To appear in the Handbook of Technology Management, H. Bidgoli (Ed.), John Wiley and Sons, 2010.

Data Mining

Gary M. Weiss, Ph.D., *Department of Computer and Information Science, Fordham University*
Brian D. Davison, Ph.D., *Department of Computer Science and Engineering, Lehigh University*

INTRODUCTION

The amount of data being generated and stored is growing exponentially, due in large part to the continuing advances in computer technology. This presents tremendous opportunities for those who can unlock the information embedded within this data, but also introduces new challenges. In this chapter we discuss how the modern field of data mining can be used to extract useful knowledge from the data that surround us. Those that can master this technology and its methods can derive great benefits and gain a competitive advantage.

In this introduction we begin by discussing what data mining is, why it developed now and what challenges it faces, and what types of problems it can address. In subsequent sections we look at the key data mining tasks: prediction, association rule analysis, cluster analysis, and text, link and usage mining. Before concluding we provide a list of data mining resources and tools for those who wish further information on the topic.

What is Data Mining?

Data mining is a *process* that takes data as input and outputs knowledge. One of the earliest and most cited definitions of the data mining process, which highlights some of its distinctive characteristics, is provided by Fayyad, Piatetsky-Shapiro and Smyth (1996), who define it as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." Note that because the process must be non-trivial, simple computations and statistical measures are not considered data mining. Thus predicting which salesperson will make the most future sales by calculating who made the most sales in the previous year would *not* be considered data mining. The connection between "patterns in data" and "knowledge" will be discussed shortly. Although not stated explicitly in this definition, it is understood that the process must be at least partially automated, relying heavily on specialized computer algorithms (i.e., data mining algorithms) that search for patterns in the data.

It is important to point out that there is some ambiguity about the term "data mining", which is in large part purposeful. This term originally referred to the algorithmic step in the data mining process, which initially was known as the Knowledge Discovery in Databases (KDD) process. However, over time this distinction has been dropped and data mining, depending on the context, may refer to the entire process or just the algorithmic step. This entire process, as originally envisioned by Fayyad, Piatetsky-Shapiro and Smyth (1996), is shown in Figure 1. In this chapter we discuss the entire process, but, as is common with most texts on the subject, we focus most of our attention on the algorithmic data mining step.

The first three steps in Figure 1 involve preparing the data for mining. The relevant data must be selected from a potentially large and diverse set of data, any necessary preprocessing must then be performed, and finally the data must be transformed into a representation suitable for the data mining algorithm that is applied in the data mining step. As an example, the preprocessing step might involve computing the day of week from a date field, assuming that the domain experts thought that having the day of week information would be useful. An example of data transformation is provided by Cortes and Pregibon (1998). If each data record describes one *phone call* but the goal is to predict whether a *phone number* belongs to a business or residential customer based on its calling patterns, then all records associated with each phone number must be *aggregated*, which will entail creating attributes corresponding to the average number of calls per day, average call duration, etc.

While data preparation does not get much attention in the research community or the data mining community in general, it is critical to the success of any data mining project because without high quality data it is often impossible to learn much from the data. Furthermore, although most research on data mining pertains to the data mining algorithms, it is commonly acknowledged that the choice of a specific data mining algorithms is generally less important than doing a good job in data preparation. In practice it is common for the data preparations steps to take more time and effort than the actual data mining step. Thus, anyone undertaking a data mining project should ensure that sufficient time and effort is allocated to the data preparation steps. For those interested in this topic, there is a book (Pyle 1999) that focuses exclusively on data preparation for data mining.

The fourth step in the data mining process is the data mining step. This step involves applying specialized computer algorithms to identify patterns in the data. Many of the most common data mining algorithms, including decision tree algorithms and neural network algorithms, are described in this chapter. The patterns that are generated may take

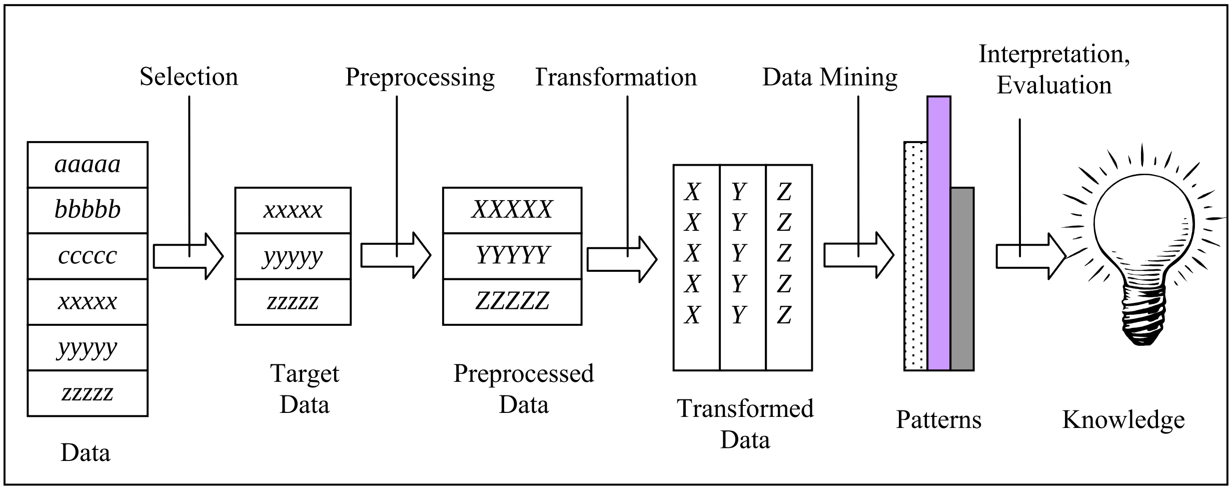


Figure 1: The Data Mining Process

various forms (e.g., decision tree algorithms generate decision trees). At least for predictive tasks, which are probably the most common type of data mining task, these patterns collectively can be viewed as a *model*. For example, if a decision tree algorithm is used to predict who will respond to a direct marketing offer, we can say that the decision tree models how a consumer will respond to a direct mail offer. Finally, the results of data mining cannot simply be accepted, but must be carefully evaluated and interpreted. As a simple example, in the case of the direct-mail example just described, we could evaluate the decision tree based on its accuracy—the percentage of its predictions that are correct. However, many other evaluation or performance metrics are possible and for this specific example return on investment might actually be a better metric.

The data mining process is an iterative process, although this is not explicitly reflected in Figure 1. After the initial run of the process is complete, the user will evaluate the results and decide whether further work is necessary or if the results are adequate. Normally, the initial results are either not acceptable or there is an expectation that further improvements are possible, so the process is repeated after some adjustments are made. These adjustments can be made at any stage of the process. For example, additional data records may be acquired, additional fields (i.e., variables) may be generated from existing information or obtained (via purchase or measurement), manual cleaning of the data may be performed, or new data mining algorithms may be selected. At some point the results may become acceptable and the mined knowledge then will be communicated and may be acted upon. However, even once the mined knowledge is acted upon the data mining process may not be complete and have to be repeated, since the data distribution may change over time, new data may become available, or new evaluation criteria may be introduced.

Motivation and Challenges

Data Mining developed as a new discipline for several reasons. First, the amount of data available for mining grew at a tremendous pace as computing technology became widely deployed. Specifically, high speed networks allowed enormous amount of data to be transferred and rapidly decreasing disk costs permitted this data to be stored cost-effectively. The size and scope of these new datasets is remarkable. According to a recent industry report (International Data Corporation 2007), in 2006 161 Exabyte's (161 Billion Gigabytes) of data were created and in 2010 988 Exabyte's of data will be created. While these figures include data in the form of email, pictures and video, these and other forms of data are increasingly being mined. Traditional corporate datasets, which include mainly fixed-format numerical data, are also quite huge, with many companies maintaining Terabyte datasets that record every customer transaction.

This remarkable growth in the data that is collected and stored presents many problems. One basic problem concerns the scalability of traditional statistical techniques, which often cannot handle data sets with millions or billions of records and hundreds or thousands of variables. A second problem is that much of the data that is available for analysis, such as text, audio, video and images, is non-numeric and highly unstructured (i.e., cannot be represented by a fixed set of variables). This data cannot easily be analyzed using traditional statistical techniques. A third problem is that the number of data analysts has not matched the exponential growth in the amount of data, which has caused much of this data to remain unanalyzed in a “data tomb” (Fayyad 2003).

Data mining strives to address these challenges. Many data mining algorithms are specifically designed to be scalable and perform well on very large data sets, without fixed size limitations, and degrade gracefully as the size of the data set and the number of variables increases. Scalable data mining algorithms, unlike many early machine learning algorithms, should not require that all data be loaded into and remain in main memory, since this would prevent very large data sets from being mined. A great deal of effort has also been expended to develop data mining methods to handle non-numeric data. This includes mining of spatial data (Ester et al. 1998), images (Hsu, Lee, and Zhang 2002), video (Zhu et al. 2005), text documents (Sebastiani 2002), and the World Wide Web (Chakrabarti 2002, Liu 2007). While a discussion of all of these methods is beyond the scope of this chapter, text mining and web mining, which have become increasingly popular due to the success of the Web, are discussed later in this chapter.

Data mining also attempts to offload some of the work from the data analyst so that more of the collected data can be analyzed. One can see how data mining aids the data analyst by contrasting data mining methods with the more conventional statistical methods. Most of statistics operates using a hypothesize-and-test paradigm where the statistician first decides on a specific hypothesis to be tested (Hand 1998). The statistician then makes assumptions about the data (e.g., that it is normally distributed) and then tries to fit a model based on these assumptions. In data mining the analyst does not need to make specific assumptions about the data nor formulate a specific hypothesis to test. Instead, more of the responsibility for finding a good model is assigned to the data mining algorithm, which will search through a large space of potential models in order to identify a good model. The data mining process, unlike the deductive process typically used by a statistician, is typically data-driven and inductive, rather than hypothesis-driven and deductive. It needs to be noted, however, that for data mining to be successful there must be a sufficient amount of high quality (i.e., relatively noise-free) data. This amount depends on the complexity of the problem and cannot easily be estimated a priori.

Overview of Data Mining Tasks

The best way to gain an understanding of data mining is to understand the types of tasks, or problems, that it can address. At a high level, most data mining tasks can be categorized as either having to do with *prediction* or *description*. Predictive tasks allow one to predict the value of a variable based on other existing information. Examples of predictive tasks include predicting when a customer will leave a company (Wei and Chiu 2002), predicting whether a transaction is fraudulent or not (Fawcett and Provost 1997), and identifying the best customers to receive direct marketing offers (Ling and Li 2000). Descriptive tasks, on the other hand, summarize the data in some manner. Examples of such tasks include automatically segmenting customers based on their similarities and differences (Chen et al. 2006) and finding associations between products in market basket data (Agrawal and Srikant 1994). Below we briefly describe the major predictive and descriptive data mining tasks. Each task is subsequently described in greater detail later in the chapter.

Classification and Regression

Classification and regression tasks are predictive tasks that involve building a model to predict a target, or dependent, variable from a set of explanatory, or independent, variables. For classification tasks the target variable usually has a small number of discrete values (e.g., “high” and “low”) whereas for regression tasks the target variable is continuous. Identifying fraudulent credit card transactions (Fawcett and Provost 1997) is a classification task while predicting future prices of a stock (Enke and Thawornwong 2005) is a regression task. Note that the term “regression” in this context should not be confused with the regression methods used by statisticians (although those methods can be used to solve regression tasks).

Association Rule Analysis

Association rule analysis is a descriptive data mining task that involves discovering patterns, or associations, between elements in a data set. The associations are represented in the form of rules, or implications. The most common association rule task is *market basket analysis*. In this case each data record corresponds to a transaction (e.g., from a supermarket checkout) and lists the items that have been purchased as part of the transaction. One possible association rule from supermarket data is {Hamburger Meat} → {Ketchup}, which indicates that those transactions that include Hamburger Meat tend to also include Ketchup. It should be noted that although this is a descriptive task, highly accurate association rules can be used for prediction (e.g., in the above example it might be possible to use the presence of “Hamburger Meat” to predict the presence of “Ketchup” in a grocery order).

Cluster Analysis

Cluster analysis is a descriptive data mining task where the goal is to group similar objects in the same cluster and dissimilar objects in different clusters. Applications of clustering include clustering customers for the purpose of market segmentation and grouping similar documents together in response to a search engine request (Zamir and Etzioni 1998).

Text Mining Tasks

Much available data is in the form of unstructured or semi-structured text, which is very different from conventional data, which is completely structured. Text is unstructured if there is no predetermined format, or structure, to the data. Text is semi-structured if there is structure associated with some of the data, as in the case for web pages, since most web pages will have a title denoted by the title tag, images denoted by image tags, etc. While text mining tasks often fall into the classification, clustering and association rule mining categories, we discuss them separately because the unstructured nature of text requires special consideration. In particular, the method for representing textual data is critical. Example applications of text mining includes the identification of specific noun phrases such as people, products and companies, which can then be used in more sophisticated co-occurrence analysis to find non-obvious relationships among people or organizations. A second application area that is growing in importance is sentiment analysis, in which blogs, discussion boards, and reviews are analyzed for opinions about products or brands.

Link Analysis Tasks

Link analysis is a form of network analysis that examines associations between objects. For example, in the context of email, the objects might be people and the associations might represent the existence of email between two people. On the Web each page can link to others, and so web link analysis considers the web graph resulting from such links. Given a graph showing relationships between objects, link analysis can find particularly important or well-connected objects and show where networks may be weak (e.g., in which all paths go through one or a small number of objects).

PREDICTION TASKS: CLASSIFICATION AND REGRESSION

Classification and regression tasks are the most commonly encountered data mining tasks. These tasks, as described earlier, involve mapping an object to either one of a set of predefined classes (classification) or to a numerical value (regression). In this section we introduce the terminology required to describe these tasks and the framework for performing predictive modeling. We then describe several key characteristics of predictive data mining algorithms and finish up by describing the most popular predictive data mining algorithms in terms of these characteristics.

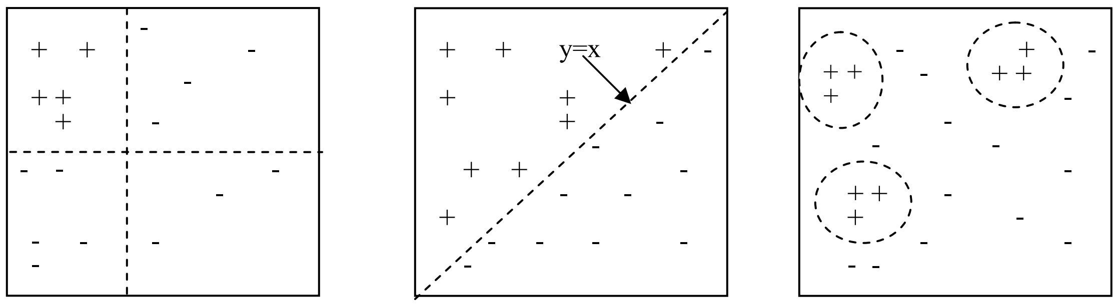
Terminology and Background

Most prediction tasks assume that the underlying data is represented as a collection of objects or records, which, in data mining, are often referred to as instances or examples. Each example is made up of a number of variables, commonly referred to as features or attributes. The attribute to be predicted is of special interest and may be referred to as the target, or, for classification tasks, the class. In the majority of cases the number of attributes is fixed and thus the data can be represented in a tabular format, as shown in Table 1. The data in Table 1 describes automobile loans and contains 10 examples with 5 attributes, where the binary target/class attribute resides in the last column and indicates whether the customer defaulted on their loan.

Table 1: Sample Auto Loan Default Data

Age	Income	Student	Credit Rating	Default
Youth	Medium	Yes	Fair	No
Youth	Low	Yes	Fair	No
Senior	Low	No	Excellent	No
Senior	Medium	No	Excellent	No
Senior	High	No	Poor	Yes
Senior	Medium	No	Poor	Yes
Senior	Low	Yes	Fair	No
Middle Age	Low	No	Fair	Yes
Middle Age	Medium	Yes	Fair	No
Middle Age	Low	No	Fair	Yes

This data in Table 1 can be used to build a predictive model to classify customers based on whether they will default on their loan. That is, the model generated by the data mining algorithm will take the values for the *age*, *income*, *student*, and *credit-*



(a) (b) (c)

Figure 3: Three Sample Classification Tasks

rating attributes and map them to “Yes” or “No”. The induced classification model could take many forms and Figure 2 provides a plausible result for a rule-based learning algorithm. In this case each rule in Figure 2 is evaluated in order and the classification is determined by the first rule for which the left-hand side evaluates to true. In this case, if the first two rules do not fire, then the third one is guaranteed to fire and thus acts as a default rule. Note that the three rules in Figure 2 can be used to correctly classify all ten examples in Table 1.

1) If credit-rating = “Poor” → Default = “Yes”
2) If Age = “Middle Aged” and Income = “Low” → Default = “Yes”
3) → Default = “No”

Figure 2: Rule Set for Predicting Auto Loan Defaults

In data mining, the predictive model is induced from a *training set* where the target or class value is provided. This is referred to as supervised learning, since it requires that someone acts as a teacher and provides the answers (i.e., class/target value) for each training example. The model induction is accomplished in the data mining step using any of a number of data mining algorithms. Once the model is generated, it can then be applied to data where the value for the target value is unknown. Because we are primarily interested in evaluating how well a model performs on new data (i.e., data not used to build the model), we typically reserve some of the labeled data and assign it to a test set for evaluation purposes. When the test set is used for evaluating the predictive performance of the model, the target value is examined only after the prediction is made. Evaluating the model on a test set that is independent of the training data is crucial because otherwise we would have an unrealistic (i.e., overly optimistic) estimate of the performance of the model.

Classification requires the data mining algorithm to partition the input space in such a way as to separate the examples based on their class. To help illustrate this, consider Figure 3, which shows representations of three data sets, where each example is described by three attributes, one of which is the class variable that may take on the values “+” and “-”.

The dashed lines in Figure 3 form decision boundaries that partition the input space in such a way that the positive examples are separated from the negative examples. In Figure 3a the two lines partition the space into four quadrants so that all positive examples are segregated in the top left quadrant. Similarly, in Figure 3b the equation $x + y = 1$ forms the decision boundary that perfectly separates the two classes. The data in Figure 3c requires three globular decision boundaries to perfectly identify the positive examples. Some classification tasks may require that complex decision boundaries be formed in order to achieve good classification performance, but even in such cases it is often impossible to perfectly separate the examples by class. This is due to the fact that domains may be complex and the data may be noisy. Note that the decision boundary in Figure 2b, because it is not parallel to either axis, requires both attribute values to be considered at once, something not required by the data in Figure 2a. This is significant because not all prediction algorithms (e.g., decision trees) produce models with this capability.

Characteristics of Predictive Data Mining Algorithms

Before the most commonly used data mining algorithms are introduced, it is useful to understand the characteristics that can be used to describe and compare them. These characteristics are described briefly in this section and then referred to in subsequent

Table 2: Summary off Predictive Data Mining Algorithms

Learning Method	Tasks Handled	Expressive Power	Training Time	Testing Time	Model Comprehensibility
Decision Trees	Classification	Fair	Fast	Fast	Good
Rule-Based	Classification	Fair	Fast	Fast	Good
ANN	Classification, Regression	Good	Slow	Fast	Poor
Nearest-Neighbor	Classification, Regression	Good	No Time	Slow	No model generated but predictions are explainable
Naïve Bayesian	Classification	Good	Fast	Fast	Poor

sections. The first characteristic concerns the type of predictive tasks that can be handled by the algorithm. Predictive data mining algorithms may handle only classification tasks, only regression tasks, or may handle both types of tasks.

The second characteristic concerns the expressive power of the data mining model. The expressive power of a model is determined by the types of decision boundaries that it can form. Some learners can only form relatively simple decision boundaries and hence have limited expressive power. Algorithms with limited expressive power may not perform well on certain tasks, although it is difficult to determine in advance which algorithms will perform best for a given task. In fact, it is not uncommon for those algorithms that generate less complex models to perform competitively with those with more expressive power.

The format of the model impacts the third criterion, which is the *comprehensibility*, or explainability, of the predictive model. Certain models are easy to comprehend or explain, while others are nearly impossible to comprehend and, due to their nature, must essentially be viewed as "black boxes" that given an input, somehow produce a result. Whether comprehensibility is important depends on the goal of the predictive task. Sometimes one only cares about the outcome—perhaps the predictive accuracy of the algorithm—but often one needs to be able to explain or defend the predictions. In some cases comprehensibility may even be the primary evaluation criteria if the main goal is to better understand the domain. For example, if one can build an effective model for predicting manufacturing errors, then one may be able to use that model to determine how to reduce the number of future errors.

The fourth criterion concerns the *computation time* of the data mining algorithm. This is especially important due to the enormous size of many data sets. With respect to computation time, we are interested in the training time, how long it will take to build the model, and the testing time, how long it will take to apply the model to new data in order to generate a prediction. Computational requirements are much more of a concern if the learning must occur in real-time, but currently most data mining occurs "off-line."

Table 2 describes some of the most popular data mining methods in terms of the characteristics just introduced (the methods themselves are described in the next section). Experience has shown that there is no one best method and that the method that performs best depends on the domain as well as the goals of the data mining task (e.g., does the induced model need to be easily understandable). The five listed methods are all in common use and are implemented by most major data mining packages.

Predictive Data Mining Algorithms

In this section we briefly describe some of the most common data mining algorithms. Because the purpose of this chapter is to provide a general description of data mining, its capabilities, and how it can be *used* to solve real-world problems, many of the technical details concerning the algorithms are omitted. A basic knowledge of the major data mining algorithms, however, is essential in order to know when each algorithm is relevant, what the advantages and disadvantages of each algorithm are, and how these algorithms can be used to solve real-world problems.

Decision Trees

Decision tree algorithms (Quinlan 1993; Breiman et al. 1984) are a very popular class of learning algorithms for classification tasks. A sample decision tree, generated from the automobile loan data in Table 1, is shown in Figure 4. The internal nodes of the decision tree each represent an attribute while the terminal nodes (i.e., leaf nodes displayed as rectangles) are labeled with a class value. Each branch is labeled with an attribute value, and, when presented with an example, one follows the branches that match the attribute values for the example, until a leaf node is reached. The class value assigned to the leaf node is then used as the predicted value for the example. In this simple example the decision tree will predict that a customer will default on their automobile loan if their credit rating is “poor” or it is not “poor” (i.e., “fair” or “excellent”) but the person is “middle aged” and their income level is “low”.

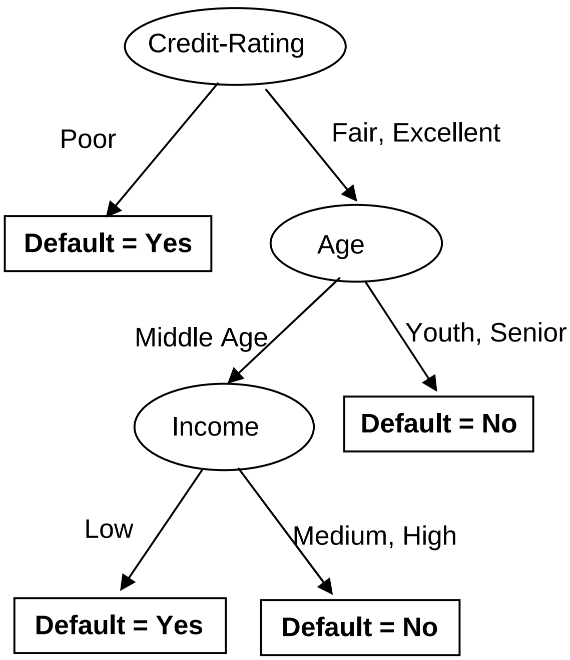


Figure 4: Decision Tree Model Induced from the Automobile Loan Data

Decision tree algorithms are very popular. The main reason for this is that the induced decision tree model is easy to understand. Additional benefits include the fact that the decision tree model can be generated quickly and new examples can be classified quickly using the induced model. The primary disadvantage of a decision tree algorithm is that it has limited expressive power, namely because only one attribute can be considered at a time. Thus while a decision tree classifier could easily classify the data set represented in Figure 3a, it could not easily classify the data set represented in Figure 3b, assuming that the true decision boundary corresponds to the line y=x. A decision tree could approximate that decision boundary if it were permitted to grow very large and complex, but could never learn it perfectly. Note that since decision trees cannot handle regression tasks, other methods must be used for those tasks.

Rule-based Classifiers

Rule-based classifiers generate classification rules, such as the rule set shown earlier in Figure 2. The way in which classifications are made from a rule set varies. For some rule-based systems the first rule to fire (i.e., have the left-hand side of the rule satisfied) determines the classification, whereas in other cases all rules are evaluated and the final classification is made based on a voting scheme. Rule-based classifiers are very similar to decision-tree learners and have similar expressive power, computation time, and comprehensibility. The connection between these two classification methods is even more direct since any decision tree can trivially be converted into a set of mutually exclusive rules, by creating one rule corresponding to the path from the root of the tree to each leaf. While some rule-based learners such as C4.5Rules (Quinlan 1993) operate this way, other rule learners, such as RIPPER (Cohen 1995), generate rules directly.

Artificial Neural Networks

Artificial Neural Networks (ANNs) were originally inspired by attempts to simulate some of the functions of the brain and can be used for both classification and regression tasks (Gurney 1997). An ANN is composed of an interconnected set of nodes that includes an input layer, zero or more hidden layers, and an output layer. The links between nodes have weights associated with them. A typical neural network is shown in Figure 5.

The ANN in Figure 5 accepts three inputs, I_1, I_2, and I_3 and generates a single output O_1. The ANN *computes* the output value from the input values as follows. First, the input values are taken from the attributes of the training example, as it is inputted to the ANN. These values are then weighted and fed into the next set of nodes, which in this example are H_1 and H_2. A non-linear activation function is then applied to this weighted sum and then the resulting value is passed to the next layer, where this process is repeated, until the final value(s) are outputted. The ANN learns by incrementally modifying its weights so that, during the training phase, the predicted output value moves closer to the observed value. The most popular algorithm for modifying the weights is the backpropagation algorithm (Rumelhart, Hinton, and William 1986). Due to the nature of ANN learning, the entire training set is applied repeatedly, where each application is referred to as an epoch.

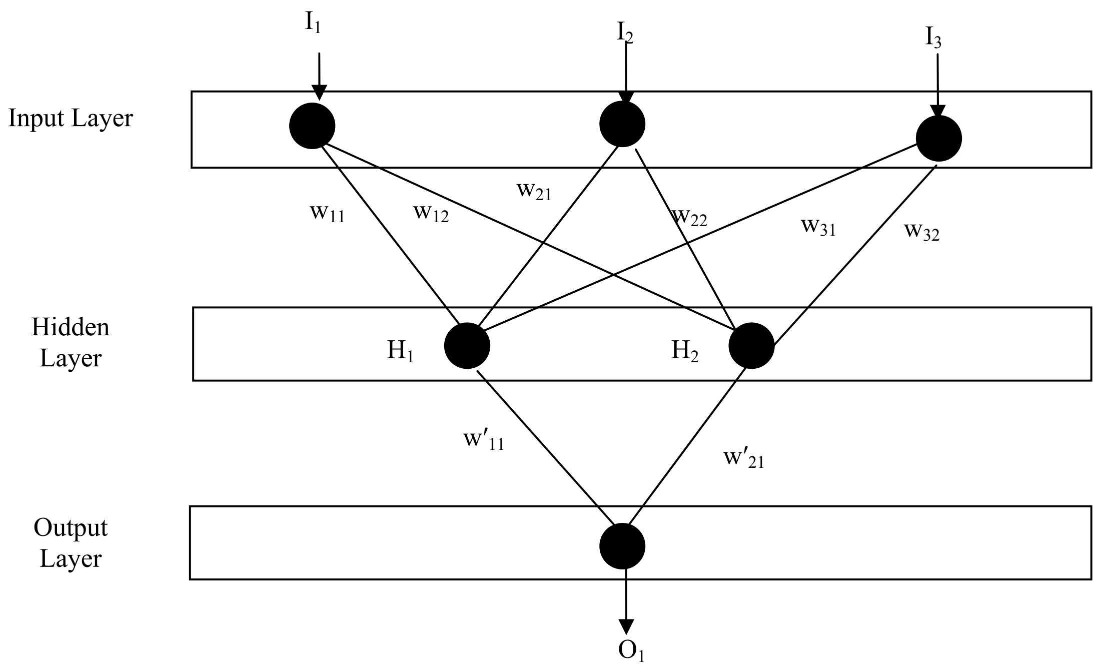


Figure 5: A Typical Artificial Neural Network

ANNs can naturally handle regression tasks, since numerical values are passed through the nodes and are ultimately passed through to the output layer. However, ANNs can also handle classification tasks by thresholding on the output values. ANNs have a great deal of expressive power and are not subject to the same limitations as decision trees. In fact, most ANNs are universal approximators in that they can approximate any continuous function to any degree of accuracy. However, this power comes at a cost. While the induced ANN can be used to quickly predict the values for unlabelled examples, training the model takes much more time than training a decision tree or rule-based learner and, perhaps most significantly, the ANN model is virtually incomprehensible and therefore cannot be used to explain or justify its predictions.

Nearest-Neighbor.

Nearest-neighbor learners (Cover and Hart 1967) are very different from any of the learning methods just described in that no explicit model is ever built. That is, there is no training phase and instead all of the work associated with making the prediction is done at the time an example is presented. Given an example the nearest-neighbor method first determines the *k* most similar examples in the training data and then determines the prediction based on the class values associated with these *k* examples, where *k* is a user-specified parameter. The simplest scheme is to predict the class value that occurs most frequently in the *k* examples, while more sophisticated schemes might use weighted voting, where those examples most similar to the example to be classified are more heavily weighted. People naturally use this type of technique in everyday life. For example, realtors typically base the sales price of a new home on the sales price of similar homes that were recently sold in the area. Nearest-neighbor learning is sometimes referred to as instance-based learning.

Nearest-neighbor algorithms are typically used for classification tasks, although they can also be used for regression tasks. These algorithms also have a great deal of expressive power. Nearest-neighbor algorithms generate no explicit model and hence have no training time. Instead, all of the computation is performed at testing time and this process may be relatively slow since all training examples may need to be examined. It is difficult to evaluate the comprehensibility of the model since none is produced. We can say that because no model is produced, one cannot gain any global (i.e., high-level) insight into the domain. However, individual predictions can easily be explained and justified in a very natural way, by referring to the nearest-neighbors. We thus say that this method does not produce a comprehensible model but its predictions are explainable.

Naïve Bayesian Classifiers

Most classification tasks are not completely deterministic. That is, even with complete knowledge about an example you may not be able to correctly classify it. Rather, the relationship between an example and the class it belongs to is often probabilistic. Naïve Bayesian classifiers (Langley et al. 1992) are probabilistic classifiers that allow us to exploit statistical properties of the data in order to predict the most likely class for an example. More specifically, these methods use the training data and the prior probabilities associated with each class and with each attribute value and then utilize Bayes' theorem to determine the most likely class given a set of observed attribute values. This method is naïve in that it assumes that the values for each attribute are independent (Bayesian Belief Networks do not make this assumption but a discussion of those methods is beyond the scope of this chapter). The naïve Bayes method is used for classification tasks. These methods are quite powerful, can express complex concepts, and are fast to generate and to classify new examples. However, these methods do not build an explicit model that can then be interpreted.

Ensemble Methods

Ensemble methods are general methods for improving the performance of predictive data mining algorithms. The most notable ensemble methods are bagging and boosting, which permit multiple models to be combined. With bagging (Breiman 1996) the training data are repeatedly randomly sampled with replacement, so that each of the resulting training sets has the same number of examples as the original training data but is composed of different training examples. A classifier is induced from each of the generated training sets and then each unlabelled test example is assigned the classification most frequently predicted by these classifiers (i.e., the classifiers "vote" on the classification). Boosting (Freund and Schapire 1997) is somewhat similar to bagging and also generates multiple classifiers, but boosting adaptively changes the distribution of training examples such that the training examples that are misclassified are better represented in the next iteration. Thus boosting focuses more attention on the examples that are difficult to classify. As with bagging, unlabeled examples are classified based on the predictions of all of the generated classifiers. Most data mining packages now implement a variety of ensemble methods, including boosting and bagging. Most of these packages also permit different models (e.g., a neural network model and a decision tree model) to be combined so that, in theory, one can get the best aspects of each.

ASSOCIATION ANALYSIS

Many businesses maintain huge databases of transactional data, which might include all purchases made from suppliers or all customer sales. Association analysis (Agrawal, Imielinki and Swami 1993) attempts to find patterns either within or between these transactions.

Simple Example using Market Basket Data

Consider the data in Table 3, which includes five transactions associated with purchases at a grocery store. These data are referred to as market basket data since each transaction includes the items found in a customer's shopping "basket" during checkout. Each record contains a transaction identifier and then a list all of the items purchased as part of the transaction.

Table 3: Market Basket Data from a Grocery Store

Transaction ID	Items
1	{Ketchup, Hamburgers, Soda}
2	{Cereal, Milk, Diapers, Bread}
3	{Hot dogs, Ketchup, Soda, Milk}
4	{Greeting Card, Cake, Soda}
5	{Greeting Card, Cake, Milk, Cereal}

The data in Table 3 are very different from the relational data used for predictive data mining, such as the data in Table 1, where each record is composed of a fixed number of fields. Here the key piece of data is a variable-length list of items, in which the list only indicates the presence or absence of an item—not the number of instances purchased.

In market basket analysis, a specific instance of association analysis, the goal is to find patterns *between* items purchased in the same transaction. As an example, using the limited data in Table 3, a data mining algorithm might generate the association rule {Ketchup} → {Soda}, indicating that a customer that purchases Ketchup is likely to also purchase Soda.

Uses of Association Analysis

There are many benefits of performing association analysis. Continuing with the grocery store example, an association rule of the form A → B can be exploited in many ways. For example, items A and B could be located physically close together in a store in order to assist the shoppers or could be located far apart to increase the chance that a shopper will encounter and purchase items that would otherwise be missed. Sales strategies can also be developed to exploit these associations. For example, the store could run a sale on item A in order to increase the sales of item B or could have a coupon printed out for item B at checkout for those who purchase item A, in order to increase the likelihood that item B will be purchased on the next shopping trip. Applications of association rule analysis are not limited to market basket data and other association mining techniques have been applied to sequence data (Agrawal and Srikant 1995), spatial data (Huang, Shekhar, and Xiong 2004) and graph-based data (Kuramochi and Karypis 2001). We discuss sequence mining toward the end of this section.

Generation and Evaluation of Association Rules

There are many algorithms for performing association rule analysis, but the earliest and most popular such algorithm is the Apriori algorithm (Agrawal and Srikant 1994). This algorithm, as well as most others, utilizes two stages. In the first stage, the sets of items that co-occur frequently are identified. In the second stage association rules are generated from these "frequent itemsets." As an example, the frequent itemset {Ketchup, Soda} could be generated from the data in Table 3, since it occurs in two transactions, and from this the association rule {Ketchup} → {Soda} could be generated.

The frequency of itemsets and association rules can be quantified using the support measure. The support of an itemset (or association rule) is the fraction of all transactions that contain all items in the itemset (or association rule). Thus, using the data in Table 3, the itemset {Ketchup, Soda} and the association rule {Ketchup} → {Soda} both have a support of 2/5, or .4. The user must specify the minimum support level, *minsup*, that is acceptable and only association rules that satisfy that support will be generated.

Not all association rules that satisfy the support constraint will be useful. The confidence of an association rule attempts to measure the predictive value of the rule. The confidence of a rule is calculated as the fraction of transactions that contain the items on the right-hand side of the rule given that the transaction contains the items in the left-hand side of the rule. Using the data in Table 3, the association rule {Ketchup} → {Soda} has a confidence of 2/2 or 1.0, since Soda is always found in the transaction if Ketchup is purchased. The confidence of the association rule {Soda} → {Ketchup} is only 2/3, or .66, since Ketchup only occurs in two of the three transactions that contain Soda. In association rule mining the user must also specify a minimum confidence value, *minconf*. Table 4 shows the results of an association rule mining algorithm, such as the Apriori algorithm, when applied to the data in Table 3 with *minsup* = .4 and *minconf* = .75. Note that more than one item may appear on either side of the association rules, but does not occur for this example due to the simplicity of the sample data.

Table 4: Association Rules Generated from Market Basket Data (*minsup*=.4 and *minconf*=.75)

Association Rule	**Support**	**Confidence**
{Ketchup} → {Soda}	0.4	1.0
{Cereal} → {Milk}	0.4	1.0
{Greeting Card} → {Cake}	0.4	1.0
{Cake} → {Greeting Card}	0.4	1.0

Sequential Pattern Mining

Sequential pattern mining in transactional data is a variant of association analysis, in that one is looking for pattern between items in a sequence, rather than in a set of items. As an example, suppose a company rents movies and keeps records of all of the rental transactions. The company can then mine these data to determine patterns within the sequences of movie rentals. Some patterns will be expected and hence may not have much business value, such as Star Wars Episode I → Star Wars Episode II, but less obvious patterns may also be found. Sequential patterns abound and sequence mining algorithms have been applied to a variety of domains. For example, sequence mining has been applied to sequences of network alarms in order to predict network equipment failures (Weiss and Hirsh 1998), to computer audit data in order to identify network intrusions (Lee, Stolfo and Mok 2000), to biological sequences to find regions of local similarity (Altschul et al. 1990), and to web clickstream data to find web pages that are frequently accessed together (Tan and Kumar 2002).

CLUSTER ANALYSIS

Cluster analysis (Jain, Murthy, and Flynn 1999; Parsons, Haque, and Liu 2004) automatically partitions data into meaningful groups based on the characteristics of the data. Similar objects are placed into the same cluster and dissimilar objects are placed into different clusters. Clustering is an unsupervised learning task in that the training data do not include the "answer" (i.e., a mapping from example to cluster). Clustering algorithms operate by measuring the similarity and dissimilarity between objects and then finding a clustering scheme that maximizes intra-cluster similarity and inter-cluster dissimilarity. Clustering requires that a similarity measure be defined between objects, which, for objects with numerical attributes, may be the Euclidean distance between the points. Figure 6 shows one possible clustering of eleven objects, each described by three attributes. The cluster boundaries are denoted by the dashed shapes.

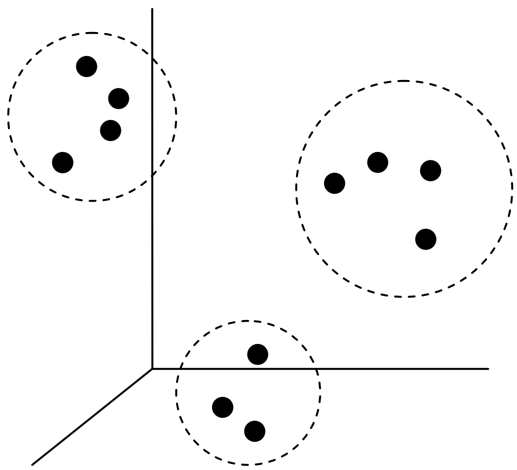


Figure 5: Eleven Examples Placed into Three Clusters

Uses of Clustering

There are many reasons to cluster data. The main reason is that it allows us to build simpler, more understandable models of the world, which can be acted upon more easily. People naturally cluster objects for this reason all the time. For example, we are able to identify objects as a "chair" even if they look quite different and this allows us to ignore the specific characteristics of a chair if they are irrelevant. Clustering algorithms automate this process and allow us to exploit the power of computer technology. A secondary use for clustering is for dimensionality reduction or data compression. For example, one could identify ten attributes for a data set, cluster the examples using these attributes, and then replace the ten attributes with one new attribute that specifies the cluster number. Reducing the number of dimensions (i.e., attributes) can simplify the data mining process. Clustering can also aid with data compression by replacing complex objects with an index into a table of the object closest to the center of that objects cluster.

There are many specific applications of clustering and we list only a few here. Clustering can be used to automatically segment customers into meaningful groups (e.g., students, retirees, etc.), so that more effective, customized, marketing plans can be developed for each group. In document retrieval tasks the returned documents may be clustered and presented to the users grouped by these clusters (Zamir and Etzioni 1998) in order to present the documents to the user in a more organized and meaningful way. For example, clustering can be employed by a search engine so that the documents retrieved from the search term "jaguar" cluster the documents related to the jaguar animal separately from those related to the Jaguar automobile (the ask.com search engine currently provides this capability). The clustering algorithm can work effectively in this case because one set of returned documents will repeatedly have the term "car", "automobile" or "S-type" in it while the other set may have the terms "jungle" or "animal" appear repeatedly.

Categories of Clustering Algorithms

Clustering algorithms can be organized by the basic approach that they employ. These approaches are also related to the type of clustering that the algorithm produces. The two main types of clusterings are hierarchical and non-hierarchical. A hierarchical clustering has multiple levels while a non-hierarchical clustering has only a single level. An example of a hierarchical clustering is the taxonomy used by biologists to classify living organisms (although that hierarchy was not formed using data mining algorithms).

The non-hierarchical clustering algorithms will take the presented objects and place each into one of *k* clusters, where each cluster must have at least one object. Most of these algorithms require the user to specify the value of *k*, which is often a liability, since the user will generally not know ahead of time the optimal number of meaningful clusters. The framework used by many of these algorithms is to form an initial random clustering of the objects and then repeatedly move objects between clusters to improve the overall quality of the clusters. One of the oldest and most notable of these methods is the K-means clustering algorithm (Jain and Dubes 1988). This algorithm randomly assigns each object to one of the *k* clusters and then computes the mean (i.e., center or centroids) of the points in the cluster. Then each object is reassigned to the cluster based on which centroid it is closest to and then the centroids of each cluster are recomputed. This cycle continues until no changes are made. This very simple method sometimes works well. Another way to generate non-hierarchical clusterings is via density-based clustering methods, such as DBSCAN (Ester et al. 1996), which find regions of high density that are separated from regions of low density. One advantage of DBSCAN is that because it is sensitive to the density differences it can form clusters with arbitrary shapes.

Hierarchical clustering algorithms are the next type of clustering algorithms. These algorithms can be divided into agglomerative and divisive algorithms. The agglomerative algorithms start with each object as an individual cluster and then at each iteration merge the most similar pair of clusters. The divisive algorithms take the opposite approach and start with all objects in a single partition and then iteratively split one cluster into two. The agglomerative techniques are by far the more popular method. These methods are appropriate when the user prefers a hierarchical clustering of the objects.

TEXT, LINK AND USAGE MINING

In this section we focus on mining unstructured and semi-structured, non-numeric data. While these data cannot be effectively stored in a conventional relational database, it is the dominant form for human communication, especially given the advent and explosive growth of email, instant messaging and the World Wide Web (WWW). In many cases the data mining tasks associated with these data are not new, but are described in this section because of their importance and because they typically utilize specialized data mining methods. As an example, text mining tasks include classification (i.e., text classification) but the methods used must take into account the unstructured nature and high dimensionality of the data. Other data mining tasks, such as link mining, can be considered a new type of data mining task, although they may still be used in conjunction with existing tasks (i.e., link mining can be used to aid in classification).

Text Mining

The basic unit for analysis in text mining is a document. A document can contain an arbitrary number of terms from an arbitrarily large vocabulary, which is the union of all terms in the collection of documents being analyzed. If one represents a document using attributes that denote the presence or absence of each term, then the number of attributes will generally be very large (in the thousands or millions, depending on the collection). This causes difficulty for most data mining algorithms and thus some text-specific methods are often needed.

Text representation.

While a number of methods for representing text have been developed, essentially all use the same framework. A document (e.g., a web page, blog post, book, or search query) is treated as a “bag” of words or terms, which means that the order of the terms is ignored and only the distinct set of terms is stored, along with a weight corresponding to its importance or prevalence. In the simplest model the weight might be a Boolean value, indicating whether the term occurs in the document. The vector space model, in contrast, encodes a document as a vector of real-valued weights that might incorporate the relative frequency of the term in the document and the relative popularity of the term in all documents in the collection.

Text classification and clustering.

In text classification, two methods are common. The naive Bayes algorithm provides a computationally inexpensive method for text classification, which can be interpreted probabilistically. When higher accuracy is required, support vector machines (Vapnik 1999) are used. The support vector machine (SVM) method operates by learning a hyperplane to separate two classes in a real-valued multidimensional space. Modern SVMs (Joachims 1998) are designed to efficiently handle large numbers of attributes but still provide high predictive accuracy.

Document clustering is similar to other forms of clustering. Typically the K-means method described earlier is used, where each example corresponds to the term vector used to represent each document. However, in this case the similarity function used in the clustering process incorporates the weights associated with each term. The most common similarity measure calculates the cosine of the angle between the two vectors of term weights.

Link Mining

Many kinds of data are characterized by relationships, or links, between entities. Such relationships include co-authorship and co-citation (i.e., scholarly articles cited in the same article). Hyperlinks between web pages form a particularly useful relationship. These relationships can form large graphs, where the entities correspond to nodes in the graph and the relationships correspond to the edges in the graph. In some cases these relationships have been studied for decades. In social network analysis (Wasserman and Faust 1994) the relationships and interactions between people are analyzed to find communities or participants with particular centrality or prestige within the network. In bibliometrics the citation network of scholarly publications is analyzed, using relationships such as co-citation (Small 1973) and bibliographic coupling (Kessler 1963), to determine the importance of authors, papers, and venues.

Research in the WWW community has rekindled interest in these ideas and has provided substantial applications. In particular, the graph of the Web provides something similar to a citation network, since links are often construed as recommendations or citations from one page to another. Google's PageRank algorithm (Page et al. 1998), for example, uses the idea of rank prestige from social network analysis. In it, a page's importance is dependent not only on the number of votes (i.e., links) received from other pages, but also on the importance of those pages. Kleinberg (1999), in contrast, used bibliometric ideas to define measures for web hubs and authorities. In his HITS model, a good web hub is a page that points to a number of good web authorities; similarly, a good web authority is a page to which many good hubs point. Both PageRank and HITS utilize recursive definitions that when applied to all pages simultaneously, correspond to the calculation of the eigenvector of the matrix form of the web graph. The authority values generated by this process are used by search engines such as Google and Ask.com in combination with estimates of query relevance to rank pages for retrieval. Given the large number of relevant results for most queries, estimates of page importance have become essential in generating useful result rankings.

Content Mining

While mining the link structure of the Web is significant, there is an enormous amount of data within web pages that is ripe for mining. Given a web page, one might first ask what the page is about (sometimes called "gisting"). Assigning a web page to one of a number of possible classes is more difficult than traditional text classification—the Web places no restrictions on the format, length, writing style, validity, or uniqueness of the content of a web page. Fortunately, careful use of the content of neighboring pages can dramatically improve classification accuracy (Qi and Davison 2006). Topical classification of web pages is particularly important for contextual advertising, for automated web directory creation, and focused crawling.

Many web pages contain structured data that are retrieved from some underlying but otherwise inaccessible database. This structured data includes product information from online stores, job postings, search results, news articles, and much more. The process of selecting these data from the page in which it is embedded is called data extraction, and can be automated in a supervised or unsupervised manner (Liu 2007). Such data can then be mined for knowledge more directly as homogeneous records. An even greater amount of data is believed to be available via the "deep Web" – those pages that result from content submitted through forms (typically for a search or database lookup) – and are available for harvesting when the appropriate form content to submit is known.

Even when pages are classified as being in a category of interest, the content within it (perhaps already extracted from the rest of the page) may still be unstructured. A common concern for many organizations is to determine not only which pages discuss topics or products of interest, but also what the attitudes are with respect to those topics or products. The Web has resulted in a huge expansion of the ways that customers can express their opinions, i.e., though blogs, discussion groups and product reviews on merchant sites. Mining these opinions provides organizations with valuable insights into product and brand reputations, insights into the competition, and consumer trends. Opinion mining (Liu, 2007) can also be of value to customers who want advice before a purchase and to advertisers who want to promote their products. The simplest form of opinion mining is sentiment classification, in which the text is classified as being positive or negative. For more detail, feature-based opinion mining and summarization might be performed to extract details, such as product characteristics mentioned, and determine whether the opinions expressed were positive or negative. A third variation would be to search for explicit comparisons within the opinions and thus be able to provide relative comparisons between similar products.

Web Usage Mining

The content and structure of the Web provide significant opportunity for web mining, as described above. Usage of the Web also provides tremendous information as to the quality, interestingness, and effectiveness of web content, and insights into the interests of users and their habits. By mining clickstream data and other data generated by users as they interact with resources on one or more web sites, behavioral patterns can be discovered and analyzed. Discovered patterns include collections of frequent queries or pages visited by users with common interests. By modeling user behavior it is possible to personalize a web site, improve web performance by fetching web pages before they are needed, and to cross-sell or up-sell a potential customer for improved profitability. Such analysis is possible on the basis of web server logs, but stronger models are possible with clickstream traffic captured by a web proxy as it can also capture cross-server activity.

Another record of web activity is from search engines. The queries submitted represent the express interests or information needs of the searchers and thus provide an unmatched view into the interests, needs, and desires of a user population. However, in order to characterize a query log, one must first be able to classify the queries. Query classification is also important for monetization of search through relevant advertising. In general, query classification is known to be difficult, primarily because typical query strings are short and often ambiguous. As a result, some additional knowledge is normally used, such as incorporating the text of the results of the query to provide expanded content for classification.

In addition to queries, search engine providers also capture the clicks corresponding to which results the searcher visited. Analysis of the patterns of clicks can provide important feedback on searcher satisfaction and how to improve the search engine rankings (Joachims et al. 2005).

Finally, it is important to note that the collection, storage, transmission and use of usage data is often subject to legal constraints in addition to privacy expectations. Not surprisingly, methods for the anonymization of user data continue to be an active research topic.

DATA MINING RESOURCES & TOOLS

For those wishing to obtain more information on data mining, there are a number of general resources. A good electronic resource for staying current in the field is KDnuggets (http://kdnuggets.com/), a website that provides information on data mining in the form of news articles, job postings, publications, courses, and conferences, and a free bimonthly email newsletter. There are a number of general textbooks on data mining. Those who have some background in computer science and are interested in the technical aspects of data mining, including how the data mining algorithms operate, should consider the texts by Han and Kamber (2006) , Tan, Steinbach and Kumar (2006) and Liu (2007). Those with a business background, or whose primary interest is in how data mining can address business problems, may want to consider the texts by Berry and Linoff (2004) and Pyle (2003). The primary journals in the area are *Data Mining and Knowledge Discovery* and *ACM's Transactions on Knowledge Discovery from Data* and the primary professional organization is ACM's Special Interest Group (SIG) on Knowledge Discovery and Data Mining (http://www.sigkdd.org). Major conferences in the field include the *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* and the *IEEE International Conference on Data Mining*. A comprehensive list of data mining texts, journals, and conferences is available from http://kdnuggets.com/.

There are a wide variety of data mining tools available. Some of these tools implement just one data mining method (e.g., decision trees) whereas others provide a comprehensive suite of methods and a uniform interface. Many of the tools provided by the academic community are available for free, while many of the commercial data mining tools can be quite expensive. The commercial tools are frequently provided by companies that also provide statistical tools and in these cases are often marketed as an extension to these tools.

One of the most frequently used tools for research has been C4.5 (Quinlan 1993), a decision tree tool, which is still freely available for download over the Internet (www.rulequest.com/Personal/). However, this tool is no longer supported and its capabilities are somewhat dated. A more powerful commercial version of this product, C5.0, is available for a modest price from Rulequest Research (www.rulequest.com). There are a number of powerful data mining packages that provide support for all major data mining algorithms. These packages also provide support for the entire data mining process, including data preparation and model evaluation, and provide access through a graphical user interface so that a programming background is not required. These data mining packages include Weka (Witten and Frank 2005), which is free and available for download over the Internet and commercial packages such as Enterprise Miner and Clementine, from SAS Institute Inc. and SPSS Inc., respectively. A more complete list of data mining tools is available from KDnuggets at www.kdnuggets.com/software.

CONCLUSION

Data mining initially generated a great deal of excitement and press coverage, and, as is common with new "technologies", overblown expectations. However, as data mining has begun to mature as a discipline, its methods and techniques have not only proven to be useful, but have begun to be accepted by the wider community of data analysts. As a consequence, courses in data mining are now not only being taught in

Computer Science departments, but also in most business schools. Even many of the social sciences that have long relied almost exclusively on statistical techniques have begun to realize that some knowledge of data mining is essential and will be required to ensure future success.

All "knowledge workers" in our information society, particularly those who need to make informed decisions based on data, should have at least a basic familiarity with data mining. This chapter provides this familiarity by describing what data mining is, its capabilities, and the types of problems that it can address. Further information on this topic can be acquired via the resources listed in the previous section.

GLOSSARY

Artificial Neural Network (ANN) – A computational device, inspired by the brain and modeled as an interconnected set of nodes, which learns to predict by adjusting the weights between its nodes so that the output it generates better matches the predicted value encoded with the training examples.

Association analysis – A data mining task that looks for associations between items that occur within a set of transactions. Example: if Hamburger Meat is purchased then Ketchup is purchased 30% of the time.

Bayesian Classifier – A probabilistic classifier that determines, via Bayes' theorem, the most likely class given a set of observed attributes.

Classification – The predictive data mining task that involves assigning an example to one of a set of predefined classes. Example: predicting who will default on a loan.

Cluster Analysis – The data mining task that automatically partitions data into clusters (i.e., groups) such that similar objects are placed into the same cluster and dissimilar objects are placed into different clusters.

Data mining process – The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al. 1996). Synonymous with the knowledge discovery process.

Data mining step – The algorithmic step in the data mining process that extracts patterns from the data.

Description task – The data mining task that summarizes the data in some manner. Clustering and association rule mining are the two main descriptive data mining tasks.

Predictive accuracy – The fraction, or percentage, of predictions that are correct.

Prediction task – The data mining task that involves predicting a value based on other existing information. The main predictions tasks are classification and regression tasks.

Regression task – The predictive data mining task that involves mapping an example to a numerical, possibly continuous, value. Example: predicting a future stock price.

Supervised learning – A type of learning task where the "answer" is provided along with the input.

Test Set – The labeled data used for predictive data mining tasks that is reserved to evaluate the effectiveness (e.g., accuracy) of the predictive model built using the training data.

Training set – The data provided as input to a data mining algorithm that is used to train or build a model.

Unsupervised learning – A type of learning task where the "answer" is not provided along with the input. Clustering and association rule mining are the main examples of unsupervised learning in data mining.

REFERENCES

Agrawal, R., Imielinki, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM-SIGMOD International Conference on Management of Data*, 207-216, Washington, DC.

Agrawal, R., and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 1994 International Conference on Very Large Databases,* 487-499, Santiago, Chile.

Agrawal, R., and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering*, 3-14, Taipei, Taiwan.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403-410.

Berry, M., and Linoff, G. S. (2004). *Data Mining Techniques for Marketing, Sales, and Customer Relationship Management.* Wiley.

Breiman, L. 1996. Bagging predictors. *Machine Learning*, 24(2):123-140.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth International Group.

Chakrabarti, S. (2002). *Mining the Web: Statistical Analysis of Hypertext and Semi-Structured Data*. Morgan Kaufmann.

Chen, Y., Zhang, G, Hu, D., and Wang, S. (2006). Customer segmentation in customer relationship management based on data mining. In *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*, 288-293. Boston: Springer.

Cohen, W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning,* 115-123, Tahoe City, CA.

Cortes, C., and Pregibon, D. (1998). Giga-mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 174-178.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13: 21-27.

Enke, D., and Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29(4):927-940.

Ester, M., Frommelt, A., Kriegel, H., and Sander, J. (1998). Algorithms for characterization and trend detection in spatial databases. In *Proceedings of the International Conference of Knowledge Discovery and Data Mining*, p. 44-50, New York, NY, August 1998.

Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 226-231.

Fayyad, U. M. (2003). Editorial. *SIGKDD Explorations*, 5(2).

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37-54.

Fawcett, T., and Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3): 291-316.

Freund, Y. and Schapire, Y.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139.

Gurney, K. (1997). *An Introduction to Neural Networks*. CRC Press.

Han, J., and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Hand, D. J. (1998). Data Mining: Statistics and More? *American Statistician*, 52(2): 112-118.

Hsu, W., Lee, M. L., and Zhang, J. (2002). Image mining: Trends and developments. *Journal of International Information Systems,* 19:7-23.

Huang, Y., Shekhar, S., and Xiong, H. (2004). Discovering co-location patterns from spatial datasets: a general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1472-1485.

International Data Corporation (2007). The expanding digital universe: A forecast of worldwide information growth through 2010. http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf.

Jain, A.K., and Dubes, R. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Jain, A. K., Murthy, M. N., and Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Reviews*, Nov 1999.

Joachims, T. (1998) Making large-scale support vector machine learning practical. In Scholkopf, Burges and Smola (ed), *Advances in Kernel Methods: Support Vector Machines*. Cambridge, MA: MIT Press.

Joachims, T., Granka, L., Pan, B., Hembrooke, H. and Gay, G. (2005). Accurately Interpreting Clickthrough Data as Implicit Feedback. In *Proceedings of the 28th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil.

Kessler, M. (1963). Bibliographic Coupling between Scientific Papers. *American Documentation*, 14.

Kleinberg, J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM* 46(5):604-632.

Kuramochi, M., and Karypis, G. (2001). Frequent Subgraph Discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, 313-320, San Jose, CA.

Langley, P., Iba, W., and Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence*, 223-228.

Lee, W., Stolfo, S. J., and Mok, K. W. (2000). *Adaptive intrusion detection: a data mining approach. Artificial Intelligence Review*, 14(6):533-567.

Ling, C. X., and Li, C. (2000). Applying Data Mining to Direct Marketing. In W. Kou and Y. Yesha (eds.), *Electronic Commerce Technology Trends: Challenges and Opportunities*, 185-198, IBM Press.

Liu, B. (2007) *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. New York: Springer-Verlag.

Page, L., Brin, S., Motwani, R. and Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Computer Science Department, Stanford University.

Parsons, L., Haque, E., and Liu, H. Subspace clustering for high dimensional data: A review. *SIGKDD Explorations*, 6:90-105.

Pyle, D. (1999). *Data preparation for Data Mining*. Morgan Kaufmann.

Pyle, D. (2003). *Business Modeling and Data Mining*. Morgan Kaufmann.

Qi, X. and Davison, B. D. (2006) Knowing a Web Page by the Company It Keeps. In *Proceedings of the 15th ACM Conference on Information and Knowledge Management (CIKM),* pp. 228-237, Arlington, VA.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533-536.

Sebastaiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1-47.

Small, H. (1973) Co-Citation in the Scientific Literature: a New Measure of the Relationship between Two Documents. *Journal of American Society for Information Science*, 24(4), pp. 265-269.

Tan, P. N., and Kumar, V. (2002). Mining association patterns in web usage data. In *Proceedings of the International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet*.

Tan, P., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Pearson Addison Wesley.

Vapnik, V. (1999) *The Nature of Statistical Learning Theory, 2nd Ed.* New York: Springer-Verlag.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge, MA: Cambridge University Press.

Wei, C., and Chiu, I (2002). Turning telecommunications call details to churn prediction: A data mining approach. *Expert Systems with Applications*, 23(2):103-112.

Weiss, G. M., and Hirsh, H. (1998). Learning to predict rare events in event sequences. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, 359-363, Menlo Park, CA.

Witten, I. H., and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques,* 2nd edition, Morgan Kaufmann, San Francisco.

Zamir, O., and Etzioni, O. (1998). Web document clustering: A feasibility study. In *Proceedings of the 21st International ACM Conference on Research and Development in Information Retrieval*, 46-54.

Zhu, X., Wu, X., Elmagarmid, A. K., Feng, Z., and Wu, L. (2005). Video data mining: semantic indexing and event detection from the association perspective. *IEEE Transactions on Knowledge and Data Engineering*, 17(5): 665-677.