the authors, the discovery of patterns within time series data is a challenging

[23] The three types of time that can be associated with an attribute in a database are: Valid time; the time at which events actually take place within the modelled reality; Transaction time; the time at which data is actually entered into the database, ie. the time transactions actually take place; and User-defined time; any attribute within the database which records time, for example, date fields etc.

problem, and the development of general-purpose tools poses several problems. For example, the number of possible patterns a linear series can follow is potentially very large. Therefore in order to develop tools to look for such patterns within data, a comprehensible and manageable set of patterns must first be defined. A shape definition language *SDL* is defined by Agrawal *et al.* to describe patterns or shapes occurring in historical data (Agrawal, *et al.* 1995). Based upon *SDL* a query language for defining time series patterns and trends is described by Agrawal and Psaila (1995). One obvious limitation of such a query language is that linear patterns are most intuitively described visually, and typically textual descriptions involve the use of informal language. Agrawal and Psaila allow the user to create their own language with complex patterns being defined in terms of primitives such as up or down.

There are two significant problems that are associated with pattern detection from time-series data, scale and proximity. The problem of scale is posed because searching for patterns at different resolutions will yield different results, and the choice of resolution is largely domain dependent. Patterns that may be significant in one set of time series data may only be considered noise within a different domain. Therefore the resolution at which to search for temporal patterns remains a parameter which must be specified by users. However patterns existing over different time scales may all be interesting and therefore methods of considering such issues must be investigated if automated tools are to be developed. The proximity of events in time determines if any significant relationship between them can be drawn. For example the fact that an engine failed after a power surge may not be interesting if the power surge occurred thirty years before the engine failed. If however the events occurred within moments of each other then the relationship may be significant. Many application domains may involve delayed reactions and therefore determining an appropriate proximity is essential. Windowing techniques are commonly employed to tackle the problem of proximity. A time window is an interval within which data is examined only in the context of the other data in the window.

The *RX* project (Blum 1982), utilises several sub-modules to discover causal relationships from temporal data within a medical domain. *RX* utilises non protocol and non randomised data or in other words, no special data acquisition techniques are assumed. The data used is from an operating database where data input is typically on a day to day basis. Other notable contributions to temporal knowledge discovery include the work of Mannila *et al.* who describe an algorithm for the discovery of frequently occurring episodes in sequences (Mannila, Toivonen and Verkamo 1995). An extension of this work employing temporal logic is described by Padmanabhon and Tuzhilin (1996).

Spatial Data

Spatial databases model multi-dimensional space and are typically found within geographical information systems (GIS) (Abraham and Roddick 1998, 1999). The complexity of spatial data necessitates the development of special purpose data mining tools. It is important to note that spatial and non-spatial information will typically need to be integrated into any learning system. Therefore, whilst existing tools and techniques remain partially valid, further extensions to accommodate spatial reasoning are required. Although the application of KDD tools to spatial databases is relatively new, some work has already been undertaken. From this initial research, two main approaches to knowledge discovery within spatial databases have emerged. The first approach is based directly upon spatial operations, where properties such as distance and proximity are used directly within discovery, in conjunction with techniques such as clustering. Clustering can be defined as the process of grouping physical or abstract objects into classes of similar objects (Chen, Han and Yu 1996). The second approach is based heavily upon the relational model, where spatial attributes are converted via spatial operations into corresponding attributes and these attributes are processed using largely conventional techniques with extensions to include spatial operators.

Bell *et al.* (1994) describe their experience of data mining within spatial databases. Their technique applies the Dempster-Shafer Theory of Evidence to the location of volcanoes on Venus. Volcanoes are located from images captured by the Magellan-Venus space probe and evidential reasoning is used to combine the spatial evidence from multiple images. Images differ in quality due to the angle at which they were taken and their resolution. With evidential theory, the evidence provided by each of the images can be assigned a weighting to reflect the image quality. This approach deals with images representing spatial data, however other techniques may be required for knowledge discovery within spatial databases where the spatial data is represented in other ways.

Rule discovery mechanisms used for non-spatial data can potentially be extended to include spatial information. For example, Koperski and Han describe spatial association rules (Koperski and Han 1995). One of the important points highlighted in their work is that existing spatial processing functions should be exploited where possible and existing techniques for relational data are often useful within spatial knowledge discovery. They define spatial association rules and provide an algorithm for their top-down induction within relational based GIS systems. Spatial association rules can be loosely described as conventional association rules as defined by

Agrawal *et al.* (1993), with at least one spatial attribute and the potential for spatial predicates such as, *close-to*, *within* or *next-to* describing the association within the rule. Therefore a simple example may be:

$$\text{is\_a}(X, \text{city}) \wedge \text{within}(X, \text{Australia}) \rightarrow \text{close\_to}(X, \text{coast})$$

Spatial relationships need to be defined by domain experts, and may have various interpretations at different levels of abstraction, which also need to be defined. For example, a country may be defined as being *close-to* a city if it is 50 km away, but a backstreet may not be defined as being *close-to* a school if it is also 50 km away. Therefore depending upon the level of abstraction being discussed, spatial terms need to be re-interpreted. Whilst this approach provides support for natural language semantics, it creates the potential for ambiguity and a requirement for an increased level of expert input.

The Koperski and Han algorithm exploits pre-defined conceptual hierarchies to conduct top-down induction. Initially the spatial data of interest is extracted and processed to find support for any associations fitting the user-specified pattern. Strongly supported rules at this high level of abstraction are then identified. More specific rules can then be derived from the high level set by descending the conceptual hierarchy and these rules tested for adequate strength within the database. The required minimum support threshold can be lowered for more specific rules, where less data is available. By continuing down the conceptual hierarchy, seeking more specific rules from the previous rule set, association rules at all levels can be discovered. This approach enables the discovery of rules at various levels of abstraction. It also reduces processing costs, by eliminating large sections of uninteresting data at a high level of abstraction, and avoiding futile searches for strong rules. However this approach relies upon detailed domain knowledge in the form of conceptual hierarchies and the definition of relationships at multiple levels.

Lu *et al.* also employ conceptual hierarchies in the induction of characteristic rules from spatial data (Lu, Han and Ooi 1993). They describe basic algorithms for attribute oriented induction of spatial and associated non-spatial data. The induced rules characterise the non-spatial properties and relationships of spatial objects, and the learning process is initiated by a user learning request that may focus on attributes of interest. Spatial generalisation may be performed with the use of existing conceptual hierarchies, clustering techniques or even spatial indexing structures. Generalisation is performed on spatial and non-spatial data, and the order in which this is done effects the resulting generalisation rules. This leads to the specification of two basic algorithms, the non-spatial-data-dominated algorithm that generalises non-spatial attributes first, and the spatial-data-dominated, which generalises spatial attributes first. The possibility of algorithms for interleaving spatial and non-spatial generalisation is also discussed.

An alternative technique for knowledge discovery in spatial databases is the application of clustering techniques. Ng and Han (1994) describe *CLARANS*, a clustering technique designed for spatial data mining. The effectiveness of *CLARANS*, and the techniques it is based upon, rests strongly in its ability to determine the central representative object or *mediod* for each cluster. This process involves testing candidate mediods against their neighbouring objects. Once this first step has been performed other objects can be assigned to the appropriate cluster based on their proximity to the mediod. However an exhaustive search for the best mediod within the dataset is impractical within very large datasets. Rather than look for the mediods within sample sets of the data, *CLARANS* works with the complete dataset, but only tests a mediod against samples of neighbouring objects. Both a spatial dominant version *SD(CLARANS)*, and a non-spatial dominant version, *NSD(CLARANS)* are described. The spatial dominant version performs clustering of spatial attributes before applying the *DBLEARN* system (Han, Cai and Cercone 1993), to the non-spatial attributes associated with each cluster. This approach provides a non-spatial description of the spatial clusterings. The non-spatial dominant approach applies *DBLEARN* algorithms to the non-spatial data first, before applying *CLARANS* to the spatial attributes associated with each of the generalised tuples. This approach reveals spatial clusterings existing within groupings of non-spatial items.

CLARANS is further investigated by Ester *et al.* who describe the efficient application of this clustering algorithm to spatial databases, (Ester, Kriegel and Xu 1995). They describe performance enhancement techniques showing a slightly reduced effectiveness for a large gain in efficiency. Such a trade-off is likely to be highly desirable in applications involving very large volumes of spatial data. They apply their technique to a large protein database, to aid in the identification of similar protein surfaces. In addition, Ester *et al.* point out that clustering techniques are not dependent upon domain knowledge.

Combinatorial Data

Combinatorial data contains complex objects such as trees, graphs and sequences that often exist in databases describing sophisticated domains such as circuit layouts, molecular structures or computer code. One application for knowledge discovery tools in combinatorial domains is to discover generalisations of data, by finding substructures which occur repeatedly and can be substituted and hence reduce the volume of the encoded data. Finding similarities between complex objects is often too difficult to perform by inspection and this can also be performed using knowledge discovery techniques. The emphasis in combinatorial data mining is upon identifying patterns in the underlying structure.

Djoko *et al.* describe SUBDUE a tool for the use of domain knowledge in the discovery of substructure within combinatorial data (Djoko, Cook and Holder 1995). The input data can be substituted with a pointer to the appropriate substructure. Statistical measures of the goodness of fit between the substructure and the input data determine if the fit is sufficient to support the substitution of the input data with the pre-defined substructure. Once discovered, these substructures can be used to form a compressed representation of the original data or employed for the purposes of providing an overview of what structures exist in the data. Substructures are modelled in a hierarchy and occur at multiple levels. Therefore they can be generalised and decomposed into other substructures within the hierarchy.

The discovery of similar substrings in protein sequences is discussed by Wang *et al.* (1994). Their approach is based upon the initial selection of appropriate patterns from a sample of the database. These patterns are then compared against the rest of the database. Their technique can find non-consecutive patterns separated by arbitrary lengths without prior knowledge of their structure or occurrence. The diversity of structure than can be potentially found in combinatorial data is limited only by the application domain. Therefore techniques for combinatorial knowledge discovery may require application specific approaches.

Data from the Internet

The exact manner in which knowledge discovery can be applied to the Internet is still an area open to investigation. The standards that exist within the Internet are largely associated with data presentation and navigation, while issues related to storing large volumes of data are comparatively overlooked. The enormous volume of data resources makes the Internet a useful target and source of input for knowledge discovery. However the lack of standardised storage formats, the wide diversity of data types and the wide distribution of data across the Internet hinders general-purpose knowledge discovery over the Internet. One possible solution is to construct multi-level databases upon the raw data; an approach described by Han and Fu (1994). By proving a generalised summary of the data at a high level of abstraction, tools could locate information of interest quickly and then perform knowledge discovery upon the low-level data. However the extent to which such an approach will be universally adopted is debatable. Another approach could be the use of intelligent agents capable of learning from diverse sources of data and then collaborating with each other in the summation of knowledge and presentation of results. Little investigation has been undertaken into the application of intelligent agents to knowledge discovery but Davies *et al.* examine an intelligent agent approach towards data mining using first order logic (Davies and Edwards 1995b).

One application of data mining to the World Wide Web is the analysis of usage logs to characterise user behaviour patterns. Once specific user groups have been identified, websites can adapt to offer them customised information and hyperlink pathways through the data. Chundi and Dayal (1997) describe a technique for providing a list of links for web users based upon previous access patterns. The approach uses clustering techniques to classify website clients into various user groups based upon their access behaviour. Once identified as belonging to a specific group the tool can offer further links based upon links most likely to be useful. The approach contains an adaptive component that can adjust the links presented over time as new behaviour patterns are observed. Although this approach is limited to presenting prospective links, it could be extended to provide alternative information, details and customised advertising offers, based upon the user profile.

Data from Data Warehouses

In a traditional database environment the processing to answer queries and support analysis is conducted as the user requests arise. This may involve the integration of data from multiple databases and information sources. This data may subsequently require reformatting and adjustment to create a single integrated source of data from which user analysis can then be conducted. As both the volume of data and the value of information has grown the importance of analysing and exploiting organisation data has increased. The desire to integrate and hence analyse data more efficiently has led to the emergence of data warehousing technology. Defined simply a data warehouse is *an analytical database that is designed for large volumes of read-only data, providing intuitive access to information that will be useful in making decisions* (Fong and Zeng 1997). Data warehouses store integrated data in predefined formats specifically selected to support user queries and analysis. This data is typically gathered from multiple databases and information sources and may contain historical data as well as metadata. The rapid adoption of data warehouse technology by organisations means that much knowledge discovery is now likely to be undertaken in the context of data warehouses.

In an effort to support data analysis data warehouses store information that can be utilised by knowledge discovery tools. Inmon identifies four types of data characteristic of a data warehouse that may be useful for knowledge discovery (Inmon 1996). Firstly, integrated data in a data warehouse reduces the need to combine, clean and reconstitute data that may originate from multiple sources and hence allow speed up the knowledge discovery process. Secondly, detailed and summarised data allows data mining tools the option of analysing data at various levels of abstraction without needing to perform any generalisation. Thirdly, historical data supports longitudinal analysis such as the detection of trends and cycles within the modelled domain. Fourthly, metadata provides useful contextual information that can aid in the knowledge discovery process. All of these data types may be derived independently however their presence in a data warehouse would avoid the need to create them hence saving time in the preparation of data for the knowledge discovery process.

In the same way that existing KDD tools can exploit the database management system, and query languages of existing databases, KDD tools can also exploit the technologies associated with data warehouses such as OLAP (Online Analytic Processing) and data cube technologies. Kamber *et al.* describe the use of data cubes for metarule-guided mining of association rules (Kamber, Han and Chiang 1997). The metarule in this context refers to a template that acts as a pattern filter, restricting the search space. When a data cube exists the large one item itemsets can be found by examining the 1-D aggregation layer of the cube. Based upon these large itemsets candidate two item large itemsets can be constructed and tested by examining the 2-D aggregation layer. This process continues until itemsets with the same number of items as appearing in the metarule have been discovered. From here the large itemsets can be evaluated as rules. This approach avoids multiple scans over the entire database by utilising the summary information held in the data cube. Overall it can be seen that data warehouse technology provides an improved environment for knowledge discovery processes and it is likely that many KDD tools will be adapted to exploit this.

THE UTILISATION OF DISCOVERED KNOWLEDGE

Both data mining and KDD are usually associated with decision support and knowledge base creation. However other opportunities exist for the exploitation of discovered knowledge. For example, knowledge can be used for the detection of inconsistencies and integrity enforcement. Other applications include semantic query optimisation and the discovery of hidden structures or dependencies within the data, which may lead to database restructuring. The choice of application largely dictates the nature of discovered knowledge and hence the discovery processes. This section will discuss four main applications of discovered knowledge and provide examples of each.

Detection of Inconsistencies and Enforcement of Integrity within Databases

The detection of inconsistencies and enforcement of integrity within databases can be partially automated with the use of inductive techniques. Semantic rules can be induced from the target database and any violation of these rules can then be flagged as an exceptional occurrence, and appropriate warnings can be activated. As pointed out by Schlimmer, a major strength of this approach to integrity enforcement is that it can be applied to domains where no domain expert is available (Schlimmer, Mitchell and McDermott 1991). Furthermore, the integrity rules can be automatically updated periodically to reflect changes in the database over time.

Schlimmer (1993) describes *Carper*, an inductive learning tool used to maintain integrity within databases. This tool combines both learnt and given knowledge to construct attribute range/value models. These models can be used to detect inconsistencies in database entries. In addition existing database entries can be checked for validity against the rest of the database. *Carper* constructs a single decision path for the entry being checked. The adaptation includes the generation of multiple trees where two attributes are equally appropriate for branching. This reduces the number of false alarms also reduces the number of actual violations detected but to a lesser degree. Therefore the choice of using a single tree or multiple tree in such situations, should be determined by the nature of the application.

Kamel describes the use of an expert system shell as a front end to a database to test the integrity of updates (Kamel 1995). This simplistic approach allows rules in the expert system to enforce integrity upon data entered into the database. Rules for such a system could be automatically generated using a rule induction technique. However for such a system to be practical the rules would need to be maintained consistently over time to ensure they facilitate changes in the database over time.

It is worth noting that active databases could be employed usefully for tasks such as the detection of inconsistencies and integrity enforcement. Active databases can respond automatically to the entry of unusual data by notifying the system user of the unusual nature of the data. The entry of some quantity of new data can be seen as an event. If any inconsistencies are found between this data and the rules derived previously via data mining then this can activate a trigger. The action taken by the trigger could be to notify the user of the exception nature of the data in question. An application of active databases in knowledge discovery is described by Agrawal and Psaila (1995) who describe the application of triggers to detect trends in rules describing the application domain. Important changes in the application domain can be automatically flagged for users attention. There is no reason however why a similar approach cannot be adopted for integrity enforcement. The major advantage of this approach is that users are not required to manually conduct regular analysis to detect interesting behaviour in the rule set and underlying data.

Semantic Query Optimisation

Semantic knowledge about the contents of a database can be useful in the optimisation of queries. Rules that are known to hold in the database can be used to transform queries to improve performance, a process known as semantic query optimisation. Semantic query optimisation can be seen as a two-phase process (Siegal, Sciore and Salveter 1991). Firstly possible transformations of the original query to produce different, semantically equivalent queries must be found. These transformations are performed based upon available rules or semantic knowledge and it is this knowledge that might be derived by knowledge discovery. The optimisation process must then determine which query will has the lowest execution cost. As an example consider the following database relation noting that there is an index on the Department attribute:

STAFF (Id, Name, Department, Building, Room No)

Consider a simple query that requests a list of all lecturing staff in the Mitchell building.

```
Select STAFF.Name
Where  STAFF.Building = 'Mitchell'
```

If we have a rule that states that all staff in the Mitchell building are from the Computing department, we may then transform the query into the following query:

```
Select STAFF.Name
Where  STAFF.Department = 'Computing'
And  STAFF.Building = 'Mitchell'
```

In this transformation of the original query we then may utilise the index structure to speed up the query by quickly selecting only the Computing staff.

As noted by Siegel *et al*, no methodology for the specification of useful semantic rules by experts has been developed (Siegal, Sciore and Salveter 1991). Therefore it cannot be guaranteed that the rules specified by experts would be optimal for semantic query optimisation. Moreover the maintenance of semantic rules may place an impractical burden on domain experts as databases are constantly changing. The automatic generation of useful semantic rules is therefore a promising application of knowledge discovery techniques.

Anand *et al.* (1994) describe *state-aware* query optimisation. This approach aims to use discovered semantic knowledge to reformulate queries to better utilise available resources. The system is described as state-aware, because it reformulates queries to reflect the hardware resources currently available on the system. The *STRIP* algorithm (Anand, *et al.* 1995a) is used for the data mining component.

Siegel *et al.* (Siegal, Sciore and Salveter 1991; Siegel, Sciore and Salveter 1992) describe the automatic derivation of rules for semantic query optimisation. This process first describes the characteristics of desirable rules. A search of the database is then undertaken to see if such rules can be derived. The explicit definition of the type of rule being searched for confines the search space and this consequently improves the efficiency of the search process. As noted by the authors the search for such rules is favoured by the availability of a definition of the rule type required and a means of evaluating the utility of the result.

Yu and Sun (1989) describe an approach to semantic query optimisation that makes use of previously executed queries as a source of knowledge. They define the difference between static integrity constraints and dynamic integrity constraints. Static integrity constraints are those constraints known to exist permanently within the current operation of an organisation. Dynamic integrity constraints are constraints that currently hold true for the data, but may not remain true in the future.

Given that learning and storing rules for semantic query optimisation uses system resources it would be beneficial to concentrate resources upon learning semantics that are useful in answering queries. One approach to this is to have a query driven approach that learns semantic rules based upon user queries. Hsu and Knoblock describe a system for semantic query optimisation that uses user query patterns to direct the learning of semantic optimisation rules. Expensive queries trigger the learning process and hence the search for optimisation rules. Therefore the knowledge available for semantic query optimisation will match user query patterns. Importantly this approach will allow complex joins across multiple relations to be used for semantic query optimisation. Because the learning process is triggered by user queries then superfluous joins will not be investigated. Testing has shown that this technique is effective however a mechanism for updating outdated semantic optimisation constraints is still being investigated.

Knowledge Base Creation for Expert Systems

Discovered knowledge can be used to construct knowledge bases that can then be exploited to create expert systems. An expert system can be described as *a computing system capable of representing and reasoning about some knowledge-rich domain ... with a view to solving problems and giving advice* (Jackson 1986). The automatic creation of knowledge bases is attractive for several reasons. In situations where human experts are unavailable knowledge bases can be constructed from data sets. Automatically discovered rules can also be used to verify rules proposed by human experts. The use of data to construct knowledge bases can also overcome problems encountered when trying to extract information from human experts such as efficiency and objectivity. Moreover, because databases may contain vast numbers of previous records they can be seen as a valuable source of data from which expert system algorithms can learn.

A distinction should be made between the two extremes of learning from sets of specially selected training data and learning from databases. Many researchers have successfully derived expert systems from prepared training data. For example, Carter and Catlett describe the use of machine learning to derive various decision trees to assess credit card applications, (Carter and Catlett 1987). The experimental decision trees were created using *ID3* and *C4*, and displayed better accuracy than the existing organisational categorisation technique. By comparison, extracting expert knowledge directly from databases poses greater problems because of the need to deal with comparatively unprocessed data, and has received less attention to date. One example of a system that can be used for deriving expert system rules from a database is the *RULEARN*® system described by Koch and Fehsenfeld (1995). Taking a large set of data, this system characterises the data into a small number of rules, describing the dependencies existing in the database.

In addition to the creation of new knowledge bases it is also possible to use KDD techniques to maintain and update existing knowledge bases. For example Schlimmer *et al.* describe *Cobble*, a system to aid in the development and refinement of knowledge bases (Schlimmer, Mitchell and McDermott 1991). *Cobble* is used to test the conditional aspect, or left hand side of rules, determining if they are too specific. The *Cobble* system uses both the knowledge base being scrutinised and appropriate supplementary or background information such as domain terms or task-independent facts. Within *Cobble*, knowledge base rules are tested by comparing them for consistency with background knowledge. Generalised conditions for a rule are derived and these are compared with the existing conditional component of the rule. Any unnecessarily restrictive conditions are then removed or replaced with more general ones.

The Use of Discovered Rules for Decision Support

A final application for the use of discovered knowledge is as a resource for decision support. Decision support typically involves ad-hoc analysis of data that is guided by the user in an exploratory way. The facilitation and utilisation of user guidance is therefore a significant factor for decision support tools. Many existing KDD tools are well suited to decision support in this respect because they allow users to direct the search for knowledge. Conventional decision support tools commonly allow decision-makers to query data directly and conduct *what-if* analysis. KDD tools complement the power of decision support systems by allowing the search for more complex patterns and semantic relationships to be undertaken. The selection of an appropriate KDD tool depends upon the nature of the database to be utilised and the nature of the desired knowledge. Association rule learning systems such as *AIS* (Agrawal, Imielinski and Swami 1993) are likely to be appropriate for the analysis of supermarket transaction databases. Similarly the classification rule learning algorithm of Cai *et al.* (1990) may be applicable to the analysis of a database of insurance claimants.

Highly autonomous KDD tools may potentially pre-empt decision making by identifying opportunities from relevant organisational data. With this in mind, another application of discovered knowledge might concern the decisions made by database administrators. Knowledge concerning hidden structures or dependencies within the database can potentially support or pre-empt decisions concerning database re-structuring. For example the identification of previously unknown functional dependencies can lead to a restructuring of the database to reduce redundancy and ensure integrity.

Security Implications of Knowledge Discovery

As the reasoning power of induction within databases has become widely acknowledged, the potential misuse of inductive learning tools to induce restricted information has become an area of concern. Organisations may unwittingly provide classified information implicitly within non classified data. At the same time security of data within databases is increasingly important as sensitive information is stored within them. In a progress report on KDD, Piatetsky-Shapiro (1994) includes privacy of data and related ethical and legal issues in a list of difficulties to be overcome within KDD. With the worldwide proliferation of electronic information systems and databases, the disclosure of data has received increasing examination both legally and ethically. However issues surrounding the social and legal implications of emerging KDD technology remain largely unresolved.

O'Leary points out that whilst traditional approaches to database security address the unauthorised access of data, they do not address the unauthorised acquisition of knowledge from data (O'Leary 1991). As a consequence authorised users may use the data that they have access to in attaining knowledge restricted from them. He also warns that confidential information relating to decision making could potentially be induced from examples of past decisions. Such information could help criminals avoid detection, allow competitors to pre-empt decisions or expose confidential decision making criteria. Given these concerns O'Leary argues that the threat posed by KDD technology to organisational security warrants specific counter-measures. Traditional techniques to avoid intrusion remain useful as induction can be performed upon data obtained with or without permission. Such techniques could however be extended where required to detect suspicious activities such as the dumping of large numbers of records, which may be used for inductive techniques. In addition O'Leary also points out that approaches which exploit the weaknesses and limitations of existing inductive tools should be investigated.

Application Area	Example Applications
Astronomy	Sky survey analysis (Fayyad, Weir and Djorgovski 1993).
Database	Semantic query optimisation (Siegal, Sciore and Salveter 1991), integrity enforcement (Schlimmer 1993). Discovering missing semantics (Li, Huang and Chen 1997). Scheme discovery (Miura and Shioya 1997).
Engineering	Automotive quality control (Wirth and Reinartz 1996). Semiconductor fault diagnosis (Saxena 1993).
Finance	Credit assessment (Feelders, le Loux and van't Zand 1995). Stock market analysis (Ziarko, Golan and Edwards 1993).
Geology	Earthquake detection and measurement (Stolorz and Dean 1996).
Insurance	Premium setting, workflow analysis (Keats and Loo 1997). Behaviour patterns in health insurance (Viveros, Nearhos and Rothman 1996).
Law	Fraud detection (Shortland and Scarfe 1994).
Marketing and Sales	Quick Market Intelligence (Alexander, Bonissone and Rau 1993). Market Survey Analysis (Ciesielski and Palstra 1996). Insurance marketing analysis (Keats and Loo 1997).
Medical	RNA study and analysis (Hofacker, *et al.* 1996; Wang, *et al.* 1996) Drug side effect detection (Wade, *et al.* 1994). Discovery of causal relationships (Blum 1982). Diagnosis of acute abdominal pain (Provan and Singh 1996). Diagnosis of headache and facial pain (Tsumoto and Tanaka 1996).
Physics	Spectral data analysis (Buntine and Patel 1995).
Sport	Basketball game analysis (Bhandari, *et al.* 1997).
World Wide Web	Analysis of usage patterns and hyperlink customisation (Chen, Park and Yu 1996; Chundi and Dayal 1997).

Table 2. KDD application areas

The threat to security imposed by inductive techniques need not imply the use of KDD tools. Miller describes the way that data within a statistical database can be compromised when supplementary knowledge exists (Miller 1991). A statistical database typically contains knowledge made public on the condition that individual subjects from whom the information has been derived are not identified with the data. This form of anonymity is considered essential when dealing with detailed surveys such as census data however supplementary knowledge can compromise this. Supplementary knowledge can be loosely defined as background knowledge known by the user. Miller argues that supplementary knowledge is a major threat to database security. This argument is supported by actual examples of database compromises, reached through the application of supplementary knowledge to statistical databases. That is to say that where a system user has background information about a subject they may use this to isolate that subject and thereby removing their anonymity. She defines a classification system for various types of supplementary knowledge and compromises.

What data is used for knowledge discovery and how the results of knowledge discovery are used are the two main security issues. The source of data used for knowledge discovery and the consent of its use by the individuals concerned are significant factors in determining its sensitivity. Klösgen distinguishes between primary data collected explicitly and secondary data collected as a by-product of other transactions (Klösgen 1995a). The application of knowledge discovery techniques to secondary data collected for purposes other than knowledge discovery is a particular concern. Klösgen also highlights the relative lack of restrictions placed upon the handling of sensitive data by private organisations. This is in contrast to government agencies that often have high levels of self-regulation imposed upon data collected for official purposes. Klösgen proposes an architecture for knowledge discovery systems where the data source is isolated from the analytical components of the KDD tool by a data management component. This data manager enforces appropriate security restrictions on the data passed onto the analysing components of the knowledge discovery system.

Application Domains

Knowledge discovery in databases has been applied in many application areas where the volume of data is large. We provide a table of application areas updated from the table provided by Frawley *et al.* (1991) see Table 2. While this list of application areas is impressive, the potential for the application of data mining technology to other areas still remains. As databases continue to grow in size and new databases are created, the motivation to automatically extract implicit knowledge will continue to grow. It is also likely that in some application areas automatic discovery will become necessary as data volumes make manual analysis infeasible.

FUTURE RESEARCH

Much work has already been undertaken in the development of data mining systems. However challenges remain, four of which are discussed below.

Interestingness

The problem of specifying the *interestingness* desirable in required rules. A system that constantly reports useless facts is unlikely to be popular amongst users. It should be noted however that for some applications, such as semantic query optimisation, the notion of what is interesting is complicated as rules which exist in the data, meaningful or not, can often be useful for optimising operations restricted to that same set of data. Whilst the search for interesting rules is clearly the objective, the problem of differentiating between interesting and non-interesting rules remains largely unsolved.

Nevertheless, some suggestions have been made (Piatetsky-Shapiro 1994), and the issue of interestingness in knowledge discovery is discussed by Silberschatz and Tuzhilin (1996). They distinguish between objective and subjective measures of pattern interestingness. Objective measures relate to the structure and statistical strength of the pattern while subjective measures also consider the user viewing the patterns. This distinction is important because while objective measures are easily applied to a set of patterns, subjective measures are not easily defined. Silberschatz and Tuzhilin propose *Unexpectedness* and *Actionability* as fundamental characteristics that make patterns interesting to users. Unexpectedness is a measure of how the discovered pattern differs from the user's beliefs and hence how surprising it is. Actionability measures the ability of an interesting piece of knowledge to be acted upon by the user. They also argue that actionability and unexpectedness typically occur together in patterns. Based upon this assumption, it is sufficient to evaluate patterns on the basis of unexpectedness and avoid difficult measures of actionability. Modelling the users current beliefs considers both hard beliefs that are fixed and soft beliefs that may change over time. By measuring the deviation of newly discovered knowledge from the users current beliefs a relative measure of interestingness can then be derived (Srikant and Agrawal 1995).

User Dependence

A major limitation of existing systems is their reliance upon user input and direction. This dependence of data mining systems upon the user comes in two forms. Firstly, there is still a heavy dependence of data mining tools upon available domain knowledge. Such information often needs to be specified explicitly by domain experts.

The development of techniques to automatically acquire domain knowledge, or facilitate its acquisition may offer one partial solution to this problem.

Secondly, KDD tools require substantial run-time support from the user. Ideally, KDD tools could operate more or less autonomously in the background, presenting interesting results as they are found, and applying this new knowledge where appropriate. One major reason why more fully automated tools have not been developed is the question of defining *interestingness* mentioned above. The use of intelligent agents to discover and report interesting findings to users is an area of significant interest. Davies and Edwards have investigated this area and developed an initial framework for the operation of agent based data mining (Davies and Edwards 1995a, 1995b). Their approach focuses on distributed agents co-operating to find interesting knowledge.

Other Application Domains

Apart from overcoming existing limitations, the potential to extend KDD tools to more diverse application domains, is a major focus of the current effort. Some issues related to this are listed below.

- The diversity of data being stored necessitates the development of specific techniques optimised to particular data types and data storage models. The development of tools capable of fully exploiting the associated semantics remains a challenge in areas such as temporal, spatial and object oriented databases.
- There has been little investigation of hybrid or co-operative systems employing multiple paradigms for machine learning. Because different learning paradigms have different strengths and weaknesses, it is likely that a system employing more than one paradigm may be more effective than a single paradigm approach.
- Whilst many applications for discovered knowledge exist, most KDD systems only incorporate support for one or two. Systems that can support many applications, making efficient use of discovered knowledge by using it in multiple ways, are still to be fully investigated.

In summary, both opportunities and challenges remain within the broad field of KDD. As might be expected, many of these issues are currently the focus of current research efforts. It is likely that as the trend towards larger and more semantic rich databases continues, the opportunities for KDD research and application and prototype development will grow. In the longer term, it is highly likely that the development and use of appropriate KDD techniques will become a commercial necessity for competitive performance.

REFERENCES

Abraham, T. and Roddick, J. F. (1998). "Opportunities for knowledge discovery in spatio-temporal information systems". **Australian Journal of Information Systems.** Vol. 5, No. 2, pp. 3-12.

Abraham, T. and Roddick, J. F. (1999). "Survey of spatio-temporal databases". **Geoinformatica.** Vol. 3, No. 1, In Press.

Agrawal, R., *et al.* (1992). "An Interval Classifier for Database Mining Applications". **Eighteenth International Conference on Very Large Data Bases,** Vancouver, Canada, pp. 560-573.

Agrawal, R., *et al.* (1993). "Mining Association Rules Between Sets of Items in Large Databases". **ACM SIGMOD International Conference on Management of Data,** Washington DC, USA, ACM Press. Vol. 22. pp. 207-216.

Agrawal, R., *et al.* (1996). "The Quest Data Mining System". **Second International Conference on Knowledge Discovery and Data Mining,** Portland, Oregon, AAAI Press.

Agrawal, R. and Psaila, G. (1995). "Active Data Mining". **First International Conference on Knowledge Discovery and Data Mining (KDD-95),** Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 3-8.

Agrawal, R., *et al.* (1995). "Querying shapes of histories". **Twenty-first International Conference on Very Large Databases (VLDB '95),** Zurich, Switzerland, Morgan Kaufmann Publishers, Inc. San Francisco, USA. pp. 490-501.

Agrawal, R. and Srikant, R. (1994). "Fast Algorithms for Mining Association Rules". **Twentieth International Conference on Very Large Data Bases,** Santiago, Chile, pp. 487-499.

Ahonen, H., *et al.* (1993). "Forming grammars for structured documents". **Knowledge Discovery in Databases: 1993 AAAI workshop,** Washington, D.C., AAAI Press.

Al-Naemi, S. (1994). "A Theoretical Framework for Temporal Knowledge Discovery". **International Workshop on Spatio-Temporal Databases,** Benicassim, Spain, pp. 23-33.

Alexander, W. P., *et al.* (1993). "Preliminary Investigations into Knowledge Discovery for Quick Market Intelligence". **AAAI-93 Workshop on Knowledge Discovery in Databases (KDD '93),** Washington D.C., AAAI Press, Menlo Park, California. pp. 52-60.

Allen, J. F. (1983). "Maintaining knowledge about temporal intervals". **Communications of the ACM.** Vol. 26, No. 11, pp. 832-843.

Anand, S. S., *et al.* (1994). "Database mining in the architecture of a semantic pre-processor for state-aware query optimization". **AAAI-94 Workshop on Knowledge Discovery in Databases (KDD '94)**, Seattle,

Anand, S. S., *et al.* (1995a). "Data Mining in Parallel". **WoTUG-18**,

Anand, S. S., *et al.* (1995b). "Evidence Based Discovery of Knowledge in Databases". **IEE Colloquium on Knowledge Discovery in Databases**,

Anand, T. and Kahn, G. (1993). "Opportunity Explorer: Navigating Large Databases Using Knowledge Discovery Templates". **AAAI-93 Workshop on Knowledge Discovery in Databases**, Washington D.C., AAAI press. pp. 45-51.

Angluin, D. and Smith, C. H. (1983). "Inductive inference: theory and methods". **ACM Computing Surveys.** Vol. 15, No. 3, pp. 237-269.

Apté, C., *et al.* (1993). "Knowledge Discovery for Document Classification". **AAAI-93 Workshop on Knowledge Discovery in Databases**, Washington D.C., AAAI press. pp. 326-336.

Asa, J. A. M. and Mangano, J. J. (1995). "Selecting Among Rules Induced From a Hurricane Database". **Journal of Intelligent Information Systems.** Vol. 4, No. 1, pp. 39-52.

Beadle, N. C. W., *et al.* (1986). **Flora of the Sydney Region**. New South Wales, Reed Books.

Bell, D. A., *et al.* (1994). "Data Mining in Spatial Databases". **International Workshop on Spatio-Temporal Databases**, Benicassim, Spain,

Berndt, D. J. and Clifford, J. (1995). "Finding patterns in time series: a dynamic programming approach". in **Advances in Knowledge Discovery and Data Mining**. AAAI Press/ MIT Press. pp. 229-248.

Bhandari, I., *et al.* (1997). "Advanced Scout: Data Mining and Knowledge Discovery in NBA Data". **Data Mining and Knowledge Discovery.** Vol. 1, No. 1, pp. 121-125.

Blum, R. L. (1982). "Discovery, Confirmation, and Incorporation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project". in **Computers and Biomedical Research**. Vol. 15. pp. 164-187.

Bouguettaya, A., *et al.* (1997). "A Comparison of Group-based and Object-based Data Clustering Techniques". **Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases.**, Hong Kong, Springer-Verlag Singapore. pp. 119-136.

Buntine, W. L. and Patel, T. (1995). "Intelligent Instruments: discovering how to turn spectral data into information". **First International Conference on Knowledge Discovery and Data Mining (KDD-95)**, Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 33-38.

Cai, Y., *et al.* (1990). "An attribute-oriented approach for learning classification rules from relational databases". **Sixth IEEE International Conference on Data Engineering**, Los Angeles, CA, IEEE Computer Science Press. pp. 281-288.

Carter, C. and Catlett, J. (1987). "Assessing Credit Card Applications Using Machine Learning". **IEEE Expert.** No. Fall, pp. 71-79.

Chan, K. C. C. and Wong, A. K. C. (1991). "A statistical technique for extracting classificatory knowledge from databases". in **Knowledge Discovery in Databases**. Cambridge, MA, AAAI Press/MIT Press. pp. 107-123, (Ch. 6).

Chen, M.-S., *et al.* (1996). "Data mining: an overview from database perspective". **IEEE Transactions on Knowledge and Data Engineering.** Vol. 8, No. 6.

Chen, M. S., *et al.* (1996). "Data mining for path traversal patterns in a web environment". **Sixteenth International Conference on Distributed Computing Systems**, pp. 385-392.

Chundi, P. and Dayal, U. (1997). "An Application of Adaptive Data Mining: Facilitating Web Information Access". **1997 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery**, Arizona, USA, pp. 31-37.

Ciesielski, V. and Palstra, G. (1996). "Using a Hybrid Neural/Expert System for Data Base Mining in Market Survey Data". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 38-43.

Czyzewski, A. (1996). "Mining Knowledge in Noisy Audio Data". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 220-225.

Davies, W. and Edwards, P. (1995a). "Distributed Learning: An Agent-Based Approach to Data-Mining". **ML95 Workshop on Agents that Learn from Other Agents**,

Davies, W. H. E. and Edwards, P. (1995b). "Agent-Based Knowledge Discovery". **AAAI Spring Symposium On Information Gathering From Heterogeneous, Distributed Environments.**, AAAI Press, menlo Park, CA, U.S.A. pp. 34-37.

Djoko, S., *et al.* (1995). "Analyzing the Benefits of Domain Knowledge in Substructure Discovery". **First International Conference on Knowledge Discovery and Data Mining (KDD-95)**, Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 75-80.

Elmasri, R. and Navathe, S. (1989). **Fundamentals of database systems**. Redwood City, CA, Benjamin/Cummings.

Ester, M., *et al.* (1995). "Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification". **Fourth International Symposium on Large Spatial Databases**, Maine,

Fayyad, U., M., *et al.* (1996a). "From Data Mining to Knowledge Discovery: An Overview". in **Advances in Knowledge Discovery and Data Mining**. AAAI Press/ MIT Press. pp. 1-34.

Fayyad, U., M., *et al.*, Ed. (1996b). **Advances in Knowledge Discovery and Data Mining**. Menlo Park, California, AAAI Press.

Fayyad, U. M., *et al.* (1993). "Automated Analysis of a Large-Scale Sky Survey: The SKICAT System". **AAAI-93 Workshop on Knowledge Discovery in Databases (KDD '93)**, Washington D.C., AAAI Press, Menlo Park, California. pp. 1-13.

Feelders, A. J., *et al.* (1995). "Data Mining for Loan Evaluation at ABN AMRO: A Case Study". **First International Conference on Knowledge Discovery and Data Mining (KDD-95)**, Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 106-111.

Feldman, R. and Dagan, I. (1995). "Knowledge Discovery in Textual Databases (KDT)". **First International Conference on Knowledge Discovery and Data Mining (KDD-95)**, Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 112-117.

Feldman, R. and Hirsh, H. (1996). "Mining Associations in Text in the Presence of Background Knowledge". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 343-346.

Fisher, D. (1995). "Optimization and simplification of hierarchical clusterings". **First International Conference on Knowledge Discovery and Data Mining (KDD-95)**, Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 118-123.

Fong, J. and Zeng, X. (1997). "Data Warehouse for Decision Support". **Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases.**, Hong Kong, Springer-Verlag Singapore. pp. 195-207.

Frawley, W. J., *et al.* (1991). "Knowledge discovery in databases: an overview". in **Knowledge Discovery in Databases**. Menlo Park, California, AAAI Press. pp. 1-27.

Freksa, C. (1992). "Temporal reasoning based on semi-intervals". **Artificial Intelligence.** Vol. 54, pp. 199-227.

Guan, J. and Bell, D. (1991). **Evidence theory and its applications**. North-Holland.

Guan, J. and Bell, D. (1992). **Evidence theory and its applications**. North-Holland.

Han, J., *et al.* (1993). "Data-Driven Discovery of Quantitative Rules in Relational Databases". **IEEE Transactions on Knowledge and Data Engineering.** Vol. 5, No. 1, pp. 29-40.

Han, J. and Fu, Y. (1994). "Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases". **AAAI'94 Workshop on Knowledge Discovery in Databases**, Seattle, WA, pp. 157-168.

Han, J. and Fu, Y. (1995). **Discovery of Multiple-Level Association Rules from Large Databases**. CMPT TR 95-05. Simon Fraser University.

Han, J., *et al.* (1996). "DBMiner: A System for Mining Knowledge in Large Relational Databases". **Second International Conference on Knowledge Discovery and Data Mining**, Portland, Oregon., pp. 250-255.

Hébrail, G. and Marsais, J. (1992). "Experiments of Textual Data Analysis at Electricite de France". **International Federation of Classification Societies Conference**,

Hofacker, I. L., *et al.* (1996). "Knowledge Discovery in RNA Sequence Families of HIV Using Scalable Computers". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 20-25.

Hoschka, P. and Klösgen, W. (1991). "A support system for interpreting statistical data". in **Knowledge Discovery in Databases**. Cambridge, MA, AAAI Press/MIT Press. pp. 325-345, (Ch. 19).

Inmon, W. H. (1996). "The data warehouse and data mining". **Communications of the ACM.** Vol. 39, No. 11, pp. 49-50.

Jackson, S. (1986). **Introduction to Expert Systems**. Addison-Wesley Publishers Limited.

Kamber, M., *et al.* (1997). **Using Data Cubes for Metarule-Guided Mining of Multi-dimensional Association Rules**. Technical Report Report CMPT-TR-97-10. School of Computing science, Simon Fraser University.

Kamel, M. N. (1995). "A Prototype Rule-Based Front End Expert System for Integrity Enforcement in Relational Data Bases: An Application to the Naval Aircraft Flight Records Data Base". **Expert Systems With Applications.** Vol. 8, No. 1, pp. 47-58.

Keats, G. and Loo, S. (1997). "An intelligent business workbench for the insurance industry: using data mining to improve decision making and performance.". **Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases.**, Hong Kong, Springer-Verlag Singapore. pp. 256-274.

Keogh, E. and Smyth, P. (1997). "A probabilistic approach to fast pattern matching in time series databases". **Third International Conference on Knowledge Discovery and Data Mining,** Newport Beach, California, AAAI Press, Menlo Park, California. pp. 24-30.

Ketterlin, A. (1997). "Clustering Sequences of Complex Objects". **Third International Conference on Knowledge Discovery and Data Mining,** Newport Beach, California, AAAI Press, Menlo Park, California. pp. 215-218.

Klösgen, W. (1993). "Some implementation aspects of a discovery system". **AAAI-93 workshop on Knowledge Discovery in Databases,** Washington D.C., AAAI Press. pp. 212-226.

Klösgen, W. (1995a). "Anonymization Techniques for Knowledge Discovery in Databases". **First International Conference on Knowledge Discovery and Data Mining (KDD-95),** Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 186-191.

Klösgen, W. (1995b). "Efficient Discovery of Interesting Statements in Databases". **Journal of Intelligent Information Systems.** No. 4, pp. 53-69.

Koch, T. and Fehsenfeld, B. (1995). "Discovering Expert System Rules in Data Sets". **Expert Systems With Applications.** Vol. 8, No. 2.

Koperski, K. and Han, J. (1995). "Discovery of Spatial Association Rules in Geographic Information Databases". **Fourth International Symposium on Large Spatial Databases,** Maine, pp. 47-66.

Lee, R. M., *et al.* (1985). "Temporal inferencing on administrative databases". **Information Systems.** Vol. 10, No. 2, pp. 197-206.

Li, S.-H., *et al.* (1997). "Discovering missing semantics from existing relational databases". **Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases.**, Hong Kong, Springer-Verlag Singapore. pp. 275-286.

Loeffen, A. (1994). "Text Databases: A Survey of Text Models and Systems". **SIGMOD Record.** Vol. 23, No. 1.

Long, J. M., *et al.* (1991). "Automating the Discovery of Causal Relationships in a Medical Records Database". in **Knowledge discovery in databases.** AAAI Press/MIT Press. pp. 465-476.

Lu, W., *et al.* (1993). "Discovery of General Knowledge in Large Spatial Databases". **1993 Far East Workshop on GIS (IEGIS 93),** Singapore, pp. 275-289.

Maier, D. and Stein, J. (1990). "Development and Implementation of an Object-Oriented DBMS". in **Readings in Object-Oriented Database Systems.** Morgan Kaufmann Publishers Inc.

Mannila, H., *et al.* (1994). "Efficient Algorithms for Discovering Association Rules". **AAAI Workshop on Knowledge Discovery in Databases,** Seattle, Washington, pp. 181-192.

Mannila, H., *et al.* (1993). **Improved Methods for Finding Association Rules.** Technical Report Report C-1993-65. University of Helsinki, Finland.

Mannila, H., *et al.* (1995). "Discovering frequent episodes in sequences". **First International Conference on Knowledge Discovery and Data Mining (KDD-95),** Montreal, Quebec, Canada, AAAI Press, Menlo Park, California. pp. 210-215.

Meo, R., *et al.* (1996). "A New SQL-Like Operator for Mining Association Rules". **Twenty-second International Conference on Very Large Data Bases (VLDB '96),** Mumbai, India, Morgan Kaufmann Publishers, Inc. San Francisco, USA. pp. 122-133.

Michalski, R. S., *et al.*, Ed. (1984). **Machine learning - an artificial intelligence approach.** Berlin, Springer-Verlag.

Miller, M. (1991). "A Model of Statistical Database Compromise Incorporating Supplementary Knowledge". **the second Australian Database-Information Systems Conference,** Sydney, Australia, World Scientific.

Miura, T. and Shioya, I. (1997). "Differentiation for scheme discovery". **Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases.**, Hong Kong, Springer-Verlag Singapore. pp. 62-77.

Nahouraii, E. and Petry, F., Ed. (1991). **Object-Oriented Databases.** Los Alamitos, California, IEEE Computer Society Press.

Ng, R. T. and Han, J. (1994). "Efficient and effective clustering methods for spatial data mining". **Twentieth International Conference on Very Large Data Bases,** Santiago, Chile,

Nishio, S., *et al.* (1993). "Knowledge Discovery in Object-Oriented Databases: The First Step". **AAAI-93 workshop on Knowledge Discovery in Databases,** Washington D.C., AAAI Press. pp. 299-313.

O'Leary, D. E. (1991). "Knowledge Discovery as a threat to Database security". in **Knowledge Discovery in Databases.** AAAI Press/MIT Press. pp. 507-516.

Padmanabhan, B. and Tuzhilin, A. (1996). "Pattern discovery in temporal databases: a temporal logic approach". **Second International Conference on Knowledge Discovery and Data Mining**, Portland, Oregon, AAAI Press.

Park, J. S., *et al.* (1995). "An Effective Hash-Based Algorithm for Mining Association Rules". **ACM SIGMOD Conference on the Management of Data**, San Jose, CA, ACM Press. Vol. 24. pp. 175-186.

Piatetsky-Shapiro, G. (1994). "Knowledge Discovery in Databases: Progress Report". **The Knowledge Engineering Review.** Vol. 9, No. 1.

Piatetsky-Shapiro, G. and Frawley, W. J., Ed. (1991). **Knowledge discovery in databases.** Menlo Park, CA, AAAI Press/MIT Press.

Provan, G. M. and Singh, M. (1996). "Data Mining and Model Simplicity: A Case Study in Diagnosis". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California.

Quinlan, J. R. (1986). "Induction of Decision Trees". **Machine Learning.** Vol. 1, pp. 81-106.

Rainsford, C. P. and Roddick, J. F. (1996). "Temporal data mining in information systems: a model". **Seventh Australasian Conference on Information Systems**, Hobart, Tasmania, Vol. 2. pp. 545-553.

Rainsford, C. P. and Roddick, J. F. (1997). "An attribute-oriented induction of rules from temporal interval data". **Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases (IDW'97)**, Hong Kong, Springer Verlag. pp. 108-118.

Roddick, J. F. (1994). **A model for temporal inductive inference and schema evolution in relational database systems.** Ph.D. Thesis. Department of Computer Science and Computer Engineering, La Trobe University.

Roddick, J. F. (1995). "A survey of schema versioning issues for database systems". **Information and Software Technology.** Vol. 37, No. 7, pp. 383-393.

Roddick, J. F., *et al.* (1996). "Handling discovered structure in database systems". **IEEE Transactions on Knowledge and Data Engineering.** Vol. 8, No. 2 (April), pp. 227-240.

Roddick, J. F. and Patrick, J. D. (1992). "Temporal semantics in information systems - a survey". Information Systems. Vol. 17, No. 3, pp. 249-267.

Roddick, J. F. and Rice, S. (1998). "Towards induction in databases". **Ninth Australasian Information Systems Conference**, University of NSW, Vol. 2. pp. 534-542.

Roddick, J. F. and Spiliopoulou, M. (1999). "A bibliography of temporal, spatial and spatio-temporal data mining research". **SIGKDD Discoveries.** Vol. 1, No. 1, In Press.

Rubinstein, Y. D. and Hastie, T. (1997). "Discriminative vs Informative Learning". **Third International Conference on Knowledge Discovery and Data Mining**, Newport Beach, California, AAAI Press, Menlo Park, California. pp. 49-53.

Savasere, A., *et al.* (1995). "An Efficient Algorithm for Mining Association Rules in Large Databases". **Twenty-first International Conference on Very Large Data Bases**, Zurich, Switzerland, pp. 432-444.

Savnik, I. and Flach, P. A. (1993). "Bottom-up Induction of Functional Dependencies from Relations". **AAAI-93 Workshop on Knowledge Discovery in Databases**, Washington D.C., AAAI Press. pp. 174-185.

Saxena, S. (1993). "Fault Isolation during Semiconductor Manufacturing using Automated Discovery from Wafer Tracking Databases". **AAAI-93 Workshop on Knowledge Discovery in Databases (KDD '93)**, Washington D.C., AAAI Press, Menlo Park, California. pp. 81-88.

Schaffer, C. (1991). "On evaluation of domain-independent scientific function-finding systems". in **Knowledge Discovery in Databases.** Cambridge, MA, AAAI Press/MIT Press. pp. 93-104, (Ch. 5).

Schlimmer, J. C. (1993). "Self Modelling Databases : Learning and Applying Partial Integrity Constraints to Detect Database Errors". **IEEE Expert.** No. April.

Schlimmer, J. C., *et al.* (1991). "Justification-Based Refinement of Expert Knowledge". in **Knowledge Discovery in Databases.** AAAI Press/MIT Press. pp. 397-410.

Shortland, R. and Scarfe, R. (1994). "Data mining applications in BT". **BT Technology Journal.** Vol. 12, No. 4, pp. 17-22.

Siegal, M. B., *et al.* (1991). "Rule discovery for query optimization". in **Knowledge Discovery in Databases.** Cambridge, MA, AAAI Press/MIT Press. pp. 411-427, (Ch. 24).

Siegel, M., *et al.* (1992). "A Method for Automatic Rule Derivation to Support Semantic Query Optimization". **ACM Transactions on Database Systems.** Vol. 17, No. 4, pp. 563-600.

Silberschatz, A., *et al.* (1996). "Database Achievements and Opportunities into the 21st Century". **SIGMOD Record.** Vol. 25, No. 1, pp. 52-63.

Silberschatz, A. and Tuzhilin, A. (1996). "What Makes Patterns Interesting in Knowledge Discovery Systems". **IEEE Transactions on Knowledge and Data Engineering.** Vol. 8, No. 6, pp. 970-974.

Smyth, P. (1996). "Clustering using Monte Carlo Cross-Validation". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 126-133.

Srikant, R. and Agrawal, R. (1995). "Mining Generalized Association Rules". **Twenty-first International Conference on Very Large Databases**, Zurich, Switzerland,

Srikant, R. and Agrawal, R. (1996). "Mining Quantitative Association Rules in Large Relational Tables". **ACM SIGMOD Conference on the Management of Data**, Montreal, Canada,

Stolorz, P. and Dean, C. (1996). "Quakefinder: A Scalable Data Mining System for Detecting Earthquakes from Space". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 208-213.

Stonebraker, M., *et al.* (1993). "DBMS research at the crossroads: the Vienna update". **Nineteenth International Conference on Very Large Databases**, Dublin, Ireland, Morgan Kaufmann, Palo Alto, CA. pp. 688-692.

Tansel, A. U., *et al.* (1993). **Temporal databases: theory, design and implementation**. Redwood City, CA, Benjamin Cummings.

Tsumoto, S. and Tanaka, H. (1996). "Automated Discovery of Medical Expert System Rules from Clinical Databases Based on Rough Sets". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 63-69.

Valiant, L. G. (1984). "A Theory of the Learnable". **Communications of the ACM.** Vol. 27, No. 11, pp. 1134-1142.

Viveros, M. S., *et al.* (1996). "Applying Data Mining Techniques to a Health Insurance Information System". **Twenty-second International Conference on Very Large Data Bases (VLDB '96)**, Mumbai, India, Morgan Kaufmann Publishers, Inc. San Francisco, USA. pp. 286-293.

Wade, T. D., *et al.* (1994). "Finding temporal patterns - a set based approach". **Artificial Intelligence in Medicine.** No. 6, pp. 263-271.

Wang, J. T.-L., *et al.* (1994). "Combinatorial Pattern Discovery for Scientific Data: Some Preliminary Results". **ACM SIGMOD International Conference on Management of Data**, Minneapolis, Minnesota, pp. 115-125.

Wang, J. T. L., *et al.* (1996). "Automated Discovery of Active Motifs in Multiple RNA Secondary Structures". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 70-75.

Wirth, R. and Reinartz, T. P. (1996). "Detecting Early Indicator Cars in an Automotive Database: A Multi-Strategy Approach". **Second International Conference on Knowledge Discovery and Data Mining (KDD96)**, Portland, Oregon, AAAI Press, Menlo Park, California. pp. 76-81.

Yu, C. T. and Sun, W. (1989). "Automatic knowledge acquisition and maintenance for semantics query optimisation". **IEEE Transactions on Knowledge and Data Engineering.** Vol. 1, No. 3, pp. 362-375.

Zamir, O., *et al.* (1997). "Fast and Intuitive Clustering of Web Documents". **Third International Conference on Knowledge Discovery and Data Mining**, Newport Beach, California, AAAI Press, Menlo Park, California. pp. 287-290.

Ziarko, W., *et al.* (1993). "An Application of Datalogic/R Knowledge Discovery Tool to Identify Strong Predictive Rules in Stock Market Data". **AAAI-93 Workshop on Knowledge Discovery in Databases (KDD '93)**, Washington D.C., AAAI Press, Menlo Park, California. pp. 89-101.