CHAPTER 41

Knowledge Discovery and Data Mining

Chih-Ping Wei[1], *Selwyn Piramuthu*[2], *and Michael J. Shaw*[3]

[1]Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan

[2]Department of Decision and Information Sciences, University of Florida, Gainesville, FL, USA

[3]Department of Business Administration and Beckman Institute for Advanced Sciences and Technology, University of Illinois, Urbana, IL, USA

With the advances of information technology and widespread diffusion of databases systems in organizations, large volumes of data are generated and collected by organizations. This dramatic expansion of data has generated an urgent need for new analysis techniques that can intelligently and automatically transform the processed data into useful information and knowledge. As a result, knowledge discovery and data mining have increased in importance and economic value. Knowledge discovery refers to the overall process of discovering useful knowledge from data, while data mining refers to the extraction of patterns from data. This chapter provides a reasonably comprehensive review of knowledge discovery and its associated data mining techniques. Based on the kinds of knowledge that can be discovered in databases, data mining techniques can be broadly structured into several categories, including classification, clustering, dependency analysis, data visualization, and text mining. Representative data mining techniques for each category are depicted in this chapter.

Keywords: Knowledge Discovery; Data Mining; Classification; Clustering; Dependency Analysis; Data Visualization; Text Mining

1 Introduction

With the advances of information technology and widespread diffusion of databases systems in organizations, large volumes of data (transactional or administrative) are generated and collected by organizations. This dramatic expansion of data, as measured by sheer volume and repositories, has generated an urgent need for new analysis techniques that can intelligently and automatically transform the processed data into useful information and knowledge (Chen et al., 1996). Consequently, knowledge discovery in databases (hereafter called knowledge discovery, for short) has increased in importance and economic value. Knowledge discovery in databases refers to a process of nontrivial extraction of implicit, previously unknown, and potentially useful knowledge from large databases for crucial business decision support (Frawley et al., 1991; Chen et al., 1996). Different researchers have defined knowledge discovery and data mining differently. For example, in the view of Frawley et al. (1991), data mining refers to the extraction of patterns from data and knowledge discovery in databases refers to the overall process of discovering useful knowledge from data. On the other hand, Chen et al. (1996) regarded both terms as synonyms. In this chapter, we take the former view and consider that data mining as a phase in the knowledge discovery process, which are detailed in Section 1.1.

Knowledge discovery (or specifically, data mining) has been successfully adopted by various industries. For example, it has been applied to the healthcare domain for improving medical decision making (e.g., the screening of breast cancer (Ronco, 1999), the prediction of the risk of coronary disease (Azuaje et al., 1999), and the diagnosis of ischaemic heart disease (Kukar et al., 1999) and colorectal cancer (Anand et al., 1999)), supporting patient management (e.g., the tracking of morbidity outcomes in trauma care (Marble and Healy, 1999)), and facilitating quality assurance (e.g., inappropriate medical treatment detection (Cabena et al., 1997)). On the other hand, to survive or maintain an advantage in an ever-increasing competitive marketplace, many telecommunications companies are turning to data mining to resolve such challenging issues as fraud detection (Ezawa and Norton, 1996), customer retention (Berson et al., 2000), and prospect profiling (Kappert and Omta, 1997).

1.1 Process of Knowledge Discovery

The process of knowledge discovery considers the entire process from the onset of data to the generation of knowledge. As shown in Figure 1, it consists of five phases, including selection, preprocessing, transformation, data mining, and interpretation/evaluation (Frawley et al., 1991).

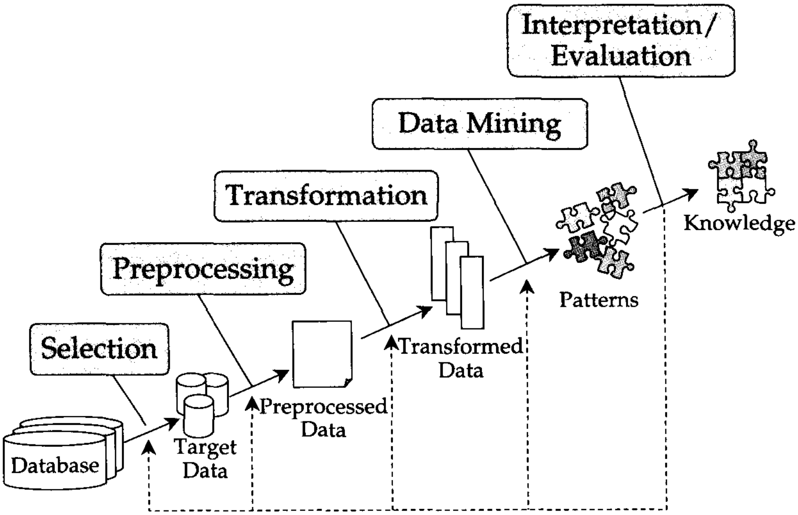


Figure 1. Process of Knowledge Discovery

The *selection* (or called data prospecting) phase aims at selecting a data set or focusing on a subset of variables or data samples from various databases, on which the knowledge discovery is to be performed. After the selection phase, the target data typically come from different data sources with varying levels of quality. Thus, the *preprocessing* phase deals with such issues as data integration (e.g., instance identification for resolving the problem when the same real-world instance is represented using different identifiers in different databases (Dey et al., 1998), attribute value transformation for converting attribute values to the same measurement unit or the same domain, etc.) and data cleaning (e.g., identification and removal of noise or outliers, inconsistency resolution, etc.), and decides on strategies for handling missing data fields. The *transformation* phase transforms the preprocessed data into a particular representational form as required by the data mining technique to be used. The *data mining* phase extracts patterns from data by using a data mining technique(s). Finally, the discovered patterns are be interpreted, checked for potential conflicts with previously believed (or extracted) knowledge, and, if needed, incorporated into the performance system in the *interpretation/evaluation* phase.

1.2 Classification and Applications of Data Mining Techniques

Based on the kinds of knowledge that can be discovered in databases, data mining techniques can be broadly classified into several categories, including classification, clustering, dependency analysis, data visualization, and text mining. Table 1 identifies some data mining applications by functional areas and identifies the data mining techniques used.

Classification analysis is a process that induces a classification model to classify a set of pre-classified instances (called training examples) into classes. Such classification model is then used to classify future instances. Major classification techniques can be classified into the following types, including decision tree induction, decision rule induction, backpropagation neural network, nearest neighbor classification, and Bayesian networks. On the other hand, clustering analysis is a process whereby a set of instances (without a predefined class attribute) is partitioned (or grouped) into several clusters in which all instances in one cluster are similar to each other and different from the instances of other clusters, according to some distance metric. Three main clustering approaches include partitioning-based, hierarchical, and neural-network-based.

Table 1. Some Data Mining Applications

Area	Application	Data Mining Technique(s) Used
Finance	Prediction of corporate failure (Lin & McClean, 2001)	Classification
	Forecasting exchange rates (Leung et al., 2000)	Classification
	Credit assessment (West, 2000; Carter & Catlett, 1987)	Classification

	Loan risk assessment (Gerritsen, 1999)	Classification
	Risk management (Wright, 1997)	Data visualization
	Forecasting interest rates (Kim & Noh, 1997)	Classification
	Portfolio management (John et al., 1996)	Classification
	Fraud detection (Ezawa & Norton, 1996)	Classification
Marketing	Detection of customer behavior change (Song et al., 2001)	Dependency analysis
	Churn prediction (Berson et al., 2000)	Classification
	Customer profiling (Kappert & Omta, 1997)	Classification
	New product introduction campaign (Berry & Linoff, 1997)	Classification
	Target marketing of home equity loans (Berry & Linoff, 1997)	Classification, clustering, and dependency analysis
	Product performance analysis (Wright, 1997)	Data visualization
	Analysis of lifestyle behavior for target marketing (Lee & Ong, 1996)	Data visualization
Healthcare	Hypertension prediction and management (Chae et al., 2001)	Classification and dependency analysis
	Discovery of clinical pathways (Lin et al., 2001)	Dependency analysis
	Survival prediction in damage control surgery (Aoki et al., 2000)	Classification
	Predicting blood transfusion requirements (Walczak & Scharf, 2000)	Classification
	Selective breast cancer screening (Ronco, 1999)	Classification
	Diagnosis of ischaemic heart disease (Kukar et al., 1999)	Classification
	Diagnosis of colorectal cancer (Anand et al., 1999)	Classification
	Inappropriate medical treatment detection (Cabena et al., 1997)	Dependency analysis
Others	Web caching (Bonchi et al., 2001)	Dependency analysis and classification
	Web personalization (Lee et al., 2001)	Dependency analysis
	Tunnel support stability prediction (Leu et al., 2001)	Classification

Weather prediction (Feng et al., 2001)	Dependency analysis
Diagnosis of river pollution (Walley & O'Connor, 2001)	Clustering
Customer service support (Hui & Jha, 2000)	Text mining, classification and clustering
Organizing of meeting comments (Roussinov & Chen, 1999)	Text mining
Network alarm analysis (Klemettinen et al., 1999)	Dependency analysis
Email routing (Weiss et al., 1999)	Text mining
Event detection (Yang et al., 1998; Yang et al., 1999)	Text mining
Prediction of road surface's temperature (Luchetta et al., 1998)	Classification

Dependency analysis discovers dependency patterns embedded in data. Types of dependency patterns include association rules, sequential patterns, temporal patterns, episode rules, etc. On the other hand, data visualization allows decision makers to view complex patterns in the data as visual objects in three dimensions and colors, and supports advanced manipulation capabilities to slice, rotate or zoom the objects to provide varying levels of details of the patterns observed. Data visualization could take several forms including pixel-oriented, geometric projection, icon-based, hierarchical, and graph-based.

Finally, text mining aims to extract patterns from textual documents and can be applied to facilitate document management and retrieval or to discover knowledge hidden in texts. Text mining techniques include text categorization, document clustering, term association discovery, routing and filtering, information extraction, and document summarization. Despite differences in purposes, a text mining technique typically involves text parsing and analysis to transform each unstructured document into an appropriate set of features (e.g., noun phrases) and subsequently applies one or more above-mentioned data mining techniques for extracting patterns from the feature space. Because text mining deals with textual documents rather than structured data, its approaches can be treated as novel data mining techniques. On the other hand, because its underlying analysis requires the use of data mining techniques, text mining can also be viewed as applications of data mining.

1.3 Roles of Data Mining in Knowledge Management

Knowledge management is a systematic process for acquiring, organizing, sustaining, applying, sharing, and renewing both tacit and explicit knowledge to enhance the organizational performance, increase organizational adaptability, increase values of existing products and services, and/or create new knowledge-intensive products, processes and services (Davenport et al., 1998; Davenport and Prusak, 1998). One view of a knowledge management process is shown in Figure 2. New knowledge can be created or acquired. The knowledge is then organized by indexing the knowledge elements, filtering based on content and establishing linkages and relationships among the elements. Subsequently, this knowledge is integrated into a knowledge repository and distributed to users for supporting their decision making process. The application of knowledge may result in the knowledge maintenance in which existing knowledge is refined or refreshed.

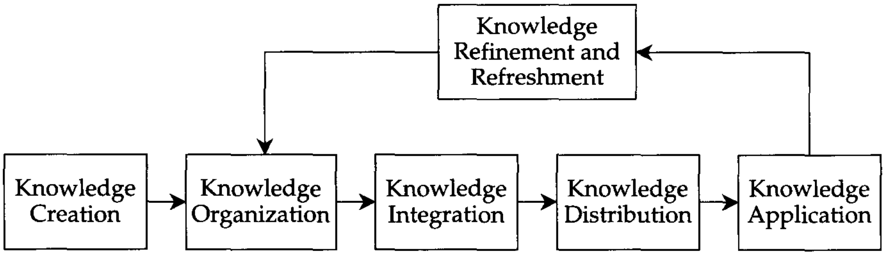


Figure 2. Knowledge Management Process

The ultimate goal of data mining is to extract useful and valid knowledge from large databases. Thus, data mining can serve an active role for knowledge creation. As an organization accumulates more data from their transactional routines, or as decisions and their effectiveness become available after the knowledge application, data mining can participate in the knowledge refinement and refreshment by re-discovering from newer data sets. Besides these apparent applications of data mining in knowledge management, data mining can also contribute to the knowledge organization and distribution. For example, if the knowledge being managed is in the textual format, the knowledge organization activity should be concerned with organizing knowledge documents in a hierarchy of categories to facilitate knowledge users to search and browse these knowledge documents. In addition, the knowledge organization activity should provide support for automatically assigning a knowledge document into one or more categories previously established. Data mining, specifically text mining, can be adopted to provide the desirable functionality to the knowledge organization activity. On the other hand, knowledge usage patterns or behavior can be discovered from the knowledge usage history of knowledge workers. Its discovery can facilitate and foster pushing knowledge proactively to potentially interested knowledge workers.

1.4 Organization of the Chapter

As shown in Figure 3, the remainder of the chapter is organized in terms of the taxonomy of data mining techniques depicted previously. Text mining is highlighted by a dash rectangle in Figure 3 because it can be viewed as involving novel data mining techniques or applications of data mining. An overview on different classification analysis techniques is presented in Section 2. In Section 3, representative clustering techniques for each clustering approach are examined.

Dependency analysis and data visualization techniques are summarized in Section 4 and Section 5, respectively. In Section 6, text mining techniques are reviewed. The chapter concludes with a summary and discussion of important practical considerations when applying data mining techniques.

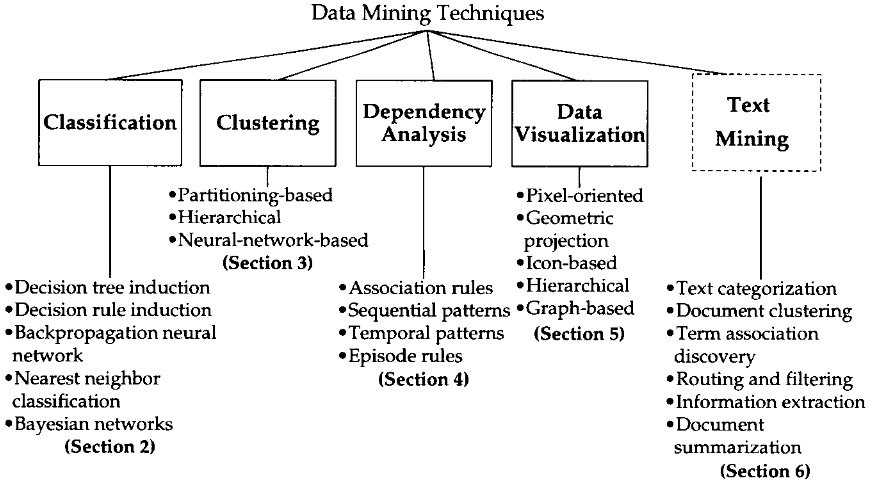


Figure 3. Taxonomy of Data Mining Techniques and Organization of the Chapter

2 Classification Analysis

As mentioned, classification analysis is a process that constructs a classification model for establishing the relationship between classes and attributes from a set of training instances. The class of an instance must be one from a finite set of possible, pre-determined class values, while attributes of the instance are descriptions of the instance potentially affecting its class. Depending on the learning strategies adopted as well as the types of classification models induced, classification techniques can be classified into the following types, including decision tree induction, backpropagation neural network, nearest neighbor classification, decision rule induction, and Bayesian networks. Due to space limitation, we only review the first three types of classification techniques. For interested readers, a well-known decision rule induction technique, CN2, can be found in (Clark and Niblett, 1989) and a survey on Bayesian networks can be found in (Heckerman, 1997).

2.1 Decision Tree Induction

A decision tree induction technique is a supervised learning method that constructs a decision tree from a set of training instances. Decision tree induction has been popular primarily due to its simplicity, both in construction as well as in interpretation. A typical decision tree induction technique, ID3 (Quinlan, 1986), adopts an *iterative* method to construct decision trees, preferring simple trees over complex ones, based on the theory that simple trees are more accurate classifiers for future instances. ID3 attempts to construct a simple (hopefully the minimal) tree by using an *information theoretic* approach that aims at minimizing the expected number of tests to classify an instance. ID3 evaluates the information gain for each attribute and selects the one achieving the highest information gain as a branching attribute in the decision tree (see Figure 4 for ID3 algorithm). Given a set of instances C, if an attribute A_i will be chosen as the branching attribute for C, the information gain of A_i is:

$$gain(A_i) = E(C) - E(A_i)$$

where $E(C) = \sum_j p_j \log_2(p_j)$ is the entropy of the set of instances C, in which p_j is the probability that an instance is in class j and $\log_2 0 = 0$, and $E(A_i) = \sum_k ({}^{n_k}\!/_n) E(C_k)$ is the resultant entropy of C if C is partitioned according to A_i, in which C_k is a disjoint subset of C based on A_i's value, n is the total number of instances in C and n_k is the number of instances in C_k.

1. Start from the root node of the decision tree and assign the root node as the current node C;
2. If all instances in C belong to the same class, then exit;
3. For each attribute A_i that was not selected by C's ancestors
   Compute the information gain for A_i (i.e., $gain(A_i) = E(C) - E(A_i)$);
4. Select the attribute with the maximum information gain as the branching attribute for C;
5. Create child nodes C_1, C_2, ..., and C_n (assume the selected attribute has n values) for C;
6. Assign all instances in C to appropriate child nodes according to the values of the branching attribute;
7. For each node C_i, assign it as the current node C and go to step 2.

Figure 4. ID3 Algorithm

Many other evaluation functions have been proposed in the literature, including *gain ratio*, *gini index* and *chi-square test*. Interested readers are referred to (Mingers, 1989a) for an empirical comparison study on different evaluation functions. In addition, many other decision tree induction techniques have been proposed, including CHAID (Kass, 1980), CART (Breiman et al., 1984), OC1 (Murthy et al., 1994), and C4.5, a descendant of ID3 (Quinlan, 1993). CART (Classification And Regression Trees) is a binary recursive partitioning method. Chi Square Automatic Interaction Detection (CHAID) segments dataset using *chi-square tests* to create multi-way splits. OC1 uses linear combinations of attributes to generate the hyper-planes that separate examples belonging to different categories.

Most of decision tree induction techniques, with OC1 being an exception, that utilize some form of counting algorithms are appropriate to nominal and ordinal attribute types. However, interval-scaled attributes are not well suited and discretization is required. A typical method to discretize an interval-scaled attribute is the binary partition where all possible mid-points between the attribute values are evaluated and the one which results in the best evaluation metric (e.g., the maximal information gain in ID3) will be chosen as the partition point for the attribute. Furthermore, in real-world applications, it is common to encounter instances with attribute values unknown or missing. Several methods have been suggested to deal with missing attribute values, including filling in the average of all the values of the attribute of interest, assigning the most likely value based on similar examples, and assigning the most frequently occurring value for this attribute.

Data noise presents additional challenge for decision tree induction techniques. When a decision tree induction technique induces from noisy data, the resulting tree tends to be very large and the over-fitting problem arises. Many of the branches will reflect chance occurrences in this particular data set rather than representing underlying relationships between attributes and classes. Several pruning methods that identify and remove the least reliable branches have been proposed (see Minger (1989b) and Esposito et al. (1997) for comparative analysis of different pruning methods). Pruning a decision tree will increase the number of classification errors made on the training data, but should decrease the error data on independent test data. Pruning can be done during the creation of a decision tree (called pre-pruning methods) or after a complete decision tree has been constructed for the training data (called post-pruning methods). For example,

Mingers' pre-pruning method (1989b) relies on estimating the importance of each node during the tree creation process. As mentioned, when creating a decision tree, an evaluation metric determines the branching attribute at a node. The evaluation metric reflects how well the chosen attribute splits the data between classes at that node. A critical value is specified and used to prune those nodes that do not reach the critical value. On the other hand, Niblett and Bratko (1986) developed a post-pruning method to find a decision tree that should, theoretically, give the minimum error rate when classifying independent sets of data. Assume a set of data with *k* classes; assume also that we have observed *n* examples of which the greatest number, n_j, are in class *j*. If we predict that all future instances will be in class *j*, the expected error rate, E_k, is:

$$E_k = \frac{n - n_j + k - 1}{n + k}$$

After a complete decision tree is created, the expected error rate is estimated for each non-terminal node if its sub-tree is pruned. Similarly, the expected error rate is estimated if the sub-tree is not pruned using the weighted average (proportional to the number of instances along each branch) of expected error rates from all of its branches. If pruning the sub-tree leads to a smaller expected error rate, the sub-tree will be pruned and the node becomes a leaf.

2.2 Backpropagation Neural Network

Inspired by biological neural networks, an artificial neural network models the biological characteristics of neurons and their interactions (Fu, 1999). An artificial neural network is represented by a set of nodes and weighted links. A node corresponds to a neuron, a link corresponds to a synapse, and the weight of a link corresponds to a synaptic strength. The nodes are often grouped together into linear arrays called *layers*. Several different artificial neural networks that vary in network topology, purpose, and algorithms used to learn the weights of the links have been proposed. For classification analysis purpose, a multi-layered perceptron is commonly adopted. As its name suggests, a multi-layered perceptron has an *input layer*, an *output layer* and one or more *hidden layers*. Hidden layers are used to identify the desired relationships between the input nodes and output nodes. A multi-layered perceptron is a fully-connected network in which each node in one layer is connected in the forward direction to every node in the next layer. It is known as a feed-forward network since data activation proceeds in a forward manner (i.e., from input to output layer).

The backpropagation algorithm is commonly used to train a feed-forward neural network (Rumelhart et al., 1986). In a backpropagation neural network, training is achieved by adjusting its weights each time it sees an input-output pair (i.e., a training instance). Each instance requires two stages: a forward pass and a backward pass. The forward pass involves presenting the training instance to the network and letting activations flow until they reach the output layer. During the backward pass, the network's actual output (from the forward pass) is compared with the target output of the instance and error estimates are computed for the output nodes. The error estimates of the output nodes are then employed to derive error estimates for the hidden nodes. Accordingly, the weights connected to the output nodes can be adjusted in order to reduce those errors in output layer. Finally, errors are propagated back to the weights stemming from the input nodes.

Backpropagation neural networks require numerically-coded input and output data representation. To avoid the effect of the variation of input attribute values on the learning effectiveness, interval-scaled input attribute values need to be standardized into the range between 0 and 1. For a nominal attribute, the following schemes can be taken: binary coding, 1-of-*N* coding (i.e., one input node is used for each value), and 1-of-*N*-1 coding (i.e., *N*-1 input nodes are used for an attribute with *N* possible values). The output of each training instance needs to be encoded into the range between 0 and 1 as well. If the output of a training instance is a continuous value, one output node is sufficient and the standardization is needed. If the output is one from *N* possible classes, 1-of-*N* coding scheme is often adopted (Berry and Linoff, 1997). Thus, based on the encoding scheme employed, the number of input and output nodes is determined by the characteristic of attributes and the classes, respectively. However, the number of hidden layers as well as the number of nodes in each hidden layer requires considerable experimentation undertaking. To lessen the required experimentation effort, some rules of thumb have been suggested. For example, one such heuristic is to choose only one hidden layer with the number of hidden nodes to be anywhere between the number of nodes in the input and output layers (Blum, 1992).

A backpropagation neural network is known for its ability to deal with noise, partial and potentially conflicting data as well as to generalize to situations not encountered previously (Atlas et al., 1990). Furthermore, the numerically coded input and output nodes make a neural network capable of managing interval-scaled input attributes and continuous outputs. However, backpropagation neural networks suffer from a long training time and limited interpretability. The backpropagation algorithm is a gradient-descent method and, by its iterative nature, takes a long time to converge. On the other hand, the backpropagation network technique represents a holistic approach to learning by encoding the classification model in the weights between nodes. Explaining the knowledge that is embedded in the network structure is a challenging task. There have been several attempts at alleviating this interpretability problem through generation of IF-THEN rules from the weights (e.g., Fu and Shortliffe, 2000; Tickle et al., 1998).

2.3 Nearest Neighbor Classification

Classification problems where no knowledge of the underlying distribution except those that can be inferred from the examples are mostly in the domain of nonparametric statistics (Cover and Hart, 1967). In these cases, it is reasonable to assume that the examples that close together, in some distance metric, belong to the same category. To this end, given an instance with unknown classification, instances with known classification that are closer to this instance are given more weights. The simplest solution is the 1-nearest-neighbor classification rule. When k nearest neighbors are considered to determine the classification, it is known as the k-nearest neighbor (k-NN) search.

The nearest neighbor search problem can be defined as: given a set of points S in an n-dimensional space, find the closest point (or k closest points) in S to a query point p (where $p \notin S$). The nearest neighbor problem has been solved for optimality for lower dimensions. For example, it has been solved optimally for the 1-dimensional case, where the nearest neighbor of a query point can be easily determined by performing a binary search on all points. For 2-dimensional cases, Voronoi diagrams can be used to determine the nearest neighbors (e.g., Dobkin and Lipton, 1976). However, there has not been much progress in cases involving higher dimensionality. Most solutions either result in data structures that require exponentially large storage space in terms of the number of dimensions or require query time that is close to that of a linear scan of the data points. For dimensions > log(number of sample points), a brute force search is usually the best both in theory and practice to circumvent problems associated with the curse of dimensionality. When the nearest neighbor search problem is relaxed to accommodate points that are almost as close as the nearest neighbor points, the search performance can be improved. The relaxed problem is known as the approximate nearest neighbor search. Several approximate nearest neighbor search algorithms have been proposed (Arya et al., 1994; Beyer et al., 1999).

3 Clustering Analysis

The goal of clustering analysis is to partition (or group) a set of instances into clusters such that each cluster indicates a group of instances that are more strongly associated with each other than with those in different clusters. Clustering is one of the unsupervised learning methods in machine learning. As mentioned, three main approaches of clustering are partitioning-based, hierarchical, and neural-network-based clustering. We examine each of clustering approaches in this section.

3.1 Partitioning-Based Clustering Approach

The partitioning-based clustering approach performs partitioning on a set of instances into non-overlapping subsets called clusters. Most classical partitioning-based algorithms include K-means (Anderberg, 1973) and PAM (Kaufman and Rousseeuw, 1990). Suppose that n objects described by the attribute vectors $\{x_1, x_2, \ldots, x_n\}$ be partitioned into k clusters, where $k < n$. Let m_i be the mean of the vectors in the cluster i. An object o_j belongs to the cluster i if the distance between o_j and m_i is the minimum. The K-means algorithm is shown as in Figure 5.

```
Randomly initialize the means $m_1, m_2, \ldots, m_k$
Repeat
    Use the means to classify all objects into the clusters
    For $i = 1$ to $k$
            Replace $m_i$ with the mean of all objects in the cluster $i$
    End-for
Until there is no change in any mean
```

Figure 5. K-Means Algorithm

The K-means algorithm is well known for its efficiency in clustering large data sets, but is limited to data sets involving only interval-scaled attributes. To avoid this deficiency, PAM (Partitioning Around Medoids) uses medoids rather than centroids to represent clusters. The medoid of a cluster is the most centrally located object in a cluster. The PAM algorithm finds k clusters in n objects by first finding a representative object for each cluster. Once k medoids have been selected, each non-selected object is classified into the closest medoid according to a distance measure. Subsequently, it repeatedly tries to make a better choice by substituting a medoid m_j with a non-selected object o_h as long as such substitution improves the quality of the clustering (i.e., reduces the average distance between an object and its closest medoid). Assume D is the data set to cluster (with n objects), M is the set of medoids, $rep(M, o_i)$ returns a medoid in M that is closest to the object o_i, and $d(o_i, o_k)$ is the distance between objects o_i and o_k. The cost (i.e., average distance between an object and its closest medoid) of M is:

$$Cost(M,D) = \frac{\sum_{i=1}^{n} d(o_i, rep(M, o_i))}{n}$$

The effect of substituting a medoid m_j with o_h is:

$TC_{jh} = Cost(M', D) - Cost(M, D)$
where M' is the new set of medoids after substituting a medoid m_j in M with an object o_h not in M.

$TC_{jh} > 0$ means that replacing the medoid m_j with o_h would result in a greater average distance between an object and the medoid of its cluster. Thus, if $TC_{jh} > 0$, o_h will not be selected to replace the medoid m_j. Accordingly, the PAM algorithm is shown in Figure 6.

```
Select $k$ medoids arbitrarily.
Repeat
    Compute $TC_{jh}$ for all pairs of objects $m_j$ and $o_h$, where $m_j$ is a medoid
        and $o_h$ is not.
    Select the pair $m_j$ and $o_h$ whose $TC_{jh}$ is the minimal.
    If the minimum $TC_{jh} < 0$ then replace $m_j$ and $o_h$.
Until the minimum $TC_{jh} \geq 0$
Return the $k$ medoids;
```

Figure 6. PAM Algorithm

The most distinct characteristics of clustering analysis is that it often encounters very large data sets, containing millions of objects described by tens or even hundreds of attributes of various types (e.g., interval-scaled, nominal, etc.). This requires that a clustering algorithm be scalable and capable of handling different attribute types. However, most classical clustering algorithms either can handle various attribute types but are not efficient when clustering large data sets (e.g., PAM) or can handle large data sets efficiently but are limited to interval-scaled attributes (e.g., K-means). To response to this requirement, several fast partitioning-based clustering algorithms have been proposed: including CLARA (Kaufman and Rousseeuw, 1990), CLARANS (Ng and Han, 1994), and genetic-algorithm-based clustering methods (Estivill-Castro and Murray, 1997). CLARA is a combination of a sampling approach and the PAM algorithm. Instead of finding medoids for the entire data set, CLARA draws a sample from the data set and uses the PAM algorithm to select an optimal set of medoids from the sample. To alleviate sampling bias, CLARA repeats the sampling and clustering process multiple times and, subsequently, selects the best set of medoids as the final clustering.

On the other hand, CLARANS views the process of finding optimal medoids as searching through a graph, in which each node represents a set of medoids. Two nodes are neighbors if their sets differ by only one object. Instead of using an exhaustive search strategy, CLARANS adopts a serial randomized search. That is, starting from an arbitrary node in the graph, CLARANS randomly checks one of its neighbors. If the neighbor results in a better clustering, CLARANS proceeds to this neighbor; otherwise, CLARANS randomly checks another neighbor until a better neighbor is found or a pre-determined maximal number of neighbors are reached. To avoid being trapped in a suboptimal solution, CLARANS repeatedly starts from different initial nodes and selects the best node as the final clustering.

Finally, genetic-algorithm-based clustering methods employ a genetic algorithm to search for the optimal set of medoids. The genetic algorithm adopts a probabilistic, parallel-randomized-search strategy similar to biological evolution. It starts with a random fixed-sized population of chromosomes, each of which encodes a possible set of medoids. Iteratively, a new generation of the same size is generated by randomly applying genetic operators (including reproduction, crossover and mutation) on probabilistically selected parent chromosomes based on their fitness as a solution. The process terminates as soon as there is no further improvement over generations or after a pre-defined maximal number of generations has been reached. The fittest chromosome of the last generation or among all generations is then selected as the final clustering.

The search strategies employed by these fast clustering algorithms are fundamentally different. Therefore, their performance in terms of clustering quality and execution time for clustering data sets with different characteristics may vary. Wei et al. (2000a) conducted an empirical evaluation, including the effects of data size, number of clusters, degree of cluster distinctness, degree of cluster asymmetry, and level of data randomness on the clustering quality and execution time of these fast clustering algorithms.

All partition-based clustering techniques assume a pre-specified number of clusters. To determine an optimal cluster number for a given set of objects, the silhouette measure that evaluates the degree of cluster separation (Kauffman and Rousseeuw, 1990) and the cover coefficient (Can and Ozkarahan, 1990) can be employed.

3.2 Hierarchical Clustering Approach

The hierarchical clustering approach builds a binary clustering hierarchy whose leaf nodes are the original objects to be clustered. Hierarchical clustering has the advantage over the partitioning-based clustering approach in that the number of clusters need not be pre-specified and that the number of clusters can be decreased (or increased) by simply moving up (or down) the hierarchy. A representative hierarchical clustering algorithm is the hierarchical agglomerative clustering (HAC) method. The HAC method starts with as many clusters as there are objects, where each cluster contains just one object (Voorhees, 1986; Berson and Smith, 1997). Two most similar clusters are merged together to form the next larger cluster. The merging continues until a hierarchy of clusters is built with just a single cluster containing all the objects at the top of the hierarchy.

Same as any partitioning-based clustering technique, the HAC method relies on a distance function. Defining a function for measuring the distance between two clusters each of which contains more than one object is not straightforward. Several methods have been proposed in the literature, including:

1. Joining the clusters whose nearest objects are as near as possible. This is called the *single-link method.* Because clusters can be joined together on the basis of just a single near pair of objects, this method can create long, snakelike clusters.
2. Joining the clusters whose most distant objects are as near as possible. This is called the *complete-link method,* because all objects within the cluster are linked together within some maximum distance. This method favors the creation of small compact clusters.
3. Joining the clusters where the average distance between all pairs of objects is as small as possible. This is called the *group-average link method,* because it considers all objects within the clusters, including the nearest and the most distant. It results in clusters somewhere in between the elongated single-link clusters and the tight complete-link clusters.
4. Joining the clusters whose resulting cluster has the minimum total distance between all objects. This is called the *Ward's method.* It tends to produce a symmetric hierarchy and is good at recovering cluster structure. However, it is sensitive to outliers and has difficulty in recovering elongated clusters.

3.3 Neural-Network-Based Clustering Approach

The self-organizing map (SOM), developed by Kohonen (1989, 1995), is an unsupervised two-layer neural network commonly used for data clustering. An advantage of SOM over other clustering algorithms is its ability to visualize high dimensional data using a two-dimensional grid while preserving similarity between data points as much as possible. In SOM, each input node corresponds to a coordinate axis in the input attribute vector space. Each output node corresponds to a node in a two-dimensional grid. The network is fully connected in that every output node i is connected to every input node j with a connection weight w_{ij}. Thus, an output node can be considered as a point in the input vector space. During the training phase, the objects to be clustered are presented multiple times in order to train the connection weights in such a way that distribution of output nodes represents distribution of the input objects.

Let $X_i(t) = \{x_{i1}(t), x_{i2}(t), \ldots, x_{in}(t)\}$ be the vector of the input object i at time t where $x_{ik}(t)$ is the kth element of the vector (i.e., the kth input node) and $W_j(t) = \{w_{j1}(t), w_{j2}(t), \ldots, w_{jn}(t)\}$ be the weight vector of the output node j at time t where w_{jk} is the connection weight between the output node j and the kth input node. A sketch of the SOM algorithm is shown in Figure 7.

4 Dependency Analysis

Dependency analysis discovers dependency patterns embedded in data. Depending on the characteristics and structure of the data, different types of dependency patterns emerge: association rules, sequential patterns, temporal patterns, episode rules, etc. In the following subsections, data mining techniques for discovering the first three types of dependency patterns are be discussed. An episode rule discovery technique is to find frequent *episodes* from a sequence of events (e.g., alarms in a telecommunication network, user interface actions, etc.), where an episode is defined to be a collection of events that occur relatively close to each other in a given partial order. Once such episodes are known, one can produce rules for describing or predicting the behavior of the sequence. In the section, the discussion on the episode rule discovery are not further be discussed. Interested readers are referred to Mannila et al. (1995) and Mannila and Toivonen (1996) for further details.

4.1 Association Rules

The problem of mining association rules over market basket data was first introduced by Agrawal et al. (1993). Given a set of transactions, each of which contains a collection of items, an association-rule mining technique finds interesting co-occurrence of items in this set of transactions. The association-rule mining technique can be applied to many applications areas. For example, the analysis of retailing transactions helps identify which products tend to be purchased together. This information can be used to suggest new store layouts, which products to put on special, which products to bundle, etc. On the other hand, the analysis of medical diagnosis records (each of which contains details on the diagnosis, prescription and/or treatments recommended) can help detect inappropriate prescriptions and medical treatments (Cabena et al., 1997).

1. Initialize the network and connection weights:
   It is to create a two-dimensional map (grid) of m output nodes (e.g., 20-by-10 map of 200 nodes) and to initialize weights $w_{ij}(0)$ from n input nodes to m output nodes to small random values at time 0.
2. Present each object to the network in order:
   Each object i is represented as an input vector $X_i(t)$ and presented to the network. Each object requires two steps: distance computation (as in the step 3) and weight update (as in the step 4). All objects will be presented to the network multiple times.
3. Compute distances between each input vector and the weight vector of each output node:
   Assume at time t, the object i is processed. The Euclidean distance d_{ij} is computed between the current input vector, $X_i(t)$, and the weight vector $W_j(t)$ for each output node j:
   $$d_{ij} = \sum_{k=1}^{n} (x_{ik}(t) - w_{jk}(t))^2$$
4. Updating weights to the winning output node and its neighbors to reduce their distance:
   Assume at time t, the object i is processed. The output node that produces minimum d_{ij} is selected as the winning node j^*. The weights to the node j^* and its neighbors are updated to reduce the distance between them

and the input vector $X_i(t)$. The neighbor selection algorithm can be found in (Kohonen, 1989). The weight connecting the kth input node to the jth output node (where the jth output node is the node j^* or its neighbor selected) is adjusted as follows:

$$w_{jk}(t+1) = w_{jk}(t) + \eta(t)(x_{ik}(t) - w_{jk}(t))$$

where $\eta(t)$ is an error-adjusting coefficient (0< $\eta(t)$ <1) that decreases over time.

After the weight update, the nodes in the neighborhood of j^* (including j^*) become more similar to the input vector $X_i(t)$.

5. Label regions in the map:
After the network is trained through repeated presentation of all objects. Each output node j is labeled by the most similar object i.
6. Map the objects to the labeled regions:
Each object is then mapped to an output node where the minimum distance is attained.

Figure 7. SOM Algorithm

The association-rule mining problem is formally defined as follows (Agrawal et al., 1993; Agrawal and Srikant, 1994). Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A transaction T contains a set of items X, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The association rule $X \Rightarrow Y$ holds in D with confidence c if c% of transactions in D that contain X also contains Y. The rule $X \Rightarrow Y$ has a support s in D if s% of transactions in D contains $X \cup Y$. Given a set of transactions D, the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum *support* and minimum *confidence*.

The problem of discovering all association rules satisfying the user-specified support and conference thresholds can be decomposed into two subproblems:

1. Find all sets of items that have a certain user-specified minimum support (called large itemsets).
2. Use the large itemsets to generate desired association rules.

Given m different items, the number of possible itemsets is 2^m-1-m. This is because for an association rule to exist, there should be at least two items. A typical supermarket often has at least 10,000 different items. Thus, the number of possible itemsets is $2^{10,000}$-10001 (can you image this number?). On the other hand, the number of transactions is also very large (e.g., millions transactions per year for a supermarket). Under this situation, calculating the supports for all possible itemsets is prohibitively expensive. One of the earlier, and hence well-known, algorithms for fast finding large itemsets is Apriori (Agrawal and Srikant, 1994). Apriori exploits the downward closure property of the support measure to improve the efficiency of search large itemsets. The downward closure property of the support measure means that if an itemset X has a support of at least s, then any subset of X must have a support of at least s; conversely, if an itemset X has a support of less than s, then any itemset Y that contains X definitely has a support of less than s. This property allows us to use large itemsets of size n-1 (i.e., referred to as large (n-1)-itemsets, each of which has *n-1* items) to construct candidate itemsets of size n (i.e., candidate n-itemsets which are potentially large n-itemsets). Assume that items within an itemset are kept in their lexicographic order. Let L_k be a set of large k-itemsets and C_k be a set of candidate k-itemsets. The Apriori algorithm for finding large itemsets is shown in Figure 8.

```
$L_l$ = {large 1-itemsets};
For ($k = 2$; $L_{k-1} \neq \emptyset$; $k$++) do begin
    $C_k$ = apriori_gen($L_{k-1}$); // generate candidate itemsets
    For each transaction $t \in D$ do begin
        $C_t$ = subset($C_k$, $t$); // find candidates that are contained in $t$
        For each candidate $c \in C_t$ do
        $c$.count++;
    End;
    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minimum support}\}$;
End;
Return $\cup_{k>1} L_k$;
```

Figure 8. Apriori Algorithm for Finding Large Itemsets

The apriori_gen takes as argument L_{k-1} and returns C_k in two steps: join step (L_{k-1} is joined with L_{k-1}) and prune step (all the itemsets $c \in C_k$ such that some (k-1)-subset of c is not in L_{k-1}). After all large itemsets are discovered by the Apriori algorithm, the second subproblem of mining association rules (i.e., using the large itemsets to generate desired association rules) is relative straightforward since its solution space is far smaller than that of the first subproblem. Thus, different search strategies are not be discussed further. For interested readers, please refer to (Agrawal and Srikant, 1994) for details.

There have been several new methods for mining association rules (e.g., Brin et al., 1997; Ramaswamy et al. 1998, Zaki et al. 1998). In many applications, taxonomies (i.e., is-a hierarchies) over the items are available. Multiple-level association-rule mining algorithms have been proposed by allowing association rules to include items defined at different abstraction levels of interest in market basket analysis (e.g., Srikant and Agrawal, 1997; Han and Fu, 1999). Since data sets that are used in association rule analysis tend to be very huge, researchers have considered sampling the large databases to generate association rules (e.g., Toivonen, 1996; Zaki et al., 1997). Finally, since it is costly to find association rules in large databases, techniques for incrementally updating and efficiently maintaining association rules with the addition of new data have been proposed (e.g., Cheung et al., 1996; Cheung et al., 1997). In their techniques, the major idea is to reuse the information on the old large itemsets and to integrate the support information on the new large itemsets in order to substantially reduce the size of candidate sets to be re-examined.

4.2 Sequential Patterns

Sequential-pattern analysis is proposed to identify frequent sequential occurrence of items across ordered transactions over time. In contrast to the association rule analysis that is concerned with intra-transaction patterns, the sequential-pattern analysis is concerned with inter-transaction patterns. Understanding customers' long-term purchasing behavior is a typical application of the sequential pattern analysis. On the other hand, analyzing medical diagnosis records of the same patients can help understand long-term medical treatments or patient care processes.

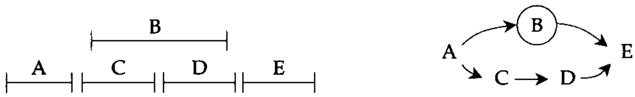
The sequential-pattern mining problem is formally defined as follows (Agrawal and Srikant, 1995). An itemset is a non-empty set of items, and a sequence is an ordered list of itemsets. A sequence $s = <a_1\ a_2\ \ldots a_n>$ is contained in another sequence $t = <b_1\ b_2\ \ldots b_m>$ if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, ..., and $a_n \subseteq b_{i_n}$. In a set of sequences, a sequence s is *maximal* if s is not contained in any other sequences. All the transactions of a customer can together be viewed as a sequence, where each transaction corresponds to an itemset and the list of transactions, ordered by transaction time in ascending order, corresponds to a sequence. Accordingly, the support for an itemset a_i is defined as the fraction of customers who purchased all items of a_i in a single transaction. A customer supports a sequence s if it is contained in the customer sequence for this customer. Likewise, the support for a sequence s is defined as the fraction of total customers who support s. A sequence whose support is no less than the minimum support is called a large sequence. Formally, given a database D of customer transactions, the problem of mining sequential patterns is to find the maximal sequences among all large sequences. Each such maximal sequence represents a sequential pattern.

Because the downward closure property holds for the support measure in the sequential-pattern mining problem as well, the sequential-pattern mining algorithm is similar to that of finding large itemsets in mining association rules. The details of the sequential-pattern mining algorithm (called AprioriAll) and its variant (AprioriSome) can be found in (Agrawal and Srikant, 1995). In their later work, the problem of mining sequential patterns is further generalized (Srikant and Agrawal, 1996). In their extended algorithm (called Generalized Sequential Patterns, GSP), users are allowed to specify a minimum and/or maximum time constraint between adjacent elements in a sequential pattern. Second, the restriction that the items in an element of a sequential pattern must come from the same transaction is relaxed. Instead, GSP allows the items to be present in a set of transactions whose transaction-times are within a user-specified time window. Third, given a user-specified taxonomy on items, GSP allows sequential patterns to include items across all levels of the taxonomy.

4.3 Temporal Patterns

As mentioned, sequential-pattern analysis is concerned with identifying patterns from sequences of transactions. However, in some applications, temporal relationships are beyond sequential ones. For example, a process instance is comprised of a set of activities and the duration of their execution. Based on their execution durations, a set of temporal relationships, including sequential and overlapping, between activities can be constructed for each process instance. Discovery of frequently occurring activities and temporal relationships within a given set of process instances is essential. For example, to provide better patient management, it is essential to discover from clinical care logs frequent clinical pathways knowledge that could be used to reduce practice variations, minimize delays in treatments, and improve resource use (Lin et al., 2001). It is also desirable to discover frequent plan execution patterns that distinguish desired plan executions from undesired ones. Since the discovered knowledge involves activities with temporal relationships, they are referred to as temporal patterns (Wei et al., 2000b).

For a given process instance, two activities are *overlapped* if their execution durations overlap. Otherwise, one activity is *followed* by another if the former finishes before the latter starts. Furthermore, an activity v_i is *directly followed* by another activity v_j if there does not exist any activity v_k such that v_i is followed by v_k and v_k is followed by v_j. To express the temporal relationships between activities in a concise way, a temporal graph is defined. Specifically, a process instance P can be represented as a directed acyclic graph (called a temporal graph) $G=(V,E)$, where V is the set of activities in P and $(v_i,v_j) \in E$ if v_i is directly followed by v_j in P. From a given temporal graph, we can infer whether an activity v_i is followed by another v_j by examining whether there exists a path from v_i to v_j. If there exists no path between v_i and v_j, they are overlapped. Figure 9 shows a sample process instance and its corresponding temporal graph.



a) Process Instance b) Temporal Graph for Process Instance in a)

Figure 9. Examples of Process Instance and Temporal Graph

Given a temporal graph TG, a process instance P supports TG if all followed and overlapped relationships that existed in TG are presented in P. Similar to a sequential pattern, a temporal graph is said to be *frequent* if it is supported by at least $s\%$ of the process instances, where $s\%$ is a user-defined support threshold. Given a set of temporal graphs T, a temporal graph $TG \in T$ is *maximal* if TG is not supported by any other temporal graph in T. Thus, the goal of temporal pattern discovery is to find *frequent* and *maximal* temporal graphs from a set of process instances.

Let G-$\{v\}$ be a subgraph of G where the vertex v is removed from G and transitive edges via the vertex v are reconstructed by connecting each source vertex of incoming edges of v to each destination vertex of outgoing edges of v. The downward closure property implies that a temporal graph G of size n (where n refers to the number of vertices in G) may be frequent only if both G-$\{v_{source}\}$ and G-$\{v_{sink}\}$ are frequent for any pair of source activity v_{source} (i.e., the activity that has no incoming edge in G) and sink activity v_{sink} (i.e., the activity that has no outgoing edge in G). Thus, frequent temporal graphs of size n-1 can be used to construct candidate temporal graphs of size n. The candidate generation process combines two frequent temporal graphs G_1 and G_2 (of size n-1) into one or more candidate temporal graphs (of size n) if there exist a source activity v_{source} in G_1 and a sink activity v_{sink} in G_2 such that G_1-$\{v_{source}\}$= G_2-$\{v_{sink}\}$ and $v_{source} \neq v_{sink}$. Valid temporal relationships between v_{source} and v_{sink} in the candidate temporal graphs need to be determined. Three possible relationships exist but some of them may be invalid: 1) v_{source} and v_{sink} are overlapped, 2) v_{source} is followed by v_{sink}, and 3) v_{sink} is followed by v_{source}. A temporal relationship between v_{source} and v_{sink} is considered valid if all temporal relationships in G_1 and G_2 are preserved. Each valid temporal graph then becomes a candidate temporal graph of size n.

Once a set of candidate temporal graphs of size n are derived from frequent temporal graphs of size n-1, the support of each candidate temporal graph is counted by scanning the set of process instances. Candidate temporal graphs that have a certain user-specified minimum support become frequent temporal graphs of size n. This candidate and frequent temporal graph generation process continues until no more candidate or frequent temporal graphs can be formed. To retain maximal temporal graphs, a pruning mechanism is incorporated. After the set of frequent temporal graphs of size n is generated, all of their subgraphs of size n-1 are derived and removed from the set of frequent temporal graphs of size n-1.

5 Data Visualization

Data visualization takes advantage of human perception as a method for analysis. Data visualization allows decision makers to view complex patterns in the data as visual objects in three dimensions and colors, and supports advanced manipulation capabilities to slice, rotate or zoom the objects to provide varying levels of details of the patterns observed (Shaw et al., 2001). The primary goals of data visualization could be exploration or confirmation. Data visualization for exploration is usually performed during early stages of data analysis when initial hypothesis about the data characteristics is not available. The ideal outcome is a set of hypotheses on the distribution of data along various dimensions as well as any interaction effects that may be visible. Visualization can also be used to identify outliers, clusters, frequencies, relationships, and general patterns in the data. On the other hand, data visualization for confirmation is usually used to confirm hypotheses that are suspected from previous data analyses using other means. Visualizing the hypotheses provides yet another evidence for the confirmation of the hypotheses of interest.

Keim and Kriegel (1996) provided an elaborate analysis of visualization techniques for mining large databases and classified data visualization techniques into pixel-oriented, geometric projection, icon-based, hierarchical, and graph-based. A pixel-oriented technique maps each data value to a colored pixel and presents the data values belonging to each attribute in separate windows. Geometric projection techniques aim at finding "interesting" projections of multidimensional data sets. The class of geometric projection techniques includes principal component analysis, factor analysis, multidimensional scaling, etc. The idea of icon-based techniques is to map each multidimensional data item to an icon (e.g., a face icon in the Chernoff face visualization (Tufte, 1983), a stick figure in the stick figure technique (Pickett and Grinstein, 1988)). In an icon-based technique, two dimensions are mapped to the display dimensions and the remaining dimensions are mapped to the properties of an icon (e.g., the shape of nose, mouth and eyes of a face icon or the angles and/or limb lengths of a stick figure icon). If the data items are relatively dense with respect to the display dimensions, the resulting visualization presents texture patterns that are vary according to the characteristics of the data and are, thus, detectable by preattentive perception. The hierarchical visualization techniques subdivide the *k*-dimensional space and present the subspaces in a hierarchical fashion. For example, the dimensional stacking technique subdivides the *k*-dimensional space into two-dimensional subspaces (LeBlanc et al., 1990). Finally, the basic idea of a graph-based technique is to effectively present a large graph using specific layout algorithms, query languages, and abstraction techniques. Examples of graph-based techniques are Hy+ (Consens and Mendelzon, 1993) and SeeNet (Becker et al., 1995).

6 Text Mining

Unlike the above-mentioned data mining techniques that deal with structured data, text mining is to extract patterns from textual documents. A text mining technique typically involves text parsing and analysis to transform each unstructured document into an appropriate set of features and subsequently applies one or more above-mentioned data mining techniques for extracting patterns from the feature space. Research in text mining has its roots in information retrieval. Text mining has grown from its origin to encompass text categorization, document clustering, term association discovery, routing and filtering, information extraction, document summarization, etc. In this section, we focus on text categorization, document clustering, and term association discovery techniques.

6.1 Text Categorization

Text categorization refers to the assignment of textual documents, on the basis of their contents, to one or more pre-defined categories (Apté et al., 1994). A challenging research issue of text categorization is the development of statistical or inductive learning methods for automatically discovering text categorization patterns, based on a training set of manually categorized documents. In general, automatic learning text categorization patterns encompasses three main phases: feature extraction and selection, document representation, and induction (Apté et al., 1994; Wei and Lee, 2001). The feature extraction and selection as well as document representation are text parsing and analysis tasks, while the induction phase deals with the classification analysis issue described in Section 2.

The feature extraction and selection phase is undertaken to determine a set or sets of features (a universal dictionary or local dictionaries) that will be used for representing individual documents. The universal dictionary is created for all categories, while each local dictionary is created for a particular category. The text portion of the training documents is parsed to produce a list of features (typically consisting of nouns or noun phrases) none of which either belongs to a pre-defined list of stop words or is a number or part of a proper name. After the feature extraction, the feature selection is initiated to reduce the number of unnecessary features, a process that not only improves learning efficiency but also reduces bias in raw data and increases the learning effectiveness (Dumais et al., 1998). Several feature selection methods have been proposed in the literature (Dumais et al., 1998; Lam and Ho, 1998; Lewis and Ringuette, 1994; Ng et al., 1997), including TF (within-document term frequency), TF×IDF (within-document term frequency×inverse document frequency), correlation coefficient, mutual information, and χ^2 metric. The top k features with the highest feature selection metric score are selected as features for representing documents.

In the document representation phase, each individual document is represented in terms of features in the dictionary (universal or local) generated in the previous phase. A document is labeled to indicate its category membership and assigned a value for each feature in the dictionary, where the values can be either boolean (e.g., indicating whether or not the feature appears in the document), or numerical (e.g., frequency of occurrence in the document being processed). Different document representation methods have been proposed (Yang and Chute, 1994), including binary, TF, IDF, and TF×IDF.

The induction phase is designed to automatically discover text categorization patterns that distinguish categories from one another, based on a training set of manually categorized documents. The learning strategies for automatically learning text categorization patterns can essentially be subdivided into the following types: decision tree induction (Weiss et al., 1999); decision rule induction (Apté et al., 1994; Cohen and Singer, 1999); k-nearest neighbor classification (Iwayama and Tokunaga, 1995; Larkey and Croft, 1996; Yang, 1994); neural network (Wiener et al., 1995; Ng et al., 1997); Bayesian networks (Baker and McCallum, 1998; Larkey and Croft, 1996; Lewis and Ringuette, 1994; McCallum and Nigam, 1998); and regression approach (Yang and Chute, 1994). For interested readers, a more detailed summary and empirical comparisons can be found in (Yang and Liu, 1999).

6.2 Document Clustering

Document clustering is to organize a large document collection into groups of documents that are related among them. Similar to text categorization, a document clustering technique generally consists of three main phases: feature extraction and selection, document representation, and clustering.

The feature extraction and selection phase is the same as that in a text categorization technique. Typical feature selection methods for document clustering include TF, TF×IDF, and a hybrid of TF and TF×IDF. The top *k* features with the highest feature selection metric score are selected as features for representing documents. Similarly, the document representation phase is the same as that in a text categorization technique. Commonly adopted document representation methods include binary, TF, and TF×IDF.

The clustering phase is to segment documents into clusters, based on the representative features and their corresponding values of each document. It resembles a typical clustering task. Understandably, each of the clustering approaches (discussed in Section 3) can be applied for clustering documents: partitioning-based (e.g., Larsen and Aone, 1999), hierarchical (e.g., El-Hamdouchi and Willett, 1986; Roussinov and Chen, 1999), and Kohonen neural network approach (e.g., Roussinov and Chen, 1999). Besides adopting basic clustering techniques, new techniques have been developed for clustering documents. For example, Agrawal et al. (1999) developed an interactive document clustering approach that involves iteratively presenting the user with a number of related documents that suggest how a specific cluster might be. The user can discard irrelevant documents from or add additional documents into the cluster. Kim and Lee (2000) developed a semi-supervised document clustering technique that combines a variant of complete-linkage agglomerative hierarchical clustering with the relevance-feedback learning technique.

6.3 Term Association Discovery

Since user queries are usually short and the word mismatch between query terms and documents is common, the information retrieval results are usually unsatisfied (Xu and Croft, 1996). To improve the information retrieval effectiveness, query expansion has been proposed. When expanding a user query, semantically similar and/or statistically associated terms with corresponding weights are added. A widely adopted approach for query expansion approach is to automatically construct term associations from documents (or user queries) and uses the constructed term associations for expanding a user query.

To build term association relationships from documents, most of existing techniques adopt the general term co-occurrence approach for determining weights between terms. Recently, the association-rule mining technique has been applied to discover associations between terms. Unlike the general term co-occurrence approach, the association rule mining technique generates directional term association (Wei et al., 2000c).

7 Conclusion

As organizations become increasingly information oriented and more knowledge conscious, mining patterns, regularities, and knowledge from data become essential to decision support. Knowledge discovery and data mining is a fast expanding field with many new research results and successful applications reported recently. This chapter attempts to provide a reasonably comprehensive review on knowledge discovery and its associated data mining techniques. However, the techniques covered in this chapter are by no means exhaustive. For example, fuzzy-based data mining techniques and web mining are not included. As a promising and expanding field, knowledge discovery and data mining approaches will continue to evolve and new techniques will be devised.

Knowledge discovery and data mining applications should be treated more as a business issue, rather from purely technology perspectives. Organizations need to first identify core business questions and analyze opportunities where organizational data can provide value. Subsequently, business strategies for addressing the targeted business question should be developed. For example, business strategies for the business question of customer retention may include the strategies of product innovation and stabilization of potential churners. The business strategy development usually requires experts from various business functional areas (e.g., marketing, strategic management, etc.). Once a set of business strategies have been devised, each of them is further decomposed into an array of data mining tasks. For example, to stabilize potential churners, two essential data mining tasks include prediction of potential churners and cross-sales. Finally, depending on the data availability and their characteristics, appropriate data mining techniques for each data mining task will then be determined or developed. This top-down process defers the selection of data mining techniques until the target business question to address is fully understood and analyzed.

A data mining application should not be a standalone system. It should be integrated with existing information systems from which data are collected and to which the discovered knowledge can be applied. Moreover, data mining applications may facilitate and foster the reengineering of business processes. As a result, the enhancement or modification of existing information systems may be inevitable. Thus, the provision of information system with high degree of modularization is essential to maximizing the value of data mining applications to organizations.

References

Agrawal, R., R. Bayardo, and R. Srikant, "Athena: Mining-based Interactive Management of Text Databases," *Proceedings of the 6th International Conference on Extending Database Technology*, July 1999, 365-379.

Agrawal, R., T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington DC, 1993, 207-216.

Agrawal, R. and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, 1994, 487-499.

Agrawal R. and R. Srikant, "Mining Sequential Patterns," *Proceedings of the 1995 Conference on Data Engineering*, Taiepi, Taiwan, 1995, 3-14.

Anand, S. S., A. E. Smith, P. W. Hamilton, J. S. Anand, J. G. Hughes, and P. H. Bartels, "An Evaluation of Intelligent Prognostic Systems for Colorectal Cancer," *Artificial Intelligence in Medicine*, Vol. 15, No. 2, 1999, 193-214.

Anderberg, M. R., *Cluster Analysis for Applications*, New York: Academic Press, 1973.

Aoki, N., M. J. Wall, J. Demsar, B. Zupan, T. Granchi, M. A. Schreiber, J. B. Holcomb, M. Byrne, K. R. Liscum, G. Goodwin, J. R. Beck, and K. K. Mattox, "Predictive Model for Survival at the Conclusion of A Damage Control Laparotomy," *The American Journal of Surgery*, Vol. 180, No. 6, December 2000, 540-545.

Apté, C., F. Damerau, and S. Weiss, S., "Automated Learning of Decision Rules for Text Categorization," *ACM Transactions on Information Systems*, Vol. 12, No. 3, 1994, 233-251.

Arya, S., D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching," *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Arlington, VA, 1994, 573-582.

Atlas, L., R. Cole, J. Connor, M. El-Sharkawi, R. J. Marks II, Y. Muthusamy, and E. Barnard, "Performance Comparisons between Backpropagation Networks and Classification Trees on Three Real-World Applications," in Turetzky, D.S. (ed.), *Neural Information Processing Systems (NIPS) 2,* San Mateo, CA: Morgan Kaufmann, 1990, 622-629.

Azuaje, F., W. Dubitzky, P. Lopes, N. Black, K. Adamson, X. Wu, and J. A. White, "Predicting Coronary Disease Risk Based on Short-term RR Interval Measurements: A Neural Network Approach," *Artificial Intelligence in Medicine*, Vol. 15, No. 3, 1999, 275-297.

Baker, L. D. and A. K. McCallum, "Distributional Clustering of Words for Text Classification," *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, August 1998, 96-103.

Becker, R. A., S. G. Eick, and A. R. Wilks, "Visualizing Network Data," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 1, March 1995, 16-28.

Berry, M. J. and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*, New York: John Wiley & Sons, Inc., 1997.

Berson, A. and S. J. Smith, *Data Warehousing, Data Mining, and OLAP*, New York: McGraw-Hill, 1997.

Berson, A., S. Smith, and K. Thearling, *Building Data Mining Applications for CRM*, New York: McGraw-Hill, 2000.

Beyer, K., J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is 'Nearest Neighbor' Meaningful?" *Proceedings of the 7th International Conference on Data Theory (ICDT)*, Jerusalem, Israel, 1999, 217-235.

Blum, A., *Neural Network in C++*, New York: Wiley, 1992.

Bonchi, F., F. Giannotti, C. Gozzi, G. Manco, M. Nanni, D. Pedreschi, C. Renso, and S. Ruggieri, "Web Log Data Warehousing and Mining for Intelligent Web Caching," *Data and Knowledge Engineering*, Vol. 39, No. 2, 2001, 165-189.

Breiman, L., J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees,* Pacific Grove, CA: Wadsworth, 1984.

Brin, S., R. Motwani, J. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Tucson, AZ, 1997, 255-264.

Cabena, P., P. Hadjinian, R. Stadler, J. Verhees and A. Zanasi, *Discovering Data Mining: From Concept to Implementation*, Upper Saddle River, NJ: Prentice Hall, 1997.

Can, F. and E. A. Ozkarahan, "Concepts and Effectiveness of the Cover-Coefficient-Based Clustering Methodology for Text Databases," *ACM Transactions on Database Systems*, Vol. 15, No. 4, 1990, 483-517.

Carter, C. and J. Catlett, "Assessing Credit Card Applications Using Machine Learning," *IEEE Expert*, Fall 1987, 71-79.

Chae, Y. M., S. H. Ho, K. W. Cho, D. H. Lee, and S. H. Ji, "Data Mining Approach to Policy Analysis in A Health Insurance Domain," *International Journal of Medical Informatics,* Vol. 62, No. 2-3, 2001, 103-111.

Chen, M. S., J. Han, and P. S. Yu, "Data Mining: An Overview from a Database Perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, 1996, 866-883.

Cheung, D., J. Han, V. Ng, and C. Y. Wong, "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique," *Proceedings of the International Conference on Data Engineering*, New Orleans, LA, 1996, 106-114.

Cheung, D., S. D. Lee, and B. Kao, "A General Incremental Technique for Maintaining Discovered Association Rules," *Proceedings of the 5th International Conference on Database Systems for Advanced Applications*, Melbourne, Australia, 1997, 185-194.

Clark, P. and T. Niblett, "The CN2 Induction Algorithm," *Machine Learning*, Vol. 3, No. 4, 1989, 261-283.

Cohen, W. W. and Y. Singer, "Context-sensitive Learning Methods for Text Categorization," *ACM Transactions on Information Systems*, 17, 2, 1999, 141-173.

Consens, M. P. and A. O. Mendelzon, "Hy+: A Hygraph-Based Query and Visualization System," *Proceedings of ACM SIGMOD International Conference on Management of Data*, Washington D.C., 1993, 511-516.

Cover, T. M. and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, IT-13, 1, 1967, 21-27.

Davenport, T. H., D. W. DeLong, and M. C. Beers, "Successful Knowledge Management Projects," *Sloan Management Review*, Winter 1998, 43-57.

Davenport, T. H. and L. Prusak, *Working Knowledge: How Organizations Manager What They Know*, Boston, MA: Harvard Business School Press, 1998.

Dey, D., S. Sarkar, and P. De, "Entity Matching in Heterogeneous Databases," *Proceedings of the 31st Hawaii International Conference on System Sciences*, Kona, Hawaii, 1998.

Dobkin, D. and R. J. Lipton, "Multidimensional Search Problems," *SIAM Journal of Computing*, 5, 2, 1976, 181-186.

Dumais, S., J. Platt, D. Heckerman, and M. Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," *Proceedings of the 1998 ACM 7th International Conference on Information and Knowledge Management (CIKM '98)*, Washington D.C., November 1998, 148-155.

El-Hamdouchi, A. and P. Willett, "Hierarchical Document Clustering Using Ward's Method," *Proceedings of ACM Conference on Research and Development in Information Retrieval*, Pisa, Italy, September 1986, 149-156.

Esposito, F., D. Malerba, and G. Semeraro, "A Comparative Analysis of Methods for Pruning Decision Trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 5, 1997, 476-491.

Estivill-Castro, V. and A. T. Murray, "Spatial Clustering for Data Mining with Generic Algorithms," Technical Report FIT-TR-97-10, Queensland University of Technology, Faculty of Information Management, September 1997.

Ezawa, K. J. and S. W. Norton, "Constructing Bayesian Networks to Predict Uncollectible Telecommunications Accounts," *IEEE Expert*, Vol. 11, No. 5, 1996, 45-51.

Feng, L., T. Dillon, and J. Liu, "Inter-transactional Association Rules for Multidimensional Contexts for Prediction and Their Application to Studying Meteorological Data," *Data and Knowledge Engineering*, Vol. 37, No. 1, 2001, 85-115.

Frawley, W., G. Piatetsky-Shapiro, and C. J. Matheus. "Knowledge Discovery in Databases: An Overview," in Piatesky-Shapiro, G. and Frawley, W.J. (eds.), *Knowledge Discovery in Databases*, Cambridge, MA: AAAI/MIT Press, 1991, 1-30.

Fu, L., "Knowledge Discovery Based on Neural Networks," *Communications of the ACM*, 42, 11, 1999, 47-50.

Fu, L. and E. H. Shortliffe. "The Application of Certainty Factors to Neural Computing for Rule Discovery," *IEEE Transactions on Neural Networks*, 11, 3, 2000, 647-657.

Gerritsen, R., "Assessing Loan Risks: A Data Mining Case Study," *IT Professional*, 1, 6, 1999, 16-21.

Han, J. and Y. Fu, "Mining Multiple-Level Association Rules in Large Databases," *IEEE Transactions on Knowledge and Data Engineering*, 11, 5, 1999, 798-805.

Heckerman, D., "Bayesian Networks for Data Mining," *Data Mining and Knowledge Discovery*, 1, 1, 1997, 79-119.

Hui, S. C. and G. Jha, "Data Mining for Customer Service Support," *Information and Management*, 38, 1, 2000, 1-13.

Iwayama, M. and T. Tokunaga, "Cluster-based Text Categorization: A Comparison of Category Search Strategies," *Proceedings of 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, July 1995, 273-281.

John, G. H., P. Miller, and R. Kerber, "Stock Selection Using Rule Induction," *IEEE Expert*, 11, 5, 1996, 52-58.

Kappert, C. B. and S. W. F. Omta, "Neural Networks and Business Modeling–An Application of Neural Modeling Techniques to Prospect Profiling in the Telecommunications Industry," *Proceedings of the 30th Hawaii International Conference on System Sciences*, Maui, Hawaii, 1997, 465-473.

Kass, G. V., "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Applied Statistics*, 29, 1980, 119-127.

Kaufman, L. and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: John Wiley & Sons, Inc., 1990.

Keim, D. A. and H. Kriegel, "Visualization Techniques for Mining Large Databases: A Comparison," *IEEE Transactions on Knowledge and Data Engineering*, 8, 6, 1996, 923-927.

Klemettinen, M., H. Mannila, and H. Toivonen, "Interactive Exploration of Interesting Findings in the Telecommunication Network Alarm Sequence Analyzer (TASA)," *Information and Software Technology*, 41, 9, 1999, 557-567.

Kim, H. and S. Lee, "A Semi-Supervised Document Clustering Technique for Information Organization," *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)*, McLean, VA, November 2000, 30-37.

Kim, S. H. and H. J. Noh, "Predictability of Interest Rates Using Data Mining Tools: A Comparative Analysis of Korea and the US," *Expert Systems with Applications*, 13, 2, 1997, 85-95.

Kohonen, T., *Self-Organization and Associative Memory*, Berlin: Springer, 1989.

Kohonen, T., *Self-Organizing Maps*, Berlin: Springer, 1995.

Kukar, M., I. Kononenko, C. Groselj, K. Kralj, and J. Fettich, "Analysing and Improving the Diagnosis of Ischaemic Heart Disease with Machine Learning," *Artificial Intelligence in Medicine*, 16, 1, 1999, 25-50.

Lam, W. and C. Y. Ho, "Using A Generalized Instance set for Automatic Text Categorization," *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, August 1998, 81-89.

Larkey, L. and W. Croft, "Combining Classifiers in Text Categorization," *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, August 1996, 289-297.

Larsen, B. and C. Aone, "Fast and Effective Text Mining Using Linear-time Document Clustering," *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, August 1999, 16-22.

LeBlanc, J., M. O. Ward, and N. Wittels, "Exploring N-Dimensional Databases," *Proceedings of Visualization*, San Francisco, CA, 1990, 230-237.

Lee, C. H., Y. H. Kim, and P. K. Rhee, "Web Personalization Expert with Combining Collaborative Filtering and Association Rule Mining Technique," *Expert Systems with Applications*, 21, 3, 2001, 131-137.

Lee, H. Y. and H. L. Ong, "Visualization Support for Data Mining," *IEEE Expert*, 11, 5, 1996, 69-75.

Leu, S. S., C. N. Chen, and S. L. Chang, "Data Mining for Tunnel Support Stability: Neural Network Approach," *Automation in Construction*, 10, 4, 2001, 429-441.

Leung, M. T., A. S. Chen, and H. Daouk, "Forecasting Exchange Rates Using General Regression Neural Networks," *Computers and Operations Research*, 27, 11-12, 2000, 1093-1110.

Lewis, D. and M. Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization," *Proceedings of Symposium on Document Analysis and Information Retrieval*, 1994.

Lin, F., S. Chou, S. Pan, and Y. Chen, "Mining Time Dependency Patterns in Clinical Pathways," *International Journal of Medical Informatics*, 62, 1, 2001, 11-25.

Lin, F. Y. and S. McClean, "A Data Mining Approach to the Prediction of Corporate Failure," *Knowledge-Based Systems*, 14, 3-4, 2001, 189-195.

Luchetta, A., S. Manetti, and F. Francini, "Forecast: A Neural System for Diagnosis and Control of Highway Surfaces," *IEEE Intelligent Systems*, 13, 3, 1998, 20-26.

Mannila, H., H. Toivonen, and A. I. Verkamo, "Discovering Frequent Episodes in Sequences," *Proceedings of First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, Montreal, Canada, August 1995, 210–215.

Mannila, H. and H. Toivonen, "Discovering Generalized Episodes Using Minimal Occurrences," *Proceedings of Second International Conference on KnowledgeDiscovery and Data Mining*, Portland, Oregon, August 2-4, 1996.

Marble, R. P. and J. C. Healy, "A Neural Network Approach to the Diagnosis of Morbidity Outcomes in Trauma Care," *Artificial Intelligence in Medicine*, 15, 3, 1999, 299-307.

McCallum, A. K. and K. Nigam, "A Comparison of Event Models for Naïve Bayes Text Classification," *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

Mingers, J., "An Empirical Comparison of Selection Measures for Decision-Tree Induction," *Machine Learning*, 3, 1989a, 319-341.

Mingers, J., "An Empirical Comparison of Pruning Methods for Decision Tree Induction," *Machine Learning*, 4, 2, 1989b, 227-243.

Murthy, S. K., S. Kasif, and S. Salzberg, "A System for Induction of Oblique Decision Trees," *Journal of Artificial Intelligence Research*, 2, 1994, 1-32.

Ng, H. T., W. B. Goh, and K. L. Low, "Feature Selection, Perceptron Learning, and A Usability Case Study for Text Categorization," *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97)*, Philadelphia, PA, July 1997, 67-73.

Ng, R. and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," *Proceedings of International Conference on Very Large Data Bases*, Santiago, Chile, Sept. 1994, 144–155.

Niblett, T. and I. Bratko, "Learning Decision Rules in Noisy Domains," *Research and Development in Expert Systems III: Proceedings of the 6th Technical Conference of the British Computer Society Specialist Group on Expert Systems*, Brignton, December 1986, 25-34.

Pickett, R. M. and G. G. Grinstein, "Iconographics Displays for Visualizing Multidimensional Data," *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, 1988, 514-519.

Quinlan, J. R., "Induction of Decision Trees," *Machine Learning*, 1, 1, 1986, 81-106.

Quinlan, J. R., *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann, 1993.

Ramaswamy, S., S. Mahajan, and A. Silberschatz, "On the Discovery of Interesting Patterns in Association Rules," *Proceedings of the 24th International Conference on Very Large Data Bases*, 1998.

Ronco, A. L., "Use of Artificial Neural Networks in Modeling Associations of Discriminant Factors: Towards An Intelligent Selective Breast Cancer Screening," *Artificial Intelligence in Medicine*, 16, 3, 1999, 299-309.

Roussinov, D. and H. Chen, "Document Clustering for Electronic Meetings: An Experimental Comparison of Two Techniques," *Decision Support Systems*, 27, 1-2, 1999, 67-79.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in Rumelhart, D.E. and McClelland J.L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, Cambridge, MA: MIT Press, 1986, 318-362.

Shaw, M. J., C. Subramaniam, G. W. Tan, and M. E. Welge, "Knowledge Management and Data Mining for Marketing," *Decision Support Systems*, 31, 1, 2001, 127-137.

Song, H. S., J. K. Kim, and S. H. Kim, "Mining the Change of Customer Behavior in An Internet Shopping Mall," *Expert Systems with Applications*, 21, 3, 2001, 157-168.

Srikant, R. and R. Agrawal, "Mining Generalized Association Rules," *Future Generation Computer Systems*, 13, 2-3, 1997, 161-180.

Srikant, R. and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," *Proceedings of the 5th International Conference on Extending Database Technology (EDBT)*, Avignon, France, March 1996, 3-17.

Tickle, A. B., R. Andrews, M. Golea, and J. Diederich, "The Truth Will Come to Light: Directions and Challenges in Extracting the Knowledge Embedded Within Trained Artificial Neural Networks," *IEEE Transactions on Neural Networks*, 9, 6, 1998, 1057-1068.

Toivonen, H., "Sampling Large Databases for Association Rules," *Proceedings of the 22nd International Conference on Very Large Data Bases*, Bombay, India, 1996, 134-145.

Tufte, E. R., *The Visual Display of Quantitative Information*, Cheshire, CT: Graphics Press, 1983.

Voorhees, E. M., "Implementing Agglomerative Hierarchical Clustering Algorithms for Use in Document Retrieval," *Information Processing and Management*, 22, 1986, 465-476.

Walczak, S. and J. E. Scharf, "Reducing Surgical Patient Costs Through Use of An Artificial Neural Network to Predict Transfusion Requirements," *Decision Support Systems*, 30, 2, 2000, 125-138.

Walley, W. J. and M. A. O'Connor, "Unsupervised Pattern Recognition for the Interpretation of Ecological Data," *Ecological Modelling*, 146, 1-3, 2001, 219-230.

Wei, C., Y. H. Lee and C. M. Hsu, "Empirical Comparison of Fast Clustering Algorithms for Large Data Sets," *Proceedings of 33rd Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2000a.

Wei, C., S. Y. Hwang, and W. S. Yang, "Mining Frequent Temporal Patterns in Process Databases," *Proceedings of 10th Workshop on Information Technologies and Systems (WITS 2000)*, Brisbane, Australia, December 2000b, 175-180.

Wei, C. and Y. H. Lee, "Event Detection for Supporting Environmental Scanning: An Information Extraction-based Approach," *Proceedings of 5th Pacific Asia Conference on Information Systems (PACIS)*, Seoul, Korea, June 2001.

Wei, J., S. Bressan, and B. C. Ooi, "Mining Term Association Rules for Automatic Global Query Expansion: Methodology and Preliminary Results," *Proceedings of the First International Conference on Web Information Systems Engineering*, 2000c, 366-373.

Weiss, S. M., C. Apte, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp, "Maximizing Text-Mining Performance," *IEEE Intelligent Systems*, 14, 4, 1999, 63-69.

West, D., "Neural Network Credit Scoring Models," *Computers and Operations Research*, 27, 11-12, 2000, 1131-1152.

Wiener, W., J. O. Pedersen, and A. S. Weigend, "A Neural Network Approach to Topic Spotting," *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR '95)*, Las Vegas, NV, 1995, 317-332.

Wright, W., "Business Visualization Applications," *IEEE Computer Graphics and Applications*, 17, 4, 1997, 66-70.

Xu, J. and W. B. Croft, "Query Expansion Using Local and Global Document Analysis," *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996, 4-11.

Yang, Y., "Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval," *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 1994, 13-22.

Yang, Y., J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu, "Learning Approaches for Detecting and Tracking News Events," *IEEE Intelligent Systems*, 14, 4, 1999, 32-43.

Yang, Y. and C. G. Chute, "An Example-Based Mapping Method for Text Categorization and Retrieval," *ACM Transactions on Information Systems*, 12, 3, 1994, 252-277.

Yang, Y. and X. Liu, "A Re-examination of Text Categorization Methods," *Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, August 1999, 42-49.

Yang, Y., T. Pierce, and J. G. Carbonell, "A Study on Retrospective and On-line Event Detection," *Proceedings of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, 28-36.

Zaki, M., S. Parthasarathy, W. Li, and M. Ogihara, "Evaluation of Sampling for Data Mining of Association Rules," *Proceedings of the 7th Workshop on Research Issues in Data Engineering*, Birmingham, UK, 1997, 42-50.

Zaki, M., S. Parthasarathy, M. Ogihara, and W. Li, "New Algorithms for Fast Discovery of Association Rules," Technical Report 651, Computer Science Department, University of Rochester, 1998.

PART VI

Outcomes of Knowledge Management