Data Mining: Delving into the Unknown

S. S. Anand, A. G. Büchner, J. G. Hughes
Northern Ireland Knowledge Engineering Laboratory

This chapter provides an introduction to the field of Data Mining (also known as Knowledge Discovery in Databases, Intelligent Data Analysis, Data Dredging and Data Archaeology). Firstly, we define Data Mining presenting our view on what its aims and challenges are. We then provide some background to Data Mining by providing a historical view on the evolution of electronically stored data. This leads into a discussion on why there was a need for Data Mining in the industry, which has been a major factor in the efforts that have gone into building the present generation of Data Mining systems. We present a number of areas in IT that are related to Data Mining in their objectives and compare and contrast these technologies with Data Mining. Next we describe different Data Mining goals identified within the literature before discussing the technologies that go towards achieving these goals. We discuss the knowledge representation models commonly used in Data Mining and the various discovery paradigms employed before describing its enabling technologies - these are the IT foundations on which successful Data Mining can be achieved. We then discuss the Data Mining process (also referred to as the Knowledge Discovery in Databases (KDD) process) and describe the various stages within the process discussing technological support required at each of the stages. We present various applications of Data Mining in manufacturing, finance, retail, medicine and science. We conclude the chapter by describing the organisation of this book and providing a comprehensive list of references and a bibliography for Data Mining.

1.1 Data Mining: Definition, Aims and Challenges

Over the past two decades there has been a huge increase in the amount of data being stored in databases as well as the number of database applications in business and the scientific domain. This explosion in the amount of electronically stored data was accelerated by the success of the relational model for storing data and the development and maturing of data retrieval and manipulation technologies. While technology for storing the data developed quickly to keep up with the demand, less emphasis was placed on developing software for analysing the data, until recently when companies realised that hidden within this mass of stored data was a resource that has been largely ignored. This huge amount of stored data contained knowledge about a number of aspects of their business, which was not being utilised. The Database Management Systems used to manage these data sets at present only allow the user to access information explicitly present in the databases i.e. the data. The data stored in the database is only the tip of the 'iceberg of information' available from it. Contained implicitly within this data is knowledge about aspects of business operations waiting to be harnessed and used for more effective business decision support.

The extraction of knowledge from large data sets is called Data Mining and is defined as the efficient extraction of non-trivial, implicit, previously unknown, potentially useful and understandable information from large data sets. Data Mining has a number of synonyms, the most frequently used ones are Knowledge Discovery in Databases (KDD), Intelligent Data Analysis, Data Archaeology and Data Dredging. Data Mining has been defined in literature as a stage in the KDD process [FAYY96, KLOE96] where automated algorithms for discovering patterns in the data are let loose on the pre-processed data. However, we use Data Mining as a synonym for KDD, mainly due to the fact that it is the prevalent term in industry.

The aim of Data Mining is to improve the performance of a knowledge intensive process by providing timely and useful knowledge in a concise manner from large, relevant data resources. The process, improved performance in which is desired, could be in manufacturing, finance, medicine, retail and even a scientific endeavour. The need for information as well as its source is, however, identical across all these sectors.

Data Mining can be considered to be an inter-disciplinary field involving concepts from Machine Learning, Database Technology, Statistics, Mathematics, High Performance Computing and Visualisation (see Figure 1. 1). While the main concern of database technologists was to find efficient ways of storing, retrieving and manipulating data, the main concern of the machine learning and statistical community was to develop techniques for learning knowledge from data and that of the Visualisation community has been on the interface between humans and their data stored electronically. The complexity of the mining algorithms and the size of the data being mined, make High Performance Computing an essential ingredient of successful, time-critical, Data Mining.

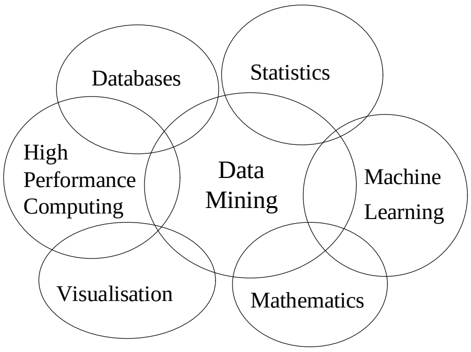

Figure 1. 1: Data Mining

The obvious benefits of Data Mining have resulted in a lot of resources being directed towards its development. Most major software and hardware vendors are involved in the development of such “futuristic” decision support tools which are now becoming a reality.

A number of challenges face Data Miners that need to be addressed for successful Data Mining. These challenges are either Data-based, i.e. caused by the state of the data that is being mined, Functionality-based, i.e. caused by the functionality required from a Data Mining system to be useful to the user, Performance-based, i.e. caused by efficiency and time critical constraints on the system, or Ethical-based, i.e. caused by ethical issues on the use of Data Mining. Real-world data is noisy, typically contain a lot of missing values, often contains residual variation (i.e. incomplete data in the sense of missing attributes), is stale (i.e. out of date), distributed and heterogeneous, stored in legacy systems, constantly being updated and enormous, both in the number of tuples as well as its dimensionality. The functionality required from a Data Mining system includes knowledge maintenance, knowledge manipulation techniques, techniques for incorporating user biases and domain knowledge, knowledge filtering and validation, knowledge visualisation and understandability and techniques for integration with existing systems. Performance-based challenges centre around efficiency and scalability of the system and its ability to provide the required knowledge in a timely fashion. Timeliness of knowledge is essential as any system that provides useful knowledge but allows the train of thought of the decision maker to be broken cannot be effective. Finally, protection and privacy issues associated with the knowledge discovered by the Data Mining system have not been addressed as such, and though it is not necessarily a technological challenge, addressing it is essential for the success of Data Mining.

1.2 Evolution of Data Management

Data management started about three decades ago when data was stored in flat ASCII or EBCDIC operating system files. At this stage no data specific information was explicitly stored along with the data. Often data had to be stored more than once across the organisation leading to inconsistencies and inefficiencies. There were no query languages, and any constraints or interrelationships among the entities were left to applications accessing the files. Database Management Systems were introduced in the late 1960’s largely triggered by the Space Race. They provided these missing facilities and were based on networks, hierarchies or (later) relations. Constraints, such as data types, value ranges, dependencies, or relationships among entities were provided and meta data stored in data dictionaries. Later still, fourth generation languages were provided to ease application development. Within the last decade these systems have been extended to handle distributed and heterogeneous data, and thus more semantics about the inter-relationships of different sites had to be embedded. Richer modelling methods such as the Semantic data model and Object-oriented data models [HUGH91] have now been developed and are making their mark in the industry.

State-of-the-art data management systems are platform-independent client-server solutions with active back-ends and visualised front-ends. In addition to user friendly manual data entery, these components allow automatic data generation, which leads to an enormous data volume increase (as [FRAW91] stated “the amount of data doubles every 20 months”). The information hidden in both, the manually entered and the automatically generated data, is tremendous.

So, the question arises - what have we been doing with this data to date and what will we do with it in the future? The data flood will not stop, but we can possibly slow down the increase in storage capacity needed by using knowledge based pre-processing techniques. That is, we can filter out the useful information within the data and only store it as opposed to storing the large data volumes that are inaccessible and, consequently, useless due to their volume. While this knowledge-based pre-processing solves the problem of the data flood, it raises new issues:

- How do we ensure that the knowledge is correct?
- How can we manage knowledge in a similar way as we handle data?
- What are the privacy issues involved?
- How can we be sure that all the useful knowledge has been extracted from the data?

These questions need answered before such knowledge based filtering techniques can be utilised to their full potential.

IT investments in the industry have to date been aimed at automation of business processes rather than at proactive decision support. However, a change in the trend of IT investments from process automators to “informators” has been taking place and it has become clear that systems adept in automating business processes are wholly inappropriate for decision support. Technologies that played a central role in the automation of businesses processes, for example, database management systems used for implementing On-line Transaction Processing (OLTP) applications, falter when required to transform data stored in OLTP applications into useful information and also for getting “richer and deeper” information. There are a number of specific reasons why RDBMS are not suitable for business analysis, however, the underlying root of the problem is the inherent nature of the two application areas being at odds with each other.

To make OLTP queries fast, RDBMS applications generally normalise the data into 50 - 200 tables. The normalised database reduces overheads for an update transaction and improves the performance of retrieval for queries with only a small subset of resulting tuples. Though efficient for OLTP operations, normalised databases are inefficient for ad-hoc business analysis applications as it means a large number of joins are required to access the data necessary for such applications. While parallel processing can be useful in table scans it offers very little performance enhancement for complex joins.

Also, to allow truly ad-hoc end-user analysis, the database administrator should, ideally, index the database on every possible combination of columns and tables that the end user may ask for. This would create an unnecessary overhead for OLTP and query response times.

Locking models, data consistency schemes, and caching algorithms are based on the RDBMS being used for OLTP applications where the transactions are small and discrete. Long running, complex queries cause problems in each of these areas.

Furthermore, SQL is not designed with common business needs in mind. There is no way of using standard SQL for retrieving information like “the top 10 salespersons”, “bottom 20% of customers”, “products with a market share of greater than 25%” or “the sales ratio of cola to root beer”. Also RDBMS do not provide common data analysis tools like data rotation, drill downs, dicing and slicing. Such functionality was never meant to be part of an RDBMS, rather it was left to client software that has failed to deliver in the last two decades.

Thus, the data management infrastructure within most organisations is shaky and requires an overhaul for Data Mining to be carried out successfully. The current “hot topic” in data management, Data Warehousing, aims at doing just that. Data Warehousing is a technique for integrating “legacy” operational systems within a corporation to provide an enterprise wide view for decision support purposes. This technology has become necessary due to the realisation on the part of large organisations that decisions about one business process cannot be made in complete isolation of other business processes within the enterprise. For example, large financial organisations may have different sections of their marketing departments maintaining their own customer data based around different products. The individual product-centred customer databases often do not link in together. While such a situation may be fine from a day-today operational perspective, clearly, from a decision support perspective a much more beneficial situation would be a customer-centred database where the same identification number is used to identify all the different products bought by each customer.

Also, most large corporations have operational data in production systems that is unreliable and disparate, making it difficult to integrate or extract for analysis purposes. Thus, the implementation of a Data Warehouse consists of the acquisition of data from multiple internal and external sources (of the corporation), the management and integration into a central, integrated repository, the provision of access, reporting and analysis tools to interpret selected data converting it into information to support managerial decision making processes.

However, in a recent survey of Data Warehousing projects [CONS96] it was reported that while most warehousing projects meet the users requirements of data quality and integration the ability to analyse and convert the data into information has been a major disappointment. Data Mining and other data analysis tools attempt to provide this ability to convert data into information for the Data Warehouse user. According to Charles Bonomo, vice-president of Advanced Technologies at J. P. Morgan, a large financial organisation in the US, “One of the primary justifications for implementing a data warehousing solution is having a Data Mining tool in place that can access the data within it”. The most logical consequence is to integrate Data Mining technology as central part of the data warehousing philosophy, without limiting any of the existing functionality.

1.3 Related Areas

In this section we discuss four disciplines that are related to Data Mining. These are On-line Analytical Processing (OLAP), Statistical Data Analysis, Machine Learning and Database Query and Reporting tools. We compare and contrast these areas with Data Mining in this section.

1.3.1 On-Line Analytical Processing

Complex statistical functionality was never intended to be accommodated within RDBMSs. Providing such functionality was left to user-friendly end-user products such as spreadsheets or statistical packages which act as front ends to the RDBMS. Though statistics packages and related tools provide a certain amount of functionality required by business analysts, none address, to any great extent, the need for analysing the data according to its multiple dimensions. Any product that intends to provide such functionality must provide the following features to allow adequate statistical data analysis - access to many different types of files, creation of multi-dimensional views of the data, experimentation with various data formats and aggregations, definition and visual animation of new information models, application of summations and other formulae to these models, data analysis tools such as drilling down, rolling up, slicing and dicing, rotation of consolidation paths and generation of a wide variety of reports, charts and diagrams.

On-line Analytical Processing (OLAP) is the name given by E. F. Codd [CODD93] to the technologies that attempt to address these user requirements. Codd defined OLAP as “the dynamic synthesis, analysis and consolidation of large volumes of multi-dimensional data”.

Codd provided 12 rules/ requirements of any OLAP system. These are support for multidimensional conceptual views, transparency, accessibility, consistent reporting performance, client-server architecture, generic dimensionality, dynamic sparse matrix handling, multi-user support, unrestricted cross-dimensional operations, intuitive data manipulation, flexible reporting and unlimited dimensions and aggregation levels. These rules are based around one principal objective - to provide an environment that the data analyst can use without needing to compromise the quality of the analysis being undertaken. The requirement for the OLAP system supporting multidimensional views is based on the fact that the most intuitive way of visualising data is in a multidimensional manner as opposed to the flat, two dimensional relational view provided by relational databases. Thus, supporting such a view allows the analyst to undertake the analysis without the need to constantly map his/ her multidimensional requirements onto two dimensions. Truly multidimensional OLAP systems utilise multidimensional database servers as opposed to relational database servers. Similarly, unlimited cross-dimensional operations, consistent reporting performance as the number of dimensions increase, accessibility, generic and unlimited dimensions, intuitive data manipulation, flexible reporting and unlimited generalisations and aggregation levels are all requirements so as to allow the analyst to focus on the analysis task as opposed to trying to reduce the impact of a compromise on the benefits of the analysis.

A number of extensions to these OLAP requirements have been suggested [DRES93, BUYT95], including: Support for multiple arrays, time series analysis, OLAP joins, procedural language and development tools, database management tools, object storage, integration of functionality, subset selection, detail drill down, local data support, incremental database refresh and an SQL interface.

While OLAP is a useful data analysis tool, its objectives and goals differ from that of Data Mining. OLAP takes analysts needs into account and provides an integrated environment for data analysis with fast access to multidimensional data as well as exploratory analysis tools such as slice and dice, rotation of consolidation paths and drill down. However, the analysis is carried out manually, that is, it is user driven with fixed, pre-determined dimensions, ruling out novel, previously unknown discoveries being made.

1.3.2 Statistical Data Analysis

Statistical data analysis has proved useful in disproving hypotheses formed by domain experts in fields as varied as the statistical methods themselves. In general, statistical methods can be classified into descriptive and inferential statistics. Descriptive statistics consists of techniques for describing the data set in a concise form providing valuable information such as means, standard deviation, distributions as well as exploratory data analysis and simple graphical techniques. Inferential statistics is mainly concerned with generalisation, analysis, interpretations, and predictions of the described observations.

Statistics falls short of the goals of Data Mining. Firstly, Statistics is ill-suited for nominal and structured data types that are common in real-world databases. Secondly, Statistics is totally data driven and does not provide techniques for incorporating domain or prior knowledge. Thirdly, the process of statistical data analysis requires expert user guidance. Lastly, the results from a statistical analysis are difficult to interpret and are overwhelming to non-statisticians.

Previous versions of statistical packages had poor interfaces to data and provided the user with poor portability of results, reducing their uptake when analysing large operational databases as is required in Data Mining. Modern statistical packages extend the providence of tools and techniques by providing loosely-coupled modules lending themselves to Data Mining. Examples are open database interfaces, parallelism support, or the embodiment of basic machine learning techniques e.g. SPSS Neural Connection. However, the future use of statistical methods in Data Mining certainly lies in closely-coupling their use with machine learning and database technologies in the form of exploratory data analysis, noise modelling, knowledge validation and significance testing.

1.3.3 Machine Learning

Learning from data has been an area of interest to machine learning enthusiasts since the 1970s. Over the past two decades machine learning research has matured with the development of a number of sophisticated techniques based on different models of human learning and reasoning. Learning by example, Cased-based Reasoning, Learning by observation, Neural Networks, Genetic Algorithms and Bayesian Belief Networks are some of the most popular learning techniques that were being used to create “the ultimate thinking machine”.

Machine Learning researchers tend to identify themselves with one of five main paradigms of machine learning [LANG95]. These are Rule Induction, Genetic Algorithms, Neural Networks, Instance-based or Case-based Reasoning and Analytic Learning. Of these, the first four paradigms have been applied to Data Mining with varying degrees of success. Langley et al. [LANG95] suggest that the distinctions made between these paradigms are more from a historical rather than a scientific viewpoint. While all these paradigms aim to achieve the same goal of improved performance in solving a task by exploiting regularities in data, they differ in the metaphors used. For example, while proponents of Neural Networks emphasise on analogies with neuroscience, proponents of Genetic Algorithms draw parallels with evolution, Case-based Reasoning with human memory and Rule Induction with heuristic search. However, these differences are now less pronounced and hybrid systems that utilise more than one of these paradigm together are becoming commonplace, for example, Neuro-Fuzzy and Neuro-Genetic algorithms.

The main factor that distinguishes Data Mining from machine learning is that it is concerned with learning from existing real-world data rather than data generated particularly for the learning tasks. In Data Mining the data sets are large therefore efficiency and scalability of algorithms is important. As mentioned earlier the data from which Data Mining algorithms learn knowledge is already existing real-world data. Therefore, typically the data contains plenty of missing values as well as noise and it is not static i.e. it is prone to updates. However, as the data is stored in databases, efficient methods for data retrieval are available that can be used to make the algorithms more efficient. Also, domain knowledge in the form of integrity constraints is available that can be used to constrain the learning algorithms search space.

In summary, machine learning algorithms form the basis for most Data Mining tools. However, to make them suitable for handling real-world Data Mining problems appropriate extensions have to be added to these techniques.

1.3.4 Database Query and Reporting Tools

Database reporting tools were originally designed to create reports derived from databases and data files. With increasing user requirements, those tools have been extended to provide basic statistical analysis, primitive visualisation of data through charts, and (mostly) proprietary macro languages.

In Section 0 we described how the relational model defines an unrealistic structure on the data that is split across a number of two-dimensional tables instead of being in a more intuitive multidimensional form. Thus, even though SQL provides a non-procedural approach to querying data, allowing the user to concentrate on the description of what data he/she requires as opposed to the actual procedure that needs to be followed to retrieve the data, due to the unintuitive two-dimensional representation of the underlying data, analysts find it a daunting task to use SQL for data analysis. Also, SQL lacks a number of features required for analysis (see section 0).

Recently, however, more and more OLAP functionality has been embedded in up-to-date database reporting tools, such as drill-down and roll-up operations, sub-reports for multidimensional views, or cross-dimensional operations.

The goal of database reporting tools is to allow the user to generate reports and simple summaries of data, which are useful for retrospective analysis. Therefore, the user must have a good idea of what data he/ she wants to report on while using these tools. On the other hand, Data Mining tools are used in situations where the user is not quite sure what he/she is looking for. Therefore, typically, Data Mining queries are "fuzzy" and less precise. For example, a Data Mining query could be "What are the characteristics of my customers that tend to lapse their motor insurance policy?". To date, no query and reporting tool can handle such a query.

1.4 Data Mining Goals

In this section we describe the different goals of Data Mining in terms of the type of problem it is being used to solve. We define nine different goals: Classification, Cluster Analysis or Data Segmentation, Regression, Temporal Modelling, Discovery of Associations, Sequential Pattern Discovery, Discovery of Characteristics, Dependency Modelling and Deviation Detection. Each of these goals require different types of data as well as different learning paradigms. We briefly describe these goals and define their data requirements. The learning paradigms are discussed in Section 0.

1.4.1 Classification

Classification rules are rules that discriminate between different partitions of a database based on their attributes. The partitions of the database are based on an attribute called the classification label. Each value within the classification label domain is called a class. Consider the sample data from a car insurance company shown in Table 1. The field "Lapse/ Renew" partitions the database into two classes, that of customers who lapsed their insurance policy ("Lapse Customers") and customers who renewed their insurance policy ("Renew Customers"). The insurance company would clearly benefit from being able to predict in advance as to whether a customer was going to lapse or renew his/ her policy as they could pursue customers at a high risk of lapsing their policy to try and change their mind. Classification rules can be discovered to discriminate between “Lapse Customers” and “Renew Customers”.

Table 1: Sample Data from Insurance Company

Cust No.	Insurance Group	Age of Car	Age of Driver	No Claims	Lapse/ Renew
10011	4	10	21	0	Lapse
10012	5	7	24	1	Lapse
10013	8	3	26	1	Lapse
00212	2	16	56	8	Renew
00131	9	2	62	10	Renew

Learning of classification rules can be classified into Single Class Learning and Multiple Class Learning. In *Single Class Learning* the data is required on a single class in the form of examples of states belonging to that class (*positive examples*) as well as examples of states that do not belong to that class (*negative examples*). In *Multiple Class Learning* the data consists of examples that belong to a number of mutually exclusive classes. The classification algorithm constructs a class description for each of the classes that distinguish states belonging to one class from those belonging to another. In this case, examples of states belonging to the class are the positive examples while examples of states of all the other classes form the negative examples for that class.

Apart from the availability of positive and negative examples in the data set, the availability of the classification label is assumed by the classification algorithms. As these learning methods assume the classification label to be available in the data set, they are often referred to as *Supervised Learning* in machine learning literature. Data used for building the classification model is called the *training data set*. Algorithms assume that the training data set is a representative sample of the underlying process that generates the data. More on this aspect of learning is covered in Chapter 23.

Neural Networks (see Chapter 8) [BIGU96], Tree Induction (see Chapter 5) [AGRA92], Genetic Algorithms, Simulated Annealing (see Chapter 7) and Statistical techniques (see Chapter 19) [CHAN91] have been employed for achieving the Classification goal of Data Mining.

1.4.2 Cluster Analysis

Cluster Analysis or Data Segmentation, often referred to in Machine Learning literature as *Unsupervised Learning,* is concerned with discovering structure in data. This goal is also known as *learning by observation and discovery*. Initial clustering techniques were based on the Euclidean distance between the data tuples and algorithms developed for clustering attempted to maximise the similarity within a class while minimising the similarity between the different classes. However, these systems could only deal with numeric data and were unable to use background information. Conceptual clustering attempts to overcome the drawbacks of traditional clustering techniques by employing not only the Euclidean distance measures for numeric values but also employing hierarchical concept generalisations made available by domain experts to deal with non-numeric attributes within the data.

Cluster Analysis differs from classification in that the classes, which the data tuples in the training set belong to, are not provided. The clustering algorithm has to identify the classes by finding similarities between different states provided as examples. A classification algorithm may then be used to discover the distinguishing features of these discovered classes within the data. Thus, often Cluster Analysis forms a precursor to the use of Classification algorithms within Data Mining.

Kohonen Neural Networks [BIGU96], Rule Induction (Chapter 6), Fuzzy Clustering [MIYO90] and Bayesian techniques [CHEE96] are the most commonly used paradigms for Clustering.

1.4.3 Regression

Regression is the learning of a function that maps attributes onto a real-valued domain. Regression can be thought of as classification, the only difference being that the classification label, instead of being discrete valued, is continuous. The attribute that is being predicted is called a *dependent variable* while the attributes that are used to predict the dependant variable are referred to as *independent variables*.

An example of a Data Mining task with a Regression goal is that of house price prediction (see Table 2). In this example the Price attribute is the dependent variable.

Table 2: Sample from Housing Database

Ward	**House Type**	**Heating**	**Bedrooms**	**Garage**	**Price**
1	Bungalow	Oil-Fired	4	Double	130, 500
1	Bungalow	Oil-Fired	3	Single	110, 750
1	Terraced	None	2	None	43, 000
2	Terraced	Economy7	2	None	44, 000
2	House	Gas-Fired	6	Single	140, 000
3	House	Gas-Fired	8	Single	150, 000

Traditional statistical techniques for regression use techniques like least squares to discover linear as well as non-linear equations that fit the data presented as input. However, they are unable to handle non-numeric data commonly found in Data Mining data sets. Neural Networks [BIGU96] and Rule Induction [BRIE90] are two of the most widely used techniques in Data Mining for regression as they can handle numeric as well as non-numeric independent variables. In general Neural Networks tend to outperform Rule Induction techniques when performing regression.

1.4.4 Temporal Modelling

This involves rules that are based on temporal data. From a business perspective, the most well known applications of temporal modelling are in financial forecasting. For example, consider stock prices data. Conclusions drawn from such data like “whenever the price of Microsoft shares falls for two days in succession, IBM shares rise by around 2%” or “Microsoft share prices never fall for more than three successive days” can be useful to financial investors.

Though very useful in financial forecasting, Temporal modelling has applications in most sectors. For example, in the medical sector, creating temporal models of patients of diabetes could lead to preventive medical actions being implemented to reduce the risk of certain complications in the diabetic life cycle.

Techniques have been developed for discovering temporal relationships using Discrete Fourier Transforms to map time sequences to the frequency domain [AGRA95]. This technique is based on two observations. Firstly, for most sequences of practical interest only the first few frequencies are strong and secondly, Fourier transforms preserve the Euclidean distance in the time or frequency domain

Other techniques used include Dynamic Time Warping, a technique adopted from the speech recognition field [BERN96], Neural Networks [BIGU96] and Rough Sets [ZIAR93].

1.4.5 Discovery of Associations

Discovery of associations involves rules that associate one set of attributes in a data set to other attribute sets. An association rule is in the form A → B, where A and B are conjunctions of expressions on attributes of the database. ‘A’ is referred to as the Antecedent and ‘B’, the consequent. For example, if we have a table containing information about people living in Belfast, an association rule could be of the type

(Age < 25) Ù (Income > 10000) ® (Car_model = Sports)

This rule associates the Age and Income of a person to the type of car he drives.

Associations are the most basic type of pattern. They are the purest form of Data Mining and include Classification (Section 0) and Characteristic (Section 0) rules as special cases. The definition of an Association is so loose that it is difficult at times to justify such patterns as knowledge as rather than providing summaries of the data used for discovering these types of patterns, the number of discoverable association patterns far exceed the amount of data being used to discover them. The usefulness of such discovery of associations is only when used in conjunction with user guidance in the form of parameter setting for filtering out obvious and non-interesting associations. The most commonly used parameters are threshold values for the support and confidence (or uncertainty) of a rule.

Set oriented approaches [AGRA93, AGRA94] developed by Agrawal et. al. are the most efficient techniques for discovery of such rules. Other approaches include attribute-oriented induction techniques [HAN94], Information Theory based Induction [SMYT91] and Minimal-Length Encoding based Induction [PEDN91].

1.4.6 Sequential Pattern Discovery

Sequential pattern discovery is similar to discovery of associations. The difference here is that Sequential Pattern Discovery techniques discover associations across time. For example, consider the sample database in Table 3, a Sequential Pattern may be that whenever somebody buys a Jacket and a Tie, the next time they shop they buy a pair of shoes.

Table 3: Example Retail Transaction Database

Cust_No	Date/ Time	Jacket	Tie	Shoes	Socks	Shirt
1001	02.06.96 09:25	1	1	0	0	0
1002	02.06.96 10.32	1	1	0	1	0
1001	03.06.96 13:34	0	0	1	0	0
1003	04.06.96 13:30	1	0	0	0	0
1003	06.06.96 10:02	0	0	1	0	1
1002	06.06.96 12:03	0	0	1	0	1

A Sequential Pattern is in the form $(A)_{t_i} \rightarrow (B)_{t_j}$, where A and B are conjunctions of expressions on attributes of the database and the attributes in A appear in the database with an earlier time stamp than B i.e. $t_i < t_j$. Using this notation the example rule above may be written as follows:

Jacket and Tie ® Shoes with support = 2/3 and confidence = 1

Extensions to Set Oriented Approaches developed for discovery of associations have been proposed for Sequential pattern discovery [AGRA95a].

1.4.7 Discovery of Characteristics

A characteristic rule is an assertion that characterises a concept. It is typically a conjunction of simple properties that objects of the concept have in common. For example, the symptoms of a disease are characteristics of that disease. The most interesting characteristic description is ***the maximal characteristic description*** that refers to the characteristic description with the maximum number of object properties in it. A characteristic rule is a kind of association rule where the antecedent is constrained as the concept being characterised. As such they are the best type of characterisation of a concept that can be arrived at in the absence of negative examples. They constitute the necessary condition for a concept as opposed to a sufficient condition represented by classification rules. Characteristic rules may be refined as negative examples are collected and may be incrementally converted into classification rules.

Characteristic rules are very useful within Data Mining since, in a number of applications, for example, cross-sales in a customer database, negative examples are not stored in the database. Rule Induction is the primary paradigm for discovery of Characteristic rules (see Chapter 15).

1.4.8 Dependency Modelling

The goal is to discover all significant dependencies that exist between variables of the real-world process being modelled by the data. Dependencies may be interpreted as causal relationships and discovering such dependencies along with their local probability distributions can be useful in building probabilistic knowledge bases modelling the real-world process and capable of probabilistic inference [HECK96]. Fayyad et al. [FAYY96] split dependency modelling into structural and quantitative modelling. Structural modelling refers to the discovery of the dependencies itself while the quantitative modelling refers to the local probability distributions or the strength of the dependencies.

Alternatively, the dependencies discovered may be in the form of functional or multi-valued dependencies [BELL93] that can be used to resolve heterogeneity issues in legacy systems or restructure the databases being mined.

1.4.9 Deviation Detection

A deviation is defined as the difference between an observed value and a reference value. Deviations are of a number of types [PIAT94]: Deviation over time, Normative Deviation and Deviation from Expectation. These three types of deviations differ in the norm used to calculate the deviation of the observed value. In deviation over time, the norm would be based on the value of the variable over a certain time period in the past. For example, last years 3rd quarter could be the norm against which this years 3rd quarter value is compared. When a standard norm is available as a reference value, deviation from that value is referred to as normative deviation. In deviation from expectation, the expected value may be generated from a model or may be based on a hypothesis provided by a domain expert.

A discovery system based on deviation detection consists of a number of components. Firstly, a component is needed for identifying significant deviations from user defined norms. Next, a component is required that can order the deviations in terms of their interestingness, and can remove any redundant results. Next, the deviations must be explained. An explanation may come from a decomposition of a formula that defines the finding measured, for example, Total_perks = Child_education * number_of_children + Sales * 0.1, where Child_education is the amount paid to the employee per child that qualifies for the “Educate Children” scheme. It may turn out that the number of children eligible for the companies “Educate Children” scheme far exceeded the norm. Thus, the more interesting deviation is the number of children of the company’s employees eligible for the “Educate Children” scheme as opposed to the deviation in Total_perks payment made by the company. Alternatively, the explanation may be arrived at by breaking down the measure into values from sub-sectors formed by decomposing the sector of the finding. For example, if the sales of a particular Whisky brand is found to be very high in Scotland in 1996, the reason may be that the product has had greater sales in Edinburgh during the summer only whereas in the rest of Scotland the sales have been normal. This in turn could be explained by a larger number of tourists visiting Edinburgh during that period due to a new advertising policy of the Scottish tourist board. The final component of the discovery system is the report generator including possible recommendations for rectifying the deviation discovered.

Deviation detection has proved useful in certain sectors where there are set measures of performance and deviations from their norms are of interest. For example, KEFIR (Key Findings Reporter) [MATH94], a domain independent system for discovering and explaining key findings based on deviation detection was very successful in the Health care sector because pre-defined measures for deviation detection such Average_hospital_payments_per_capita and Admission_rate_per_1000_people were available. In this respect, KEFIR is user driven and discovery is performed in pre-defined paths. However, deviation detection can be automated as well and is the basis of belief system based interestingness measures for knowledge discovered using techniques other than deviation detection.

1.5 Knowledge Representations

In this section we discuss seven of the most common knowledge representation techniques employed in Data Mining. While the knowledge representation technique required depends on the domain in which Data Mining is being performed, a common requirement of most Data Mining exercises is the understandability of the discovered knowledge. Understandability of a knowledge structure is normally measured by its simplicity, however, an over simplistic knowledge representation technique could restrict the quality of the discoverable knowledge. The Data Mining goal and paradigm used often dictate the knowledge representation, however, when alternative paradigms are available for achieving the goal, the knowledge representation technique used by the paradigm could be the deciding factor.

We now describe Decision Trees, Rules, Neural Networks, Bayesian Belief Networks, Exception Directed Acyclic Graphs, Decision Graphs and Predicate Logic. We do not discuss knowledge representations such as frames and semantic networks as their use in Data Mining is very uncommon. For a more comprehensive discussion on knowledge representation techniques, the reader is referred to [BENC90].

1.5.1 Decision Trees

A Decision Tree is a tree-based knowledge representation methodology used to represent classification rules. The leaf nodes represent the class labels while non-leaf nodes (as known as decision nodes) represent the attributes associated with the objects being classified. The branches of the tree represent each possible value of the decision node from which they originate.

Given a training data set, the number of possible decision trees that can be induced is large, therefore, tree induction algorithms use heuristics to guide their search for the best decision tree. The heuristics help in picking the attributes that form the decision nodes in the tree, splitting of numeric attributes, defining the stopping criterion for when to stop further splitting of the tree and defining the extent to which the tree should be pruned to reduce over-fitting of the training data. Measures for goodness of the induced decision tree are based on size, understandability and accuracy.

Once the decision tree has been built using a training set of data it may be used to classify new objects. To do so we start at the root node of the tree and follow the branches associated with the attribute values of the object until we reach a leaf node representing the class of the object.

Though decision trees have been used successfully in a number of applications, they have a number of disadvantages. Firstly, even for small training sets decision trees can be quite large and thus opaque. Quinlan [QUIN87] points out that it is questionable whether opaque structures like decision trees can be described as knowledge, no matter how well they function. In the presence of missing values for attributes of objects in the test data set, trees can exhibit poor performance. The knowledge representation is very limited. Firstly, trees that test on multiple attributes at a single node or examine relations between attributes are excluded. Also, when the knowledge to be discovered contains a disjunction, the trees induced are large and contain replicated sub-trees (see Figure 1. **6**). They are not incremental in nature as the search for the best tree is greedy and *one-step optimal* i.e. at each node the best attribute is chosen and there is no way of backtracking later. Thus, when new data is presented to the tree, the tree must be re-discovered. Due to the greedy nature of the search mechanism used by induction algorithms they tend to favour attributes for a decision node that contain a larger number of unique values as they tend to split the data set into purer, i.e. homogeneous, subsets even though the resulting subsets may have little statistical significance.

A number of solutions have been suggested in literature that are based around either making decision trees more expressive by removing some of the representational constraints on them, for example, Decision Graphs (Section 0) or by refining heuristics used in selection of the attributes for the decision nodes, for example, the information gain ratio improves on the information gain measure that favoured multi-valued attributes.

The main advantage of decision trees is their execution efficiency mainly due to their simple and economical representation and ability to perform even though they lack the expressive power of semantic networks or first order logic methods of knowledge representation.

1.5.2 Rules

Rules are probably the most common form of data representation. A rule is a conditional statement that specifies an *action* for a certain set of *conditions*, normally represented as X → Y. The *action*, Y, is normally called the *consequent* of the rule and the set of *conditions*, X, its *antecedent*. A set of rules is an unstructured group of IF.. THEN statements.

There are two main types of rules - Exact rules and Probabilistic rules. Exact rules are rules where the consequent of the rule is always true when the antecedent of the rule is true. Probabilistic rules are rules where the consequent of the rule is true to some degree of belief when the antecedent is true. Therefore, a probabilistic rule has a belief measure associated with it that indicates the belief in the fact that when the antecedent of the rule is true, the consequent will also be true.

The popularity of rules as a method of knowledge representation is mainly due to their simple form. Humans easily interpret them, as they are a very intuitive and natural way of representing knowledge, unlike decision trees and neural networks. Also as a system of rules is unstructured, it is less rigid, which can be advantageous at the early stages of the development of a knowledge based system.

But representing knowledge as rules has a number of disadvantages. Rule representation is inadequate to represent many types of knowledge e.g. causal knowledge. As the number of rules in the system increases the performance of the system decreases and the system becomes more difficult to maintain and modify. New rules cannot be added arbitrarily to the system as they may contradict existing rules of the system leading to erroneous conclusions. The degradation in performance of a rule based system is not graceful.

The lack of structure in rule based representations makes the modelling of the real-world difficult if not impossible. Thus, a more organised and structured representation for knowledge is desirable that can make partial inferences and degrade gracefully with size.

1.5.3 Neural Networks

The human brain consists of a network of approximately 10^{11} neurones. Each biological neurone consists of a number of nerve fibres called ***dendrites*** connected to the ***cell body*** where the ***cell nucleus*** is located. The ***axon*** is a long, single fibre that originates from the cell body and branches near its end into a number of ***strands***. At the ends of these strands are the transmitting ends of the ***synapse*** that connect to other biological neurones through the receiving ends of the synapse found on dendrites as well as the cell body of biological neurones. A single axon typically makes thousands of synapses with other neurones. The transmission process is a complex chemical process that effectively increases or decreases the electrical potential within the cell body of the receiving neurone. When this electrical potential reaches a threshold value (action potential) it enters its excitatory state and is said to fire. It is the connectivity of the neurones that give these simple ‘devices’ their real power. Figure 1. 2 shows a typical biological neurone.

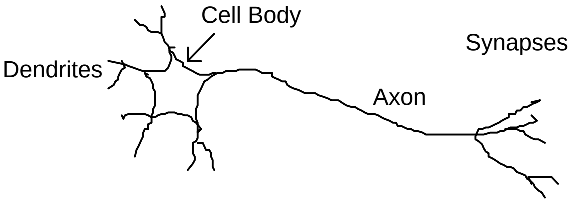

Figure 1. 2: A Biological Neurone

An Artificial Neurone (or processing element, PE) is a highly simplified model of the biological neurone. As in biological neurones an artificial neurone has a number of inputs, a cell body (consisting of the summing node and a semi-linear activation function see Figure 1. 3) and an output which can be connected to a number of other artificial neurones.

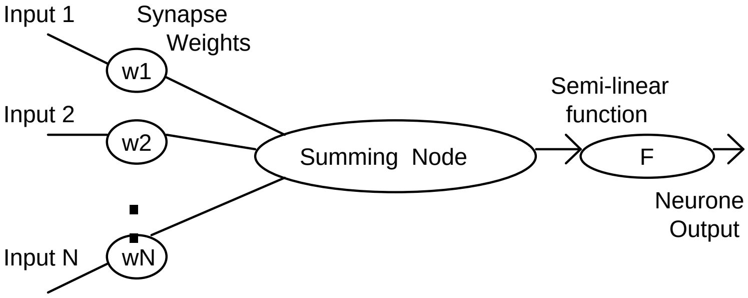

Figure 1. 3: An Artificial Neurone

Neural Networks are densely interconnected networks of PEs together with a rule to adjust the strength of the connections between the units in response to externally supplied data. The overall behaviour of a network is determined by its connectivity rather than by the detailed operation of any element. Different topologies for neural networks are suitable for different tasks e.g. Hopfield Networks for optimisation problems, Multi-layered Perceptron for classification problems and Kohonen Networks for data coding.

Thus, a neural network can be considered as a flexible knowledge representation method that can be used to handle a number of different types of knowledge. The knowledge being represented in the form of the connectivity of the neurones and the synapse weights. Though neural networks are effective in representing knowledge, they are difficult to interpret and, despite their sound biological underpinning, are not intuitive to humans.

1.5.4 Bayesian Networks

Bayesian Networks are a graphical representation for uncertain knowledge. A Bayesian Network consists of a directed acyclic graph, with nodes representing the variables of the problem domain and arcs representing dependencies between the variables, and a set of local conditional probability distributions that together describe the global joint probability distribution of the problem domain. Figure 1. 4 shows an example Bayesian Belief Network for diagnosis of engine problems.

Though the arcs represent dependencies between variables within the problem domain, they can be interpreted as cause-effect relationships or influence relationships. Thus, Bayesian Networks are also known as Cause-Effect Networks and Influence diagrams. The interpretation of the independence relationships as cause-effect relationships make Bayesian Networks an intuitive mode of communication between experts and knowledge engineers which has been a major reason for their success in recent expert system development.

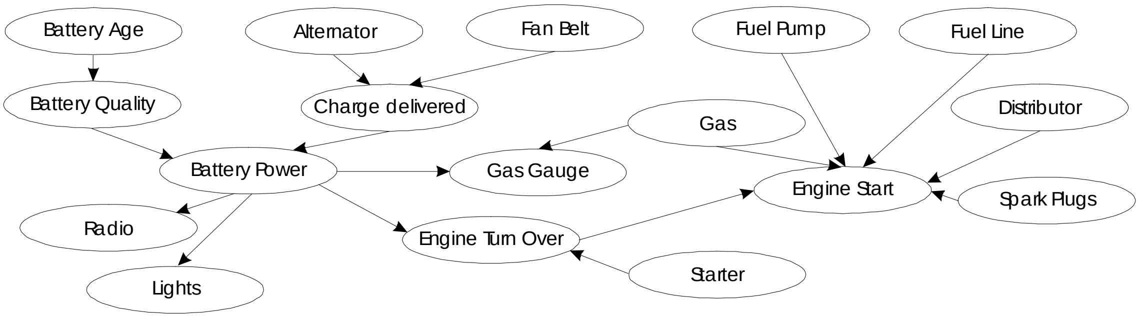

Figure 1. 4: A Bayesian Network for Diagnosis of Engine Problems [HECK95]

The model represented by the graph and local conditional probability distributions may be utilised for probabilistic inference. Capturing the independence relationships result in the chain rule of probability a feasible approach to calculating the joint probability distribution, making probabilistic inference more efficient.

1.5.5 Exception Directed Acyclic Graphs

Exception Directed Acyclic Graphs (EDAGs) [GAIN96] is a knowledge representation technique that subsumes production rules, decision trees and rules with exceptions in an attempt to make knowledge structures more comprehensible.

There are two main parts to an EDAG - Nodes and Arcs. Nodes represent premises (antecedent), some attached to conclusions (consequent) while Arcs represent inheritance links with disjunctive multiple inheritance. Such a knowledge structure allows for a much more concise knowledge representation, an important measure for comprehensibility of a knowledge structure. Figure 1.5 shows an example EDAG.

Traversing of an EDAG starts at the root nodes. If the premise of the node is found to be true the conclusion associated with it is taken as the intermediate conclusion and the various arcs originating at the node are followed. Each path is followed until a premise is not satisfied or a leaf node is arrived at. Any conclusions arrived at during this traversing of the EDAG replace the intermediate conclusion. When no paths can be traversed any further, the current intermediate conclusion is asserted. For example in Figure 1.5 conclusion 1 is a default conclusion, if premise 3 is true, conclusion 3 replaces conclusion 1 as the intermediate conclusion. Next if premise 5 is found to be false, conclusion 3 is asserted.

EDAGs do not necessarily have a single root node and may consist of several disconnected parts as in the case of Figure 1.5. Multiple conclusions may arise from traversing the EDAG as the premises need not

Figure 1.5: Exception Directed Acyclic Graph

be mutually exclusive. EDAGs are not binary structures nor are they tree structures and can therefore have more than two arcs originating or culminating at them. A node need not have a conclusion with the resulting node representing a common premise to all the conclusions of the child nodes. Similarly a node need not have a premise. Such a node represents a common conclusion of all the premises of its parent nodes or it may represent a default conclusion. A null node may be used to avoid arc crossing leading to incomprehensibility of the EDAG. Conclusions may or may not be fully decidable due to missing information as a premise may take a truth value of true, false or unknown.

Each conclusion within an EDAG can be explained by the negation of its child nodes' premises and the premises from it back to the root node. For example, conclusion 11 has the complete premise

(premise 2) Ù (premise 11) Ù Ø(premise 12 Ú premise 13)

As can be seen, rules without exceptions or a default form the trivial EDAG without any connections while a decision tree is an EDAG restricted to a tree structure. Rules with exceptions may be represented as EDAGs as well where each node has a premise and a conclusion. However, both trees and rules with exceptions can be re-represented in the form of general EDAGs by factoring out common sub-premises and the removal of replications within decision trees occurring due to their inability to represent disjunctive concepts.

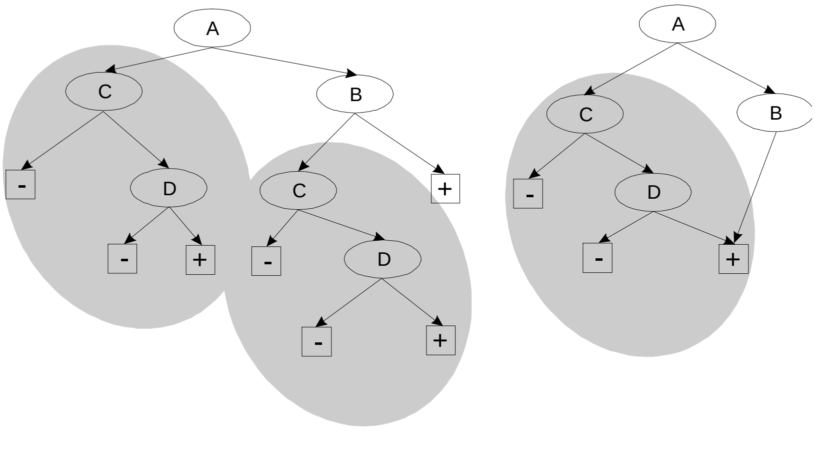

Figure 1. 6: Decision Trees Vs Decision Graphs

1.5.6 Decision Graphs

Decision graphs [OLIV93] or Decision Pylons are directed acyclic graphs used to represent knowledge in a succinct way. Decision graphs are an extension to decision trees that attempt to reduce the size of the knowledge structure by addressing two problems faced by decision trees, namely, Replication and Fragmentation. Decision trees are excellent at representing conjunctions, however, when representing disjunctive models, they result in very large trees due to the same sub-tree appearing in more than one part of the decision tree. For example, the decision trees and decision graphs representing $(A \wedge B) \vee (C \wedge D)$ are shown in Figure 1. 6. The decision tree contains the sub-tree with root node C and child node D twice, unnecessarily increasing its size. Decision graphs avoid this replication by removing the restriction that the knowledge structure must be a tree.

The distinguishing feature of decision trees is the join. A join is a node that has more than one parent. Joins can be used to remove replication from decision trees thus improving the clarity of the knowledge structure by reducing its size.

Induction algorithms for Decision Graphs differ from those inducing decision trees in that they have to choose between splitting and joining the leaf nodes at each stage as opposed to just choosing an attribute for a new decision node. Thus, these algorithms are computationally more expensive but as can be seen from Figure 1. 6, they induce knowledge structures that are, in the worst case, decision trees.

1.5.7 Predicate Logic

Logic was developed in philosophy and mathematics with the sole objective of representing and assessing the soundness of argumnets. Thus, knowledge representation requirements such as mataphysical adequacy, epistemic adequacy, lack of ambiguity and clarity [BENC90] are also features of logic. In addition to the representational compatibility between knowledge representation and logics, logics have the additional advantage of having a well-defined deductive engine available, aiding in deduction of knowledge. Thus, logics are a natural choice for a knowledge representation scheme.

The simplest form of logic is propositional calculus. A number of knowledge representations utilise propositional calculus as their basis for representation, for example, decision trees use conjunctive normal form expressions for representation of knowledge. However, propositional calculus is too restrictive in a number of domains [DZER96], where predicate logic is used instead. Most knowledge-based systems, however, utilise only a subset of predicate calculus to achieve computational tractability, for example, Annotated Predicate Calculus [MICH83]. Predicate calculus allows the representation of relations between object properties in addition to statements that can be represented in propositional calculus.

Logics with even greater expressive power such as temporal logic, non-monotonic logic, polyvalent logic and fuzzy logic have also been developed to address the needs of more complex domains.

1.6 Data Mining Paradigms

In this section we describe the main Data Mining paradigms. We classify the paradigms into five main groups: Machine Learning methods, Statistical methods, Uncertainty based methods, Database methods and Visualisation. We discuss these paradigms individually, however, the reader should keep in mind the fact that often hybrid systems that utilise a number of these paradigms together are used in Data Mining. For example, Neuro-fuzzy systems, Neuro-genetic systems and Case-based reasoning and Rule Induction hybrid systems.

1.6.1 Machine Learning Methods

Automating the process of learning has enthralled AI researchers for some years now. The basic idea is to build a model of the environment using data describing the environment.

Over the last few decades machine learning has developed into a mature discipline. Biological systems have been the source of inspiration to a number of different paradigms with the same goal in mind - an improved performance of a process that is being modelled by the intelligent system by exploiting regularities in the data collected from the process. Four major paradigms that have emerged in machine learning are Rule Induction, Connectionist approaches, Genetic approaches and Case-base reasoning. We discuss these paradigms in this section.

1.6.1.1 Rule Induction

The dictionary definition of Induction is “A process of reasoning by which a general conclusion is drawn from a set of premises, based mainly on experience and experimental evidence”. Thus, the basic idea of Rule Induction is to build a model of the environment using sets of data describing the environment. The simplest model clearly is to store all the states of the environment along with all the transitions between them over time. For example, a chess game may be modelled by storing each state of the chess board along with the transitions from one state to the other. But the usefulness of such a model is limited as the number of states and transitions between them are infinite. Thus, it is unlikely that a state that occurs in the future would match, exactly, a state from the past. Thus, a better model would be to store abstractions/ generalisations of the states and the associated transitions. The process of generalisation is called *Induction*.

A general paradigm for inductive inference is defined in [MICH83] as follows. Here, H is a *generalisation* of F and F is a *specialisation* of H. Deriving F from H is truth-preserving, however, inducing H from F is falsity-preserving. Allowing for weak implications means that probabilistic hypotheses are included in the paradigm where the hypothesis only accounts of a subset of F rather than the whole set.

Given:

- Observational Statements, F, representing specific statements about a process or object
- Tentative Inductive Assertions and
- Background Knowledge including assumptions and constraints imposed on the observations, candidate assertions, domain knowledge and preference criteria describing the desirable characteristics of the induced assertion.

Find

- an Inductive assertion (hypothesis), H, that tautologically or weakly implies the observation statements and satisfies the background knowledge.

The preference criterion is important, as the number of hypotheses that can possibly be drawn from F is infinite. The typical way to define such a criterion is to specify the preferable properties of the induced hypotheses e.g. Occam’s Razor (“biased-choice”). Alternatively, a restrictive language could be used to represent the hypotheses, which would reduce the number of plausible hypotheses (“biased-language”).

The induced hypothesis is a *concept recognition rule* such that if an object satisfies the rule it would be an instance of the concept. A concept recognition rule must satisfy the *completeness condition* i.e. every object in the training set that satisfies the concept description is an instance of the concept as well as the *consistency condition* i.e. any object in the training set that satisfies the concept description is not an instance of any other concept.

1.6.1.2 Connectionist Paradigms

Neural Networks is a non-symbolic or connectionist paradigm of machine learning that finds its inspiration from neuroscience. The realisation that most symbolic learning paradigms are not satisfactory in a number of domains e.g. pattern recognition, that are regarded by humans as trivial lead to research into trying to model the human brain.

Using neural networks as a basis for a computational model has its origins in pioneering work conducted by McCulloch and Pitts in 1943 [MCCU43]. They suggested a simple model of a neurone that computed the weighted sum of the inputs to the neurone and output a 1 or a 0 according to weather the sum was over a threshold value or not. A zero output would correspond to the inhibitory state of the neurone while an output of 1 would correspond to the excitatory state of the neurone. But the model was far from a true model of a biological neurone. For a start the biological neurone output is a continuous function rather than a step function. The step function has been replaced by other more general, continuous functions called *activation functions*. The most popular of these is the Sigmoid function defined as:

$$f(x) = 1/(1+e^{-x})$$

There are three main ingredients to a neural network: the neurones and the links between them, the algorithm for the training phase and a method for interpreting the response from the network in the testing phase.

The learning algorithms used during the training phase are normally iterative, e.g. back-propagation algorithm, attempting to reduce the error in the output of the network. Once the error is reduced (not necessarily minimised) the network can be used to classify other unseen objects.

Though neural networks seem an attractive concept they have a number of disadvantages. Firstly, the learning process is very slow compared to other learning methods. Secondly, the learned knowledge is in the form of a weighted network, and is difficult for a user to interpret. Thirdly, user intervention in the learning process, interactively, which is normally required in Data Mining applications is difficult to incorporate. However, neural networks are known to perform better than symbolic learning techniques on noisy data found in most real-world data sets.

1.6.1.3 Genetic Algorithms

Genetic Algorithms [SRIN94] were first proposed by Holland in the early 1970s [HOLL75] but have gained momentum only in the last decade. They model the evolutionary process and are based on the principle of survival of the fittest. Genetic Algorithms start with a population of solutions represented as binary strings that are the equivalent of a gene pool in nature. Each string has an associated fitness value that directly influences its chances of surviving in the next population. The selected candidate strings are subjected to crossover and mutation - two genetic operations that take place during reproduction, in nature. Crossover is the primary operator utilised in genetic algorithms, used for exchange of genetic material between strings in a population. Mutation is used as a secondary operator and is used only in situations when lost genetic information needs to be regenerated. For example, if the optimal solution contains a 1 in a particular position of the string but the population only consists of strings with 0 in that position, the only way to rectify this is through mutation, crossover can never achieve a 1 in the desired position under these circumstances. The crossover and mutation are controlled using two parameters called the crossover probability and the mutation probability. The crossed and mutated strings form the new population and the whole process is repeated until a certain stopping criterion is arrived at. The stopping criteria is the criteria picked which if satisfied stops the cycle of generating new populations and the string with the greatest fitness value is the solution. The chosen criteria may be a degree of homogeneity within the population, a threshold on the maximum fitness value or simply the number of genetic cycles.

As compared to traditional optimisation techniques like hill-climbing, genetic algorithms tend to perform better due to their inherent parallelism and ability to avoid local minima.

1.6.1.4 Case Based Reasoning

Case based reasoning (CBR) is a machine learning paradigm modelled on the human problem solving process. When a new problem is encountered, the problem solving process followed by humans is to remember a previous problem that is similar to the present problem, adapt the solution to the problem based on the difference between it and the present problem and then applying the new, adapted solution. This is the basis of CBR.

A CBR system consists of a case-base containing previous problems and their solutions, a similarity measure that is used as a measure of how similar two cases are, an indexing mechanism for accessing similar cases, a set of adaptation rules used to adapt the retrieved case to solve the present problem and domain knowledge that may be used in defining the similarity measure.

Normally case bases incorporate a certain amount of domain knowledge and structuring that transform databases into case-bases. However, in their simplest form, a case base could be a database. Thus, using this simplistic model of a CBR system, CBR has been used for Data Mining tasks with Classification or Regression goals. However, one of the major drawbacks of using CBR systems is the initial set-up costs associated with it i.e. acquiring the adaptation knowledge. This bottleneck is reminiscent of the bottleneck associated with building rule-base systems. However, CBR has great potential in being used in conjunction with other Data Mining paradigms providing goal-oriented mining [ANAN97b].

Other paradigms of machine learning related to CBR are instance-based learning and learning by analogy.

1.6.2 Statistical Methods

Statistical techniques may be employed for data mining at a number of stages of the mining process (Section 0). In fact statistical techniques have been employed by analysts to detect unusual patterns and explain patterns using statistical models. However, using statistical techniques and interpreting their results is difficult and requires a considerable amount of knowledge of statistics. Data Mining seeks to provide non-statisticians with useful information that is not difficult to interpret. We now discuss how statistical techniques can be used within Data Mining.

The presence of outliers may be detected by methods that involve thresholding the difference between particular attribute values and the average, using either parametric or non-parametric methods. Exploratory Data Analysis concentrates on simple arithmetic and easy-to-draw pictures to provide *Descriptive Statistical Measures and Presentation*, such as frequency counts and table construction (including frequencies, row, column and total percentages), building histograms, computing measures of location (mean, median) and spread (standard deviation, quartiles and semi inter-quartile range, range).

Principal Component Analysis (PCA) is of particular interest to Data Mining as most Data Mining algorithms have linear time complexity with respect to the number of tuples in the database but are exponential with respect to the number of attributes in the data. Attribute Reduction using PCA thus provides a facility to account for a large proportion of the variability of the original attributes by considering only relatively few new attributes (called Principal Components) which are specially constructed as weighted linear combinations of the original attributes. The first Principal Component (PC) is that weighted linear combination of attributes with the maximum variation; the second PC is that weighted linear combination which is orthogonal to the first PC whilst maximising the variation, etc. The new attributes formed by PCA may possibly themselves be assigned individual meaning if domain knowledge is invoked. The facility for PCA requires the partial computation of the eigensystem of the correlation matrix, as the PC weights are the eigenvector components, with the eigenvalues giving the proportions of the variance explained by each PC.

Sampling is an efficient way of discovering knowledge, and resampling offers opportunities for cross-validation and bootstrapping for knowledge validation. Hierarchical data structures may be explored by segmentation and stratification.

Statistics provides a number of tools for data analysis some of which may be employed within Data Mining. Measures of association and relationships between attributes, such as computation of expected frequencies and construction of cross-tabulations, computation of chi-squared statistics of association, presentation of scatterplots and computation of correlation coefficients. The interestingness of rules may be assessed by considering measures of statistical significance [PIAT91]. Inferential Statistics for hypothesis testing, such as construction of confidence intervals, parametric and non-parametric hypothesis tests for average values and for group comparisons have also been employed in Data Mining. In addition, classification may be carried out using discriminant analysis (supervised) or cluster analysis (unsupervised).

1.6.3 Uncertainty Based Methods

Uncertainty modelling has been an important issue in knowledge based systems since their inception. Modelling uncertainty is made difficult due to the different sources of uncertainty as well as its different aspects. Initial systems utilised probability theory as the uncertainty model, though it soon became apparent that alternative techniques would be required. This was mainly due to two reasons. Firstly, the representation used by probability theory was found to be too restrictive and secondly, probabilistic techniques available at the time required many parameters or assumed unrealistic sets of independence relationships [HECK95]. A number of alternative approaches have been suggested in AI literature, the most common being Certainty Factors, Evidence Theory, Fuzzy Set Theory and more recently Rough Set Theory. While Evidence Theory and Fuzzy Set Theory have provided a richer representation that covers more aspects of uncertainty than could be represented using probabilistic methods, Rough Set Theory goes one step further by removing the requirement for any parameters. Rough Set Theory uses only information that can be gathered from the data itself, thus making it a useful tool in Data Mining. Recently, probabilistic methods are making a comeback in the form of Bayesian Belief Networks. In this section we discuss some of these uncertainty modelling techniques in greater detail.

1.6.3.1 Bayesian Belief Networks

A Bayesian Belief Network (BBN), also known as influence diagram and causal network, is a graphical technique that is used to model uncertain information within an environment based on relationships between variables of the environment. These networks are underpinned by well-established Bayesian probability techniques allowing for information to be inferred. The discovery of these models from data are still in their infancy though real-world applications using such models within knowledge-based systems have been reported recently [HECK95].

A BBN represents the joint probability distribution over the problem domain. It consists of a set of local conditional probabilities, combined with a set of independence assertions. Using the **chain rule of probability**:

$$p(x_1, x_2, \ldots. x_n) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots. x_{i-1})$$

the joint probability distribution for the domain can be constructed using the local conditional probability distributions and independence assertions. To make the calculations more efficient, for each variable x_i, a subset of the set $\{x_1, x_2, \ldots x_{i-1}\}$ that consists of the variables that x_i is dependent on is used instead. This subset is called the conditioning set of x_i, Π_i.

A BBN is a directed acyclic graph that has variables from the domain represented as nodes and directed arcs connecting them. The parent nodes in the Bayesian network, of the variable x_i corresponds to the variables in $\{x_1, x_2, \ldots x_{i-1}\}$ that x_i is dependent on. Thus, the Bayesian network encodes the assertions of conditional independence. The inverting of the arcs within the graph is equivalent to an application of Bayes theorem. Deterministic nodes can be used to simplify the network where the number of parent nodes is large.

Given the BNN, in principle it should be possible to induce any probability of interest. However, the exponential number of joint distributions that need to be taken into account can be daunting. Using the condition sets for each variable instead can reduce this number. This is equivalent to dimensionality reduction in other inference methods.

$$p(w|z) = \frac{p(wz)}{p(z)} = \frac{\sum_{xy} p(w, x, y, z)}{\sum_{wxy} p(w, x, y, z)} = \frac{\sum_{xy} p(w).\, p(x|w).\, p(y|wx).\, p(z|wxy)}{\sum_{wxy} p(w).\, p(x|w).\, p(y|wx).\, p(z|wxy)}$$

The inference of probabilities using such a mechanism is termed as probabilistic inference. While there are a number of applications of Bayesian Networks in industry, these are in domains where the number of independence assertions are high and well understood so that the dimensionality reduction is effective. In areas where such a clear understanding is not available, probabilistic inference is NP-hard.

1.6.3.2 Evidence-based Methods

Probability Theory has been termed as a prescriptive model of uncertain reasoning as opposed to a descriptive model, as it imposes reasoning mechanisms that are not used by humans. In general, humans do not associate belief in the exhaustive set of propositions prior to reasoning. Instead, they tend to associate belief to a number of propositions based on the available evidence. Evidence Theory [SHAF76] is a generalisation of probability theory that attempts to model uncertainty epistemologically. Belief is associated with propositions as evidence is received. As new evidence is received the belief in the propositions gets transferred to other propositions.

Evidence Theory provides an explicit representation of ignorance, thus once belief has been assigned to all the propositions supported by the current evidence, the residual belief can be associated with ignorance. Support is associated with the various propositions within the frame of discernment of the problem in hand using evidential functions such as the mass function, belief function and plausibility function. Evidence

Theory has operators defined on these evidential functions for combining, refining, coarsening and discounting belief associated with the different propositions that are supported by the available evidence.

Anand et al. [ANAN96] have proposed a framework for Data Mining using Evidence Theory that utilises its representational richness and combines it with rule induction techniques for developing robust Data Mining solutions.

1.6.3.3 Fuzzy Set Theory and Fuzzy Logic

Fuzzy Set theory was introduced by Zadeh [ZADE65] as a generalisation of conventional set theory. As opposed to the “crisp” character of conventional set theory where an element either is or is not a member of a set, in Fuzzy Set theory, a membership function is associated with the elements of the universe that signifies its degree of membership of the set. Based on Fuzzy Set theory, Fuzzy Logic is defined as a generalisation of conventional logic. Though Fuzzy Logic has been most successful in control applications, it also has applications in knowledge based data analysis. Two of its most widely applied areas are developing linguistic summaries of data [YAGE91] and fuzzy clustering. As opposed to crisp clustering, in fuzzy clustering each cluster has a membership function defined on the universe of objects. Clustering may be carried out using iterative strategies as in the case of Fuzzy C-means [ZIMM91] or using hierarchical clustering methods (agglomerative as well as divisive) [MIYA90].

Fuzzy Logic is increasingly being used in hybrid systems along with learning paradigm, for example, neural networks and genetic algorithms [MUNA94]. While genetic algorithms have been used to fine tune membership functions in fuzzy systems, neural networks have been employed to aid fuzzy systems by adding learning and pattern recognition capabilities to them.

1.6.3.4 Rough Set Theory

Rough Set Analysis was introduced by Pawlak in 1982 [PAWL82] as a model for handling uncertain information. The starting point of Rough Set Analysis is the Information or Decision Table. Rows within the decision table are called examples while the columns are called attributes. The attributes are partitioned into condition attributes and decision attributes. The fundamental concept in Rough Set Analysis is the Indiscernability Relation (IR) which is associated with a set of attributes. All other concepts within Rough Set Analysis are defined in terms of the indiscernability relation. The IR is an equivalence relation that partitions the decision table into elementary sets based on the unique values of the corresponding attributes. Examples within each elementary set are indiscernable from each other i.e. they have the same values for the associated attributes. A special case of elementary sets called concepts are elementary sets resulting from the IR corresponding to the decision attribute set. A finite union of elementary sets is said to be a definable set.

If the IR corresponding to a set of attributes, P, as well as its superset define the same elementary sets, the attributes in the superset that are not in P are said to be redundant. A set, P, with no redundant attributes is said to be minimal and is a reduct of another set Q if Q defines the same elementary sets.

A decision table is said to be inconsistent if the elementary sets defined by the IR corresponding to the condition attributes are not subsets of the concepts. When presented with an inconsistent decision table, Rough Set Analysis, uses upper and lower approximations to deal with the inconsistencies. For each concept X, the lower approximation is the greatest definable set contained in X while the upper approximation is the least definable set containing X. The set difference between the upper and lower approximations is known as the boundary region. A concept with a non-empty boundary region is called a rough set. Rules induced from the lower approximation are certain while those in the upper approximation are possible. Associated with the lower and upper approximations are the uncertainty values quality of lower approximation and quality of upper approximation that are defined as the cardinality of the lower approximation to the number of rows in the decision table and the cardinality of the upper approximation to the number of rows in the decision table, respectively.

Rough Set Analysis has a strong correspondence with Evidence Theory, with the lower and upper approximations corresponding to belief and plausibility functions. It is useful to note that Rough Set Analysis is objective in its measure of uncertainty and requires no external parameters as is the case with other uncertainty handling and analysis tools, for example, membership functions in fuzzy set theory and probability distributions in statistical analysis. This makes it a very attractive tool for Data Mining. Another useful aspect of Rough Set Analysis for Data Mining is the simple, mathematically sound definition of redundant attributes making it a powerful tool for data dimensionality reduction.

1.6.4 Database Methods

Set-oriented approaches to data mining attempt to employ facilities provided by present day DBMSs to discover knowledge. This allows the use of years of research into database performance enhancement to be used within the Data Mining processes. However, SQL is very limited in what it can provide for Data Mining and therefore techniques based solely on this approach are very limited in applicability. Though these techniques have shown that certain aspects of Data Mining can be performed within the DBMS efficiently, providing a challenge for researchers into investigating how the data mining operations can be divided into DBMS operations and non-DBMS operations to make the most of both worlds. Database methods include set-oriented techniques used for discovery of associations [AGRA93] and sequential pattern discovery [AGRA95a] and attribute oriented induction [HAN96].

Discovering knowledge from multi-databases is mainly concerned about extending existing Data Mining algorithms to handle vertically split distributed and heterogeneous database environments. Data Mining algorithms operating on vertically and hybridly split distributed data have to keep track of primary and foreign keys when locally discovered knowledge is combined [RIBE96]. Data Mining mechanisms for heterogeneous databases have to incorporate available schematic information as domain knowledge and provide communication facilities among the local discovery sites [BÜCH97].

Another stream of database method development for Data Mining is the development of the Knowledge and Data Discovery Management Systems [IMIE96] and extended SQL for Data Mining [MEO96].

1.6.5 Data and Knowledge Visualisation

The general idea of visualisation in the Data Mining context is to map multidimensional data (in form of relations) into a two- or three-dimensional Cartesian coordinate system. This view can then be displayed on a two-dimensional device/display and allows finding interesting patterns in the data. Dependent on the amount and dimensionality of data as well as the kind of knowledge to be discovered, several visualisation techniques can be applied. Keim et al. [KEIM96] classify these techniques into the following four groups:

- *Pixel-oriented Techniques* map each data value of an attribute to a coloured pixel which are displayed in attribute windows. Pixel-oriented techniques can either be data-driven, i.e. query-independent, or query-driven, i.e. query-dependent. The advantage of this technique is that the number of attribute values that may be visualised is only limited by the number of supported colours, which is more than sufficient with hardware facilities available today.
- *Geometric Projection Techniques* are mainly concerned with finding projections of multidimensional data sets. The overall idea is to visualise similarities relative to appropriate geometric constructs, such as axes and spirals. The technique is not suitable for vast amounts of data, because of overlapping lines in the output.
- *Icon-Based Techniques* map each multidimensional data item to one iconic structure. The structure of an icon depends on the data to be represented, the structure corresponding to the distribution of the data values.
- *Hierarchical and Graph-based Techniques* order data visually in a tree- or network-like style, respectively. Each node and its corresponding sub-nodes built a sub-space which can be used as input for further Data Mining.

Techniques belonging to the four groups of visualisation techniques are aimed at interacting with the user. The interaction is twofold: the provided information might be sufficient for decision making or sub areas of the output might be chosen for further Data Mining. In the first case, visualisation acts as knowledge visualisation tool, in the latter case it aids in data pre-processing.

1.7 Enabling Technologies

In this section we discuss the enabling technologies that can considerably enhance the quality of Data Mining and the ease with which it is achievable. However, they do not form an integral part of the Data Mining solution, as in the absence of these technologies Data Mining can still take place.

1.7.1 High Performance Computing

Scalability and performance are two major issues in Data Mining. Performance is an important consideration as batch Data Mining processes would result in users loosing their train of thought, decreasing the effectiveness and usefulness of Data Mining. Size of the search space as well as model validation techniques make Data Mining computationally expensive and therefore even the most efficient sequential algorithms can prove to be too slow. Thus the need for scalability.

Scalability refers to the fact that Data Mining techniques should be able to utilise additional resources made available to it and accomplish acceptable speed-ups. Recent advances in high performance computing and networking have made the goal of scalability possible.

There are two main architectures in parallel computing: Distributed Computing and Massively Parallel Processor (MPP) machines. Distributed Computing, often referred to as shared nothing machines, are a heterogeneous set of workstations and PCs networked using high speed communication facilities such as ATM to transfer data and messages between them. MPP machines consist of a large number of processors, usually ranging from a 100 to 1000 individual processors, and a large shared memory.

Three main parallel computing paradigms for implementing software on these parallel architectures are Message Passing, Shared Memory and Parallelising Compilers. In the message passing paradigm a number of processes run concurrently, communicating with each other by passing messages and data. In general, message passing is used when algorithms that are to run in parallel can be subdivided into a number of components capable of executing fairly independently. Important issues in this paradigm are the cost of message passing, cost of synchronisation of concurrent processes and load balancing. Developments of message passing libraries like Parallel Virtual Machine [GEIS94], that make underlying hardware heterogeneity transparent, as well as high speed networking hardware have made this type of computing viable.

The shared memory paradigm is utilised in situations where message-passing costs would be high. However, algorithms paralleised using the shared memory paradigm have their own set of overheads like locking and synchronisation. Shared memory implementations may utilise underlying hardware architecture consisting of distributed physical memory. Distributed shared memory systems like Linda [DOUG95] use an abstraction called the tuple space for communication between concurrent processes, thus simulating the shared memory paradigm.

As opposed to the Message Passing and Shared Memory paradigms that require redesigning and reprogramming of algorithms specifically for parallel execution, often tailored to the underlying hardware, the parallelising compiler paradigm automatically parallelises existing sequential code by utilising the inherent parallelism within it. While using such compilers does reduce the redevelopment overheads, they are not as efficient as specifically tailored implementations of the algorithms. Thus, in performance critical applications, the use of such compilers is not viable.

Two main techniques are used in parallelising algorithms: Data Partitioning and Functional parallelism. In Data Partitioning, the data is split into a number of partitions and processing carried out in parallel on each of the partitions. The results obtained from each of the partitions then need to be integrated in some way before use. Examples of Data Mining algorithms that utilise this technique are the Arbiter and Combiner trees for classification algorithms [CHAN96], the Evidential Association Rule (EAR) algorithm [ANAN97a], Data Distribution Apriori algorithm [AGRA96] for Association algorithms and Distributed Mining of Association (DMA) rules [CHEUN96]. All these algorithms utilise horizontal partitioning of the data. Issues concerning discovery using vertical partitioning of data have been discussed in [BÜCH96].

Functional parallelism requires the algorithm to be partitioned into functional components that can be executed in parallel. In the case of Data Mining, the discovery process may be considered as a set of operators on data and intermediate representations of the discovered knowledge (for example, itemsets used by most association algorithms [AGRA94]), each carrying out a specific task towards the goal of discovering knowledge [AGRA93a, ANAN96]. The execution of these operators in parallel has been studied in the case of association algorithms [AGRA96, ANAN95].

1.7.2 Database Management Systems

Existing database management systems (DBMSs) are transaction orientated and contain components such as query languages, data dictionaries, transaction monitors, security facilities, etc. Most Data Mining solutions access those databases, but are not integrated in the database system. The next logical step in the evolution of DBMSs is the synergy of database technology and Data Mining underneath a new umbrella, which Imielinski et al.[IMIE96] call knowledge and data discovery management system. The support from DBMSs for Data Mining is manifold and the main parts that data miners are concerned with are extended query languages, the database including the data dictionary itself and multi-databases.

Incorporating existing Data Mining algorithms, no matter what methodology is used and what paradigm it is based on, requires the extension of existing database languages for querying and manipulating information. An extension of the de-facto query language SQL has been suggested by [MEO96] to express knowledge discovery queries including threshold values. The two main problems of query language extensions is the variety of Data Mining algorithms, most of which require different parameters, and design limitations of such languages, such as the lack of expressiveness of SQL. Agrawal et al. [AGRA96a] give an example how to couple their apriori association algorithm to a pseudo SQL host language.

To handle discovered knowledge, the DBMS has to provide facilities to store and retrieve rules, patterns, associations, etc, which requires the extension of central parts. Firstly, the underlying data dictionary has to support such knowledge, no matter what structure is has. Secondly, the embedded query language (and possibly existing programming language bindings) have to be enhanced to query and manipulate knowledge, and thirdly, new indexing techniques have to be developed to guarantee acceptable performance. An example of a two-level architecture is given in [KERS95]. Further, if a report generator is part of the DBMS, this should be extended to be able to handle discovered knowledge, such as rules, decision trees, associations, etc.

Multi-databases, i.e. distributed and heterogeneous data sources provide supplementary information which can be harnessed in Data Mining. In addition to partitioned information itself, for example customer data, product data and order data, schematic meta data can be utilised. Useful meta data are primary and foreign keys which express dependencies among entities as well as integrity constraints and semantic equivalence operations which can be used as domain knowledge.

1.7.3 Data Warehousing

On-line Transaction Processing (OLTP) systems are inherently inappropriate for decision support querying and, therefore, the need for Data Warehousing. Data warehouses [INMO92] are a new breed of databases that are optimised for use in decision support [REDB95]. They provide efficient access to corporate wide data in a format that is understandable to decision makers using meta-knowledge built into the data loading algorithm that loads data from various legacy systems into the Data warehouse.

Data warehouses principally, integrate “legacy” systems within a corporation to provide an enterprise wide view to decision making. This technology has become necessary due to the realisation on the part of large organisations that decisions about one business process cannot be made in complete isolation of other business processes within the enterprise. Also, most large corporations have operational data in production systems that is unreliable and disparate, making it difficult to integrate or extract for analysis purposes. Thus, the implementation of a Data Warehouse consists of the acquisition of data from multiple internal and external sources (of the corporation), the cleansing of this data, the management and integration into a central, integrated repository, the automatic updating of summary information in the warehouse based on the new data, the provision of access, reporting and analysis tools to interpret selected data converting it into information to support managerial decision making processes.

Table 4: OLTP vs. Data Warehouses

OLTP Systems	Data Warehouses
Data Source generally Human Users through Data Entry Screens	Data Source normally Legacy Systems requiring automatic loading algorithms
Pre-defined Queries	Ad-hoc Queries
Optimised Retrieval paths	Complex Retrievals
Small Query Results	Large Query Results
Frequent Updates	Batch, Infrequent Updates

Yet another reason for not being able to use OLTP databases for such decision support activities is that queries for decision support are normally complex and could span over 25 or more tables within an OLTP database that have been created with a view to optimising the performance of the OLTP operations and to reduce the amount of redundant information in the database. Joining these tables and producing a result would be very inefficient as join operations are computationally the most expensive database operation especially when they are created in ad-hoc queries not optimised by adding indexes to the tables accessed. Also, most OLTP databases are being used to their limits and can, therefore, not support such computationally expensive queries. Historical data is normally stored in backup storage devices and cannot be accessed directly. Such data though not important for OLTP applications are essential for trend analysis. Data stored in OLTP applications are not stored in formats that are understandable to the decision makers and would need a data expert to convert it into an understandable format. During the loading of data from OLTP systems to the Data Warehouse, data is reformatted to make it available in terms of business concepts.

Table 4 summarises the differences between OLTP and Data Warehouses in terms of their usage. It is clear from this that Data Warehouses require special methods for joins and data retrieval e.g. STAR indexes and joins [REDB95], data loading and new query tools supporting common decision support operations.

Data Warehouses are normally either employed in the role of a provider of business relevant information to managers and analysts or in a “closed-loop” system performing information driven functions such as intelligent inventory reordering.

From the point of view of Data Mining, a data warehouse can greatly reduce the effort required at the data pre-processing stage of Data Mining [INMO96] as it provides the data miner with integrated, clean, historical data in a business oriented manner. Along with the data, the data warehouse contains metadata that provides the data miner with the context of the data within the warehouse which is very useful during data prospecting and domain knowledge elicitation.

1.8 The Data Mining Process

Data Mining is recognised to be a process, rather than a stand alone automated algorithm that discovers knowledge from data without human intervention. While such a system would clearly be ideal, it is far from possible using present Data Mining techniques. In this section we describe our view of the Data Mining process (also known as the KDD process).

The beginning of the Data Mining process is the identification of a problem requiring IT support for decision making. The process that follows is comprised of a number of components beginning with the identification of the human resources required to carry out the Data Mining process.

1.8.1 Human Resource Identification

After a problem has been identified at the management level of an organisation, ***Human Resource Identification*** is the first stage of the Data Mining process. In most real-world Data Mining problems the human resources required are: the domain expert, the data expert and the data mining expert. Normally, Data Mining is carried out in large organisations where the prospect of finding a domain expert who is also an expert in the data stored by the organisation is rare. For example, in a large bank, the domain expert would belong to the sales department while the data expert will probably belong to the IT department. The data mining expert would normally belong to an organisation outside the bank deployed by the bank for the purpose of delivering the Data Mining goal. The synergy of these human resources as early as possible within any Data Mining project is imperative to the success of the project.

1.8.2 Problem Specification

Problem Specification is the second stage of the process. Here a better understanding of the problem is developed by the human resources identified in the ***Human Resource Identification*** stage of the process. The problem is decomposed into sub-problems and those tasks that can be solved using a Data Mining approach are identified. We refer to these sub-problems as Data Mining Tasks (DMT). Each DMT is now associated with a particular Data Mining goal (see Section 0).

The second part of the problem specification stage is to identify the ultimate user of the knowledge. Clearly if the knowledge discovered is to be used by a human, it must be in a format that the user can understand and is used to. However, if Data Mining is only a small part of a larger project and the output from knowledge discovery is to be interpreted by a computerised system, the format of the discovered knowledge will have to strictly adhere to the expected format.

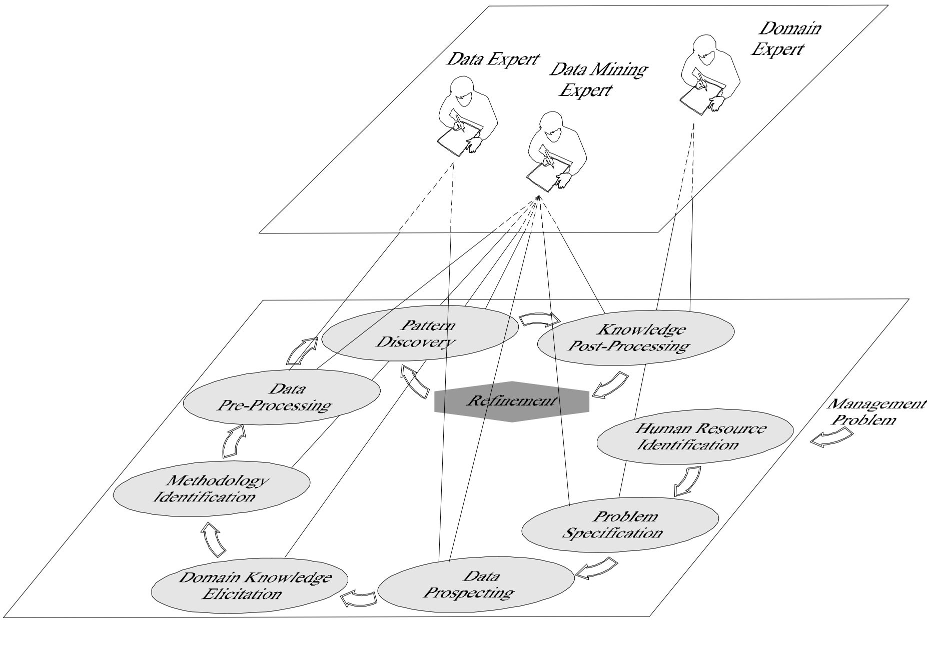

Figure 1. 7: The Data Mining Process

It is critical for the Data Mining expert to clarify in his/her mind what type of data mining task the solution would require and for that reason it is imperative that he/she gains a clear understanding of the problem at hand.

1.8.3 Data Prospecting

Data Prospecting is the next stage in the process. It consists of analysing the state of the data required for solving the problem at hand. There are four main considerations within this stage: What are the relevant attributes? Is the data required electronically stored and accessible? Are the data attributes required populated? Is the data distributed, heterogeneous, stored in legacy systems or is it centrally stored? If heterogeneous, are there any semantic inconsistencies that can be accounted for by the data expert or do they need to be discovered before the data can be used for discovering knowledge for decision support? For example, three computerised systems within a hospital in Northern Ireland: the Patient Administration System, the Laboratory System and the Radiology System contain different aspects of a patients previous medical history but they do not contain a common key with which this information can be aggregated. Clearly, such inconsistencies must be removed if Data Mining is to be performed on the patient data to discover health care patterns. It is very important to include within the data set all data attributes that could be related to the problem at hand and not just those that are "relevant" according to the domain expert as the domain expert may be using biases that are not entirely justified. Only the data attributes that have obvious values associated with the problem at hand should be removed from the discovery data set. For example, if we want to discover characteristics of customers of an insurance product that is aimed at house owners there is no point in including the data attribute 'House Status' as it will always have the value 'Owner'. During this stage the Data Mining expert, having gained a clear understanding of the problem in the previous stage, and the data expert work closely to map the problem onto the data sources modelling the business.

1.8.4 Domain Knowledge Elicitation

The next stage is that of *Domain Knowledge Elicitation*. During this stage of the Data Mining process, having identified the data that is relevant to the problem being tackled, the Data Mining expert now attempts to elicit any domain knowledge that the domain expert may be interested in incorporating into the discovery process. The domain knowledge may take the form of domain specific constraints on the search space as well as hierarchical generalisations defined on the various attributes indentified during data prospecting. The domain knowledge must be verified for consistency before proceeding to the next stage of the process.

Another form of domain knowledge utilised in Inductive Logic Programming [DZER96] is described in the form of relations. ILP allows other relations within the database that are relevant to the problem at hand to be incorporated into the discovery in the form of background knowledge.

1.8.5 Methodology Identification

The main task of the *Methodology Identification* stage is to find the best Data Mining methodology to solve the specified mining problem. Often a combination of paradigms is required to solve the problem at hand. For example, clustering or data partitioning may be required before the application of a classification algorithm. The most commonly used paradigms are described in Section 0. The chosen paradigm depends on the type of information that is required, the state of the available data (accessed at the Data Prospecting stage), the problem at hand and the domain of knowledge being discovered. For example, if an explanation of the discovered knowledge is required neural networks would clearly not be an appropriate methodology. The selected technique may influence the format of the input data, whose preparation is part of the following knowledge discovery step. For example, when using neural networks, data transformation may be required to map input data into the interval [0, 1] or when Association rule induction is used, the data may need to be discretised or converted into a binary format depending on the association algorithm used. The success of this stage mainly depends on the experience and expertise of the Data Mining expert.

1.8.6 Data Pre-processing

Depending on the state of the data this stage of the process may or may not constitute the stage where most of the effort of the knowledge discovery process is concentrated. Data pre-processing involves removing outliers in the data, predicting and filling-in missing values, noise modelling, data dimensionality reduction, data quantisation, transformation, coding and heterogeneity resolution. Outliers and noise in the data can skew the learning process and result in less accurate knowledge being discovered. They must be dealt with before discovery is carried out. Missing values in the data must either be filled in or a paradigm used that can take them into account during the discovery process so as to account for the incompleteness of the data model. Data dimensionality reduction is an important aid to improve the efficiency of the discovery algorithm as most of these have execution times that increase exponentially with respect to the number of attributes within the data set. Depending on the paradigm chosen the data may need to be coded or discretised.

Data pre-processing technologies can consist of quite a variety of tools, such as exploratory data analysis and thresholding for removal of outliers, interactive graphics for data selection, principal component analysis, factor analysis or feature subset selection [KOHA95] for data dimensionality reduction, statistical models for handling noise in the data [HICK96], techniques for filling in missing values [LITT87, QUIN86], information theoretic measures for data discretisation, linear or non-linear transformation of data [BIGU96] and semantic equivalence relationship handling for solving heterogeneity conflicts.

1.8.7 Pattern Discovery

The *Pattern Discovery*[1] stage follows the data pre-processing stage. It consists of using algorithms that automatically discover patterns from the pre-processed data. The choice of algorithm depends on the Data Mining goal at hand. Due to the large amounts of data from which knowledge is to be discovered, the algorithms used in this stage must be efficient. In our experience, it is better that the DMT is not totally automated and independent of user intervention. The domain expert can often provide domain knowledge that can be used by the discovery algorithm for making patterns in the data more visible, pruning of the search space or for filtering the discovered knowledge based on a user driven interestingness measure.

Different paradigms require different parameters to be set by the user. Example parameters are number of hidden layers, number of nodes per layer and various learning parameters like learning rate and error tolerance for Neural Networks, population size, mutation and cross-over probabilities for Genetic Algorithms, membership functions in Fuzzy systems, Support and Confidence thresholds in Association algorithms and so on. Tuning these parameters is normally an iterative process and forms part of the refinement process within Data Mining (see Section 0).

1.8.8 Knowledge Post-processing

The last stage of the Data Mining process is *Knowledge Post-processing*. Trivial and obsolete information must be filtered out and discovered knowledge must be presented in a user-readable way, using either visualisation techniques or natural language constructs. Often the knowledge filtering process is domain as well as user dependent. The most common way to filter knowledge discovered is to rank the knowledge and threshold based on the ranking. The ranking is often based on support, uncertainty and interestingness measures of the knowledge.

Gebhardt [GEBH94] formalised interestingness by providing four facets of interestingness: the subject field under-consideration, the conspicuousness of a finding, the novelty of the finding and the deviation from prior knowledge. In general, measures of interestingness can be classified into Objective Measures and Subjective Measures. An Objective Measure depends on the structure of the pattern and the underlying data used e.g. J-Measure [SMYT91] and Piatetsky-Shapiro’s Measure [PIAT91]. Subjective Measures depend on the class of the users who examine the patterns. These are based on two concepts [SILB95]:

[1] This stage is also referred to as Model Development [BRAC96] and Data Mining [FAYY96]. We, however, use the term Pattern Discovery instead of Data Mining to refer to this stage to avoid confusion with the use of Data Mining as the overall process in industry.

Unexpectedness (a pattern is interesting if it is unexpected) and Actionability (a pattern is interesting if the user can do something with it to his or her advantage e.g. KEFIR) of the pattern.

A measure of interstingness can be used as an interestingness filter where all patterns are discovered and those that are interesting are presented to the user. Alternatively, it may be used as an interestingness engine that focuses the search for interesting patterns, dynamically pruning the less interesting facts.

Another aspect of Knowledge Post-processing is knowledge validation before it can be used for critical decision support. The most commonly used techniques here are holdout sampling, random resampling, n-fold cross-validation [KOHA95a] and bootstrapping [EFRO93].

1.8.9 Knowledge Maintenance

Due to the fact that the data used as input to the knowledge discovery process is often dynamic and prone to updates, the discovered knowledge has to be maintained. *Knowledge Maintenance* may consist of re-applying the already set up Data Mining process for the particular problem or using an incremental methodology that would update the knowledge as the data changes keeping them consistent.

1.8.10 The Refinement Process

It is an accepted fact that the Data Mining process is iterative. After the Knowledge Post-processing stage, the knowledge discovered is examined by the domain expert and the data mining expert. This examination of the knowledge may lead to the *Refinement Process* of Data Mining. During the refinement process the domain knowledge as well as the actual goal posts of the discovery may be refined. Refinement could take the form of redefining the data used in the discovery, a change in the methodology used, the user defining additional constraints on the mining algorithm, refinement of the domain knowledge used or refinement of the parameters of the mining algorithm. Once the refinement is complete the pattern discovery stage and the knowledge post-processing stages are repeated. Note that the Refinement Process is not a stage of the Data Mining Process. Instead in constitutes the iterative aspects of the Data Mining Process and may make use of the initial stages of the Data Mining Process i.e. Data Prospecting, Methodology Identification, Domain Knowledge Elicitation and Data Pre-processing.

1.9 Applications of Data Mining

We now give a representative list of applications in which Data Mining has been utilised successfully, and describe the overall technological challenges faced during the development of these applications. The taxonomy is not exhaustive; Data Mining has also been applied to other disciplines, such as administration, geology, geography, linguistics, telecommunication, meteorology, biochemistry, etc.

1.9.1 Manufacturing

Most modern industrial processes are subject to technological control and monitoring, during which vast quantities of manufacturing data are generated. The main industrial areas in which Data Mining has been applied are process and quality control, process analysis, supply forecasting, machine maintenance and fault diagnosis.

Texas Instruments has isolated faults during semiconductor manufacturing using automated discovery from wafer tracking databases [SAXE93]. Firstly, associations (called classes of queries) are generated which are based on prior wafer grinding and polishing data. These classes have the potential to identify interrelationships among processing steps, which can isolate faults during the manufacturing process. Secondly, domain filters are incorporated to minimise the search space of the discovered associations. Thirdly, the interestingness evaluator tries to detect outliers, clusters (using the minimum description length) and trends (using Kendall’s t-coefficient) which are feed back to the query generator. Lastly, another domain filter has been implemented to set interestingness thresholds, before finally a list of detected patterns is output.

Apte et al. facilitated five classification methods (k-nearest neighbour, linear discriminant analysis, decision trees, neural networks and rule induction) to predict defects in hard drive manufacturing [APTE93]. Error rates at a critical step of the manufacturing process were used as input to identify knowledge (classes fail or pass) for further assistance of engineers. In the particular environment, none of the methods achieved outstanding results, but rule induction was superior, in order to minimise the high dimensionality of given data and thus, to improve the performance of the manufacturing quality control bottleneck.

1.9.2 Finance

Financial organisations such as banks, insurance companies, building societies, etc., are at a crucial period in their development. The competition among financial institutions is becoming harder and new services are being introduced frequently. Due to this variety of services, better informed customers, and international as well as political aspects of financial factors, prediction of market trends is becoming more complex.

IBM has used modified rule induction and classification techniques to predict equity returns [APTE96] and to select investment cocktails [JOHN96]. Both systems have a similar objective, in that they try to predict portfolios of stocks with exceptional return for the near future. The first approach is based on minimal rule generation and contextual feature analysis, whereas the latter project uses the Recon Data Mining system. Both systems achieved encouraging results with respect to established financial indices.

Additionally, fraud detection and prevention is becoming a topic of high interest with a view to protect both customers as well as the financial institution from fraud. Data Mining has been applied successfully in this area. For instance, MasterCard International and Los Alamos National Laboratory are working together to develop a computer model to predict fraud. They have evaluated more than 30 different modelling techniques based on neural networks, fuzzy logic, and genetic algorithms.

1.9.3 Retail

The two major customer concerns for business organisations are the behaviour of existing clientele and to identify potential new customers. The first area usually encompasses market basket analysis and cross-sales while the latter task deals with the identification of characteristics and behaviours of potential new clients.

Data Mining at WalMart, a large shopping chain in the United States, produced one of the most cited rules discovered by Data Mining, which showed that male customers who buy nappies on a Friday afternoon or evening are likely to buy a six pack of beer, too. The objective of the Data Mining exercise was to find associations across products for rearranging goods in shelves and to gain more information to modify the “valued-customer” schemes.

Cross-Sales is the term given to the problem of attempting to sell a product to existing customers of an organisation who are not already customers for that particular product. Anand et al. tackled the problem, using algorithms to discover characteristics rules [ANAN97]. Classification algorithms are not suitable, because no negative data is available. An association algorithm based on Evidence Theory, which is a generalisation of earlier association algorithms and therefore allows the incorporation of support and uncertainty thresholds and syntactic constraints, has been used to discover characteristic rules and to detect deviations. Additionally, for refinement purposes, domain knowledge has been incorporated to discover more accurate rules.

1.9.4 Medicine

Experts systems have been established in the medical area from the early days, and thus it is not surprising that the medical sector is a suitable test bed for Data Mining technologies. The two main medical areas Data Mining can be applied to are diagnosis and treatment. Diagnosis is mainly concerned with classifying a patient into one (or more) possible disease class(es). Treatment of a disease might involve various medications, which have negative effects on each other. Such inter-relationships among drugs can be discovered using association algorithms.

To predict risks of gastro-oesophageal cancer patients White et. al. applied classification techniques to medical records [WHIT96]. Logistic regression and tree-based classification has been utilised on epidemiological and clinical variables. The results showed that the tree-based classification using the developed algorithm PREDICTOR was better at discriminating cancers than logistic regression. This performance was validated using cross-validation techniques.

1.9.5 Science

1.9.5.1 Astronomy

Today the amount of data confronting astronomy and space scientists is enormous, and manual browsing – let alone interpreting – is beyond human capabilities, e.g. it has been estimated that manually locating all 10^6 visible volcanoes on Venus scattered throughout 30000 images would take a planetary geologist about 10 man-years [SMYT96]. The major objective is the automated classification and detection of new astronomical sky objects, i.e. patterns in images.

A well known project in the area of astronomy is the SKICAT system [FAYY96] which aims to classify and analyse an expected $2 * 10^9$ sky objects collected at an extensive sky survey. Due to the vast amounts of data (about 3 Terabyte), special techniques for data management, image processing and Data Mining were required. Supervised image classification has been carried out using decision tree induction, using the GID3* and O-BTREE algorithms. To generate better results, i.e. to provide optimised rules, multiple tree induction was implemented through the RULER learning system. Also, in addition to existing base-level attributes, feature extraction was applied, facilitating attribute normalisation as well as scale and fraction scaling. The results showed that the classification system RULER, which managed to identify about 94% of sky objects per frame, was superior to traditional decision tree algorithms such as ID3, GID3* and O-Btree.

1.9.5.2 Molecular Biology

The goal of the Human Genome Project is to identify, map and sequence all human genes. Large information generated from this project is being stored in databases. Simply generating and storing this information is not an end in itself. Methods are needed for scientific discovery that uses this mass of data and generates new information that can be useful for our understanding of the human genome.

Once the primary base sequences of a gene have been determined, it is a relatively easy matter to determine the amino-acid sequence of the encoded polypeptide. Databases of these sequences such as GenBank, EMBL and PIR currently contain in excess of 80,000 entries. Searching these databases can provide important information on the role of homologous genes in distantly related organisms [DONN94]. In contrast the Brookhaven Protein Data Bank (PDB) which contains protein 3D structures derived from x-ray crystallo-graphic and nmr analyses contains only 2000 chains. Our ability to determine the primary structure of a polypeptide far outstrips our ability to determine the 3D structure it will adopt. The Data Mining tool PEBLS was applied to the prediction of protein secondary structure and to the identification of DNA promoter sequences [COST93]. Nearest neighbour methods have been applied to classify DNA sequences.

1.9.5.3 Climate

The challenge of predicting long-term climate changes and short-term weather forecasts has been tackled by many different scientific disciplines, but was never solved in a satisfactory manner. Due to the complexity of data – qualitative as well as quantitative – a complete model is hardly realisable. Spatio-temporal data, on which climate forecasts are mainly based, is hard to analyse automatically, and different local phenomena make designing a generic solution difficult.

One of the major objectives of the Oasis projects at UCLA and JPL [STOL96] is to provide a platform for storing and accessing heterogeneous spatio-temporal geophysical data to track atmospheric constructs, such as cyclones, mete fronts or blocking events. Modified supervised sequential pattern extraction algorithms have been applied to discover such phenomena, and principal component analysis and hierarchical cluster analysis was used to minimise the search space.

1.10 Organisation of the Book

This book attempts to provide the reader with a comprehensive text on current techniques employed in the various stages of the Data Mining/ Knowledge Discovery in Databases process as outlined in Section 0. The book is divided into four parts - Data Pre-processing, Data Mining Paradigms, The Role of the Human and Knowledge Post-processing. These are the four main areas within the process that require technological support.

Part 1 of the book is on Data Pre-processing and consists of three chapters. The first chapter (Chapter 2) covers techniques for handling missing data. Relational databases provided a representation for missing data in the form of NULL values. Though this is an acceptable situation from a Database Management Systems point of view it is clearly not enough to have an explicit representation of missing information when reasoning about the real-world process that it models. Techniques used to handle missing information can be divided into two main approaches. The first, and more widely used, approach uses various techniques of varying complexity to fill-in the missing data before discovery is undertaken. The second approach is to represent the missing information explicitly as ignorance and deal with it during the discovery itself. Chapter 2 surveys techniques used in the first approach while Chapter 11 discusses the second approach as part of an overall Data Mining framework that uses Evidence Theory as the underlying uncertainty model.

Chapter 3 addresses the need for data-dimensionality reduction techniques within Data Mining. While most Discovery techniques used in Data Mining can achieve linear time complexity with respect to the number of tuples in the data set, they are generally exponential with respect to the number of attributes in the data set. Thus, it is useful to reduce the dimensionality of the data set, without loosing any valuable information, prior to discovery. Another reason for reducing dimensionality is results from machine learning research that shows that the performance of learning algorithms suffers in the presence of irrelevant data. This chapter covers the area of Data-dimensionality reduction from two different viewpoints - Machine Learning, filter as well as wrapper methods, and Statistics.

Chapter 4 addresses the area of Noise Modelling. This again, is an important aspect within Data Mining as noise is inevitable in real-world databases. The chapter discusses statistical techniques used to help discovery algorithms in coping with inaccurate data.

Part 2 of the book is on Data Mining Paradigms. The Data Mining Paradigms are further classified into Machine Leaning Based Techniques, Uncertainty Based Techniques, Database Support for Data Mining and Statistical Support. Within Machine Learning based techniques the book consists of four chapters. The first of these chapters (Chapter 5) discusses in detail the use of Information Theory in Rule Induction. Chapter 6 surveys techniques in Conceptual Clustering while Chapter 7 discusses the use of Heuristic Techniques specifically Genetic Algorithms, Simulated Annealing and Hybrid Systems within Data Mining. Chapter 8 discusses the use of Neural Networks in Data Mining. Neural Networks have been criticised in Data Mining literature on two accounts. Firstly, Data Mining is considered as a “black box” technology and patterns learned by the Neural Network during training are not easily understood. Secondly, when using Neural Networks, using background knowledge is difficult. This chapter attempts to address these criticisms by surveying recent developments in rule extraction from Neural Networks as well as the incorporation of background knowledge into the training of Neural Networks.

Uncertainty Based Techniques are discussed in Chapters 9 through to 12. Chapter 9 discusses various techniques for the discovery of Cause-Effect Networks from Data and identifies challenges of this novel approach to Data Mining. Chapter 10 first describes the basic concepts of the Rough Set approach to Data Mining and then discusses two techniques for obtaining a logic of rough sets. The chapter then describes techniques for discovering dependencies from data and introduces statistical procedures for evaluating the validity of the predictions based on the discovered dependencies. Chapter 11 describes the benefits of using Evidence Theory as the underlying uncertainty model for Data Mining while Chapter 12 describes various approaches to Data Mining based on Fuzzy Set Theory.

Chapters 13 through to 16 discuss Database Support for Data Mining. Chapter 13 discusses Set-Oriented Approaches to Data Mining which are the most widely used technology for discovery of Association rules. The chapter discusses the general approach as well as the use of Database Management System facilities such as indexes, joins and query optimisation within the implementation of these algorithms. Chapter 14 discusses database support for discovery of temporal relationships while Chapter 15 discusses Database Support for Attribute Oriented Induction. Chapter 16 addresses the area of discovery in Distributed and Heterogeneous Databases. While most data in the real-world is distributed, first generation Data Mining systems tend to ignore this fact and work with data stored in a single relation. This chapter discusses techniques that have been developed to address the problems that are introduced when attempting to discover knowledge from distributed and heterogeneous databases.

Chapters 17 through to 20 discuss Statistical support for Data Mining. Chapter 17 may be considered as a companion chapter to Chapter 16 as it discusses the use of local Statistical Databases as Data Mining primitives when discovering knowledge from distributed and heterogeneous databases. Chapter 18 discusses Log-linear modelling, a statistical approach specifically designed for analysis of nominal data types. Chapter 19 discusses statistical approaches to discovering classification rules while Chapter 20 discusses the use of sampling in Data Mining.

Chapter 21 constitutes the third part of the book that discusses the role of the Human in the Data Mining process. Specifically it describes how domain knowledge and biases can be incorporated into the discovery process and what the different benefits of using such information are.

The fourth and final part of the book concentrates on the aspect of Knowledge Post-processing. The first of the two chapters in this part (Chapter 22) discusses various approaches to Knowledge Filtering. Approaches discussed here vary from simple thresholding of measures associated with the discovered knowledge to more complex user models of interesting knowledge. Chapter 23 surveys various approaches to validating the discovered knowledge before use in decision making.

The final chapter in the book (Chapter 24), describes future enhancements and directions that the authors foresee in the field of Data Mining. This chapter hopes to provide directionality to the research community as well as provide motivation to newcomers in the field.

1.11 References

[AGRA92] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, A. Swami. An interval classifier for database mining applications, in *Proc. 18th Int. Conf. on VLDB*, pp. 560-573, 1992.

[AGRA93] R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases, in *Proc. of the ACM SIGMOD Conf. on Management of Data*, pp. 207-216, 1993.

[AGRA93a] R. Agrawal, T. Imielinski, A. Swami. Database mining: A performance perspective, in *IEEE Trans. on Knowledge and Data Engineering*, 5(6):914-925, 1993.

[AGRA94] R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases, in *Proc. 20th Int. Conf. on VLDB*, pp. 487 - 499, 1994.

[AGRA95] R. Agrawal, K. Lin, H. S. Sawhney, K. Shim. Fast Similarity Search in the Presence of Noise, Scaling and Translation in Time-Series Databases, in *Proc. 21st Int. Conf. on VLDB*, pp. 490-501, 1995.

[AGRA95a] R. Agrawal, R. Srikant. Mining Sequential Patterns, in *Proc. Int. Conf. on Data Engineering (ICDE)*, Taipei, Taiwan, March 1995.

[AGRA96] R. Agrawal, J.C. Shafer. Parallel Mining of Association Rules: Design, Implementation and Experience, in IEEE Trans. on Knowledge and Data Engineering, 8(6):962–969, 1996.

[AGRA96a] R. Agrawal, K. Shim. Developing Tightly-Coupled Data Mining Applications on a Relational Database System, in *Proc. 2nd Int. Conf. on Knowledge Discovery in Databases and Data Mining*, pp. 287-290, 1996.

[ANAN95] S.S. Anand, D.A. Bell and J.G. Hughes. Evidential Techniques for Parallel Database Mining, High Performance Computing and Networking, in *Lecture Notes in Computer Science*, Springer Verlag, pp. 190-195, 1995.

[ANAN96] S.S. Anand, D.A. Bell and J.G. Hughes. A General Framework for Data Mining Based on Evidence Theory, in Data and Knowledge Engineering Journal, 18:189–223, 1996.

[ANAN97] S.S. Anand, J.G. Hughes, D.A. Bell, A. Patrick. Tackling the Cross-Sales Problem using Data Mining, in Proc. 1st Pacific-Asia Conf. on Knowledge Discovery and Data Mining, pp. 331-343, 1997.

[ANAN97a] S.S. Anand, D.A. Bell and J.G. Hughes. The Evidence-based Association Rule Algorithm, *Internal Report*, Faculty of Informatics, University of Ulster, 1997.

[ANAN97b] S. S. Anand, W. Dubitzky, D. Patterson, A. Schuster, J.G. Hughes. M^2: A First Step Towards Automated Generation and Updating of Case-Knowledge from Databases, submitted for publication to ICCBR'97, 1997.

[APTE93] C. Apté, S. Weiss, G. Grout. Predicting Defects in Disk Drive Manufacturing: A Case Study in High-Dimensional Classification, in *Proc. 9th Conf. Artificial Intelligence on Applications*, pp. 212-218, 1993.

[APTE96] C. AptéP, S.J. Hong. Predicting Equity Returns from Securities Data, in *[FAYY96b]*, pp. 541-560, 1996.

[BELL93] D.A. Bell. From Data Properties to Evidence, in *IEEE Transactions on Knowledge and Data Engineering*, 5(6), pp. 965-969, 1993.

[BENC90] T.J.M. Bench-Capon. Knowledge Representation: An Approach to Artificial Intelligence, The APIC Series 32, Academic Press, 1990.

[BERN96] D.J. Berndt, J. Clifford. Finding Patterns in Time Series: A Dynamic Programming Approach, in *[FAYY96b]*, pp. 229-248, 1996.

[BIGU96] J.P. Bigus, Data Mining with Neural Networks, McGraw-Hill, 1996.

[BRAC96] R.J. Brachman, T. Anand. The Process of Knowledge Discovery in Databases: A Human-Centred Approach, in *[FAYY96b]*, pp. 37-58, 1996.

[BRIE90] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone. Classification and Regression Trees, Chapman and Hall, 1990.

[BÜCH96] A.G. Büchner, S.S. Anand, D.A. Bell, J.G. Hughes. A Framework for Discovering Knowledge from Distributed and Heterogeneous Databases, in *IEE Colloquium on Knowledge Discovery in Databases*, pp. 8/1-8/4, 1996.

[BÜCH97] A. G. Büchner, B. Yang, S. Ram, D. A. Bell, J. G. Hughes. A Holistic Architecture for Knowledge Discovery in Multi-Database Environments, submitted for publication to DMKD'97, 1997.

[BUYT95] F. A. Buytendijk, OLAP: Playing for keeps, 1995.

[CAI91] Y. Cai, N. Cerecone, J. Han, Attribute-Oriented Induction in Relational Databases, in *[PIAT91a]*, pp. 213-228, 1991.

[CHAN91] K.C.C. Chan, A.K.C. Wong. A Statistical Technique for Extracting Classificatory Knowledge from Databases, in *[PIAT91a]*, pp. 107-124, 1991.

[CHAN96] P. K. Chan, S. J. Stolfo. Sharing Learned Models among Remote Database Partitions by Local Meta-learning, in Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, pp. 2–7, 1996.

[CHEE96] P. Cheesman, J. Stutz. Bayesian Classification (AutoClass): Theory and Results, in *[FAYY96b]*, pp. 153-180, 1996.

[CHEU96] D.W. Cheung, V.T. Ng, A.W. Fu, Y. Fu: Efficient Mining of Association Rules in Distributed Databases, in *IEEE Trans. on Knowledge and Data Engineering*, 8(6):911-922, 1996.

[CODD93] E.F. Codd, S.B. Codd, C.T. Salley, Providing OLAP to User-Analysts: An IT Mandate, White Paper produced by Codd and Date Inc., 1993.

[CONS96] Conspectus. Data Warehousing and Decision Support Environment, February, 1996.

[COST93] S. Cost and S. Salzberg. A Weighted Nearest Neighbour Algorithm for Learning with Symbolic Features, in *Machine Learning*, 10(1):57-78, 1993.

[DONN94] E. Donnelly, Y.A. Barnett, W. McCullough. Germinating conidiospores of Aspergillus amino acid auxotrophs are hypersensitive to heat shock, oxidative stress and DNA damage. in *FEBS Letters*, 355:201-204, 1994.

[DOUG95] A. Douglas, A. Wood, A. Rowstron. Linda Implementation Revisited, in *Transputer and Occam Developments*, P. Nixon (Ed.), pp. 125–138, 1995.

[DRES93] Dresner, OLAP: Heightened Industry Focus on Business Intelligence, Gartner Group, October 4, 1993

[DZER96] S. Dzeroski. Inductive Logic Programming and Knowledge Discovery in Databases, in *[FAYY96b]*, pp. 117-152, 1996.

[EFRO93] B. Efron, R.J. Tibshirani. An Introduction to the Bootstrap, Chapman and Hall, 1993.

[FAYY96] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From Data Mining to Knowledge Discovery in *[FAYY96b]*, pp. 471-493, 1996.

[FAYY96a] U.M. Fayyad, S.G. Djorgovski, N. Weir, Automating the Analysis and Cataloging of Sky Surveys, in *[FAYY96b]*, pp. 471-493, 1996.

[FAYY96b] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.). Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996.

[FRAW91] W. J. Frawley, G. Piatetsky-Shapiro, C. J. Matheus, Knowledge Discovery in Databases : An Overview, in *[PIAT91a]*, pp. 1-27, 1991.

[GAIN96] B.R. Gaines, Transforming Rules and Trees into Comprehensible Knowledge Structures, in *[FAYY96b]*, pp. 205-228, 1996.

[GEBH94] F. Gebhardt. Discovering interesting statements from a database, in *Applied Stochastic Models and Data Analysis*, 10:1-14, 1994.

[GEIS94] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam. PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing, MIT Press, 1994.

[HAN94] J. Han, Towards Efficient Induction Mechanisms in Database Systems, in *Theoretical Computer Science 133*, Elsevier, pp. 361-385, 1994.

[HAN96] J. Han, Y. Fu. Exploration of the Power of Attribute-Oriented Induction in Data Mining, in *[FAYY96b]*, pp. 399 -424, 1996.

[HECK95] D. Heckerman, A. Mamdani, M. P. Wellman. Guest Editors of Communications of the ACM, Special Issue on Real-World Applications of Bayesian Networks, 38(3), March, 1995

[HECK96] D. Heckerman. Bayesian Networks for Knowledge Discovery, in *[FAYY96b]*, pp. 471-493, 1996.

[HICK96] R.J. Hickey. Noise Modelling and Evaluating Learning from Examples, in *Artificial Intelligence*, 80, 1996.

[HOLL75] J. H. Holland. Adaptation in Natural and Artificial Systems, Univ. of Michigan Press, Ann Arbor, Michigan, 1975.

[HUGH91] J.G. Hughes. Object-Oriented Databases, Prentice Hall, 1991.

[IMIE96] T. Imielinski, H. Mannila. A Database Perspective on Knowledge Discovery, in *Communications of the ACM*, 39(11):58-64, 1996.

[INMO92] W.H. Inmon, Data Warehouse, Wellesley, 1992.

[INMO96] W.H. Inmon, The Data Warehouse and Data Mining, in *Communications of the ACM*, 39(11): 49-50, 1996.

[JOHN96] G.H. John, P. Miller, R. Kerber, Stock Selection Using Rule Induction, in *IEEE Expert: Intelligent Systems & their Applications*, 11(5):52-58, 1996.

[KEIM96] D.A. Keim, H.-P. Kriegel. Visualization Techniques for Mining Large Databases: A Comparison, in *IEEE Trans. on Knowledge and Data Engineering*, 8(6):923-938, 1996.

[KERS95] M.L. Kersten, M. Holsheimer: On the symbiosis of a data mining environment and a DBMS, CWI Research Report CS-R9512, 1995.

[KOHA95] R. Kohavi, D. Sommerfield. Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. in *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining*, pp. 192-197, 1995.

[KOHA95a] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in *Proc. of IJCAI'95*, 1995.

[LANG95] P. Langley and H.A. Simon, Applications of Machine Learning and Rule Induction, in *Communications of the ACM*, 38(11), 1995.

[LITT87] R.J.A. Little, D.B. Ruthlin. Statistical Analysis with Missing Values, Wiley, 1987.

[MATH94] C.J. Matheus, G. Piatetsky-Shapiro, D. McNeill. An Application of KEFIR to the Analysis of Healthcare Information, in *AAAI Workshop on Knowledge Discovery in Databases*, pp. 441-452, 1994.

[MCCU43] W.S. McCulloch, W. Pitts. A Logical Calculus of Ideas Immanent in Nervous Activity. in *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[MEO96] R. Meo, G. Psaila, S. Ceri: A new SQL-like Operator for Mining Association Rules, in *Proc. 22nd Int. Conf. VLDB*, pp. 122-133, 1996.

[MICH83] R.S. Michalski. A Theory and Methodology of Inductive Learning, in *Machine Learning: An Artificial Intelligence Approach*, R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), pp. 83-134, 1983.

[MIYA90] S. Miyamoto. Fuzzy Sets in Information Retrieval and Cluster Analysis, Kluwer Academic Publications, Dordrecht, Boston, London, 1990.

[MUNA94] T. Munakata, Y. Jani. Fuzzy Systems: An Overview, in *Communications of the ACM*, 37(3): 69-76, 1994.

[PEDN91] E.P.D. Pendault. Minimum-Length Encoding and Inductive Inference, in *[PIAT91a]*, pp. 71-92, 1991.

[PIAT91] G. Piatetsky-Shapiro. Discovery, Analysis and Presentation of Strong Rules, in *[PIAT91a]*, pp. 229-248, 1991.

[PIAT91a] G. Piatetsky-Shapiro, W.J. Frawley (Eds.). Discovery in Databases, AAAI/ MIT Press, 1991.

[PIAT94] The Interestingness of Deviations, in *AAAI Workshop on Knowledge Discovery in Databases*, pp. 25 – 36, 1994.

[QUIN86] J. R. Quinlan, Induction of Decision Tree , in *Machine Learning*, 1:81-106, 1986.

[QUIN87] J. R. Quinlan. Simplifying Decision Trees, in Int. J. Man-Machine Studies, 27:221–234, 1987.

[REDB95] Red Brick Systems. A Red Brick White Paper, available from http://www.redbrick.com/rbs/whitepapers/datawh_wp.html.

[SAXE93] S. Saxena. Fault Isolation during Semiconductor Manufacturing using Automated Discovery from Wafer Tracking Databases, in *Proc. Knowledge Discovery in Databases Workshop*, pp. 81-88, 1993.

[SILB95] A. Silberschatz, A. Tuzhilin. On Subjective Measures of Interestingness in Knowledge Discovery, in Proc. of 1st Int. Conf. on Knowledge Discovery and Data Mining, pp. 275-281, 1995.

[SMYT91] P. Smyth and R.M. Goodman. Rule Induction Using Information Theory, in *[PIAT91a]*, pp. 159 - 176, 1991.

[SMYT96] P. Smyth, U.M. Fayyad, M.C. Burl, P. Perona. Modeling Subjective Uncertainty in Image Annotation, in *[FAYY96b]*, pp. 517-539, 1996.

[SRIN94] M. Srinivas, L. M. Patnaik, Genetic Algorithms: A Survey, in *IEEE Computer,* 27(6), 1994.

[STOL96] P. Stolorz, H. Nakamura, E. Mesrobian, R. Muntz, E. Shek, J.R. Santos, J. Yi, K. Ng, S.-Y. Chien, C. Mechoso, J. Farrara. Fast Spatio-Temporal Data Mining of Large Geophysical Datasets, in *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining*, pp. 300-305, 1996.

[WHIT96] A.P. White, M.T. Hallissey, L.W.L. Fielding, W.Z. Liu: A Comparison of Two Classification Techniques in Screening for Gastro-oesophageal Cancer, in *Applications and Innovations: Proc. Expert Systems '96*, pp. 83-97.

[YAGE91] R. Yager. On Linguistic Summaries of Data, in *[PIAT91a]*, pp. 347-366, 1991.

[ZADE65] L. A. Zadeh. Fuzzy Sets, in *Information and Control*, 8:338-353, 1965.

[ZIAR93] W. Ziarko, R. Golan, D. Edwards. An Application of Datalogic/R Knowledge Discovery Tool to Identify Strong Predictive Rules in Stock Market Data, in AAAI Workshop on Knowledge Discovery in Databases, pp. 89–101, 1993.

[ZIMM91] H. -J. Zimmermann. Fuzzy Set Theory - And Its Applications, Kluwer, 1991.