23

Data Mining in Information Analysis and Delivery

Objectives:

- *Information analysis* is an umbrella term that applies to a multitude of techniques for extracting from massive quantities of information various types of important, interesting, or unexpected phenomena.
- The data mining component prepares data for analysis by the automated deduction component. The automated deduction system may discover equivalences that can be used by the data mining component to simplify its search.
- Automated alerts are analytical agents that are designed to automatically find managerially interesting and important information in a database. Alerts can be a powerful analytical tool to keep managers informed as to critical problems and important business opportunities.
- The implications of "alerts run rampant" lead to the ability of the alerts system to provide actionable knowledge to the organization.
- The concept of intelligent information delivery (IID) systems is that it forms a metalayer on top of an alert, or other type of data mining system. The IID monitors the alerts produced and decides which information is most critical to bring to users' attention.
- Information Discovery, Inc. is the leading provider of large-scale data mining oriented decision support software and solutions, introducing pattern management with its breakthrough Pattern Warehouse^{TM} technology.
- **The Data Mining Techniques have been used to discover patterns; the machine-man approaches provide interactive access to the patterns.**
- The modern data mining techniques rely on pattern distillation, rather than data retention. Pattern distillation can be classified into logical, equational, and cross-tabulation methods.

S. Sumathi and S.N. Sivanandam: *Data Mining in Information Analysis and Delivery*, Studies in Computational Intelligence (SCI) **29**, 591–613 (2006)
www.springerlink.com

Abstract. The Information analysis and delivery, and a characterization of data mining technologies and processes by Information discovery Inc., case study are the subject of this chapter.

Information analysis is an umbrella term that applies to a multitude of techniques for extracting important, interesting, or unexpected phenomena from massive quantities of information various types.

In this chapter, we examine the implications of "alerts run rampant" on the ability of the alerts system to provide actionable knowledge to the organization. We then provide a simple example of an intelligent information delivery (IID) mechanism, that functions as a metalayer to the alerts system. The IID layer evaluates the importance and criticality of alert-created information across all alerts in the system. It then decides on the disposition of specific pieces of information. Finally, we describe how the IID layer can be used as a mechanism to derive knowledge out of analyzed information from data mining systems, in general. This chapter gives information regarding the data mining in information analysis and delivery.

23.1 Information Analysis: Overview

The analysis of information is an area of computer science rapidly growing in importance. Because the information of interest is of a wide variety of nature – and because the type of phenomena, which we seek, varies and often is ill defined, many diverse technologies must be developed and applied in novel ways.

It is often convenient to view information analysis as involving three main steps: data acquisition, information extraction and representation, and analysis.

23.1.1 Data Acquisition

Data of a variety of natures is acquired from a possibly large number of diverse sources. Examples of data and their sources include:

Structured Data

Computer audit trails, financial data, attributes of a complex system, and, generally, data from existing database systems may be subjected to a variety of analyses in an attempt to detect behaviors such as intrusion, fraud, or system malfunction.

Numerical and scientific data from an important subclass of structured data, a subclass whose analysis warrants special consideration. For example, autonomous sensors play an important role in safeguard and nonproliferation applications. While the data produced may be structured (i.e., in a prespecified format with well-defined features), the challenges differ from the above-mentioned sources in several ways. For example, these data may include real-valued vectors of variable length and data that is of a temporal nature (where change of state often is the critical component of the analysis).

It also often is the case that a large amount of complex metadata (e.g., scientific formulas and other types of rules) is required to capture the semantics of this type of data.

Images

Satellite and other types of image data are important for such applications as nonproliferation, climatology, and environmental studies. Such digitized images introduce a variety of new information attributes, such as three-dimensional special and four-dimensional time-space relationships.

Free Text Data

\ Documents, reports, technical articles, and articles from the popular press contain a wealth of information to mine. These sources present a particularly formidable challenge. Though not as difficult as natural language understanding, useful analyses well require context to establish semantics, similarities of topics, patterns of usage, and relevance to target queries.

23.1.2 Extraction and Representation

A crucial aspect of the analysis system is the representation, storage, and retrieval of the information under study. Rather than develop distinct analysis techniques for each type of data we might encounter, the best approach, we argue, is to represent and exploit the salient features of data within a common data model and to develop a uniform analysis methodology that operates upon this common model. Once a suitable representation is chosen, extraction tools are defined for each type of data source to map data from the form gathered into the common representation and to store the resulting data in the underlying database.

The data model devised must be sufficiently rich and flexible to support that variety of data we expect, and it also must be capable of supporting efficiently sophisticated analyses against massively large data sets, including retrieval operations required for data mining. To address these issues, we are interested in customizing and adapting one or more data models well studied in computer science so as to be suitable for the problem. Adaptations may include support for statistical analyses, expert rule bases, and axiom systems; complex hierarchical relationship such as/or relationships; and the identification of data equivalence classes.

23.1.3 Information Analysis

Information analysis requires a suite of sophisticated tools, including:

Data mining tools for discovering and prioritizing potentially interesting information: The research in data mining both explores foundational issues

and seeks to apply the results by incorporating the data exploration methodology and algorithmic advances into an experimental software system.

The data mining foundation we have built is based on "information prioritization," a problem model where we are presented with a large number of data points, which must be prioritized. The prioritization produced allows one to pursue items from the highest to lowest ranked until time, money, or interest is exhausted. Another defining characteristic of our work is that we have developed methods, which perform analyses even in the most information-deprived environments (for example, environments lacking labeled training sets, expert rules, and feedback).

The challenges presented by future research include the mining of temporal, special, and textual patterns; the construction of abstract statistical models of undesirable behavior in new domains of interest; and the integration of our statistical data mining techniques with automated reasoning techniques. This last project is considered below.

Automated deduction tools for reasoning about data: An automated deduction tool allows us to make references and draw logical conclusions about retrieved data based on general data based on general rules and relationships. The work in automated deduction focuses on the development of inference rules and strategies needed to reason effectively about problems from mathematics and logic and for application areas such as the analysis of information. In order to develop an effective reasoning component for an information analysis system, we are working on problems such as the following:

Data to be interpreted will be at various levels of abstraction, ranging from raw sensor data to high-level terms that are the output of other data mining and analysis steps. We are working to enhance the inference and search capabilities of the automated deduction system in order to be able to reason effectively at multiple levels of abstraction.

One aspect of the analysis of information is to search for sets of observables that are considered to be evidence for activities or conditions of interest. It often will be the case that several sets of observables will be considered to be "equivalent" evidence for some activity or condition. We are developing strategies to account for equivalence classes in the search for evidence. Specifically, we are attempting to use the automated deduction system to search for evidence using functional rather than strictly syntactic matching criteria.

Integration of data mining and automated reasoning: Data mining and automated reasoning techniques traditionally are applied to quite different types of information analysis problems. We believe each technology can benefit from the other, and together can form the core of powerful information analysis architecture. The directions this integration may take include, for example, the following.

The statistically based data mining system operates as a last line of defense, inspecting data, which the automated deduction system does not flag as violating constraints specified in its rule base.

The data mining component prepares data for analysis by the automated deduction component. For example, the data mining component can provide data at a level of abstraction well suited for analysis by the automated deduction component and can prioritize this information to help guide the automated deduction system's search. Further, clustering of data values suggested by the data mining component can affect the application of inference rules (e.g., whether or not a rule fires).

23.2 Intelligent Information Delivery – Case Study

Recently, users of advanced information systems have begun to realize the value of incorporating automated alerts into their systems. Automated alerts are analytical agents that are designed to automatically find managerially interesting and important information in a database. The agents operate without user intervention, but report important information back to users whenever critical events are found in the database. This case study is taken from Analytic Solutions for Ceres Integrated Solutions, NY.

Alerts can be a powerful analytical tool to keep managers informed as to critical problems and important business opportunities. All it takes, it seems, is having the correct underlying sources of data for the alerts to operate on and then creating the appropriate set of alerts. The problem with automated alerts is that the volume of information automatically returned to the user can quickly become overwhelming. Analysis is easy. Knowledge is hard. A vital component in the development of knowledge is the recognition that an event is something that is important for the recipient to know about; in fact, that it is more important to know about than other significant events.

23.2.1 Alerts Run Rampant

Data mining systems, in general, are geared toward the analysis of vast amounts of data and are designed to produce large quantities of analyzed information that, essentially, have to be sifted through and analyzed before they become useful as business decision making aids. This fact can become a critical issue when applied to automated alert systems. These systems are designed to perform data mining automatically and continuously. An example from the consumer packaged goods (CPG) industry will show the magnitude of the problem.

The CPG industry has, for many years, had the availability of rich sources of data. For most grocery products, vendors such as A.C. Nielsen and IRI sell sales scanner data that tracks all competitive products in a category, by UPC (the individual product, the lowest level information that manufacturers track for sales purposes), in each of fifty or more markets. A typical category can have 1,200 or more UPCs in each market. Most packaged goods manufacturers receive updates weekly. This means that *a single alert measure* can track 60,000 possible events each week.

There are, however, many more than a single important alert measure that packaged goods manufacturers need to track. Some key alert measures for the packaged goods industry include short-term market share changes for all UPCs in the market, trends in market share changes, introductions of new competitive items, competitor price changes, and changes in competitive promotional activity. Competitive activity is inferred by observing such things as retailer promotion pricing actions, increased levels of distribution for a competitor's UPCs, and retailer promotions, such as increases in point of purchase displays, major ads, and coupon activity. Since packaged goods marketers typically micromarket, each UPC has to be tracked, by market, for each alert measure.

There can be easily hundreds of thousands (or many more) events being tracked automatically. Because the CPG marketing environment is highly competitive and dynamic, there can easily be thousands of events that set off trigger conditions to alert a user. The situation gets even more overwhelming when we consider the fact that advanced marketing analysis systems in the CPG industry sometimes also embed sophisticated data mining technology that automatically analyzes causal factors associated with some alert conditions. The ensuing report, then, includes not only alert information, but also details of an analysis. The information overload that results can set up a condition where the user has to either spend all his or her time on reviewing the results of alerts, or ends up just ignoring the output of the alerts system.

23.2.2 What an Intelligent Information Delivery System is

An intelligent information delivery system is essentially a knowledge-based system that:

1. Monitors and intercepts all outputs from the alert system,
2. Performs some analyses on the set of alerts that evaluates alert output based on the totality of what is known, and
3. Applies business rules to the output of the process to determine which outputs from the alert system are critically important for a user to know about.

The IID system functions as a meta-analysis layer for the alerts system. It evaluates alert-created information across all alerts in the system. Based on results of analysis and the rules contained in the system, it decides on the relative importance of the various alerts. The IID system also decides whom, that is, which users should receive specific pieces of information. The IID knowledge base contains rules related to managerial objectives that guide the selection of output for individual users. Development of this knowledge base is based on conducting knowledge engineering sessions with key business users to determine specific business rules to incorporate in the system. Actual application to individual users is based on creating user settings stored in a database table and accessed by the meta-analysis layer.

23.2.3 Simple Example of an Intelligent Information Delivery Mechanism

As an illustration, we provide a simple example of an IID system. The system has all three components:

1) An alert monitor
2) Meta-analysis capabilities, and
3) A business rule knowledge base.

The IID system is designed to support a consumer packaged goods alert system.

Alert Monitor

The alert system in the example polls the database and performs its analyses weekly to coincide with database updates based on marketplace scanner data purchased from IRI or A.C. Nielsen. The focus of the system is on information contained in this data. The IID Alert Monitor intercepts all alerts that are in its domain of knowledge. No alerts are passed onto users at this time. The Monitor holds the alert information until all the alerts have finished processing the updated information in the database.

Meta-Analysis Layer

The Meta-Analysis Layer synthesizes results of the alert process and performs further analysis. For example, it will do cross-market analysis of alerts to discover whether or not an alert condition is specific to a single market, or whether it reflects a more general condition. It will also check if the alert is a one-time occurrence or whether there has been a pattern of these conditions over time.

The Meta-Analysis Layer also makes an assessment of the overall volatility in the marketplace. Highly volatile markets can be expected to have many fluctuations in market share, retailer promotional activity, and competitor product introductions. Some alerts that might be considered significant in a nonvolatile market, after this assessment, may no longer be important enough to report.

Business Rules Knowledge Base

The Business Rules Knowledge Base contains rules developed by conducting in-depth interviews with key business managers in the organization responsible for taking action based on the results of the alert system. The business rules are mapped against the meta-analyzed alerts to determine which alerts are really important to know about, and who receives which alerts.

Consumer packaged goods marketers often focus on Brand Development Index (BDI) and Category Development Index (CDI) measures when running their business. BDI ranks markets as to the strength of the brand in that market. Markets where the brand has a high market share, high BDI markets, are ranked ahead of markets where the brand has a low market share. CDI ranks markets as to the strength of the overall category in the market. Markets where category sales are high (high CDI markets) are ranked ahead of markets where category sales are low.

Brand strategies often incorporate the relative importance of these measures and how to use them. For example, a brand strategy that focuses on increasing market share may often focus on high opportunity markets – those with high CDI, but low BDI. A brand strategy that focuses on maintaining current brand strength may focus on high BDI markets. An important element, then, of the business rules knowledge base may be the incorporation of rules related to BDI and CDI.

An emphasis on BDI and CDI could lead to the following rules:

Rule 1:
IF Brand Strategy is to focus on High Opportunity Markets
THEN Alerts should be ranked by the CDI of the market they relate to
Rule 2:
IF Brand Strategy is to focus on High Brand Strength Markets
THEN Alerts should be ranked by the BDI of the market they relate to

There will be other rules relating to other strategies that incorporate additional factors. Other rules in the knowledge base may relate to results of cross-market analysis, prioritization of negative information about the marketers brand, prioritization of positive information about competitors' brands, priority given to trends versus one-time events, and thresholds related to when to consider competitive activity important.

Rule 3:
IF Cross-market analysis shows an over all strong pattern THEN this is an important alert
Rule 4:
IF UPC is for our Brand
AND there is a downward trend in market share THEN this is an important alert
Rule 5:
IF UPC is for Key Competitors' Brand AND there has been a Highly Significant increase in market share THEN this is an important alert
Rule 6:
IF Market share change for a UPC is > twice the average Market Share change THEN This is a Highly Significant increase in market share
Rule 7:
IF UPC is for Key Competitors' Brand AND there has been at least a three-month trend in price decreases THEN this is an important alert

The above is just a small sample of the business rules knowledge base that would be developed for even a simple intelligent Information delivery system. However, even a simple IID system can reduce the volume of output of alerts from hundreds of pages containing thousands of analyses to just the few most important findings.

23.3 A Characterization of Data Mining Technologies and Processes – Case Study

While myriad of approaches to data mining has been proposed, just a few fundamental techniques form the basis of most systems. Here we provide a characterization of these fundamental technologies and outline a framework for viewing data mining processes in terms of three families of techniques. Information Discovery, Inc. is the leading provider of large-scale data mining-oriented decision support software and solutions, introducing pattern management with its breakthrough Pattern Warehouse^{TM} technology and offering two comprehensive product suites. The Data Mining Suite^{TM} products directly access very large multitable SQL repositories to find powerful multiform patterns. The Knowledge Access Suite^{TM} incrementally stores these premined patterns in a Pattern Warehouse^{TM} for access by business users. The company also offers a wide range of discovery and data mining solutions, strategic consulting, and warehouse architecture design, as well as customized solutions for banking, financial services, retail, customer packaged goods, manufacturing, and web log analysis. This case study is taken from Information Discovery, Inc., Marketing Communications, CA.

The past year has seen a dramatic surge in the level of interest in data mining, with business users wanting to take advantage of the technology for a competitive edge. The IT departments in most Fortune 500 companies are suddenly tasked to respond to deployment questions relating to data mining. The growing interest in data mining has also resulted in the introduction of a myriad of commercial products, each described with a set of terms that sound similar, but in fact refer to very different functionality and based on distinct technical approaches.

The IT managers charged with the task of selecting a decision support system often face a challenge in responding to the needs of the business users because the underlying concept of data mining are far more complex than traditional query and reporting, and to add to the pressure the needs of the business users are usually urgent, requiring decisions that need to be made quickly.

However, while various approaches to data mining seem to offer distinct features and benefits, in fact just a few fundamental techniques form the basis of most data mining systems. Here we provide a characterization of these fundamental technologies, outline a framework for viewing data mining processes, and provide suggestions for the suitability of each technology for various business applications.

We define data mining as "a decision support process in which we search for patterns of information in data." This search may be done just by the user (i.e., by performing queries; in which case it is quite hard) or may be assisted by a smart program that automatically searches the database for the user and finds significant patterns. Once found, the information needs to be presented in a suitable form, with graphs, reports etc. The approaches we discuss here characterize the data mining paradigms from the point of view of discovery and prediction, not from the view point of machine–man pattern access. As in Parsaye, 1997, after the techniques discussed here have been used to discover patterns, the machine–man approaches provide interactive access to the patterns.

23.3.1 Data Mining Processes

Traditionally, there have been two types of statistical analyses: confirmatory analysis and exploratory analysis. In confirmatory analysis, one has a hypothesis and either confirms or refutes it. However, the bottleneck for confirmatory analysis is the shortage of hypotheses on the part of the analyst. In "exploratory analysis," one finds suitable hypothesis to confirm or refute. Here the system takes the initiative in data analysis, not the user.

The concept of "initiative" also applies to multidimensional spaces. In a simple OLAP access system, the user may have to think of a hypothesis and generate a graph. But in OLAP data mining, the system thinks of the questions by itself. We use the term *data mining* to refer to the automated process of data analysis in which the system takes the initiative to generate patterns by itself. From a process-oriented view, there are three classes of data mining activity: *discovery, predictive modeling, and forensic analysis.*

- *Discovery* is the process of looking in a database to find hidden patterns without a predetermined idea or hypothesis about what the patterns may be. In other words, the program takes the initiative in finding what the interesting patterns are, without the user thinking of the relevant questions first. In large databases, there are so many patterns that the user can never practically think of the right questions to ask. The key issue here is the richness of the patterns that can be expressed and discovered and the quality of the information delivered – determining the power and usefulness of the discovery technique.

As a simple example of discovery with system initiative, suppose we have a demographic database of the US. The user may take the initiative to ask a question from the database, such as "what is the average age of bakers?" The system may then print 47 as the average age. The user may then ask the system to take the initiative and find something interesting about "age" by itself. The system will then act as a human analyst would. It will look at some data characteristics, distributions, etc. and try to find some data densities that might be away from ordinary. In this case the system may print the rule: "IF Profession = Athlete THEN Age <30, with a 71% confidence."

This rule means that if we pick 100 athletes from the database, 71 of them are likely to be under 30. The system may also print: "IF Profession = Athlete THEN Age, <60, with a 97% confidence." This rule means that if we pick 100 athletes from the database, 97 of them are likely to be under 60. This delivers information to the user by distilling pattern from data.

- In *predictive modeling* patterns discovered from the database are used to predict the future. Predictive modeling thus allows the user to submit records with some unknown field values, and the system will guess the unknown values based on previous patterns discovered from the database. While discovery finds patterns in data, predictive modeling applies the patterns to guess values for new data items.

To use the example above, once we know that athletes are usually under 30, we can guess someone's age if we know that they are an athlete. For instance, if we are shown a record for John Smith whose profession is athlete by applying the rules we found above, we can be over 70% sure that he is under 30 years old, and we can be almost certain that he is under 60. Note that discovery helps us find "general knowledge," but prediction just guesses that value for the age of a specific individual. Also note that in this case the prediction is "transparent" (i.e., we know why we guess the age as under 30). In some systems the age is guessed, but the reason for the guess is not provided, making the system "opaque."

- *Forensic analysis* is the process of applying the extracted patterns to find anomalous or unusual data elements. To discover the unusual, we first find what is the norm, then we detect those items that deviate from the usual within a given threshold. Again, to use the example above, once we notice that 97% of athletes are fewer than 60, we can wonder about the 3% who are over 60 and still listed as athletes. These are unusual, but we still do not know why. They may be unusually healthy or play sports where age is less important (e.g., golf) or the database may contain errors, etc. Note that discovery helps us find "usual knowledge," but forensic analysis looks for unusual and specific cases.

Each of these processes can be further classified. There are several types of pattern discovery such as If/Then rules, associations, etc. While the rules discussed above have an IF-THEN nature, association rules refer to item groupings (e.g., when someone buys one product at a store, they may buy another product at the same time – a process usually called *market basket analysis*). The power of a discovery system is measured by the types and generality of the patterns it can find and express in a suitable language.

23.3.2 Data Mining Users and Activities

It is necessary to distinguish the data mining processes discussed above from the data mining activities in which the processes may be performed, and

the users who perform them. First, the users. Data mining activities are usually performed by three different classes of users: *executives, end users, and analysts.*

- *Executives* need top-level insights and spend far less time with computers than the other groups – their attention span is usually less than 30 minutes. They may want information beyond what is available in their executive information system (EIS). End users and analysts usually assist executives.
- *End users* know how to use a spreadsheet, but they do not program – they can spend several hours a day with computers. Examples of end users are sales people, market researchers, scientists, engineers, physicians, etc. At times, managers assume the role of both executive and end user.
- *Analysts* know how to interpret data and do occasional computing but are not programmers. They may be financial analysts, statisticians, consultants, or database designers. Analysts usually know some statistics and SQL.

These users usually perform three types of data mining activity within a corporate environment: *episodic, strategic, and continuous data mining.*

- In episodic mining we look at data from one specific episode such as a specific direct marketing campaign. We may try to understand this data set, or use it for prediction on new marketing campaigns. Episodic mining is usually performed by analysts.
- In strategic mining we look at larger sets of corporate data with the intention of gaining an overall understanding of specific measures such as profitability. Hence, a strategic mining exercise may look to answer questions such as: "where do our profits come from?" or "how do our customer segments and product usage patterns relate to each other?"
- In continuous mining we try to understand how the world has changed within a given time period and try to gain an understanding of the factors that influence change. For instance, we may ask: "how have sales patterns changed this month?" or "what were the changing sources of customer attrition last quarter?"

Obviously continuous mining is an on-going activity and usually takes place once strategic mining has been performed to provide a first understanding of the issues. Continuous and strategic mining are often directed toward executives and managers, although analysts may help them here. As we see later, different technologies are best suited to each of these types of data mining activity.

23.3.3 The Technology Tree

The top-level dichotomization of the data mining technologies can be based on the retention of data; that is, do we still keep or need the data after we

have mined it? In most cases, not. However, in some early approaches much of the data set was still maintained for future pattern matching. Obviously, these retention-based techniques only apply to the tasks of predictive modeling and forensic analysis, and not knowledge discovery since they do not distill any patterns.

As one would expect, approaches based on data retention quickly run into problems because of large data sets. However, in some cases predictive results can be obtained with these techniques and for the sake of completeness. The approaches based on pattern distillation fall into three categories: logical, cross-tabulatioal, and equational.

Not all approaches based on pattern distillation provide knowledge, since the patterns may be distilled into an "opaque" language or formalism not easily readable by humans such as very complex equations. Hence, some of these approaches produce "transparent" and understandable patterns of knowledge, others just produce patterns used for opaque prediction.

Data Retention

While in pattern distillation we analyze data, extract patterns and then leave the data behind, in the retention approaches the data is kept for pattern matching. When new data items are presented, they are matched against the previous data set.

A well-known example of an approach based on data retention is the "nearest neighbor" method. Here, a data set is kept (usually in memory) for comparison with new data items. When a new record is presented for prediction, the "distance" between it and similar records in the data set is found, and the most similar (or nearest neighbors) are identified.

For instance, given a prospective customer for banking services, the attributes of the prospect are compared with all existing bank customers (e.g., the age and income of the prospect are compared with the age and income of existing customers). Then a set of closest "neighbors" for the prospect are selected (based on closest income, age, etc.).

The term *K-nearest neighbor* is used to mean that we select the top K (e.g. top 10) neighbors for the prospect. Next, a closer comparison is performed to select which new product is most suited to the prospect, based on the products used by the Top K (e.g., top 10) neighbors. Of course, it is quite expensive to keep all the data, and hence sometimes just a set of "typical cases" is retained. We may select a set of 100 "typical customers" as the basis for comparison. This is often called *case-based reasoning*.

Obviously, the key problem here is that of selecting the "typical" customers as cases. If we do not really understand the customers, how can we expect to select the typical cases, and if the customer base changes, how do we change the typical customers?

Another usually fatal problem for these approaches has to do with databases with a large number of non-numeric values (e.g., many supermarket

products or car parts). Since distances between these non-numeric values are not easily computed, some measures of approximation need to be used – and this is often hard to come by. And if there are many non-numeric values, there will be too many cases to manage.

Pattern Distillation

These technologies extract patterns from a data set then use the patterns for various purposes. Naturally, the first two questions to ask here are: What types of patterns can be extracted and how are they represented?

Obviously, patterns need to be expressed within formalism and a language. This choice given rise to three distinct approaches: logic, equations, or cross-tabulations. Each of these approaches traces its historical roots to a distinct mathematical origin.

For instance, let us consider the distinction between equations and logic. In an equational system operators such as "plus" and "times" may be used to relate variables together, e.g., $(a * X) + b)$ while in a logical system the key operators are conditional (e.g., IF $6 < X < 7$ THEN $1 < y < 2$).

Logic can deal with both numeric and non-numeric data. Equations require all data to be numeric, while cross-tabulations are the reverse and only work on non-numeric data; a key source of problems. But more importantly, equations compute distances form surfaces (such as lines) while cross-tabs focus on co-occurrences.

Neural networks are opaque equational techniques since internally they compute surfaces within a numeric space. As data is repeatedly fed into the network, the parameters are changed so that the surface becomes closer to the data point.

When discussing data mining, it is necessary to distinguish between "directed analysis" and free form "roams" through the database. In directed analysis, also called supervised learning, there is a "teacher" who teaches the system, by saying when prediction was correct or incorrect. Here the data has a specific column that is used as the goal for discovery or prediction.

In unsupervised learning, the system has number teacher, but simply tries to find interesting clusters of patterns within the data set. Most of the business applications of data mining involve directed data mining, while unsupervised discovery can sometimes be used for data segmentation or clustering (e.g., finding classes of customers that group together).

Logical Approaches

Logic forms the basis of most written languages and is essential for "left-brain" thinking. Patterns expressed in logical languages are distinguished by two main features: on one hand they are readable and understandable, on the

other hand they are excellent for representing crisp boxes and groupings of data elements.

The central operator in a logical language is usually a variation on the well known If/Then statements (e.g., "If it is raining, then it is cloudy"). However, let us note that while the most common form of logic is conditional logic, often we may need to use other logical forms such as association logic with When/Also rules, (e.g., When paint is purchased, also a paint brush is purchased. While the propositional and predicate logic (i.e., conditional logics) are best known, other forms of logic (e.g., variational and trend logics) are also useful in business data analysis.

Conditional logic systems can be separated into two distinct groups: rules and decision trees. Conditional rules may be implemented by induction or genetic algorithms and there are several approaches for generating decision trees (e.g., CART, CHAID, C4.5)

Rules

Logical relationships are usually represented as rules. The simplest types of rules express conditional or association relationships. A conditional rule is a statement of the form:

If Condition1
Then Condition2

For instance, in a demographic database we may have a rule: If Profession = Athlete Then Age <30. Here we compare the values within fields of a given table (i.e., we have an "attribute-value" representation). Here Profession is the attribute and Athlete the value. Another example of an attribute-value expression is "State = Arizona", where State is the attribute and Arizona the value.

Conditional rules usually work on tables with attributes (i.e. fields) and values, such as below.

Name	Profession	Age
John	Smith	Athlete
27	...	...
...	...	...

Rules may easily go beyond attribute-value representations. They may have statements such as "Shipping_State = Receiving_State". Here, in attribute logic, we compare the values of two fields, without explicitly naming any values. This relationship cannot be stated by decision trees or cross-tabs.

Affinity logic is distinct from conditional logic both in terms of the language of expression and the data structures it uses. Affinity analysis (or association analysis) is the search for patterns and conditions that describe how various items "group together" or "happen together" within a series of events or transactions. An affinity rule has the form:

When Item1 Also Item2

An example of this is, When Paint, Also Paint Brush. A simple affinity analysis uses a transaction table such as:

Transaction #	Item
123	Paint
123	Paint Brush
123	Nails
124	Paint
124	Paint Brush
124	Wood
125	–

to identify items that group together within transactions. Here, the transaction# field is used to group items together, while the item# field includes the entities being grouped. In this example, the affinity for transactions 123 and 124 is the pair (Paint, Paint Brush). Please note that this is a distinct data structure from the conditional logic rule above.

As pointed out in "Data Mining with OLAP Affinities," "flat affinities" need to be extended to dimensional or OLAP affinities for better results. A dimensional affinity has the form:

Confidence = 95%
IF
Day = Saturday
WHEN
Item = Paint Brush
ALSO
Item = Paint

Here logical conditions and associations are combined. This form of hybrid structure delivers the real power of transparent logic. Rules have the advantage of being able to deal with numeric and non-numeric data in a uniform manner. When dealing with numeric data, some approaches have to break numeric fields into "code" or specific values. This may effectively remove all numeric consideration form the codes, thus resulting in the loss of patterns. For instance, the field Age may need to be broken into 3 ranges (1–30), (31–60), (61–100), corresponding to young, middle-aged, and old. Of course, the data may hold patterns that overlap any of these ranges (e.g., the range 27–34) may be very significant for some patterns and any approach based on code assignment will miss these.

Rules can also work well on multidimensional and OLAP data because they can deal with ranges of numeric data and their logical formats allows their patterns to be merged along multiple dimensions.

Rules do at times look like decision trees, but despite the surface level similarity they are a distinct and different technique. This is easy to see when we consider the fact that decision trees do not express associations, or attribute-based patterns such as "Shipping_State = Receiving_State" where the values of two fields are compared, without explicitly naming any values.

The main weakness of rules stems from their inability to deal with smooth surfaces that typically occur in nature (e.g., finger-print identification, facial recognition). These "naturally smooth" surfaces are often best approximated by equational approaches such as neural nets.

Below we review two approaches to rule generation, namely induction and genetic algorithms. However, these are not the only approaches to data mining with rules. Some approaches try to precompute every possible rule that a data set could include. In these cases, only a few columns of data may be used because the logical space is so large. Hence we do not review these since they are not practical for large-scale applications.

Rule Induction

Rule induction is the process of looking at a data set and generating patterns. By automatically exploring the data set, the induction system forms hypotheses that lead to patterns.

The process is in essence similar to what a human analyst would do in exploratory analysis. For example, given a database of demographic information, the induction system may first look at how ages are distributed, and it may notice an interesting variation for those people whose profession is listed as professional athlete. This hypothesis is then found to be relevant and the system will print a rule such as.

IF Profession = Athlete
THEN Age <30.

This rule may have "a confidence" of 70% attached to it. However, this pattern may not hold for the ages of bankers or teachers in the same database.

We must also distinguish between fuzzy and inexact rules. Inexact rules often have a "fixed" confidence factor attached to them, i.e., each rule has a specific integer or percentage (such as 70%) representing its validity. However, the confidence in fuzzy rules can vary in terms of the numeric values in the body of the rule; for instance the confidence may be proportional to the age of a person and as the age varies so does the confidence. In this way fuzzy rules can produce much more compact expression of knowledge and lead to stable to behavior.

Rule induction can discover very general rules, which deal with both numeric and non-numeric data. And rules can combine conditional and affinity statements into hybrid patterns. A key issue here is the ability to go beyond flat databases and deal with OLAP patterns.

Genetic Algorithms

Genetic algorithms also generate rules from data sets but do not follow the exploration-oriented protocol of rule induction. Instead, they rely on the idea of "mutation" to make changes in patterns until a suitable form of pattern emerges via selective breeding. The genetic crossover operation is in fact very

similar to the operation breeders use when they cross-breed plants and/or animals. The exchange of genetic material by chromosomes is also based on the same method. In the case of rules, the material exchange is a part of the pattern the rule describes.

Let us note that this is different from rule induction since the main focus in genetic algorithms is the combination of patterns from rules that have been discovered so far, while in rule induction the main focus of the activity is the dataset. Genetic algorithms are not just for rule generation and may be applied to a variety of other tasks to which rules do not immediately apply, such as the discovery of patterns in text, planning and control, system optimization, etc.

Decision Trees

Decision trees express a simple form of conditional logic. A decision tree system simply partitions a table into smaller tables by selecting subsets based on values for a given attribute. Based on how the table is partitioned, we get a different decision tree algorithm such as CART, CHAID, and C4.5.

For example, consider the table:

Manufacturer	**State**	**City**	**Product Color**	**Profit**
Smith	CA	Los Angeles	Blue	High
Smith	AZ	Flagstaff	Green	Low
Adams	NY	NYC	Blue	High
Adams	AZ	Flagstaff	Red	Low
Johnson	NY	NYC	Green	Avg
Johnson	CA	Los Angeles	Red	Avg

A decision tree from this table can be drawn. This decision tree first selected the attribute State to start the partitioning operation, then the attribute Manufacturer. Of course, if there are 100 columns in the table, the question of which attribute to select first becomes crucial. In fact, in many cases, including the table above, there is no best attribute, and whichever attribute the tree chooses there will be information loss. For example the two facts:

(a). "Blue products are high profit."
(b). "Arizona is low profit."

can never be obtained from the table above with a decision tree. We can either get fact (a) or fact (b) from the tree, not both, because a decision tree selects one specific attribute for partitioning at each stage. Rules and cross-tabs, on the other hand, can discover both of these facts.

23.3.4 Cross-Tabulation

Cross-tabulation is a very basic and simple form of data analysis, well known in statistics, and widely used for reporting. A two dimensional cross-tab is similar to a spreadsheet, with both row and column headings as attribute values. The cells in the spreadsheet represent an aggregate operation, usually the number of co-occurrences of the attribute values together. Many cross-tabs are effectively equivalent to a 3D bar graph that displays co-occurrence counts. Consider the table in the previous section. A cross-tab for the profit level could look as follows:

CA	**AZ**	**NY**	**Blue**	**Green**	**Red**	**Profit**
1	0	1	2	0	0	High
1	0	1	0	1	1	Avg
0	2	0	0	1	1	Low

Here we have not included the fields Manufacturer and City because the cross-tab would look too large. However, as is readily seen here, the fact that the count of co-occurrence of Blue and High is above the others indicates a stronger relationship.

When dealing with a small number of non-numeric values, cross-tabs are simple enough to use and find some conditional logic relationships (but not attribute logic, affinities, or other forms of logic). Cross-tabs usually run into four classes of problems: first when the number of non-numeric values goes up, second when one has to deal with numeric values, goes up, second when one has to deal with numeric values, third when several conjunction are involved, and fourth when the relationships are not just based on counts.

Agents and belief networks are variations on the cross-tab theme and are discussed next.

Agents

The term *agent* is sometimes used (among its other uses) to refer to cross-tabs that are graphically displayed in a network and allow some conjunctions (i.e., ANDs). In this context the term *agent* is effectively equivalent to the term *field-value pair*.

Like other cross-tab techniques, when dealing with numeric values, agents have to break the numbers into fixed "codes" (e.g., break Age into three age classes: (1–30), (31–60), (61–100)). Of course, the data may hold patterns that overlap any of these ranges (e.g., the range (27–34) and these will not be detected by the agent). And if the ranges selected are too small, there will be too many of them and larger patterns will be missed. Moreover, this inability to deal with numeric data causes problems with multidimensional data.

Belief Networks

Belief networks (sometimes called *causal networks*) also rely on co-occurrence counts, but both the graphic rendering and the probabilistic representation are slightly different from agents. Belief networks usually illustrated using a graphical representation of probability distributions (derived from counts). A belief network is thus a directed graph, consisting of nodes (representing variables) and arcs (representing probabilities dependencies) between the node variables. Each node contains a conditional probability distribution that describes the relationship between the node and the parents of that node. The belief network graph is acyclic, meaning that there are no cycles.

Equational Approaches

The underlying method of pattern expression in these systems is "surface construction" rather than logical expression or co-occurrence counts. Such systems usually use a set of equations to define a "surface" within a numeric space, then measure distances from this surface for a prediction.

The best known example of such a surface is a straight line in a two-dimensional space with the simple equation $Y = (a * X) + b$ and leads to the well-known approach of linear regression in statistics. As the parameter "a" varies in this equation, the slope of the line changes. Regression works well when the points to be approximated lie on a straight line; it is also possible to use nonlinear equations to approximate smoother surfaces.

When the surfaces are even more complex (e.g., $Y = (X2 + X + (1/X))$), or when there are several dimensions, the ability of humans to understand the equations and surfaces decreases rather quickly. The system becomes opaque or "black box." However, it is still possible to construct such surfaces. In fact, neural nets are known to be "universal approximators" in theory. They can come close to any function. However, the present theory does not specify the practical limits of nets for achieving such approximation on large data sets and most neural net implementations rely on sampling.

The equational approaches almost always require the data set to be all numeric. Non-numeric data needs to be "coded" into numbers (the reverse of what cross-tabs do). This often causes a number of problems, as discussed below.

23.3.5 Neural Nets

Neural nets are a class of predictive modeling system that work by iterative parameter adjustment. Structurally, a neural network consists of a number of interconnected elements (called neurons) organized in layers, which learn by modifying the connection strengths (i.e., the parameters) connecting the layers.

Neural nets usually construct complex equational surfaces through repeated iterations, each time adjusting the parameters that define the surface. After many iterations, a surface may be "internally" defined that approximates many of the points within the data set.

The basic function of each neuron is to: (a) evaluate input values, (b) calculate a total for the combined input values, (c) compare the total with a threshold value, and (d) determine what its own output will be. While the operation of each neuron is fairly simple, complex behavior can be created by connecting a number of neurons together. Typically, the input neurons are connected to a middle layer (or several intermediate layers), which is then connected to an outer layer.

To build a neural model, we first train the net on a "training dataset," then use the trained net to make predictions. We may, at times, also use a "monitoring data set" during the training phase to check on the progress of the training. Each neuron usually has a set of weights that determine how it evaluates the combined strength of the input signals. Inputs coming into a neuron can be either positive (excitatory) or negative (inhibitory). Learning takes place by changing the weights used by the neuron in accordance with classification errors that were made by the net as a whole.

The inputs are usually scaled and normalized to produce a smooth behavior. During the training phase, the net sets the weights that determine the behavior of the intermediate layer. A popular approach is called *backpropagation* in which the weights are adjusted based on how closely the network has made guesses. Incorrect guesses reduce the thresholds for the appropriate connections. Neural nets can be trained to reasonably approximate the behavior of functions on small- and medium-sized data sets since they are universal approximate. However, in practice they work only on subsets and samples of data and at times run into problems when dealing with larger data sets (e.g., failure to converge or being stuck in a local minimum).

It is well known that backpropagation networks are similar to regression. There are several other network training paradigms that go beyond backpropagation, but still have problems in dealing with large data sets. One key problem for applying neural nets to large data sets is the preparation problem. The data in the warehouse has to be mapped into real numbers before the net can use it. This is a difficult task for commercial data with non-numerous values.

Since input to a neural net has to be numeric (and scaled), interfacing to a large data warehouse may become a problem. For each data field used in a neural net, we need to perform scaling and coding. The numeric (and date) fields are scaled. They are mapped into a scale that makes them uniform (i.e., if ages range between 1 and 100 and number of children between 1 and 5, then we scale these into the same interval, such as -1 to $+1$). This is not a very difficult task.

However, non-numeric values cannot easily be mapped to numbers in a direct manner since this will introduce "unexpected relationship" into the

data, leading to errors later. For instance, if we have 100 cities, and assign 100 numbers to them, cities with values 98 and 99 will seem more related together than those with numbers 21 and 77. The net will think these cities are somehow related, and this may not be so.

To be used in a neural net, values for nonscalar fields such as City, State, or Product need to be coded and mapped into "new fields", taking the values 0 and 1. This means that the field State, which may have the 7 values: {CA, NY, AZ, GA, MI, TX, VA), is no longer used. Instead, we have 7 new fields, called CA, NY, AZ, CA, MI, TX, VA each taking the value 0 to 1, depending on the value in the record. For each record, only of these fields has the value 1, and the others have the value 0. In practice, there are often 50 states requiring 50 new inputs.

Now the problem should be obvious. "What if the field City has 1,000 values?" Do we need to introduce 1,000 new input elements for the net? In the strict sense, yes, we have to. But in practice this is not easy, since the internal matrix representation for the net will become astronomically large and totally unmanageable. Hence, by-pass approaches are often used.

Some systems try to overcome this problem by grouping the 1,000 cities into 10 groups of 100 cities each. Yet, this often introduces bias into the system, since in practice it is hard to know what the optimal groups are, and for large warehouses this requires too much human intervention. In fact, the whole purpose of data mining is to find these clusters, not ask the human analysts to construct them.

The distinguishing power of neural nets comes from their ability to deal with smooth surfaces that can be expressed in equations. These suitable application areas are varied and include fingerpoint identification and facial pattern recognition. However, with suitable analytical effort neural net works can also succeed in many other areas such as financial analysis and adaptive control. Eventually, the best way to use neural nets on large data sets will be to combine them with rules, allowing them to make predictions within hybrid architecture.

23.4 Summary

Automated deduction can be used to identify a subset of data for further interpretation using data mining strategies. Information prioritization results obtained by mining a database of successful proofs can be used to develop new search strategies for automated deduction.

The automated deduction system may discover equivalences that can be used by the data mining component to simplify its search. Eventually, the two components may be even more tightly coupled, iterating to extract information at successively higher levels of abstraction, and interacting in a hypothesize-test mode.

The concept of intelligent information delivery systems – systems that form a metalayer on top of an alert, or other type of data mining system was introduced. The IID monitors the alerts produced and decide which information is most critical to bring to users' attention. We have also given a brief example of an IID system.

As data mining systems and systems of alerts that repetitively and automatically analyze information in databases become more prevalent, the problem of what to do with all the answers that come out will become increasingly important. The alternative is that over time, users of these systems will find that the more analysis they receive, the less they end up knowing. Also a case study based on the characterization of data mining technologies and processes is given.

The fundamental techniques used for data mining can be classified into distinct groups, each offering advantages and trade-offs. The modern techniques rely on pattern distillation, rather than data retention. Pattern distillation can be classified into logical, equational, and cross-tabulation methods. The underlying structure of these approaches was discussed and compared. Hybrid approaches are likely to succeed best, merging logic and equations with multidimensional analysis. However, the overstructure of how these techniques are used should be viewed in the context of machine–man interaction.

23.5 Review Questions

1. Define extraction and representation.
2. What is information analysis?
3. Explain with case study intelligent information delivery with its components.
4. How is data mining process characterized?
5. State the data mining users and their activities.
6. What is data retention and pattern distillation?
7. How are logical approaches formed in the information analysis?
8. How are decision tree and cross-tabulation formed in the intelligent information delivery model?