The Emerging Field of Data Mining

Patrick Whalin
University of Minnesota – Morris
Morris, MN 56267

whalinp@mrs.umn.edu

ABSTRACT

Until recently, data analysts would pour over hundreds, maybe even thousands of bits of data looking for underlying patterns that would reliably predict future outcomes. With the advent of the computer age, people have begun using computers to automate the data gathering process and store the information in databases. Computers are so well suited to this task that gigantic databases with terabytes of information have been generated. It is well beyond the scope of the human mind to sort through all of this data and find any useful patterns for predicting future events. The field of data mining has been created to deal with this new problem by using computers to automate the process of searching data for useful patterns.

Keywords

Knowledge Discovery from Databases, Data Mining

1. INTRODUCTION

NASA will soon launch a satellite network devoted exclusively to earth science, the Earth Orbiting System (EOS). This complicated array of sensors will generate forty-six megabytes of data per second, which is almost four terabytes a day. For comparison, four terabytes is enough space to store fifteen hundred copies of the thirty two-volume text of the Encyclopedia Britannica [2]. So once NASA has all of this data, they will need some way to analyze it to produce useful conclusions, otherwise EOS will not be of any use to earth scientists. The only viable option for examining such a volume of information today is data mining.

Data mining, also known as knowledge discovery from databases, is the analysis of data to search for underlying patterns that will hold for all occurrences of the data source. These patterns can then be used to predict future events with a fair degree of certainty. These predications can be used to enact measures to prevent undesired events and to promote desirable trends. Data mining has numerous applications in science, security, and business.

2. THE BASIC DATA MINING PROCESS

Before any data mining systems can be used on a set of data, automated data reduction techniques often need to be used to trim down the data to a manageable size. These data reduction techniques often include cataloging, classification, clustering, segmentation, and partitioning, as well as other forms of sorting [2, 5, 6]. This tends to be a tedious and time-consuming part of the process because tremendous amounts of data need to be manipulated. Data cleaning should also occur at this stage, meaning that incorrect, inconsistent, incomplete, or missing data needs to be accounted for. Unfortunately, most current data mining tools do not deal with data cleaning very well.

Data mining programs can use any of a number of algorithms including advanced data visualization, a multitude of statistical procedures, tree-based modeling and segmentation, genetic algorithms, machine learning, inductive reasoning and association, neural networks [4, 6], and pattern/image recognition algorithms.

Whatever method the data-mining program uses, it is executed on a sample of the reduced and cleaned data set. The trends and rules the program generates are then tested against another segment of the data set to see if they still hold true, and therefore represent true patterns rather than coincidences specific to the data sample [7].

At this point, human analysts examine the patterns found by the data mining to determine which can be useful or should be explored more intricately. Truly useful and important patterns are often repeatedly tested with new data sets to confirm their reliability [7].

3. A BASIC DATA MINING ALGORITHM

Decision tree learning is one of the most widely used and practical methods for data mining. It is a method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions. I will describe the basic decision tree algorithm ID3 [6]. More complex algorithms include ASSISTANT and C4.5. Decision trees are popular because they are robust against errors in the data and missing attributes.

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this

attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the subtree rooted at the new node [6].

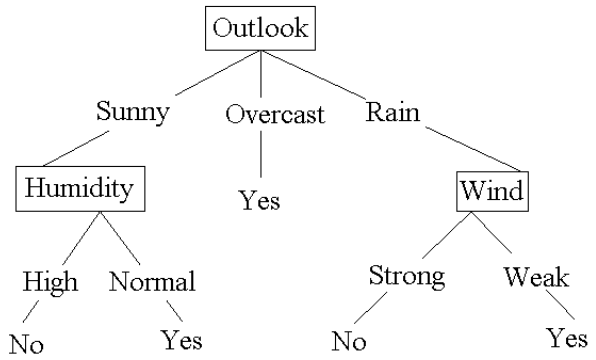

Figure 1. Basic decision tree

Figure 1 illustrates a simple learned decision tree. This decision tree classifies Saturday mornings according to whether they are suitable for playing tennis. For example, the instance

<Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong>

would be sorting down the leftmost branch of this decision tree and would therefore be classified as a negative instance because it leads to a "No". Note that the Wind and Temperature attributes are ignored because they are not part of the branch that leads to the conclusion. This allows decision trees to avoid considering unnecessary information [6].

ID3 learns decision trees by constructing them top-down, beginning with the question "which attribute should be tested at the root of the tree?" To answer this question, each attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node. This process is repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree. This forms a greedy search for an acceptable decision tree. Note that the algorithm never backtracks to reconsider earlier choices [6].

In order to decide which attribute to test at each node in the tree, we begin by defining entropy. Entropy characterizes the (im)purity of an arbitrary collection of examples. Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is

$$\text{Entropy}(S) = -p(+)\log_2 p(+) - p(-)\log_2 p(-) \quad (1)$$

where p(+) is the proportion of positive examples in S and p(-) is the proportion of negative examples in S [6]. To illustrate, suppose S is a collection of 14 examples of some Boolean concept, including 9 positive and 5 negative examples (we adopt the notation [9+, 5-] to summarize such a sample of data). Then the entropy of S is

$$\text{Entropy}([9+, 5-]) = -(9/14)\log_2 (9/14) - (5/14)\log_2 (5/14) = 0.940 \quad (2)$$

Thus far we have discussed entropy in the special case where the target classification is Boolean. For calculating entropy in a more general way, if the target attribute can take on c different values, then the entropy of S relative to this c-wise classification is defined as

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \quad (3)$$

where p_i is the proportion of S belonging to class i. Note the logarithm is still in base 2 because entropy is a measure of the expected encoding length measured in bits [6].

Entropy is important for deciding which attribute would be the best test for a node of a decision tree because it is used to calculate the information gain. Information gain is the expected reduction in entropy caused by partitioning the examples according to this attribute. More precisely, the information gain, Gain(S, A) of an attribute A, relative to a collection of examples S, is defined as

$$\text{Gain}(S, A) \equiv \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} |S_v|/|S| \, \text{Entropy}(S_v) \quad (4)$$

Where Values(A) is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v [6]. Note that the first term in Equation (4) is just the entropy of the original collection S, and the second term is the expected value of the entropy after S is partitioned using attribute A.

For example, suppose S is a collection of training-examples described by attributes including Wind, which can have the values Weak and Strong [6]. As before, assume S is a collection containing 14 examples, [9+, 5-]. Of these 14 examples, suppose 6 of the positive and 2 of the negative examples have Wind = Weak, and the remainder have Wind = Strong. The information gain due to sorting the original 14 examples by the attribute Wind may then be calculated as

$$\begin{aligned}
\text{Values(Wind)} &= \text{Weak, Strong} \\
S &= [9+, 5-] \\
S_{\text{Weak}} &\leftarrow [6+, 2-] \\
S_{\text{Strong}} &\leftarrow [3+, 3-] \\
\text{Gain}(S, \text{Wind}) &= \text{Entropy}(S) - \sum_{v \in (\text{Weak, Strong})} |S_v|/|S| \, \text{Entropy}(S_v) \\
&= \text{Entropy}(S) - (8/14)\text{Entropy}(S_{\text{Weak}}) \\
&\quad - (6/14)\text{Entropy}(S_{\text{Strong}}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}$$

At each node of a decision tree ID3 calculates all the information gains of the possible attributes, then it chooses the attribute that will yield the highest information gain and makes that the test of the current node [6]. In this way ID3 maximizes the utility of each node and generally generates shorter trees. The definition of the ID3 algorithm is

ID3(Examples, Target_attribute, Attributes)

Examples are the training examples. Target_attribute is the attribute whose values are to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a Root node for the tree
- If all Examples are positive, Return the single-node tree Root, with label = +
- If all Examples are negative, Return the single-node tree Root, with label = –
- If Attributes is empty, Return the single-node tree Root, with label = most common value of Target_attribute in Examples
- Otherwise Begin
  - A ← the attribute from Attributes that best* classifies Examples
  - The decision for Root ← A
  - For each possible value, v_i, of A,
    - Add a new tree branch below Root, corresponding to the test A = v_i
    - Let $Examples_{vi}$ be the subset of Examples that have value v_i for A
    - If $Examples_{vi}$ is empty
      - Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples
      - Else below this new branch add the subtree ID3(Examples, Target_attribute, Attributes – {A})
- End
- Return Root

* The best attribute is the one with the highest information gain, as defined in Equation (4).

4. DATA MINING IN BUSINESS

Many businesses are interested in data mining because of the falling cost of data storage, the increasing ease of collecting data over networks, and the immense computational power available at low prices. The development of robust and efficient data mining algorithms has caused most businesses to create huge databases containing as much information about their activities as possible. Already available on the market are generic multitask data mining tools to perform a variety of discovery operations. Examples include Clementine, IMACS, MLC++, MOBAL, and Recon [1].

Making data mining programs useful to businesses requires several elements. First, the problem needs to be stated in the business users' terms, including viewing the data in a business model perspective. Second, the program needs to support specific key business analyses such as segmentation, which is very important in marketing applications. Third, the results of the data mining need to be presented in a form geared to the business problem being solved. Finally, there has to be support for protracted data mining on an increasing data set, since business databases are continually growing to store increasing numbers of business transactions [1].

Data mining applications have been developed for a variety of businesses including marketing, finance, banking, manufacturing, and telecommunications [1, 6]. Many of these applications use a predictive modeling approach, but they encompass the full range of methods previously mentioned.

Data mining in marketing falls into the broad area called database marketing. It consists of analysis of customer databases to select the best potential customers for a particular product. Business Weekly estimated that more than fifty percent of all U.S. retailers use or plan to use database marketing. American Express has had good results from database marketing, experiencing a ten to fifteen percent increase in credit card use [1].

The BBC of the U.K. hired Integral Solutions Ltd. to develop a system for predicting the size of television audiences [1]. Integral Solutions Ltd.'s program used neural networks and rule induction to determine the factors playing the most important roles in relating the size of a program's audience to its scheduling slot. The final version performed as well as human experts but adapted more quickly to changes because it was constantly retrained with current data.

Early developments in data mining included Coverstory and Spotlight [1], programs that analyzed supermarket sales data and generated reports on the most significant changes in volume and share broken down by region, product type, and other qualities. Causal factors like price changes and distribution channels were analyzed and related to changes in volume and share. Spotlight later grew into the Opportunity Explorer system, which includes support for sales representatives of consumer packaged-goods companies to examine their business with individual retailers. This is accomplished by creating presentations showing the advantages of stocking additional products or having special promotions. It even generates interactive reports using hyperlinks for easy navigation.

The Management Discovery Tool (MDT), developed by AT&T and NCR [1], incorporates a set of business rules so that users can easily set up monitors for detecting significant changes in important business indicators. MDT also allows automatic HTML report generation, making it easier for users to understand the causes of changes while allowing deeper analysis through point and click links. To appeal more to mainstream business users, MDT provides a limited set of analysis types including summarization, trend analysis, change analysis, and measure and segment comparison.

The Fidelity Stock Selector fund uses a neural network data mining system to select investments [1]. It performed quite well in the stock market overall. A human fund manager evaluates the output of the system before the action is taken however, so it is not ascertainable how to divide the credit between human and machine.

A data mining system developed by Carlberg & Associates using neural networks was used to predict the Standard & Poor's 500 Index [1, 4]. It incorporated interest rates, earnings, dividends, the dollar index, and oil prices in its analysis. The system was amazingly successful, accounting for ninety six percent of the variation in the Index from 1986 to 1995.

The Clonedetector system developed by GTE uses customer profiles to detect cellular cloning fraud [1]. If a particular user suddenly starts calling in an unusual way, the Clonedetector

automatically informs GTE security. A similar system was developed by AT&T to detect international calling fraud, but that system is much more interactive with and reliant upon human operators.

Acknosoft developed a data-mining program called CASSIOPEE for General Electric and SNECMA [1]. CASSIOPEE is being used by three European airlines to diagnose and predict technical problems in Boeing 737 aircraft. Clustering methods are used to derive families of faults.

5. DATA MINING IN SECURITY SYSTEMS

In June of 1994, a computer expert in St. Petersburg, Russia, Vladimir Leonidovich Levin, penetrated the CitiBank electronic funds-transfer network. Over the course of five months, he funneled 10 million dollars into accounts in California, Israel, Germany, Finland, the Netherlands and Switzerland. He was eventually apprehended, and most of the money was recovered, but the incident revealed the vulnerability of large databases to computer hackers [1].

Incidents such as the one described above make the security of a company's computer system a serious issue in today's business world. System administrators and security officers monitor these computer networks, often comprised of thousands of computers and terabytes of storage space. Their job is daunting; especially since a security violation on one workstation could become a multimillion-dollar incident. The Computer Emergency Response Team, an organization of computer security professionals, estimates that only five percent of companies whose security has been compromised are even aware that they have been infiltrated. Although the raw information needed to detect an intrusion is often available in the audit data recorded by each computer, there is far too much of it generated each day for the system administrators and security officers to inspect it. Even if they tried, the vast majority of the audit record would be completely mundane and innocuous actions [1].

Data mining offers a convenient way to monitor these large computer networks. By detecting anomalous activities in the logs of computers, a data mining system could flag suspicious events for later inspection by system administrators, allowing them to avoid checking all the normal daily activities. The data mining system does this by developing a profile of the typical activities of each user in the network. Deviations from the expected pattern could be harmful or abusive behavior and would therefore be flagged. The system would have to be flexible enough to compensate for normal deviations from expected behavior like users learning new programs or doing new tasks. One study done at Purdue University [1] found that a data mining system was able to identify a profiled user ninety-nine percent of the time and differentiated between a profiled user and another user with almost ninety-four percent accuracy.

Although only a few companies currently have data mining systems checking their audit records, the number is expected to dramatically increase in the near future as companies desperately try to ensure that their computer systems are secure from intrusion.

6. DATA MINING IN SCIENCE DATABASES

Today's advanced scientific instruments can easily generate terabytes and even petabytes (a million gigabytes) of information. Although data mining can be an invaluable tool in analyzing this data, it faces the additional challenge that scientific data frequently is not in a convenient flat file format. Scientific data is frequently in the form of images, which are relatively easily examined by humans, but which present a myriad of problems for data mining programs. There is also time-series and sequence data such as DNA sequences, which need special algorithms to be dealt with effectively. Finally there are categorical values such as protein sequences [3]. The problem with such data is that many algorithms rely on feature vectors allowed by numerical data, so these algorithms cannot be used on categorical data sets. Despite these extra difficulties, scientific data mining has still been making rapid progress [5].

An example of mining scientific data was the cataloging of a sky survey. The Second Palomar Observatory Sky Survey took six years to collect three terabytes of image data containing an estimated two billion sky objects [3]. The three thousand photographic images were scanned into 16-bit pixel resolution digital images at 23,040x23,040 pixels per image. The problem was generating a survey catalog of all these sky objects from this information. Additionally, the attributes and class of each object needed to be determined and recorded in the catalog. To solve this problem, the Sky Image Cataloging and Analysis Tool (SKI-CAT) system was developed.

The majority of objects in each image were faint, making determination of their class by visual inspection or classical computational approaches in astronomy impossible. SKI-CAT used decision-tree learning algorithms to accurately predict sky object classes [3]. This accuracy was verified by comparison with a set of high-resolution charged-couple device images. SKI-CAT was ninety four percent accurate at predicting the class of sky objects, which increased the number of reliably classified objects by three hundred percent. These results have already helped astronomers discover sixteen new high red-shift quasars. Such quasars are difficult to find and provide clues about the early history of the universe.

The Magellan spacecraft orbited the planet Venus for over five years and used synthetic aperture radar to penetrate the gas and cloud cover to map the surface of the planet [3]. The result is that we have a unique high-resolution map of the entire planet. In fact, we have more of Venus mapped at the 75-m pixel resolution than we do of the Earth because so much of the Earth is covered by water. This dataset is valuable because of its completeness and because Venus is the most similar to Earth in size. It is hoped that learning about the geological evolution of Venus will produce valuable lessons about the Earth.

The immense size of this dataset prevents planetary geologists from personally examining all the images. To assist geologists in analyzing the Venus map, the Jet Propulsion Laboratory developed the Adaptive Recognition Tool (JARtool) [3]. The system seeks to automate the search for small volcanoes by training the system via examples. The geologists would label a

small sample of the images and the system would then use these to train itself to recognize small volcanoes. The system would then attempt to locate and measure the planet's estimated one million small volcanoes. It used classification learning to distinguish true detections of volcanoes from false alarms. It performed as well as scientists in identifying common types of small volcanoes, but rarely detected those scientists are not sure about.

The geoscientific data mining system Quakefinder [3] automatically detects and measures tectonic activity in the Earth's crust using satellite data. It was used to map the direction and magnitude of ground displacements due to the 1992 Landers earthquake in Southern California over a spatial region of several hundred square kilometers at a resolution of 10 m to a sub-pixel precision of 1 m. Quakefinder is implemented on a 256-node Gray T3D parallel supercomputer so that the gathered data can rapidly produce scientific results. Besides automatically measuring known faults, it also allows automatic knowledge discovery by indicating novel unexplained tectonic activity away from the primary faults never before observed. Future work will focus on the measurement of continuous processes over many images, instead of simply measuring abrupt behavior seen during earthquakes.

7. CONCLUSIONS

Although data mining is still limited in its functionality, its potential is nearly unlimited. Already business, science, and security have derived benefits from its development. Databases that used to store millions of bits of useless information can be mined for insights that can greatly profit the miners. Recently, scientific instruments and business systems have been gathering extra information that was apparently useless simply because it was so easy to do. The creation of data mining makes this excess information useful.

Research to expand the types and magnitude of data that data mining systems can effectively mine is well underway. The needs of business, security, and science will provide incentive to invest time and money into such development. Perhaps someday data mining will advance faster than the growth of databases and allow the mining of nearly infinite databases, such as mining the entire World Wide Web.

8. REFERENCES

[1] Brachman, Ronald J., Tom Khabaza, Willi Kloesgen, Gregory Piatetsky-Shapiro, Evangelos Simoudis, "Mining Business Databases," *Communications of the ACM*, v39, p42(7) (Nov 1996).

[2] Brodley, Carla E., Terran Lane, and Timothy M. Stough, "Knowledge Discovery and Data Mining," *American Scientist*, v86, p54(8) (Jan-Feb 1999).

[3] Fayyad, Usama, David Haussler, and Paul Stolorz, "Mining Scientific Data," *Communications of the ACM*, v39, p51(7) (Nov 1996).

[4] Fu, LiMin, "Knowledge Discovery Based on Neural Networks," *Communications of the ACM*, v42, p48 (Nov 1999).

[5] Mitchell, Tom M., "Machine Learning and Data Mining", *Communications of the ACM*, v42, p30 (Nov 1999).

[6] Mitchell, Tom M., Machine Learning. McGraw-Hill Companies, 1997.

[7] Studt, Tim, "Scientific Data Miners Make use of all the Tools Available: Data Mining can Extract Previously Unknown and Potentially Useful Information from Extremely Large and Complex Data Bases," *R & D*, v39, p62C(2) (April 1997).