Data Mining - The Gaining Knowledge Progression

Introduction

Over the past couple of decades, the proliferation of information and data that is being stored in some electronic format has been astonishing. The accumulation of all that data has really taken place at an explosive rate. Research studies have shown that the amount of information (world-wide) doubles every 20 months, as the size and number of database installations increases at an even faster rate. Very sophisticated electronic data gathering devices, as well as high-speed interconnect solutions in use today contribute to the explosion of available data/information. Further, high-performance computing systems and storage devices are getting cheaper and cheaper. The crux of the (data accumulation) issue though is that (as most companies focused their efforts on collecting the data) a vast majority of these firms do not know what to do with all the information on storage. While companies recognize that the collected information reflects the core of their business operations and hence, that their decision makers could leverage the data to gain valuable insight into the business operations, the personnel, techniques, and methodologies to accomplish this task are at many companies just not in place. The database management systems at most firms allow accessing the data, but only explore a fraction of the potential that could be gained from further deciphering the vast amount of data. While online transaction processing (OLTP) systems excel at efficiently and safely storing data into databases, they are less effective in delivering meaningful analysis results in return. Analyzing and deciphering data provides (business) knowledge beyond the actual stored data pool. In other words, while the data pool is often rather voluminous, the current *knowledge value* may be rather low as no direct use can be made of it. It is the hidden information in the data that is sometimes much more useful to a company. Data Mining or Knowledge Discovery in Databases (KDD) provides these obvious benefits. This paper elaborates on the *gaining knowledge path* from the raw data to the extracted knowledge layer, elaborating on some of the techniques, tools, and issues surrounding data mining. It has to be pointed out that the term Data Mining has been stretched beyond its limits, and has become a buzzword to describe basically any form of data analysis. Therefore, the following definition of Data Mining or Knowledge Discovery in Databases is provided:

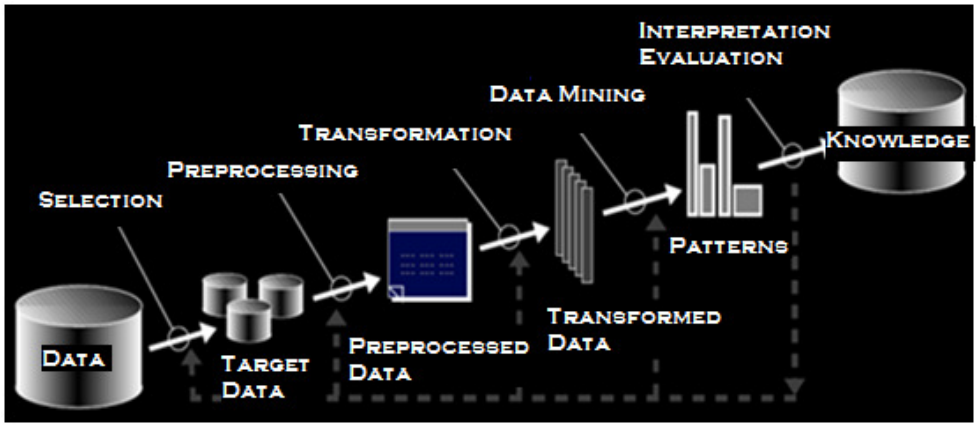
"Data Mining, or Knowledge Discovery in Databases (KDD) as it is also known, is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. This encompasses a number of different technical approaches, such as clustering, data summarization, learning classification rules, finding dependency net works, analyzing changes, and detecting anomalies." (MIT Press - Frawley, Piatetsky-Shapiro, Matheus)

Data Mining Process

The analogy with the actual mining process can be described as Data Mining referring to the utilization of a variety of techniques to identify nuggets of information (or decision-making knowledge) in bodies of data, and extracting these so that they can be used in areas such as decision support, prediction, forecasting, or estimation. Basically, Data Mining is concerned with the analysis of data and the use of software techniques for finding patterns and regularities in data sets. Normally high-performance parallel computer systems and software solutions are used to explore patterns by identifying the underlying rules and features in the data set (a process normally executed by high-skilled, expensive analysts). The underlying philosophy is that it is possible to strike gold in unexpected places, as the Data Mining SW extracts patterns that were either not previously discernible, or were so obvious that no one noticed them before. Data Mining analysis tends to work from the data set upwards, and the best techniques are those developed with a focus on large volumes of data. In other words, the idea is to utilize as much of the collected data as possible to arrive at reliable conclusions and decisions. The analysis process commences with a set of data, and uses a methodology to develop an optimal representation of the structure of the data to acquire knowledge. Once knowledge has been acquired, expansion to larger sets of data can be pursued based on the assumption that the larger data set mimics the structure of the sample data set. This process is analogous to an actual mining operation where large amounts of low-grade materials are sifted through in order to find something of value. Figure 1 depicts the major stages/processes identified in Data Mining and KDD. The process starts with the raw data and finishes with the extracted knowledge.

- ***Selection Stage*** - This stage encompasses selecting or segmenting the data according to some criteria (e.g. people who own a horse or people who own a cow) to establish or determine actual subsets of the data pool.
- ***Preprocessing Stage*** - This stage represents the data cleansing process where certain information is removed (deemed unnecessary and/or may slow down the queries). To illustrate, it is unnecessary to track the sex of a patient when studying pregnancy. Further, the data is reevaluated to ensure a consistent data format, as the data may have been compiled from different sources (sources where a patient's sex may have been recorded as *f* or *m* or as *1* or *0* or where dates reflect either a *mm/dd/yyyy* or a *dd/mm/yyyy* format).
- ***Transformation Stage*** - In this stage, the data may be transformed via introducing overlays (such as demographic overlays commonly used in market research studies). In other words, the data is made more useable and navigable.
- ***Data Mining Stage*** - This stage is concerned with the extraction of patterns from the data. A pattern can be defined as given set of facts (data) *F*, a language *L*, and some measure of certainty *C*. A pattern is defined as a statement *S* in *L* that describes relationships among a subset *Fs* of *F* with a certainty *C* such that *S* is simpler in some sense than the enumeration of all the facts in *Fs*.
- ***Interpretation and Evaluation Stage*** - In this stage, the patterns identified by the system are interpreted into knowledge that can then be used to support the (human) decision-making process. The focus can be on prediction and classification tasks, or on elaborating on some observed phenomena.

Figure 1: The Raw Data to Gaining Knowledge Process



Note: Figure courtesy of Data Mining Wiki

Data Mining verses Data Analytics

It is paramount to distinguish between Data Mining and Data Analytics related projects. While both, Data Analytics and Data Mining aim at information that is actionable, Data Analytics and Data Mining are as different as cows and horses. Data Analytics projects usually encompass hypotheses testing. The analyst has something in mind, and is focused on answering a question, but he/she has a hypotheses about the outcome. Data Mining on the other hand represents the act of discovery that in most cases lacks a hypotheses. Data Mining seeks for patterns (often by processing vast amounts of data), but these patterns and relationships were not previously known or anticipated.

Data Mining - The Learning Environment

The field of Data Mining research has been influenced by a number of other research areas such as inductive learning, machine learning (a branch of artificial intelligence), or statistics (to name a few).

Inductive learning - Induction represents the inference of information from data, while inductive learning describes the model building process where the environment (dataset/database) is analyzed focusing on finding (extracting)

patterns. Similar objects are grouped in classes or generations, and rules are formulated so that it is possible to predict the class/generation of unseen objects. The classification process identifies classes in a way that each class represents a unique pattern of values that form the class description. The nature of the environment is dynamic and hence, the model has to be adaptive (be able to learn). Inductive learning where the system infers knowledge itself via observing its environment is based on 2 distinct strategies; supervised and unsupervised learning. Supervised learning refers to learning from examples, where a supervisor component aids the system at constructing a model by defining classes and supplying examples for each class. The system has to find a description of each class (such as the common properties) and once the description has been formulated, the description and the class form a classification rule that can be used to predict class' of previously unseen objects. Unsupervised learning refers to systems that learn from observation and discovery. To illustrate, the Data Mining system is supplied with objects, but no classes are defined. Hence, the system has to observe the examples and recognize the patterns by itself. The result reflects a set of class descriptions, one for each class that was discovered (this approach is *similar* to the cluster analysis method used in statistics).

Deductive verses Inductive Arguments

A deductive argument is one in which it is impossible for the premises to be true but the conclusion to be false. Thus, the conclusion follows necessarily from the premises and inferences. In this way, it is supposed to be a definitive proof of the truth of the claim (conclusion). Here is a classic example:

1. *All men are mortal. (premise)*
2. *Plato was a man. (premise)*
3. *Plato was mortal. (conclusion)*

If the premises are true (and they are), then it simply is not possible for the conclusion to be false. If one has a deductive argument, and one accepts the truth of the premises, then one must also accept the truth of the conclusion, as if one rejects it, then one is basically rejecting logic itself. An inductive argument is one in which the premises are supposed to support the conclusion in such a way that if the premises are true, it is improbable that the conclusion would be false. Thus, the conclusion (*probably*) follows from the premises and inferences. Here is another classic example:

1. *Plato was Greek. (premise)*
2. *Most Greeks eat fish. (premise)*
3. *Plato ate fish. (conclusion)*

In this example, even if both premises are true, it is still possible for the conclusion to be false (maybe Plato was either allergic to or just didn't like fish). Words which tend to mark an argument as inductive (and hence probabilistic rather than definite) include probably, likely, possibly, or reasonably. It may seem that inductive arguments are weaker than deductive arguments, as with inductive arguments, there is always the possibility of arriving at a false conclusion. That suspicion is not entirely true though. With deductive arguments, the conclusions are already contained, even if implicitly, in the premises. This implies that with deductive arguments, one does not arrive at new information. In a best case scenario, one is shown information that was obscured or unrecognized previously. Hence, the sure truth-preserving nature of deductive arguments comes at a cost. Inductive arguments on the other hand do provide new ideas, and hence may expand knowledge about the world in a way that is impossible for deductive arguments to achieve. Thus, while deductive arguments may be used most often with mathematical problems, most other fields of research make extensive use of inductive arguments.

Statistics - While statistics is based on a solid, sound theoretical foundation, the results provided by statistics can be overwhelming to some, and rather difficult to interpret by others. In most cases, some guidance on where and how to analyze the data is required by most companies. Data Mining on the other hand allows (1) the expert's knowledge of the data and (2) the advanced analysis techniques of the computer systems to work as a unit. Statistical analysis solutions such as SAS (SAS, Statistical Analysis System) or SPSS (IBM, originally called Statistical Package for the Social Sciences) have been used to detect, decipher, and explain unusual patterns by utilizing statistical models. Statistics does have a role to play, especially when used for a more directed analysis of the results that are provided by Data Mining systems. To illustrate, statistical induction can be used to quantify the average failure rate of some parts or machines.

Machine Learning - Machine learning (ML) represents the automation of a learning process. ML is actually a branch of artificial intelligence that is focused on the design and development of algorithms that allow computer systems to evolve behaviors based on empirical data. A learner can take advantage of examples (data) to capture the characteristics of interest of the unknown, underlying probability distribution. Data can be seen as examples that illustrate relations among observed variables. A major focus is on automated learning to recognize complex patterns, and to compile intelligent decisions based on the data set (generalizing is sometimes necessary to keep the input space manageable). For Data Mining purposes, learning is tantamount to the construction of rules based on observations of environmental states and transitions. ML reflects a rather broad field of research that includes not only learning from examples, but also supervised or reinforced learning. In a nutshell, a learning algorithm utilizes the data set (and its accompanying information) as input and returns a statement (a concept representing the results of learning) as output. Machine learning examines previous/past examples and their outcomes, learns how to reproduce them, and defines generalizations about new/future cases. In general, ML systems utilize an entire finite set that is labeled the training set as input to acquire knowledge. This set contains examples (observations that are coded in some machine readable form). Next to the training phase, the ML process includes a test and an actual forecast cycle as well.

Data Mining and Machine Learning

Data Mining and the subsection of Machine Learning (ML) that focuses on learning from examples overlap in some areas (by some of the problems being addressed, as well as by the algorithms being used). At the same time, Data Mining and ML differ in others. While Data Mining is concerned with exploring understandable knowledge, ML is focused on improving the performance of an agent. In other words, training an artificial neural network to balance a scale would be addressed via ML and not by Data Mining. It has to be pointed out though that on the other hand, extracting knowledge via an artificial neural network can be very relevant for Data Mining purposes. Further, Data Mining projects normally operate on very large, real-world database systems, while ML normally utilizes smaller data sets. Hence, efficiency criteria's are much more paramount to Data Mining. Data Mining though is considered part of the ML framework that is concerned with exploring understandable knowledge in large sets of real-world examples. While integrating ML techniques into database systems (to implement Data Mining), the database systems require more efficient learning algorithms, as real-world databases are normally very large and noisy. In most circumstances, the deployed database systems are designed for purposes other than ML and/or Data Mining and hence, properties or attributes that would simplify the learning task are not present. A vast number of database installations are contaminated (i.e. errors are common) and therefore, the Data Mining algorithms have to cope with noise. This is different from most ML systems that normally operate in a lab environment that utilizes pristine data sets.

Data Mining - Modes of Operation

Based on Data Mining studies conducted by IBM over the years, 2 types of model (or modes of operation) emerged that show the most promise to be successfully applied to disinter, and ultimately propagate information of interest to the user community. The 2 modes of operation are know in the literature as verification and discovery model, respectively.

Verification Model - A verification model engages a user-defined hypothesis, and tests the validity of the hypothesis against the data. The emphasis is on the user/expert, who is responsible for formulating the hypothesis and issuing the query against the data to either affirm or negate the hypothesis. The challenge with this approach is that no new information is created in the retrieval process. The executed queries will always return records to either verify or negate the hypothesis. The search process is iterative in that the output is reviewed, a new set of questions (or hypothesis') are formulated to refine the search while the entire process is repeated. The user/expert is discovering the facts about the data by utilizing a variety of techniques such as queries, multidimensional analysis, and visualization to guide the exploration of the data under scrutiny.

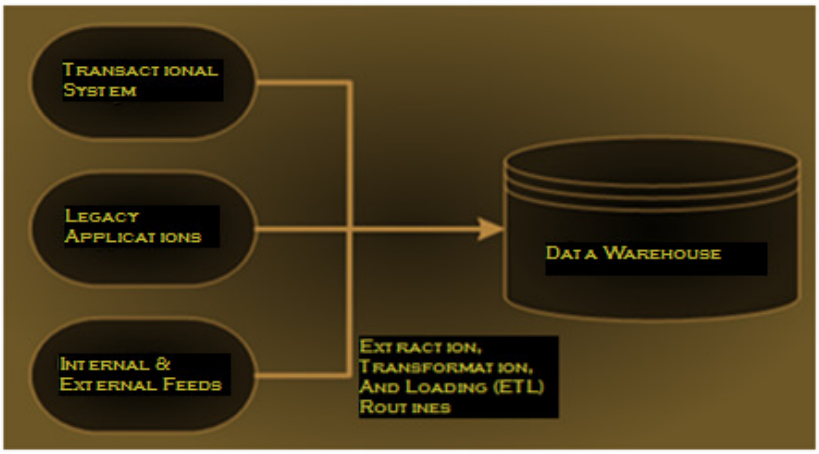
Discovery Model - A discovery model differs in its focus, as the system automatically discovers crucial information that is hidden in the data. The data is deceivered in search of (frequently occurring) patterns, trends, and generalizations without any intervention or guidance from the user/expert. The discovery tools aim at revealing a

large number of facts about the data in an as short as possible timeframe (efficiency and effectiveness arguments). To reiterate, based on the discovery mode, the data is scrutinized without a hypothesis, focusing (as an example) on grouping the records according to common characteristics being found by the system.

Data Warehouse Concepts

The Data Mining prospective may greatly be enhanced if the appropriate data has been collected and stored in a data warehouse. A data warehouse represents a relational database management system (RDMS) designed specifically to meet the needs of transaction processing systems. A data warehouse can be loosely defined as any centralized data repository that can be queried for business benefits (see Figure 2). In general, data warehousing is characterized as a technique to extract archived, operational data and to overcome inconsistencies among different (sometimes legacy) data formats. By integrating data throughout an enterprise, regardless of location, format, or communication requirements, it is feasible to incorporate additional expert information. Hence, data warehousing represents the logical link between what management is presented via the company's decision support (EIS - executive information systems) solutions and the company's operational activities. In other words, a data warehouse provides data that is already transformed and summarized and hence, reflects a solution environment that is appropriate for DSS (decision support systems) and EIS applications, respectively.

Figure 2: Conceptional Data Warehouse Setup



Note: Figure courtesy of Data Warehouse Wiki

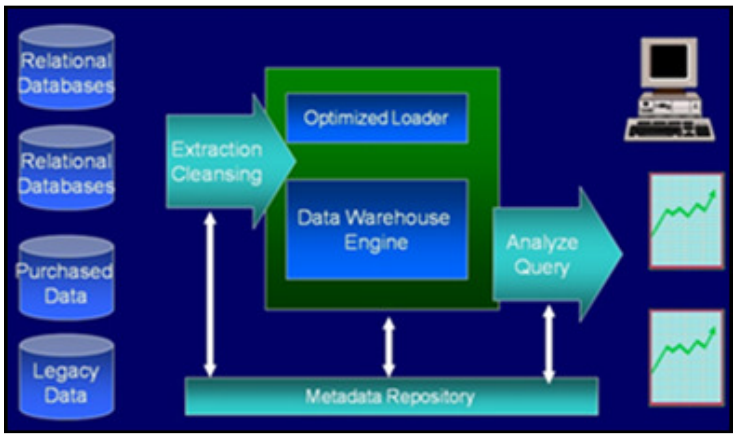
In general, there are 4 major characteristics that describe a data warehouse solution:

1. ***Subject Oriented*** - The data is organized according to subject. To illustrate, an insurance company utilizing a data warehouse would categorize their data by subject (customer, premium, claim) instead of by different product offerings (auto, life, home, mortgage). The data that is organized by subject only contains the information necessary for decision support processing purposes.
2. ***Integrated*** - As the data may be stored in separate applications throughout the operational environment, the encoding of the data may be inconsistent. To illustrate, some applications may classify gender differently while others use different date formats. As the data is moved from the operational environment into the data warehouse, a consistent coding convention is required, hence (as an example) some gender data or date format may have to be adjusted to accomplish the necessary consistency.
3. ***Time Variant*** - A data warehouse solution requires storing data is 5, 10, 15 or more years old. This data is used for comparison, trend, and forecasting purposes. While this data is not updated per se, properly sizing the data warehouse storage solution to accommodate for all the data is paramount.
4. ***Non Volatile*** - In a data warehouse, the data is neither updated nor changed in any way once the data is moved into the data warehouse solution. In other words, the data is only accessed and analyzed.

While establishing a data warehouse solution, the initial step is to insulate the current operational information. In other words, it is important to preserve the security and integrity of the company's mission-critical OLTP applications while still allowing access to as much of the data as possible. The resulting data warehouse may consume Tera or Peta Bytes of disk space. Hence, efficient techniques for storing, retrieving, and processing massive amounts of information are required to successfully operate in a data warehouse environment. Hence, well designed, adequately sized, optimally configured, performance centric installations that include high-speed SAN and parallel processing solutions are required. To reiterate, a data warehouse normally retrieves data from a variety of heterogeneous, operational database/application systems. The data is transformed and delivered to the data warehouse based on a selected model (or mapping definition). The data transformation, as well as the data movement processes are executed whenever an update to the warehouse data is required. Hence, some automated process to manage and execute these functions is normally required. The information that describes the model and the definition of the source data elements is labeled the *metadata* (see Figure 3). The metadata has to incorporate at a minimum:

- the structure of the data
- the summarization algorithm
- the mapping from the operational to the data warehouse environment

Figure 3: Metadata Repository



Note: Figure courtesy of Data Warehouse Wiki

To expand on the metadata discussion above, the data warehouse literature normally categorizes the metadata in:

- ***Technical metadata*** that defines the objects and processes in a data warehouse system. The technical metadata includes the systems metadata that defines the data structures such as the tables, fields, data types, indexes and partitions in the relational engine, as well as the databases, dimensions, measures, and data mining models. The technical metadata defines the data model, as well as the way it is displayed to the users via the reports, schedules, distribution lists, and user security settings.

- ***Business metadata*** that describes the data warehouse in more user-friendly terms. The business metadata outlines the available data, elaborates on the source of the data, and reports on the relationships to other data in the data warehouse. The business metadata normally serves as the documentation baseline for the system. Users normally operate on business metadata.

- ***Process metadata*** that is used to describe the results of various operations in the data warehouse. Within the ETL (extraction, transformation, and loading) process framework, all the key data that is generated by the tasks are logged during execution. The logged data includes the start time, the end time, the CPU usage, IO operations, as well as the rows that were processed. For ETL troubleshooting purposes, the process

metadata becomes very valuable. Further, some companies focus on collecting and selling process metadata, basically converting the process metadata into business metadata. The process metadata is normally of great interest to a business, as the data can be used to identify their users, which products they use, or what level of service they are receiving.

The data cleansing process (see Figure 3) is considered an important aspect of creating an efficient data warehouse solution. The process entails the removal of certain aspects of the operational data, such as low-level transaction information that may slow down the queries. The cleansing stage has to be as dynamic as possible (to accommodate all types of queries, even those that may require low-level information). The data should be extracted from production sources on a regular basis (intervals) and should be pooled centrally. The cleansing process has to remove duplications, and reconcile differences among various styles of the data. Once the data has been cleaned, the data is transferred to the data warehouse repository. A data warehouse can be used as a central store (to run the queries) or it can be setup as a set of data marts (see Figure 8). The data marts represent small warehouses that can be established to provide subsets of the main store by providing summarized information tailored towards the requirements of a specific group or department. A central store approach normally is based on rather simple data structures with very little assumptions on the relationships among the data, whereas the data marts generally utilize multidimensional databases that expedite query processing (as they entail data structures that reflect the most likely inquiry scenarios). Depending on the complexity of the data warehouse setup, a significant amount of work (design, sizing, specialized programming, testing, deployment) may be necessary to provide the interoperability among different products (maybe from multiple vendors), and to enable and optimize the required data warehouse processes.

Data Warehousing & OLTP Systems

A database that is designed for online transaction processing (OLTP), is commonly regarded as not being suitable for data warehousing purposes, as OLTP solutions are designed with different business perspectives in mind (such as maximizing the transaction capacity). Data warehouse solutions are by and large focused on query processing as opposed to transaction processing. Table 1 summarizes some of the differences (and similarities) of a data warehouse and an OLTP system.

Table 1: Data Warehouse vs. OLTP Systems - Attributes

Attribute	OLTP	Data Warehouse
Purpose	Execute Day-to-Day Operations	Information Retrieval & Analysis
Structure	RDBMS	RDBMS, Columnar & Correlation
Data Model	Normalized	Multi-Dimensional
Access	SQL	SQL plus Data Analysis Extensions
Type of Data	Data that runs the Business	Data that analyzes the Business
Condition/State of the Data	Changing, Dynamic	Historical, Descriptive

In a nutshell, a data warehouse serves a different business purpose than an OLTP system. While an OLTP system answers (as an example) aggregation questions like *what is the current account balance for customer xyz*, a data warehouse solution provides answers (as an example) to questions such as *what product line sells best in the Midwest, and how does this correlates to other demographic data*.

Data Mining Issues

As already discussed, Data Mining relies on database systems to supply the raw data for input. This may cause issues though as database systems tend be dynamic, incomplete, noisy, and normally rather large. Other issues may revolve around the adequacy and relevance of the information stored in database systems. As most database systems are designed for other than Data Mining purposes, the properties or attributes that simplify the learning task are not present. In practice, inconclusive data issues where some essential attribute (about the application domain) is missing in the dataset may make it impossible to discover significant knowledge about a given domain. To illustrate,

a certain illness (such as malaria) cannot be diagnosed from a patient database if that database does not contain the patients red blood cell counts. Database systems may be contaminated (error-prone) and hence, it cannot be assumed that the dataset is entirely valid/correct. Attributes that rely on subjective or measurement judgments can lead to error scenarios where some examples may even be miss-classified. An error in either the value of an attribute or the class information is labeled as noise. Where possible, the goal has to be to eliminate noise from the classification information to enhance the overall accuracy of the generated rules. Missing data can be treated by discovery systems in a number of ways:

- disregard missing values
- omit the corresponding records
- infer missing values from known values
- treat missing data as a special value to be included additionally in the attribute domain
- average over the missing values by using Bayesian techniques

Noisy data typically fits a regular statistical distribution (such as Gauss), while wrong values reflect data entry errors. Statistical methods can be used to address the issues of noisy data and to segregate the noise into different types. Uncertainty refers to the severity of the error and the degree of noise in the dataset (data precision is an important consideration in a discovery system). Database systems tend to be large and dynamic. The data is in an ever-changing state as information is added, modified, or removed. The crux of the issue here is how to ensure (from a Data Mining perspective) that the rules are current and consistent based on the status quo. Further, the learning process is time-sensitive, and as some data values may vary over time, the discovery system may be affected by the timeliness nature of the data. Further, the relevance or irrelevance of some of the fields in the database impact the discovery cycle. To illustrate, zip codes are fundamental to any studies trying to establish a geographical connection to an item of interest such as the sales of a product. Some areas where Data Mining based applications are successfully being utilized are:

Retail/Marketing

- Identify customer buying patterns
- Find associations among customer demographics
- Predict response behavior to various marketing campaigns
- Market basket analysis

Banking

- Detect patterns of fraudulent credit card use
- Identify loyal customers (for awards programs)
- Predict customers likely to change their credit card affiliation
- Determine credit card spending by customer groups
- Find hidden correlations among different financial indicators
- Identify stock trading rules based on historical market data

Insurance, Health Care, Medicine

- Claims analysis (such as what medical procedures are claimed together)
- Predict customers that will buy new policies
- Identify behavior patterns of risky customers
- Identify fraudulent behavior
- Characterize patient behavior to predict office visits
- Identify successful medical therapies for different illnesses

Transportation

- Determine the distribution schedules among factories
- Analyze loading patterns
- Analyze order patterns

Association Mining

Data Mining methods may be classified either by the function they perform or according to the class of application they address. Data Mining tools have to infer a model from the database, and in the case of supervised learning, this process requires the user to define one or more classes. The database contains one or more attributes that denote the class of a tuple (known as predicted attributes), whereas the remaining attributes are labeled predicting attributes. A combination of values for the predicted attributes defines a class. While learning the classification rules, the system has to identify the rules that predict the class from the predicting attributes and hence, the user has to define the conditions for each class. Based on that, the data mining system constructs the descriptions for the classes. Association mining that discovers dependencies among values of an attribute was introduced by Agrawal (1993), and has emerged as an important research area. The problem of association mining, also referred to as the *market basket* problem, is formally defined as follows. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items and $S = \{s_1, s_2, \ldots, s_m\}$ be a set of transactions, where each transaction $s_i \in S$ is a set of items that is $s_i \subseteq I$. An *association rule* denoted by $X \Rightarrow Y$, $X, Y \subset I$, and $X \cap Y = \Phi$, describes the existence of a relationship between the two item-sets X and Y. Several measures have been introduced to define the *strength* of the relationship between item-sets X and Y such as *Support*, *Confidence*, and *Interest*. The definitions of these measures (from a probabilistic view point) are:

- $SUPPORT(X \Rightarrow Y) = P(X, Y)$, or the % of transactions in the database that contain both X and Y
- $CONFIDENCE \quad (X \Rightarrow Y) = P(X, Y) / P(X)$, or the % of transactions containing Y in those transactions containing X
- $INTEREST(X \Rightarrow Y) = P(X, Y) / P(X)P(Y)$ represents a test of statistical independence

SUPPORT for an item-set S is calculated as $SUPPORT(S) = \frac{F(S)}{F}$, where $F(S)$ represents the number of transactions having S, and F reflects the total number of transactions. For a minimum *SUPPORT* value *MINSUP*, S is a large (or frequent) item-set if

$SUPPORT(S) \geq MINSUP$, or $F(S) \geq F * MINSUP$

Suppose the transaction set T is divided into two subsets T_1 and T_2, corresponding to two consecutive time intervals, where F_1 represents the number of transactions in T_1 , and F_2 reflects the number of transactions in T_2, ($F=F_1+F_2$). $F_1(S)$ symbolizes the number of transactions having S in T_1. $F_2(S)$ characterizes the number of transactions having S in T_2, ($F(S)=F_1(S)+F_2(S)$). By calculating the *SUPPORT* of S, in each of the two subsets, the conclusion is:

$$SUPPORT_1(S) = \frac{F_1(S)}{F_1} \text{ and } SUPPORT_2(S) = \frac{F_2(S)}{F_2}$$

S reflects a large item-set if

$$\frac{F_1(S) + F_2(S)}{F_1 + F_2} \geq MINSUP \text{ , or } F_1(S) + F_2(S) \geq (F_1 + F_2) * MINSUP$$

In order to find out if S is a large item-set or not, 4 cases are considered:

1. S is a large item-set in T_1 and also a large item-set in T_2, i.e., $F_1(S) \geq F_1 * MINSUP$ and $F_2(S) \geq F_2 * MINSUP$
2. S is a large item-set in T_1, but a small item-set in T_2, i.e., $F_1(S) \geq F_1 * MINSUP$ and $F_2(S) < F_2 * MINSUP$
3. S is a small item-set in T_1, but a large item-set in T_2, i.e., $F_1(S) < F_1 * MINSUP$ and $F_2(S) \geq F_2 * minsup$
4. S is a small item-set in T_1 and also a small item-set in T_2, i.e., $F_1(S) < F_1 * MINSUP$ and $F_2(S) < F_2 * MINSUP$

In the 1st and 4th cases, S is a large item-set and a small item-set in transaction set T, respectively. In the2nd and 3d cases, it is not unambiguous to determine if S is a small item-set or a large item-set. Formally speaking, let

SUPPORT(S) = MINSUP + δ, where $\delta \geq 0$ if *S* is a large item-set, and $\delta < 0$ if *S* is a small item-set. The above four cases have the following characteristics:

- $\delta_1 \geq 0$ *and* $\delta_2 \geq 0$
- $\delta_1 \geq 0$ *and* $\delta_2 < 0$
- $\delta_1 < 0$ *and* $\delta_2 \geq 0$
- $\delta_1 < 0$ *and* $\delta_2 < 0$

S is a large item-set if $\frac{F_1 * (MINSUP + \delta_1) + F_2 * (MINSUP + \delta_2)}{F_1 + F_2} \geq MINSUP$, or

$F_1 * (MINSUP + \delta_1) + F_2 * (MINSUP + \delta_2) \geq MINSUP * (F_1 + F_2)$

which can be written as $F_1 * \delta_1 + F_2 * \delta_2 \geq 0$

Generally, let the transaction set *T* be divided into *n* transaction subsets T_i 's, $1 \leq i \leq n$. *S* is a large item-set if $\sum_{i=1}^{n} F_i * \delta_i \geq 0$, where F_i represents the number of transactions in T_i and $\delta_i = SUPPORT_i(S) - MINSUP$, $1 \leq i \leq n$. $-MINSUP \leq \delta_i \leq 1-MINSUP$, $1 \leq i \leq n$.

For those cases where $\sum_{i=1}^{n} F_i * \delta_i < 0$, there are two options, either (1) discard *S* as a large item-set (a *small item-set* with no *history* record maintained), or (2) keep it for future calculations (a *small item-set* with *history* record maintained). In this case, it is not reported as a large item-set, but its $\sum_{i=1}^{n} F_i * \delta_i$ formula will be maintained and checked through the future intervals.

Data Mining - Some (Popular) Techniques

Data mining techniques are very important to the data mining process as a whole. The type of technique to use can vary from project to project and has to depend on the type of data to be analyzed. The analyst should also have a good understanding of the business, and what the data/information is to be used for. This is imperative to determine what data represents information, and how to establish the rules that will generate useful data that is beneficial to the business. It is paramount to point out though that no single technique can be defined as the optimal method for data mining.

Clustering represents a hard combinatorial problem and is defined as the unsupervised classification of patterns. The formation of clusters is based on the principle of maximizing the similarity among objects of the same cluster, while simultaneously minimizing the similarity between objects belonging to distinct clusters. There are four basic types of clustering algorithms normally being used, known as *partitioning algorithms*, *hierarchical algorithms, density-based algorithms* and *grid-based algorithms.* Partitioning algorithms construct a partition of *N* objects into a set of *k* clusters. Hierarchical algorithms create a hierarchical decomposition of the database that can be presented as *dendrogram* [Zhang]. Density-based algorithms search for regions in the data space that are denser than a threshold, and form clusters from these dense regions. Grid-based algorithms quantize the search space into a finite number of cells and then operate on the quantized space. Genetic algorithms (GA) have been proposed for clustering, because they avoid local optima and are insensitive to initialization. The individuals encode a fixed number (*k*) of clusters, using a binary representation for the center of clusters. The minimization of the squared error reflects the fitness function used to guide the search. The classical *k-means* clustering algorithm (and its variation the *k-medoids*) are representatives of partitioning techniques, and have widely been used in clustering applications. The reason behind the popularity of the k-means algorithm has to do with the simplicity and the speed of the algorithm. Broadly speaking, given the number of desired clusters, a k-means algorithm attempts to determine *k* partitions that optimize a criterion function. The square-error criterion *E* is the most commonly used, and is defined

as the sum of the squared Euclidean distances between each multidimensional pattern p belonging to C_i cluster and the center m_i of this cluster.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left\| p - m_i \right\|^2$$

With a k-means algorithm, each cluster is represented by a vector corresponding to the center of gravity for the cluster. In k-medoids, each cluster is described by one of the patterns, which are closely located to the center of gravity of the cluster. Both k-means and k-medoids assign pattern to clusters, trying to minimize the square-error function in order to obtain k partitions that are as compact and separated as possible. However, there are some well-known drawbacks, such as (1) sensitivity to the initialization process, which can lead to local optima, (2) sensitivity to the presence of noise, (3) discovery of clusters with similar sizes and densities, and (4) discovery of hyperspherical clusters. The k-means method works well for clusters that are compact clouds (hyperspherical in shape) and are well separated from each other. However, when there are large differences in the sizes or geometries or densities of different clusters, the square error method could split large clusters to minimize the Euclidean distances equation presented above.

Figure 4: Workload Clustering - Linux MPP Environment

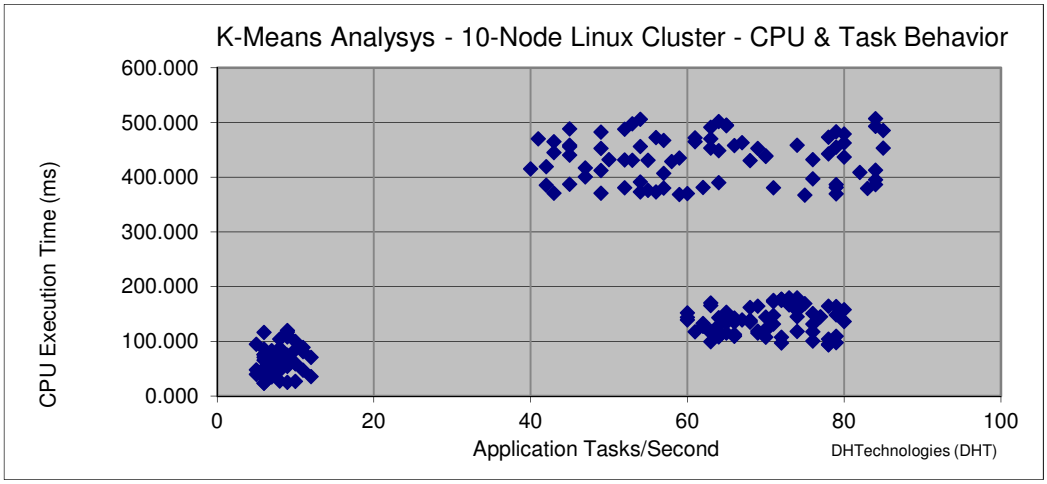


Figure 4 outlines the results of a K-Means analysis conducted for a Linux MPP (massive parallel computing) environment. 12-month worth of production OS and application performance data reflected the input data set. The goal of the study was to identify (performance specific) patterns among the systems (CPU, memory, SAN, Interconnect, network) and application subsystems that could be used to conduct forecasting and future cluster sizing (design) studies. The sample data outlined in Figure 4 reveal 3 distinct clusters while examining the collected task CPU execution time data in relation to the application task throughput behavior.

Market basket analysis (also known as *association rule discovery* or *affinity analysis*) is a rather popular data mining method. In the simplest scenario, the data consists of two variables: a *transaction* and an *item*. For each transaction, there is a list of items. Typically, a transaction is a single customer purchase, and the items are the items that were bought. An *association rule* is a statement of the form (item set A) $\Rightarrow$ (item set B).

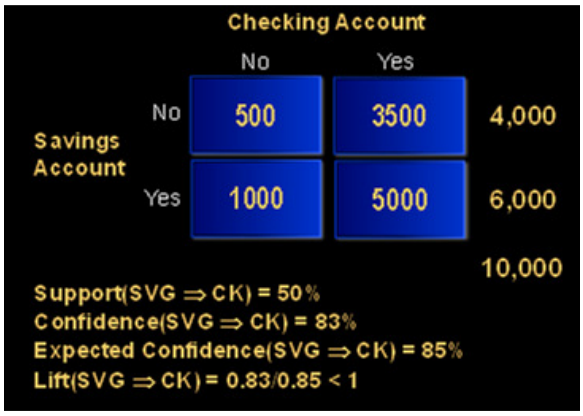
The aim of the analysis is to determine the strength of all the association rules among a set of items. The strength of the association is measured by the *support* and *confidence* of the rule. The support for the rule $A \Rightarrow B$ is the probability that the two item sets occur together. The support of the rule $A \Rightarrow B$ is estimated by:

$$\frac{transactions\ that\ contain\ every\ item\ in\ A\ and\ B}{all\ transactions}$$

It has to be pointed out that support is symmetric. In other words, the support for the rule $A \Rightarrow B$ is the same as the support for the rule $B \Rightarrow A$. The confidence of an association rule $A \Rightarrow B$ represents the conditional probability of a transaction containing item set B given that it contains item set A. The confidence is estimated by:

$$\frac{transactions\ that\ contain\ every\ item\ in\ A\ and\ B}{transactions\ that\ contain\ the\ items\ in\ A}$$

Figure 5: Market Basket Analysis



Note: Figure courtesy of Data Mining Wiki

The interpretation of the implication (⇒) in association rules is precarious. High confidence and support does not imply cause and effect. The rule is not necessarily interesting. The two items might not even be correlated. The term *confidence* is not related to the statistical usage. Hence, there is no repeated sampling interpretation. Considering the association rule (saving account) ⇒ (checking account) (see Figure 5). This rule has a 50% support (5,000/10,000) and 83% confidence (5,000/6,000). Based on these two measures, this scenario may be considered a strong rule. On the contrary, those without a savings account are even more likely to have a checking account (87.5%). Saving and checking are in fact, negatively correlated. If the two accounts were independent, then knowing that a person has a saving account does not help in knowing whether that person has a checking account. The expected confidence if the two accounts were independent is 85% (8,500/10,000). This is higher than the confidence of SVG ⇒ CK. The *lift* of the rule $A \Rightarrow B$ reflects the confidence of the rule divided by the expected confidence, assuming that the item sets are independent. The lift can be interpreted as a general measure of association between the two item sets. Values greater than 1 indicate positive correlation, values equal to 1 indicate zero correlation, and values less than 1 indicate negative correlation. It has to be noticed that the lift is symmetric as well. Ergo, the lift of the rule $A \Rightarrow B$ is the same as the lift of the rule $B \Rightarrow A$.

Link Analysis refers to a graphical approach to data analysis, focusing on establishing connections and relationships in order to determine knowledge and facts. Link analysis aids at exploring data that identifies relationships among the values in the databases. There are various types of link analysis strategies in use, but the most common are *stratification, association discovery, and sequence discovery.* Stratification represents the process of retrieving records from the data storage by utilizing the hypothesized links as the retrieval keys. Association discovery identifies rules about items that appear together in an event (such as in a purchase transaction). The final and most commonly applied link analysis technique *sequence discovery* is similar to association discovery, but augments on the association relation by adding a time epoch into the equation framework.

Data Visualization's role in data mining is to allow the analyst to gain perception on the observed data. Data visualization solutions are used to assist the analyst in fostering a deep, intuitive understanding of the data to be mined. Either a *conventional* or a *spatial* visualization technique is being used. In general, scatter plots, pie charts, or some other graphical representation format that depict the relationships among the data sets are employed. Conventional visualization refers to the usage of graphs, charts, or other graphical data representation. The goal is to graphically depict overall scenarios and not individual data points in itself. Conventional visualization involves the computation of descriptive statistics such as counts, frequencies, ranges, or proportions. Spatial visualization on the other hand employs plots that portray actual data in their feature spaces. With spatial visualization, there is no computation of descriptive statistics, as the representation preserves the geometric relationships among the data (i.e. the spatial characteristic is sampled and visualized).

Artificial Neural Networks (ANN) represent an information processing paradigm that is inspired by the way biological nervous systems (such as the brain) process information. The key element is the unique structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in accord to solve specific problems. ANN's, like people, learn by example. An ANN is tuned for a specific application through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist among the neurons (ANN's utilize the same approach). ANN's have verified their success in many applications due to their ability to solve most problems with relative ease of use, as well as the model-free property they enjoy. ANN's can solve problems without the need to understand or learn the analytical and statistical properties of neither the problem nor the solution steps. Hence, ANN's are already being used for many mining purposes (searching for the hidden patterns within the explicitly stored information in database systems) via e.g. Hopfield-Tank networks to solve classification, scheduling, or optimization related projects. Classification is one of the data mining problems that is recently receiving great attention. For that, the approach of symbolic classification rules using ANN's has been appreciated. To illustrate one possible approach, concise symbolic rules with high accuracy can be extracted from an ANN. The ANN can first be trained to achieve a required accuracy rate. Redundant connections of the network can then be removed via some network pruning algorithm. The activation values of the hidden units in the network can now be analyzed, and classification rules can be generated by utilizing the results of this analysis. It has to be pointed out that next to ANN's, Fuzzy Logic has been successfully used in some data mining related projects. To illustrate, Fuzzy logic provides techniques for addressing cognitive issues in the real world. Ergo, fuzzy logic techniques and data mining practices have been successfully fused in IT security, medical imaging, or multimedia related projects.

Figure 6: Prediction Analysis

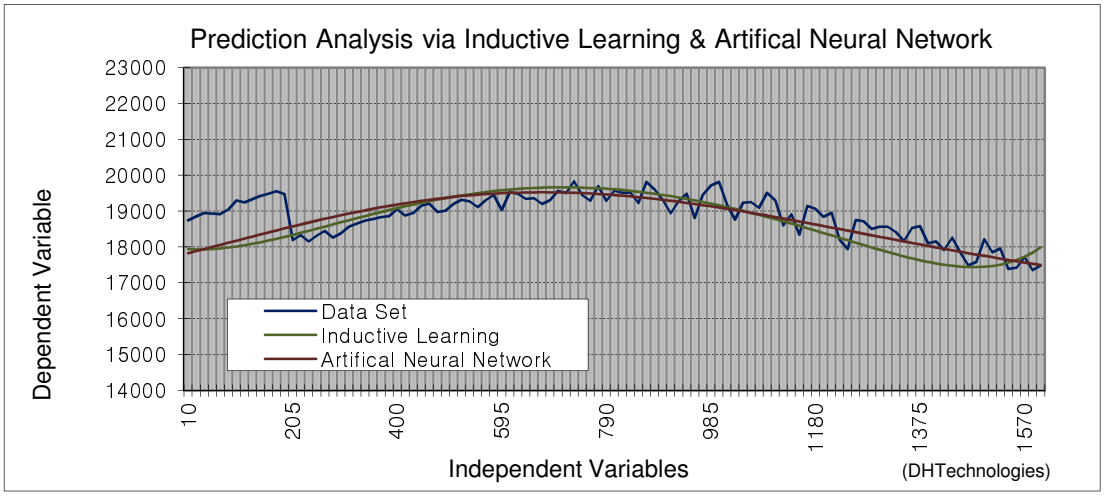
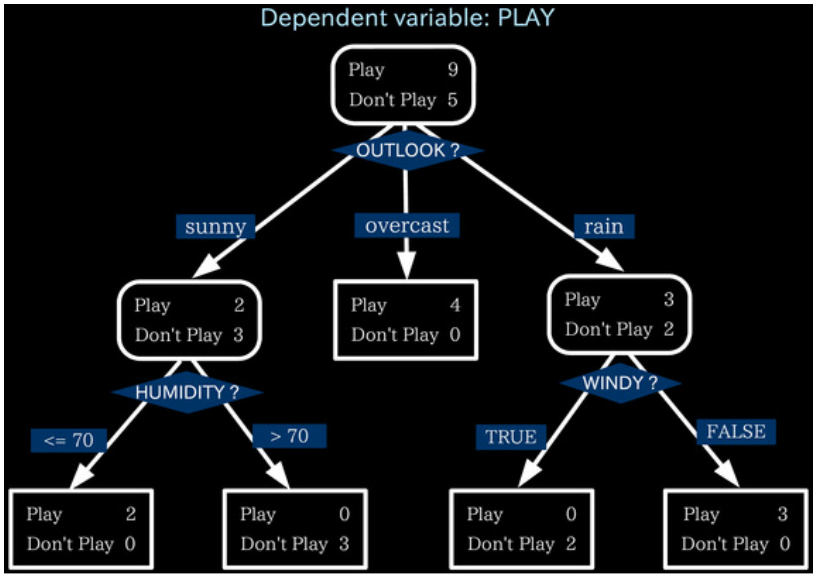


Figure 6 outlines a (small) subset of the results of an estimation study for a financial database application, where for a given data pool, 8 independent and 1 dependent variable comprised the application environment. The study was conducted on a training data set of 2,500,000 records and a test data set of 1,500,000 records, respectively. The goal of the study was (after the learning phase) to predict 500,000 records. The study utilized an

inductive learning, as well as an artificial neural network based modeling approach. After the learning phase, both techniques reflected the actual base data pool with high fidelity. In data mining studies, it is not uncommon to utilize multiple techniques concurrently to cross-compare results and to identify any potential implementation issues.

Decision Trees are generated by algorithms that identify various ways of splitting a data set into branch-like segments. It has to be pointed out that decision trees lie at the heart of most *rule induction* algorithms used in data mining. These segments form an inverted decision tree that originates with a root node on top of the tree. The object of analysis is reflected in this root node as a simple, 1-dimensional display in the decision tree interface.

Figure 7: Sample Decision Tree



Note: Figure courtesy of Decision Tree Wiki

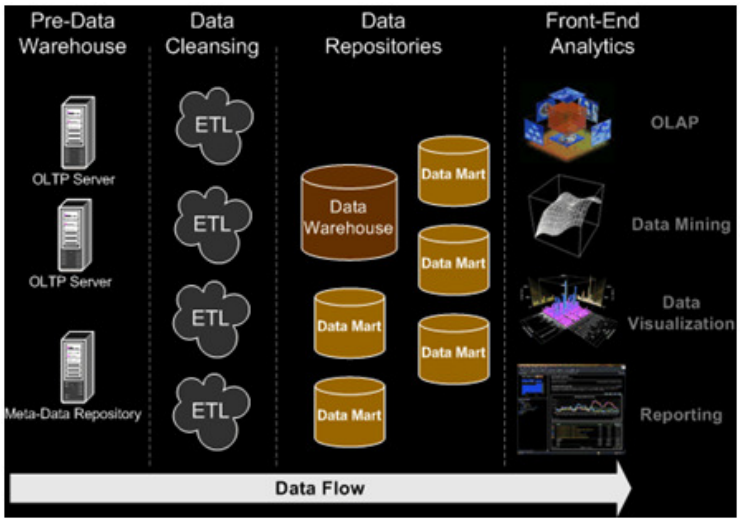
The name of the field of data that reflects the object of analysis is usually displayed along with the spread or distribution of the values that are contained in that field. A sample decision tree is illustrated in Figure 7, which shows that the decision tree can reflect both, a continuous and categorical object of analysis. The nodes reflects all the data set records, fields, and field values that are found in the object of analysis. The discovery of the decision rule to form the branches/segments underneath the root node is based on a method that extracts the relationship between the object of analysis (that serves as the target field in the data), and one or more fields that serve as input fields to create the branches/segments. The values in the input field are used to estimate the likely value in the target field. The target field is also called an outcome, response, or dependent field or variable.

Some limitations with decision trees revolve around the fact that the problem of learning an optimal decision tree is known to be NP-complete (under several aspects of optimality, as well as for rather simple concepts). Hence, practical decision-tree learning algorithms are based on heuristic algorithms (such as the greedy algorithm where locally optimal decisions are made at each node). Such algorithms do not guarantee a globally optimal decision tree though. Recent developments suggest the usage of genetic algorithms to avoid local optimal decisions. Over-fitting may be another issue (where the decision trees are too complex and do not generalize the data very well). Pruning techniques are normally required to avoid scenarios like that. Solving problems involving XOR statements, parity, or multiplexing scenarios are rather difficult to solve via decision trees, as the trees get prohibitively large. In circumstances like that, it may be beneficial to assess the usage of statistical relational learning or inductive logic programming based approaches.

Online Analytical Processing (OLAP)

One of the major challenges faced by many companies today is how to process ever larger database systems (that contain increasingly complex data structures) while at the same time not sacrificing the aggregate performance behavior of the IT solutions. Client-server, cluster, GRID, or even Cloud based architectures allow organizations to handle specific data management problems in a more scalable manner. Many companies have realized that RDBMS solutions are not suitable for their entire spectrum of database applications. Hence, most SW vendors offer other tools and solutions to address complex data scenarios such as imaging, audio, or multimedia. OLAP reflects a class of software tools that provides analysis of data stored in a database. OLAP tools enable users to analyze different dimensions of multidimensional data (for time series and trend analysis) (see Figure 8). In general, the major component of an OLAP setup consists of one or more OLAP server systems that are situated between the clients and some form of database management system. The OLAP server is aware of the data organization in the database solution and is equipped with special functions to analyze the data (OLAP servers are available for basically all the major database systems). In a nutshell, OLAP can be defined as the dynamic synthesis, analysis and consolidation of large volumes of multidimensional data (E.F. Codd, 1993).

Figure 8: Data Flow



Note: Figure courtesy of OLAP Wiki

Some of the major rules and requirements for an OLAP system revolve around terms such as multidimensional conceptual view, intuitive navigation, business mathematics, transparency, client/server architecture, or flexible reporting. Another focus is on performance. Theoretically, OLAP systems should provide response time values in the low second range (i.e. instant response so that the users do not lose their train of thought). One fundamental question though is what is multidimensional data, and when does it become OLAP? OLAP essentially represents a way to establish associations among dissimilar components of information (by utilizing predefined business rules). A multidimensional database has to be able to express complex business calculations in a rather simple way. Hence, the data has to be referenced, and the mathematics have to be defined. Dimensional database systems are not without major challenges, as these systems are not suited to store just any data type (such as lists). With dimensional systems, users do have the liberty to freely explore the data and to generate reports without being restricted to a set format. As an illustration, an OLAP database may be comprised of sales data aggregated by region, product type, and sales channel. A representative OLAP query may access a multi Tera-Byte, multi-year sales database to locate all product sales in each region for each product type. After reviewing the results, an analyst may refine the query to isolate sales volume for each sales channel within the region/product classifications. As a final step, the analyst may perform year-to-year or quarter-to-quarter comparisons for each sales channel. This entire process has to be processed online (with a rapid response time behavior) so that the analysis process remains serene.

OLAP verses OLTP Solutions

OLAP solutions differ rather significantly from OLTP applications that normally consist of a large number of relatively simple transactions. The OLTP transactions usually retrieve and update a small number of records that are contained in several distinct tables. The relationships among the tables are generally straightforward. A typical customer order entry OLTP transaction may retrieve all of the data relating to a specific customer, and then insert a new order record for the customer. Information is selected from the customer information, customer order, and customer product tables. Each row in each table contains a customer identification number that is used to relate the rows from the different tables. The relationships among the records are simple, and only a few records are actually retrieved or updated by a single transaction. Hence, major disparities between OLAP and OLTP can be described as:

- OLTP solutions normally handle mission-critical production data that is accessed via rather simple queries
- OLAP servers handle management-critical data that is accessed via an iterative analytical investigation process
- OLAP and OLTP alike encompass special requirements, and hence necessitate optimized servers for the 2 types of processing paradigms
- OLAP database servers utilize multidimensional structures to store information and relationships among the data. Multidimensional structures can be visualized as cubes of data, and cubes within cubes of data. Each side of the cube is considered a dimension. Each dimension represents a different category (such as product type, region, sales channel, or time). Each cell within the multidimensional structure contains aggregated data relating elements along each of the dimensions. To illustrate, a single cell may contain the total sales numbers for a given product in a region based on a specific sales channel in a single month.
- OLAP database servers support common analytical operations such as consolidation, drill-down, or slice and dice. Consolidation involves the aggregation of data such as simple roll-ups or complex expressions involving interrelated data. To illustrate, sales offices can be rolled-up to districts, and districts rolled-up to regions. Via drill-down, OLAP servers can reverse the direction and automatically display detailed data that comprises consolidated data. The slice and dice approach describes the ability to glance at the database from various angles. One slice of the sales database may disclose all sales for product *X* within regions. Another slice may reveal all sales by sales channel within each product type. Slice and dice is often performed along a time axis (to analyze trends and to identify patterns).

To summarize, OLAP solutions logically organize data in multiple dimensions that allow the analysts to quickly and easily analyze complex data relationships. The database in itself is physically organized in a way that related data can be rapidly retrieved across multiple dimensions. Further, OLAP servers have to be very efficient (aka a proper design is paramount) in regards to storing and processing multidimensional data.

Summary

The overarching issue with data mining is the need for a repository of detailed (level) data. The detailed level is needed as the variance of aggregated data tends to understate the true variability of the source data, and thereby invalidating many of the assumptions of commonly used (statistical) methods. The low level data often results in Terabytes of granular level data that carries a significant cost to maintain. These issues are compounded by the need for robust data mining tools that often come with very high software licensing costs, and the need for highly skilled (and expensive) labor. The risk of using less-skilled labor is that these analysts often do not fully comprehend what data mining techniques are appropriate for a given problem, or even if the proper technique is chosen, tend to not understand the assumptions of the techniques and the risks to validity and reliability when these assumptions are violated. As a result, less-skilled data miners tend to identify irrelevant and insignificant findings. Due to the labor and high cost issues, next to science and research, data mining (at any significant level) has so far been an approach chosen only by primarily large corporations that are engaged in high-volume transactional activities, or for micro/macro economists at banks, insurance companies, universities, or the government. Data mining may get more mainstream though as the costs for cloud based (parallel) computing solutions are already very reasonable, and tools such as R (an open source programming language for statistical computing and graphics) are becoming more and more popular . As a matter of fact, R already provides a powerful (low-cost) platform for data mining but nevertheless, the labor issues discussed above obviously still exist.

References

Inmon, B., " Building the Data Warehouse", 4th Edition, Wiley Publishing, 2005

Breslin, M., "Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models", Business Intelligence Journal, 2004

Kimball, R. and M. Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, New York: John Wiley & Sons, 2000.

R. Agrawal, T. Imilienski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. of the ACM SIGMOD Int'l Conf. On Management of data, May 1993.

R. Agrawal, J. Shafer, "Parallel Mining of Association Rules," IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, Dec. 1996.

T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp. 103--114, Montreal, Canada, 1996.

S. Brin, R. Motwani, et al, "Dynamic Itemset Counting and Implication Rules for Market Basket Data," SIGMOD Record (SCM Special Interset Group on Management of Data), 26,2, 1997.

S. Chaudhuri, "Data Mining and Database Systems: Where is the Intersection," Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 1997.

M. Chen, J. Han, and P. Yu, "Data Mining: An Overview from a Database Prospective", IEEE Trans. Knowledge and Data Engineering, 8, 1996.

J. Han and M. Kamber, "Data Mining: Concepts and Techniques," Morgan Kaufman Publishers, 2000.

A.K. Jain and R.C. Dubes, "Algorithms for Clustering Data", Prentice-Hall, Englewood Cliffs, NJ, 1988.

Bennell, J.,D. Crabbe, S. Thomas, and O. Gwilym, "Modeling Sovereign Credit Ratings: Neural Networks versus Ordered Probit," Expert Systems with Applications., 2006

Fadalla, A., Lin, Chien-Hua. "An Analysis of the Applications of Neural Networks in Finance", Interfaces 31: 4 July- August 2001