

1.8 Summary

- ❖ *Necessity is the mother of invention.* With the mounting growth of data in every application; data mining meets the imminent need for effective, scalable, and flexible data analysis in our society. Data mining can be considered as a natural evolution of information technology and a confluence of several related disciplines and application domains.
- ❖ **Data mining** is the process of discovering interesting patterns from massive amounts of data. As a *knowledge discovery process*, it typically involves data cleaning, data integration, data selection, data transformation, pattern discovery, pattern evaluation, and knowledge presentation.
- ❖ A pattern is *interesting* if it is valid on test data with some degree of certainty, novel, potentially useful (e.g., can be acted on or validates a hunch about which the user was curious), and easily understood by humans. Interesting patterns represent **knowledge**. Measures of **pattern interestingness**, either *objective* or *subjective*, can be used to guide the discovery process.
- ❖ We present a **multidimensional view** of data mining. The major dimensions are **data, knowledge, technologies, and applications**.
- ❖ Data mining can be conducted on any kind of **data** as long as the data are meaningful for a target application, such as database data, data warehouse data, transactional data, and advanced data types. Advanced data types include time-related or sequence data, data streams, spatial and spatiotemporal data, text and multimedia data, graph and networked data, and Web data.
- ❖ A **data warehouse** is a repository for long-term storage of data from multiple sources, organized so as to facilitate management decision making. The data are stored under a unified schema and are typically summarized. Data warehouse systems provide multidimensional data analysis capabilities, collectively referred to as **online analytical processing**.
- ❖ **Multidimensional data mining** (also called **exploratory multidimensional data mining**) integrates core data mining

techniques with OLAP-based multidimensional analysis. It searches for interesting patterns among multiple combinations of dimensions (attributes) at varying levels of abstraction, thereby exploring multidimensional data space.

- ❖ **Data mining functionalities** are used to specify the kinds of patterns or **knowledge** to be found in data mining tasks. The functionalities include characterization and discrimination; the mining of frequent patterns, associations, and correlations; classification and regression; cluster analysis; and outlier detection. As new types of data, new applications, and new analysis demands continue to emerge, there is no doubt we will see more and more novel data mining tasks in the future.
- ❖ Data mining, as a highly application-driven domain, has incorporated **technologies** from many other domains. These include statistics, machine learning, database and data warehouse systems, and information retrieval. The **interdisciplinary nature of data mining research and development** contributes significantly to the success of data mining and its extensive applications.
- ❖ Data mining has many successful **applications**, such as business intelligence, Web search, bioinformatics, health informatics, finance, digital libraries, and digital governments.
- ❖ There are many challenging **issues in data mining research**. Areas include mining methodology, user interaction, efficiency and scalability, and dealing with diverse data types. Data mining research has strongly impacted society and will continue to do so in the future.

2.5 Summary

- ❖ Data sets are made up of data objects. A **data object** represents an entity. Data objects are described by attributes. Attributes can be nominal, binary, ordinal, or numeric.
- ❖ The values of a **nominal** (or **categorical**) **attribute** are symbols or names of things, where each value represents some kind of category, code, or state.

- ❖ **Binary attributes** are nominal attributes with only two possible states (such as 1 and 0 or true and false). If the two states are equally important, the attribute is *symmetric*; otherwise it is *asymmetric*.
- ❖ An **ordinal attribute** is an attribute with possible values that have a meaningful order or ranking among them, but the magnitude between successive values is not known.
- ❖ A **numeric attribute** is *quantitative* (i.e., it is a measurable quantity) represented in integer or real values. Numeric attribute types can be *interval-scaled* or *ratio scaled*. The values of an **interval-scaled attribute** are measured in fixed and equal units. **Ratio-scaled attributes** are numeric attributes with an inherent zero-point. Measurements are ratio-scaled in that we can speak of values as being an order of magnitude larger than the unit of measurement.
- ❖ **Basic statistical descriptions** provide the analytical foundation for data preprocessing. The basic statistical measures for data summarization include *mean*, *weighted mean*, *median*, and *mode* for measuring the central tendency of data; and *range*, *quantiles*, *quartiles*, *interquartile range*, *variance*, and *standard deviation* for measuring the dispersion of data. Graphical representations (e.g., *boxplots*, *quantile plots*, *quantile–quantile plots*, *histograms*, and *scatter plots*) facilitate visual inspection of the data and are thus useful for data preprocessing and mining.
- ❖ **Data visualization** techniques may be *pixel-oriented*, *geometric-based*, *icon-based*, or *hierarchical*. These methods apply to multidimensional relational data. Additional techniques have been proposed for the visualization of complex data, such as text and social networks.
- ❖ Measures of object **similarity** and **dissimilarity** are used in data mining applications such as clustering, outlier analysis, and nearest-neighbor classification. Such measures of *proximity* can be computed for each attribute type studied in this chapter, or for combinations of such attributes. Examples include the *Jaccard coefficient* for asymmetric binary attributes and *Euclidean*, *Manhattan*, *Minkowski*, and *supremum* distances for numeric attributes. For applications involving sparse numeric data vectors, such as term-frequency vectors, the *cosine measure* and the *Tanimoto coefficient* are often used in the assessment of similarity.

3.6 Summary

- ❖ **Data quality** is defined in terms of *accuracy, completeness, consistency, timeliness, believability, and interpretability*. These qualities are assessed based on the intended use of the data.
- ❖ **Data cleaning** routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. Data cleaning is usually performed as an iterative two-step process consisting of discrepancy detection and data transformation.
- ❖ **Data integration** combines data from multiple sources to form a coherent data store. The resolution of semantic heterogeneity, metadata, correlation analysis, tuple duplication detection, and data conflict detection contribute to smooth data integration.
- ❖ **Data reduction** techniques obtain a reduced representation of the data while minimizing the loss of information content. These include methods of *dimensionality reduction, numerosity reduction, and data compression*. **Dimensionality reduction** reduces the number of random variables or attributes under consideration. Methods include *wavelet transforms, principal components analysis, attribute subset selection, and attribute creation*. **Numerosity reduction** methods use parametric or nonparametric models to obtain smaller representations of the original data. Parametric models store only the model parameters instead of the actual data. Examples include regression and log-linear models. Nonparametric methods include histograms, clustering, sampling, and data cube aggregation. **Data compression** methods apply transformations to obtain a reduced or “compressed” representation of the original data. The data reduction is *lossless* if the original data can be reconstructed from the compressed data without any loss of information; otherwise, it is *lossy*.
- ❖ **Data transformation** routines convert the data into appropriate forms for mining. For example, in **normalization**, attribute data are scaled so as to fall within a small range such as 0.0 to 1.0. Other examples are **data discretization** and **concept hierarchy generation**.
- ❖ **Data discretization** transforms numeric data by mapping values to interval or concept labels. Such methods can be used to automatically generate *concept hierarchies* for the data, which allows for mining at

multiple levels of granularity. Discretization techniques include binning, histogram analysis, cluster analysis, decision tree analysis, and correlation analysis. For nominal data, **concept hierarchies** may be generated based on schema definitions as well as the number of distinct values per attribute.

- ❖ Although numerous methods of data preprocessing have been developed, data preprocessing remains an active area of research, due to the huge amount of inconsistent or dirty data and the complexity of the problem.

4.6 Summary

- ❖ A **data warehouse** is a *subject-oriented, integrated, time-variant, and nonvolatile* data collection organized in support of management decision making. Several factors distinguish data warehouses from operational databases. Because the two systems provide quite different functionalities and require different kinds of data, it is necessary to maintain data warehouses separately from operational databases.
- ❖ Data warehouses often adopt a **three-tier architecture**. The bottom tier is a *warehouse database server*, which is typically a relational database system. The middle tier is an *OLAP server*, and the top tier is a *client* that contains query and reporting tools.
- ❖ A data warehouse contains **back-end tools and utilities** for populating and refreshing the warehouse. These cover data extraction, data cleaning, data transformation, loading, refreshing, and warehouse management.
- ❖ Data warehouse **metadata** are data defining the warehouse objects. A metadata repository provides details regarding the warehouse structure, data history, the algorithms used for summarization, mappings from the source data to the warehouse form, system performance, and business terms and issues.
- ❖ A **multidimensional data model** is typically used for the design of corporate *data warehouses* and *departmental data marts*. Such a model can adopt a *star schema*, *snowflake schema*, or *fact constellation schema*. The core of the *multidimensional model* is the **data cube**, which consists of a large set of *facts* (or *measures*) and a

number of *dimensions*. Dimensions are the entities or perspectives with respect to which an organization wants to keep records and are hierarchical in nature.

- ❖ A data cube consists of a **lattice of cuboids**, each corresponding to a different degree of summarization of the given multidimensional data.
- ❖ **Concept hierarchies** organize the values of attributes or dimensions into gradual abstraction levels. They are useful in mining at multiple abstraction levels.
- ❖ **Online analytical processing** can be performed in data warehouses/marts using the multidimensional data model. Typical OLAP operations include *roll-up*, and *drill-down*, *across*, *through*, *slice-and-dice*, and *pivot .rotate/*, as well as statistical operations such as ranking and computing moving averages and growth rates. OLAP operations can be implemented efficiently using the data cube structure.
- ❖ Data warehouses are used for *information processing* (querying and reporting), *analytical processing* (which allows users to navigate through summarized and detailed data by OLAP operations), and *data mining* (which supports knowledge discovery). OLAP-based data mining is referred to as **multidimensional data mining** (also known as exploratory multidimensional data mining, online analytical mining, or OLAM). It emphasizes the interactive and exploratory nature of data mining.
- ❖ OLAP servers may adopt a **relational OLAP (ROLAP)**, a **multidimensional OLAP (MOLAP)**, or a **hybrid OLAP (HOLAP)** implementation. A ROLAP server uses an extended relational DBMS that maps OLAP operations on multidimensional data to standard relational operations. A MOLAP server maps multidimensional data views directly to array structures. A HOLAP server combines ROLAP and MOLAP. For example, it may use ROLAP for historic data while maintaining frequently accessed data in a separate MOLAP store.
- ❖ **Full materialization** refers to the computation of all of the cuboids in the lattice defining a data cube. It typically requires an excessive amount of storage space, particularly as the number of dimensions and size of associated concept hierarchies grow. This problem is known as the **curse of dimensionality**. Alternatively, **partial materialization** is the selective computation of a subset of the cuboids or subcubes in the

lattice. For example, an **iceberg cube** is a data cube that stores only those cube cells that have an aggregate value (e.g., count) above some minimum support threshold.

- ❖ OLAP query processing can be made more efficient with the use of indexing techniques. In **bitmap indexing**, each attribute has its own bitmap index table. Bitmap indexing reduces join, aggregation, and comparison operations to bit arithmetic. **Join indexing** registers the joinable rows of two or more relations from a relational database, reducing the overall cost of OLAP join operations. **Bitmapped join indexing**, which combines the bitmap and join index methods, can be used to further speed up OLAP query processing.
- ❖ **Data generalization** is a process that abstracts a large set of task-relevant data in a database from a relatively low conceptual level to higher conceptual levels. Data generalization approaches include data cube-based data aggregation and attribute-oriented induction. **Concept description** is the most basic form of descriptive data mining. It describes a given set of task-relevant data in a concise and summarative manner, presenting interesting general properties of the data. Concept (or class) description consists of **characterization** and **comparison** (or **discrimination**). The former summarizes and describes a data collection, called the **target class**, whereas the latter summarizes and distinguishes one data collection, called the **target class**, from other data collection(s), collectively called the **contrasting class(es)**.
- ❖ **Concept characterization** can be implemented using **data cube (OLAP-based) approaches** and the **attribute-oriented induction approach**. These are attributes or dimension-based generalization approaches. The **attribute-oriented induction approach** consists of the following techniques: *data focusing, data generalization by attribute removal or attribute generalization, count and aggregate value accumulation, attribute generalization control, and generalization data visualization.*
- ❖ **Concept comparison** can be performed using the attribute-oriented induction or data cube approaches in a manner similar to concept characterization. Generalized tuples from the target and contrasting classes can be quantitatively compared and contrasted.

5.5 Summary

- ❖ **Data cube computation and exploration** play an essential role in data warehousing and are important for flexible data mining in multidimensional space.
- ❖ A data cube consists of a **lattice of cuboids**. Each cuboid corresponds to a different degree of summarization of the given multidimensional data. **Full materialization** refers to the computation of all the cuboids in a data cube lattice. **Partial materialization** refers to the selective computation of a subset of the cuboid cells in the lattice. Iceberg cubes and shell fragments are examples of partial materialization. An **iceberg cube** is a data cube that stores only those cube cells that have an aggregate value (e.g., count) above some minimum support threshold. For **shell fragments** of a data cube, only some cuboids involving a small number of dimensions are computed, and queries on additional combinations of the dimensions can be computed on-the-fly.
- ❖ There are several efficient **data cube computation methods**. In this chapter, we discussed four cube computation methods in detail: (1) **MultiWay** array aggregation for materializing full data cubes in sparse-array-based, bottom-up, shared computation; (2) **BUC** for computing iceberg cubes by exploring ordering and sorting for efficient top-down computation; (3) **Star-Cubing** for computing iceberg cubes by integrating top-down and bottom-up computation using a star-tree structure; and (4) **shell fragment cubing**, which supports high-dimensional OLAP by precomputing only the partitioned cube shell fragments.
- ❖ **Multidimensional data mining in cube space** is the integration of knowledge discovery with multidimensional data cubes. It facilitates systematic and focused knowledge discovery in large structured and semi-structured data sets. It will continue to endow analysts with tremendous flexibility and power at multidimensional and multigranularity exploratory analysis. This is a vast open area for researchers to build powerful and sophisticated data mining mechanisms.
- ❖ Techniques for processing advanced queries have been proposed that take advantage of cube technology. These include **sampling cubes** for

multidimensional analysis on sampling data, and **ranking cubes** for efficient top- k (ranking) query processing in large relational data sets.

- ❖ This chapter highlighted three approaches to multidimensional data analysis with data cubes. **Prediction cubes** compute prediction models in multidimensional cube space. They help users identify interesting data subsets at varying degrees of granularity for effective prediction. **Multifeature cubes** compute complex queries involving multiple dependent aggregates at multiple granularities. **Exception-based, discovery-driven exploration** of cube space displays visual cues to indicate discovered data exceptions at all aggregation levels, thereby guiding the user in the data analysis process.

6.4 Summary

- ❖ The discovery of frequent patterns, associations, and correlation relationships among huge amounts of data is useful in selective marketing, decision analysis, and business management. A popular area of application is **market basket analysis**, which studies customers' buying habits by searching for itemsets that are frequently purchased together (or in sequence).
- ❖ **Association rule mining** consists of first finding **frequent itemsets** (sets of items, such as A and B , satisfying a *minimum support threshold*, or percentage of the task relevant tuples), from which **strong** association rules in the form of $A \Rightarrow B$ are generated. These rules also satisfy a *minimum confidence threshold* (a prespecified probability of satisfying B under the condition that A is satisfied). Associations can be further analyzed to uncover **correlation rules**, which convey statistical correlations between itemsets A and B .
- ❖ Many efficient and scalable algorithms have been developed for **frequent itemset mining**, from which association and correlation rules can be derived. These algorithms can be classified into three categories: (1) *Apriori-like algorithms*, (2) *frequent pattern growth-based algorithms* such as FP-growth, and (3) *algorithms that use the vertical data format*.
- ❖ The **Apriori algorithm** is a seminal algorithm for mining frequent itemsets for Boolean association rules. It explores the level-wise mining Apriori property that *all nonempty subsets of a frequent itemset must also be frequent*. At the k th iteration (for $k \geq 2$), it forms

frequent k -itemset candidates based on the frequent $(k - 1)$ -itemsets, and scans the database once to find the *complete* set of frequent k itemsets, L_k . Variations involving hashing and transaction reduction can be used to make the procedure more efficient. Other variations include partitioning the data (mining on each partition and then combining the results) and sampling the data (mining on a data subset). These variations can reduce the number of data scans required to as little as two or even one.

- ❖ **Frequent pattern growth** is a method of mining frequent itemsets without candidate generation. It constructs a highly compact data structure (an *FP-tree*) to compress the original transaction database. Rather than employing the generate-and-test strategy of Apriori-like methods, it focuses on frequent pattern (fragment) growth, which avoids costly candidate generation, resulting in greater efficiency.
- ❖ **Mining frequent itemsets using the vertical data format (Eclat)** is a method that transforms a given data set of transactions in the horizontal data format of *TID-itemset* into the vertical data format of *item-TID set*. It mines the transformed data set by *TID set* intersections based on the Apriori property and additional optimization techniques such as *diffset*.
- ❖ Not all strong association rules are interesting. Therefore, the support–confidence framework should be augmented with a pattern evaluation measure, which promotes the mining of *interesting* rules. A measure is **null-invariant** if its value is free from the influence of **null-transactions** (i.e., the *transactions that do not contain any of the itemsets being examined*). Among many pattern evaluation measures, we examined *lift*, *_2*, *all confidence*, *max confidence*, *Kulczynski*, and *cosine*, and showed that only the latter four are null-invariant. We suggest using the Kulczynski measure, together with the imbalance ratio, to present pattern relationships among itemsets.

7.7 Summary

- ❖ The **scope** of frequent pattern mining research reaches far beyond the basic concepts and methods introduced in Chapter 6 for mining frequent itemsets and associations. This chapter presented a road map of the field, where topics are organized with respect to the kinds of patterns and rules that can be mined, mining methods, and applications.

- ❖ In addition to mining for basic frequent itemsets and associations, **advanced forms of patterns** can be mined such as multilevel associations and multidimensional associations, quantitative association rules, rare patterns, and negative patterns. We can also mine high-dimensional patterns and compressed or approximate patterns.
- ❖ **Multilevel associations** involve data at more than one abstraction level (e.g., “*buys computer*” and “*buys laptop*”). These may be mined using multiple minimum support thresholds. **Multidimensional associations** contain more than one dimension. Techniques for mining such associations differ in how they handle repetitive predicates. **Quantitative association rules** involve quantitative attributes. Discretization, clustering, and statistical analysis that discloses exceptional behavior can be integrated with the pattern mining process.
- ❖ **Rare patterns** occur rarely but are of special interest. **Negative patterns** are patterns with components that exhibit negatively correlated behavior. Care should be taken in the definition of negative patterns, with consideration of the null-invariance property. Rare and negative patterns may highlight exceptional behavior in the data, which is likely of interest.
- ❖ **Constraint-based mining** strategies can be used to help direct the mining process toward patterns that match users’ intuition or satisfy certain constraints. Many user specified constraints can be pushed deep into the mining process. Constraints can be categorized into **pattern-pruning** and **data-pruning** constraints. Properties of such constraints include *monotonicity*, *antimonotonicity*, *data-antimonotonicity*, and *succinctness*. Constraints with such properties can be properly incorporated into efficient pattern mining processes.
- ❖ Methods have been developed for mining patterns in **high-dimensional space**. This includes a pattern growth approach based on *row enumeration* for mining data sets where the number of dimensions is large and the number of data tuples is small (e.g., for microarray data), as well as mining **colossal patterns** (i.e., patterns of very long length) by a *Pattern-Fusion* method.
- ❖ To reduce the number of patterns returned in mining, we can instead mine compressed patterns or approximate patterns. *Compressed patterns* can be mined with representative patterns defined based on

the concept of clustering, and *approximate patterns* can be mined by extracting **redundancy-aware top- k patterns** (i.e., a small set of k -representative patterns that have not only high significance but also low redundancy with respect to one another).

- ❖ **Semantic annotations** can be generated to help users understand the meaning of the frequent patterns found, such as for textual terms like “{*frequent, pattern*}.” These are dictionary-like annotations, providing semantic information relating to the term. This information consists of *context indicators* (e.g., terms indicating the context of that pattern), the most *representative data transactions* (e.g., fragments or sentences containing the term), and the most *semantically similar patterns* (e.g., “{*maximal, pattern*}” is semantically similar to “{*frequent, pattern*}”). The annotations provide a view of the pattern’s context from different angles, which aids in their understanding.
- ❖ Frequent pattern mining has many diverse applications, ranging from pattern-based data cleaning to pattern-based classification, clustering, and outlier or exception analysis. These methods are discussed in the subsequent chapters in this book.

8.7 Summary

- ❖ **Classification** is a form of data analysis that extracts models describing data classes. A classifier, or classification model, predicts categorical labels (classes). **Numeric prediction** models continuous-valued functions. Classification and numeric prediction are the two major types of prediction problems.
- ❖ **Decision tree induction** is a top-down recursive tree induction algorithm, which uses an attribute selection measure to select the attribute tested for each non leaf node in the tree. **ID3**, **C4.5**, and **CART** are examples of such algorithms using different attribute selection measures. **Tree pruning** algorithms attempt to improve accuracy by removing tree branches reflecting noise in the data. Early decision tree algorithms typically assume that the data are memory resident. Several scalable algorithms, such as **RainForest**, have been proposed for scalable tree induction.
- ❖ **Naive Bayesian classification** is based on Bayes’ theorem of posterior probability. It assumes class-conditional independence—that

the effect of an attribute value on a given class is independent of the values of the other attributes.

- ❖ A **rule-based classifier** uses a set of IF-THEN rules for classification. Rules can be extracted from a decision tree. Rules may also be generated directly from training data using sequential covering algorithms.
- ❖ A **confusion matrix** can be used to evaluate a classifier's quality. For a two-class problem, it shows the *true positives*, *true negatives*, *false positives*, and *false negatives*. Measures that assess a classifier's predictive ability include **accuracy**, **sensitivity** (also known as **recall**), **specificity**, **precision**, F , and F_- . Reliance on the accuracy measure can be deceiving when the main class of interest is in the minority.
- ❖ Construction and evaluation of a classifier require partitioning labeled data into a training set and a test set. **Holdout**, **random sampling**, **cross-validation**, and **bootstrapping** are typical methods used for such partitioning.
- ❖ Significance tests and ROC curves are useful tools for model selection. **Significance tests** can be used to assess whether the difference in accuracy between two classifiers is due to chance. **ROC curves** plot the true positive rate (or sensitivity) versus the false positive rate (or $1 - \text{specificity}$) of one or more classifiers.
- ❖ **Ensemble methods** can be used to increase overall accuracy by learning and combining a series of individual (base) classifier models. **Bagging**, **boosting**, and **random forests** are popular ensemble methods.
- ❖ The **class imbalance problem** occurs when the main class of interest is represented by only a few tuples. Strategies to address this problem include **oversampling**, **undersampling**, **threshold moving**, and **ensemble techniques**.

9.8 Summary

- ❖ Unlike naïve Bayesian classification (which assumes class conditional independence), **Bayesian belief networks** allow class conditional independencies to be defined between subsets of variables. They

provide a graphical model of causal relationships, on which learning can be performed. Trained Bayesian belief networks can be used for classification.

- ❖ **Backpropagation** is a neural network algorithm for classification that employs a method of gradient descent. It searches for a set of weights that can model the data so as to minimize the mean-squared distance between the network's class prediction and the actual class label of data tuples. Rules may be extracted from trained neural networks to help improve the interpretability of the learned network.
- ❖ A **support vector machine** is an algorithm for the classification of both linear and nonlinear data. It transforms the original data into a higher dimension, from where it can find a hyperplane for data separation using essential training tuples called **support vectors**.
- ❖ *Frequent patterns* reflect strong associations between attribute–value pairs (or items) in data and are used in **classification based on frequent patterns**. Approaches to this methodology include associative classification and discriminant frequent pattern–based classification. In **associative classification**, a classifier is built from association rules generated from frequent patterns. In **discriminative frequent pattern–based classification**, frequent patterns serve as combined features, which are considered in addition to single features when building a classification model.
- ❖ Decision tree classifiers, Bayesian classifiers, classification by backpropagation, support vector machines, and classification based on frequent patterns are all examples of **eager learners** in that they use training tuples to construct a generalization model and in this way are ready for classifying new tuples. This contrasts with **lazy learners** or **instance-based** methods of classification, such as nearest-neighbor classifiers and case-based reasoning classifiers, which store all of the training tuples in pattern space and wait until presented with a test tuple before performing generalization. Hence, lazy learners require efficient indexing techniques.
- ❖ In **genetic algorithms**, populations of rules “evolve” via operations of crossover and mutation until all rules within a population satisfy a specified threshold. **Rough set theory** can be used to approximately define classes that are not distinguishable based on the available attributes. **Fuzzy set** approaches replace “brittle” threshold cutoffs for continuous-valued attributes with membership degree functions.

- ❖ Binary classification schemes, such as support vector machines, can be adapted to handle **multiclass classification**. This involves constructing an ensemble of binary classifiers. Error-correcting codes can be used to increase the accuracy of the ensemble.
- ❖ **Semi-supervised classification** is useful when large amounts of unlabeled data exist. It builds a classifier using both labeled and unlabeled data. Examples of semi-supervised classification include *self-training* and *cotraining*.
- ❖ **Active learning** is a form of supervised learning that is also suitable for situations where data are abundant, yet the class labels are scarce or expensive to obtain. The learning algorithm can actively query a user (e.g., a human oracle) for labels. To keep costs down, the active learner aims to achieve high accuracy using as few labeled instances as possible.
- ❖ **Transfer learning** aims to extract the knowledge from one or more *source tasks* and apply the knowledge to a *target task*. TrAdaBoost is an example of the *instance-based approach* to transfer learning, which reweights some of the data from the source task and uses it to learn the target task, thereby requiring fewer labeled target-task tuples.

10.7 Summary

- ❖ A **cluster** is a collection of data objects that are *similar* to one another within the same cluster and are *dissimilar* to the objects in other clusters. The process of grouping a set of physical or abstract objects into classes of *similar* objects is called **clustering**.
- ❖ Cluster analysis has extensive **applications**, including business intelligence, image pattern recognition, Web search, biology, and security. Cluster analysis can be used as a standalone data mining tool to gain insight into the data distribution, or as a preprocessing step for other data mining algorithms operating on the detected clusters.
- ❖ Clustering is a dynamic field of research in data mining. It is related to **unsupervised learning** in machine learning.
- ❖ Clustering is a challenging field. Typical **requirements** of it include scalability, the ability to deal with different types of data and

attributes, the discovery of clusters in arbitrary shape, minimal requirements for domain knowledge to determine input parameters, the ability to deal with noisy data, incremental clustering and insensitivity to input order, the capability of clustering high-dimensionality data, constraint-based clustering, as well as interpretability and usability.

- ❖ Many clustering algorithms have been developed. These can be categorized from several **orthogonal aspects** such as those regarding partitioning criteria, separation of clusters, similarity measures used, and clustering space. This chapter discusses major fundamental clustering methods of the following categories: *partitioning methods*, *hierarchical methods*, *density-based methods*, and *grid-based methods*. Some algorithms may belong to more than one category.
- ❖ A **partitioning method** first creates an initial set of k partitions, where parameter k is the number of partitions to construct. It then uses an *iterative relocation technique* that attempts to improve the partitioning by moving objects from one group to another. Typical partitioning methods include k -means, k -medoids, and CLARANS.
- ❖ A **hierarchical method** creates a hierarchical decomposition of the given set of data objects. The method can be classified as being either *agglomerative (bottom-up)* or *divisive (top-down)*, based on how the hierarchical decomposition is formed. To compensate for the rigidity of *merge* or *split*, the quality of hierarchical agglomeration can be improved by analyzing object linkages at each hierarchical partitioning (e.g., in Chameleon), or by first performing *microclustering* (that is, grouping objects into “microclusters”) and then operating on the microclusters with other clustering techniques such as iterative relocation (as in BIRCH).
- ❖ A **density-based method** clusters objects based on the notion of density. It grows clusters either according to the density of neighborhood objects (e.g., in DBSCAN) or according to a density function (e.g., in DENCLUE). OPTICS is a density-based method that generates an augmented ordering of the data’s clustering structure.
- ❖ A **grid-based method** first quantizes the object space into a finite number of cells that form a grid structure, and then performs clustering on the grid structure. STING is a typical example of a grid-based method based on statistical information stored in grid cells. CLIQUE is a grid-based and subspace clustering algorithm.

- ❖ **Clustering evaluation** assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method. The tasks include assessing clustering tendency, determining the number of clusters, and measuring clustering quality.

11.5 Summary

- ❖ In conventional cluster analysis, an object is assigned to one cluster exclusively. However, in some applications, there is a need to assign an object to one or more clusters in a fuzzy or probabilistic way. **Fuzzy clustering** and **probabilistic model-based clustering** allow an object to belong to one or more clusters. A **partition matrix** records the membership degree of objects belonging to clusters.
- ❖ **Probabilistic model-based clustering** assumes that a cluster is a parameterized distribution. Using the data to be clustered as the observed samples, we can estimate the parameters of the clusters.
- ❖ A **mixture model** assumes that a set of observed objects is a mixture of instances from multiple probabilistic clusters. Conceptually, each observed object is generated independently by first choosing a probabilistic cluster according to the probabilities of the clusters, and then choosing a sample according to the probability density function of the chosen cluster.
- ❖ An **expectation-maximization algorithm** is a framework for approaching maximum likelihood or maximum a posteriori estimates of parameters in statistical models. Expectation-maximization algorithms can be used to compute fuzzy clustering and probabilistic model-based clustering.
- ❖ **High-dimensional data** pose several challenges for cluster analysis, including how to model high-dimensional clusters and how to search for such clusters.
- ❖ There are two major categories of clustering methods for high-dimensional data: subspace clustering methods and dimensionality reduction methods. **Subspace clustering methods** search for clusters in subspaces of the original space. Examples include **subspace search methods**, **correlation-based clustering methods**, and **biclustering methods**. **Dimensionality reduction methods** create a new space of lower dimensionality and search for clusters there.

- ❖ **Biclustering methods** cluster objects and attributes simultaneously. Types of biclusters include biclusters with **constant values**, **constant values on rows/columns**, **coherent values**, and **coherent evolutions on rows/columns**. Two major types of biclustering methods are **optimization-based methods** and **enumeration methods**.
- ❖ **Spectral clustering** is a **dimensionality reduction method**. The general idea is to construct new dimensions using an affinity matrix.
- ❖ **Clustering graph and network data** has many applications such as social network analysis. Challenges include how to measure the similarity between objects in a graph, and how to design clustering models and methods for graph and network data.
- ❖ **Geodesic distance** is the number of edges between two vertices on a graph. It can be used to measure similarity. Alternatively, similarity in graphs, such as social networks, can be measured using structural context and random walk. **SimRank** is a similarity measure that is based on both structural context and random walk.
- ❖ Graph clustering can be modeled as computing **graph cuts**. A **sparsest cut** may lead to a good clustering, while **modularity** can be used to measure the clustering quality.
- ❖ **SCAN** is a graph clustering algorithm that searches graphs to identify well-connected components as clusters.
- ❖ **Constraints** can be used to express application-specific requirements or background knowledge for cluster analysis. Constraints for clustering can be categorized as constraints on **instances**, on **clusters**, or on **similarity measurement**. Constraints on instances include **must-link** and **cannot-link** constraints. A constraint can be **hard** or **soft**.
- ❖ **Hard constraints for clustering** can be enforced by strictly respecting the constraints in the cluster assignment process. **Clustering with soft constraints** can be considered an optimization problem. Heuristics can be used to speed up constrained clustering.

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.

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.