data on a variety of advanced database systems. Chapter 11 describes major data mining applications as well as typical commercial data mining systems. Criteria for choosing a data mining system are also provided.

1.7 Data Mining Task Primitives

Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed. A data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining task primitives. These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths. The data mining primitives specify the following, as illustrated in Figure 1.13.

- The set of task-relevant data to be mined: This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions).
- The kind of knowledge to be mined: This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
- The background knowledge to be used in the discovery process: This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction. An example of a concept hierarchy for the attribute (or dimension) age is shown in Figure 1.14. User beliefs regarding relationships in the data are another form of background knowledge.
- The interestingness measures and thresholds for pattern evaluation: They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.
- The expected representation for visualizing the discovered patterns: This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

A data mining query language can be designed to incorporate these primitives, allowing users to flexibly interact with data mining systems. Having a data mining query language provides a foundation on which user-friendly graphical interfaces can be built.
Task-relevant data
Database or data warehouse name
Database tables or data warehouse cubes
Conditions for data selection
Relevant attributes or dimensions
Data grouping criteria

Knowledge type to be mined
Characterization
Discrimination
Association/correlation
Classification/prediction
Clustering

Background knowledge
Concept hierarchies
User beliefs about relationships in the data

Pattern interestingness measures
Simplicity
Certainty (e.g., confidence)
Utility (e.g., support)
Novelty

Visualization of discovered patterns
Rules, tables, reports, charts, graphs, decision trees, and cubes
Drill-down and roll-up

Figure 1.13 Primitives for specifying a data mining task.

This facilitates a data mining system’s communication with other information systems and its integration with the overall information processing environment.

Designing a comprehensive data mining language is challenging because data mining covers a wide spectrum of tasks, from data characterization to evolution analysis. Each task has different requirements. The design of an effective data mining query language requires a deep understanding of the power, limitation, and underlying mechanisms of the various kinds of data mining tasks.
There are several proposals on data mining languages and standards. In this book, we use a data mining query language known as DMQL (Data Mining Query Language), which was designed as a teaching tool, based on the above primitives. Examples of its use to specify data mining queries appear throughout this book. The language adopts an SQL-like syntax, so that it can easily be integrated with the relational query language, SQL. Let’s look at how it can be used to specify a data mining task.

Example 1.11 Mining classification rules. Suppose, as a marketing manager of AllElectronics, you would like to classify customers based on their buying patterns. You are especially interested in those customers whose salary is no less than $40,000, and who have bought more than $1,000 worth of items, each of which is priced at no less than $100. In particular, you are interested in the customer’s age, income, the types of items purchased, the purchase location, and where the items were made. You would like to view the resulting classification in the form of rules. This data mining query is expressed in DMQL as follows, where each line of the query has been enumerated to aid in our discussion.

(1) use database AllElectronics_db
(2) use hierarchy location_hierarchy for T.branch, age_hierarchy for C.age
(3) mine classification as promising_customers
(4) in relevance to C.age, C.income, I.type, I.place_made, T.branch
(5) from customer C, item I, transaction T
(6) where I.item_ID = T.item_ID and C.cust_ID = T.cust_ID
   and C.income ≥ 40,000 and I.price ≥ 100
(7) group by T.cust_ID

Note that in this book, query language keywords are displayed in sans serif font.