JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

KNOWLEDGE DISCOVERY IN DATABASES: A REVIEW AND ANALYSIS

[1]Senthil Kumar Ramadoss, [2]Puviarasan N

[1]Research Scholar, [2]Professor & Head
Department of Computer and Information Science,
Annamalai University, TN, INDIA

Abstract: ***This paper aims to review and analyze the various technical procedures being used to extract knowledge from databases. The challenges and advantages of different database knowledge discovery methods, particularly from very large databases that have relations naturally not explicit in database were analyzed. The machine learning techniques that are acting in the background of those methods were compared and analyzed. The efficiencies of various machine learning methods used for database knowledge discovery were analyzed and classified. In totality, the state of art in database knowledge discovery in several research areas, the progress and development of knowledge discovery models including multi-strategy discovery systems were reviewed, analyzed and their efficiencies were reported.***

Index Terms - Database knowledge Discovery, Machine Learning, Knowledge Discovery Models, Multi-strategic Knowledge Discovery Systems

I. INTRODUCTION

The need to take a look at the newly developing application of artificial intelligence in database knowledge discovery and database analysis in general firstly proposed during the first international conference on knowledge discovery and data mining [1], by The Holding [2] in his survey volume. The need was confirmed on finding that the *Journal of the American Society for Information Science* has called for papers for a special issue on the subject to be published in summer 1997.Large databases, both numerical and textual, contain vast amounts of data that are not explicitly displayed by the structure of the database. 'Knowledge discovery' has been defined as the 'extraction of implicit, previously unknown, and potentially useful information from data' [4].

The information extracted includes concepts, concept interrelations, classifications, decision rules, and other patterns of interest. In 1989, the total number of databases in the world was estimated at five million, though most of these were small. The falling costs of data storage and of processing power have encouraged organizations to accumulate great quantities of data. The recording of commercial and financial transactions generates data- bases now measured in gigabytes and terabytes [5].

One of the largest databases in the world has been created by the US retailer Wal-Mart, which handles over twenty million transactions a day; US census data amount to a million bytes; most health-care transactions in the US are being stored in multi-gigabyte databases; Mobil Oil is developing a 'data warehouse' of over 100 terabytes of data relating to oil exploration. Scientific databases are also rapidly growing: e.g. earth observation satellites can generate one terabyte of data every day; the Human Genome Project stores thousands of bytes of data for each of the several million genetic sites. It has been estimated that the total volume of recorded data in the world is now doubling every twenty months. There is a growing gap between data generation and data understanding. There is a need to develop advanced techniques of intelligent data analysis to make sense of these mountains of data.

A variety of methods of knowledge discovery and 'data mining' has been developed in recent years. Some examples of areas in which they have been applied are:

- analysis of point-of-sale data to aid understanding of consumer behavior;
- examination of medical records to understand health trends;
- automatic analysis and cataloguing of stellar surveys;
- analysis of subscriber databases of trade journals to provide editions tailored to particular customers;
- analysis of car defect reports to derive diagnostic expert systems;
- Detecting patterns in time series data.

Other areas that have been analyzed are: drug side effects, biological classification of river water quality, cancer diagnosis, protein structure, chemical structure identification, hospital cost containment, voting trends, fraud detection, population analysis, quality control, forest fire prevention, wind damage risk assessment, modeling of global climatic change.

In this paper some of the methods and applications of knowledge discovery will be discussed. The main concern is to examine inputs and outputs – what data are analyzed, and what knowledge is produced. The details of the intervening technical algorithms will not be explored – as stated in the subtitle; this is an introductory view of the subject. A helpful introduction to machine learning techniques is the book by Weiss and Kulikowski [6].

II. **BACKGROUND** The techniques of knowledge discovery stem from several decades of research into machine learning, pattern recognition, statistics and visualization techniques. The discovery of patterns in collections of observations has been a research topic of long-standing interest – it is the process of inductive learning. Inductive learning is the acquisition of knowledge by drawing inductive inferences from facts (provided either by examples or from the environment). The goal of inference is to formulate plausible general assertions that explain the given facts and are able to predict new facts. An omnipresent problem in science is to construct meaningful classifications of observed objects or situations, to facilitate comprehension and subsequent development of theory. The process has been studied in such areas as cluster analysis and numerical taxonomy. Automatic classification techniques have also been applied in keyword or document clustering for information retrieval [7].

How do these ideas apply to database analysis? At its simplest, a database consists of a set of records, each of which represents an 'entity' whose attributes are listed as a set of fields. We can extract information that may indeed be new knowledge, by using whatever 'select' and 'summarize' operations the database software may provide. In this situation, the user has to think of a question to ask, and the system responds. One potential shortcoming of this approach is that the user may not think of all the relevant things that could be asked and many crucial items may go unnoticed.

A database may be analyzed in other ways. For example, the data in a numeric field may be averaged, their range (highest and lowest values) may be identified, and the scatter of values about the average. In a nominal field (e.g. publisher names), the data can be counted and so ranked as to frequency of occurrence. Two fields can be compared to see if they co-vary, and how closely one may be predicted from the other – e.g. the values for height and weight in medical records. But to do this case by case for a database with many fields and large volumes of data becomes prohibitively laborious. Something more automatic, and more intelligent, is required: software that will look for patterns in the data on its own initiative, with or without some interaction from the user. This interaction can be valuable because there are a potentially very large number of patterns that can be generated in relation to any set of facts. So the user needs to provide criteria for the selection of patterns – background knowledge that defines the assumptions and constraints imposed on the facts and on candidate inductive patterns, including a preference criterion for reducing the chosen patterns to one or a few. When databases are very large, sampling must be used to provide a more manageable set of records for analysis, and the resulting uncertainty can be measured statistically.

Discovered knowledge can describe inter-field patterns relating values of fields in the same record, or inter-record patterns relating values aggregated over groups of records, or identifying useful clusters or interesting trends in time-dependent data. Discovered knowledge can take the form of a simple rule relating variables; or putting several rules together to form a causal chain or network, leading to semantic models or domain theories.

A pattern may be regarded as a collection or class of records sharing some- thing in common. Numeric discovery methods include, as already noted, cluster analysis and mathematical taxonomy – maximizing similarity within classes and minimizing similarity between them. Conceptual clustering uses not only similarity but also what has been called 'conceptual cohesiveness' as defined by background information. Interactive clustering includes contributions from the human user's knowledge. Learning algorithms work by identifying commonalities and/or differences among class members – e.g. by means of decision tree inducers, neural networks, genetic algorithms. Further knowledge discovery tasks are the summarization of class records by describing their common or characteristic features; the discrimination of records of one class from those of another; the comparison of a class with other (as yet un-classed) records; and the detection of anomalies or deviations from the normal pattern.

In more detail, the tasks of pattern discovery within databases may be summarized as follows: (1) class identification: the division of the database records into a set or hierarchy of classes based on attribute similarity; (2) concept description: here the task is to set a target concept and to generate a listing of the attributes that characterize it or that discriminate it from other records; (3) deviation detection: analysis to detect anomalous values or significant trends over time; (4) dependency analysis: functional dependencies between fields in a database are often already specified, but analysis may detect 'unknown' dependencies.

Learning from data can be considered to be a knowledge engineering strategy if the data represent records of expert decision-making. Alternatively, learning from performance data can derive new patterns and relationships that improve understanding of a certain process and therefore enable better decisions to be made in the future. Data models (rules and patterns) derived from historic data can be used to predict the outcome of future events. One potential application of discovered knowledge is the automated construction of knowledge bases for expert systems, or at least the refinement of a knowledge base initially developed by human experts.

III. DATABASE DISCOVERY HIERARCHY

To make an inductive inference is to assign an entity to a class: to state that 'all (or most) swans are white' is to class swans among the white entities. An early approach to inductive learning was to generate a classification hierarchy from a database [8]. More than a decade ago, Quinlan developed the algorithm known as the 'Interactive Dichotomizer' (ID3) [9]. A set of sample records of entities and their attributes is initially divided intellectually into two or more classes. The algorithm determines which attribute most effectively discriminates between the entities and allocates the data into the established classes. Next a significant attribute of each of the subsets is used to partition them further, and the process is repeated until subsets ('leaves') are reached that can no longer be partitioned. The resulting structure may be called a 'discrimination tree'. ID3 has been much used in subsequent developments.

A simple example (adapted from Fayyad [10]) will illustrate this form of classification. Figure 1 portrays a number of entities (persons, represented by X or O) graphed according to the numerical values of two attributes: their incomes and their bank loan debts. A third attribute is represented by the difference between the symbols X (has defaulted on loan) and O (has not defaulted). This attribute provides the predefined classes.

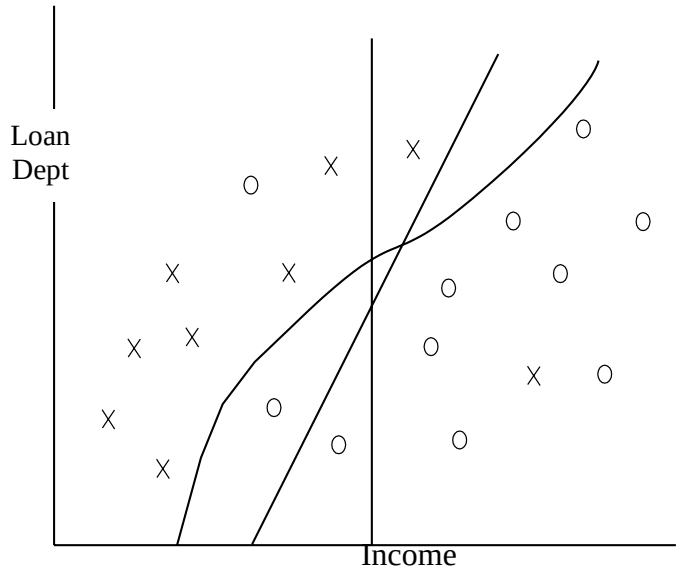

Figure 1. ***Possible outcomes of a classification algorithm***

A classification algorithm seeks to find the best way of mapping the data into these two classes. One such mapping might be represented by line A, which could be expressed as 'class persons with income less than A as likely defaulters'. To the left of line A are seven X and three O, so the rule classes correctly with a probability of 7/10 – the confidence level is 70%. Seven out of the nine X on the graph are to the left of line A, so the proportion of X covered by the rule is 7/9 – the goal coverage is 78%. A preferred mapping might be line B, which correctly assigns 8/10 = 80% to its left, and covers 8/9 = 89% of X; but the classing rule would not be so simple – it would be an expression involving both income and debt. A nonlinear technique such as can be provided by neural networks might generate the mapping C for the data, which correctly assigns 8/9 = 89% to its left, and still covers 8/9 = 89% of X, but could this curved line be expressed as an understandable classing rule? Neural networks may produce a superior classing strategy, but do not necessarily express it as a rule understandable by people and hence acceptable to them.

Now let us look at more realistic examples of classing. Consider an example from the use of the software Knowledgeseeker [11]. The database consists of medical records for 360 individuals. Each record lists twenty- five attributes relating to blood pressure level, diet, age and other characteristics– in detail: HypertensionLevel, TypeOfMilk, DeepFriedLastWeek, BeefLastWeek, PorkLastWeek, PoultryLastWeek, FishLastWeek, LambLastWeek, OtherMeat- LastWeek, CheeseLastWeek, EggsLastWeek, Meat2MealsLstWk, SaltInFood, SaltConsumption, ButterFood, SportsActivity, SleepTime, Smoking, Drink-Pattern, DrinksEveryDay, Age, YrsEducation, Income, Gender, Weight(kg). The goal set for the analysis is to examine all fields to determine the relative importance of each factor to the existence of abnormally high or low blood pressure. The first product of the analysis is a top-level hierarchy (Figure 2: the percentages represent the proportion of the parent population assigned to each category of hypertension, the bracketed figure is the total number of records in the sample). The program has identified which attributes appear to covary with hyper- tension; assessed Age as the most significant; and displayed the proportions within each age group of those having low, normal or high Hypertension values. In effect, the whole set of 360 records has been divided first according to the Age characteristic, and the Hypertension characteristics of each Age class have been displayed. The process can be continued – for example, the next most significant attribute affecting hypertension in the oldest age group can be displayed in Figure 3.

Similarly, analysis of the youngest age group gives the results in Figure 4. The software identifies other attributes that significantly affect hypertension, e.g. PorkLastWeek, SaltInFood and SportsActivity, and the numerical effect of each of these can be displayed. The data can if required be presented graphically.

The hierarchy developed is a classification of the hypertension records, with the sequence of characteristics used being chosen according to the significance of their effect on blood pressure. But it can be viewed in other ways. If we treat the percentages as estimates of probabilities, the hierarchy can be seen as a succession of IF-THEN rules, for example:

IF Age is 63-72
THEN likelihood of high Hypertension is 48.9%
but
IF Age is 63-72
AND FishLastWeek is 2-7 times
THEN likelihood of high Hypertension is only 23.1%

Hypertension Level			
low	18.3 %		
normal	60.3 %		
high	21.4 %		
	[360]		
	Age		
	32–50	*51–62*	*63–72*
low	35.1%	13.6%	5.4%
normal	56.1%	72.1%	45.7%
high	8.8%	14.3%	48.9%
	[114]	[154]	[92]

Figure 2. ***Hypertension relative to age***

Age 63–72		
FishLast Week		
	0–1 times	*2–7 times*
low	0.0%	19.2%
normal	40.9%	57.7%
high	59.1%	23.1%
	[66]	[26]

Figure 3. ***Hypertension of oldest group relative to fish diet***

Age 32–50			
DrinkPattern			
	regular	*occasional*	*never*
low	33.3%	33.3%	44.4%
normal	61.5%	33.3%	55.6%
high	5.1%	33.3%	0.0%
	[78]	[18]	[18]

Figure 4. ***Hypertension of youngest group relative to drink pattern***

If all the other influencing factors are included, we have a 'decision tree': for a given patient, if the value of each influencing factor is known, a 'decision' or estimate of the probability of high or low hypertension can be made. The decision tree in effect constitutes a small expert system to diagnose the likelihood of high or low blood pressure. Software such as Analyser [12] or Sipina [13] provides refinements in the induction of decision trees, for example:

(a) dividing the available data into a training set and a test set;
(b) offering a preliminary 'data view' of the training set: this displays a list of the fields in a table, showing the field names, types, values, usage and other information; for numeric fields it shows the minimum value, maximum value, average value, standard deviation and the number of non-numeric values; for nominal fields, it displays the frequency of occurrence of each discrete value. Studying this data view may help to suggest the direction of subsequent analysis;
(c) allowing selection of the field which is to be the outcome or dependent variable, chosen for the initial classification of the records. In some cases (e.g. the hypertension database) this is straightforward, but in more complex databases various fields may be chosen in turn for analysis;
(d) allowing selection of which fields are to be considered as influencing factors in the analysis, thus giving the user a chance to use his own knowledge of the domain to restrict the scope of the analysis;
(e) permitting the grouping of field values into manageable clumps – as in the Age field above;
(f) providing ways of 'pruning' an induced hierarchy, to remove less significant branches and simplify it;
(g) allowing the user to suggest changes to the tree and to re-analyze the data on this basis;
(h) giving facilities to check the performance of the decision tree against the test set of data.

Rule induction technology has been established for nearly two decades. However, when applied to large data sets in the real world, early exponents found that noise in data files prevented the building of accurate decision trees. Noise can be any or all of: insufficient attributes and insufficient data; not having all the relevant attributes to classify the outcome; the presence of irrelevant

attributes; errors and omissions in the data. The data fields may be corrupted due to typing errors or transmission errors. The contents of the data fields may contain inaccurate information as a result of human errors in making decisions, or machine errors in logging events. The data file may not contain enough data to cover the patterns one is trying to discover, or sufficient attribute fields to classify the outcome field. The data file may contain attribute fields which are irrelevant to the classification of the outcome field. The patterns which are contained in the data file may have been changing throughout the time span over which the data were collected.

The effect of noise is particularly strong at the extremities of a decision tree. By extending the algorithms and adding statistical techniques to the process, probabilistic rule induction produces pruned decision trees, less prone to distortion by noise. These trees are more compact and understandable, giving an improved probability of correctly classifying the training data set at the new leaf endings. The ultimate accuracy of the trees can be verified against new data, and the stability of each tree leaf population profile can then be accurately established.

IV. RULE GENEATION

To illustrate further the extraction of rules from data we may consider the example of IDIS, an Information Discovery System [14]. It comprises three modules: an induction module IXL (Induction in eXtremely Large databases) that automatically discovers rules, relationships and correlations from the data; a visualization module DVT (Data Visualization Tool) that generates graphical views of the data in a variety of selectable forms; and an anomaly module DBS (DataBase Supervisor) that automatically analyses the database to find anomalous data items and errors. Let us concentrate on the IXL module.

The sequence of operations used is:

- Select database from file menu.
- Select goal field.
- Select fields of interest as potential influencing factors (alternatively, select all fields if not sure which are relevant).
- Set discovery parameters (or use defaults).
- Generate rules.
- Browse rules.
- Read narrative report.

The conclusion of the IXL execution is thus a narrative report on the rules discovered. As an example we may look at the report on the analysis of a database Arcus, containing records relating to heart disease (this area seems to be a favorite for demonstrators of knowledge discovery software!).

The relative distribution for the data relating to HeartDisease is as follows:

HeartDisease = Yes 15.3%
others 84.7%

The fields having significant influence on the goal HeartDisease = Yes are: AGE, SBP, HDL, DBP [these are codes for measured factors that are believed to contribute to heart disease]. The factors discussed below are of interest due to the relatively high confidence level they render to HeartDisease = Yes.

(1)		AGE between 46 and 69
		Confidence: 52.7% Goal Coverage: 71.6%
(2)		SEX = Female
	AND	SBP is between 130 and 231
		Confidence: 41.3% Goal Coverage: 28.4%
(3)		SBP is between 127 and 231
	AND	DBP is between 82 and 112
	AND	GLUC is between 2.86 and 5.17
		Confidence: 40.7% Goal Coverage: 22.0%
(4)		HDL is between 0.98 and 2.23
	AND	LDL is between 4.35 and 9.63
	AND	SBP is between 117 and 231
		Confidence: 40.7% Goal Coverage: 20.2%
(5)		HDL is between 0.78 and 1.17
	AND	SBP is between 130 and 231
		Confidence: 42.0% Goal Coverage: 19.3%
(6)		LDL is between 4.95 and 9.63
	AND	SBP is between 127 and 231
		Confidence: 41.2% Goal Coverage: 19.3%

The following rules are interesting because they provide a relatively high gain in confidence by focusing on a specific cluster of data: HeartDisease= Yes if

(1)		SEX = Female
	AND	AGE is between 46 and 69
		Confidence: 66.7% Goal Coverage: 47.7%
(2)		AGE is between 51 and 69
	AND	DBP is between 30 and 84
		Confidence: 61.8% Goal Coverage: 38.5%
(3)		AGE is between 46 and 69
	AND	HDL is between 0.16 and 1.17

AND SBP is between 127 and 231
Confidence: 61.7% Goal Coverage: 34.0%
(4) AGE is between 51 and 69
AND HDL is between 0.16 and 1.27
AND DBP is between 30 and 84
Confidence: 62.5% Goal Coverage: 32.1%
(5) AGE is between 51 and 69
AND SBP is between 130 and 179
Confidence: 62.5% Goal Coverage: 27.5%

The report picks out the most significant rules that have been generated, and presents them for inspection by the user. Some knowledge discovery software offers more than one algorithm for generating rules. For example, Analyzer uses not only decision tree induction, but also genetic algorithms and neural networks. Genetic algorithms produce not a single decision tree, but various trees covering particular sets of attributes. For certain types of database, the discovery of even a handful of accurate pattern rules can be invaluable, while the likelihood of classifying the entire data set accurately might be low, and in such cases the use of genetic algorithms is recommended. Another scenario for evolving multiple pattern rules or decision trees is when the data set is small (500 records) and therefore only a fraction of the available attributes may appear in a single induced decision tree.

V. APPLICATIONS OF CLASSIFICATION

A few recent applications of classification algorithms may be noted. Two involve the use of the 'inductive logic' program Golem [15]. The first of these aimed at developing rules for the selection of chemicals to be used as drugs [16], in particular by relating chemical structure to biological activity. A training set of forty-four chemicals, all variants of a common basic molecule, was used. Their structures were described in terms of substituent radicals (such as OH, NO, NH, CH) at various positions on the basic molecule, and each radical was characterized according to various properties – size, polarity, flexibility, polarisability, hydrogen donor or acceptor characteristics. The prior 'classification' was the ranking of the test chemicals as to biological activity, based on experimental results. The aim of the analysis was to discover rules that would rank the chemicals as nearly as possible in the order of the observed ranking. Analysis of the data by Golem generated a number of rules relating to level of biological activity, in the form:

Drug A ranks more highly than Drug B

IF B has no substituent at positions 4 and 5

AND B at position 3 has substituent with polarisability = 1

AND A at position 3 has substituent with size = 2

AND A at position 3 has substituent with H-donor character = 0

AND A at position 3 has substituent with polarity > 0

By studying the set of rules generated, the ranking of the chemicals by Golem could be established. This ranking was compared to that known from experimental data, and correlated very strongly with it – more strongly than did previously used predictive methods.

The second Golem application is for the biological classification of river water quality [17]. The biological flora and fauna found in the water are often used as indicators of environmental quality, to supplement the more traditional chemical tests. The benthic (bottom-dwelling) invertebrates are currently regarded as a suitable group for monitoring. Data are available recording the presence and abundance of different types of benthic invertebrates in a number of river water samples, together with the assessment by an expert of the quality of the sample, assigning it to one of four classes, A, B, C, D. The classification task was to learn general rules that would enable new samples to be classed as to quality on the basis of the presence and abundance of test organisms. From a training set of 292 samples, Golem produced thirty-five rules for assigning samples to classes A, B or C. For example, 'if Leuctridae are present at high abundance level, assign sample to class A', which was established with a confidence level of 92%. Of the thirty-five discovered rules, twenty-five were considered good or acceptable by experts.

The classification application which has recently received a great deal of attention uses a system called SKICAT (Sky Image Cataloging and Analysis Tool) [18]. The database analyzed is the product of the second Palomar Observatory sky survey at the California Institute of Technology. Digitizing the photographic images will provide over three terabytes of data, relating to about a billion stellar objects. The initial task was to attempt to class stellar objects into four major categories: star, star with fuzz, galaxy, artifact. A total of forty optical attributes for each detected object were measured automatically in the survey. The experimenters discovered that these basic attributes were not sufficiently invariant between different regions of a photographic plate and between plates. They were able to develop four normalized and relatively consistent 'features' to describe each stellar object, and to use these as attributes for analysis. The training set of data was drawn from objects on four (of the total 3,000) photographic plates, concentrating on the small regions of the plates for which firm object classification decisions were available: in all, 1,688 objects were humanly classed to make up a test set. Analyzing this, SKICAT developed decision trees that, using the optical 'features', mapped stellar objects into the classes with an accuracy of 94%.

Subsequently, SKICAT was set the task of detecting objects that could possibly be quasars. These objects have unusual optical properties – in particular, anomalously high red shifts in their spectra – and appear to be highly luminescent but very distant galaxies. On photographs, quasars are indistinguishable from ordinary stars in our local galaxy. The Caltech group used SKICAT to select quasar candidates from objects detected in the sky survey, sorting through roughly one million objects to find each quasar. In December 1995, Caltech confirmed the discovery of sixteen new quasars [19].

VI. CLUSTERING

The process of 'classification' in knowledge discovery begins, as we have seen, with an intellectual division of the data into two or more classes on the basis of one attribute, and the derivation of a hierarchy of other attributes that best allocate the data to those classes. This is considered to be a form of 'supervised' learning. In contrast, the process known as 'clustering' is regarded as 'unsupervised'. Here there is no prior classing. The data are analyzed to find evidence for the existence of classes or clusters within them – a process more akin to the automatic classification procedures used within information science.

An example of software that carries out clustering is Cobweb/3, described as follows [20]. While examining their environment, humans form concepts by organizing observations on the basis of shared characteristics. These concepts provide a framework into which new observations having similar characteristics can be classified. Cobweb/3 is an approach to 'conceptual clustering', a computational framework for forming concepts from any data that can be represented as conjunctions of attributes and their values. As the program learns, it forms a hierarchy organizing the concepts, each of which summarizes some of the encountered instances. As in its human counterpart, this process occurs incrementally – concepts become more discriminating over time as the system encounters more examples. The system accepts as input a series of item descriptions in the form of a set of attribute-value pairs. The attributes can be represented with either nominal or numeric values, and they provide the basis for determining where the item is to be incorporated in the hierarchy. In addition to this learning mode, Cobweb/3 can also operate in a prediction mode, in which the acquired concepts are used to fill in missing information. Given a partial description and a hierarchy (either user-specified or one the system has construct- ed), it will classify a new item and predict the missing attribute's value.

Another clustering program is Autoclass, which has been used for example to analyse infrared astronomical data [21]. The Iras Low Resolution Spectral Atlas contains infrared data on 5,425 stellar sources. From the attributes in the data, Auto-class generated seventy-seven classes, significantly different from the humanly constructed classification adopted in the atlas. Autoclass was able to make many subtle distinctions between spectra previously regarded as similar, distinctions that experts accepted as significant.

VII. DEVIATION DETECTION

Several successful applications of knowledge discovery techniques have been developed for the analysis of change in data. These include Coverstory [22] and Spotlight [23] for supermarket sales data. The system to be discussed here is Kefir, for health management databases [24]. These contain data on the cost, price, usage and quality of various aspects of health care, and are maintained by hospitals, insurance companies and large corporations. The data are grouped according to the population group studied, e.g. region, business unit, union employees, and according to the medical area involved, e.g. inpatient, outpatient, surgical, maternity. A 'sector' is the combination of a particular medical area and a particular population group, and is associated with a set of measures relevant to the specified medical area. Kefir works on the data so organised, together with a set of 'norms' – the expected values for the measures used, established by previous data analysis. The system analyses the database to detect deviations from the norms. In many cases, the discovery of a deviation can be followed by a recommendation of action to handle the problem. A simple recommendation rule is:

IF measure = admissions-per-thousand
AND sector = premature-pregnancies
AND percent-deviation > +0.10
THEN 'initiate an early prenatal care programme'

The recommendations are defined, as instances of deviation occur, by a health expert. Approximately thirty-five recommendations covered the most general medical areas. The final output from Kefir is a report on the deviation detected. An extract from such a report is as follows: Total inpatient payments in this analysis fell 22.5%, from 1.4 to 1.1 million dollars. This total was less than the expected (norm) value of 1.2 million, giving rise to a saving of $147,000. If current conditions continue into the next period, this trend will result in $388,000 saving. The main cause of the decrease in total inpatient payments was the decrease in one of the sectors included, namely Medical Circulatory, which fell 58.7%.

In this case, the observed deviation of payments from the norm resulted, not in a recommendation, but in a prediction about future savings.

VIII. DEVIATION DETECTION

The various technical procedures being used to extract knowledge from databases were reviewed and analyzed. The challenges and advantages of different database knowledge discovery methods, particularly from very large databases that have relations naturally not explicit in database were analyzed. The machine learning techniques that are acting in the background of those methods were compared and analyzed. The efficiencies of various machine learning methods used for database knowledge discovery were analyzed and classified. In totality, the state of art in database knowledge discovery in several research areas, the progress and development of knowledge discovery models including multi-strategy discovery systems were reviewed, analyzed and their efficiencies were reported.

REFERENCES

[1] Fayyad, U.M. and Uthurusamy, R. eds. Proceedings of the First International Conference on Knowledge Discovery and Data Mining. Menlo Park, Cal.:AAAI Press, 1995.
[2] Fayyad, U.M. et al., eds. Advances in knowledge discovery and data mining. Menlo Park, Cal.: AAAI Press and Cambridge, Mass., London: MIT Press, 1996.
[3] Piatetsky-Shapiro, G. and Frawley, W.J., eds. Knowledge discovery in databases. Menlo Park, Cal.: AAAI Press, 1991.
[4] Frawley, W.J. et al. Knowledge discovery in databases: an overview. AI Magazine, Fall 1992, 57–70.
[5] Weiss, S.I. and Kulikowski, C. Computer systems that learn. San Francisco: Morgan Kaufmann, 1991.
[6] Van Rijsbergen, C.J. Information retrieval. 2nd edition. London: Butterworths, 1979.
Michalski, R.S. et al., eds. Machine learning. Berlin: Springer-Verlag, 1984.
[7] Quinlan, J.R. Learning efficient classification procedures. In: Michalski, R.S. et al., eds. Machine learning. Berlin: Springer-Verlag, 1984, pp. 463–482. See also his valuable review article: Induction of decision trees. Machine Learning, 1, 1986, 81–106.

[8] Fayyad, U.M. et al. From data mining to knowledge discovery: an overview. In: Fayyad, U.M. et al., eds. Advances in knowledge discovery and data mining. Menlo Park, Cal.: AAAI Press and Cambridge, Mass., London: MIT Press, 1996, pp. 1–34.

[9] Knowledgeseeker is a commercial program developed by Angoss Software. A demonstration version can be downloaded from the Internet at URL http://www.angoss.com/ks/ks.htm.

[10] Analyser is a commercial program developed by Attar Software. An evaluation version can be downloaded from the Internet at URL http://www.attar.com. The 'help' files in this provide very useful tutorial and explanatory guidance. Attar also produce an expert system shell, Xpertrule, into which the rules and decision trees generated by Analyser can be transferred.

[11] Sipina is a shareware program developed by D.A. Zighed. It can be downloaded from the Internet at URL ftp://eric.univ-lyon2.fr/pub/sipina/.

[12] IDIS is a commercial program developed by Information Discovery. A demonstration version is available on the Internet at URL http://www. datamining.com/. For some reason, its transfer to my computer is very slow and error-prone, and repeated attempts to download it have not succeeded.

[13] Muggleton, S. and Feng, C. Efficient induction of logic programs. In: Proceedings of First Conference on Algorithmic Learning Theory. Tokyo: Ohmsha, 1990.

[14] King, R. et al. Drug design by machine learning. Proceedings of the National Academy of Sciences, USA , 89, 1992, 11322–26.

[15] Dzeroski, S. Inductive logic programming and knowledge discovery in databases. In: Fayyad, U.M. et al., eds. Advances in knowledge discovery and data mining. Menlo Park, Cal.: AAAI Press and Cambridge, Mass.,London: MIT Press, 1996, pp. 117–152.

[16] Fayyad, U.M. *et al.* Automating the analysis and cataloging of sky surveys. *In*: Fayyad, U.M. *et al., eds. Advances in knowledge discovery anddata mining*. Menlo Park, Cal.: AAAI Press and Cambridge, Mass., London: MIT Press, 1996, pp. 471–493.

[17] *KDD Nuggets*, *96*(1), January 1996.

[18] Cobweb/3 is a commercial program developed at the University of Georgia. Information about it is available on the Internet at URL http://cognac.cosmic.uga.edu/abstracts/arc-13186.html.

[19] Cheesman, P. *and* Stutz, J. Bayesian classification (AutoClass). *In*: Fayyad, U.M. *et al., eds. Advances in knowledge discovery and datamining*. Menlo Park, Cal.: AAAI Press and Cambridge, Mass., London: MIT Press, 1996, pp. 153–180.

[20] Schmitz, J. *et al.* Coverstory – automated news finding in marketing. *In*: Volinio, L. *DSS Transactions*. Providence: Institute of Management Sciences, 1990.

[21] Anand, T. *and* Kahn, G. Spotlight – a data explanation system. *In: Proceedings of Eighth IEEE Conference on Applied AI*. Washington: IEEE Press, 1992.

[22] Matheus, C.J. *et al.* Selecting and reporting what is interesting. *In*: Fayyad, U.M. *et al., eds. Advances in knowledge discovery and datamining*. Menlo Park, Cal.: AAAI Press and Cambridge, Mass., London: MIT Press, 1996, pp. 495–515.

[23] Chen, H. Relevance feedback and probabilistic models in IR. *Journal ofthe American Society for Information Science*, *46*, 194–216, 1995.

[24] *Byte*, October 1995, 83–103.

[25] Carbonell, J.G., *ed*. *Machine learning, paradigms and methods*. Cambridge,Mass.: MIT Press, 1990.

[26] Knowledge Discovery Mine. Internet URL http://info.gte.com/~kdd/.

[27] *KDD Nuggets*. Subscribe to kdd-request@gte.com.

[28] Data Mine. Internet URL http://www.cs.bham.ac.uk/~anp/TheDataMine.html.

[29] *Machine Learning List*. Subscribe to ml-request@ics.uci.edu.Internet URL http://www.cs.wisc.edu/~belew/MLIA.html.