set) to mine all outliers by scanning the data set three times. First, a sample, S, is created of the given data set, D, using sampling by replacement. Each object in S is considered the centroid of a partition. The objects in D are assigned to the partitions based on distance. The preceding steps are completed in one scan of D. Candidate outliers are identified in a second scan of D. After a third scan, all $DB(r,\pi)$-outliers have been found.

12.4.3 Density-Based Outlier Detection

Distance-based outliers, such as $DB(r,\pi)$-outliers, are just one type of outlier. Specifically, distance-based outlier detection takes a global view of the data set. Such outliers can be regarded as "global outliers" for two reasons:

- A $DB(r,\pi)$-outlier, for example, is far (as quantified by parameter r) from at least $(1-\pi)\times 100\%$ of the objects in the data set. In other words, an outlier as such is remote from the majority of the data.
- To detect distance-based outliers, we need two global parameters, r and π, which are applied to every outlier object.

Many real-world data sets demonstrate a more complex structure, where objects may be considered outliers with respect to their local neighborhoods, rather than with respect to the global data distribution. Let's look at an example.

Example 12.14 **Local proximity-based outliers.** Consider the data points in Figure 12.8. There are two clusters: C_1 is dense, and C_2 is sparse. Object $\boldsymbol{o}_3$ can be detected as a distance-based outlier because it is far from the majority of the data set.

Now, let's consider objects $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$. Are they outliers? On the one hand, the distance from $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ to the objects in the dense cluster, C_1, is smaller than the average distance between an object in cluster C_2 and its nearest neighbor. Thus, $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ are not distance-based outliers. In fact, if we were to categorize $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ as $DB(r,\pi)$-outliers, we would have to classify all the objects in clusters C_2 as $DB(r,\pi)$-outliers.

On the other hand, $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ can be identified as outliers when they are considered locally with respect to cluster C_1 because $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ deviate significantly from the objects in C_1. Moreover, $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ are also far from the objects in C_2.

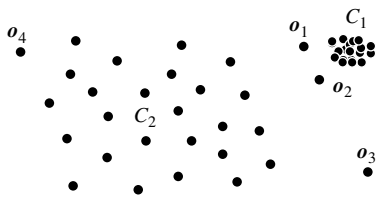

Figure 12.8 Global outliers and local outliers.

To summarize, distance-based outlier detection methods cannot capture local outliers like $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$. Note that the distance between object $\boldsymbol{o}_4$ and its nearest neighbors is much greater than the distance between $\boldsymbol{o}_1$ and its nearest neighbors. However, because $\boldsymbol{o}_4$ is local to cluster C_2 (which is sparse), $\boldsymbol{o}_4$ is not considered a local outlier. ■

"*How can we formulate the local outliers as illustrated in Example 12.14?*" The critical idea here is that we need to compare the density around an object with the density around its local neighbors. The basic assumption of density-based outlier detection methods is that the density around a nonoutlier object is similar to the density around its neighbors, while the density around an outlier object is significantly different from the density around its neighbors.

Based on the preceding, density-based outlier detection methods use the relative density of an object against its neighbors to indicate the degree to which an object is an outlier.

Now, let's consider how to measure the *relative density* of an object, $\boldsymbol{o}$, given a set of objects, D. The *k-distance* of $\boldsymbol{o}$, denoted by $dist_k(\boldsymbol{o})$, is the distance, $dist(\boldsymbol{o}, \boldsymbol{p})$, between $\boldsymbol{o}$ and another object, $\boldsymbol{p} \in D$, such that

- There are at least k objects $\boldsymbol{o'} \in D-\{\boldsymbol{o}\}$ such that $dist(\boldsymbol{o}, \boldsymbol{o'}) \leq dist(\boldsymbol{o}, \boldsymbol{p})$.
- There are at most $k-1$ objects $\boldsymbol{o''} \in D-\{\boldsymbol{o}\}$ such that $dist(\boldsymbol{o}, \boldsymbol{o''}) < dist(\boldsymbol{o}, \boldsymbol{p})$.

In other words, $dist_k(\boldsymbol{o})$ is the distance between $\boldsymbol{o}$ and its k-nearest neighbor. Consequently, the *k-distance neighborhood* of $\boldsymbol{o}$ contains all objects of which the distance to $\boldsymbol{o}$ is not greater than $dist_k(\boldsymbol{o})$, the k-distance of $\boldsymbol{o}$, denoted by

$$N_k(\boldsymbol{o}) = \{\boldsymbol{o'} | \boldsymbol{o'} \in D, dist(\boldsymbol{o}, \boldsymbol{o'}) \leq dist_k(\boldsymbol{o})\}. \tag{12.11}$$

Note that $N_k(\boldsymbol{o})$ may contain more than k objects because multiple objects may each be the same distance away from $\boldsymbol{o}$.

We can use the average distance from the objects in $N_k(\boldsymbol{o})$ to $\boldsymbol{o}$ as the measure of the local density of $\boldsymbol{o}$. However, such a straightforward measure has a problem: If $\boldsymbol{o}$ has very close neighbors $\boldsymbol{o'}$ such that $dist(\boldsymbol{o}, \boldsymbol{o'})$ is very small, the statistical fluctuations of the distance measure can be undesirably high. To overcome this problem, we can switch to the following reachability distance measure by adding a smoothing effect.

For two objects, $\boldsymbol{o}$ and $\boldsymbol{o'}$, the *reachability distance* from $\boldsymbol{o'}$ to $\boldsymbol{o}$ is $dist(\boldsymbol{o} \leftarrow \boldsymbol{o'})$ if $dist(\boldsymbol{o}, \boldsymbol{o'}) > dist_k(\boldsymbol{o})$, and $dist_k(\boldsymbol{o})$ otherwise. That is,

$$reachdist_k(\boldsymbol{o} \leftarrow \boldsymbol{o'}) = \max\{dist_k(\boldsymbol{o}), dist(\boldsymbol{o}, \boldsymbol{o'})\}. \tag{12.12}$$

Here, k is a user-specified parameter that controls the smoothing effect. Essentially, k specifies the minimum neighborhood to be examined to determine the local density of an object. Importantly, reachability distance is not symmetric, that is, in general, $reachdist_k(\boldsymbol{o} \leftarrow \boldsymbol{o'}) \neq reachdist_k(\boldsymbol{o'} \leftarrow \boldsymbol{o})$.

Now, we can define the *local reachability density* of an object, $\boldsymbol{o}$, as

$$lrd_k(\boldsymbol{o}) = \frac{\|N_k(\boldsymbol{o})\|}{\sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} reachdist_k(\boldsymbol{o}' \leftarrow \boldsymbol{o})}. \tag{12.13}$$

There is a critical difference between the density measure here for outlier detection and that in density-based clustering (Section 12.5). In density-based clustering, to determine whether an object can be considered a core object in a density-based cluster, we use two parameters: a radius parameter, r, to specify the range of the neighborhood, and the minimum number of points in the r-neighborhood. Both parameters are global and are applied to every object. In contrast, as motivated by the observation that relative density is the key to finding local outliers, we use the parameter k to quantify the neighborhood and do not need to specify the minimum number of objects in the neighborhood as a requirement of density. We instead calculate the local reachability density for an object and compare it with that of its neighbors to quantify the degree to which the object is considered an outlier.

Specifically, we define the *local outlier factor* of an object $\boldsymbol{o}$ as

$$LOF_k(\boldsymbol{o}) = \frac{\sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} \frac{lrd_k(\boldsymbol{o}')}{lrd_k(\boldsymbol{o})}}{\|N_k(\boldsymbol{o})\|} = \sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} lrd_k(\boldsymbol{o}') \cdot \sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} reachdist_k(\boldsymbol{o}' \leftarrow \boldsymbol{o}). \tag{12.14}$$

In other words, the local outlier factor is the average of the ratio of the local reachability density of $\boldsymbol{o}$ and those of $\boldsymbol{o}$'s k-nearest neighbors. The lower the local reachability density of $\boldsymbol{o}$ (i.e., the smaller the item $\sum_{\boldsymbol{o}' \in N_k(\boldsymbol{o})} reachdist_k(\boldsymbol{o}' \leftarrow \boldsymbol{o})$) and the higher the local reachability densities of the k-nearest neighbors of $\boldsymbol{o}$, the higher the LOF value is. This exactly captures a local outlier of which the local density is relatively low compared to the local densities of its k-nearest neighbors.

The local outlier factor has some nice properties. First, for an object deep within a consistent cluster, such as the points in the center of cluster C_2 in Figure 12.8, the local outlier factor is close to 1. This property ensures that objects inside clusters, no matter whether the cluster is dense or sparse, will not be mislabeled as outliers.

Second, for an object $\boldsymbol{o}$, the meaning of $LOF(\boldsymbol{o})$ is easy to understand. Consider the objects in Figure 12.9, for example. For object $\boldsymbol{o}$, let

$$direct_{min}(\boldsymbol{o}) = \min\{reachdist_k(\boldsymbol{o}' \leftarrow \boldsymbol{o}) | \boldsymbol{o}' \in N_k(\boldsymbol{o})\} \tag{12.15}$$

be the minimum reachability distance from $\boldsymbol{o}$ to its k-nearest neighbors. Similarly, we can define

$$direct_{max}(\boldsymbol{o}) = \max\{reachdist_k(\boldsymbol{o}' \leftarrow \boldsymbol{o}) | \boldsymbol{o}' \in N_k(\boldsymbol{o})\}. \tag{12.16}$$

We also consider the neighbors of $\boldsymbol{o}$'s k-nearest neighbors. Let

$$indirect_{min}(\boldsymbol{o}) = \min\{reachdist_k(\boldsymbol{o}'' \leftarrow \boldsymbol{o}') | \boldsymbol{o}' \in N_k(\boldsymbol{o}) \text{ and } \boldsymbol{o}'' \in N_k(\boldsymbol{o}')\} \tag{12.17}$$

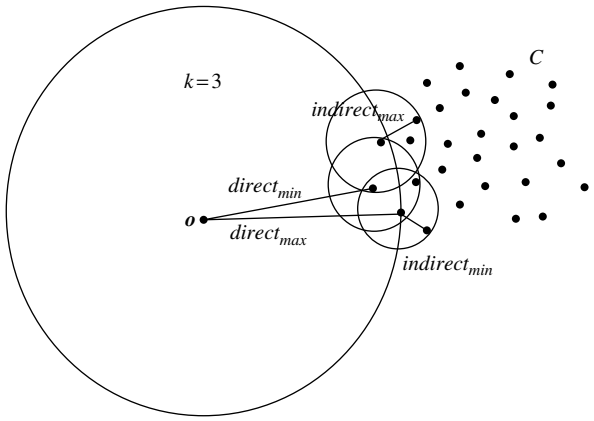

Figure 12.9 A property of $LOF(\boldsymbol{o})$.

and

$$indirect_{max}(\boldsymbol{o}) = \max\{reachdist_k(\boldsymbol{o''} \leftarrow \boldsymbol{o'}) | \boldsymbol{o'} \in N_k(\boldsymbol{o}) \text{ and } \boldsymbol{o''} \in N_k(\boldsymbol{o'})\}. \quad (12.18)$$

Then, it can be shown that $LOF(\boldsymbol{o})$ is bounded as

$$\frac{direct_{min}(\boldsymbol{o})}{indirect_{max}(\boldsymbol{o})} \leq LOF(\boldsymbol{o}) \leq \frac{direct_{max}(\boldsymbol{o})}{indirect_{min}(\boldsymbol{o})}. \quad (12.19)$$

This result clearly shows that LOF captures the relative density of an object.

12.5 Clustering-Based Approaches

The notion of outliers is highly related to that of clusters. Clustering-based approaches detect outliers by examining the relationship between objects and clusters. Intuitively, an outlier is an object that belongs to a small and remote cluster, or does not belong to any cluster.

This leads to three general approaches to clustering-based outlier detection. Consider an object.

- Does the object belong to any cluster? If not, then it is identified as an outlier.
- Is there a large distance between the object and the cluster to which it is closest? If yes, it is an outlier.
- Is the object part of a small or sparse cluster? If yes, then all the objects in that cluster are outliers.

Let's look at examples of each of these approaches.

Example 12.15 **Detecting outliers as objects that do not belong to any cluster.** Gregarious animals (e.g., goats and deer) live and move in flocks. Using outlier detection, we can identify outliers as animals that are not part of a flock. Such animals may be either lost or wounded.

In Figure 12.10, each point represents an animal living in a group. Using a density-based clustering method, such as DBSCAN, we note that the black points belong to clusters. The white point, ***a***, does not belong to any cluster, and thus is declared an outlier. ■

The second approach to clustering-based outlier detection considers the distance between an object and the cluster to which it is closest. If the distance is large, then the object is likely an outlier with respect to the cluster. Thus, this approach detects individual outliers with respect to clusters.

Example 12.16 **Clustering-based outlier detection using distance to the closest cluster.** Using the *k*-means clustering method, we can partition the data points shown in Figure 12.11 into three clusters, as shown using different symbols. The center of each cluster is marked with a +.

For each object, ***o***, we can assign an outlier score to the object according to the distance between the object and the center that is closest to the object. Suppose the closest center to ***o*** is $\boldsymbol{c}_o$; then the distance between ***o*** and $\boldsymbol{c}_o$ is $dist(\boldsymbol{o}, \boldsymbol{c}_o)$, and the average

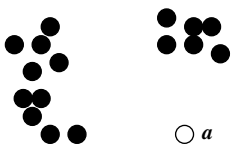

Figure 12.10 Object ***a*** is an outlier because it does not belong to any cluster.

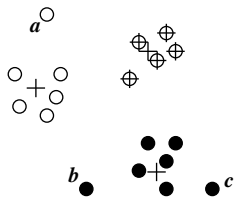

Figure 12.11 Outliers ($\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$) are far from the clusters to which they are closest (with respect to the cluster centers).

distance between $\boldsymbol{c}_o$ and the objects assigned to $\boldsymbol{o}$ is $l_{\boldsymbol{c}_o}$. The ratio $\frac{dist(\boldsymbol{o},\boldsymbol{c}_o)}{l_{\boldsymbol{c}_o}}$ measures how $dist(\boldsymbol{o}, \boldsymbol{c}_o)$ stands out from the average. The larger the ratio, the farther away $\boldsymbol{o}$ is relative from the center, and the more likely $\boldsymbol{o}$ is an outlier. In Figure 12.11, points $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are relatively far away from their corresponding centers, and thus are suspected of being outliers. ■

This approach can also be used for intrusion detection, as described in Example 12.17.

Example 12.17 **Intrusion detection by clustering-based outlier detection.** A bootstrap method was developed to detect intrusions in TCP connection data by considering the similarity between data points and the clusters in a training data set. The method consists of three steps.

1. A training data set is used to find patterns of normal data. Specifically, the TCP connection data are segmented according to, say, dates. Frequent itemsets are found in each segment. The frequent itemsets that are in a majority of the segments are considered patterns of normal data and are referred to as "base connections."
2. Connections in the training data that contain base connections are treated as attack-free. Such connections are clustered into groups.
3. The data points in the original data set are compared with the clusters mined in step 2. Any point that is deemed an outlier with respect to the clusters is declared as a possible attack.

■

Note that each of the approaches we have seen so far detects only individual objects as outliers because they compare objects one at a time against clusters in the data set. However, in a large data set, some outliers may be similar and form a small cluster. In intrusion detection, for example, hackers who use similar tactics to attack a system may form a cluster. The approaches discussed so far may be deceived by such outliers.

To overcome this problem, a third approach to cluster-based outlier detection identifies small or sparse clusters and declares the objects in those clusters to be outliers as well. An example of this approach is the *FindCBLOF* algorithm, which works as follows.

1. Find clusters in a data set, and sort them according to decreasing size. The algorithm assumes that most of the data points are not outliers. It uses a parameter $\alpha\,(0 \leq \alpha \leq 1)$ to distinguish large from small clusters. Any cluster that contains at least a percentage α (e.g., $\alpha = 90\%$) of the data set is considered a "large cluster." The remaining clusters are referred to as "small clusters."
2. To each data point, assign a *cluster-based local outlier factor* (CBLOF). For a point belonging to a large cluster, its CBLOF is the product of the cluster's size and the similarity between the point and the cluster. For a point belonging to a small cluster, its CBLOF is calculated as the product of the size of the small cluster and the similarity between the point and the closest large cluster.

CBLOF defines the similarity between a point and a cluster in a statistical way that represents the probability that the point belongs to the cluster. The larger the value, the more similar the point and the cluster are. The CBLOF score can detect outlier points that are far from any clusters. In addition, small clusters that are far from any large cluster are considered to consist of outliers. The points with the lowest CBLOF scores are suspected outliers.

Example 12.18 **Detecting outliers in small clusters.** The data points in Figure 12.12 form three clusters: large clusters, C_1 and C_2, and a small cluster, C_3. Object ***o*** does not belong to any cluster.

Using CBLOF, *FindCBLOF* can identify ***o*** as well as the points in cluster C_3 as outliers. For ***o***, the closest large cluster is C_1. The CBLOF is simply the similarity between ***o*** and C_1, which is small. For the points in C_3, the closest large cluster is C_2. Although there are three points in cluster C_3, the similarity between those points and cluster C_2 is low, and $|C_3| = 3$ is small; thus, the CBLOF scores of points in C_3 are small. ■

Clustering-based approaches may incur high computational costs if they have to find clusters before detecting outliers. Several techniques have been developed for improved efficiency. For example, **fixed-width clustering** is a linear-time technique that is used in some outlier detection methods. The idea is simple yet efficient. A point is assigned to a cluster if the center of the cluster is within a predefined distance threshold from the point. If a point cannot be assigned to any existing cluster, a new cluster is created. The distance threshold may be learned from the training data under certain conditions.

Clustering-based outlier detection methods have the following advantages. First, they can detect outliers without requiring any labeled data, that is, in an unsupervised way. They work for many data types. Clusters can be regarded as summaries of the data. Once the clusters are obtained, clustering-based methods need only compare any object against the clusters to determine whether the object is an outlier. This process is typically fast because the number of clusters is usually small compared to the total number of objects.

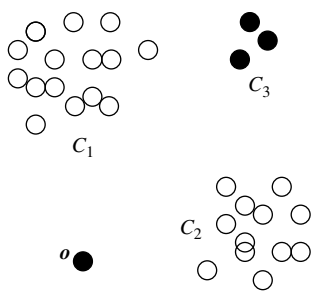

Figure 12.12 Outliers in small clusters.

A weakness of clustering-based outlier detection is its effectiveness, which depends highly on the clustering method used. Such methods may not be optimized for outlier detection. Clustering methods are often costly for large data sets, which can serve as a bottleneck.

12.6 Classification-Based Approaches

Outlier detection can be treated as a classification problem if a training data set with class labels is available. The general idea of classification-based outlier detection methods is to train a classification model that can distinguish normal data from outliers.

Consider a training set that contains samples labeled as "normal" and others labeled as "outlier." A classifier can then be constructed based on the training set. Any classification method can be used (Chapters 8 and 9). This kind of brute-force approach, however, does not work well for outlier detection because the training set is typically heavily biased. That is, the number of normal samples likely far exceeds the number of outlier samples. This imbalance, where the number of outlier samples may be insufficient, can prevent us from building an accurate classifier. Consider intrusion detection in a system, for example. Because most system accesses are normal, it is easy to obtain a good representation of the normal events. However, it is infeasible to enumerate all potential intrusions, as new and unexpected attempts occur from time to time. Hence, we are left with an insufficient representation of the outlier (or intrusion) samples.

To overcome this challenge, classification-based outlier detection methods often use a *one-class model*. That is, a classifier is built to describe only the normal class. Any samples that do not belong to the normal class are regarded as outliers.

Example 12.19 Outlier detection using a one-class model. Consider the training set shown in Figure 12.13, where white points are samples labeled as "normal" and black points are samples labeled as "outlier." To build a model for outlier detection, we can learn the decision boundary of the normal class using classification methods such as SVM (Chapter 9), as illustrated. Given a new object, if the object is within the decision boundary of the normal class, it is treated as a normal case. If the object is outside the decision boundary, it is declared an outlier.

An advantage of using only the model of the normal class to detect outliers is that the model can detect new outliers that may not appear close to any outlier objects in the training set. This occurs as long as such new outliers fall outside the decision boundary of the normal class. ■

The idea of using the decision boundary of the normal class can be extended to handle situations where the normal objects may belong to multiple classes such as in fuzzy clustering (Chapter 11). For example, *AllElectronics* accepts returned items. Customers can return items for a number of reasons (corresponding to class categories) such as "product design defects" and "product damaged during shipment." Each such

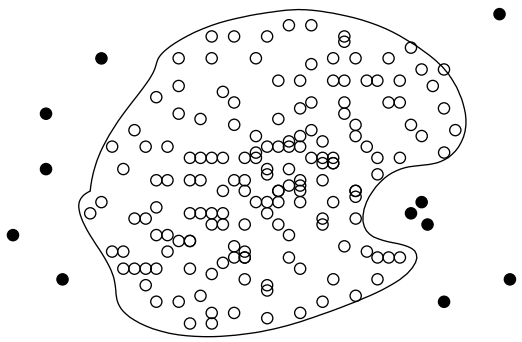

Figure 12.13 Learning a model for the normal class.

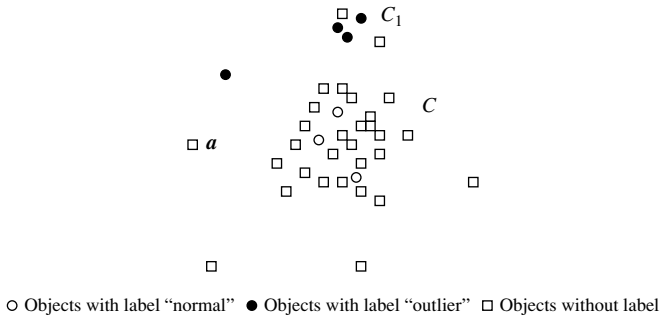

Figure 12.14 Detecting outliers by semi-supervised learning.

class is regarded as normal. To detect outlier cases, *AllElectronics* can learn a model for each normal class. To determine whether a case is an outlier, we can run each model on the case. If the case does not fit any of the models, then it is declared an outlier.

Classification-based methods and clustering-based methods can be combined to detect outliers in a semi-supervised learning way.

Example 12.20 **Outlier detection by semi-supervised learning.** Consider Figure 12.14, where objects are labeled as either "normal" or "outlier," or have no label at all. Using a clustering-based approach, we find a large cluster, C, and a small cluster, C_1. Because some objects in C carry the label "normal," we can treat all objects in this cluster (including those without labels) as normal objects. We use the one-class model of this cluster to identify normal objects in outlier detection. Similarly, because some objects in cluster C_1 carry the label "outlier," we declare all objects in C_1 as outliers. Any object that does not fall into the model for C (e.g., ***a***) is considered an outlier as well. ■

Classification-based methods can incorporate human domain knowledge into the detection process by learning from the labeled samples. Once the classification model is constructed, the outlier detection process is fast. It only needs to compare the objects to be examined against the model learned from the training data. The quality of classification-based methods heavily depends on the availability and quality of the training set. In many applications, it is difficult to obtain representative and high-quality training data, which limits the applicability of classification-based methods.

12.7 Mining Contextual and Collective Outliers

An object in a given data set is a **contextual outlier** (or *conditional outlier*) if it deviates significantly with respect to a specific context of the object (Section 12.1). The context is defined using **contextual attributes**. These depend heavily on the application, and are often provided by users as part of the contextual outlier detection task. Contextual attributes can include spatial attributes, time, network locations, and sophisticated structured attributes. In addition, **behavioral attributes** define characteristics of the object, and are used to evaluate whether the object is an outlier in the context to which it belongs.

Example 12.21 **Contextual outliers.** To determine whether the temperature of a location is exceptional (i.e., an outlier), the attributes specifying information about the location can serve as contextual attributes. These attributes may be spatial attributes (e.g., longitude and latitude) or location attributes in a graph or network. The attribute *time* can also be used. In customer-relationship management, whether a customer is an outlier may depend on other customers with similar profiles. Here, the attributes defining customer profiles provide the context for outlier detection. ■

In comparison to outlier detection in general, identifying contextual outliers requires analyzing the corresponding contextual information. Contextual outlier detection methods can be divided into two categories according to whether the contexts can be clearly identified.

12.7.1 Transforming Contextual Outlier Detection to Conventional Outlier Detection

This category of methods is for situations where the contexts can be clearly identified. The idea is to transform the contextual outlier detection problem into a typical outlier detection problem. Specifically, for a given data object, we can evaluate whether the object is an outlier in two steps. In the first step, we identify the context of the object using the contextual attributes. In the second step, we calculate the outlier score for the object in the context using a conventional outlier detection method.

Example 12.22 **Contextual outlier detection when the context can be clearly identified.** In customer-relationship management, we can detect outlier customers in the context of customer groups. Suppose *AllElectronics* maintains customer information on four attributes, namely *age_group* (i.e., under 25, 25-45, 45-65, and over 65), *postal_code*, *number_of_transactions_per_year*, and *annual_total_transaction_amount*. The attributes *age_group* and *postal_code* serve as contextual attributes, and the attributes *number_of_transactions_per_year* and *annual_total_transaction_amount* are behavioral attributes. ■

To detect contextual outliers in this setting, for a customer, $\boldsymbol{c}$, we can first locate the context of $\boldsymbol{c}$ using the attributes *age_group* and *postal_code*. We can then compare $\boldsymbol{c}$ with the other customers in the same group, and use a conventional outlier detection method, such as some of the ones discussed earlier, to determine whether $\boldsymbol{c}$ is an outlier.

Contexts may be specified at different levels of granularity. Suppose *AllElectronics* maintains customer information at a more detailed level for the attributes *age*, *postal_code*, *number_of_transactions_per_year*, and *annual_total_transaction_amount*. We can still group customers on *age* and *postal_code*, and then mine outliers in each group. What if the number of customers falling into a group is very small or even zero? For a customer, $\boldsymbol{c}$, if the corresponding context contains very few or even no other customers, the evaluation of whether $\boldsymbol{c}$ is an outlier using the exact context is unreliable or even impossible.

To overcome this challenge, we can assume that customers of similar age and who live within the same area should have similar normal behavior. This assumption can help to generalize contexts and makes for more effective outlier detection. For example, using a set of training data, we may learn a mixture model, U, of the data on the contextual attributes, and another mixture model, V, of the data on the behavior attributes. A mapping $p(V_i|U_j)$ is also learned to capture the probability that a data object $\boldsymbol{o}$ belonging to cluster U_j on the contextual attributes is generated by cluster V_i on the behavior attributes. The outlier score can then be calculated as

$$S(\boldsymbol{o}) = \sum_{U_j} p(\boldsymbol{o} \in U_j) \sum_{V_i} p(\boldsymbol{o} \in V_i) p(V_i|U_j). \tag{12.20}$$

Thus, the contextual outlier problem is transformed into outlier detection using mixture models.

12.7.2 Modeling Normal Behavior with Respect to Contexts

In some applications, it is inconvenient or infeasible to clearly partition the data into contexts. For example, consider the situation where the online store of *AllElectronics* records customer browsing behavior in a search log. For each customer, the data log contains the sequence of products searched for and browsed by the customer. *AllElectronics* is interested in contextual outlier behavior, such as if a customer suddenly purchased a product that is unrelated to those she recently browsed. However, in this application, contexts cannot be easily specified because it is unclear how many products browsed

earlier should be considered as the context, and this number will likely differ for each product.

This second category of contextual outlier detection methods models the normal behavior with respect to contexts. Using a training data set, such a method trains a model that predicts the expected behavior attribute values with respect to the contextual attribute values. To determine whether a data object is a contextual outlier, we can then apply the model to the contextual attributes of the object. If the behavior attribute values of the object significantly deviate from the values predicted by the model, then the object can be declared a contextual outlier.

By using a prediction model that links the contexts and behavior, these methods avoid the explicit identification of specific contexts. A number of classification and prediction techniques can be used to build such models such as regression, Markov models, and finite state automaton. Interested readers are referred to Chapters 8 and 9 on classification and the bibliographic notes for further details (Section 12.11).

In summary, contextual outlier detection enhances conventional outlier detection by considering contexts, which are important in many applications. We may be able to detect outliers that cannot be detected otherwise. Consider a credit card user whose income level is low but whose expenditure patterns are similar to those of millionaires. This user can be detected as a contextual outlier if the income level is used to define context. Such a user may not be detected as an outlier without contextual information because she does share expenditure patterns with many millionaires. Considering contexts in outlier detection can also help to avoid false alarms. Without considering the context, a millionaire's purchase transaction may be falsely detected as an outlier if the majority of customers in the training set are not millionaires. This can be corrected by incorporating contextual information in outlier detection.

12.7.3 Mining Collective Outliers

A group of data objects forms a **collective outlier** if the objects as a whole deviate significantly from the entire data set, even though each individual object in the group may not be an outlier (Section 12.1). To detect collective outliers, we have to examine the *structure* of the data set, that is, the relationships between multiple data objects. This makes the problem more difficult than conventional and contextual outlier detection.

"*How can we explore the data set structure?*" This typically depends on the nature of the data. For outlier detection in temporal data (e.g., time series and sequences), we explore the structures formed by time, which occur in segments of the time series or subsequences. To detect collective outliers in spatial data, we explore local areas. Similarly, in graph and network data, we explore subgraphs. Each of these structures is inherent to its respective data type.

Contextual outlier detection and collective outlier detection are similar in that they both explore structures. In contextual outlier detection, the structures are the contexts, as specified by the contextual attributes explicitly. The critical difference in collective outlier detection is that the structures are often not explicitly defined, and have to be discovered as part of the outlier detection process.

As with contextual outlier detection, collective outlier detection methods can also be divided into two categories. The first category consists of methods that reduce the problem to conventional outlier detection. Its strategy is to identify *structure units*, treat each structure unit (e.g., a subsequence, a time-series segment, a local area, or a subgraph) as a data object, and extract features. The problem of collective outlier detection is thus transformed into outlier detection on the set of "structured objects" constructed as such using the extracted features. A structure unit, which represents a group of objects in the original data set, is a collective outlier if the structure unit deviates significantly from the expected trend in the space of the extracted features.

Example 12.23 **Collective outlier detection on graph data.** Let's see how we can detect collective outliers in *AllElectronics*' online social network of customers. Suppose we treat the social network as an unlabeled graph. We then treat each possible subgraph of the network as a structure unit. For each subgraph, S, let $|S|$ be the number of vertices in S, and $freq(S)$ be the frequency of S in the network. That is, $freq(S)$ is the number of different subgraphs in the network that are isomorphic to S. We can use these two features to detect outlier subgraphs. An *outlier subgraph* is a collective outlier that contains multiple vertices.

In general, a small subgraph (e.g., a single vertex or a pair of vertices connected by an edge) is expected to be frequent, and a large subgraph is expected to be infrequent. Using the preceding simple method, we can detect small subgraphs that are of very low frequency or large subgraphs that are surprisingly frequent. These are outlier structures in the social network. ■

Predefining the structure units for collective outlier detection can be difficult or impossible. Consequently, the second category of methods models the expected behavior of structure units directly. For example, to detect collective outliers in temporal sequences, one method is to learn a Markov model from the sequences. A subsequence can then be declared as a collective outlier if it significantly deviates from the model.

In summary, collective outlier detection is subtle due to the challenge of exploring the structures in data. The exploration typically uses heuristics, and thus may be application-dependent. The computational cost is often high due to the sophisticated mining process. While highly useful in practice, collective outlier detection remains a challenging direction that calls for further research and development.

12.8 Outlier Detection in High-Dimensional Data

In some applications, we may need to detect outliers in high-dimensional data. The dimensionality curse poses huge challenges for effective outlier detection. As the dimensionality increases, the distance between objects may be heavily dominated by noise. That is, the distance and similarity between two points in a high-dimensional space may not reflect the real relationship between the points. Consequently, conventional outlier detection methods, which mainly use proximity or density to identify outliers, deteriorate as dimensionality increases.

Ideally, outlier detection methods for high-dimensional data should meet the challenges that follow.

- **Interpretation of outliers:** They should be able to not only detect outliers, but also provide an interpretation of the outliers. Because many features (or dimensions) are involved in a high-dimensional data set, detecting outliers without providing any interpretation as to why they are outliers is not very useful. The interpretation of outliers may come from, for example, specific subspaces that manifest the outliers or an assessment regarding the "outlier-ness" of the objects. Such interpretation can help users to understand the possible meaning and significance of the outliers.
- **Data sparsity:** The methods should be capable of handling sparsity in high-dimensional spaces. The distance between objects becomes heavily dominated by noise as the dimensionality increases. Therefore, data in high-dimensional spaces are often sparse.
- **Data subspaces:** They should model outliers appropriately, for example, adaptive to the subspaces signifying the outliers and capturing the local behavior of data. Using a fixed-distance threshold against all subspaces to detect outliers is not a good idea because the distance between two objects monotonically increases as the dimensionality increases.
- **Scalability with respect to dimensionality:** As the dimensionality increases, the number of subspaces increases exponentially. An exhaustive combinatorial exploration of the search space, which contains all possible subspaces, is not a scalable choice.

Outlier detection methods for high-dimensional data can be divided into three main approaches. These include extending conventional outlier detection (Section 12.8.1), finding outliers in subspaces (Section 12.8.2), and modeling high-dimensional outliers (Section 12.8.3).

12.8.1 Extending Conventional Outlier Detection

One approach for outlier detection in high-dimensional data extends conventional outlier detection methods. It uses the conventional proximity-based models of outliers. However, to overcome the deterioration of proximity measures in high-dimensional spaces, it uses alternative measures or constructs subspaces and detects outliers there.

The **HilOut** algorithm is an example of this approach. HilOut finds distance-based outliers, but uses the ranks of distance instead of the absolute distance in outlier detection. Specifically, for each object, $\boldsymbol{o}$, HilOut finds the k-nearest neighbors of $\boldsymbol{o}$, denoted by $nn_1(\boldsymbol{o}), \ldots, nn_k(\boldsymbol{o})$, where k is an application-dependent parameter. The weight of object $\boldsymbol{o}$ is defined as

$$w(\boldsymbol{o}) = \sum_{i=1}^{k} dist(\boldsymbol{o}, nn_i(\boldsymbol{o})). \tag{12.21}$$

All objects are ranked in weight-descending order. The top-l objects in weight are output as outliers, where l is another user-specified parameter.

Computing the k-nearest neighbors for every object is costly and does not scale up when the dimensionality is high and the database is large. To address the scalability issue, HilOut employs space-filling curves to achieve an approximation algorithm, which is scalable in both running time and space with respect to database size and dimensionality.

While some methods like HilOut detect outliers in the full space despite the high dimensionality, other methods reduce the high-dimensional outlier detection problem to a lower-dimensional one by dimensionality reduction (Chapter 3). The idea is to reduce the high-dimensional space to a lower-dimensional space where normal instances can still be distinguished from outliers. If such a lower-dimensional space can be found, then conventional outlier detection methods can be applied.

To reduce dimensionality, general feature selection and extraction methods may be used or extended for outlier detection. For example, principal components analysis (PCA) can be used to extract a lower-dimensional space. Heuristically, the principal components with low variance are preferred because, on such dimensions, normal objects are likely close to each other and outliers often deviate from the majority.

By extending conventional outlier detection methods, we can reuse much of the experience gained from research in the field. These new methods, however, are limited. First, they cannot detect outliers with respect to subspaces and thus have limited interpretability. Second, dimensionality reduction is feasible only if there exists a lower-dimensional space where normal objects and outliers are well separated. This assumption may not hold true.

12.8.2 Finding Outliers in Subspaces

Another approach for outlier detection in high-dimensional data is to search for outliers in various subspaces. A unique advantage is that, if an object is found to be an outlier in a subspace of much lower dimensionality, the subspace provides critical information for interpreting *why* and *to what extent* the object is an outlier. This insight is highly valuable in applications with high-dimensional data due to the overwhelming number of dimensions.

Example 12.24 **Outliers in subspaces.** As a customer-relationship manager at *AllElectronics*, you are interested in finding outlier customers. *AllElectronics* maintains an extensive customer information database, which contains many attributes and the transaction history of customers. The database is high dimensional.

Suppose you find that a customer, Alice, is an outlier in a lower-dimensional subspace that contains the dimensions *average_transaction_amount* and *purchase_frequency*, such that her average transaction amount is substantially larger than the majority of the customers, and her purchase frequency is dramatically lower. The subspace itself speaks for why and to what extent Alice is an outlier. Using this information, you strategically decide to approach Alice by suggesting options that could improve her purchase frequency at *AllElectronics*. ■

"*How can we detect outliers in subspaces?*" We use a *grid-based subspace outlier detection method* to illustrate. The major ideas are as follows. We consider projections of the data onto various subspaces. If, in a subspace, we find an area that has a density that is much lower than average, then the area may contain outliers. To find such projections, we first discretize the data into a grid in an equal-depth way. That is, each dimension is partitioned into ϕ equal-depth ranges, where each range contains a fraction, f, of the objects $\left(f = \frac{1}{\phi}\right)$. Equal-depth partitioning is used because data along different dimensions may have different localities. An equal-width partitioning of the space may not be able to reflect such differences in locality.

Next, we search for regions defined by ranges in subspaces that are significantly sparse. To quantify what we mean by "significantly sparse," let's consider a k-dimensional cube formed by k ranges on k dimensions. Suppose the data set contains n objects. If the objects are independently distributed, the expected number of objects falling into a k-dimensional region is $\left(\frac{1}{\phi}\right)^k n = f^k n$. The standard deviation of the number of points in a k-dimensional region is $\sqrt{f^k(1-f^k)n}$. Suppose a specific k-dimensional cube C has $n(C)$ objects. We can define the **sparsity coefficient** of C as

$$S(C) = \frac{n(C) - f^k n}{\sqrt{f^k(1-f^k)n}}. \tag{12.22}$$

If $S(C) < 0$, then C contains fewer objects than expected. The smaller the value of $S(C)$ (i.e., the more negative), the sparser C is and the more likely the objects in C are outliers in the subspace.

By assuming $S(C)$ follows a normal distribution, we can use normal distribution tables to determine the probabilistic significance level for an object that deviates dramatically from the average for an a priori assumption of the data following a uniform distribution. In general, the assumption of uniform distribution does not hold. However, the sparsity coefficient still provides an intuitive measure of the "outlier-ness" of a region.

To find cubes of significantly small sparsity coefficient values, a brute-force approach is to search every cube in every possible subspace. The cost of this, however, is immediately exponential. An *evolutionary search* can be conducted, which improves efficiency at the expense of accuracy. For details, please refer to the bibliographic notes (Section 12.11). The objects contained by cubes of very small sparsity coefficient values are output as outliers.

In summary, searching for outliers in subspaces is advantageous in that the outliers found tend to be better understood, owing to the context provided by the subspaces. Challenges include making the search efficient and scalable.

12.8.3 Modeling High-Dimensional Outliers

An alternative approach for outlier detection methods in high-dimensional data tries to develop new models for high-dimensional outliers directly. Such models typically avoid

proximity measures and instead adopt new heuristics to detect outliers, which do not deteriorate in high-dimensional data.

Let's examine *angle-based outlier detection* (ABOD) as an example.

Example 12.25 **Angle-based outliers.** Figure 12.15 contains a set of points forming a cluster, with the exception of $\boldsymbol{c}$, which is an outlier. For each point $\boldsymbol{o}$, we examine the angle $\angle \boldsymbol{xoy}$ for every pair of points $\boldsymbol{x}$, $\boldsymbol{y}$ such that $\boldsymbol{x} \neq \boldsymbol{o}$, $\boldsymbol{y} \neq \boldsymbol{o}$. The figure shows angle $\angle \boldsymbol{dae}$ as an example.

Note that for a point in the center of a cluster (e.g., $\boldsymbol{a}$), the angles formed as such differ widely. For a point that is at the border of a cluster (e.g., $\boldsymbol{b}$), the angle variation is smaller. For a point that is an outlier (e.g., $\boldsymbol{c}$), the angle variable is substantially smaller. This observation suggests that we can use the variance of angles for a point to determine whether a point is an outlier. ■

We can combine angles and distance to model outliers. Mathematically, for each point $\boldsymbol{o}$, we use the distance-weighted angle variance as the outlier score. That is, given a set of points, D, for a point, $\boldsymbol{o} \in D$, we define the **angle-based outlier factor** (ABOF) as

$$ABOF(\boldsymbol{o}) = VAR_{x,y \in D, x \neq o, y \neq o} \frac{\langle \overrightarrow{\boldsymbol{ox}}, \overrightarrow{\boldsymbol{oy}} \rangle}{dist(\boldsymbol{o},\boldsymbol{x})^2 \, dist(\boldsymbol{o},\boldsymbol{y})^2}, \tag{12.23}$$

where $\langle , \rangle$ is the scalar product operator, and $dist(,)$ is a norm distance.

Clearly, the farther away a point is from clusters and the smaller the variance of the angles of a point, the smaller the ABOF. The ABOD computes the ABOF for each point, and outputs a list of the points in the data set in ABOF-ascending order.

Computing the exact ABOF for every point in a database is costly, requiring a time complexity of $O(n^3)$, where n is the number of points in the database. Obviously, this exact algorithm does not scale up for large data sets. Approximation methods have been developed to speed up the computation. The angle-based outlier detection idea has been generalized to handle arbitrary data types. For additional details, see the bibliographic notes (Section 12.11).

Developing native models for high-dimensional outliers can lead to effective methods. However, finding good heuristics for detecting high-dimensional outliers is difficult. Efficiency and scalability on large and high-dimensional data sets are major challenges.

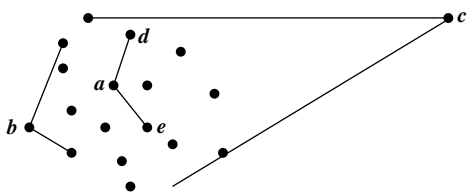

Figure 12.15 Angle-based outliers.

12.9 Summary

- Assume that a given statistical process is used to generate a set of data objects. An **outlier** is a data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism.
- **Types of outliers** include global outliers, contextual outliers, and collective outliers. An object may be more than one type of outlier.
- **Global outliers** are the simplest form of outlier and the easiest to detect. A **contextual outlier** deviates significantly with respect to a specific context of the object (e.g., a Toronto temperature value of 28°C is an outlier if it occurs in the context of winter). A subset of data objects forms a **collective outlier** if the objects as a whole deviate significantly from the entire data set, even though the individual data objects may not be outliers. Collective outlier detection requires background information to model the relationships among objects to find outlier groups.
- **Challenges** in outlier detection include finding appropriate data models, the dependence of outlier detection systems on the application involved, finding ways to distinguish outliers from noise, and providing justification for identifying outliers as such.
- Outlier detection methods can be **categorized** according to whether the sample of data for analysis is given with expert-provided labels that can be used to build an outlier detection model. In this case, the detection methods are *supervised, semi-supervised*, or *unsupervised*. Alternatively, outlier detection methods may be organized according to their assumptions regarding normal objects versus outliers. This categorization includes *statistical* methods, *proximity-based* methods, and *clustering-based* methods.
- **Statistical outlier detection methods** (or **model-based methods**) assume that the normal data objects follow a statistical model, where data not following the model are considered outliers. Such methods may be *parametric* (they assume that the data are generated by a parametric distribution) or *nonparametric* (they learn a model for the data, rather than assuming one a priori). Parametric methods for multivariate data may employ the Mahalanobis distance, the χ^2-statistic, or a mixture of multiple parametric models. Histograms and kernel density estimation are examples of nonparametric methods.
- **Proximity-based outlier detection methods** assume that an object is an outlier if the proximity of the object to its nearest neighbors significantly deviates from the proximity of most of the other objects to their neighbors in the same data set. *Distance-based outlier detection methods* consult the *neighborhood* of an object, defined by a given radius. An object is an outlier if its neighborhood does not have enough other points. In *density-based outlier detection methods*, an object is an outlier if its density is relatively much lower than that of its neighbors.

- **Clustering-based outlier detection methods** assume that the normal data objects belong to large and dense clusters, whereas outliers belong to small or sparse clusters, or do not belong to any clusters.
- **Classification-based outlier detection methods** often use a one-class model. That is, a classifier is built to describe only the normal class. Any samples that do not belong to the normal class are regarded as outliers.
- **Contextual outlier detection** and **collective outlier detection** explore structures in the data. In contextual outlier detection, the structures are defined as contexts using contextual attributes. In collective outlier detection, the structures are implicit and are explored as part of the mining process. To detect such outliers, one approach transforms the problem into one of conventional outlier detection. Another approach models the structures directly.
- **Outlier detection methods for high-dimensional data** can be divided into three main approaches. These include extending conventional outlier detection, finding outliers in subspaces, and modeling high-dimensional outliers.

12.10 Exercises

12.1 Give an application example where global outliers, contextual outliers, and collective outliers are all interesting. What are the attributes, and what are the contextual and behavioral attributes? How is the relationship among objects modeled in collective outlier detection?

12.2 Give an application example of where the border between normal objects and outliers is often unclear, so that the degree to which an object is an outlier has to be well estimated.

12.3 Adapt a simple semi-supervised method for outlier detection. Discuss the scenario where you have (a) only some labeled examples of normal objects, and (b) only some labeled examples of outliers.

12.4 Using an equal-depth histogram, design a way to assign an object an outlier score.

12.5 Consider the nested loop approach to mining distance-based outliers (Figure 12.6). Suppose the objects in a data set are arranged randomly, that is, each object has the same probability to appear in a position. Show that when the number of outlier objects is small with respect to the total number of objects in the whole data set, the expected number of distance calculations is linear to the number of objects.

12.6 In the density-based outlier detection method of Section 12.4.3, the definition of local reachability density has a potential problem: $lrd_k(o) = \infty$ may occur. Explain why this may occur and propose a fix to the issue.

12.7 Because clusters may form a hierarchy, outliers may belong to different granularity levels. Propose a clustering-based outlier detection method that can find outliers at different levels.

12.8 In outlier detection by semi-supervised learning, what is the advantage of using objects without labels in the training data set?

12.9 To understand why angle-based outlier detection is a heuristic method, give an example where it does not work well. Can you come up with a method to overcome this issue?

12.11 Bibliographic Notes

Hawkins [Haw80] defined outliers from a statistics angle. For surveys or tutorials on the subject of outlier and anomaly detection, see Chandola, Banerjee, and Kumar [CBK09]; Hodge and Austin [HA04]; Agyemang, Barker, and Alhajj [ABA06]; Markou and Singh [MS03a, MS03b]; Patcha and Park [PP07]; Beckman and Cook [BC83]; Ben-Gal [B-G05]; and Bakar, Mohemad, Ahmad, and Deris [BMAD06]. Song, Wu, Jermaine, et al. [SWJR-07] proposed the notion of conditional anomaly and contextual outlier detection.

Fujimaki, Yairi, and Machida [FYM05] presented an example of semi-supervised outlier detection using a set of labeled "normal" objects. For an example of semi-supervised outlier detection using labeled outliers, see Dasgupta and Majumdar [DM02].

Shewhart [She31] assumed that most objects follow a Gaussian distribution and used 3σ as the threshold for identifying outliers, where σ is the standard deviation. Boxplots are used to detect and visualize outliers in various applications such as medical data (Horn, Feng, Li, and Pesce [HFLP01]). Grubb's test was described by Grubbs [Gru69], Stefansky [Ste72], and Anscombe and Guttman [AG60]. Laurikkala, Juhola, and Kentala [LJK00] and Aggarwal and Yu [AY01] extended Grubb's test to detect multivariate outliers. Use of the χ^2-statistic to detect multivariate outliers was studied by Ye and Chen [YC01].

Agarwal [Aga06] used Gaussian mixture models to capture "normal data." Abraham and Box [AB79] assumed that outliers are generated by a normal distribution with a substantially larger variance. Eskin [Esk00] used the EM algorithm to learn mixture models for normal data and outliers.

Histogram-based outlier detection methods are popular in the application domain of intrusion detection (Eskin [Esk00] and Eskin, Arnold, Prerau, et al. [EAP+02]) and fault detection (Fawcett and Provost [FP97]).

The notion of distance-based outliers was developed by Knorr and Ng [KN97]. The index-based, nested loop–based, and grid-based approaches were explored (Knorr and Ng [KN98] and Knorr, Ng, and Tucakov [KNT00]) to speed up distance-based outlier detection. Bay and Schwabacher [BS03] and Jin, Tung, and Han [JTH01] pointed out that the CPU runtime of the nested loop method is often scalable with respect to database size. Tao, Xiao, and Zhou [TXZ06] presented an algorithm that finds all distance-based outliers by scanning the database three times with fixed main memory. For larger memory size, they proposed a method that uses only one or two scans.

The notion of density-based outliers was first developed by Breunig, Kriegel, Ng, and Sander [BKNS00]. Various methods proposed with the theme of density-based

outlier detection include Jin, Tung, and Han [JTH01]; Jin, Tung, Han, and Wang [JTHW06]; and Papadimitriou, Kitagawa, Gibbons, et al. [PKG-F03]. The variations differ in how they estimate density.

The bootstrap method discussed in Example 12.17 was developed by Barbara, Li, Couto, et al. [BLC+03]. The FindCBOLF algorithm was given by He, Xu, and Deng [HXD03]. For the use of fixed-width clustering in outlier detection methods, see Eskin, Arnold, and Prerau, et al. [EAP+02]; Mahoney and Chan [MC03]; and He, Xu, and Deng [HXD03]. Barbara, Wu, and Jajodia [BWJ01] used multiclass classification in network intrusion detection.

Song, Wu, Jermaine, et al. [SWJR07] and Fawcet and Provost [FP97] presented a method to reduce the problem of contextual outlier detection to one of conventional outlier detection. Yi, Sidiropoulos, Johnson, Jagadish, et al. [YSJJ+00] used regression techniques to detect contextual outliers in co-evolving sequences. The idea in Example 12.22 for collective outlier detection on graph data is based on Noble and Cook [NC03].

The HilOut algorithm was proposed by Angiulli and Pizzuti [AP05]. Aggarwal and Yu [AY01] developed the sparsity coefficient–based subspace outlier detection method. Kriegel, Schubert, and Zimek [KSZ08] proposed angle-based outlier detection.

13 Data Mining Trends and Research Frontiers

As a young research field, data mining has made significant progress and covered a broad spectrum of applications since the 1980s. Today, data mining is used in a vast array of areas. Numerous commercial data mining systems and services are available. Many challenges, however, still remain. In this final chapter, we introduce the mining of complex data types as a prelude to further in-depth study readers may choose to do. In addition, we focus on trends and research frontiers in data mining. Section 13.1 presents an overview of methodologies for mining complex data types, which extend the concepts and tasks introduced in this book. Such mining includes mining time-series, sequential patterns, and biological sequences; graphs and networks; spatiotemporal data, including geospatial data, moving-object data, and cyber-physical system data; multimedia data; text data; web data; and data streams. Section 13.2 briefly introduces other approaches to data mining, including statistical methods, theoretical foundations, and visual and audio data mining.

In Section 13.3, you will learn more about data mining applications in business and in science, including the financial retail, and telecommunication industries, science and engineering, and recommender systems. The social impacts of data mining are discussed in Section 13.4, including ubiquitous and invisible data mining, and privacy-preserving data mining. Finally, in Section 13.5 we speculate on current and expected data mining trends that arise in response to new challenges in the field.

13.1 Mining Complex Data Types

In this section, we outline the major developments and research efforts in mining complex data types. Complex data types are summarized in Figure 13.1. Section 13.1.1 covers mining sequence data such as time-series, symbolic sequences, and biological sequences. Section 13.1.2 discusses mining graphs and social and information networks. Section 13.1.3 addresses mining other kinds of data, including spatial data, spatiotemporal data, moving-object data, cyber-physical system data, multimedia data, text data,

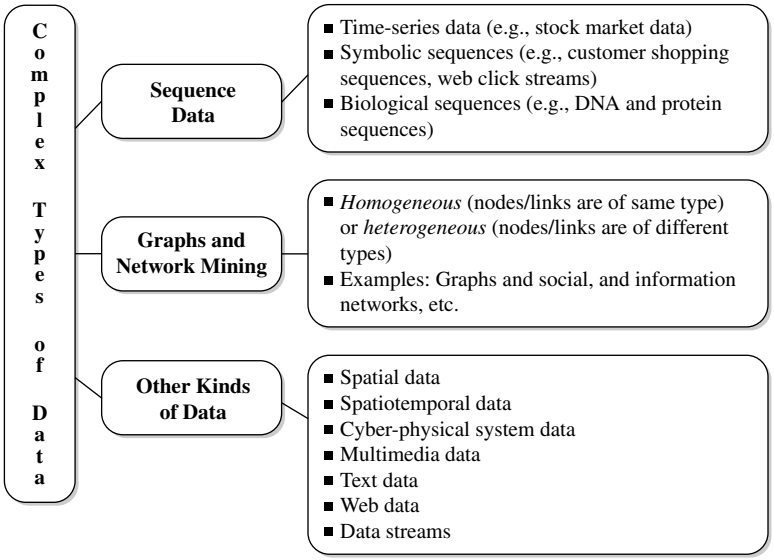

Figure 13.1 Complex data types for mining.

web data, and data streams. Due to the broad scope of these themes, this section presents only a high-level overview; these topics are not discussed in-depth in this book.

13.1.1 Mining Sequence Data: Time-Series, Symbolic Sequences, and Biological Sequences

A **sequence** is an ordered list of events. Sequences may be categorized into three groups, based on the characteristics of the events they describe: (1) *time-series data*, (2) *symbolic sequence data*, and (3) *biological sequences*. Let's consider each type.

In **time-series data**, sequence data consist of long sequences of numeric data, recorded at equal time intervals (e.g., per minute, per hour, or per day). Time-series data can be generated by many natural and economic processes such as stock markets, and scientific, medical, or natural observations.

Symbolic sequence data consist of long sequences of event or nominal data, which typically are not observed at equal time intervals. For many such sequences, *gaps* (i.e., lapses between recorded events) do not matter much. Examples include customer shopping sequences and web click streams, as well as sequences of events in science and engineering and in natural and social developments.

Biological sequences include DNA and protein sequences. Such sequences are typically very long, and carry important, complicated, but hidden semantic meaning. Here, gaps are usually important.

Let's look into data mining for each of these sequence data types.

Similarity Search in Time-Series Data

A **time-series data set** consists of sequences of numeric values obtained over repeated measurements of time. The values are typically measured at equal time intervals (e.g., every minute, hour, or day). Time-series databases are popular in many applications such as stock market analysis, economic and sales forecasting, budgetary analysis, utility studies, inventory studies, yield projections, workload projections, and process and quality control. They are also useful for studying natural phenomena (e.g., atmosphere, temperature, wind, earthquake), scientific and engineering experiments, and medical treatments.

Unlike normal database queries, which find data that match a given query *exactly*, a **similarity search** finds data sequences that *differ only slightly* from the given query sequence. Many time-series similarity queries require **subsequence matching**, that is, finding a set of sequences that contain subsequences that are similar to a given query sequence.

For similarity search, it is often necessary to first perform *data or dimensionality reduction and transformation* of time-series data. Typical *dimensionality reduction* techniques include (1) the *discrete Fourier transform* (*DFT*), (2) *discrete wavelet transforms (DWT)*, and (3) *singular value decomposition* (*SVD*) based on *principle components analysis* (*PCA*). Because we touched on these concepts in Chapter 3, and because a thorough explanation is beyond the scope of this book, we will not go into great detail here. With such techniques, the data or signal is mapped to a signal in a *transformed space*. A small subset of the "strongest" transformed coefficients are saved as features.

These features form a *feature space*, which is a projection of the transformed space. Indices can be constructed on the original or transformed time-series data to speed up a search. For a query-based similarity search, techniques include normalization transformation, atomic matching (i.e., finding pairs of gap-free windows of a small length that are similar), window stitching (i.e., stitching similar windows to form pairs of large similar subsequences, allowing gaps between atomic matches), and subsequence ordering (i.e., linearly ordering the subsequence matches to determine whether enough similar pieces exist). Numerous software packages exist for a similarity search in time-series data.

Recently, researchers have proposed transforming time-series data into piecewise aggregate approximations so that the data can be viewed as a sequence of symbolic representations. The problem of similarity search is then transformed into one of matching subsequences in symbolic sequence data. We can identify *motifs* (i.e., frequently occurring sequential patterns) and build index or hashing mechanisms for an efficient search based on such motifs. Experiments show this approach is fast and simple, and has comparable search quality to that of DFT, DWT, and other dimensionality reduction methods.

Regression and Trend Analysis in Time-Series Data

Regression analysis of time-series data has been studied substantially in the fields of statistics and signal analysis. However, one may often need to go beyond pure regression

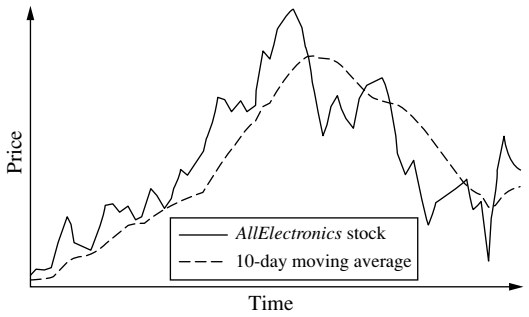

Figure 13.2 Time-series data for the stock price of *AllElectronics* over time. The *trend* is shown with a dashed curve, calculated by a moving average.

analysis and perform *trend analysis* for many practical applications. Trend analysis builds an integrated model using the following four major *components* or *movements* to characterize time-series data:

1. **Trend or long-term movements**: These indicate the general direction in which a time-series graph is moving over time, for example, using *weighted moving average* and the *least squares* methods to find *trend curves* such as the dashed curve indicated in Figure 13.2.
2. **Cyclic movements**: These are the long-term oscillations about a trend line or curve.
3. **Seasonal variations**: These are nearly identical patterns that a time series appears to follow during corresponding seasons of successive years such as holiday shopping seasons. For effective trend analysis, the data often need to be "deseasonalized" based on a **seasonal index** computed by autocorrelation.
4. **Random movements**: These characterize sporadic changes due to chance events such as labor disputes or announced personnel changes within companies.

Trend analysis can also be used for **time-series forecasting**, that is, finding a mathematical function that will approximately generate the historic patterns in a time series, and using it to make long-term or short-term predictions of future values. *ARIMA (auto-regressive integrated moving average)*, *long-memory time-series modeling*, and *autoregression* are popular methods for such analysis.

Sequential Pattern Mining in Symbolic Sequences

A **symbolic sequence** consists of an ordered set of elements or events, recorded with or without a concrete notion of time. There are many applications involving data of

symbolic sequences such as customer shopping sequences, web click streams, program execution sequences, biological sequences, and sequences of events in science and engineering and in natural and social developments. Because biological sequences carry very complicated semantic meaning and pose many challenging research issues, most investigations are conducted in the field of bioinformatics.

Sequential pattern mining has focused extensively on mining symbolic sequences. A sequential pattern is a frequent subsequence existing in a single sequence or a set of sequences. A sequence $\alpha = \langle a_1 a_2 \cdots a_n \rangle$ is a **subsequence** of another sequence $\beta = \langle b_1 b_2 \cdots b_m \rangle$ if there exist integers $1 \leq j_1 < j_2 < \cdots < j_n \leq m$ such that $a_1 \subseteq b_{j_1}$, $a_2 \subseteq b_{j_2}, \ldots, a_n \subseteq b_{j_n}$. For example, if $\alpha = \langle \{ab\}, d \rangle$ and $\beta = \langle \{abc\}, \{be\}, \{de\}, a \rangle$, where a, b, c, d, and e are items, then α is a subsequence of β. Mining of sequential patterns consists of mining the set of subsequences that are frequent in one sequence or a set of sequences. Many scalable algorithms have been developed as a result of extensive studies in this area. Alternatively, we can mine only the *set of closed* sequential patterns, where a sequential pattern s is **closed** if there exists no sequential pattern s', where s is a *proper* subsequence of s', and s' has the same (frequency) support as s. Similar to its frequent pattern mining counterpart, there are also studies on efficient mining of **multidimensional, multilevel sequential patterns**.

As with constraint-based frequent pattern mining, user-specified constraints can be used to reduce the search space in sequential pattern mining and derive only the patterns that are of interest to the user. This is referred to as **constraint-based sequential pattern mining**. Moreover, we may relax constraints or enforce additional constraints on the problem of sequential pattern mining to derive different kinds of patterns from sequence data. For example, we can enforce gap constraints so that the patterns derived contain only consecutive subsequences or subsequences with very small gaps. Alternatively, we may derive periodic sequential patterns by folding events into proper-size windows and finding recurring subsequences in these windows. Another approach derives *partial order patterns* by relaxing the requirement of strict sequential ordering in the mining of subsequence patterns. Besides mining partial order patterns, sequential pattern mining methodology can also be extended to mining trees, lattices, episodes, and some other ordered patterns.

Sequence Classification

Most classification methods perform model construction based on feature vectors. However, sequences do not have explicit features. Even with sophisticated feature selection techniques, the dimensionality of potential features can still be very high and the sequential nature of features is difficult to capture. This makes sequence classification a challenging task.

Sequence classification methods can be organized into three categories: (1) feature-based classification, which transforms a sequence into a feature vector and then applies conventional classification methods; (2) sequence distance–based classification, where the distance function that measures the similarity between sequences determines the

quality of the classification significantly; and (3) model-based classification such as using hidden Markov model (HMM) or other statistical models to classify sequences.

For time-series or other numeric-valued data, the feature selection techniques for symbolic sequences cannot be easily applied to time-series data without discretization. However, discretization can cause information loss. A recently proposed time-series *shapelets method* uses the time-series subsequences that can maximally represent a class as the features. It achieves quality classification results.

Alignment of Biological Sequences

Biological sequences generally refer to sequences of nucleotides or amino acids. **Biological sequence analysis** compares, aligns, indexes, and analyzes biological sequences and thus plays a crucial role in bioinformatics and modern biology.

Sequence alignment is based on the fact that all living organisms are related by evolution. This implies that the nucleotide (DNA, RNA) and protein sequences of species that are closer to each other in evolution should exhibit more similarities. An **alignment** is the process of lining up sequences to achieve a maximal identity level, which also expresses the degree of similarity between sequences. Two sequences are **homologous** if they share a common ancestor. The degree of similarity obtained by sequence alignment can be useful in determining the possibility of homology between two sequences. Such an alignment also helps determine the relative positions of multiple species in an evolution tree, which is called a **phylogenetic tree**.

The problem of alignment of biological sequences can be described as follows: *Given two or more input biological sequences, identify similar sequences with long conserved subsequences.* If the number of sequences to be aligned is exactly two, the problem is known as **pairwise sequence alignment**; otherwise, it is **multiple sequence alignment**. The sequences to be compared and aligned can be either nucleotides (DNA/RNA) or amino acids (proteins). For nucleotides, two symbols align if they are identical. However, for amino acids, two symbols align if they are identical, or if one can be derived from the other by substitutions that are likely to occur in nature. There are two kinds of alignments: *local alignments* and *global alignments*. The former means that only portions of the sequences are aligned, whereas the latter requires alignment over the entire length of the sequences.

For either nucleotides or amino acids, insertions, deletions, and substitutions occur in nature with different probabilities. **Substitution matrices** are used to represent the probabilities of substitutions of nucleotides or amino acids and probabilities of insertions and deletions. Usually, we use the gap character, −, to indicate positions where it is preferable not to align two symbols. To evaluate the quality of alignments, a *scoring* mechanism is typically defined, which usually counts identical or similar symbols as positive scores and gaps as negative ones. The algebraic sum of the scores is taken as the alignment measure. The goal of alignment is to achieve the maximal score among all the possible alignments. However, it is very expensive (more exactly, an NP-hard problem) to find optimal alignment. Therefore, various heuristic methods have been developed to find suboptimal alignments.

The dynamic programming approach is commonly used for sequence alignments. Among many available analysis packages, BLAST (Basic Local Alignment Search Tool) is one of the most popular tools in biosequence analysis.

Hidden Markov Model for Biological Sequence Analysis

Given a biological sequence, biologists would like to analyze what that sequence represents. To represent the structure or statistical regularities of sequence classes, biologists construct various probabilistic models such as *Markov chains* and *hidden Markov models*. In both models, the probability of a state depends only on that of the previous state; therefore, they are particularly useful for the analysis of biological sequence data. The most common methods for constructing hidden Markov models are the forward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm. Given a sequence of symbols, x, the *forward algorithm* finds the probability of obtaining x in the model; the *Viterbi algorithm* finds the most probable path (corresponding to x) through the model, whereas the *Baum-Welch algorithm* learns or adjusts the model parameters so as to best explain a set of training sequences.

13.1.2 Mining Graphs and Networks

Graphs represents a more general class of structures than sets, sequences, lattices, and trees. There is a broad range of graph applications on the Web and in social networks, information networks, biological networks, bioinformatics, chemical informatics, computer vision, and multimedia and text retrieval. Hence, graph and network mining have become increasingly important and heavily researched. We overview the following major themes: (1) graph pattern mining; (2) statistical modeling of networks; (3) data cleaning, integration, and validation by network analysis; (4) clustering and classification of graphs and homogeneous networks; (5) clustering, ranking, and classification of heterogeneous networks; (6) role discovery and link prediction in information networks; (7) similarity search and OLAP in information networks; and (8) evolution of information networks.

Graph Pattern Mining

Graph pattern mining is the mining of *frequent subgraphs* (also called **(sub)graph patterns**) in one or a set of graphs. Methods for mining graph patterns can be categorized into Apriori-based and pattern growth–based approaches. Alternatively, we can mine the set of *closed graphs* where a graph g is *closed* if there exists no proper supergraph g' that carries the same support count as g. Moreover, there are many *variant graph patterns*, including approximate frequent graphs, coherent graphs, and dense graphs. User-specified constraints can be pushed deep into the graph pattern mining process to improve mining efficiency.

Graph pattern mining has many interesting applications. For example, it can be used to generate compact and effective *graph index structures* based on the concept of

frequent and discriminative graph patterns. Approximate *structure similarity search* can be achieved by exploring graph index structures and multiple graph features. Moreover, classification of graphs can also be performed effectively using frequent and discriminative subgraphs as features.

Statistical Modeling of Networks

A **network** consists of a set of *nodes*, each corresponding to an *object* associated with a set of properties, and a set of *edges* (or *links*) connecting those nodes, representing relationships between objects. A network is **homogeneous** if all the nodes and links are of the same type, such as a friend network, a coauthor network, or a web page network. A network is **heterogeneous** if the nodes and links are of different types, such as publication networks (linking together authors, conferences, papers, and contents), and health-care networks (linking together doctors, nurses, patients, diseases, and treatments).

Researchers have proposed multiple statistical models for modeling homogeneous networks. The most well-known generative models are the random graph model (i.e., the Erdös-Rényi model), the Watts-Strogatz model, and the scale-free model. The scale-free model assumes that the network follows the *power law distribution* (also known as the *Pareto distribution* or the *heavy-tailed distribution*). In most large-scale social networks, a **small-world phenomenon** is observed, that is, the network can be characterized as having a high degree of local clustering for a small fraction of the nodes (i.e., these nodes are interconnected with one another), while being no more than a few degrees of separation from the remaining nodes.

Social networks exhibit certain evolutionary characteristics. They tend to follow the **densification power law**, which states that networks become increasingly *dense* over time. **Shrinking diameter** is another characteristic, where the effective diameter often *decreases* as the network grows. Node *out-degrees* and *in-degrees* typically follow a heavy-tailed distribution.

Data Cleaning, Integration, and Validation by Information Network Analysis

Real-world data are often incomplete, noisy, uncertain, and unreliable. Information redundancy may exist among the multiple pieces of data that are interconnected in a large network. Information redundancy can be explored in such networks to perform quality data cleaning, data integration, information validation, and trustability analysis by network analysis. For example, we can distinguish authors who share the same names by examining the networked connections with other heterogeneous objects such as coauthors, publication venues, and terms. In addition, we can identify inaccurate author information presented by booksellers by exploring a network built based on author information provided by multiple booksellers.

Sophisticated information network analysis methods have been developed in this direction, and in many cases, portions of the data serve as the "training set." That is, relatively clean and reliable data or a consensus of data from multiple information

providers can be used to help consolidate the remaining, unreliable portions of the data. This reduces the costly efforts of labeling the data by hand and of training on massive, dynamic, real-world data sets.

Clustering and Classification of Graphs and Homogeneous Networks

Large graphs and networks have cohesive structures, which are often hidden among their massive, interconnected nodes and links. Cluster analysis methods have been developed on large networks to uncover network structures, discover hidden communities, hubs, and outliers based on network topological structures and their associated properties. Various kinds of network clustering methods have been developed and can be categorized as either partitioning, hierarchical, or density-based algorithms. Moreover, given human-labeled training data, the discovery of network structures can be guided by human-specified heuristic constraints. Supervised classification and semi-supervised classification of networks are recent hot topics in the data mining research community.

Clustering, Ranking, and Classification of Heterogeneous Networks

A heterogeneous network contains interconnected nodes and links of different types. Such interconnected structures contain rich information, which can be used to mutually enhance nodes and links, and propagate knowledge from one type to another. Clustering and ranking of such heterogeneous networks can be performed hand-in-hand in the context that highly ranked nodes/links in a cluster may contribute more than their lower-ranked counterparts in the evaluation of the cohesiveness of a cluster. Clustering may help consolidate the high ranking of objects/links dedicated to the cluster. Such mutual enhancement of ranking and clustering prompted the development of an algorithm called RankClus. Moreover, users may specify different ranking rules or present labeled nodes/links for certain data types. Knowledge of one type can be propagated to other types. Such propagation reaches the nodes/links of the same type via heterogeneous-type connections. Algorithms have been developed for supervised learning and semi-supervised learning in heterogeneous networks.

Role Discovery and Link Prediction in Information Networks

There exist many hidden roles or relationships among different nodes/links in a heterogeneous network. Examples include advisor–advisee and leader–follower relationships in a research publication network. To discover such hidden roles or relationships, experts can specify constraints based on their background knowledge. Enforcing such constraints may help cross-checking and validation in large interconnected networks. Information redundancy in a network can often be used to help weed out objects/links that do not follow such constraints.

Similarly, *link prediction* can be performed based on the assessment of the ranking of the expected relationships among the candidate nodes/links. For example, we may predict which papers an author may write, read, or cite, based on the author's recent publication history and the trend of research on similar topics. Such studies often require analyzing the proximity of network nodes/links and the trends and connections of their similar neighbors. Roughly speaking, people refer to link prediction as **link mining**; however, link mining covers additional tasks including *link-based object classification*, *object type prediction*, *link type prediction, link existence prediction*, *link cardinality estimation*, and *object reconciliation* (which predicts whether two objects are, in fact, the same). It also includes *group detection* (which clusters objects), as well as *subgraph identification* (which finds characteristic subgraphs within networks) and *metadata mining* (which uncovers schema-type information regarding unstructured data).

Similarity Search and OLAP in Information Networks

Similarity search is a primitive operation in database and web search engines. A heterogeneous information network consists of multityped, interconnected objects. Examples include bibliographic networks and social media networks, where two objects are considered similar if they are linked in a similar way with multityped objects. In general, object similarity within a network can be determined based on network structures and object properties, and with similarity measures. Moreover, network clusters and hierarchical network structures help organize objects in a network and identify subcommunities, as well as facilitate similarity search. Furthermore, similarity can be defined differently per user. By considering different linkage paths, we can derive various similarity semantics in a network, which is known as *path-based similarity*.

By organizing networks based on the notion of similarity and clusters, we can generate multiple hierarchies within a network. Online analytical processing (OLAP) can then be performed. For example, we can drill down or dice information networks based on different levels of abstraction and different angles of views. OLAP operations may generate multiple, interrelated networks. The relationships among such networks may disclose interesting hidden semantics.

Evolution of Social and Information Networks

Networks are dynamic and constantly evolving. Detecting evolving communities and evolving regularities or anomalies in homogeneous or heterogeneous networks can help people better understand the structural evolution of networks and predict trends and irregularities in evolving networks. For homogeneous networks, the evolving communities discovered are subnetworks consisting of objects of the same type such as a set of friends or coauthors. However, for heterogeneous networks, the communities discovered are subnetworks consisting of objects of different types, such as a connected set of papers, authors, venues, and terms, from which we can also derive a set of evolving objects for each type, like evolving authors and themes.

13.1.3 Mining Other Kinds of Data

In addition to sequences and graphs, there are many other kinds of semi-structured or unstructured data, such as spatiotemporal, multimedia, and hypertext data, which have interesting applications. Such data carry various kinds of semantics, are either stored in or dynamically streamed through a system, and call for specialized data mining methodologies. Thus, mining multiple kinds of data, including *spatial data, spatiotemporal data, cyber-physical system data, multimedia data, text data*, *web data*, and *data streams*, are increasingly important tasks in data mining. In this subsection, we overview the methodologies for mining these kinds of data.

Mining Spatial Data

Spatial data mining discovers patterns and knowledge from spatial data. Spatial data, in many cases, refer to geospace-related data stored in geospatial data repositories. The data can be in "vector" or "raster" formats, or in the form of imagery and geo-referenced multimedia. Recently, large *geographic data warehouses* have been constructed by integrating thematic and geographically referenced data from multiple sources. From these, we can construct *spatial data cubes* that contain spatial dimensions and measures, and support *spatial OLAP* for *multidimensional spatial data analysis*. Spatial data mining can be performed on spatial data warehouses, spatial databases, and other geospatial data repositories. Popular topics on geographic knowledge discovery and spatial data mining include *mining spatial associations and co-location patterns*, *spatial clustering*, *spatial classification*, *spatial modeling*, and *spatial trend and outlier analysis*.

Mining Spatiotemporal Data and Moving Objects

Spatiotemporal data are data that relate to both space and time. **Spatiotemporal data mining** refers to the process of discovering patterns and knowledge from spatiotemporal data. Typical examples of spatiotemporal data mining include discovering the evolutionary history of cities and lands, uncovering weather patterns, predicting earthquakes and hurricanes, and determining global warming trends. Spatiotemporal data mining has become increasingly important and has far-reaching implications, given the popularity of mobile phones, GPS devices, Internet-based map services, weather services, and digital Earth, as well as satellite, RFID, sensor, wireless, and video technologies.

Among many kinds of spatiotemporal data, *moving-object data* (i.e., data about moving objects) are especially important. For example, animal scientists attach telemetry equipment on wildlife to analyze ecological behavior, mobility managers embed GPS in cars to better monitor and guide vehicles, and meteorologists use weather satellites and radars to observe hurricanes. Massive-scale moving-object data are becoming rich, complex, and ubiquitous. Examples of **moving-object data mining** include mining *movement patterns of multiple moving objects* (i.e., the discovery of relationships among multiple moving objects such as moving clusters, leaders and followers, merge, convoy, swarm, and pincer, as well as other collective movement patterns). Other examples of

moving-object data mining include mining *periodic patterns* for one or a set of moving objects, and mining *trajectory patterns, clusters, models*, and *outliers*.

Mining Cyber-Physical System Data

A **cyber-physical system** (CPS) typically consists of a large number of interacting physical and information components. CPS systems may be interconnected so as to form large heterogeneous *cyber-physical networks*. Examples of cyber-physical networks include a patient care system that links a patient monitoring system with a network of patient/medical information and an emergency handling system; a transportation system that links a transportation monitoring network, consisting of many sensors and video cameras, with a traffic information and control system; and a battlefield commander system that links a sensor/reconnaissance network with a battlefield information analysis system. Clearly, cyber-physical systems and networks will be ubiquitous and form a critical component of modern information infrastructure.

Data generated in cyber-physical systems are dynamic, volatile, noisy, inconsistent, and interdependent, containing rich spatiotemporal information, and they are critically important for real-time decision making. In comparison with typical spatiotemporal data mining, mining cyber-physical data requires linking the current situation with a large information base, performing real-time calculations, and returning prompt responses. Research in the area includes rare-event detection and anomaly analysis in cyber-physical data streams, reliability and trustworthiness in cyber-physical data analysis, effective spatiotemporal data analysis in cyber-physical networks, and the integration of stream data mining with real-time automated control processes.

Mining Multimedia Data

Multimedia data mining is the discovery of interesting patterns from multimedia databases that store and manage large collections of multimedia objects, including image data, video data, audio data, as well as sequence data and hypertext data containing text, text markups, and linkages. Multimedia data mining is an interdisciplinary field that integrates image processing and understanding, computer vision, data mining, and pattern recognition. Issues in multimedia data mining include *content-based retrieval and similarity search*, and *generalization and multidimensional analysis*. Multimedia data cubes contain additional dimensions and measures for multimedia information. Other topics in multimedia mining include *classification and prediction analysis*, *mining associations*, and *video and audio data mining* (Section 13.2.3).

Mining Text Data

Text mining is an interdisciplinary field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics. A substantial portion of information is stored as text such as news articles, technical papers, books, digital libraries, email messages, blogs, and web pages. Hence, research in text mining has been very active. An important goal is to derive high-quality information from text. This is

typically done through the discovery of patterns and trends by means such as statistical pattern learning, topic modeling, and statistical language modeling. Text mining usually requires structuring the input text (e.g., parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database). This is followed by deriving patterns within the structured data, and evaluation and interpretation of the output. "High quality" in text mining usually refers to a combination of relevance, novelty, and interestingness.

Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity-relation modeling (i.e., learning relations between named entities). Other examples include multilingual data mining, multidimensional text analysis, contextual text mining, and trust and evolution analysis in text data, as well as text mining applications in security, biomedical literature analysis, online media analysis, and analytical customer relationship management. Various kinds of text mining and analysis software and tools are available in academic institutions, open-source forums, and industry. Text mining often also uses WordNet, Sematic Web, Wikipedia, and other information sources to enhance the understanding and mining of text data.

Mining Web Data

The World Wide Web serves as a huge, widely distributed, global information center for news, advertisements, consumer information, financial management, education, government, and e-commerce. It contains a rich and dynamic collection of information about web page contents with hypertext structures and multimedia, hyperlink information, and access and usage information, providing fertile sources for data mining. **Web mining** is the application of data mining techniques to discover patterns, structures, and knowledge from the Web. According to analysis targets, web mining can be organized into three main areas: *web content mining*, *web structure mining*, and *web usage mining*.

Web content mining analyzes web content such as text, multimedia data, and structured data (within web pages or linked across web pages). This is done to understand the content of web pages, provide scalable and informative keyword-based page indexing, entity/concept resolution, web page relevance and ranking, web page content summaries, and other valuable information related to web search and analysis. Web pages can reside either on the *surface web* or on the *deep Web*. The *surface web* is that portion of the Web that is indexed by typical search engines. The *deep Web* (or *hidden Web*) refers to web content that is not part of the surface web. Its contents are provided by underlying database engines.

Web content mining has been studied extensively by researchers, search engines, and other web service companies. Web content mining can build links across multiple web pages for individuals; therefore, it has the potential to inappropriately disclose personal information. Studies on privacy-preserving data mining address this concern through the development of techniques to protect personal privacy on the Web.

Web structure mining is the process of using graph and network mining theory and methods to analyze the nodes and connection structures on the Web. It extracts patterns from hyperlinks, where a hyperlink is a structural component that connects a

web page to another location. It can also mine the document structure within a page (e.g., analyze the treelike structure of page structures to describe HTML or XML tag usage). Both kinds of web structure mining help us understand web contents and may also help transform web contents into relatively structured data sets.

Web usage mining is the process of extracting useful information (e.g., user click streams) from server logs. It finds patterns related to general or particular groups of users; understands users' search patterns, trends, and associations; and predicts what users are looking for on the Internet. It helps improve search efficiency and effectiveness, as well as promotes products or related information to different groups of users at the right time. Web search companies routinely conduct web usage mining to improve their quality of service.

Mining Data Streams

Stream data refer to data that flow into a system in vast volumes, change dynamically, are possibly infinite, and contain multidimensional features. Such data cannot be stored in traditional database systems. Moreover, most systems may only be able to read the stream once in sequential order. This poses great challenges for the effective mining of stream data. Substantial research has led to progress in the development of efficient methods for mining data streams, in the areas of mining frequent and sequential patterns, multidimensional analysis (e.g., the construction of stream cubes), classification, clustering, outlier analysis, and the online detection of rare events in data streams. The general philosophy is to develop single-scan or a-few-scan algorithms using limited computing and storage capabilities.

This includes collecting information about stream data in sliding windows or *tilted time windows* (where the most recent data are registered at the finest granularity and the more distant data are registered at a coarser granularity), and exploring techniques like microclustering, limited aggregation, and approximation. Many applications of stream data mining can be explored—for example, real-time detection of anomalies in computer network traffic, botnets, text streams, video streams, power-grid flows, web searches, sensor networks, and cyber-physical systems.

13.2 Other Methodologies of Data Mining

Due to the broad scope of data mining and the large variety of data mining methodologies, not all methodologies of data mining can be thoroughly covered in this book. In this section, we briefly discuss several interesting methodologies that were not fully addressed in the previous chapters. These methodologies are listed in Figure 13.3.

13.2.1 Statistical Data Mining

The data mining techniques described in this book are primarily drawn from computer science disciplines, including data mining, machine learning, data warehousing, and algorithms. They are designed for the efficient handling of huge amounts of data that are

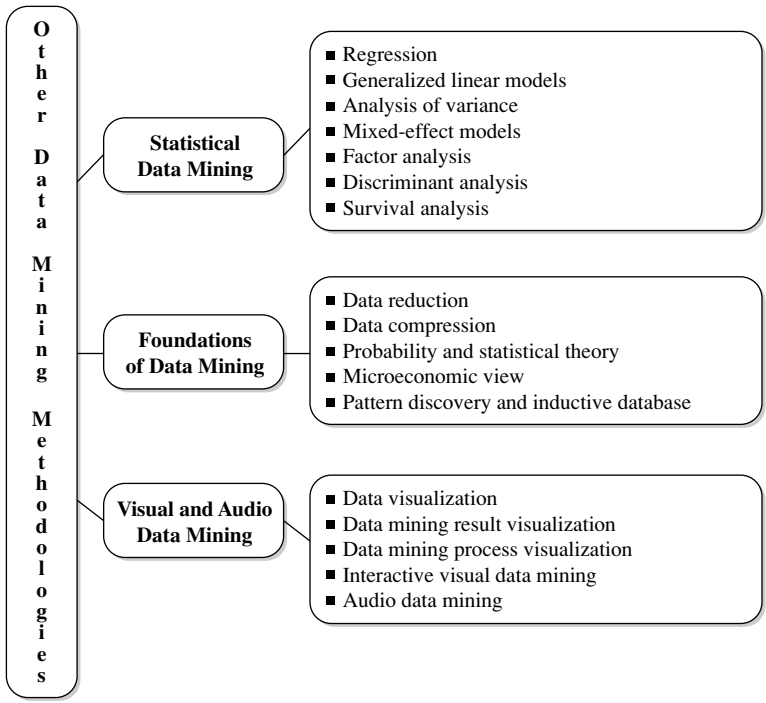

Figure 13.3 Other data mining methodologies.

typically multidimensional and possibly of various complex types. There are, however, many well-established statistical techniques for data analysis, particularly for numeric data. These techniques have been applied extensively to scientific data (e.g., data from experiments in physics, engineering, manufacturing, psychology, and medicine), as well as to data from economics and the social sciences. Some of these techniques, such as principal components analysis (Chapter 3) and clustering (Chapters 10 and 11), have already been addressed in this book. A thorough discussion of major statistical methods for data analysis is beyond the scope of this book; however, several methods are mentioned here for the sake of completeness. Pointers to these techniques are provided in the bibliographic notes (Section 13.8).

- **Regression**: In general, these methods are used to predict the value of a *response* (dependent) variable from one or more *predictor* (independent) variables, where the variables are numeric. There are various forms of regression, such as linear, multiple, weighted, polynomial, nonparametric, and robust (robust methods are useful when errors fail to satisfy normalcy conditions or when the data contain significant outliers).
- **Generalized linear models**: These models, and their generalization (*generalized additive models*), allow a *categorical* (nominal) response variable (or some transformation

of it) to be related to a set of predictor variables in a manner similar to the modeling of a numeric response variable using linear regression. Generalized linear models include logistic regression and Poisson regression.

- **Analysis of variance**: These techniques analyze experimental data for two or more populations described by a numeric response variable and one or more categorical variables (*factors*). In general, an ANOVA (single-factor analysis of variance) problem involves a comparison of k population or treatment means to determine if at least two of the means are different. More complex ANOVA problems also exist.
- **Mixed-effect models**: These models are for analyzing grouped data—data that can be classified according to one or more grouping variables. They typically describe relationships between a response variable and some covariates in data grouped according to one or more factors. Common areas of application include multilevel data, repeated measures data, block designs, and longitudinal data.
- **Factor analysis**: This method is used to determine which variables are combined to generate a given factor. For example, for many psychiatric data, it is not possible to measure a certain factor of interest directly (e.g., intelligence); however, it is often possible to measure other quantities (e.g., student test scores) that reflect the factor of interest. Here, none of the variables is designated as dependent.
- **Discriminant analysis**: This technique is used to predict a categorical response variable. Unlike generalized linear models, it assumes that the independent variables follow a multivariate normal distribution. The procedure attempts to determine several discriminant functions (linear combinations of the independent variables) that discriminate among the groups defined by the response variable. Discriminant analysis is commonly used in social sciences.
- **Survival analysis**: Several well-established statistical techniques exist for survival analysis. These techniques originally were designed to predict the probability that a patient undergoing a medical treatment would survive at least to time t. Methods for survival analysis, however, are also commonly applied to manufacturing settings to estimate the life span of industrial equipment. Popular methods include Kaplan-Meier estimates of survival, Cox proportional hazards regression models, and their extensions.
- **Quality control**: Various statistics can be used to prepare charts for quality control, such as Shewhart charts and CUSUM charts (both of which display group summary statistics). These statistics include the mean, standard deviation, range, count, moving average, moving standard deviation, and moving range.

13.2.2 Views on Data Mining Foundations

Research on the theoretical foundations of data mining has yet to mature. A solid and systematic theoretical foundation is important because it can help provide a coherent

framework for the development, evaluation, and practice of data mining technology. Several theories for the basis of data mining include the following:

- **Data reduction**: In this theory, the basis of data mining is to reduce the data representation. Data reduction trades accuracy for speed in response to the need to obtain quick approximate answers to queries on very large databases. Data reduction techniques include singular value decomposition (the driving element behind principal components analysis), wavelets, regression, log-linear models, histograms, clustering, sampling, and the construction of index trees.
- **Data compression**: According to this theory, the basis of data mining is to compress the given data by encoding in terms of bits, association rules, decision trees, clusters, and so on. Encoding based on the *minimum description length principle* states that the "best" theory to infer from a data set is the one that minimizes the length of the theory and of the data when encoded, using the theory as a predictor for the data. This encoding is typically in bits.
- **Probability and statistical theory**: According to this theory, the basis of data mining is to discover joint probability distributions of random variables, for example, Bayesian belief networks or hierarchical Bayesian models.
- **Microeconomic view**: The microeconomic view considers data mining as the task of finding patterns that are interesting only to the extent that they can be used in the decision-making process of some enterprise (e.g., regarding marketing strategies and production plans). This view is one of utility, in which patterns are considered interesting if they can be acted on. Enterprises are regarded as facing optimization problems, where the object is to maximize the utility or value of a decision. In this theory, data mining becomes a nonlinear optimization problem.
- **Pattern discovery and inductive databases**: In this theory, the basis of data mining is to discover patterns occurring in the data such as associations, classification models, sequential patterns, and so on. Areas such as machine learning, neural network, association mining, sequential pattern mining, clustering, and several other subfields contribute to this theory. A knowledge base can be viewed as a database consisting of data and patterns. A user interacts with the system by querying the data and the theory (i.e., patterns) in the knowledge base. Here, the knowledge base is actually an inductive database.

These theories are not mutually exclusive. For example, pattern discovery can also be seen as a form of data reduction or data compression. Ideally, a theoretical framework should be able to model typical data mining tasks (e.g., association, classification, and clustering), have a probabilistic nature, be able to handle different forms of data, and consider the iterative and interactive essence of data mining. Further efforts are required to establish a well-defined framework for data mining that satisfies these requirements.

13.2.3 Visual and Audio Data Mining

Visual data mining discovers implicit and useful knowledge from large data sets using data and/or knowledge visualization techniques. The human visual system is controlled by the eyes and brain, the latter of which can be thought of as a powerful, highly parallel processing and reasoning engine containing a large knowledge base. Visual data mining essentially combines the power of these components, making it a highly attractive and effective tool for the comprehension of data distributions, patterns, clusters, and outliers in data.

Visual data mining can be viewed as an integration of two disciplines: data visualization and data mining. It is also closely related to computer graphics, multimedia systems, human–computer interaction, pattern recognition, and high-performance computing. In general, data visualization and data mining can be integrated in the following ways:

- **Data visualization**: Data in a database or data warehouse can be viewed at different granularity or abstraction levels, or as different combinations of attributes or dimensions. Data can be presented in various visual forms, such as boxplots, 3-D cubes, data distribution charts, curves, surfaces, and link graphs, as shown in the data visualization section of Chapter 2. Figures 13.4 and 13.5 from StatSoft show

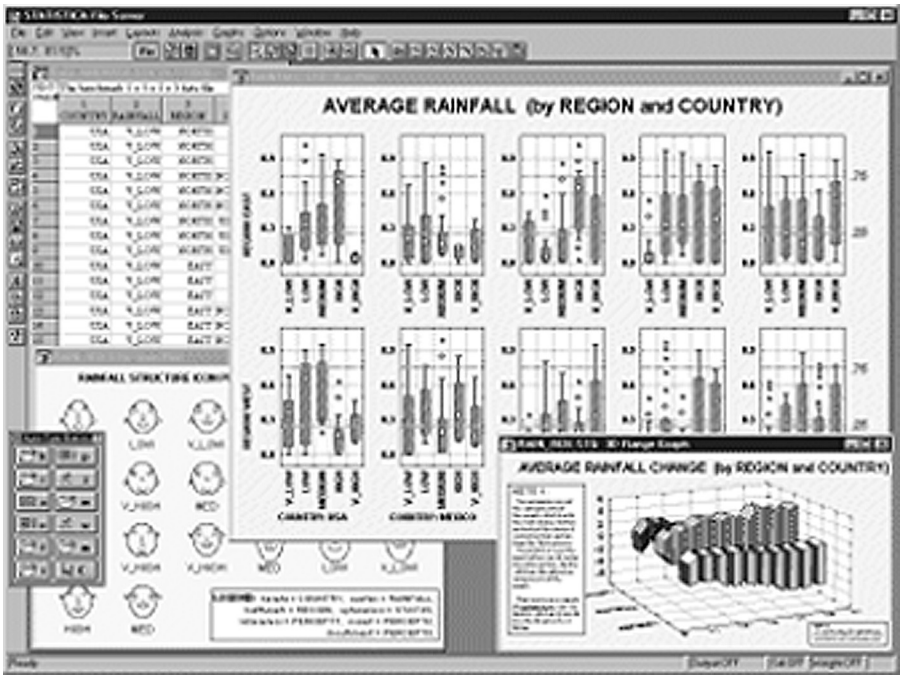

Figure 13.4 Boxplots showing multiple variable combinations in StatSoft. *Source: www.statsoft.com.*

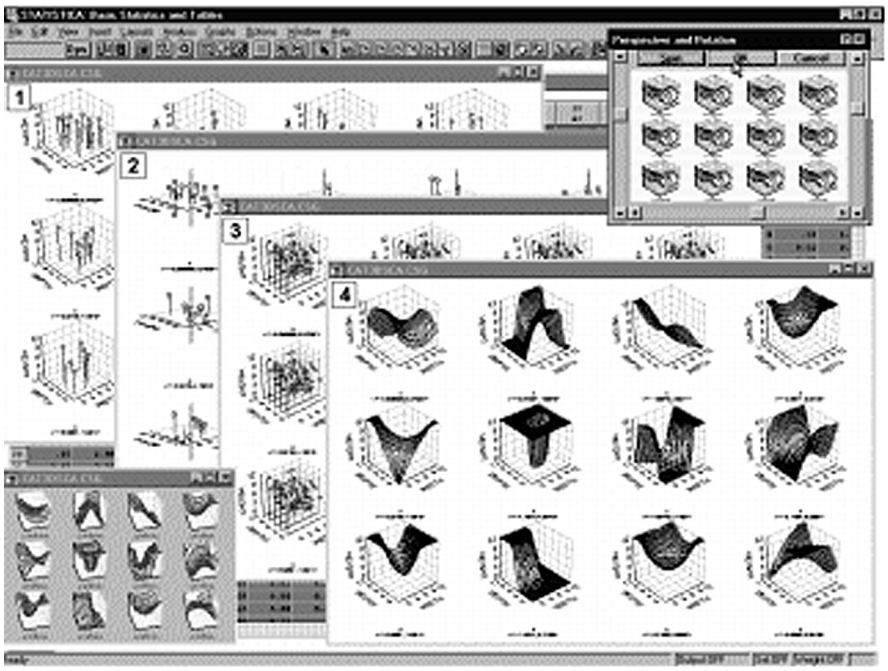

Figure 13.5 Multidimensional data distribution analysis in StatSoft. *Source: www.statsoft.com.*

data distributions in multidimensional space. Visual display can help give users a clear impression and overview of the data characteristics in a large data set.

- **Data mining result visualization**: Visualization of data mining results is the presentation of the results or knowledge obtained from data mining in visual forms. Such forms may include scatter plots and boxplots (Chapter 2), as well as decision trees, association rules, clusters, outliers, and generalized rules. For example, scatter plots are shown in Figure 13.6 from SAS Enterprise Miner. Figure 13.7, from MineSet, uses a plane associated with a set of pillars to describe a set of association rules mined from a database. Figure 13.8, also from MineSet, presents a decision tree. Figure 13.9, from IBM Intelligent Miner, presents a set of clusters and the properties associated with them.
- **Data mining process visualization**: This type of visualization presents the various processes of data mining in visual forms so that users can see how the data are extracted and from which database or data warehouse they are extracted, as well as how the selected data are cleaned, integrated, preprocessed, and mined. Moreover, it may also show which method is selected for data mining, where the results are stored, and how they may be viewed. Figure 13.10 shows a visual presentation of data mining processes by the Clementine data mining system.

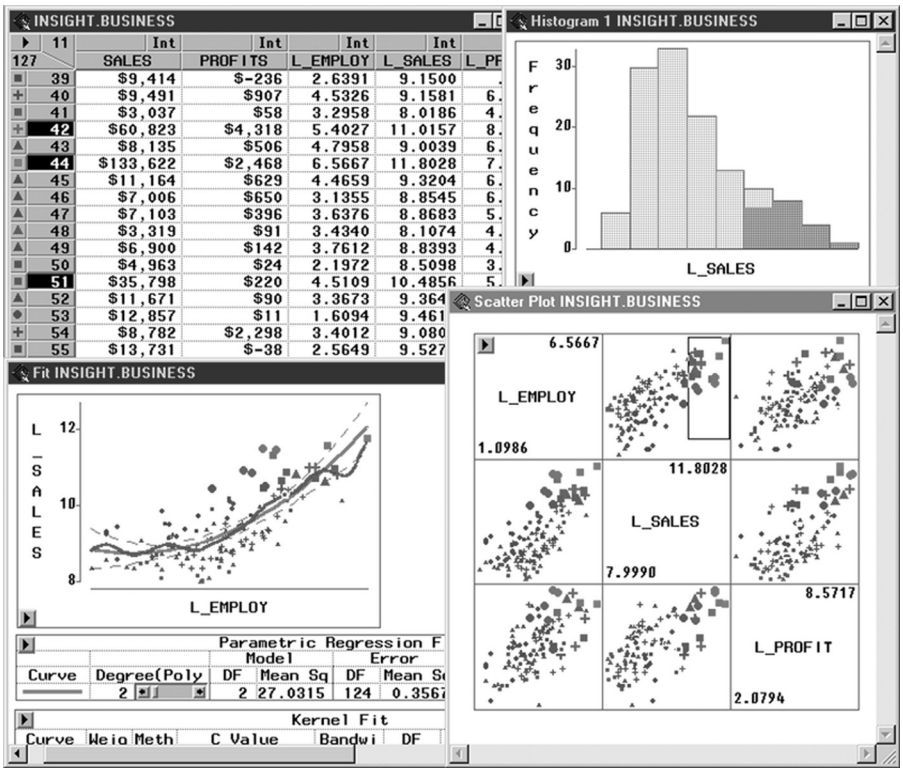

Figure 13.6 Visualization of data mining results in SAS Enterprise Miner.

- **Interactive visual data mining**: In (interactive) visual data mining, visualization tools can be used in the data mining process to help users make smart data mining decisions. For example, the data distribution in a set of attributes can be displayed using colored sectors (where the whole space is represented by a circle). This display helps users determine which sector should first be selected for classification and where a good split point for this sector may be. An example of this is shown in Figure 13.11, which is the output of a perception-based classification (PBC) system developed at the University of Munich.

Audio data mining uses audio signals to indicate the patterns of data or the features of data mining results. Although visual data mining may disclose interesting patterns using graphical displays, it requires users to concentrate on watching patterns and identifying interesting or novel features within them. This can sometimes be quite tiresome. If patterns can be transformed into sound and music, then instead of watching pictures, we can listen to pitchs, rhythm, tune, and melody to identify anything interesting or unusual. This may relieve some of the burden of visual concentration and be more

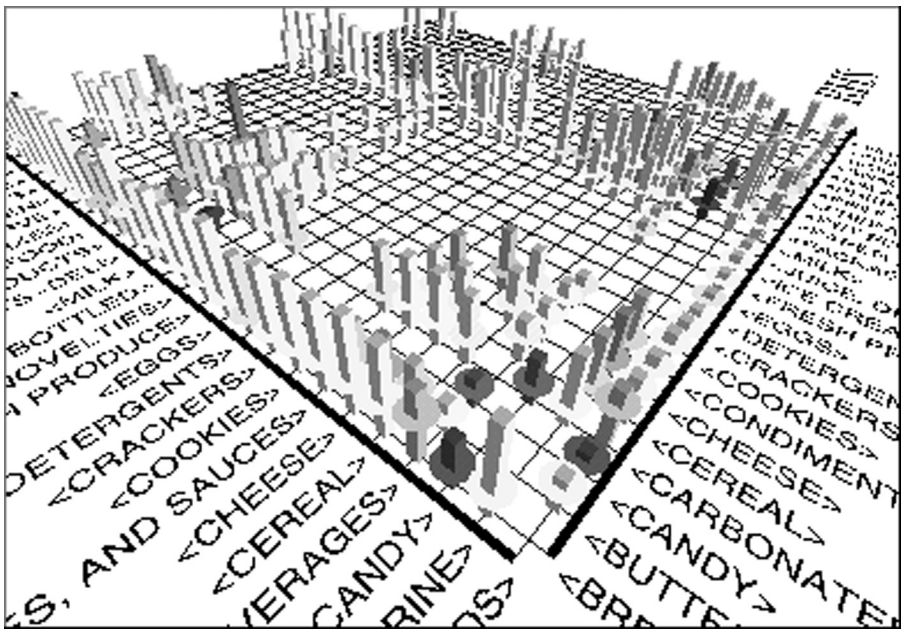

Figure 13.7 Visualization of association rules in MineSet.

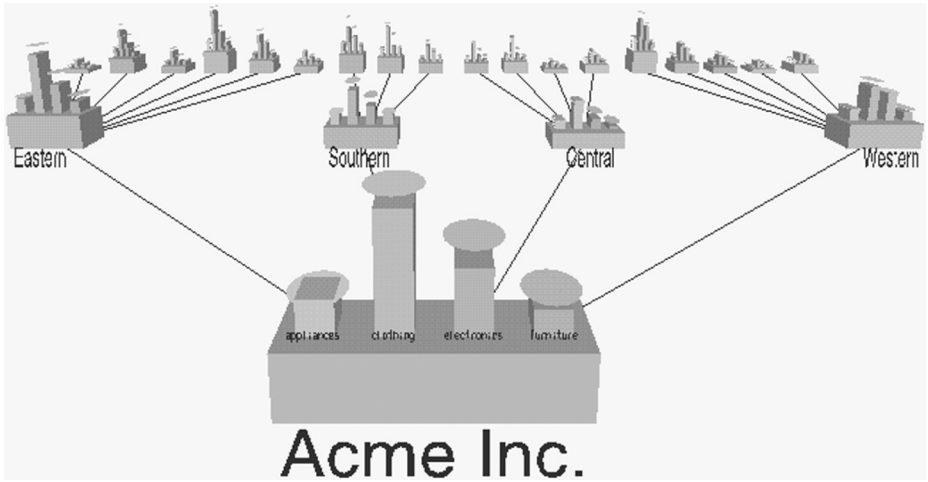

Figure 13.8 Visualization of a decision tree in MineSet.

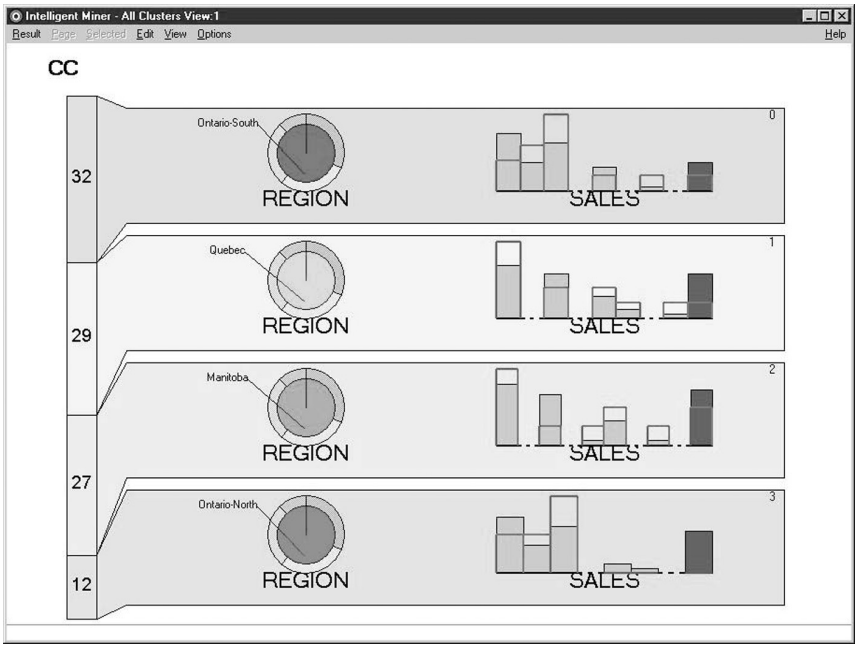

Figure 13.9 Visualization of cluster groupings in IBM Intelligent Miner.

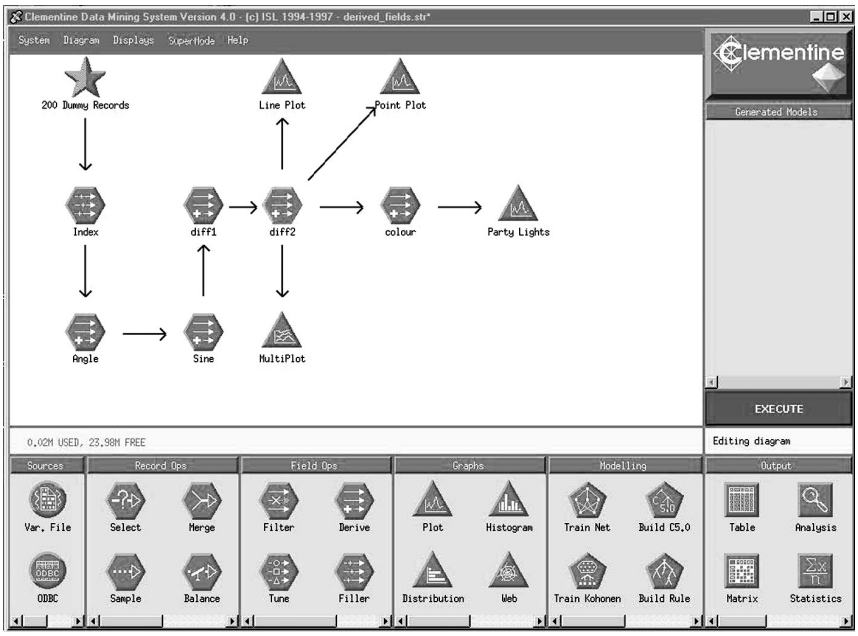

Figure 13.10 Visualization of data mining processes by Clementine.

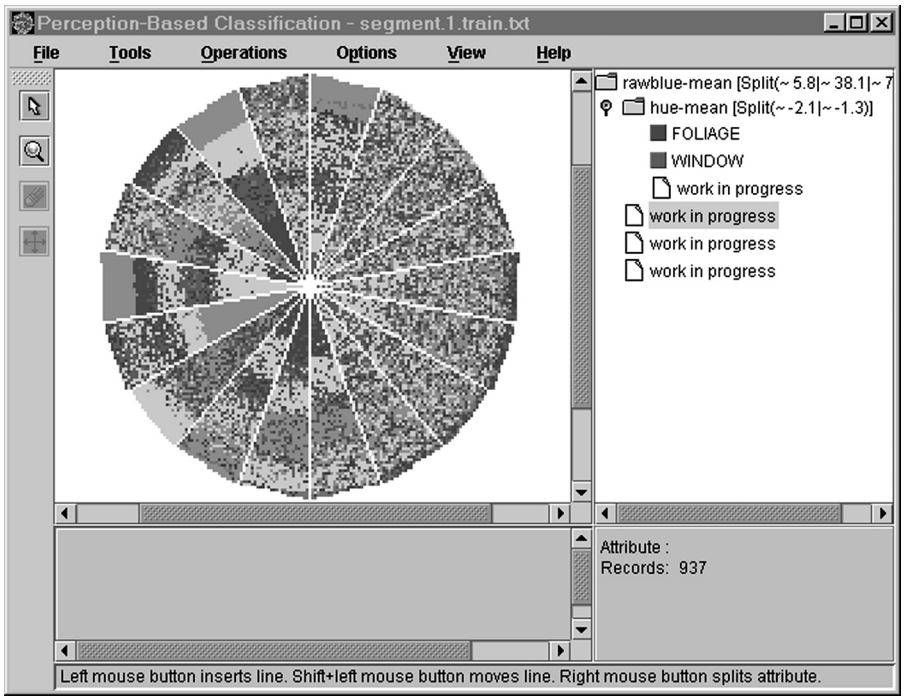

Figure 13.11 Perception-based classification, an interactive visual mining approach.

relaxing than visual mining. Therefore, audio data mining is an interesting complement to visual mining.

13.3 Data Mining Applications

In this book, we have studied principles and methods for mining relational data, data warehouses, and complex data types. Because data mining is a relatively young discipline with wide and diverse applications, there is still a nontrivial gap between general principles of data mining and application-specific, effective data mining tools. In this section, we examine several application domains, as listed in Figure 13.12. We discuss how customized data mining methods and tools should be developed for such applications.

13.3.1 Data Mining for Financial Data Analysis

Most banks and financial institutions offer a wide variety of banking, investment, and credit services (the latter include business, mortgage, and automobile loans and credit cards). Some also offer insurance and stock investment services.

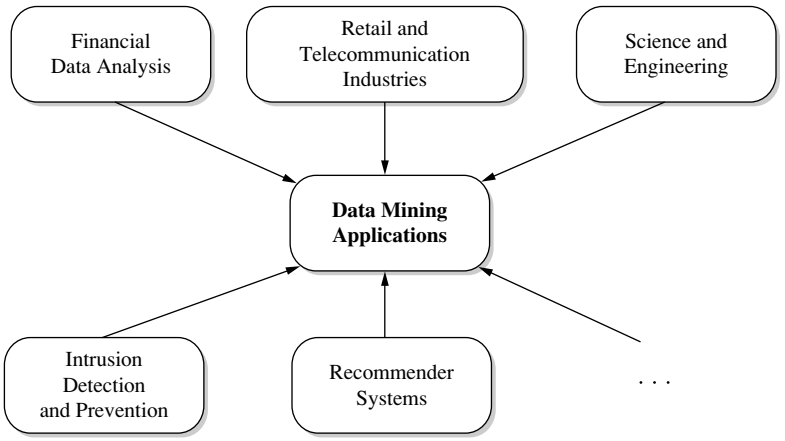

Figure 13.12 Common data mining application domains.

Financial data collected in the banking and financial industry are often relatively complete, reliable, and of high quality, which facilitates systematic data analysis and data mining. Here we present a few typical cases.

- **Design and construction of data warehouses for multidimensional data analysis and data mining:** Like many other applications, data warehouses need to be constructed for banking and financial data. Multidimensional data analysis methods should be used to analyze the general properties of such data. For example, a company's financial officer may want to view the debt and revenue changes by month, region, and sector, and other factors, along with maximum, minimum, total, average, trend, deviation, and other statistical information. Data warehouses, data cubes (including advanced data cube concepts such as multifeature, discovery-driven, regression, and prediction data cubes), characterization and class comparisons, clustering, and outlier analysis will all play important roles in financial data analysis and mining.
- **Loan payment prediction and customer credit policy analysis:** Loan payment prediction and customer credit analysis are critical to the business of a bank. Many factors can strongly or weakly influence loan payment performance and customer credit rating. Data mining methods, such as attribute selection and attribute relevance ranking, may help identify important factors and eliminate irrelevant ones. For example, factors related to the risk of loan payments include loan-to-value ratio, term of the loan, debt ratio (total amount of monthly debt versus total monthly income), payment-to-income ratio, customer income level, education level, residence region, and credit history. Analysis of the customer payment history may find that, say, payment-to-income ratio is a dominant factor, while education level and debt ratio are not. The bank may then decide to adjust its loan-granting policy so

as to grant loans to those customers whose applications were previously denied but whose profiles show relatively low risks according to the critical factor analysis.

- **Classification and clustering of customers for targeted marketing:** Classification and clustering methods can be used for customer group identification and targeted marketing. For example, we can use classification to identify the most crucial factors that may influence a customer's decision regarding banking. Customers with similar behaviors regarding loan payments may be identified by multidimensional clustering techniques. These can help identify customer groups, associate a new customer with an appropriate customer group, and facilitate targeted marketing.
- **Detection of money laundering and other financial crimes:** To detect money laundering and other financial crimes, it is important to integrate information from multiple, heterogeneous databases (e.g., bank transaction databases and federal or state crime history databases), as long as they are potentially related to the study. Multiple data analysis tools can then be used to detect unusual patterns, such as large amounts of cash flow at certain periods, by certain groups of customers. Useful tools include data visualization tools (to display transaction activities using graphs by time and by groups of customers), linkage and information network analysis tools (to identify links among different customers and activities), classification tools (to filter unrelated attributes and rank the highly related ones), clustering tools (to group different cases), outlier analysis tools (to detect unusual amounts of fund transfers or other activities), and sequential pattern analysis tools (to characterize unusual access sequences). These tools may identify important relationships and patterns of activities and help investigators focus on suspicious cases for further detailed examination.

13.3.2 Data Mining for Retail and Telecommunication Industries

The retail industry is a well-fit application area for data mining, since it collects huge amounts of data on sales, customer shopping history, goods transportation, consumption, and service. The quantity of data collected continues to expand rapidly, especially due to the increasing availability, ease, and popularity of business conducted on the Web, or **e-commerce**. Today, most major chain stores also have web sites where customers can make purchases online. Some businesses, such as Amazon.com (*www.amazon.com*), exist solely online, without any brick-and-mortar (i.e., physical) store locations. Retail data provide a rich source for data mining.

Retail data mining can help identify customer buying behaviors, discover customer shopping patterns and trends, improve the quality of customer service, achieve better customer retention and satisfaction, enhance goods consumption ratios, design more effective goods transportation and distribution policies, and reduce the cost of business.

A few examples of data mining in the retail industry are outlined as follows:

- **Design and construction of data warehouses:** Because retail data cover a wide spectrum (including sales, customers, employees, goods transportation, consumption,

and services), there can be many ways to design a data warehouse for this industry. The levels of detail to include can vary substantially. The outcome of preliminary data mining exercises can be used to help guide the design and development of data warehouse structures. This involves deciding which dimensions and levels to include and what preprocessing to perform to facilitate effective data mining.

- **Multidimensional analysis of sales, customers, products, time, and region:** The retail industry requires timely information regarding customer needs, product sales, trends, and fashions, as well as the quality, cost, profit, and service of commodities. It is therefore important to provide powerful multidimensional analysis and visualization tools, including the construction of sophisticated data cubes according to the needs of data analysis. The *advanced data cube structures* introduced in Chapter 5 are useful in retail data analysis because they facilitate analysis on multidimensional aggregates with complex conditions.
- **Analysis of the effectiveness of sales campaigns:** The retail industry conducts sales campaigns using advertisements, coupons, and various kinds of discounts and bonuses to promote products and attract customers. Careful analysis of the effectiveness of sales campaigns can help improve company profits. Multidimensional analysis can be used for this purpose by comparing the amount of sales and the number of transactions containing the sales items during the sales period versus those containing the same items before or after the sales campaign. Moreover, association analysis may disclose which items are likely to be purchased together with the items on sale, especially in comparison with the sales before or after the campaign.
- **Customer retention—analysis of customer loyalty:** We can use customer loyalty card information to register sequences of purchases of particular customers. Customer loyalty and purchase trends can be analyzed systematically. Goods purchased at different periods by the same customers can be grouped into sequences. Sequential pattern mining can then be used to investigate changes in customer consumption or loyalty and suggest adjustments on the pricing and variety of goods to help retain customers and attract new ones.
- **Product recommendation and cross-referencing of items:** By mining associations from sales records, we may discover that a customer who buys a digital camera is likely to buy another set of items. Such information can be used to form product recommendations. *Collaborative recommender systems* (Section 13.3.5) use data mining techniques to make personalized product recommendations during live customer transactions, based on the opinions of other customers. Product recommendations can also be advertised on sales receipts, in weekly flyers, or on the Web to help improve customer service, aid customers in selecting items, and increase sales. Similarly, information, such as "hot items this week" or attractive deals, can be displayed together with the associative information to promote sales.
- **Fraudulent analysis and the identification of unusual patterns:** Fraudulent activity costs the retail industry millions of dollars per year. It is important to (1) identify potentially fraudulent users and their atypical usage patterns; (2) detect attempts to gain fraudulent entry or unauthorized access to individual and organizational

accounts; and (3) discover unusual patterns that may need special attention. Many of these patterns can be discovered by multidimensional analysis, cluster analysis, and outlier analysis.

As another industry that handles huge amounts of data, the **telecommunication industry** has quickly evolved from offering local and long-distance telephone services to providing many other comprehensive communication services. These include cellular phone, smart phone, Internet access, email, text messages, images, computer and web data transmissions, and other data traffic. The integration of telecommunication, computer network, Internet, and numerous other means of communication and computing has been under way, changing the face of telecommunications and computing. This has created a great demand for data mining to help understand business dynamics, identify telecommunication patterns, catch fraudulent activities, make better use of resources, and improve service quality.

Data mining tasks in telecommunications share many similarities with those in the retail industry. Common tasks include constructing large-scale data warehouses, performing multidimensional visualization, OLAP, and in-depth analysis of trends, customer patterns, and sequential patterns. Such tasks contribute to business improvements, cost reduction, customer retention, fraud analysis, and sharpening the edges of competition. There are many data mining tasks for which customized data mining tools for telecommunication have been flourishing and are expected to play increasingly important roles in business.

Data mining has been popularly used in many other industries, such as *insurance, manufacturing, and health care*, as well as for the *analysis of governmental and institutional administration data*. Although each industry has its own characteristic data sets and application demands, they share many common principles and methodologies. Therefore, through effective mining in one industry, we may gain experience and methodologies that can be transferred to other industrial applications.

13.3.3 Data Mining in Science and Engineering

In the past, many scientific data analysis tasks tended to handle relatively small and homogeneous data sets. Such data were typically analyzed using a "*formulate hypothesis, build model, and evaluate results*" paradigm. In these cases, statistical techniques were typically employed for their analysis (see Section 13.2.1). Massive data collection and storage technologies have recently changed the landscape of scientific data analysis. Today, scientific data can be amassed at much higher speeds and lower costs. This has resulted in the accumulation of huge volumes of high-dimensional data, stream data, and heterogenous data, containing rich spatial and temporal information. Consequently, scientific applications are shifting from the "*hypothesize-and-test*" paradigm toward a "*collect and store data, mine for new hypotheses, confirm with data or experimentation*" process. This shift brings about new challenges for data mining.

Vast amounts of data have been collected from scientific domains (including geosciences, astronomy, meteorology, geology, and biological sciences) using sophisticated

telescopes, multispectral high-resolution remote satellite sensors, global positioning systems, and new generations of biological data collection and analysis technologies. Large data sets are also being generated due to fast numeric simulations in various fields such as climate and ecosystem modeling, chemical engineering, fluid dynamics, and structural mechanics. Here we look at some of the challenges brought about by emerging scientific applications of data mining.

- **Data warehouses and data preprocessing**: Data preprocessing and data warehouses are critical for information exchange and data mining. Creating a warehouse often requires finding means for resolving inconsistent or incompatible data collected in multiple environments and at different time periods. This requires reconciling semantics, referencing systems, geometry, measurements, accuracy, and precision. Methods are needed for integrating data from heterogeneous sources and for identifying events.

  For instance, consider climate and ecosystem data, which are spatial and temporal and require cross-referencing geospatial data. A major problem in analyzing such data is that there are too many events in the spatial domain but too few in the temporal domain. For example, El Nino events occur only every four to seven years, and previous data on them might not have been collected as systematically as they are today. Methods are also needed for the efficient computation of sophisticated spatial aggregates and the handling of spatial-related data streams.

- **Mining complex data types**: Scientific data sets are heterogeneous in nature. They typically involve semi-structured and unstructured data, such as multimedia data and georeferenced stream data, as well as data with sophisticated, deeply hidden semantics (e.g., genomic and proteomic data). Robust and dedicated analysis methods are needed for handling spatiotemporal data, biological data, related concept hierarchies, and complex semantic relationships. For example, in bioinformatics, a research problem is to identify regulatory influences on genes. *Gene regulation* refers to how genes in a cell are switched on (or off) to determine the cell's functions. Different biological processes involve different sets of genes acting together in precisely regulated patterns. Thus, to understand a biological process we need to identify the participating genes and their regulators. This requires the development of sophisticated data mining methods to analyze large biological data sets for clues about regulatory influences on specific genes, by finding DNA segments ("regulatory sequences") mediating such influence.

- **Graph-based and network-based mining**: It is often difficult or impossible to model several physical phenomena and processes due to limitations of existing modeling approaches. Alternatively, labeled graphs and networks may be used to capture many of the spatial, topological, geometric, biological, and other relational characteristics present in scientific data sets. In graph or network modeling, each object to be mined is represented by a vertex in a graph, and edges between vertices represent relationships between objects. For example, graphs can be used to model chemical structures, biological pathways, and data generated by numeric

simulations such as fluid-flow simulations. The success of graph or network modeling, however, depends on improvements in the scalability and efficiency of many graph-based data mining tasks such as classification, frequent pattern mining, and clustering.

- **Visualization tools and domain-specific knowledge**: High-level graphical user interfaces and visualization tools are required for scientific data mining systems. These should be integrated with existing domain-specific data and information systems to guide researchers and general users in searching for patterns, interpreting and visualizing discovered patterns, and using discovered knowledge in their decision making.

Data mining in engineering shares many similarities with data mining in science. Both practices often collect massive amounts of data, and require data preprocessing, data warehousing, and scalable mining of complex types of data. Both typically use visualization and make good use of graphs and networks. Moreover, many engineering processes need real-time responses, and so mining data streams in real time often becomes a critical component.

Massive amounts of human communication data pour into our daily life. Such communication exists in many forms, including news, blogs, articles, web pages, online discussions, product reviews, twitters, messages, advertisements, and communications, both on the Web and in various kinds of social networks. Hence, **data mining in social science and social studies** has become increasingly popular. Moreover, user or reader feedback regarding products, speeches, and articles can be analyzed to deduce general opinions and sentiments on the views of those in society. The analysis results can be used to predict trends, improve work, and help in decision making.

Computer science generates unique kinds of data. For example, computer programs can be long, and their execution often generates huge-size traces. Computer networks can have complex structures and the network flows can be dynamic and massive. Sensor networks may generate large amounts of data with varied reliability. Computer systems and databases can suffer from various kinds of attacks, and their system/data accessing may raise security and privacy concerns. These unique kinds of data provide fertile land for data mining.

Data mining in computer science can be used to help monitor system status, improve system performance, isolate software bugs, detect software plagiarism, analyze computer system faults, uncover network intrusions, and recognize system malfunctions. Data mining for software and system engineering can operate on static or dynamic (i.e., stream-based) data, depending on whether the system dumps traces beforehand for postanalysis or if it must react in real time to handle online data.

Various methods have been developed in this domain, which integrate and extend methods from machine learning, data mining, software/system engineering, pattern recognition, and statistics. Data mining in computer science is an active and rich domain for data miners because of its unique challenges. It requires the further development of sophisticated, scalable, and real-time data mining and software/system engineering methods.

13.3.4 Data Mining for Intrusion Detection and Prevention

The security of our computer systems and data is at continual risk. The extensive growth of the Internet and the increasing availability of tools and tricks for intruding and attacking networks have prompted **intrusion detection and prevention** to become a critical component of networked systems. An intrusion can be defined as any set of actions that threaten the integrity, confidentiality, or availability of a network resource (e.g., user accounts, file systems, system kernels, and so on). Intrusion detection systems and intrusion prevention systems both monitor network traffic and/or system executions for malicious activities. However, the former produces reports whereas the latter is placed in-line and is able to actively prevent/block intrusions that are detected. The main functions of an intrusion prevention system are to identify malicious activity, log information about said activity, attempt to block/stop activity, and report activity.

The majority of intrusion detection and prevention systems use either *signature-based detection* or *anomaly-based detection*.

- **Signature-based detection**: This method of detection utilizes *signatures*, which are attack patterns that are preconfigured and predetermined by domain experts. A signature-based intrusion prevention system monitors the network traffic for matches to these signatures. Once a match is found, the intrusion detection system will report the anomaly and an intrusion prevention system will take additional appropriate actions. Note that since the systems are usually quite dynamic, the signatures need to be updated laboriously whenever new software versions arrive or changes in network configuration or other situations occur. Another drawback is that such a detection mechanism can only identify cases that match the signatures. That is, it is unable to detect new or previously unknown intrusion tricks.
- **Anomaly-based detection**: This method builds models of normal network behavior (called *profiles*) that are then used to detect new patterns that significantly deviate from the profiles. Such deviations may represent actual intrusions or simply be new behaviors that need to be added to the profiles. The main advantage of anomaly detection is that it may detect novel intrusions that have not yet been observed. Typically, a human analyst must sort through the deviations to ascertain which represent real intrusions. A limiting factor of anomaly detection is the high percentage of false positives. New patterns of intrusion can be added to the set of signatures to enhance signature-based detection.

Data mining methods can help an intrusion detection and prevention system to enhance its performance in various ways as follows.

- **New data mining algorithms for intrusion detection**: Data mining algorithms can be used for both signature-based and anomaly-based detection. In signature-based detection, training data are labeled as either "normal" or "intrusion." A classifier can then be derived to detect known intrusions. Research in this area has

included the application of classification algorithms, association rule mining, and cost-sensitive modeling. Anomaly-based detection builds models of normal behavior and automatically detects significant deviations from it. Methods include the application of clustering, outlier analysis, and classification algorithms and statistical approaches. The techniques used must be efficient and scalable, and capable of handling network data of high volume, dimensionality, and heterogeneity.

- **Association, correlation, and discriminative pattern analyses help select and build discriminative classifiers**: Association, correlation, and discriminative pattern mining can be applied to find relationships between system attributes describing the network data. Such information can provide insight regarding the selection of useful attributes for intrusion detection. New attributes derived from aggregated data may also be helpful such as summary counts of traffic matching a particular pattern.
- **Analysis of stream data**: Due to the transient and dynamic nature of intrusions and malicious attacks, it is crucial to perform intrusion detection in the data stream environment. Moreover, an event may be normal on its own, but considered malicious if viewed as part of a sequence of events. Thus, it is necessary to study what sequences of events are frequently encountered together, find sequential patterns, and identify outliers. Other data mining methods for finding evolving clusters and building dynamic classification models in data streams are also necessary for real-time intrusion detection.
- **Distributed data mining**: Intrusions can be launched from several different locations and targeted to many different destinations. Distributed data mining methods may be used to analyze network data from several network locations to detect these distributed attacks.
- **Visualization and querying tools**: Visualization tools should be available for viewing any anomalous patterns detected. Such tools may include features for viewing associations, discriminative patterns, clusters, and outliers. Intrusion detection systems should also have a graphical user interface that allows security analysts to pose queries regarding the network data or intrusion detection results.

In summary, computer systems are at continual risk of breaks in security. Data mining technology can be used to develop strong intrusion detection and prevention systems, which may employ signature-based or anomaly-based detection.

13.3.5 Data Mining and Recommender Systems

Today's consumers are faced with millions of goods and services when shopping online. **Recommender systems** help consumers by making product recommendations that are likely to be of interest to the user such as books, CDs, movies, restaurants, online news articles, and other services. Recommender systems may use either a *content-based* approach, a *collaborative* approach, or a *hybrid* approach that combines both content-based and collaborative methods.

The **content-based approach** recommends items that are similar to items the user preferred or queried in the past. It relies on product features and textual item descriptions. The **collaborative approach** (or *collaborative filtering approach*) may consider a user's social environment. It recommends items based on the opinions of other customers who have similar tastes or preferences as the user. Recommender systems use a broad range of techniques from information retrieval, statistics, machine learning, and data mining to search for similarities among items and customer preferences. Consider Example 13.1.

Example 13.1 **Scenarios of using a recommender system.** Suppose that you visit the web site of an online bookstore (e.g., Amazon) with the intention of purchasing a book that you have been wanting to read. You type in the name of the book. This is not the first time you have visited the web site. You have browsed through it before and even made purchases from it last Christmas. The web store remembers your previous visits, having stored click stream information and information regarding your past purchases. The system displays the description and price of the book you have just specified. It compares your interests with other customers having similar interests and recommends additional book titles, saying *"Customers who bought the book you have specified also bought these other titles as well."* From surveying the list, you see another title that sparks your interest and decide to purchase that one as well.

Now suppose you go to another online store with the intention of purchasing a digital camera. The system suggests additional items to consider based on previously mined sequential patterns, such as *"Customers who buy this kind of digital camera are likely to buy a particular brand of printer, memory card, or photo editing software within three months."* You decide to buy just the camera, without any additional items. A week later, you receive coupons from the store regarding the additional items. ■

An advantage of recommender systems is that they provide *personalization* for customers of e-commerce, promoting one-to-one marketing. Amazon, a pioneer in the use of collaborative recommender systems, offers "a personalized store for every customer" as part of their marketing strategy. Personalization can benefit both consumers and the company involved. By having more accurate models of their customers, companies gain a better understanding of customer needs. Serving these needs can result in greater success regarding cross-selling of related products, upselling, product affinities, one-to-one promotions, larger baskets, and customer retention.

The recommendation problem considers a set, C, of users and a set, S, of items. Let u be a utility function that measures the usefulness of an item, s, to a user, c. The utility is commonly represented by a rating and is initially defined only for items previously rated by users. For example, when joining a movie recommendation system, users are typically asked to rate several movies. The space $C \times S$ of all possible users and items is huge. The recommendation system should be able to extrapolate from known to unknown ratings so as to predict item–user combinations. Items with the highest predicted rating/utility for a user are recommended to that user.

"*How is the utility of an item estimated for a user*?" In content-based methods, it is estimated based on the utilities assigned by the same user to other items that are similar. Many such systems focus on recommending items containing textual information, such as web sites, articles, and news messages. They look for commonalities among items. For movies, they may look for similar genres, directors, or actors. For articles, they may look for similar terms. Content-based methods are rooted in information theory. They make use of keywords (describing the items) and user profiles that contain information about users' tastes and needs. Such profiles may be obtained explicitly (e.g., through questionnaires) or learned from users' transactional behavior over time.

A collaborative recommender system tries to predict the utility of items for a user, u, based on items previously rated by other users who are similar to u. For example, when recommending books, a collaborative recommender system tries to find other users who have a history of agreeing with u (e.g., they tend to buy similar books, or give similar ratings for books). Collaborative recommender systems can be either memory (or heuristic) based or model based.

Memory-based methods essentially use heuristics to make rating predictions based on the entire collection of items previously rated by users. That is, the unknown rating of an item–user combination can be estimated as an aggregate of ratings of the most similar users for the same item. Typically, a k-nearest-neighbor approach is used, that is, we find the k other users (or neighbors) that are most similar to our target user, u. Various approaches can be used to compute the similarity between users. The most popular approaches use either Pearson's correlation coefficient (Section 3.3.2) or cosine similarity (Section 2.4.7). A weighted aggregate can be used, which adjusts for the fact that different users may use the rating scale differently. Model-based collaborative recommender systems use a collection of ratings to learn a model, which is then used to make rating predictions. For example, probabilistic models, clustering (which finds clusters of like-minded customers), Bayesian networks, and other machine learning techniques have been used.

Recommender systems face major challenges such as scalability and ensuring quality recommendations to the consumer. For example, regarding scalability, collaborative recommender systems must be able to search through millions of potential neighbors in real time. If the site is using browsing patterns as indications of product preference, it may have thousands of data points for some of its customers. Ensuring quality recommendations is essential to gain consumers' trust. If consumers follow a system recommendation but then do not end up liking the product, they are less likely to use the recommender system again.

As with classification systems, recommender systems can make two types of errors: false negatives and false positives. Here, *false negatives* are products that the system fails to recommend, although the consumer would like them. *False positives* are products that are recommended, but which the consumer does not like. False positives are less desirable because they can annoy or anger consumers. Content-based recommender systems are limited by the features used to describe the items they recommend.

Another challenge for both content-based and collaborative recommender systems is how to deal with new users for which a buying history is not yet available.

Hybrid approaches integrate both content-based and collaborative methods to achieve further improved recommendations. The Netflix Prize was an open competition held by an online DVD-rental service, with a payout of $1,000,000 for the best recommender algorithm to predict user ratings for films, based on previous ratings. The competition and other studies have shown that the predictive accuracy of a recommender system can be substantially improved when blending multiple predictors, especially by using an ensemble of many substantially different methods, rather than refining a single technique.

Collaborative recommender systems are a form of **intelligent query answering**, which consists of analyzing the intent of a query and providing generalized, neighborhood, or associated information relevant to the query. For example, rather than simply returning the book description and price in response to a customer's query, returning additional information that is related to the query but that was not explicitly asked for (e.g., book evaluation comments, recommendations of other books, or sales statistics) provides an intelligent answer to the same query.

13.4 Data Mining and Society

For most of us, data mining is part of our daily lives, although we may often be unaware of its presence. Section 13.4.1 looks at several examples of "ubiquitous and invisible" data mining, affecting everyday things from the products stocked at our local supermarket, to the ads we see while surfing the Internet, to crime prevention. Data mining can offer the individual many benefits by improving customer service and satisfaction as well as lifestyle, in general. However, it also has serious implications regarding one's right to privacy and data security. These issues are the topic of Section 13.4.2.

13.4.1 Ubiquitous and Invisible Data Mining

Data mining is present in many aspects of our daily lives, whether we realize it or not. It affects how we shop, work, and search for information, and can even influence our leisure time, health, and well-being. In this section, we look at examples of such **ubiquitous** (or ever-present) **data mining**. Several of these examples also represent **invisible data mining**, in which "smart" software, such as search engines, customer-adaptive web services (e.g., using recommender algorithms), "intelligent" database systems, email managers, ticket masters, and so on, incorporates data mining into its functional components, often unbeknownst to the user.

From grocery stores that print personalized coupons on customer receipts to online stores that recommend additional items based on customer interests, data mining has innovatively influenced what we buy, the way we shop, and our experience while shopping. One example is Wal-Mart, which has hundreds of millions of customers visiting its tens of thousands of stores every week. Wal-Mart allows suppliers to access data on

their products and perform analyses using data mining software. This allows suppliers to identify customer buying patterns at different stores, control inventory and product placement, and identify new merchandizing opportunities. All of these affect which items (and how many) end up on the stores' shelves—something to think about the next time you wander through the aisles at Wal-Mart.

Data mining has shaped the online shopping experience. Many shoppers routinely turn to online stores to purchase books, music, movies, and toys. Recommender systems, discussed in Section 13.3.5, offer personalized product recommendations based on the opinions of other customers. Amazon.com was at the forefront of using such a personalized, data mining–based approach as a marketing strategy. It has observed that in traditional brick-and-mortar stores, the hardest part is getting the customer into the store. Once the customer is there, he or she is likely to buy something, since the cost of going to another store is high. Therefore, the marketing for brick-and-mortar stores tends to emphasize drawing customers in, rather than the actual in-store customer experience. This is in contrast to online stores, where customers can "walk out" and enter another online store with just a click of the mouse. Amazon.com capitalized on this difference, offering a "personalized store for every customer." They use several data mining techniques to identify customer's likes and make reliable recommendations.

While we are on the topic of shopping, suppose you have been doing a lot of buying with your credit cards. Nowadays, it is not unusual to receive a phone call from one's credit card company regarding suspicious or unusual patterns of spending. Credit card companies use data mining to detect fraudulent usage, saving billions of dollars a year.

Many companies increasingly use data mining for **customer relationship management (CRM)**, which helps provide more customized, personal service addressing individual customer's needs, in lieu of mass marketing. By studying browsing and purchasing patterns on web stores, companies can tailor advertisements and promotions to customer profiles, so that customers are less likely to be annoyed with unwanted mass mailings or junk mail. These actions can result in substantial cost savings for companies. The customers further benefit in that they are more likely to be notified of offers that are actually of interest, resulting in less waste of personal time and greater satisfaction.

Data mining has greatly influenced the ways in which people use computers, search for information, and work. Once you get on the Internet, for example, you decide to check your email. Unbeknownst to you, several annoying emails have already been deleted, thanks to a spam filter that uses classification algorithms to recognize spam. After processing your email, you go to Google (*www.google.com*), which provides access to information from billions of web pages indexed on its server. Google is one of the most popular and widely used Internet search engines. Using Google to search for information has become a way of life for many people.

Google is so popular that it has even become a new verb in the English language, meaning "to search for (something) on the Internet using the Google search engine or, by extension, any comprehensive search engine."[1] You decide to type in some keywords

[1] *http://open-dictionary.com.*

for a topic of interest. Google returns a list of web sites on your topic, mined, indexed, and organized by a set of data mining algorithms including PageRank. Moreover, if you type "Boston New York," Google will show you bus and train schedules from Boston to New York; however, a minor change to "Boston Paris" will lead to flight schedules from Boston to Paris. Such smart offerings of information or services are likely based on the frequent patterns mined from the click streams of many previous queries.

While you are viewing the results of your Google query, various ads pop up relating to your query. Google's strategy of tailoring advertising to match the user's interests is one of the typical services being explored by every Internet search provider. This also makes you happier, because you are less likely to be pestered with irrelevant ads.

Data mining is omnipresent, as can be seen from these daily-encountered examples. We could go on and on with such scenarios. In many cases, data mining is invisible, as users may be unaware that they are examining results returned by data mining or that their clicks are actually fed as new data into some data mining functions. For data mining to become further improved and accepted as a technology, continuing research and development are needed in the many areas mentioned as challenges throughout this book. These include efficiency and scalability, increased user interaction, incorporation of background knowledge and visualization techniques, effective methods for finding interesting patterns, improved handling of complex data types and stream data, real-time data mining, web mining, and so on. In addition, the *integration* of data mining into existing business and scientific technologies, to provide domain-specific data mining tools, will further contribute to the advancement of the technology. The success of data mining solutions tailored for e-commerce applications, as opposed to generic data mining systems, is an example.

13.4.2 Privacy, Security, and Social Impacts of Data Mining

With more and more information accessible in electronic forms and available on the Web, and with increasingly powerful data mining tools being developed and put into use, there are increasing concerns that data mining may pose a threat to our privacy and data security. However, it is important to note that many data mining applications do not even touch personal data. Prominent examples include applications involving natural resources, the prediction of floods and droughts, meteorology, astronomy, geography, geology, biology, and other scientific and engineering data. Furthermore, most studies in data mining research focus on the development of scalable algorithms and do not involve personal data.

The focus of data mining technology is on the *discovery of general or statistically significant patterns*, not on specific information regarding individuals. In this sense, we believe that the real privacy concerns are with unconstrained access to individual records, especially access to privacy-sensitive information such as credit card transaction records, health-care records, personal financial records, biological traits, criminal/justice investigations, and ethnicity. For the data mining applications that do involve personal data, in many cases, simple methods such as removing sensitive IDs from data may protect the privacy of most individuals. Nevertheless, privacy concerns exist wherever

personally identifiable information is collected and stored in digital form, and data mining programs are able to access such data, even during data preparation.

Improper or nonexistent disclosure control can be the root cause of privacy issues. To handle such concerns, numerous data security-enhancing techniques have been developed. In addition, there has been a great deal of recent effort on developing *privacy-preserving* data mining methods. In this section, we look at some of the advances in protecting privacy and data security in data mining.

"*What can we do to secure the privacy of individuals while collecting and mining data?*" Many **data security–enhancing techniques** have been developed to help protect data. Databases can employ a *multilevel security model* to classify and restrict data according to various security levels, with users permitted access to only their authorized level. It has been shown, however, that users executing specific queries at their authorized security level can still infer more sensitive information, and that a similar possibility can occur through data mining. *Encryption* is another technique in which individual data items may be encoded. This may involve *blind signatures* (which build on public key encryption), *biometric encryption* (e.g., where the image of a person's iris or fingerprint is used to encode his or her personal information), and *anonymous databases* (which permit the consolidation of various databases but limit access to personal information only to those who need to know; personal information is encrypted and stored at different locations). Intrusion detection is another active area of research that helps protect the privacy of personal data.

Privacy-preserving data mining is an area of data mining research in response to privacy protection in data mining. It is also known as *privacy-enhanced* or *privacy-sensitive* data mining. It deals with obtaining valid data mining results without disclosing the underlying sensitive data values. Most privacy-preserving data mining methods use some form of transformation on the data to perform privacy preservation. Typically, such methods reduce the granularity of representation to preserve privacy. For example, they may generalize the data from individual customers to customer groups. This reduction in granularity causes loss of information and possibly of the usefulness of the data mining results. This is the natural trade-off between information loss and privacy. Privacy-preserving data mining methods can be classified into the following categories.

- **Randomization methods**: These methods add noise to the data to mask some attribute values of records. The noise added should be sufficiently large so that individual record values, especially sensitive ones, cannot be recovered. However, it should be added skillfully so that the final results of data mining are basically preserved. Techniques are designed to derive aggregate distributions from the perturbed data. Subsequently, data mining techniques can be developed to work with these aggregate distributions.
- **The *k*-anonymity and *l*-diversity methods**: Both of these methods alter individual records so that they cannot be uniquely identified. In the *k-anonymity method*, the granularity of data representation is reduced sufficiently so that any given record maps onto at least *k* other records in the data. It uses techniques like generalization and suppression. The *k*-anonymity method is weak in that, if there is a homogeneity

of sensitive values within a group, then those values may be inferred for the altered records. The *l-diversity model* was designed to handle this weakness by enforcing intragroup diversity of sensitive values to ensure anonymization. The goal is to make it sufficiently difficult for adversaries to use combinations of record attributes to exactly identify individual records.

- **Distributed privacy preservation**: Large data sets could be partitioned and distributed either *horizontally* (i.e., the data sets are partitioned into different subsets of records and distributed across multiple sites) or *vertically* (i.e., the data sets are partitioned and distributed by their attributes), or even in a combination of both. While the individual sites may not want to share their entire data sets, they may consent to limited information sharing with the use of a variety of protocols. The overall effect of such methods is to maintain privacy for each individual object, while deriving aggregate results over all of the data.
- **Downgrading the effectiveness of data mining results**: In many cases, even though the data may not be available, the output of data mining (e.g, association rules and classification models) may result in violations of privacy. The solution could be to downgrade the effectiveness of data mining by either modifying data or mining results, such as hiding some association rules or slightly distorting some classification models.

Recently, researchers proposed new ideas in privacy-preserving data mining such as the notion of **differential privacy**. The general idea is that, for any two data sets that are close to one another (i.e., that differ only on a tiny data set such as a single element), a given *differentially private algorithm* will behave approximately the same on both data sets. This definition gives a strong guarantee that the presence or absence of a tiny data set (e.g., representing an individual) will not affect the final output of the query significantly. Based on this notion, a set of differential privacy-preserving data mining algorithms have been developed. Research in this direction is ongoing. We expect more powerful privacy-preserving data publishing and data mining algorithms in the near future.

Like any other technology, data mining can be misused. However, we must not lose sight of all the benefits that data mining research can bring, ranging from insights gained from medical and scientific applications to increased customer satisfaction by helping companies better suit their clients' needs. We expect that computer scientists, policy experts, and counterterrorism experts will continue to work with social scientists, lawyers, companies, and consumers to take responsibility in building solutions to ensure data privacy protection and security. In this way, we may continue to reap the benefits of data mining in terms of time and money savings and the discovery of new knowledge.

13.5 Data Mining Trends

The diversity of data, data mining tasks, and data mining approaches poses many challenging research issues in data mining. The development of efficient and effective data

mining methods, systems and services, and interactive and integrated data mining environments is a key area of study. The use of data mining techniques to solve large or sophisticated application problems is an important task for data mining researchers and data mining system and application developers. This section describes some of the trends in data mining that reflect the pursuit of these challenges.

- **Application exploration:** Early data mining applications put a lot of effort into helping businesses gain a competitive edge. The exploration of data mining for businesses continues to expand as e-commerce and e-marketing have become mainstream in the retail industry. Data mining is increasingly used for the exploration of applications in other areas such as web and text analysis, financial analysis, industry, government, biomedicine, and science. Emerging application areas include data mining for counterterrorism and mobile (wireless) data mining. Because generic data mining systems may have limitations in dealing with application-specific problems, we may see a trend toward the development of more application-specific data mining systems and tools, as well as invisible data mining functions embedded in various kinds of services.
- **Scalable and interactive data mining methods:** In contrast with traditional data analysis methods, data mining must be able to handle huge amounts of data efficiently and, if possible, interactively. Because the amount of data being collected continues to increase rapidly, scalable algorithms for individual and integrated data mining functions become essential. One important direction toward improving the overall efficiency of the mining process while increasing user interaction is **constraint-based mining**. This provides users with added control by allowing the specification and use of constraints to guide data mining systems in their search for interesting patterns and knowledge.
- **Integration of data mining with search engines, database systems, data warehouse systems, and cloud computing systems:** Search engines, database systems, data warehouse systems, and cloud computing systems are mainstream information processing and computing systems. It is important to ensure that data mining serves as an essential data analysis component that can be smoothly integrated into such an information processing environment. A data mining subsystem/service should be tightly coupled with such systems as a seamless, unified framework or as an invisible function. This will ensure data availability, data mining portability, scalability, high performance, and an integrated information processing environment for multidimensional data analysis and exploration.
- **Mining social and information networks:** Mining social and information networks and link analysis are critical tasks because such networks are ubiquitous and complex. The development of scalable and effective knowledge discovery methods and applications for large numbers of network data is essential, as outlined in Section 13.1.2.
- **Mining spatiotemporal, moving-objects, and cyber-physical systems:** Cyber-physical systems as well as spatiotemporal data are mounting rapidly due to the

popular use of cellular phones, GPS, sensors, and other wireless equipment. As outlined in Section 13.1.3, there are many challenging research issues realizing real-time and effective knowledge discovery with such data.

- **Mining multimedia, text, and web data:** As outlined in Section 13.1.3, mining such kinds of data is a recent focus in data mining research. Great progress has been made, yet there are still many open issues to be solved.
- **Mining biological and biomedical data:** The unique combination of complexity, richness, size, and importance of biological and biomedical data warrants special attention in data mining. Mining DNA and protein sequences, mining high-dimensional microarray data, and biological pathway and network analysis are just a few topics in this field. Other areas of biological data mining research include mining biomedical literature, link analysis across heterogeneous biological data, and information integration of biological data by data mining.
- **Data mining with software engineering and system engineering:** Software programs and large computer systems have become increasingly bulky in size sophisticated in complexity, and tend to originate from the integration of multiple components developed by different implementation teams. This trend has made it an increasingly challenging task to ensure software robustness and reliability. The analysis of the executions of a buggy software program is essentially a data mining process—tracing the data generated during program executions may disclose important patterns and outliers that could lead to the eventual automated discovery of software bugs. We expect that the further development of data mining methodologies for software/system debugging will enhance software robustness and bring new vigor to software/system engineering.
- **Visual and audio data mining:** Visual and audio data mining is an effective way to integrate with humans' visual and audio systems and discover knowledge from huge amounts of data. A systematic development of such techniques will facilitate the promotion of human participation for effective and efficient data analysis.
- **Distributed data mining and real-time data stream mining:** Traditional data mining methods, designed to work at a centralized location, do not work well in many of the distributed computing environments present today (e.g., the Internet, intranets, local area networks, high-speed wireless networks, sensor networks, and cloud computing). Advances in distributed data mining methods are expected. Moreover, many applications involving stream data (e.g., e-commerce, Web mining, stock analysis, intrusion detection, mobile data mining, and data mining for counterterrorism) require dynamic data mining models to be built in real time. Additional research is needed in this direction.
- **Privacy protection and information security in data mining:** An abundance of personal or confidential information available in electronic forms, coupled with increasingly powerful data mining tools, poses a threat to data privacy and security. Growing interest in data mining for counterterrorism also adds to the concern.

Further development of privacy-preserving data mining methods is foreseen. The collaboration of technologists, social scientists, law experts, governments, and companies is needed to produce a rigorous privacy and security protection mechanism for data publishing and data mining.

With confidence, we look forward to the next generation of data mining technology and the further benefits that it will bring.

13.6 Summary

- Mining complex data types poses challenging issues, for which there are many dedicated lines of research and development. This chapter presents a high-level overview of **mining complex data types**, which includes *mining sequence data* such as time series, symbolic sequences, and biological sequences; *mining graphs and networks*; and mining other kinds of data, including *spatiotemporal and cyber-physical system data*, *multimedia, text and Web data*, and *data streams*.
- Several well-established **statistical methods** have been proposed for data analysis such as regression, generalized linear models, analysis of variance, mixed-effect models, factor analysis, discriminant analysis, survival analysis, and quality control. Full coverage of statistical data analysis methods is beyond the scope of this book. Interested readers are referred to the statistical literature cited in the bibliographic notes (Section 13.8).
- Researchers have been striving to build **theoretical foundations** for data mining. Several interesting proposals have appeared, based on data reduction, data compression, probability and statistics theory, microeconomic theory, and pattern discovery–based inductive databases.
- **Visual data mining** integrates data mining and data visualization to discover implicit and useful knowledge from large data sets. Visual data mining includes *data visualization*, *data mining result visualization*, *data mining process visualization,* and *interactive visual data mining*. **Audio data mining** uses audio signals to indicate data patterns or features of data mining results.
- Many customized data mining tools have been developed for **domain-specific applications**, including finance, the retail and telecommunication industries, science and engineering, intrusion detection and prevention, and recommender systems. Such application domain-based studies integrate domain-specific knowledge with data analysis techniques and provide mission-specific data mining solutions.
- **Ubiquitous data mining** is the constant presence of data mining in many aspects of our daily lives. It can influence how we shop, work, search for information, and use a computer, as well as our leisure time, health, and well-being. In **invisible data mining**, "smart" software, such as search engines, customer-adaptive web services

(e.g., using recommender algorithms), email managers, and so on, incorporates data mining into its functional components, often unbeknownst to the user.

- A major social concern of data mining is the issue of *privacy and data security*. **Privacy-preserving data mining** deals with obtaining valid data mining results without disclosing underlying sensitive values. Its goal is to ensure privacy protection and security while preserving the overall quality of data mining results.
- **Data mining trends** include further efforts toward the exploration of new application areas; improved scalable, interactive, and constraint-based mining methods; the integration of data mining with web service, database, warehousing, and cloud computing systems; and mining social and information networks. Other trends include the mining of spatiotemporal and cyber-physical system data, biological data, software/system engineering data, and multimedia and text data, in addition to web mining, distributed and real-time data stream mining, visual and audio mining, and privacy and security in data mining.

13.7 Exercises

13.1 Sequence data are ubiquitous and have diverse applications. This chapter presented a general overview of sequential pattern mining, sequence classification, sequence similarity search, trend analysis, biological sequence alignment, and modeling. However, we have not covered sequence clustering. Present an overview of methods for *sequence clustering*.

13.2 This chapter presented an overview of sequence pattern mining and graph pattern mining methods. Mining tree patterns and partial order patterns is also studied in research. *Summarize the methods for mining structured patterns*, including sequences, trees, graphs, and partial order relationships. Examine what kinds of structural pattern mining have not been covered in research. Propose applications that can be created for such new mining problems.

13.3 Many studies analyze homogeneous information networks (e.g., social networks consisting of friends linked with friends). However, many other applications involve *heterogeneous information networks* (i.e., networks linking multiple types of object such as research papers, conference, authors, and topics). What are the major differences between methodologies for mining heterogeneous information networks and methods for their homogeneous counterparts?

13.4 Research and describe a *data mining application* that was not presented in this chapter. Discuss how different forms of data mining can be used in the application.

13.5 Why is the establishment of *theoretical foundations* important for data mining? Name and describe the main theoretical foundations that have been proposed for data mining. Comment on how they each satisfy (or fail to satisfy) the requirements of an ideal theoretical framework for data mining.

13.6 (**Research project**) Building a theory of data mining requires setting up a *theoretical framework* so that the major data mining functions can be explained under this framework. Take one theory as an example (e.g., data compression theory) and examine how the major data mining functions fit into this framework. If some functions do not fit well into the current theoretical framework, can you propose a way to extend the framework to explain these functions?

13.7 There is a strong linkage between *statistical data analysis* and data mining. Some people think of data mining as automated and scalable methods for statistical data analysis. Do you agree or disagree with this perception? Present one statistical analysis method that can be automated and/or scaled up nicely by integration with current data mining methodology.

13.8 What are the differences between *visual data mining* and *data visualization*? Data visualization may suffer from the data abundance problem. For example, it is not easy to visually discover interesting properties of network connections if a social network is huge, with complex and dense connections. Propose a visualization method that may help people see through the network topology to the interesting features of a social network.

13.9 Propose a few implementation methods for *audio data mining*. Can we integrate audio and *visual data mining* to bring fun and power to data mining? Is it possible to develop some video data mining methods? State some scenarios and your solutions to make such integrated audiovisual mining effective.

13.10 General-purpose computers and domain-independent relational database systems have become a large market in the last several decades. However, many people feel that generic data mining systems will not prevail in the data mining market. What do you think? For data mining, should we focus our efforts on developing *domain-independent* data mining tools or on developing *domain-specific* data mining solutions? Present your reasoning.

13.11 What is a *recommender system*? In what ways does it differ from a customer or product-based clustering system? How does it differ from a typical classification or predictive modeling system? Outline one method of collaborative filtering. Discuss why it works and what its limitations are in practice.

13.12 Suppose that your local bank has a data mining system. The bank has been studying your debit card usage patterns. Noticing that you make many transactions at home renovation stores, the bank decides to contact you, offering information regarding their special loans for home improvements.

(a) Discuss how this may conflict with your right to *privacy*.

(b) Describe another situation in which you feel that data mining can infringe on your privacy.

(c) Describe a *privacy-preserving data mining* method that may allow the bank to perform customer pattern analysis without infringing on its customers' right to privacy.

(d) What are some examples where data mining could be used to help society? Can you think of ways it could be used that may be detrimental to society?

13.13 What are the major challenges faced in bringing data mining research to *market*? Illustrate one data mining research issue that, in your view, may have a strong impact on the market and on society. Discuss how to approach such a research issue.

13.14 Based on your view, what is the most *challenging research problem* in data mining? If you were given a number of years and a good number of researchers and implementors, what would your plan be to make good progress toward an effective solution to such a problem?

13.15 Based on your experience and knowledge, suggest a *new frontier* in data mining that was not mentioned in this chapter.

13.8 Bibliographic Notes

For mining complex data types, there are many research papers and books covering various themes. We list here some recent books and well-cited survey or research articles for references.

Time-series analysis has been studied in statistics and computer science communities for decades, with many textbooks such as Box, Jenkins, and Reinsel [BJR08]; Brockwell and Davis [BD02]; Chatfield [Cha03b]; Hamilton [Ham94]; and Shumway and Stoffer [SS05]. A fast subsequence matching method in time-series databases was presented by Faloutsos, Ranganathan, and Manolopoulos [FRM94]. Agrawal, Lin, Sawhney, and Shim [ALSS95] developed a method for fast **similarity search** in the presence of noise, scaling, and translation in time-series databases. Shasha and Zhu present an overview of the methods for high-performance discovery in time series [SZ04].

Sequential pattern mining methods have been studied by many researchers, including Agrawal and Srikant [AS95]; Zaki [Zak01]; Pei, Han, Mortazavi-Asl, et al. [PHM-A$^+$04]; and Yan, Han, and Afshar [YHA03]. The study on **sequence classification** includes Ji, Bailey, and Dong [JBD05] and Ye and Keogh [YK09], with a survey by Xing, Pei, and Keogh [XPK10]. Dong and Pei [DP07] provide an overview on **sequence data mining** methods.

Methods for **analysis of biological sequences** including **Markov chains** and **hidden Markov models** are introduced in many books or tutorials such as Waterman [Wat95]; Setubal and Meidanis [SM97]; Durbin, Eddy, Krogh, and Mitchison [DEKM98]; Baldi and Brunak [BB01]; Krane and Raymer [KR03]; Rabiner [Rab89]; Jones and Pevzner [JP04]; and Baxevanis and Ouellette [BO04]. Information about BLAST (see also Korf, Yandell, and Bedell [KYB03]) can be found at the NCBI web site *www.ncbi.nlm.nih.gov/BLAST/*.

Graph pattern mining has been studied extensively, including Holder, Cook, and Djoko [HCD94]; Inokuchi, Washio, and Motoda [IWM98]; Kuramochi and Karypis [KK01]; Yan and Han [YH02, YH03a]; Borgelt and Berthold [BB02]; Huan, Wang, Bandyopadhyay, et al. [HWB$^+$04]; and the Gaston tool by Nijssen and Kok [NK04].

There has been a great deal of research on **social and information network analysis**, including Newman [New10]; Easley and Kleinberg [EK10]; Yu, Han, and Faloutsos [YHF10]; Wasserman and Faust [WF94]; Watts [Wat03]; and Newman, Barabasi, and Watts [NBW06]. **Statistical modeling of networks** is studied popularly such as Albert and Barbasi [AB99]; Watts [Wat03]; Faloutsos, Faloutsos, and Faloutsos [FFF99]; Kumar, Raghavan, Rajagopalan, et al. [KRR+00]; and Leskovec, Kleinberg, and Faloutsos [LKF05]. **Data cleaning, integration, and validation by information network analysis** was studied by many, including Bhattacharya and Getoor [BG04] and Yin, Han, and Yu [YHY07, YHY08].

Clustering, ranking, and classification in networks has been studied extensively, including in Brin and Page [BP98]; Chakrabarti, Dom, and Indyk [CDI98]; Kleinberg [Kle99]; Getoor, Friedman, Koller, and Taskar [GFKT01]; Newman and M. Girvan [NG04]; Yin, Han, Yang, and Yu [YHYY04]; Yin, Han, and Yu [YHY05]; Xu, Yuruk, Feng, and Schweiger [XYFS07]; Kulis, Basu, Dhillon, and Mooney [KBDM09]; Sun, Han, Zhao, et al. [SHZ+09]; Neville, Gallaher, and Eliassi-Rad [NGE-R09]; and Ji, Sun, Danilevsky et al. [JSD+10]. **Role discovery and link prediction in information networks** have been studied extensively as well, such as by Krebs [Kre02]; Kubica, Moore, and Schneider [KMS03]; Liben-Nowell and Kleinberg [L-NK03]; and Wang, Han, Jia, et al. [WHJ+10].

Similarity search and OLAP in information networks has been studied by many, including Tian, Hankins, and Patel [THP08] and Chen, Yan, Zhu, et al. [CYZ+08]. **Evolution of social and information networks** has been studied by many researchers, such as Chakrabarti, Kumar, and Tomkins [CKT06]; Chi, Song, Zhou, et al. [CSZ+07]; Tang, Liu, Zhang, and Nazeri [TLZN08]; Xu, Zhang, Yu, and Long [XZYL08]; Kim and Han [KH09]; and Sun, Tang, and Han [STH+10].

Spatial and spatiotemporal data mining has been studied extensively, with a collection of papers by Miller and Han [MH09], and was introduced in some textbooks, such as Shekhar and Chawla [SC03] and Hsu, Lee, and Wang [HLW07]. Spatial clustering algorithms have been studied extensively in Chapters 10 and 11 of this book. Research has been conducted on spatial warehouses and OLAP, such as by Stefanovic, Han, and Koperski [SHK00], and spatial and spatiotemporal data mining, such as by Koperski and Han [KH95]; Mamoulis, Cao, Kollios, Hadjieleftheriou, et al. [MCK+04]; Tsoukatos and Gunopulos [TG01]; and Hadjieleftheriou, Kollios, Gunopulos, and Tsotras [HKGT03]. **Mining moving-object data** has been studied by many, such as Vlachos, Gunopulos, and Kollios [VGK02]; Tao, Faloutsos, Papadias, and Liu [TFPL04]; Li, Han, Kim, and Gonzalez [LHKG07]; Lee, Han, and Whang [LHW07]; and Li, Ding, Han, et al. [LDH+10]. For the bibliography of temporal, spatial, and spatiotemporal data mining research, see a collection by Roddick, Hornsby, and Spiliopoulou [RHS01].

Multimedia data mining has deep roots in image processing and pattern recognition, which have been studied extensively in many textbooks, including Gonzalez and Woods [GW07]; Russ [Rus06]; Duda, Hart, and Stork [DHS01]; and Z. Zhang and R. Zhang [ZZ09]. Searching and mining of multimedia data has been studied by many (see, e.g., Fayyad and Smyth [FS93]; Faloutsos and Lin [FL95]; Natsev, Rastogi, and

Shim [NRS99]; and Zaïane, Han, and Zhu [ZHZ00]). An overview of image mining methods is given by Hsu, Lee, and Zhang [HLZ02].

Text data analysis has been studied extensively in information retrieval, with many textbooks and survey articles such as Croft, Metzler, and Strohman [CMS09]; S. Buttcher, C. Clarke, G. Cormack [BCC10]; Manning, Raghavan, and Schutze [MRS08]; Grossman and Frieder [GR04]; Baeza-Yates and Riberio-Neto [BYRN11]; Zhai [Zha08]; Feldman and Sanger [FS06]; Berry [Ber03]; and Weiss, Indurkhya, Zhang, and Damerau [WIZD04]. Text mining is a fast-developing field with numerous papers published in recent years, covering many topics such as topic models (e.g., Blei and Lafferty [BL09]); sentiment analysis (e.g., Pang and Lee [PL07]); and contextual text mining (e.g., Mei and Zhai [MZ06]).

Web mining is another focused theme, with books like Chakrabarti [Cha03a], Liu [Liu06], and Berry [Ber03]. Web mining has substantially improved search engines with a few influential milestone works, such as Brin and Page [BP98]; Kleinberg [Kle99]; Chakrabarti, Dom, Kumar, et al. [CDK+99]; and Kleinberg and Tomkins [KT99]. Numerous results have been generated since then, such as search log mining (e.g., Silvestri [Sil10]); blog mining (e.g., Mei, Liu, Su, and Zhai [MLSZ06]); and mining online forums (e.g., Cong, Wang, Lin, et al. [CWL+08]).

Books and surveys on stream data systems and stream data processing include Babu and Widom [BW01]; Babcock, Babu, Datar, et al. [BBD+02]; Muthukrishnan [Mut05]; and Aggarwal [Agg06].

Stream data mining research covers stream cube models (e.g., Chen, Dong, Han, et al. [CDH+02]), stream frequent pattern mining (e.g., Manku and Motwani [MM02] and Karp, Papadimitriou and Shenker [KPS03]), stream classification (e.g., Domingos and Hulten [DH00]; Wang, Fan, Yu, and Han [WFYH03]; Aggarwal, Han, Wang, and Yu [AHWY04b]), and stream clustering (e.g., Guha, Mishra, Motwani, and O'Callaghan [GMMO00] and Aggarwal, Han, Wang, and Yu [AHWY03]).

There are many books that discuss **data mining applications**. For financial data analysis and financial modeling, see, for example, Benninga [Ben08] and Higgins [Hig08]. For retail data mining and customer relationship management, see, for example, books by Berry and Linoff [BL04] and Berson, Smith, and Thearling [BST99]. For telecommunication-related data mining, see, for example, Horak [Hor08]. There are also books on scientific data analysis, such as Grossman, Kamath, Kegelmeyer, et al. [GKK+01] and Kamath [Kam09].

Issues in the **theoretical foundations of data mining** have been addressed by many researchers. For example, Mannila presents a summary of studies on the foundations of data mining in [Man00]. The data reduction view of data mining is summarized in *The New Jersey Data Reduction Report* by Barbará, DuMouchel, Faloutos, et al. [BDF+97]. The data compression view can be found in studies on the minimum description length principle, such as Grunwald and Rissanen [GR07].

The pattern discovery point of view of data mining is addressed in numerous machine learning and data mining studies, ranging from association mining, to decision tree induction, sequential pattern mining, clustering, and so on. The probability theory point of view is popular in the statistics and machine learning literature, such

as Bayesian networks and hierarchical Bayesian models in Chapter 9, and probabilistic graph models (e.g., Koller and Friedman [KF09]). Kleinberg, Papadimitriou, and Raghavan [KPR98] present a microeconomic view, treating data mining as an optimization problem. Studies on the inductive database view include Imielinski and Mannila [IM96] and de Raedt, Guns, and Nijssen [RGN10].

Statistical methods for data analysis are described in many books, such as Hastie, Tibshirani, Friedman [HTF09]; Freedman, Pisani, and Purves [FPP07]; Devore [Dev03]; Kutner, Nachtsheim, Neter, and Li [KNNL04]; Dobson [Dob01]; Breiman, Friedman, Olshen, and Stone [BFOS84]; Pinheiro and Bates [PB00]; Johnson and Wichern [JW02b]; Huberty [Hub94]; Shumway and Stoffer [SS05]; and Miller [Mil98].

For **visual data mining**, popular books on the visual display of data and information include those by Tufte [Tuf90, Tuf97, Tuf01]. A summary of techniques for visualizing data is presented in Cleveland [Cle93]. A dedicated visual data mining book, *Visual Data Mining: Techniques and Tools for Data Visualization and Mining*, is by Soukup and Davidson [SD02]. The book *Information Visualization in Data Mining and Knowledge Discovery*, edited by Fayyad, Grinstein, and Wierse [FGW01], contains a collection of articles on visual data mining methods.

Ubiquitous and invisible data mining has been discussed in many texts including John [Joh99], and some articles in a book edited by Kargupta, Joshi, Sivakumar, and Yesha [KJSY04]. The book *Business @ the Speed of Thought: Succeeding in the Digital Economy* by Gates [Gat00] discusses e-commerce and customer relationship management, and provides an interesting perspective on data mining in the future. Mena [Men03] has an informative book on the use of data mining to detect and prevent crime. It covers many forms of criminal activities, ranging from fraud detection, money laundering, insurance crimes, identity crimes, and intrusion detection.

Data mining issues regarding **privacy and data security** are addressed popularly in literature. Books on privacy and security in data mining include Thuraisingham [Thu04]; Aggarwal and Yu [AY08]; Vaidya, Clifton, and Zhu [VCZ10]; and Fung, Wang, Fu, and Yu [FWFY10]. Research articles include Agrawal and Srikant [AS00]; Evfimievski, Srikant, Agrawal, and Gehrke [ESAG02]; and Vaidya and Clifton [VC03]. Differential privacy was introduced by Dwork [Dwo06] and studied by many such as Hay, Rastogi, Miklau, and Suciu [HRMS10].

There have been many discussions on **trends and research directions of data mining** in various forums. Several books are collections of articles on these issues such as Kargupta, Han, Yu, et al. [KHY+08].

This page intentionally left blank

Bibliography

[AAD+96] S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In *Proc. 1996 Int. Conf. Very Large Data Bases (VLDB'96)*, pp. 506–521, Bombay, India, Sept. 1996.

[AAP01] R. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. *J. Parallel and Distributed Computing*, 61:350–371, 2001.

[AB79] B. Abraham and G. E. P. Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, 66:229–248, 1979.

[AB99] R. Albert and A.-L. Barabasi. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[ABA06] M. Agyemang, K. Barker, and R. Alhajj. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intell. Data Anal.*, 10:521–538, 2006.

[ABKS99] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pp. 49–60, Philadelphia, PA, June 1999.

[AD91] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proc. 1991 Nat. Conf. Artificial Intelligence (AAAI'91)*, pp. 547–552, Anaheim, CA, July 1991.

[AEEK99] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, pp. 392–396, San Diego, CA, Aug. 1999.

[AEMT00] K. M. Ahmed, N. M. El-Makky, and Y. Taha. A note on "beyond market basket: Generalizing association rules to correlations." *SIGKDD Explorations*, 1:46–48, 2000.

[AG60] F. J. Anscombe, and I. Guttman. Rejection of outliers. *Technometrics*, 2:123–147, 1960.

[Aga06] D. Agarwal. Detecting anomalies in cross-classified streams: A Bayesian approach. *Knowl. Inf. Syst.*, 11:29–44, 2006.

[AGAV09] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

[Agg06] C. C. Aggarwal. *Data Streams: Models and Algorithms*. Kluwer Academic, 2006.

[AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 94–105, Seattle, WA, June 1998.

[AGM04] F. N. Afrati, A. Gionis, and H. Mannila. Approximating a collection of frequent sets. In *Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'04)*, pp. 12–19, Seattle, WA, Aug. 2004.

[AGS97] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In *Proc. 1997 Int. Conf. Data Engineering (ICDE'97)*, pp. 232–243, Birmingham, England, Apr. 1997.

[Aha92] D. Aha. Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *Int. J. Man-Machine Studies*, 36:267–287, 1992.

[AHS96] P. Arabie, L. J. Hubert, and G. De Soete. *Clustering and Classification*. World Scientific, 1996.

[AHWY03] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proc. 2003 Int. Conf. Very Large Data Bases (VLDB'03)*, pp. 81–92, Berlin, Germany, Sept. 2003.

[AHWY04a] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for projected clustering of high dimensional data streams. In *Proc. 2004 Int. Conf. Very Large Data Bases (VLDB'04)*, pp. 852–863, Toronto, Ontario, Canada, Aug. 2004.

[AHWY04b] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. On demand classification of data streams. In *Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'04)*, pp. 503–508, Seattle, WA, Aug. 2004.

[AIS93] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'93)*, pp. 207–216, Washington, DC, May 1993.

[AK93] T. Anand and G. Kahn. Opportunity explorer: Navigating large databases using knowledge discovery templates. In *Proc. AAAI-93 Workshop Knowledge Discovery in Databases*, pp. 45–51, Washington, DC, July 1993.

[AL99] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, pp. 261–270, San Diego, CA, Aug. 1999.

[All94] B. P. Allen. Case-based reasoning: Business applications. *Communications of the ACM*, 37:40–42, 1994.

[Alp11] E. Alpaydin. *Introduction to Machine Learning* (2nd ed.). Cambridge, MA: MIT Press, 2011.

[ALSS95] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 490–501, Zurich, Switzerland, Sept. 1995.

[AMS+96] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger. The Quest data mining system. In *Proc. 1996 Int. Conf. Data Mining and Knowledge Discovery (KDD'96)*, pp. 244–249, Portland, OR, Aug. 1996.

[Aok98] P. M. Aoki. Generalizing "search" in generalized search trees. In *Proc. 1998 Int. Conf. Data Engineering (ICDE'98)*, pp. 380–389, Orlando, FL, Feb. 1998.

[AP94] A. Aamodt and E. Plazas. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7:39–52, 1994.

[AP05] F. Angiulli, and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Trans. on Knowl. and Data Eng.*, 17:203–215, 2005.

[APW+99] C. C. Aggarwal, C. Procopiuc, J. Wolf, P. S. Yu, and J.-S. Park. Fast algorithms for projected clustering. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pp. 61–72, Philadelphia, PA, June 1999.

[ARV09] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):1–37, 2009.

[AS94a] R. Agrawal and R. Srikant. Fast algorithm for mining association rules in large databases. In *Research Report RJ 9839*, IBM Almaden Research Center, San Jose, CA, June 1994.

[AS94b] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94)*, pp. 487–499, Santiago, Chile, Sept. 1994.

[AS95] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 1995 Int. Conf. Data Engineering (ICDE'95)*, pp. 3–14, Taipei, Taiwan, Mar. 1995.

[AS96] R. Agrawal and J. C. Shafer. Parallel mining of association rules: Design, implementation, and experience. *IEEE Trans. Knowledge and Data Engineering*, 8:962–969, 1996.

[AS00] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00)*, pp. 439–450, Dallas, TX, May 2000.

[ASS00] E. Allwein, R. Shapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[AV07] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proc. 2007 ACM-SIAM Symp. on Discrete Algorithms (SODA'07)*, pp. 1027–1035, Tokyo, 2007.

[Avn95] S. Avner. Discovery of comprehensible symbolic rules in a neural network. In *Proc. 1995 Int. Symp. Intelligence in Neural and Biological Systems*, pp. 64–67, Washington, DC, 1995.

[AY99] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proc. 1998 ACM Symp. Principles of Database Systems (PODS'98)*, pp. 18–24, Seattle, WA, June 1999.

[AY01] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *Proc. 2001 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'01)*, pp. 37–46, Santa Barbara, CA, May 2001.

[AY08] C. C. Aggarwal and P. S. Yu. *Privacy-Preserving Data Mining: Models and Algorithms*. New York: Springer, 2008.

[BA97] L. A. Breslow and D. W. Aha. Simplifying decision trees: A survey. *Knowledge Engineering Rev.*, 12:1–40, 1997.

[Bay98] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 85–93, Seattle, WA, June 1998.

[BB98] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proc. 1998 Annual Meeting of the Association for Computational Linguistics and Int. Conf. Computational Linguistics (COLING-ACL'98)*, Montreal, Quebec, Canada, Aug. 1998.

[BB01] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach* (2nd ed.). Cambridge, MA: MIT Press, 2001.

[BB02] C. Borgelt and M. R. Berthold. Mining molecular fragments: Finding relevant substructures of molecules. In *Proc. 2002 Int. Conf. Data Mining (ICDM'02)*, pp. 211–218, Maebashi, Japan, Dec. 2002.

[BBD+02] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proc. 2002 ACM Symp. Principles of Database Systems (PODS'02)*, pp. 1–16, Madison, WI, June 2002.

[BC83] R. J. Beckman and R. D. Cook. Outlier...s. *Technometrics*, 25:119–149, 1983.

[BCC10] S. Buettcher, C. L. A. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. Cambridge, MA: MIT Press, 2010.

[BCG01] D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, pp. 443–452, Heidelberg, Germany, Apr. 2001.

[BCP93] D. E. Brown, V. Corruble, and C. L. Pittard. A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problems. *Pattern Recognition*, 26:953–961, 1993.

[BD01] P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*, Vol. 1. Prentice-Hall, 2001.

[BD02] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting* (2nd ed.). New York: Springer, 2002.

[BDF+97] D. Barbará, W. DuMouchel, C. Faloutsos, P. J. Haas, J. H. Hellerstein, Y. Ioannidis, H. V. Jagadish, T. Johnson, R. Ng, V. Poosala, K. A. Ross, and K. C. Servcik. The New Jersey data reduction report. *Bull. Technical Committee on Data Engineering*, 20:3–45, Dec. 1997.

[BDG96] A. Bruce, D. Donoho, and H.-Y. Gao. Wavelet analysis. *IEEE Spectrum*, 33:26–35, Oct. 1996.

[BDJ+05] D. Burdick, P. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. OLAP over uncertain and imprecise data. In *Proc. 2005 Int. Conf. Very Large Data Bases (VLDB'05)*, pp. 970–981, Trondheim, Norway, Aug. 2005.

[Ben08] S. Benninga. *Financial Modeling* (3rd. ed.). Cambridge, MA: MIT Press, 2008.

[Ber81] J. Bertin. *Graphics and Graphic Information Processing*. Walter de Gruyter, Berlin, 1981.

[Ber03] M. W. Berry. *Survey of Text Mining: Clustering, Classification, and Retrieval*. New York: Springer, 2003.

[Bez81] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[BFR98] P. Bradley, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)*, pp. 9–15, New York, Aug. 1998.

[BG04] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *Proc. SIGMOD 2004 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'04)*, pp. 11–18, Paris, France, June 2004.

[B-G05] I. Ben-Gal. Outlier detection. In O. Maimon and L. Rockach (eds.), *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*. Kluwer Academic, 2005.

[BGKW03] C. Bucila, J. Gehrke, D. Kifer, and W. White. DualMiner: A dual-pruning algorithm for itemsets with constraints. *Data Mining and Knowledge Discovery*, 7:241–272, 2003.

[BGMP03] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi. ExAnte: Anticipated data reduction in constrained pattern mining. In *Proc. 7th European Conf. Principles and Pratice of Knowledge Discovery in Databases (PKDD'03)*, Vol. 2838/2003, pp. 59–70, Cavtat-Dubrovnik, Croatia, Sept. 2003.

[BGRS99] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proc. 1999 Int. Conf. Database Theory (ICDT'99)*, pp. 217–235, Jerusalem, Israel, Jan. 1999.

[BGV92] B. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, ACM Press, San Mateo, CA, 1992.

[Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[BJR08] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control* (4th ed.). Prentice-Hall, 2008.

[BKNS00] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00)*, pp. 93–104, Dallas, TX, May 2000.

[BL99] M. J. A. Berry and G. Linoff. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. John Wiley & Sons, 1999.

[BL04] M. J. A. Berry and G. S. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. John Wiley & Sons, 2004.

[BL09] D. Blei and J. Lafferty. Topic models. In A. Srivastava and M. Sahami (eds.), *Text Mining: Theory and Applications*, Taylor and Francis, 2009.

[BLC+03] D. Barbará, Y. Li, J. Couto, J.-L. Lin, and S. Jajodia. Bootstrapping a data mining intrusion detection system. In *Proc. 2003 ACM Symp. on Applied Computing (SAC'03)*, Melbourne, FL, March 2003.

[BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. 11th Conf. Computational Learning Theory (COLT'98)*, pp. 92–100, Madison, WI, 1998.

[BMAD06] Z. A. Bakar, R. Mohemad, A. Ahmad, and M. M. Deris. A comparative study for outlier detection techniques in data mining. In *Proc. 2006 IEEE Conf. Cybernetics and Intelligent Systems*, pp. 1–6, Bangkok, Thailand, 2006.

[BMS97] S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pp. 265–276, Tucson, AZ, May 1997.

[BMUT97] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pp. 255–264, Tucson, AZ, May 1997.

[BN92] W. L. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–85, 1992.

[BO04] A. Baxevanis and B. F. F. Ouellette. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins* (3rd ed.). John Wiley & Sons, 2004.

[BP92] J. C. Bezdek and S. K. Pal. *Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data*. IEEE Press, 1992.

[BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. 7th Int. World Wide Web Conf. (WWW'98)*, pp. 107–117, Brisbane, Australia, Apr. 1998.

[BPT97] E. Baralis, S. Paraboschi, and E. Teniente. Materialized view selection in a multidimensional database. In *Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97)*, pp. 98–12, Athens, Greece, Aug. 1997.

[BPW88] E. R. Bareiss, B. W. Porter, and C. C. Weir. Protos: An exemplar-based learning apprentice. *Int. J. Man-Machine Studies*, 29:549–561, 1988.

[BR99] K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pp. 359–370, Philadelphia, PA, June 1999.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[BS97] D. Barbará and M. Sullivan. Quasi-cubes: Exploiting approximation in multidimensional databases. *SIGMOD Record*, 26:12–17, 1997.

[BS03] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 29–38, Washington, DC, Aug. 2003.

[BST99] A. Berson, S. J. Smith, and K. Thearling. *Building Data Mining Applications for CRM*. McGraw-Hill, 1999.

[BT99] D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. *Communications of the ACM*, 42:73–78, 1999.

[BU95] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45–77, 1995.

[Bun94] W. L. Buntine. Operations for learning with graphical models. *J. Artificial Intelligence Research*, 2:159–225, 1994.

[Bur98] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–168, 1998.

[BW00] D. Barbará and X. Wu. Using loglinear models to compress datacubes. In *Proc. 1st Int. Conf. Web-Age Information Management (WAIM'00)*, pp. 311–322, Shanghai, China, 2000.

[BW01] S. Babu and J. Widom. Continuous queries over data streams. *SIGMOD Record*, 30: 109–120, 2001.

[BYRN11] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval* (2nd ed.). Boston: Addison-Wesley, 2011.

[Cat91] J. Catlett. *Megainduction: Machine Learning on Very large Databases*. Ph.D. Thesis, University of Sydney, 1991.

[CBK09] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41:1–58, 2009.

[CC00] Y. Cheng and G. Church. Biclustering of expression data. In *Proc. 2000 Int. Conf. Intelligent Systems for Molecular Biology (ISMB'00)*, pp. 93–103, La Jolla, CA, Aug. 2000.

[CCH91] Y. Cai, N. Cercone, and J. Han. Attribute-oriented induction in relational databases. In G. Piatetsky-Shapiro and W. J. Frawley (eds.), *Knowledge Discovery in Databases*, pp. 213–228. AAAI/MIT Press, 1991.

[CCLR05] B.-C. Chen, L. Chen, Y. Lin, and R. Ramakrishnan. Prediction cubes. In *Proc. 2005 Int. Conf. Very Large Data Bases (VLDB'05)*, pp. 982–993, Trondheim, Norway, Aug. 2005.

[CCS93] E. F. Codd, S. B. Codd, and C. T. Salley. Beyond decision support. *Computer World*, 27(30):5–12, July 1993.

[CD97] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26:65–74, 1997.

[CDH+02] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multidimensional regression analysis of time-series data streams. In *Proc. 2002 Int. Conf. Very Large Data Bases (VLDB'02)*, pp. 323–334, Hong Kong, China, Aug. 2002.

[CDH+06] Y. Chen, G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang. Regression cubes with lossless compression and aggregation. *IEEE Trans. Knowledge and Data Engineering*, 18:1585–1599, 2006.

[CDI98] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext classification using hyperlinks. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 307–318, Seattle, WA, June 1998.

[CDK+99] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. M. Kleinberg. Mining the web's link structure. *COMPUTER*, 32:60–67, 1999.

[CGC94] A. Chaturvedi, P. Green, and J. Carroll. *k*-means, *k*-medians and *k*-modes: Special cases of partitioning multiway data. In *The Classification Society of North America (CSNA) Meeting Presentation*, Houston, TX, 1994.

[CGC01] A. Chaturvedi, P. Green, and J. Carroll. *k*-modes clustering. *J. Classification*, 18:35–55, 2001.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory*, 13:21–27, 1967.

[CH92] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[CH07] D. J. Cook and L. B. Holder. *Mining Graph Data*. John Wiley & Sons, 2007.

[Cha03a] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2003.

[Cha03b] C. Chatfield. *The Analysis of Time Series: An Introduction* (6th ed.). Chapman & Hall, 2003.

[CHN+96] D. W. Cheung, J. Han, V. Ng, A. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In *Proc. 1996 Int. Conf. Parallel and Distributed Information Systems*, pp. 31–44, Miami Beach, FL, Dec. 1996.

[CHNW96] D. W. Cheung, J. Han, V. Ng, and C. Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Proc. 1996 Int. Conf. Data Engineering (ICDE'96)*, pp. 106–114, New Orleans, LA, Feb. 1996.

[CHY96] M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. *IEEE Trans. Knowledge and Data Engineering*, 8:866–883, 1996.

[CK98] M. Carey and D. Kossman. Reducing the braking distance of an SQL query engine. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 158–169, New York, Aug. 1998.

[CKT06] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'06)*, pp. 554–560, Philadelphia, PA, Aug. 2006.

[Cle93] W. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[CSZ06] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised Learning*. Cambridge, MA: MIT Press, 2006.

[CM94] S. P. Curram and J. Mingers. Neural networks, decision tree induction and discriminant analysis: An empirical comparison. *J. Operational Research Society*, 45:440–450, 1994.

[CMC05] H. Cao, N. Mamoulis, and D. W. Cheung. Mining frequent spatio-temporal sequential patterns. In *Proc. 2005 Int. Conf. Data Mining (ICDM'05)*, pp. 82–89, Houston, TX, Nov. 2005.

[CMS09] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Boston: Addison-Wesley, 2009.

[CN89] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[Coh95] W. Cohen. Fast effective rule induction. In *Proc. 1995 Int. Conf. Machine Learning (ICML'95)*, pp. 115–123, Tahoe City, CA, July 1995.

[Coo90] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[CPS98] K. Cios, W. Pedrycz, and R. Swiniarski. *Data Mining Methods for Knowledge Discovery*. Kluwer Academic, 1998.

[CR95] Y. Chauvin and D. Rumelhart. *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum, 1995.

[Cra89] S. L. Crawford. Extensions to the CART algorithm. *Int. J. Man-Machine Studies*, 31:197–217, Aug. 1989.

[CRST06] B.-C. Chen, R. Ramakrishnan, J. W. Shavlik, and P. Tamma. Bellwether analysis: Predicting global aggregates from local regions. In *Proc. 2006 Int. Conf. Very Large Data Bases (VLDB'06)*, pp. 655–666, Seoul, Korea, Sept. 2006.

[CS93a] P. K. Chan and S. J. Stolfo. Experiments on multistrategy learning by metalearning. In *Proc. 2nd. Int. Conf. Information and Knowledge Management (CIKM'93)*, pp. 314–323, Washington, DC, Nov. 1993.

[CS93b] P. K. Chan and S. J. Stolfo. Toward multi-strategy parallel & distributed learning in sequence analysis. In *Proc. 1st Int. Conf. Intelligent Systems for Molecular Biology (ISMB'93)*, pp. 65–73, Bethesda, MD, July 1993.

[CS96] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. In D. Touretzky, M. Mozer, and M. Hasselmo (eds.), *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1996.

[CS97] M. W. Craven and J. W. Shavlik. Using neural networks in data mining. *Future Generation Computer Systems*, 13:211–229, 1997.

[CS-T00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[CSZ+07] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proc. 2007 ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining (KDD'07)*, pp. 153–162, San Jose, CA, Aug. 2007.

[CTTX05] G. Cong, K.-Lee Tan, A. K. H. Tung, and X. Xu. Mining top-k covering rule groups for gene expression data. In *Proc. 2005 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'05)*, pp. 670–681, Baltimore, MD, June 2005.

[CWL+08] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proc. 2008 Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR'08)*, pp. 467–474, Singapore, July 2008.

[CYHH07] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proc. 2007 Int. Conf. Data Engineering (ICDE'07)*, pp. 716–725, Istanbul, Turkey, Apr. 2007.

[CYHY08] H. Cheng, X. Yan, J. Han, and P. S. Yu. Direct discriminative pattern mining for effective classification. In *Proc. 2008 Int. Conf. Data Engineering (ICDE'08)*, pp. 169–178, Cancun, Mexico, Apr. 2008.

[CYZ+08] C. Chen, X. Yan, F. Zhu, J. Han, and P. S. Yu. Graph OLAP: Towards online analytical processing on graphs. In *Proc. 2008 Int. Conf. Data Mining (ICDM'08)*, pp. 103–112, Pisa, Italy, Dec. 2008.

[Dar10] A. Darwiche. Bayesian networks. *Communications of the ACM*, 53:80–90, 2010.

[Das91] B. V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1991.

[Dau92] I. Daubechies. *Ten Lectures on Wavelets*. Capital City Press, 1992.

[DB95] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *J. Artificial Intelligence Research*, 2:263–286, 1995.

[DBK+97] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. N. Vapnik. Support vector regression machines. In M. Mozer, M. Jordan, and T. Petsche (eds.), *Advances in Neural Information Processing Systems 9*, pp. 155–161. Cambridge, MA: MIT Press, 1997.

[DE84] W. H. E. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *J. Classification*, 1:7–24, 1984.

[De01] S. Dzeroski and N. Lavrac (eds.). *Relational Data Mining*. New York: Springer, 2001.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probability Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[Dev95] J. L. Devore. *Probability and Statistics for Engineering and the Sciences* (4th ed.). Duxbury Press, 1995.

[Dev03] J. L. Devore. *Probability and Statistics for Engineering and the Sciences* (6th ed.). Duxbury Press, 2003.

[DH73] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Research and Development*, 17:420–425, 1973.

[DH00] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proc. 2000 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'00)*, pp. 71–80, Boston, MA, Aug. 2000.

[DHL+01] G. Dong, J. Han, J. Lam, J. Pei, and K. Wang. Mining multi-dimensional constrained gradients in data cubes. In *Proc. 2001 Int. Conf. Very Large Data Bases (VLDB'01)*, pp. 321–330, Rome, Italy, Sept. 2001.

[DHL+04] G. Dong, J. Han, J. Lam, J. Pei, K. Wang, and W. Zou. Mining constrained gradients in multi-dimensional databases. *IEEE Trans. Knowledge and Data Engineering*, 16:922–938, 2004.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification* (2nd ed.). John Wiley & Sons, 2001.

[DJ03] T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, 2003.

[DJMS02] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or how to build a data quality browser. In *Proc. 2002 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'02)*, pp. 240–251, Madison, WI, June 2002.

[DL97] M. Dash and H. Liu. Feature selection methods for classification. *Intelligent Data Analysis*, 1:131–156, 1997.

[DL99] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, pp. 43–52, San Diego, CA, Aug. 1999.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society, Series B*, 39:1–38, 1977.

[DLY97] M. Dash, H. Liu, and J. Yao. Dimensionality reduction of unsupervised data. In *Proc. 1997 IEEE Int. Conf. Tools with AI (ICTAI'97)*, pp. 532–539, Newport Beach, CA, IEEE Computer Society, 1997.

[DM02] D. Dasgupta and N. S. Majumdar. Anomaly detection in multidimensional data using negative selection algorithm. In *Proc. 2002 Congress on Evolutionary Computation (CEC'02)*, Chapter 12, pp. 1039–1044, Washington, DC, 2002.

[DNR+97] P. Deshpande, J. Naughton, K. Ramasamy, A. Shukla, K. Tufte, and Y. Zhao. Cubing algorithms, storage estimation, and storage and processing alternatives for OLAP. *Bull. Technical Committee on Data Engineering*, 20:3–11, 1997.

[Dob90] A. J. Dobson. *An Introduction to Generalized Linear Models*. Chapman & Hall, 1990.

[Dob01] A. J. Dobson. *An Introduction to Generalized Linear Models* (2nd ed.). Chapman & Hall, 2001.

[Dom94] P. Domingos. The RISE system: Conquering without separating. In *Proc. 1994 IEEE Int. Conf. Tools with Artificial Intelligence (TAI'94)*, pp. 704–707, New Orleans, LA, 1994.

[Dom99] P. Domingos. The role of Occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3:409–425, 1999.

[DP96] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proc. 1996 Int. Conf. Machine Learning (ML'96)*, pp. 105–112, Bari, Italy, July 1996.

[DP97] J. Devore and R. Peck. *Statistics: The Exploration and Analysis of Data*. Duxbury Press, 1997.

[DP07] G. Dong and J. Pei. *Sequence Data Mining*. New York: Springer, 2007.

[DR99] D. Donjerkovic and R. Ramakrishnan. Probabilistic optimization of top N queries. In *Proc. 1999 Int. Conf. Very Large Data Bases (VLDB'99)*, pp. 411–422, Edinburgh, UK, Sept. 1999.

[DR05] I. Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the *k*-means algorithm. In *Proc. 2005 SIAM Int. Conf. Data Mining (SDM'05)*, Newport Beach, CA, Apr. 2005.

[DT93] V. Dhar and A. Tuzhilin. Abstract-driven pattern discovery in databases. *IEEE Trans. Knowledge and Data Engineering*, 5:926–938, 1993.

[Dun03] M. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice-Hall, 2003.

[DWB06] I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proc. 10th European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD'06)*, pp. 115–126, Berlin, Germany, Sept. 2006.

[Dwo06] C. Dwork. Differential privacy. In *Proc. 2006 Int. Col. Automata, Languages and Programming (ICALP)*, pp. 1–12, Venice, Italy, July 2006.

[DYXY07] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. In *Proc. 24th Intl. Conf. Machine Learning*, pp. 193–200, Corvallis, OR, June 2007.

[Ega75] J. P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

[EK10] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[Esk00] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th Int. Conf. Machine Learning (ICML'00)*, Stanford, CA, 2000.

[EKSX96] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In *Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, pp. 226–231, Portland, OR, Aug. 1996.

[EKX95] M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. In *Proc. 1995 Int. Symp. Large Spatial Databases (SSD'95)*, pp. 67–82, Portland, ME, Aug. 1995.

[Elk97] C. Elkan. Boosting and naïve Bayesian learning. In *Technical Report CS97-557*, Dept. Computer Science and Engineering, University of California at San Diego, Sept. 1997.

[Elk01] C. Elkan. The foundations of cost-sensitive learning. In *Proc. 17th Intl. Joint Conf. Artificial Intelligence (IJCAI'01)*, pp. 973–978, Seattle, WA, 2001.

[EN10] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems* (6th ed.). Boston: Addison-Wesley, 2010.

[Eng99] L. English. *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits*. John Wiley & Sons, 1999.

[ESAG02] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'02)*, pp. 217–228, Edmonton, Alberta, Canada, July 2002.

[ET93] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[FB74] R. A. Finkel and J. L. Bentley. Quad-trees: A data structure for retrieval on composite keys. *ACTA Informatica*, 4:1–9, 1974.

[FB08] J. Friedman and E. P. Bogdan. Predictive learning via rule ensembles. *Ann. Applied Statistics*, 2:916–954, 2008.

[FBF77] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Math Software*, 3:209–226, 1977.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proc. ACM SIGCOMM'99 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 251–262, Cambridge, MA, Aug. 1999.

[FG02] M. Fishelson and D. Geiger. Exact genetic linkage computations for general pedigrees. *Disinformation*, 18:189–198, 2002.

[FGK+05] R. Fagin, R. V. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Multi-structural databases. In *Proc. 2005 ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS'05)*, pp. 184–195, Baltimore, MD, June 2005.

[FGW01] U. Fayyad, G. Grinstein, and A. Wierse. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2001.

[FH51] E. Fix and J. L. Hodges Jr. Discriminatory analysis, non-parametric discrimination: Consistency properties. In *Technical Report 21-49-004(4)*, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[FH87] K. Fukunaga and D. Hummels. Bayes error estimation using Parzen and *k-nn* procedure. *IEEE Trans. Pattern Analysis and Machine Learning*, 9:634–643, 1987.

[FH95] Y. Fu and J. Han. Meta-rule-guided mining of association rules in relational databases. In *Proc. 1995 Int. Workshop Integration of Knowledge Discovery with Deductive and Object-Oriented Databases (KDOOD'95)*, pp. 39–46, Singapore, Dec. 1995.

[FI90] U. M. Fayyad and K. B. Irani. What should be minimized in a decision tree? In *Proc. 1990 Nat. Conf. Artificial Intelligence (AAAI'90)*, pp. 749–754, Boston, MA, 1990.

[FI92] U. M. Fayyad and K. B. Irani. The attribute selection problem in decision tree generation. In *Proc. 1992 Nat. Conf. Artificial Intelligence (AAAI'92)*, pp. 104–110, San Jose, CA, 1992.

[FI93] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. 1993 Int. Joint Conf. Artificial Intelligence (IJCAI'93)*, pp. 1022–1029, Chambery, France, 1993.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical J.*, 23:298–305, 1973.

[FL90] S. Fahlman and C. Lebiere. The cascade-correlation learning algorithm. In *Technical Report CMU-CS-90-100*, Computer Sciences Department, Carnegie Mellon University, 1990.

[FL95] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. 1995 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'95)*, pp. 163–174, San Jose, CA, May 1995.

[Fle87] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 1987.

[FMMT96] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96)*, pp. 13–23, Montreal, Quebec, Canada, June 1996.

[FP05] J. Friedman and B. E. Popescu. Predictive learning via rule ensembles. In *Technical Report*, Department of Statistics, Stanford University, 2005.

[FPP07] D. Freedman, R. Pisani, and R. Purves. *Statistics* (4th ed.). W. W. Norton & Co., 2007.

[FPSS+96] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[FP97] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1:291–316, 1997.

[FR02] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *J. American Statistical Association*, 97:611–631, 2002.

[Fri77] J. H. Friedman. A recursive partitioning decision rule for nonparametric classifiers. *IEEE Trans. Computer*, 26:404–408, 1977.

[Fri01] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statistics*, 29:1189–1232, 2001.

[Fri03] N. Friedman. Pcluster: Probabilistic agglomerative clustering of gene expression profiles. In *Technical Report 2003-80*, Hebrew University, 2003.

[FRM94] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proc. 1994 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'94)*, pp. 419–429, Minneapolis, MN, May 1994.

[FS93] U. Fayyad and P. Smyth. Image database exploration: Progress and challenges. In *Proc. AAAI'93 Workshop Knowledge Discovery in Databases (KDD'93)*, pp. 14–27, Washington, DC, July 1993.

[FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, 55:119–139, 1997.

[FS06] R. Feldman and J. Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006.

[FSGM+98] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 299–310, New York, NY, Aug. 1998.

[FW94] J. Furnkranz and G. Widmer. Incremental reduced error pruning. In *Proc. 1994 Int. Conf. Machine Learning (ICML'94)*, pp. 70–77, New Brunswick, NJ, 1994.

[FWFY10] B. C. M. Fung, K. Wang, A. W.-C. Fu, and P. S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman & Hall/CRC, 2010.

[FYM05] R. Fujimaki, T. Yairi, and K. Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proc. 2005 Int. Workshop Link Discovery (LinkKDD'05)*, pp. 401–410, Chicago, IL, 2005.

[Gal93] S. I. Gallant. *Neural Network Learning and Expert Systems*. Cambridge, MA: MIT Press, 1993.

[Gat00] B. Gates. *Business @ the Speed of Thought: Succeeding in the Digital Economy*. Warner Books, 2000.

[GCB+97] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–54, 1997.

[GFKT01] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *Proc. 2001 Int. Conf. Machine Learning (ICML'01)*, pp. 170–177, Williamstown, MA, 2001.

[GFS+01] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. In *Proc. 2001 Int. Conf. Very Large Data Bases (VLDB'01)*, pp. 371–380, Rome, Italy, Sept. 2001.

[GG92] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, 1992.

[GG98] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30:170–231, 1998.

[GGR99] V. Ganti, J. E. Gehrke, and R. Ramakrishnan. CACTUS—clustering categorical data using summaries. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, pp. 73–83, San Diego, CA, 1999.

[GGRL99] J. Gehrke, V. Ganti, R. Ramakrishnan, and W.-Y. Loh. BOAT—optimistic decision tree construction. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pp. 169–180, Philadelphia, PA, June 1999.

[GHL06] H. Gonzalez, J. Han, and X. Li. Flowcube: Constructuing RFID flowcubes for multi-dimensional analysis of commodity flows. In *Proc. 2006 Int. Conf. Very Large Data Bases (VLDB'06)*, pp. 834–845, Seoul, Korea, Sept. 2006.

[GHLK06] H. Gonzalez, J. Han, X. Li, and D. Klabjan. Warehousing and analysis of massive RFID data sets. In *Proc. 2006 Int. Conf. Data Engineering (ICDE'06)*, p. 83, Atlanta, GA, Apr. 2006.

[GKK+01] R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. R. Namburu. *Data Mining for Scientific and Engineering Applications*. Kluwer Academic, 2001.

[GKR98] D. Gibson, J. M. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 311–323, New York, NY, Aug. 1998.

[GM99] A. Gupta and I. S. Mumick. *Materialized Views: Techniques, Implementations, and Applications*. Cambridge, MA: MIT Press, 1999.

[GMMO00] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *Proc. 2000 Symp. Foundations of Computer Science (FOCS'00)*, pp. 359–366, Redondo Beach, CA, 2000.

[GMP+09] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, Feb. 2009.

[GMUW08] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database Systems: The Complete Book* (2nd ed.). Prentice Hall, 2008.

[GMV96] I. Guyon, N. Matic, and V. Vapnik. Discoverying informative patterns and data cleaning. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 181–203. AAAI/MIT Press, 1996.

[Gol89] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[GR04] D. A. Grossman and O. Frieder. *Information Retrieval: Algorithms and Heuristics*. New York: Springer, 2004.

[GR07] P. D. Grunwald and J. Rissanen. *The Minimum Description Length Principle*. Cambridge, MA: MIT Press, 2007.

[GRG98] J. Gehrke, R. Ramakrishnan, and V. Ganti. RainForest: A framework for fast decision tree construction of large datasets. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 416–427, New York, NY, Aug. 1998.

[GRS98] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 73–84, Seattle, WA, June 1998.

[GRS99] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In *Proc. 1999 Int. Conf. Data Engineering (ICDE'99)*, pp. 512–521, Sydney, Australia, Mar. 1999.

[Gru69] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21, 1969.

[Gup97] H. Gupta. Selection of views to materialize in a data warehouse. In *Proc. 7th Int. Conf. Database Theory (ICDT'97)*, pp. 98–112, Delphi, Greece, Jan. 1997.

[Gut84] A. Guttman. R-Tree: A dynamic index structure for spatial searching. In *Proc. 1984 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'84)*, pp. 47–57, Boston, MA, June 1984.

[GW07] R. C. Gonzalez and R. E. Woods. *Digital Image Processing* (3rd ed.). Prentice Hall, 2007.

[GZ03a] B. Goethals and M. Zaki. An introduction to workshop frequent itemset mining implementations. In *Proc. ICDM'03 Int. Workshop Frequent Itemset Mining Implementations (FIMI'03)*, pp. 1–13, Melbourne, FL, Nov. 2003.

[GZ03b] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03)*, Melbourne, FL, Nov. 2003.

[HA04] V. J. Hodge, and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126, 2004.

[HAC+99] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. Interactive data analysis: The control project. *IEEE Computer*, 32:51–59, 1999.

[Ham94] J. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

[Han98] J. Han. Towards on-line analytical mining in large databases. *SIGMOD Record*, 27:97–107, 1998.

[Har68] P. E. Hart. The condensed nearest neighbor rule. *IEEE Trans. Information Theory*, 14:515–516, 1968.

[Har72] J. Hartigan. Direct clustering of a data matrix. *J. American Stat. Assoc.*, 67:123–129, 1972.

[Har75] J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, 1975.

[Hay99] S. S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1999.

[Hay08] S. Haykin. *Neural Networks and Learning Machines*. Prentice-Hall, 2008.

[HB87] S. J. Hanson and D. J. Burr. Minkowski-r back-propagation: Learning in connectionist models with non-euclidian error signals. In *Neural Information Proc. Systems Conf.*, pp. 348–357, Denver, CO, 1987.

[HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *J. Intelligent Information Systems*, 17:107–145, 2001.

[HCC93] J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. *IEEE Trans. Knowledge and Data Engineering*, 5:29–40, 1993.

[HCD94] L. B. Holder, D. J. Cook, and S. Djoko. Substructure discovery in the subdue system. In *Proc. AAAI'94 Workshop on Knowledge Discovery in Databases (KDD'94)*, pp. 169–180, Seattle, WA, July 1994.

[Hec96] D. Heckerman. Bayesian networks for knowledge discovery. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 273–305. Cambridge, MA: MIT Press, 1996.

[HF94] J. Han and Y. Fu. Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases. In *Proc. AAAI'94 Workshop Knowledge Discovery in Databases (KDD'94)*, pp. 157–168, Seattle, WA, July 1994.

[HF95] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 420–431, Zurich, Switzerland, Sept. 1995.

[HF96] J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 399–421. AAAI/MIT Press, 1996.

[HFLP01] P. S. Horn, L. Feng, Y. Li, and A. J. Pesce. Effect of outliers and nonhealthy individuals on reference interval estimation. *Clinical Chemistry*, 47:2137–2145, 2001.

[HG05] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proc. 22nd Int. Conf. Machine Learning (ICML'05)*, pp. 297–304, Bonn, Germany, 2005.

[HG07] A. Hinneburg and H.-H. Gabriel. DENCLUE 2.0: Fast clustering based on kernel density estimation. In *Proc. 2007 Int. Conf. Intelligent Data Analysis (IDA'07)*, pp. 70–80, Ljubljana, Slovenia, 2007.

[HGC95] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[HH01] R. J. Hilderman and H. J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic, 2001.

[HHW97] J. Hellerstein, P. Haas, and H. Wang. Online aggregation. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pp. 171–182, Tucson, AZ, May 1997.

[Hig08] R. C. Higgins. *Analysis for Financial Management with S&P Bind-In Card*. Irwin/McGraw-Hill, 2008.

[HK91] P. Hoschka and W. Klösgen. A support system for interpreting statistical data. In G. Piatetsky-Shapiro and W. J. Frawley (eds.), *Knowledge Discovery in Databases*, pp. 325–346. AAAI/MIT Press, 1991.

[HK98] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)*, pp. 58–65, New York, NY, Aug. 1998.

[HKGT03] M. Hadjieleftheriou, G. Kollios, D. Gunopulos, and V. J. Tsotras. Online discovery of dense areas in spatio-temporal databases. In *Proc. 2003 Int. Symp. Spatial and Temporal Databases (SSTD'03)*, pp. 306–324, Santorini Island, Greece, July 2003.

[HKKR99] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. Wiley, 1999.

[HKP91] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991.

[HLW07] W. Hsu, M. L. Lee, and J. Wang. *Temporal and Spatio-Temporal Data Mining*. IGI Publishing, 2007.

[HLZ02] W. Hsu, M. L. Lee, and J. Zhang. Image mining: Trends and developments. *J. Intelligent Information Systems*, 19:7–23, 2002.

[HMM86] J. Hong, I. Mozetic, and R. S. Michalski. Incremental learning of attribute-based descriptions from examples, the method and user's guide. In *Report ISG 85-5, UIUCDCS-F-86-949*, Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[HMS66] E. B. Hunt, J. Marin, and P. T. Stone. *Experiments in Induction*. Academic Press, 1966.

[HMS01] D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining* (*Adaptive Computation and Machine Learning*). Cambridge, MA: MIT Press, 2001.

[HN90] R. Hecht-Nielsen. *Neurocomputing*. Reading, MA: Addison-Wesley, 1990.

[Hor08] R. Horak. *Telecommunications and Data Communications Handbook* (2nd ed.). Wiley-Interscience, 2008.

[HP07] M. Hua and J. Pei. Cleaning disguised missing data: A heuristic approach. In *Proc. 2007 ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining (KDD'07)*, pp. 950–958, San Jose, CA, Aug. 2007.

[HPDW01] J. Han, J. Pei, G. Dong, and K. Wang. Efficient computation of iceberg cubes with complex measures. In *Proc. 2001 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'01)*, pp. 1–12, Santa Barbara, CA, May 2001.

[HPS97] J. Hosking, E. Pednault, and M. Sudan. A statistical perspective on data mining. *Future Generation Computer Systems*, 13:117–134, 1997.

[HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00)*, pp. 1–12, Dallas, TX, May 2000.

[HRMS10] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially-private queries through consistency. In *Proc. 2010 Int. Conf. Very Large Data Bases (VLDB'10)*, pp. 1021–1032, Singapore, Sept. 2010.

[HRU96] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96)*, pp. 205–216, Montreal, Quebec, Canada, June 1996.

[HS05] J. M. Hellerstein and M. Stonebraker. *Readings in Database Systems* (4th ed.). Cambridge, MA: MIT Press, 2005.

[HSG90] S. A. Harp, T. Samad, and A. Guha. Designing application-specific neural networks using the genetic algorithm. In D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems II*, pp. 447–454. Morgan Kaufmann, 1990.

[HT98] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Ann. Statistics*, 26:451–471, 1998.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer Verlag, 2009.

[Hua98] Z. Huang. Extensions to the *k*-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304, 1998.

[Hub94] C. H. Huberty. *Applied Discriminant Analysis*. Wiley-Interscience, 1994.

[Hub96] B. B. Hubbard. *The World According to Wavelets*. A. K. Peters, 1996.

[HWB+04] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. Mining spatial motifs from protein structure graphs. In *Proc. 8th Int. Conf. Research in Computational Molecular Biology (RECOMB)*, pp. 308–315, San Diego, CA, Mar. 2004.

[HXD03] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern Recognition Lett.*, 24:1641–1650, June, 2003.

[IGG03] C. Imhoff, N. Galemmo, and J. G. Geiger. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. John Wiley & Sons, 2003.

[IKA02] T. Imielinski, L. Khachiyan, and A. Abdulghani. Cubegrades: Generalizing association rules. *Data Mining and Knowledge Discovery*, 6:219–258, 2002.

[IM96] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39:58–64, 1996.

[Inm96] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 1996.

[IWM98] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proc. 2000 European Symp. Principles of Data Mining and Knowledge Discovery (PKDD'00)*, pp. 13–23, Lyon, France, Sept. 1998.

[Jac88] R. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307, 1988.

[Jai10] A. K. Jain. Data clustering: 50 years beyond *k*-means. *Pattern Recognition Lett.*, 31(8):651–666, 2010.

[Jam85] M. James. *Classification Algorithms*. John Wiley & Sons, 1985.

[JBD05] X. Ji, J. Bailey, and G. Dong. Mining minimal distinguishing subsequence patterns with gap constraints. In *Proc. 2005 Int. Conf. Data Mining (ICDM'05)*, pp. 194–201, Houston, TX, Nov. 2005.

[JD88] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[Jen96] F. V. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, 1996.

[JL96] G. H. John and P. Langley. Static versus dynamic sampling for data mining. In *Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, pp. 367–370, Portland, OR, Aug. 1996.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A survey. *ACM Computing Surveys*, 31:264–323, 1999.

[Joh97] G. H. John. *Enhancements to the Data Mining Process*. Ph.D. Thesis, Computer Science Department, Stanford University, 1997.

[Joh99] G. H. John. Behind-the-scenes data mining: A report on the KDD-98 panel. *SIGKDD Explorations*, 1:6–8, 1999.

[JP04] N. C. Jones and P. A. Pevzner. *An Introduction to Bioinformatics Algorithms*. Cambridge, MA: MIT Press, 2004.

[JSD+10] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. In *Proc. 2010 European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD'10)*, pp. 570–586, Barcelona, Spain, Sept. 2010.

[JTH01] W. Jin, K. H. Tung, and J. Han. Mining top-*n* local outliers in large databases. In *Proc. 2001 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'01)*, pp. 293–298, San Fransisco, CA, Aug. 2001.

[JTHW06] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proc. 2006 Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'06)*, Singapore, Apr. 2006.

[JW92] R. A. Johnson and D. A. Wichern. *Applied Multivariate Statistical Analysis* (3rd ed.). Prentice-Hall, 1992.

[JW02a] G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In *Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'02)*, pp. 538–543, Edmonton, Alberta, Canada, July 2002.

[JW02b] R. A. Johnson and D. A. Wichern. *Applied Multivariate Statistical Analysis* (5th ed.). Prentice Hall, 2002.

[Kam09] C. Kamath. *Scientific Data Mining: A Practical Perspective*. Society for Industrial and Applied Mathematic (SIAM), 2009.

[Kas80] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29:119–127, 1980.

[KBDM09] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: A kernel approach. *Machine Learning*, 74:1–22, 2009.

[Kec01] V. Kecman. *Learning and Soft Computing*. Cambridge, MA: MIT Press, 2001.

[Kei97] D. A. Keim. Visual techniques for exploring databases. In *Tutorial Notes, 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, Newport Beach, CA, Aug. 1997.

[Ker92] R. Kerber. ChiMerge: Discretization of numeric attributes. In *Proc. 1992 Nat. Conf. Artificial Intelligence (AAAI'92)*, pp. 123–128, San Jose, CA, 1992.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press, 2009.

[KH95] K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. In *Proc. 1995 Int. Symp. Large Spatial Databases (SSD'95)*, pp. 47–66, Portland, ME, Aug. 1995.

[KH97] I. Kononenko and S. J. Hong. Attribute selection for modeling. *Future Generation Computer Systems*, 13:181–195, 1997.

[KH09] M.-S. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. In *Proc. 2009 Int. Conf. Very Large Data Bases (VLDB'09)*, Lyon, France, Aug. 2009.

[KHC97] M. Kamber, J. Han, and J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. In *Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, pp. 207–210, Newport Beach, CA, Aug. 1997.

[KHK99] G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *COMPUTER*, 32:68–75, 1999.

[KHY+08] H. Kargupta, J. Han, P. S. Yu, R. Motwani, and V. Kumar. *Next Generation of Data Mining*. Chapman & Hall/CRC, 2008.

[KJ97] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[KJSY04] H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha. *Data Mining: Next Generation Challenges and Future Directions*. Cambridge, MA: AAAI/MIT Press, 2004.

[KK01] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proc. 2001 Int. Conf. Data Mining (ICDM'01)*, pp. 313–320, San Jose, CA, Nov. 2001.

[KKW+10] H. S. Kim, S. Kim, T. Weninger, J. Han, and T. Abdelzaher. NDPMine: Efficiently mining discriminative numerical features for pattern-based classification. In *Proc. 2010 European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD'10)*, Barcelona, Spain, Sept. 2010.

[KKZ09] H.-P. Kriegel, P. Kroeger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowledge Discovery from Data (TKDD)*, 3(1):1–58, 2009.

[KLA+08] M. Khan, H. Le, H. Ahmadi, T. Abdelzaher, and J. Han. DustMiner: Troubleshooting interactive complexity bugs in sensor networks. In *Proc. 2008 ACM Int. Conf. Embedded Networked Sensor Systems (SenSys'08)*, pp. 99–112, Raleigh, NC, Nov. 2008.

[Kle99] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46: 604–632, 1999.

[KLV+98] R. L. Kennedy, Y. Lee, B. Van Roy, C. D. Reed, and R. P. Lippman. *Solving Data Mining Problems Through Pattern Recognition*. Prentice-Hall, 1998.

[KM90] Y. Kodratoff and R. S. Michalski. *Machine Learning, An Artificial Intelligence Approach*, Vol. 3. Morgan Kaufmann, 1990.

[KM94] J. Kivinen and H. Mannila. The power of sampling in knowledge discovery. In *Proc. 13th ACM Symp. Principles of Database Systems*, pp. 77–85, Minneapolis, MN, May 1994.

[KMN+02] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient *k*-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 24:881–892, 2002.

[KMR+94] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proc. 3rd Int. Conf. Information and Knowledge Management*, pp. 401–408, Gaithersburg, MD, Nov. 1994.

[KMS03] J. Kubica, A. Moore, and J. Schneider. Tractable group detection on large link data sets. In *Proc. 2003 Int. Conf. Data Mining (ICDM'03)*, pp. 573–576, Melbourne, FL, Nov. 2003.

[KN97] E. Knorr and R. Ng. A unified notion of outliers: Properties and computation. In *Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, pp. 219–222, Newport Beach, CA, Aug. 1997.

[KNNL04] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li. *Applied Linear Statistical Models with Student CD*. Irwin, 2004.

[KNT00] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB J.*, 8:237–253, 2000.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. 14th Joint Int. Conf. Artificial Intelligence (IJCAI'95)*, Vol. 2, pp. 1137–1143, Montreal, Quebec, Canada, Aug. 1995.

[Kol93] J. L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[Kon95] I. Kononenko. On biases in estimating multi-valued attributes. In *Proc. 14th Joint Int. Conf. Artificial Intelligence (IJCAI'95)*, Vol. 2, pp. 1034–1040, Montreal, Quebec, Canada, Aug. 1995.

[Kot88] P. Koton. Reasoning about evidence in causal explanation. In *Proc. 7th Nat. Conf. Artificial Intelligence (AAAI'88)*, pp. 256–263, St. Paul, MN, Aug. 1988.

[KPR98] J. M. Kleinberg, C. Papadimitriou, and P. Raghavan. A microeconomic view of data mining. *Data Mining and Knowledge Discovery*, 2:311–324, 1998.

[KPS03] R. M. Karp, C. H. Papadimitriou, and S. Shenker. A simple algorithm for finding frequent elements in streams and bags. *ACM Trans. Database Systems*, 28:51–55, 2003.

[KR90] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[KR02] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (2nd ed.). John Wiley & Sons, 2002.

[KR03] D. Krane and R. Raymer. *Fundamental Concepts of Bioinformatics*. Benjamin Cummings, 2003.

[Kre02] V. Krebs. Mapping networks of terrorist cells. *Connections*, 24:43–52 (Winter), 2002.

[KRR+00] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proc. 2000 IEEE Symp. Foundations of Computer Science (FOCS'00)*, pp. 57–65, Redondo Beach, CA, Nov. 2000.

[KRTM08] R. Kimball, M. Ross, W. Thornthwaite, and J. Mundy. *The Data Warehouse Lifecycle Toolkit*. Hoboken, NJ: John Wiley & Sons, 2008.

[KSZ08] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proc. 2008 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'08)*, pp. 444–452, Las Vegas, NV, Aug. 2008.

[KT99] J. M. Kleinberg and A. Tomkins. Application of linear algebra in information retrieval and hypertext analysis. In *Proc. 18th ACM Symp. Principles of Database Systems (PODS'99)*, pp. 185–193, Philadelphia, PA, May 1999.

[KYB03] I. Korf, M. Yandell, and J. Bedell. *BLAST*. Sebastopol, CA: O'Reilly Media, 2003.

[Lam98] W. Lam. Bayesian network refinement via machine learning approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20:240–252, 1998.

[Lau95] S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

[LCH+09] D. Lo, H. Cheng, J. Han, S. Khoo, and C. Sun. Classification of software behaviors for failure detection: A discriminative pattern mining approach. In *Proc. 2009 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'09)*, pp. 557–566, Paris, France, June 2009.

[LDH+08] C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text cube: Computing IR measures for multidimensional text database analysis. In *Proc. 2008 Int. Conf. Data Mining (ICDM'08)*, pp. 905–910, Pisa, Italy, Dec. 2008.

[LDH+10] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *Proc. 2010 ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'10)*, pp. 1099–1108, Washington, DC, July 2010.

[LDR00] J. Li, G. Dong, and K. Ramamohanrarao. Making use of the most expressive jumping emerging patterns for classification. In *Proc. 2000 Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'00)*, pp. 220–232, Kyoto, Japan, Apr. 2000.

[LDS90] Y. Le Cun, J. S. Denker, and S. A. Solla. Optimal brain damage. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*. Morgan Kaufmann, 1990.

[Lea96] D. B. Leake. CBR in context: The present and future. In D. B. Leake (ed.), *Cased-Based Reasoning: Experiences, Lessons, and Future Directions*, pp. 3–30. AAAI Press, 1996.

[LGT97] S. Lawrence, C. L. Giles, and A. C. Tsoi. Symbolic conversion, grammatical inference and rule extraction for foreign exchange rate prediction. In Y. Abu-Mostafa, A. S. Weigend, and P. N. Refenes (eds.), *Neural Networks in the Capital Markets*. London: World Scientific, 1997.

[LHC97] B. Liu, W. Hsu, and S. Chen. Using general impressions to analyze discovered classification rules. In *Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, pp. 31–36, Newport Beach, CA, Aug. 1997.

[LHF98] H. Lu, J. Han, and L. Feng. Stock movement and *n*-dimensional inter-transaction association rules. In *Proc. 1998 SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery (DMKD'98)*, pp. 12:1–12:7, Seattle, WA, June 1998.

[LHG04] X. Li, J. Han, and H. Gonzalez. High-dimensional OLAP: A minimal cubing approach. In *Proc. 2004 Int. Conf. Very Large Data Bases (VLDB'04)*, pp. 528–539, Toronto, Ontario, Canada, Aug. 2004.

[LHKG07] X. Li, J. Han, S. Kim, and H. Gonzalez. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *Proc. 2007 SIAM Int. Conf. Data Mining (SDM'07)*, Minneapolis, MN, Apr. 2007.

[LHM98] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)*, pp. 80–86, New York, Aug. 1998.

[LHP01] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proc. 2001 Int. Conf. Data Mining (ICDM'01)*, pp. 369–376, San Jose, CA, Nov. 2001.

[LHTD02] H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6:393–423, 2002.

[LHW07] J.-G. Lee, J. Han, and K. Whang. Clustering trajectory data. In *Proc. 2007 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'07)*, Beijing, China, June 2007.

[LHXS06] H. Liu, J. Han, D. Xin, and Z. Shao. Mining frequent patterns on very high dimensional data: A top-down row enumeration approach. In *Proc. 2006 SIAM Int. Conf. Data Mining (SDM'06)*, Bethesda, MD, Apr. 2006.

[LHY+08] X. Li, J. Han, Z. Yin, J.-G. Lee, and Y. Sun. Sampling Cube: A framework for statistical OLAP over sampling data. In *Proc. 2008 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'08)*, pp. 779–790, Vancouver, British Columbia, Canada, June 2008.

[Liu06] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data.* New York: Springer, 2006.

[LJK00] J. Laurikkala, M. Juhola, and E. Kentala. Informal identification of outliers in medical data. In *Proc. 5th Int. Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, Berlin, Germany, Aug. 2000.

[LKCH03] Y.-K. Lee, W.-Y. Kim, Y. D. Cai, and J. Han. CoMine: Efficient mining of correlated patterns. In *Proc. 2003 Int. Conf. Data Mining (ICDM'03)*, pp. 581–584, Melbourne, FL, Nov. 2003.

[LKF05] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proc. 2005 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'05)*, pp. 177–187, Chicago, IL, Aug. 2005.

[LLLY03] G. Liu, H. Lu, W. Lou, and J. X. Yu. On computing, storing and querying frequent patterns. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 607–612, Washington, DC, Aug. 2003.

[LLMZ04] Z. Li, S. Lu, S. Myagmar, and Y. Zhou. CP-Miner: A tool for finding copy-paste and related bugs in operating system code. In *Proc. 2004 Symp. Operating Systems Design and Implementation (OSDI'04)*, pp. 20–22, San Francisco, CA, Dec. 2004.

[Llo57] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28:128–137, 1982 (original version: Technical Report, Bell Labs, 1957).

[LLS00] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40:203–228, 2000.

[LM97] K. Laskey and S. Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Proc. 13th Annual Conf. Uncertainty in Artificial Intelligence*, pp. 334–341, San Francisco, CA, Aug. 1997.

[LM98a] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic, 1998.

[LM98b] H. Liu and H. Motoda (eds.). *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Kluwer Academic, 1998.

[LNHP99] L. V. S. Lakshmanan, R. Ng, J. Han, and A. Pang. Optimization of constrained frequent set queries with 2-variable constraints. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pp. 157–168, Philadelphia, PA, June 1999.

[L-NK03] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. 2003 Int. Conf. Information and Knowledge Management (CIKM'03)*, pp. 556–559, New Orleans, LA, Nov. 2003.

[Los01] D. Loshin. *Enterprise Knowledge Management: The Data Quality Approach*. Morgan Kaufmann, 2001.

[LP97] A. Lenarcik and Z. Piasta. Probabilistic rough classifiers with mixture of discrete and continuous variables. In T. Y. Lin and N. Cercone (eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, pp. 373–383, Kluwer Academic, 1997.

[LPH02] L. V. S. Lakshmanan, J. Pei, and J. Han. Quotient cube: How to summarize the semantics of a data cube. In *Proc. 2002 Int. Conf. Very Large Data Bases (VLDB'02)*, pp. 778–789, Hong Kong, China, Aug. 2002.

[LPWH02] J. Liu, Y. Pan, K. Wang, and J. Han. Mining frequent itemsets by opportunistic projection. In *Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'02)*, pp. 239–248, Edmonton, Alberta, Canada, July 2002.

[LPZ03] L. V. S. Lakshmanan, J. Pei, and Y. Zhao. QC-Trees: An efficient summary structure for semantic OLAP. In *Proc. 2003 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'03)*, pp. 64–75, San Diego, CA, June 2003.

[LS95] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proc. 1995 IEEE Int. Conf. Tools with AI (ICTAI'95)*, pp. 388–391, Washington, DC, Nov. 1995.

[LS97] W. Y. Loh and Y. S. Shih. Split selection methods for classification trees. *Statistica Sinica*, 7:815–840, 1997.

[LSBZ87] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow. *Scientific Discovery: Computational Explorations of the Creative Processes*. Cambridge, MA: MIT Press, 1987.

[LSL95] H. Lu, R. Setiono, and H. Liu. Neurorule: A connectionist approach to data mining. In *Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 478–489, Zurich, Switzerland, Sept. 1995.

[LSW97] B. Lent, A. Swami, and J. Widom. Clustering association rules. In *Proc. 1997 Int. Conf. Data Engineering (ICDE'97)*, pp. 220–231, Birmingham, England, Apr. 1997.

[Lux07] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

[LV88] W. Y. Loh and N. Vanichsetakul. Tree-structured classificaiton via generalized discriminant analysis. *J. American Statistical Association*, 83:715–728, 1988.

[LZ05] Z. Li and Y. Zhou. PR-Miner: Automatically extracting implicit programming rules and detecting violations in large software code. In *Proc. 2005 ACM SIGSOFT Symp. Foundations of Software Engineering (FSE'05)*, Lisbon, Portugal, Sept. 2005.

[MA03] S. Mitra and T. Acharya. *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. John Wiley & Sons, 2003.

[MAE05] A. Metwally, D. Agrawal, and A. El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *Proc. 2005 Int. Conf. Database Theory (ICDT'05)*, pp. 398–412, Edinburgh, Scotland, Jan. 2005.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, 1:281–297, Berkeley, CA, 1967.

[Mag94] J. Magidson. The CHAID approach to segmentation modeling: CHI-squared automatic interaction detection. In R. P. Bagozzi (ed.), *Advanced Methods of Marketing Research*, pp. 118–159. Blackwell Business, 1994.

[Man00] H. Mannila. Theoretical frameworks of data mining. *SIGKDD Explorations*, 1:30–32, 2000.

[MAR96] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. 1996 Int. Conf. Extending Database Technology (EDBT'96)*, pp. 18–32, Avignon, France, Mar. 1996.

[Mar09] S. Marsland. *Machine Learning: An Algorithmic Perspective*. Chapman & Hall/CRC, 2009.

[MB88] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. John Wiley & Sons, 1988.

[MC03] M. V. Mahoney and P. K. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proc. 2003 Int. Conf. Data Mining (ICDM'03)*, Melbourne, FL, Nov. 2003.

[MCK+04] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'04)*, pp. 236–245, Seattle, WA, Aug. 2004.

[MCM83] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. *Machine Learning, An Artificial Intelligence Approach*, Vol. 1. Morgan Kaufmann, 1983.

[MCM86] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. *Machine Learning, An Artificial Intelligence Approach*, Vol. 2. Morgan Kaufmann, 1986.

[MD88] M. Muralikrishna and D. J. DeWitt. Equi-depth histograms for extimating selectivity factors for multi-dimensional queries. In *Proc. 1988 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'88)*, pp. 28–36, Chicago, IL, June 1988.

[Mei03] M. Meilă. Comparing clusterings by the variation of information. In *Proc. 16th Annual Conf. Computational Learning Theory (COLT'03)*, pp. 173–187, Washington, DC, Aug. 2003.

[Mei05] M. Meilă. Comparing clusterings: An axiomatic view. In *Proc. 22nd Int. Conf. Machine Learning (ICML'05)*, pp. 577–584, Bonn, Germany, 2005.

[Men03] J. Mena. *Investigative Data Mining with Security and Criminal Detection*. Butterworth-Heinemann, 2003.

[MFS95] D. Malerba, E. Floriana, and G. Semeraro. A further comparison of simplification methods for decision tree induction. In D. Fisher and H. Lenz (eds.), *Learning from Data: AI and Statistics*. Springer Verlag, 1995.

[MH95] J. K. Martin and D. S. Hirschberg. The time complexity of decision tree induction. In *Technical Report ICS-TR 95-27*, pp. 1–27, Department of Information and Computer Science, University of California, Irvine, CA, Aug. 1995.

[MH09] H. Miller and J. Han. *Geographic Data Mining and Knowledge Discovery* (2nd ed.). Chapman & Hall/CRC, 2009.

[Mic83] R. S. Michalski. A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, pp. 83–134. Morgan Kaufmann, 1983.

[Mic92] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1992.

[Mil98] R. G. Miller. *Survival Analysis*. Wiley-Interscience, 1998.

[Min89] J. Mingers. An empirical comparison of pruning methods for decision-tree induction. *Machine Learning*, 4:227–243, 1989.

[Mir98] B. Mirkin. Mathematical classification and clustering. *J. Global Optimization*, 12:105–108, 1998.

[Mit96] M. Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.

[Mit97] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[MK91] M. Manago and Y. Kodratoff. Induction of decision trees from complex structured data. In G. Piatetsky-Shapiro and W. J. Frawley (eds.), *Knowledge Discovery in Databases*, pp. 289–306. AAAI/MIT Press, 1991.

[MLSZ06] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proc. 15th Int. Conf. World Wide Web (WWW'06)*, pp. 533–542, Edinburgh, Scotland, May 2006.

[MM95] J. Major and J. Mangano. Selecting among rules induced from a hurricane database. *J. Intelligent Information Systems*, 4:39–52, 1995.

[MM02] G. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proc. 2002 Int. Conf. Very Large Data Bases (VLDB'02)*, pp. 346–357, Hong Kong, China, Aug. 2002.

[MN89] M. Mézard and J.-P. Nadal. Learning in feedforward layered networks: The tiling algorithm. *J. Physics*, 22:2191–2204, 1989.

[MO04] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 1(1):24–25, 2004.

[MP69] M. L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969.

[MRA95] M. Metha, J. Rissanen, and R. Agrawal. MDL-based decision tree pruning. In *Proc. 1995 Int. Conf. Knowledge Discovery and Data Mining (KDD'95)*, pp. 216–221, Montreal, Quebec, Canada, Aug. 1995.

[MRS08] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MS03a] M. Markou and S. Singh. Novelty detection: A review—part 1: Statistical approaches. *Signal Processing*, 83:2481–2497, 2003.

[MS03b] M. Markou and S. Singh. Novelty detection: A review—part 2: Neural network based approaches. *Signal Processing*, 83:2499–2521, 2003.

[MST94] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Chichester, England: Ellis Horwood, 1994.

[MT94] R. S. Michalski and G. Tecuci. *Machine Learning, A Multistrategy Approach*, Vol. 4. Morgan Kaufmann, 1994.

[MTV94] H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. In *Proc. AAAI'94 Workshop Knowledge Discovery in Databases (KDD'94)*, pp. 181–192, Seattle, WA, July 1994.

[MTV97] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997.

[Mur98] S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2:345–389, 1998.

[Mut05] S. Muthukrishnan. *Data Streams: Algorithms and Applications*. Now Publishers, 2005.

[MXC+07] Q. Mei, D. Xin, H. Cheng, J. Han, and C. Zhai. Semantic annotation of frequent patterns. *ACM Trans. Knowledge Discovery from Data (TKDD)*, 15:321–348, 2007.

[MY97] R. J. Miller and Y. Yang. Association rules over interval data. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pp. 452–461, Tucson, AZ, May 1997.

[MZ06] Q. Mei and C. Zhai. A mixture model for contextual text mining. In *Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'06)*, pp. 649–655, Philadelphia, PA, Aug. 2006.

[NB86] T. Niblett and I. Bratko. Learning decision rules in noisy domains. In M. A. Brammer (ed.), *Expert Systems '86: Research and Development in Expert Systems III*, pp. 25–34. British Computer Society Specialist Group on Expert Systems, Dec. 1986.

[NBW06] M. Newman, A.-L. Barabasi, and D. J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[NC03] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 631–636, Washington, DC, Aug. 2003.

[New10] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Rev. E*, 69:113–128, 2004.

[NGE-R09] J. Neville, B. Gallaher, and T. Eliassi-Rad. Evaluating statistical tests for within-network classifiers of relational data. In *Proc. 2009 Int. Conf. Data Mining (ICDM'09)*, pp. 397–406, Miami, FL, Dec. 2009.

[NH94] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94)*, pp. 144–155, Santiago, Chile, Sept. 1994.

[NJW01] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems 14*. pp. 849–856, Cambridge, MA: MIT Press, 2001.

[NK04] S. Nijssen and J. Kok. A quick start in frequent structure mining can make a difference. In *Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'04)*, pp. 647–652, Seattle, WA, Aug. 2004.

[NKNW96] J. Neter, M. H. Kutner, C. J. Nachtsheim, and L. Wasserman. *Applied Linear Statistical Models* (4th ed.). Irwin, 1996.

[NLHP98] R. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 13–24, Seattle, WA, June 1998.

[NRS99] A. Natsev, R. Rastogi, and K. Shim. Walrus: A similarity retrieval algorithm for image databases. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pp. 395–406, Philadelphia, PA, June 1999.

[NW99] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

[OFG97] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Proc. 1997 IEEE Workshop Neural Networks for Signal Processing (NNSP'97)*, pp. 276–285, Amelia Island, FL, Sept. 1997.

[OG95] P. O'Neil and G. Graefe. Multi-table joins through bitmapped join indices. *SIGMOD Record*, 24:8–11, Sept. 1995.

[Ols03] J. E. Olson. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann, 2003.

[Omi03] E. Omiecinski. Alternative interest measures for mining associations. *IEEE Trans. Knowledge and Data Engineering*, 15:57–69, 2003.

[OMM+02] L. O'Callaghan, A. Meyerson, R. Motwani, N. Mishra, and S. Guha. Streaming-data algorithms for high-quality clustering. In *Proc. 2002 Int. Conf. Data Engineering (ICDE'02)*, pp. 685–696, San Fransisco, CA, Apr. 2002.

[OQ97] P. O'Neil and D. Quass. Improved query performance with variant indexes. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pp. 38–49, Tucson, AZ, May 1997.

[ORS98] B. Özden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. In *Proc. 1998 Int. Conf. Data Engineering (ICDE'98)*, pp. 412–421, Orlando, FL, Feb. 1998.

[Pag89] G. Pagallo. Learning DNF by decision trees. In *Proc. 1989 Int. Joint Conf. Artificial Intelligence (IJCAI'89)*, pp. 639–644, San Francisco, CA, 1989.

[Paw91] Z. Pawlak. *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer Academic, 1991.

[PB00] J. C. Pinheiro and D. M. Bates. *Mixed Effects Models in S and S-PLUS*. Springer Verlag, 2000.

[PBTL99] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. 7th Int. Conf. Database Theory (ICDT'99)*, pp. 398–416, Jerusalem, Israel, Jan. 1999.

[PCT+03] F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. Zaki. CARPENTER: Finding closed patterns in long biological datasets. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 637–642, Washington, DC, Aug. 2003.

[PCY95a] J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. In *Proc. 1995 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'95)*, pp. 175–186, San Jose, CA, May 1995.

[PCY95b] J. S. Park, M. S. Chen, and P. S. Yu. Efficient parallel mining for association rules. In *Proc. 4th Int. Conf. Information and Knowledge Management*, pp. 31–36, Baltimore, MD, Nov. 1995.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[PHL01] J. Pei, J. Han, and L. V. S. Lakshmanan. Mining frequent itemsets with convertible constraints. In *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, pp. 433–442, Heidelberg, Germany, Apr. 2001.

[PHL+01] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. In *Proc. 2001 Int. Conf. Data Mining (ICDM'01)*, pp. 441–448, San Jose, CA, Nov. 2001.

[PHL04] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *SIGKDD Explorations*, 6:90–105, 2004.

[PHM00] J. Pei, J. Han, and R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *Proc. 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge Discovery (DMKD'00)*, pp. 11–20, Dallas, TX, May 2000.

[PHM-A+01] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, pp. 215–224, Heidelberg, Germany, Apr. 2001.

[PHM-A+04] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixSpan approach. *IEEE Trans. Knowledge and Data Engineering*, 16:1424–1440, 2004.

[PI97] V. Poosala and Y. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97)*, pp. 486–495, Athens, Greece, Aug. 1997.

[PKGF03] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Proc. 2003 Int. Conf. Data Engineering (ICDE'03)*, pp. 315–326, Bangalore, India, Mar. 2003.

[PKMT99] A. Pfeffer, D. Koller, B. Milch, and K. Takusagawa. SPOOK: A system for probabilistic object-oriented knowledge representation. In *Proc. 15th Annual Conf. Uncertainty in Artificial Intelligence (UAI'99)*, pp. 541–550, Stockholm, Sweden, 1999.

[PKZT01] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao. Efficient OLAP operations in spatial data warehouses. In *Proc. 2001 Int. Symp. Spatial and Temporal Databases (SSTD'01)*, pp. 443–459, Redondo Beach, CA, July 2001.

[PL07] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 2007.

[Pla98] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. Smola (eds.), *Advances in Kernel Methods—Support Vector Learning*, pp. 185–208. Cambridge, MA: MIT Press, 1998.

[PP07] A. Patcha, and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer Verlag, 1985.

[P-S91] G. Piatetsky-Shapiro. *Notes AAAI'91 Workshop Knowledge Discovery in Databases (KDD'91)*. Anaheim, CA, July 1991.

[P-SF91] G. Piatetsky-Shapiro and W. J. Frawley. *Knowledge Discovery in Databases.* AAAI/MIT Press, 1991.

[PTCX04] F. Pan, A. K. H. Tung, G. Cong, and X. Xu. COBBLER: Combining column and row enumeration for closed pattern discovery. In *Proc. 2004 Int. Conf. Scientific and Statistical Database Management (SSDBM'04)*, pp. 21–30, Santorini Island, Greece, June 2004.

[PTVF07] W. H. Press, S. A. Teukolosky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 2007.

[PY10] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowledge and Data Engineering*, 22:1345–1359, 2010.

[Pyl99] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[PZC+03] J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. Maple: A fast algorithm for maximal pattern-based clustering. In *Proc. 2003 Int. Conf. Data Mining (ICDM'03)*, pp. 259–266, Melbourne, FL, Dec. 2003.

[QC-J93] J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. In *Proc. 1993 European Conf. Machine Learning (ECML'93)*, pp. 3–20, Vienna, Austria, 1993.

[QR89] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, Mar. 1989.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Qui87] J. R. Quinlan. Simplifying decision trees. *Int. J. Man-Machine Studies*, 27:221–234, 1987.

[Qui88] J. R. Quinlan. An empirical comparison of genetic and decision-tree classifiers. In *Proc. 1988 Int. Conf. Machine Learning (ICML'88)*, pp. 135–141, Ann Arbor, MI, June 1988.

[Qui89] J. R. Quinlan. Unknown attribute values in induction. In *Proc. 1989 Int. Conf. Machine Learning (ICML'89)*, pp. 164–168, Ithaca, NY, June 1989.

[Qui90] J. R. Quinlan. Learning logic definitions from relations. *Machine Learning*, 5:139–166, 1990.

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Qui96] J. R. Quinlan. Bagging, boosting, and C4.5. In *Proc. 1996 Nat. Conf. Artificial Intelligence (AAAI'96)*, Vol. 1, pp. 725–730, Portland, OR, Aug. 1996.

[RA87] E. L. Rissland and K. Ashley. HYPO: A case-based system for trade secret law. In *Proc. 1st Int. Conf. Artificial Intelligence and Law*, pp. 60–66, Boston, MA, May 1987.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[RBKK95] S. Russell, J. Binder, D. Koller, and K. Kanazawa. Local learning in probabilistic networks with hidden variables. In *Proc. 1995 Joint Int. Conf. Artificial Intelligence (IJCAI'95)*, pp. 1146–1152, Montreal, Quebec, Canada, Aug. 1995.

[RC07] R. Ramakrishnan and B.-C. Chen. Exploratory mining in cube space. *Data Mining and Knowledge Discovery*, 15:29–54, 2007.

[Red92] T. Redman. *Data Quality: Management and Technology*. Bantam Books, 1992.

[Red01] T. Redman. *Data Quality: The Field Guide*. Digital Press (Elsevier), 2001.

[RG03] R. Ramakrishnan and J. Gehrke. *Database Management Systems* (3rd ed.). McGraw-Hill, 2003.

[RGN10] L. De Raedt, T. Guns, and S. Nijssen. Constraint programming for data mining and machine learning. In *Proc. 2010 AAAI Conf. Artificial Intelligence (AAAI'10)*, pp. 1671–1675, Atlanta, GA, July 2010.

[RH01] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *Proc. 2001 Int. Conf. Very Large Data Bases (VLDB'01)*, pp. 381–390, Rome, Italy, Sept. 2001.

[RH07] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proc. 2007 Joint Conf. Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*, pp. 410–420, Prague, Czech Republic, June 2007.

[RHS01] J. F. Roddick, K. Hornsby, and M. Spiliopoulou. An updated bibliography of temporal, spatial, and spatio-temporal data mining research. In J. F. Roddick and K. Hornsby (eds.), TSDM 2000, *Lecture Notes in Computer Science 2007*, pp. 147–163. New York: Springer, 2001.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland (eds.), *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.

[Rip96] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[RM86] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.

[RMS98] S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 368–379, New York, Aug. 1998.

[RN95] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.

[RNI09] M. Radovanović, A. Nanopoulos, and M. Ivanović. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In *Proc. 2009 Int. Conf. Machine Learning (ICML'09)*, pp. 865–872, Montreal, Quebec, Canada, June 2009.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Rev.*, 65:386–498, 1958.

[RS89] C. Riesbeck and R. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum, 1989.

[RS97] K. Ross and D. Srivastava. Fast computation of sparse datacubes. In *Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97)*, pp. 116–125, Athens, Greece, Aug. 1997.

[RS98] R. Rastogi and K. Shim. Public: A decision tree classifer that integrates building and pruning. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 404–415, New York, Aug. 1998.

[RS01] F. Ramsey and D. Schafer. *The Statistical Sleuth: A Course in Methods of Data Analysis*. Duxbury Press, 2001.

[RSC98] K. A. Ross, D. Srivastava, and D. Chatziantoniou. Complex aggregation at multiple granularities. In *Proc. Int. Conf. Extending Database Technology (EDBT'98)*, pp. 263–277, Valencia, Spain, Mar. 1998.

[Rus06] J. C. Russ. *The Image Processing Handbook* (5th ed.). CRC Press, 2006.

[SA95] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 407–419, Zurich, Switzerland, Sept. 1995.

[SA96] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proc. 5th Int. Conf. Extending Database Technology (EDBT'96)*, pp. 3–17, Avignon, France, Mar. 1996.

[SAM96] J. Shafer, R. Agrawal, and M. Mehta. SPRINT: A scalable parallel classifier for data mining. In *Proc. 1996 Int. Conf. Very Large Data Bases (VLDB'96)*, pp. 544–555, Bombay, India, Sept. 1996.

[SAM98] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In *Proc. Int. Conf. Extending Database Technology (EDBT'98)*, pp. 168–182, Valencia, Spain, Mar. 1998.

[SBSW99] B. Schölkopf, P. L. Bartlett, A. Smola, and R. Williamson. Shrinking the tube: A new support vector regression algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn (eds.), *Advances in Neural Information Processing Systems 11*, pp. 330–336. Cambridge, MA: MIT Press, 1999.

[SC03] S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Prentice-Hall, 2003.

[Sch86] J. C. Schlimmer. Learning and representation change. In *Proc. 1986 Nat. Conf. Artificial Intelligence (AAAI'86)*, pp. 511–515, Philadelphia, PA, 1986.

[Sch07] S. E. Schaeffer. Graph clustering. *Computer Science Rev.*, 1:27–64, 2007.

[SCZ98] G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 428–439, New York, Aug. 1998.

[SD90] J. W. Shavlik and T. G. Dietterich. *Readings in Machine Learning*. Morgan Kaufmann, 1990.

[SD02] T. Soukup and I. Davidson. *Visual Data Mining: Techniques and Tools for Data Visualization and Mining*. Wiley, 2002.

[SDJL96] D. Srivastava, S. Dar, H. V. Jagadish, and A. V. Levy. Answering queries with aggregation using views. In *Proc. 1996 Int. Conf. Very Large Data Bases (VLDB'96)*, pp. 318–329, Bombay, India, Sept. 1996.

[SDN98] A. Shukla, P. M. Deshpande, and J. F. Naughton. Materialized view selection for multidimensional datasets. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pp. 488–499, New York, Aug. 1998.

[SE10] G. Seni and J. F. Elder. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan and Claypool, 2010.

[Set10] B. Settles. Active learning literature survey. In *Computer Sciences Technical Report 1648*, University of Wisconsin–Madison, 2010.

[SF86] J. C. Schlimmer and D. Fisher. A case study of incremental concept induction. In *Proc. 1986 Nat. Conf. Artificial Intelligence (AAAI'86)*, pp. 496–501, Philadelphia, PA, 1986.

[SFB99] J. Shanmugasundaram, U. M. Fayyad, and P. S. Bradley. Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, pp. 223–232, San Diego, CA, Aug. 1999.

[SG92] P. Smyth and R. M. Goodman. An information theoretic approach to rule induction. *IEEE Trans. Knowledge and Data Engineering*, 4:301–316, 1992.

[She31] W. A. Shewhart. *Economic Control of Quality of Manufactured Product*. D. Van Nostrand, 1931.

[Shi99] Y.-S. Shih. Families of splitting criteria for classification trees. *Statistics and Computing*, 9:309–315, 1999.

[SHK00] N. Stefanovic, J. Han, and K. Koperski. Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Trans. Knowledge and Data Engineering*, 12:938–958, 2000.

[Sho97] A. Shoshani. OLAP and statistical databases: Similarities and differences. In *Proc. 16th ACM Symp. Principles of Database Systems*, pp. 185–196, Tucson, AZ, May 1997.

[Shu88] R. H. Shumway. *Applied Statistical Time Series Analysis*. Prentice-Hall, 1988.

[SHX04] Z. Shao, J. Han, and D. Xin. MM-Cubing: Computing iceberg cubes by factorizing the lattice space. In *Proc. 2004 Int. Conf. Scientific and Statistical Database Management (SSDBM'04)*, pp. 213–222, Santorini Island, Greece, June 2004.

[SHZ+09] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. RankClus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proc. 2009 Int. Conf. Extending Data Base Technology (EDBT'09)*, pp. 565–576, Saint Petersburg, Russia, Mar. 2009.

[Sil10] F. Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4:1–174, 2010.

[SK08] J. Shieh and E. Keogh. iSAX: Indexing and mining terabyte sized time series. In *Proc. 2008 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'08)*, pp. 623–631, Las Vegas, NV, Aug. 2008.

[SKS10] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts* (6th ed.). McGraw-Hill, 2010.

[SLT+01] S. Shekhar, C.-T. Lu, X. Tan, S. Chawla, and R. R. Vatsavai. Map cube: A visualization tool for spatial data warehouses. In H. J. Miller and J. Han (eds.), *Geographic Data Mining and Knowledge Discovery*, pp. 73–108. Taylor and Francis, 2001.

[SM97] J. C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Co., 1997.

[SMT91] J. W. Shavlik, R. J. Mooney, and G. G. Towell. Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning*, 6:111–144, 1991.

[SN88] K. Saito and R. Nakano. Medical diagnostic expert system based on PDP model. In *Proc. 1988 IEEE Int. Conf. Neural Networks*, pp. 225–262, San Mateo, CA, 1988.

[SOMZ96] W. Shen, K. Ong, B. Mitbander, and C. Zaniolo. Metaqueries for data mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 375–398. AAAI/MIT Press, 1996.

[SON95] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95)*, pp. 432–443, Zurich, Switzerland, Sept. 1995.

[SON98] A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proc. 1998 Int. Conf. Data Engineering (ICDE'98)*, pp. 494–502, Orlando, FL, Feb. 1998.

[SR81] R. Sokal and F. Rohlf. *Biometry*. Freeman, 1981.

[SR92] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In R. Slowinski (ed.), *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Set Theory*, pp. 331–362. Kluwer Academic, 1992.

[SS88] W. Siedlecki and J. Sklansky. On automatic feature selection. *Int. J. Pattern Recognition and Artificial Intelligence*, 2:197–220, 1988.

[SS94] S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. In *Proc. 1994 Int. Conf. Data Engineering (ICDE'94)*, pp. 328–336, Houston, TX, Feb. 1994.

[SS01] G. Sathe and S. Sarawagi. Intelligent rollups in multidimensional OLAP data. In *Proc. 2001 Int. Conf. Very Large Data Bases (VLDB'01)*, pp. 531–540, Rome, Italy, Sept. 2001.

[SS05] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications*. New York: Springer, 2005.

[ST96] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. Knowledge and Data Engineering*, 8:970–974, Dec. 1996.

[STA98] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 343–354, Seattle, WA, June 1998.

[STH+10] Y. Sun, J. Tang, J. Han, M. Gupta, and B. Zhao. Community evolution detection in dynamic heterogeneous information networks. In *Proc. 2010 KDD Workshop Mining and Learning with Graphs (MLG'10)*, Washington, DC, July 2010.

[Ste72] W. Stefansky. Rejecting outliers in factorial designs. *Technometrics*, 14:469–479, 1972.

[Sto74] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. Royal Statistical Society*, 36:111–147, 1974.

[SVA97] R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In *Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, pp. 67–73, Newport Beach, CA, Aug. 1997.

[SW49] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[Swe88] J. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

[Swi98] R. Swiniarski. Rough sets and principal component analysis and their applications in feature extraction and selection, data model building and classification. In S. K. Pal and A. Skowron (eds.), *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, Springer Verlag, Singapore, 1999.

[SWJR07] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Trans. on Knowledge and Data Engineering*, 19(5):631–645, 2007.

[SZ04] D. Shasha and Y. Zhu. *High Performance Discovery in Time Series: Techniques and Case Studies*. New York: Springer, 2004.

[TD02] D. M. J. Tax and R. P. W. Duin. Using two-class classifiers for multiclass classification. In *Proc. 16th Intl. Conf. Pattern Recognition (ICPR'2002)*, pp. 124–127, Montreal, Quebec, Canada, 2002.

[TFPL04] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proc. 2004 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04)*, pp. 611–622, Paris, France, June 2004.

[TG01] I. Tsoukatos and D. Gunopulos. Efficient mining of spatiotemporal patterns. In *Proc. 2001 Int. Symp. Spatial and Temporal Databases (SSTD'01)*, pp. 425–442, Redondo Beach, CA, July 2001.

[THH01] A. K. H. Tung, J. Hou, and J. Han. Spatial clustering in the presence of obstacles. In *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, pp. 359–367, Heidelberg, Germany, Apr. 2001.

[THLN01] A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-based clustering in large databases. In *Proc. 2001 Int. Conf. Database Theory (ICDT'01)*, pp. 405–419, London, Jan. 2001.

[THP08] Y. Tian, R. A. Hankins, and J. M. Patel. Efficient aggregation for graph summarization. In *Proc. 2008 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'08)*, pp. 567–580, Vancouver, British Columbia, Canada, June 2008.

[Thu04] B. Thuraisingham. Data mining for counterterrorism. In H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), *Data Mining: Next Generation Challenges and Future Directions*, pp. 157–183. AAAI/MIT Press, 2004.

[TK08] S. Theodoridis and K. Koutroumbas. *Pattern Recognition* (4th ed.) Academic Press, 2008.

[TKS02] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'02)*, pp. 32–41, Edmonton, Alberta, Canada, July 2002.

[TLZN08] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *Proc. 2008 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'08)*, pp. 677–685, Las Vegas, NV, Aug. 2008.

[Toi96] H. Toivonen. Sampling large databases for association rules. In *Proc. 1996 Int. Conf. Very Large Data Bases (VLDB'96)*, pp. 134–145, Bombay, India, Sept. 1996.

[TS93] G. G. Towell and J. W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13:71–101, Oct. 1993.

[TSK05] P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Boston: Addison-Wesley, 2005.

[TSS04] A. Tanay, R. Sharan, and R. Shamir. Biclustering algorithms: A survey. In S. Aluru (ed.), *Handbook of Computational Molecular Biology*, pp. 26:1–26:17. London: Chapman & Hall, 2004.

[Tuf83] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

[Tuf90] E. R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[Tuf97] E. R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, 1997.

[Tuf01] E. R. Tufte. *The Visual Display of Quantitative Information* (2nd ed.). Graphics Press, 2001.

[TXZ06] Y. Tao, X. Xiao, and S. Zhou. Mining distance-based outliers from large databases in any metric space. In *Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'06)*, pp. 394–403, Philadelphia, PA, Aug. 2006.

[UBC97] P. E. Utgoff, N. C. Berkman, and J. A. Clouse. Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29:5–44, 1997.

[UFS91] R. Uthurusamy, U. M. Fayyad, and S. Spangler. Learning useful rules from inconclusive data. In G. Piatetsky-Shapiro and W. J. Frawley (eds.), *Knowledge Discovery in Databases*, pp. 141–157. AAAI/MIT Press, 1991.

[Utg88] P. E. Utgoff. An incremental ID3. In *Proc. Fifth Int. Conf. Machine Learning (ICML'88)*, pp. 107–120, San Mateo, CA, 1988.

[Val87] P. Valduriez. Join indices. *ACM Trans. Database Systems*, 12:218–246, 1987.

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[Vap98] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.

[VC03] J. Vaidya and C. Clifton. Privacy-preserving *k*-means clustering over vertically partitioned data. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, Washington, DC, Aug 2003.

[VC06] M. Vuk and T. Curk. ROC curve, lift chart and calibration plot. *Metodološki zvezki*, 3:89–108, 2006.

[VCZ10] J. Vaidya, C. W. Clifton, and Y. M. Zhu. *Privacy Preserving Data Mining*. New York: Springer, 2010.

[VGK02] M. Vlachos, D. Gunopulos, and G. Kollios. Discovering similar multidimensional trajectories. In *Proc. 2002 Int. Conf. Data Engineering (ICDE'02)*, pp. 673–684, San Fransisco, CA, Apr. 2002.

[VMZ06] A. Veloso, W. Meira, and M. Zaki. Lazy associative classificaiton. In *Proc. 2006 Int. Conf. Data Mining (ICDM'06)*, pp. 645–654, Hong Kong, China, 2006.

[vR90] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1990.

[VWI98] J. S. Vitter, M. Wang, and B. R. Iyer. Data cube approximation and histograms via wavelets. In *Proc. 1998 Int. Conf. Information and Knowledge Management (CIKM'98)*, pp. 96–104, Washington, DC, Nov. 1998.

[Wat95] M. S. Waterman. *Introduction to Computational Biology: Maps, Sequences, and Genomes (Interdisciplinary Statistics)*. CRC Press, 1995.

[Wat03] D. J. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, 2003.

[WB98] C. Westphal and T. Blaxton. *Data Mining Solutions: Methods and Tools for Solving Real-World Problems*. John Wiley & Sons, 1998.

[WCH10] T. Wu, Y. Chen, and J. Han. Re-examination of interestingness measures in pattern mining: A unified framework. *Data Mining and Knowledge Discovery*, 21(3):371–397, 2010.

[WCRS01] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained *k*-means clustering with background knowledge. In *Proc. 2001 Int. Conf. Machine Learning (ICML'01)*, pp. 577–584, Williamstown, MA, June 2001.

[Wei04] G. M. Weiss. Mining with rarity: A unifying framework. *SIGKDD Explorations*, 6:7–19, 2004.

[WF94] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[WF05] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). Morgan Kaufmann, 2005.

[WFH11] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations* (3rd ed.). Boston: Morgan Kaufmann, 2011.

[WFYH03] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 226–235, Washington, DC, Aug. 2003.

[WHH00] K. Wang, Y. He, and J. Han. Mining frequent itemsets using support constraints. In *Proc. 2000 Int. Conf. Very Large Data Bases (VLDB'00)*, pp. 43–52, Cairo, Egypt, Sept. 2000.

[WHJ+10] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo. Mining advisor-advisee relationships from research publication networks. In *Proc. 2010 ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'10)*, Washington, DC, July 2010.

[WHLT05] J. Wang, J. Han, Y. Lu, and P. Tzvetkov. TFP: An efficient algorithm for mining top-*k* frequent closed itemsets. *IEEE Trans. Knowledge and Data Engineering*, 17:652–664, 2005.

[WHP03] J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the best strategies for mining frequent closed itemsets. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 236–245, Washington, DC, Aug. 2003.

[WI98] S. M. Weiss and N. Indurkhya. *Predictive Data Mining*. Morgan Kaufmann, 1998.

[Wid95] J. Widom. Research problems in data warehousing. In *Proc. 4th Int. Conf. Information and Knowledge Management*, pp. 25–30, Baltimore, MD, Nov. 1995.

[WIZD04] S. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. New York: Springer, 2004.

[WK91] S. M. Weiss and C. A. Kulikowski. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann, 1991.

[WK05] J. Wang and G. Karypis. HARMONY: Efficiently mining the best rules for classification. In *Proc. 2005 SIAM Conf. Data Mining (SDM'05)*, pp. 205–216, Newport Beach, CA, Apr. 2005.

[WLFY02] W. Wang, H. Lu, J. Feng, and J. X. Yu. Condensed cube: An effective approach to reducing data cube size. In *Proc. 2002 Int. Conf. Data Engineering (ICDE'02)*, pp. 155–165, San Fransisco, CA, Apr. 2002.

[WRL94] B. Widrow, D. E. Rumelhart, and M. A. Lehr. Neural networks: Applications in industry, business and science. *Communications of the ACM*, 37:93–105, 1994.

[WSF95] R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. *IEEE Trans. Knowledge and Data Engineering*, 7:623–640, 1995.

[Wu83] C. F. J. Wu. On the convergence properties of the EM algorithm. *Ann. Statistics*, 11:95–103, 1983.

[WW96] Y. Wand and R. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39:86–95, 1996.

[WWYY02] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proc. 2002 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'02)*, pp. 418–427, Madison, WI, June 2002.

[WXH08] T. Wu, D. Xin, and J. Han. ARCube: Supporting ranking aggregate queries in partially materialized data cubes. In *Proc. 2008 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'08)*, pp. 79–92, Vancouver, British Columbia, Canada, June 2008.

[WXMH09] T. Wu, D. Xin, Q. Mei, and J. Han. Promotion analysis in multi-dimensional space. In *Proc. 2009 Int. Conf. Very Large Data Bases (VLDB'09)*, 2(1):109–120, Lyon, France, Aug. 2009.

[WYM97] W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97)*, pp. 186–195, Athens, Greece, Aug. 1997.

[XCYH06] D. Xin, H. Cheng, X. Yan, and J. Han. Extracting redundancy-aware top-k patterns. In *Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'06)*, pp. 444–453, Philadelphia, PA, Aug. 2006.

[XHCL06] D. Xin, J. Han, H. Cheng, and X. Li. Answering top-k queries with multi-dimensional selections: The ranking cube approach. In *Proc. 2006 Int. Conf. Very Large Data Bases (VLDB'06)*, pp. 463–475, Seoul, Korea, Sept. 2006.

[XHLW03] D. Xin, J. Han, X. Li, and B. W. Wah. Star-cubing: Computing iceberg cubes by top-down and bottom-up integration. In *Proc. 2003 Int. Conf. Very Large Data Bases (VLDB'03)*, pp. 476–487, Berlin, Germany, Sept. 2003.

[XHSL06] D. Xin, J. Han, Z. Shao, and H. Liu. C-cubing: Efficient computation of closed cubes by aggregation-based checking. In *Proc. 2006 Int. Conf. Data Engineering (ICDE'06)*, p. 4, Atlanta, GA, Apr. 2006.

[XHYC05] D. Xin, J. Han, X. Yan, and H. Cheng. Mining compressed frequent-pattern sets. In *Proc. 2005 Int. Conf. Very Large Data Bases (VLDB'05)*, pp. 709–720, Trondheim, Norway, Aug. 2005.

[XOJ00] Y. Xiang, K. G. Olesen, and F. V. Jensen. Practical issues in modeling large diagnostic systems with multiply sectioned Bayesian networks. *Intl. J. Pattern Recognition and Artificial Intelligence (IJPRAI)*, 14:59–71, 2000.

[XPK10] Z. Xing, J. Pei, and E. Keogh. A brief survey on sequence classification. *SIGKDD Explorations*, 12:40–48, 2010.

[XSH+04] H. Xiong, S. Shekhar, Y. Huang, V. Kumar, X. Ma, and J. S. Yoo. A framework for discovering co-location patterns in data sets with extended spatial objects. In *Proc. 2004 SIAM Int. Conf. Data Mining (SDM'04)*, Lake Buena Vista, FL, Apr. 2004.

[XYFS07] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. SCAN: A structural clustering algorithm for networks. In *Proc. 2007 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'07)*, pp. 824–833, San Jose, CA, Aug. 2007.

[XZYL08] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long. Evolutionary clustering by hierarchical Dirichlet process with hidden Markov state. In *Proc. 2008 Int. Conf. Data Mining (ICDM'08)*, pp. 658–667, Pisa, Italy, Dec. 2008.

[YC01] N. Ye and Q. Chen. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International*, 17:105–112, 2001.

[YCHX05] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: A profile-based approach. In *Proc. 2005 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'05)*, pp. 314–323, Chicago, IL, Aug. 2005.

[YFB01] C. Yang, U. Fayyad, and P. S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In *Proc. 2001 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'01)*, pp. 194–203, San Fransisco, CA, Aug. 2001.

[YFM+97] K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. In *Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, pp. 96–103, Newport Beach, CA, Aug. 1997.

[YH02] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proc. 2002 Int. Conf. Data Mining (ICDM'02)*, pp. 721–724, Maebashi, Japan, Dec. 2002.

[YH03a] X. Yan and J. Han. CloseGraph: Mining closed frequent graph patterns. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 286–295, Washington, DC, Aug. 2003.

[YH03b] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *Proc. 2003 SIAM Int. Conf. Data Mining (SDM'03)*, pp. 331–335, San Fransisco, CA, May 2003.

[YHA03] X. Yan, J. Han, and R. Afshar. CloSpan: Mining closed sequential patterns in large datasets. In *Proc. 2003 SIAM Int. Conf. Data Mining (SDM'03)*, pp. 166–177, San Fransisco, CA, May 2003.

[YHF10] P. S. Yu, J. Han, and C. Faloutsos. *Link Mining: Models, Algorithms and Applications*. New York: Springer, 2010.

[YHY05] X. Yin, J. Han, and P. S. Yu. Cross-relational clustering with user's guidance. In *Proc. 2005 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'05)*, pp. 344–353, Chicago, IL, Aug. 2005.

[YHY07] X. Yin, J. Han, and P. S. Yu. Object distinction: Distinguishing objects with identical names by link analysis. In *Proc. 2007 Int. Conf. Data Engineering (ICDE'07)*, Istanbul, Turkey, Apr. 2007.

[YHY08] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the Web. *IEEE Trans. Knowledge and Data Engineering*, 20:796–808, 2008.

[YHYY04] X. Yin, J. Han, J. Yang, and P. S. Yu. CrossMine: Efficient classification across multiple database relations. In *Proc. 2004 Int. Conf. Data Engineering (ICDE'04)*, pp. 399–410, Boston, MA, Mar. 2004.

[YK09] L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In *Proc. 2009 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'09)*, pp. 947–956, Paris, France, June 2009.

[YWY07] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: From visual words to visual phrases. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'07)*, pp. 1–8, Minneapolis, MN, June 2007.

[YYH03] H. Yu, J. Yang, and J. Han. Classifying large data sets using SVM with hierarchical clusters. In *Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03)*, pp. 306–315, Washington, DC, Aug. 2003.

[YYH05] X. Yan, P. S. Yu, and J. Han. Graph indexing based on discriminative frequent structure analysis. *ACM Trans. Database Systems*, 30:960–993, 2005.

[YZ94] R. R. Yager and L. A. Zadeh. *Fuzzy Sets, Neural Networks and Soft Computing*. Van Nostrand Reinhold, 1994.

[YZYH06] X. Yan, F. Zhu, P. S. Yu, and J. Han. Feature-based substructure similarity search. *ACM Trans. Database Systems*, 31:1418–1453, 2006.

[Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[Zad83] L. Zadeh. Commonsense knowledge representation based on fuzzy logic. *Computer*, 16:61–65, 1983.

[Zak00] M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowledge and Data Engineering*, 12:372–390, 2000.

[Zak01] M. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 40:31–60, 2001.

[ZDN97] Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pp. 159–170, Tucson, AZ, May 1997.

[ZH02] M. J. Zaki and C. J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proc. 2002 SIAM Int. Conf. Data Mining (SDM'02)*, pp. 457–473, Arlington, VA, Apr. 2002.

[Zha08] C. Zhai. *Statistical Language Models for Information Retrieval*. Morgan and Claypool, 2008.

[ZHL+98] O. R. Zaïane, J. Han, Z. N. Li, J. Y. Chiang, and S. Chee. MultiMedia-Miner: A system prototype for multimedia data mining. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pp. 581–583, Seattle, WA, June 1998.

[Zhu05] X. Zhu. Semi-supervised learning literature survey. In *Computer Sciences Technical Report 1530*, University of Wisconsin–Madison, 2005.

[ZHZ00] O. R. Zaïane, J. Han, and H. Zhu. Mining recurrent items in multimedia with progressive resolution refinement. In *Proc. 2000 Int. Conf. Data Engineering (ICDE'00)*, pp. 461–470, San Diego, CA, Feb. 2000.

[Zia91] W. Ziarko. The discovery, analysis, and representation of data dependencies in databases. In G. Piatetsky-Shapiro and W. J. Frawley (eds.), *Knowledge Discovery in Databases*, pp. 195–209. AAAI Press, 1991.

[ZL06] Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowledge and Data Engineering*, 18:63–77, 2006.

[ZPOL97] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. *Data Mining and Knowledge Discovery*, 1:343–374, 1997.

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96)*, pp. 103–114, Montreal, Quebec, Canada, June 1996.

[ZS02] N. Zapkowicz and S. Stephen. The class imbalance program: A systematic study. *Intelligence Data Analysis*, 6:429–450, 2002.

[ZYH+07] F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng. Mining colossal frequent patterns by core pattern fusion. In *Proc. 2007 Int. Conf. Data Engineering (ICDE'07)*, pp. 706–715, Istanbul, Turkey, Apr. 2007.

[ZYHY07] F. Zhu, X. Yan, J. Han, and P. S. Yu. gPrune: A constraint pushing framework for graph pattern mining. In *Proc. 2007 Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'07)*, pp. 388–400, Nanjing, China, May 2007.

[ZZ09] Z. Zhang and R. Zhang. *Multimedia Data Mining: A Systematic Introduction to Concepts and Theory*. Chapman & Hall, 2009.

[ZZH09] D. Zhang, C. Zhai, and J. Han. Topic cube: Topic modeling for OLAP on multi-dimensional text databases. In *Proc. 2009 SIAM Int. Conf. Data Mining (SDM'09)*, pp. 1123–1134, Sparks, NV, Apr. 2009.

This page intentionally left blank

Index

B

C

D

E

F

G

H

I

M

N

P

Q

R

S

T