CHAPTER

1

Introduction to Data Mining

Data mining is getting more and more attention in today's business organizations. You may often hear people saying, "we should segment our customers using data mining tools," "data mining will increase customer satisfaction," or even "our competitors are using data mining to gain market share — we need to catch up!"

So, what is data mining and what benefits will using it bring you? How can you leverage this technology to solve your daily business problems? What are the technologies behind data mining? What is the life cycle of a typical data mining project? In this chapter, we will answer all these questions and give you an extended introduction to the data mining world.

In this chapter, you will learn about:

- A definition of data mining
- Determining which business problems can be solved with data mining
- Data mining tasks
- Using various data mining techniques
- Data mining flow
- The data mining project life cycle
- Current data mining standards
- A few new trends in data mining

What Is Data Mining

Data mining is a key member in the Business Intelligence (BI) product family, together with Online Analytical Processing (OLAP), enterprise reporting and ETL.

Data mining is about analyzing data and finding hidden patterns using automatic or semiautomatic means. During the past decade, large volumes of data have been accumulated and stored in databases. Much of this data comes from business software, such as financial applications, Enterprise Resource Management (ERP), Customer Relationship Management (CRM), and Web logs. The result of this data collection is that organizations have become data-rich and knowledge-poor. The collections of data have become so vast and are increasing so rapidly in size that the practical use of these stores of data has become limited. The main purpose of data mining is to extract patterns from the data at hand, increase its intrinsic value and transfer the data to knowledge.

You may wonder, why can't we dig out the knowledge by using SQL queries? In other words, you may wonder what the fundamental differences between data mining and relational database technologies are. Let's have a look of the following example.

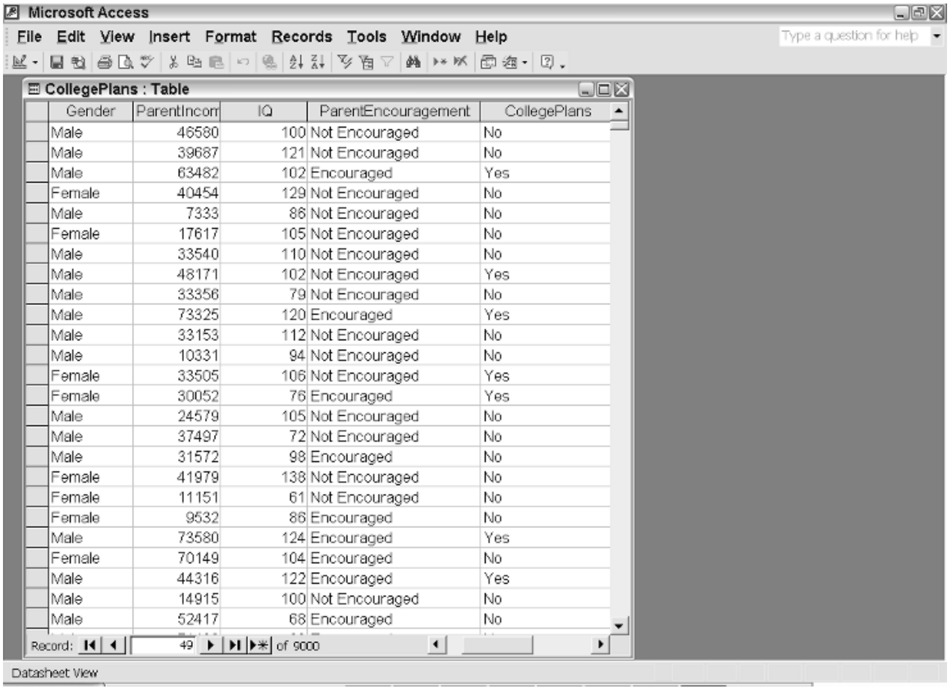
Figure 1.1 displays a relational table containing a list of high school graduates. The table records information such as gender, IQ, the level of parental encouragement, and the parental income of each student along with that student's intention to attend college. Someone asks you a question: What drives high school graduates to attend college?

You may write a query to find out how many male students attend college versus how many female students do. You may also write a query to determine the impact of the Parent Encouragement column. But what about male students who are encouraged by their parents? Or female students who are not encouraged by their parents? You would need to write hundreds of these queries to cover all the possible combinations. Data in numerical forms, such as that in Parent Income or IQ, is even more difficult to analyze. You would need to choose arbitrary ranges in these numeric values. What if there are hundreds of columns in your table? You would quickly end up with an impossible to manage number of SQL queries to answer a basic question about the meaning of your data.

In contrast, the data mining approach to this question is rather simple. All you need to do is select the right data mining algorithm and specify the column usage, meaning the input columns and the predictable columns (which are the targets for the analysis). A decision tree model would work well to determine the importance of parental encouragement in a student's decision to continue to college. You would select IQ, Gender, Parent Income, and Parent Encouragement as the input columns and College Plans as the predictable column. As the decision tree algorithm scans the data, it analyzes the impact of each input attribute related to the target and selects the most significant attribute to split. Each split divides the dataset into two subsets so that the value distribution of CollegePlans is as different as possible among these two subsets. This process is repeated recursively on each subset until the tree is completely built. Once the training process is complete, you can view the discovered patterns by browsing the tree.

Figure 1.2 shows a decision tree for the College Plan dataset. Each path from the root node to a leaf node forms a rule. Now, we can say that students with an IQ greater than 100 and who are encouraged by their parents have a 94% probability of attending college. We have extracted knowledge from the data.

As exemplified in Figure 1.2, data mining applies algorithms, such as decision trees, clustering, association, time series, and so on, to a dataset and analyzes its contents. This analysis produces patterns, which can be explored for valuable information. Depending on the underlying algorithm, these patterns can be in the form of trees, rules, clusters, or simply a set of mathematical formulas. The information found in the patterns can be used for reporting, as a guide to marketing strategies, and, most importantly, for prediction. For example, based on the rules produced by the previous decision tree, you can predict with significant accuracy whether high school students who are not represented in the original dataset will attend college.



Gender	ParentIncom	IQ	ParentEncouragement	CollegePlans
Male	46580	100	Not Encouraged	No
Male	39687	121	Not Encouraged	No
Male	63482	102	Encouraged	Yes
Female	40454	129	Not Encouraged	No
Male	7333	86	Not Encouraged	No
Female	17617	105	Not Encouraged	No
Male	33540	110	Not Encouraged	No
Male	48171	102	Not Encouraged	Yes
Male	33356	79	Not Encouraged	No
Male	73325	120	Encouraged	Yes
Male	33153	112	Not Encouraged	No
Male	10331	94	Not Encouraged	No
Female	33505	106	Not Encouraged	Yes
Female	30052	76	Encouraged	Yes
Male	24579	105	Not Encouraged	No
Male	37497	72	Not Encouraged	No
Male	31572	98	Encouraged	No
Female	41979	138	Not Encouraged	No
Female	11151	61	Not Encouraged	No
Female	9532	86	Encouraged	No
Male	73580	124	Encouraged	Yes
Female	70149	104	Encouraged	No
Male	44316	122	Encouraged	Yes
Male	14915	100	Not Encouraged	No
Male	52417	68	Encouraged	No

Figure 1.1 Student table

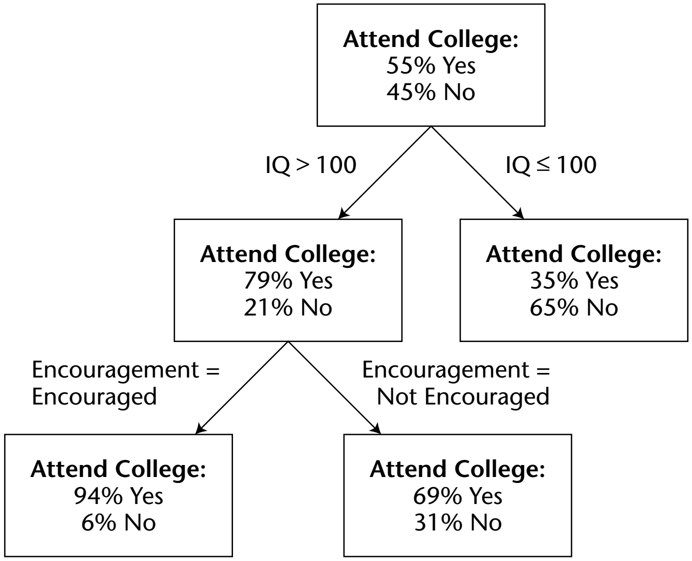


Figure 1.2 Decision tree

Data mining provides a lot of business value for enterprises. Why are we interested in data mining now? The following are a number of reasons:

A large amount of available data: Over the last decade, the price of hardware, especially hard disk space, has dropped dramatically. In conjunction with this, enterprises have gathered huge amounts of data through many applications. With all of this data to explore, enterprises want to be able to find hidden patterns to help guide their business strategies.

Increasing competition: Competition is high as a result of modern marketing and distribution channels such as the Internet and telecommunications. Enterprises are facing worldwide competition, and the key to business success is the ability to retain existing customers and acquire new ones. Data mining contains technologies that allow enterprises to analyze factors that affect these issues.

Technology ready: Data mining technologies previously existed only in the academic sphere, but now many of these technologies have matured and are ready to be applied in industry. Algorithms are more accurate, are more efficient and can handle increasingly complicated data. In addition, data mining application programming interfaces (APIs) are being standardized, which will allow developers to build better data mining applications.

Business Problems for Data Mining

Data mining techniques can be applied to many applications, answering various types of businesses questions. The following list illustrates a few typical problems that can be solved using data mining:

Churn analysis: Which customers are most likely to switch to a competitor? The telecom, banking, and insurance industries are facing severe competition these days. On average, each new mobile phone subscriber costs phone companies over 200 dollars in marketing investment. Every business would like to retain as many customers as possible. Churn analysis can help marketing managers understand the reason for customer churn, improve customer relations, and eventually increase customer loyalty.

Cross-selling: What products are customers likely to purchase? Cross-selling is an important business challenge for retailers. Many retailers, especially online retailers, use this feature to increase their sales. For example, if you go to online bookstores such as Amazon.com or Barnes andNoble.com to purchase a book, you may notice that the Web site gives you a set of recommendations about related books. These recommendations can be derived from data mining analysis.

Fraud detection: Is this insurance claim fraudulent? Insurance companies process thousands of claims a day. It is impossible for them to investigate each case. Data mining can help to identify those claims that are more likely to be false.

Risk management: Should the loan be approved for this customer? This is the most common question in the banking scenario. Data mining techniques can be used to score the customer's risk level, helping the manager make an appropriate decision for each application.

Customer segmentation: Who are my customers? Customer segmentation helps marketing managers understand the different profiles of customers and take appropriate marketing actions based on the segments.

Targeted ads: What banner ads should be displayed to a specific visitor? Web retailers and portal sites like to personalize their content for their Web customers. Using customers' navigation or online purchase patterns, these sites can use data mining solutions to display targeted advertisements to their customers' navigators.

Sales forecast: How many cases of wines will I sell next week in this store? What will the inventory level be in one month? Data mining forecasting techniques can be used to answer these types of time-related questions.

Data Mining Tasks

Data mining can be used to solve hundreds of business problems. Based on the nature of these problems, we can group them into the following data mining tasks.

Classification

Classification is one of the most popular data mining tasks. Business problems like churn analysis, risk management and ad targeting usually involve classification.

Classification refers to assigning cases into categories based on a predictable attribute. Each case contains a set of attributes, one of which is the *class* attribute (predictable attribute). The task requires finding a model that describes the class attribute as a function of input attributes. In the College Plans dataset previously described, the *class* is the College Plans attribute with two states: Yes and No. To train a classification model, you need to know the class value of input cases in the training dataset, which are usually the historical data. Data mining algorithms that require a target to learn against are considered *supervised* algorithms.

Typical classification algorithms include decision trees, neural network, and Naïve Bayes.

Clustering

Clustering is also called segmentation. It is used to identify natural groupings of cases based on a set of attributes. Cases within the same group have more or less similar attribute values.

Figure 1.3 displays a simple customer dataset containing two attributes: age and income. The clustering algorithm groups the dataset into three segments based on these two attributes. Cluster 1 contains the younger population with a low income. Cluster 2 contains middle-aged customers with higher incomes. Cluster 3 is a group of senior individuals with a relatively low income.

Clustering is an *unsupervised* data mining task. No single attribute is used to guide the training process. All input attributes are treated equally. Most clustering algorithms build the model through a number of iterations and stop when the model converges, that is, when the boundaries of these segments are stabilized.

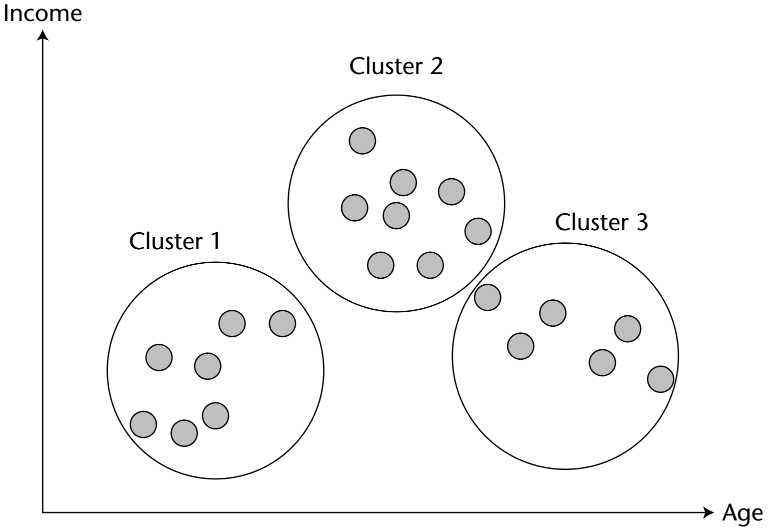


Figure 1.3 Clustering

Association

Association is another popular data mining task. Association is also called market basket analysis. A typical association business problem is to analyze a sales transaction table and identify those products often sold in the same shopping basket. The common usage of association is to identify common sets of items (frequent itemsets) and rules for the purpose of cross-selling.

In terms of association, each product, or more generally, each attribute/value pair is considered an item. The association task has two goals: to find frequent itemsets and to find association rules.

Most association type algorithms find frequent itemsets by scanning the dataset multiple times. The frequency threshold (support) is defined by the user before processing the model. For example, support = 2% means that the model analyzes only items that appear in at least 2% of shopping carts. A frequent itemset may look like {Product = "Pepsi", Product = "Chips", Product = "Juice"}. Each itemset has a size, which is the number of items that it contains. The size of this particular itemset is 3.

Apart from identifying frequent itemsets based on support, most association type algorithms also find rules. An association rule has the form A, B => C with a probability, where A, B, C are all frequent item sets. The probability is also referred to as the *confidence* in data mining literature. The probability is a threshold value that the user needs to specify before training an association model. For example, the following is a typical rule: Product = "Pepsi", Product = "Chips" => Product = "Juice" with an 80% probability. The interpretation of this rule is straightforward. If a customer buys Pepsi and chips, there is an 80% chance that he or she may also buy juice. Figure 1.4 displays the product association patterns. Each node in the figure represents a product, each edge represents the relationship. The direction of the edge represents the direction of the prediction. For example, the edge from Milk to Cheese indicates that those who purchase milk might also purchase cheese.

Regression

The regression task is similar to classification. The main difference is that the predictable attribute is a continuous number. Regression techniques have been widely studied for centuries in the field of statistics. Linear regression and logistic regression are the most popular regression methods. Other regression techniques include regression trees and neural networks.

Regression tasks can solve many business problems. For example, they can be used to predict coupon redemption rates based on the face value, distribution method, and distribution volume, or to predict wind velocities based on temperature, air pressure, and humidity.

Forecasting

Forecasting is yet another important data mining task. What will the stock value of MSFT be tomorrow? What will the sales amount of Pepsi be next month? Forecasting can help to answer these questions. It usually takes as an input time series dataset, for example a sequence of numbers with an attribute representing time. The time series data typically contains adjacent observations, which are order-dependant. Forecasting techniques deal with general trends, periodicity, and noisy noise filtering. The most popular time series technique is ARIMA, which stands for AutoRegressive Integrated Moving Average model.

Figure 1.5 contains two curves. The solid line curve is the actual time series data on Microsoft stock value, while the dotted curve is a time series model based on the moving average forecasting technique.

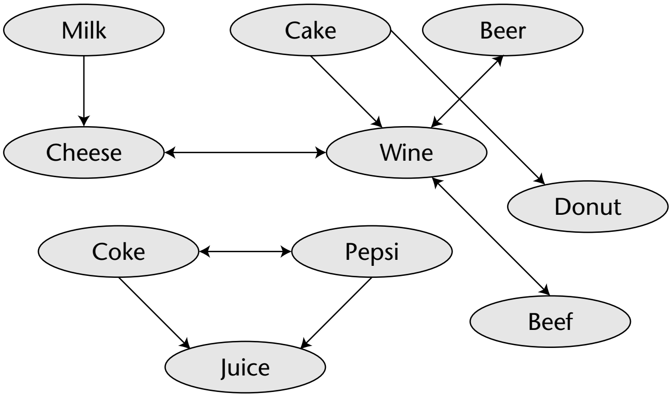


Figure 1.4 Products association

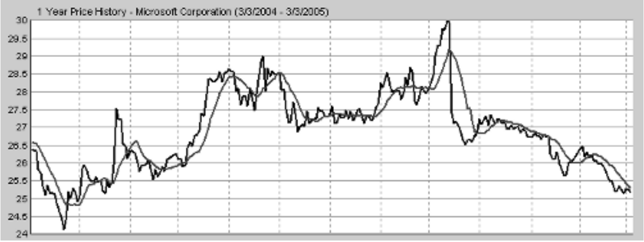


Figure 1.5 Time series

Sequence Analysis

Sequence analysis is used to find patterns in a discrete series. A sequence is composed of a series of discrete values (or states). For example, a DNA sequence is a long series composed of four different states: A, G, C, and T. A Web click sequence contains a series of URLs. Customer purchases can also be modeled as sequence data. For example, a customer first buys a computer, then speakers, and finally a Webcam. Both sequence and time series data contain adjacent observations that are dependant. The difference is that the sequence series contains discrete states, while the time series contains continuous numbers.

Sequence and association data are similar in the sense that each individual case contains a set of items or states. The difference between sequence and association models is that sequence models analyze the state transitions, while the association model considers each item in a shopping cart to be equal and independent. With the sequence model, buying a computer before buying speakers is a different sequence than buying speakers before a computer. With an association algorithm, these are considered to be the same itemset.

Figure 1.6 displays Web click sequences. Each node is a URL category. Each line has a direction, representing a transition between two URLs. Each transition is associated with a weight, representing the probability of the transition between one URL and the other.

Sequence analysis is a relatively new data mining task. It is becoming more important mainly due to two types of applications: Web log analysis and DNA analysis. There are several different sequence techniques available today such as Markov chains. Researchers are actively exploring new algorithms in this field. Figure 1.6 displays the state transitions among a set of URL categories based on Web click data.

Deviation Analysis

Deviation analysis is for finding those rare cases that behave very differently from others. It is also called outlier detection, which refers to the detection of significant changes from previously observed behavior. Deviation analysis can be used in many applications. The most common one is credit card fraud detection. To identify abnormal cases from millions of transactions is a very challenging task. Other applications include network intrusion detection, manufacture error analysis, and so on.

There is no standard technique for deviation analysis. It is still an actively researched topic. Usually analysts employ some modified versions of decision trees, clustering, or neural network algorithms for this task. In order to generate significant rules, analysts need to oversample the anomaly cases in the training dataset.

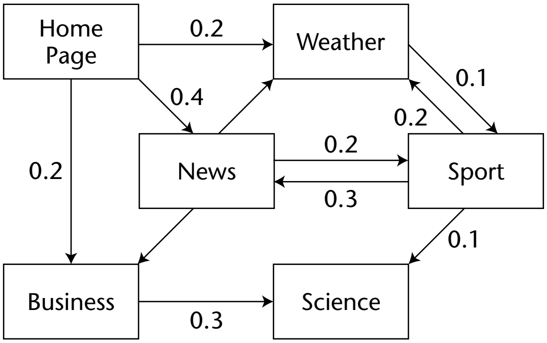


Figure 1.6 Web navigation sequence

Data Mining Techniques

Although data mining as a term is relatively new, most data mining techniques have existed for years. If we look at the roots of those popular data mining algorithms, we find that they are mainly derived from three fields: statistics, machine learning, and database.

Most of data mining tasks listed in the previous section have been addressed in the statistics community. A number of data mining algorithms, including regression, time series, and decision trees, were invented by statisticians. Regression techniques have existed for centuries. Time series algorithms have been studied for decades. The decision tree algorithm is one of the more recent techniques, dating from the mid-1980s.

Data mining focuses on automatic or semiautomatic pattern discovery. Several machine learning algorithms have been applied to data mining. Neural networks are one of these techniques and are excellent for classification and regression, especially when the attribute relationships are nonlinear. The genetic algorithm is yet another machine learning technique. It simulates the natural evolution process by working with a set of candidates and a survival (fitness) function. The survival function repeatedly selects the most suitable candidates for the next generation. Genetic algorithms can be used for classification and clustering tasks. They can also be used in conjunction with other algorithms, for instance, helping a neural network to find the best set of weights among neurons.

A database is the third technical source for data mining. Traditional statistics assumes that all the data can be loaded into memory for statistical analysis. Unfortunately, this is not always the case in the modern world. Database experts know how to handle large amounts of data that do not fit in memory, for example, finding association rules in a fact table containing millions of sales transactions. As a matter of fact, the most efficient association algorithms come from the database research community. There are also a few scalable versions of classification and clustering algorithms that use database techniques, including the Microsoft Clustering algorithm.

Data Flow

Data mining is one key member in the data warehouse family. Where does data mining fit in terms of the overall flow of data in a typical business scenario? Figure 1.7 illustrates a typical enterprise data flow to which data mining can be applied in various stages.

A business application stores transaction data in an online transaction processing (OLTP) database. The OLTP data is extracted, transformed and loaded into data warehouse on a regular basis. The schema of the data warehouse is generally different from an OLTP schema. A typical data warehouse schema has the form of a star or snowflake, with fact tables (transaction tables) in the middle of the schema, surrounded by a set of dimension tables. Once the data warehouse is populated, OLAP cubes can be built on the warehouse data.

Where can data mining add value in this typical enterprise data flow? First, and most commonly, data mining can be applied to the data warehouse where data has already been cleaned. The patterns discovered by mining models can be presented to marketing managers through reports. Usually in small enterprises there is no data warehouse. Consequently, people directly mine OLTP tables (usually by making a copy of the related tables on a separate database).

Data mining may have a direct link to business applications, most commonly through predictions. Embedding data mining features within business applications is becoming more and more common. In a Web cross-selling scenario, once a Web customer places a product in the shopping cart, a data mining prediction query is executed to get a list of recommended products based on association analysis.

Data mining can also be applied to analyze OLAP cubes. A cube is a multidimensional database with many dimensions and measures. Large dimensions may have millions of members. The total number of cells in a cube is exponential to the number of dimensions and members in a dimension. It becomes difficult to find interesting patterns manually. Data mining techniques can be applied to discover hidden patterns in a cube. For example, an association algorithm can be applied to a sales cube, analyzing customer purchase patterns for a specific region and time period. We can apply data mining techniques to forecast the measures such as store sales and profit. Another example is clustering. Data mining can group customers based on dimension properties and measures. Data mining can not only find patterns in a cube but also reorganize cube design. For instance, we can create a new customer dimension based on the results of the clustering model, grouping customers of the same cluster together in the new dimension.

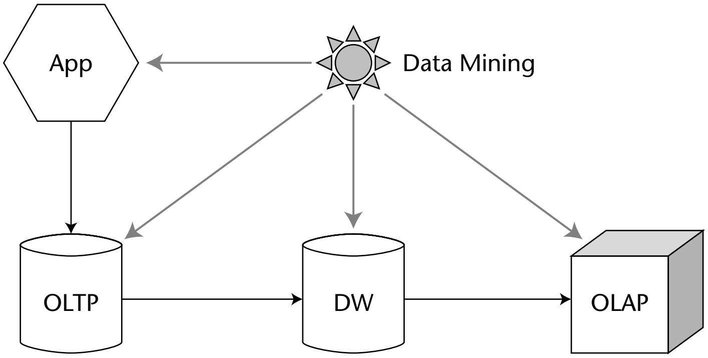


Figure 1.7 Data flow

Data Mining Project Cycle

What is the life cycle of a data mining project? What are the challenging steps? Who should be involved in a data mining project? To answer these questions, let's go over a typical data mining project step by step.

Step 1: Data Collection

The first step of data mining is usually data collection. Business data is stored in many systems across an enterprise. For example, there are hundreds of OLTP databases and over 70 data warehouses inside Microsoft. The first step is to pull the relevant data to a database or a data mart where the data analysis is applied. For instance, if you want to analyze the Web click stream and your company has a dozen Web servers, the first step is to download the Web log data from each Web server.

Sometimes you might be lucky. The data warehouse on the subject of your analysis already exists. However, the data in the data warehouse may not be rich enough. You may still need to gather data from other sources. Suppose that there is a click stream data warehouse containing all the Web clicks on the Web site of your company. You have basic information about customers' navigation patterns. However, because there is not much demographic information about your Web visitors, you may need to purchase or gather some demographic data from other sources in order to build a more accurate model.

After the data is collected, you can sample the data to reduce the volume of the training dataset. In many cases, the patterns contained in 50,000 customers are the same as in 1 million customers.

Step 2: Data Cleaning and Transformation

Data cleaning and transformation is the most resource-intensive step in a data mining project. The purpose of data cleaning is to remove noise and irrelevant information out of the dataset. The purpose of data transformation is to modify the source data into different formats in terms of data types and values. There are various techniques you can apply to data cleaning and transformation, including:

Data type transform: This is the simplest data transform. An example is transforming a Boolean column type to integer. The reason for this transform is that some data mining algorithms perform better on integer data, while others prefer Boolean data.

Continuous column transform: For continuous data such as that in Income and Age columns, a typical transform is to bin the data into buckets. For example, you may want to bin Age into five predefined age groups. Apart from binning, techniques such as normalization are popular for transforming continuous data. Normalization maps all numerical values to a number between 0 and 1 (or –1 to 1) to ensure that large numbers do not dominate smaller numbers during the analysis.

Grouping: Sometimes there are too many distinct values (states) for a discrete column. You need to group these values into a few groups to reduce the model's complexity. For example, the column Profession may have tens of different values such as Software Engineer, Telecom Engineer, Mechanical Engineer, Consultant, and so on. You can group various engineering professions by using a single value: Engineer. Grouping also makes the model easier to interpret.

Aggregation: Aggregation is yet another important transform. Suppose that there is a table containing the telephone call detail records (CDR) for each customer, and your goal is to segment customers based on their monthly phone usage. Since the CDR information is too detailed for the model, you need to aggregate all the calls into a few derived attributes such as total number of calls and the average call duration. These derived attributes can later be used in the model.

Missing value handling: Most datasets contain missing values. There are a number of causes for missing data. For instance, you may have two customer tables coming from two OLTP databases. Merging these tables can result in missing values, since table definitions are not exactly the same. In another example, your customer demographic table may have a column for age. But customers don't always like to give you this information during the registration. You may have a table of daily closing values for the stock MSFT. Because the stock market closes on weekends, there will be null values for those dates in the table. Addressing missing values is an important issue. There are a few ways to deal with this problem. You may replace the missing values with the most popular value (constant). If you don't know a customer's age, you can replace it with the average age of all the customers. When a record has too many missing values, you may simply remove it. For more advanced cases, you can build a mining model using those complete cases, and then apply the model to predict the most likely value for each missing case.

Removing outliers: Outliers are abnormal cases in a dataset. Abnormal cases affect the quality of a model. For example, suppose that you want to build a customer segmentation model based on customer telephone usage (average duration, total number of calls, monthly invoice, international calls, and so on) There are a few customers (0.5%) who behave very differently. Some of these customers live aboard and use roaming all the time. If you include those abnormal cases in the model, you may end up by creating a model with majority of customers in one segment and a few other very small segments containing only these outliers. The best way to deal with outliers is to simply remove them before the analysis. You can remove outliers based on an individual attribute; for instance, removing 0.5% customers with highest or lowest income. You may remove outliers based on a set of attributes. In this case, you can use a clustering algorithm. Many clustering algorithms, including Microsoft Clustering, group outliers into a few particular clusters.

There are many other data-cleaning and transformation techniques, and there are many tools available in the market. SQL Server Integration Services (SSIS) provides a set of transforms covering most of the tasks listed here.

Step 3: Model Building

Once the data is cleaned and the variables are transformed, we can start to build models. Before building any model, we need to understand the goal of the data mining project and the type of the data mining task. Is this project a classification task, an association task or a segmentation task? In this stage, we need to team up with business analysts with domain knowledge. For example, if we mine telecom data, we should team up with marketing people who understand the telecom business.

Model building is the core of data mining, though it is not as time- and resource-intensive as data transformation. Once you understand the type of data mining task, it is relatively easy to pick the right algorithms. For each data mining task, there are a few suitable algorithms. In many cases, you won't know which algorithm is the best fit for the data before model training. The accuracy of the algorithm depends on the nature of the data such as the number of states of the predictable attribute, the value distribution of each attribute, the relationships among attributes, and so on. For example, if the relationship among all input attributes and predictable attributes were linear, the decision tree algorithm would be a very good choice. If the relationships among attributes are more complicated, then the neural network algorithm should be considered.

The correct approach is to build multiple models using different algorithms and then compare the accuracy of these models using some tool, such as a lift chart, which is described in the next step. Even for the same algorithm, you may need to build multiple models using different parameter settings in order to fine-tune the model's accuracy.

Step 4: Model Assessment

In the model-building stage, we build a set of models using different algorithms and parameter settings. So what is the best model in terms of accuracy? How do you evaluate these models? There are a few popular tools to evaluate the quality of a model. The most well-known one is the lift chart. It uses a trained model to predict the values of the testing dataset. Based on the predicted value and probability, it graphically displays the model in a chart. We will give a better description of lift charts in Chapter 3.

In the model assessment stage, not only do you use tools to evaluate the model accuracy but you also need to discuss the meaning of discovered patterns with business analysts. For example, if you build an association model on a dataset, you may find rules such as *Relationship = Husband => Gender = Male with 100% confidence*. Although the rule is valid, it doesn't contain any business value. It is very important to work with business analysts who have the proper domain knowledge in order to validate the discoveries.

Sometimes the model doesn't contain useful patterns. This may occur for a couple of reasons. One is that the data is completely random. While it is possible to have random data, in most cases, real datasets do contain rich information. The second reason, which is more likely, is that the set of variables in the model is not the best one to use. You may need to repeat the data-cleaning and transformation step in order to derive more meaningful variables. Data mining is a cyclic process; it usually takes a few iterations to find the right model.

Step 5: Reporting

Reporting is an important delivery channel for data mining findings. In many organizations, the goal of data miners is to deliver reports to the marketing executives. Most data mining tools have reporting features that allow users to generate predefined reports from mining models with textual or graphic outputs. There are two types of reports: reports about the findings (patterns) and reports about the prediction or forecast.

Step 6: Prediction (Scoring)

In many data mining projects, finding patterns is just half of the work; the final goal is to use these models for prediction. Prediction is also called scoring in data mining terminology. To give predictions, we need to have a trained model and a set of new cases. Consider a banking scenario in which you have built a model about loan risk evaluation. Every day there are thousands of new loan applications. You can use the risk evaluation model to predict the potential risk for each of these loan applications.

Step 7: Application Integration

Embedding data mining into business applications is about applying intelligence back to business, that is, closing the analysis loop. According to Gartner Research, in the next few years, more and more business applications will embed a data mining component as a value-added. For example, CRM applications may have data mining features that group customers into segments. ERP applications may have data mining features to forecast production. An online bookstore can give customers real-time recommendations on books. Integrating data mining features, especially a real-time prediction component into applications is one of the important steps of data mining projects. This is the key step for bringing data mining into mass usage.

Step 8: Model Management

It is challenging to maintain the status of mining models. Each mining model has a life cycle. In some businesses, patterns are relatively stable and models don't require frequent retraining. But in many businesses patterns vary frequently. For example, in online bookstores, new books appear every day. This means that new association rules appear every day. The duration of a mining model is limited. A new version of the model must be created frequently. Ultimately, determining the model's accuracy and creating new versions of the model should be accomplished by using automated processes.

Like any data, mining models also have security issues. Mining models contain patterns. Many of these patterns are the summary of sensitive data. We need to maintain the read, write, and prediction rights for different user profiles. Mining models should be treated as first-class citizens in a database, where administrators can assign and revoke user access rights to these models.

Data Mining and the Current Market

In this section, we give an overview of the current data mining market and discuss a few major vendors in this field.

Data Mining Market Size

Giga Research estimates the size of the market for data mining to have passed the billion dollar mark, including software and services (consulting and service bureau). Other research organizations disagree and make more conservative estimations of its market size, from $200 to $700 million. However, one research conclusion is shared by various analysts: the data mining market is the fastest growing business intelligence component (reporting, OLAP, packaged data marts, and so on). Data mining currently represents about 15% of the business intelligence market. It is evolving from transitional horizontal packages toward embedded data mining applications, integrated with CRM, ERP, or other business applications.

Major Vendors and Products

There are hundreds of data mining product and consulting companies. KDNuggets (`kdnuggets.com`) has an extended list of most of these companies and their products in the data mining field. Here we list a few the major data mining product companies.

SAS: SAS is probably the largest data mining product vendor in terms of the market share. SAS has been in the statistics field for decades. SAS Base contains a very rich set of statistical functions that can be used for all sorts of data analysis. It also has a powerful script language called SAS Script. SAS Enterprise Miner was introduced in 1997. It provides the user with a graphical flow environment for model building, and it has a set of popular data mining algorithms, including decision trees, neural network, regression, association, and so on. It also supports text mining.

SPSS: SPSS is another major statistics company. It has a number of data mining products including SPSS base and Answer Tree (decision trees). SPSS acquired a British company ISL in late 1998 and inherited the Clementine data mining package. Clementine was one of the first companies to introduce the data mining flow concept, allowing users to clean data, transform data, and train models in the same workflow environment. Clementine also has tools to manage data mining project cycle.

IBM: IBM has a data mining product called Intelligent Miner, developed by an IBM German subsidiary. Intelligent Miner contains a set of algorithms and visualization tools. Intelligent Miner exports mining models in Predictive Modeling Markup Language (PMML), which was defined by the Data Mining Group (DMG), an industry organization. PMML documents are Extensible Markup Language (XML) files containing the descriptions of model patterns and statistics of training dataset. These files can be loaded by DB2 database for prediction purpose.

Microsoft Corporation: Microsoft was the first major database vendor to include data mining features in a relational database. SQL Server 2000, released in September 2000, contains two patented data mining algorithms: Microsoft Decision Trees and Microsoft Clustering. Apart from these algorithms, the most important data mining feature is the implementation of OLE DB for Data Mining. OLE DB for Data Mining is an industry standard that defines a SQL-style data mining language and a set of schema rowsets targeted at database developers. This API makes it very easy to embed data mining components, especially prediction features, into user applications. We will detail the OLE DB for Data Mining API in a later chapter.

Oracle: Oracle 9i shipped in 2000, containing a couple of data mining algorithms based on association and Naïve Bayes. Oracle 10g includes many more data mining tools and algorithms. Oracle also incorporated the Java Data Mining API, which is a Java package for data mining tasks.

Angoss: Angoss' KnowledgeSTUDIO is a data mining tool that includes the power to build decision trees, cluster analysis, and several predictive models, allowing users to mine and understand their data from many different perspectives. It includes powerful data visualization tools to support and explain the discoveries. Angoss also has a set of content viewer controls, which work with data mining algorithms in SQL Server 2000. Its algorithms can also be plugged into the SQL Server platform.

KXEN: KXEN is a data mining software provider based in France. It has a number of data mining algorithms, including SVM, regression, time series, segmentation, and so forth. It also provides data mining solutions for OLAP cubes. It developed an Excel add-in that allows users to do data mining in a familiar Excel environment.

Current Issues and Challenges

Although data mining has been talked about more frequently in recent years, it is still a relatively small market. Most data mining users are the data analysts of large businesses in the sector of finance, telecom, and insurance. Data mining is still considered as an optional high-end feature. Because it seems to be too sophisticated for most developers to understand, very few business applications include data mining features.

Data mining is not yet a main stream technology, although it has the potential to bring added value to almost any kind of business application. There are a few challenges to overcome before data mining will become a mass technology:

Proprietary horizontal packages without a standard API: The majority of the data mining products available in the market are horizontal packages. These tools include a few data mining algorithms, a graphic interface for model building, some data extraction and transformation functions, and a reporting tool. Some products also include their own storage engines with special formats. Because there are so many different components, it is hard to find a good product with satisfactory features across all these areas. Most products are strong in data mining algorithms, but relatively weak in other components. Probably the biggest issue is that these products are proprietary systems. There is no dominant standard API. Thus, it is hard for developers to integrate the results of data mining with standard reporting tools or use model prediction functions in applications.

Analyst-oriented instead of developer-oriented: Most data mining products are oriented toward data analysts, most likely statisticians. Many data mining products originate from statistical packages with hundreds of statistical functions, requiring users to have strong mathematical backgrounds. To make data mining a main stream technology, we need to help millions of application developers who know more about database technologies and less about math to apply data mining techniques in an easy way.

Limited user education: Data analysis is becoming more and more important. However, most developers are not familiar with data analysis techniques. Accordingly we need to improve user education in this area.

Limited algorithm features: Most data mining algorithms are quite general. It is easy to generate hundreds of rules using these algorithms; however, most of these rules may be just common sense. Integrating subjects of interest and domain knowledge with the algorithm is still an open issue. Some new areas such as DNA sequence analysis require more advanced techniques than just horizontal data mining packages. There is still a lot of research to do on data mining algorithms.

Data Mining Standards

Data mining is a relatively new field. You can compare today's data mining market with the database market about 20 years ago, when there was no relational concept or SQL. Each vendor had its own propriety storage format, and there was no easy way to query different data sources. During the past few years, data mining vendors have begun to recognize similar issues in the data mining field and have made some effort to standardize the data mining metadata, APIs and content formats. These efforts are mostly led by industrial bodies or independent data mining software vendors. In this section, we will give you a high-level overview of these data mining standards.

OLE DB for DM and XML for Analysis

Data Mining (OLE DB for DM) was initialized by Microsoft in 1999 and supported by a number of data mining vendors including Angoss, KXEN, and Megaputer. OLE DB for Data Mining doesn't define any new COM or OLE DB interfaces. Instead, it defines powerful data mining languages for model creation, training, and prediction. It also defines a set of schema rowsets, which store the metadata for mining models and mining algorithms. The key philosophy of OLE DB for Data Mining is to map relational concepts to data mining by leveraging SQL and OLE DB. In the specification, a mining model is considered to be a first-class object, just like a relational table. All the operations on mining models are relational. Prediction is a special joint query between a mining model and a relational table. Developers can connect to the data mining algorithms provider through ADO, in the same way that they connect to a database server. Through the Command object, a prediction query can be defined and executed. The query results are presented in the form of a record set. It is very natural for database developers to learn the concept of OLE DB for DM.

XML for Analysis is another industrial standard initialized by Microsoft in 2001 and owned by XML/A Council. The council is co-chaired by Microsoft and two other major BI vendors: Hyperion and SAS. A dozen of BI vendors are the members of the council. The XML/A Council is in charge of the definition of XML for Analysis Specification. This standard leverages technologies from OLE DB for OLAP and OLE DB for Data Mining, supporting the OLAP Query Language (MDX) and the data mining query language Data Mining eXtensions (DMX). It allows consumer applications to query OLAP and data mining servers through the XML Simple Object Access Protocol (SOAP) across different platforms. We will explain the details of OLE DB for DM and XML for Analysis in later chapters.

SQL/Multimedia for Data Mining

ISO SQL/Multimedia (SQL MM) is structured as multipart SQL extensions covering framework, full text, spatial data, images and data mining. The data mining section was proposed by IBM. The key concept is to define a set of user-defined types and methods in a database for the purpose of data mining. These types and methods can then be used in database queries.

Figure 1.8 shows an overview of user-defined types for model training, test and application. The user-defined type `DM_MiningData` is an abstraction of source data contained in tables or views. It also stores the metadata needed to access the data source. The `DM_MiningSchema` type defines the input fields used by data mining training, test, or application runs. The data mining field type defines how a field should be handled by the data mining techniques. For instance, a field can be declared as categorical. DM_ClasTask is a data mining task type. There are four data mining types supported by SQL/MM for DM, including association, clustering, classification, and regression. There is also a set of methods defined for these tasks for the purpose of model training and parameter setting.

DM_ClasModel is a mining model type that is an abstraction for a classification type of mining model. It provides methods to access the model properties as well as methods to apply (predict) or to test the model. DM_ClasTestResult is a data mining test result type. It is used to hold the result information of a test run computed for a data mining model. Finally, the DM_ApplicationData type is defined as a container for data used to apply a mining model. Basically, it is an abstraction for a set of values with associated names representing a single row of input data. The result of model application is stored in the result type such as DM_ClasResult.

The following is an example of creating a classification mining model based on these user-defined types:

```
With MyData As (
     DM_MiningData::DM_defMiningData('CT')
)
Insert into MT (ID, TASK)
Values (
1, DM_ClasTask::DM_defClasTask) (
     MyData, NULL,
     (
       (new DM_ClasSettings())
       .DM_clasUseSchema(MyData.DM_genMiningSchema())
     ).DM_clasSetTarget('r')
)
)
```

The preceding statement does the following tasks:

1. Creates a DM_MiningData value using the DM_defMiningData method
2. Creates a DM_MiningSchema value using the DM_genMiningSchema of the DM_MiningData type
3. Creates a DM_ClasSettings value using the default constructor and assign the DM_MiningSchema value as the schema to use
4. Declares column named “r” as a predictable field
5. Creates a DM_ClasTask value using the DM_defClasTask method
6. Stores the newly created DM_ClasTask value in table MT

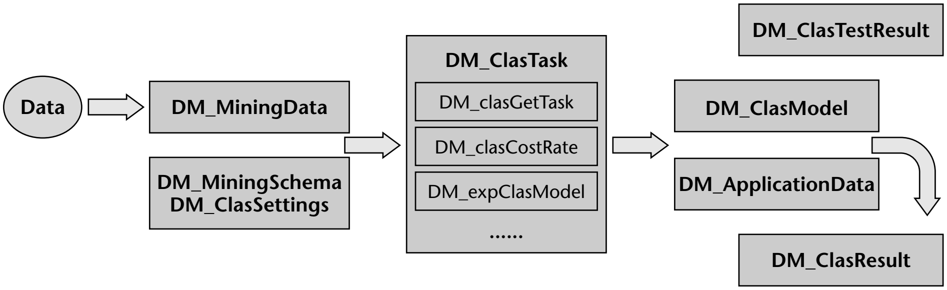


Figure 1.8 Overview of user-defined types for model training, testing, and application

The following statement invokes the training for the classification model using the task value in the MT table. The trained model is stored in the MM table with two columns: ID and Model. It can later be used for the application and test run.

```
Insert Into MM (ID, MODEL)
Values (1, MyTask.DM_buildClasModel())
```

Java Data Mining API

The Java Data Mining API (JSR-73 API) is a Java package for data mining. JSR 73 work is led by Oracle. The goal is to allow Java applications to communicate with data mining engines to build, test, and apply mining models.

JSR-73 separates source data (physical data) and logical data concepts. Source data can be any relational data or text files. There are three key sources: individual record case (single case), single-record case table (simple table) and multirecord case table. A multirecord case table is more like a transaction table, and each case has multiple records. Logical data contains a set of logical attributes. A logical attribute is an abstraction of a physical attribute and contains the definition of content type such as categorical, ordinal, or numerical.

Another important class in JSR 73 is schema, which is a folder for storing named mining objects. Schema is maintained in metadata repository.

JSR 73 defines a set of Java classes for different data mining tasks such as classification and clustering. Each task contains a set of methods for prediction, validation, and other purposes. Users can also specify the type of the mining model settings, such as cost matrix.

The code in Listing 1.1 creates a mining model using JSR:

```
//Get a connection.
javax.datamining.ConnectionSpec spec =
     connectionFactory.createConnectionSpec("myDMS", "me", "myPass");
```

Listing 1.1 Mining model using JSR *(continued)*

```
javax.datamining,Connection dmsCom =
     connectionFactory.getConnection (spec);

//Set schema
dmsCon.setCurrentSchema("mySchema";

//Create and populate PhysicalData object
String uri = new String ("...customer.data");
PhysicalDataSet data = new PhysicalDataSet (uri);
Data.getMetadata();

//Create LogicalData object
LogicalData ld = new LogicalData (data);
//Specify logical attribute
LogicalAttribute income = ld.getAttribute ("income");
Income.setAttributeType(AttributeType.numerical);

//Create FunctionSettings.
FunctionSettings settings = new
     ClassficationSettings(ld, "credit_risk");
Settings.setCostMatrix(costs);

//Create the build task.
buildTask = new BuildTask(data, settings, "myModel");
dmsCon.addObject("myBuildTask", buildTask);
dmsCon.save();

//Execute the task.
ExecutetionHandle handle = dmsCon.execute(task);
Handle.waitForCompletion();

//Access the model.
Model model = (Model)dmsCon;getObject("myModel")
```

Listing 1.1 *(continued)*

Predictive Model Markup Language

Predictive Model Markup Language (PMML) is defined by an industrial organization called Data Mining Group (DMG, dmg.org). DMG includes most of major data mining product vendors. SAS, SPSS, IBM, Microsoft, Oracle, and a few others are the members of DMG. The goal of PMML is to define a standard XML format for persisting mining model content. Without PMML, mining models are application-dependent, system-dependent, and architecture-dependent. PMML standardizes the model content for common data mining algorithms, eases the model deployment, and allows models to be exchanged among various software packages.

Because each data mining algorithm has different types of content, the formats of the XML documents needed to persist these contents are different. PMML defines the XML representation for a set of popular data mining algorithm contents, including decision trees, regression, neuron network, clustering, and so on. For example, PMML for the Decision Tree algorithms specifies tags to describe tree topology, node splitting condition, node statistics, and so on.

Apart from its algorithm part, a PMML document also has sections to hold a data dictionary, statistics, transformations, and so on. The following is a list of components in a PMML document:

Data dictionary: The data dictionary contains definitions for fields as used in mining models. It specifies the types and value ranges.

Mining schema: The mining schema is a subset of the fields as defined in the data dictionary. Each model contains one mining schema that lists fields as used in that model.

Transformation dictionary: The transformation dictionary contains descriptions of mining fields derived by using transformations such as aggregation and binning.

Statistics: This section contains the statistics of training dataset.

Taxonomy: Taxonomy is the section for defining attribute hierarchies. For example, the attributes Country, State, and City form a geographic hierarchy.

One or more PMML models: The section describes the content of the mining model. It is algorithm-specific.

PMML supports the content definition for following mining algorithms:

- Polynomial regression
- Logistic regression
- General regression
- Center-based clusters
- Density-based clusters
- Trees
- Associations
- Neural nets
- Naïve Bayes
- Sequences
- Text model
- Vector machine

While OLE DB for DM, SQL MM/DM, and JSR 73 are more or less competing against each other, PMML remains in a neutral position. It is not a programming interface for data mining; instead, it focuses on the model content. All the major data mining vendors support the PMML standard. There are two advantages to PMML. The first one is about model interchange, that is, a model created by using an algorithm of product A can be loaded by product B. The second advantage of PMML is the ease of deployment. It is rather simple to deploy an XML document to different servers and platforms. More information about PMML can be found at `dmg.org`. Listing 1.2 is the extraction of a PMML document for a decision tree model.

```
<?xml version="1.0" ?>
  <PMML version="3.0" >
    <Header copyright="www.dmg.org" description="A very small binary
tree model to show structure."/>
    <DataDictionary numberOfFields="5" >
      <DataField name="temperature" optype="continuous"/>
      <DataField name="humidity" optype="continuous"/>
      <DataField name="windy" optype="categorical" >
        <Value value="true"/>
        <Value value="false"/>
      </DataField>
      <DataField name="outlook" optype="categorical" >
        <Value value="sunny"/>
        <Value value="overcast"/>
        <Value value="rain"/>
      </DataField>
      <DataField name="whatIdo" optype="categorical" >
        <Value value="will play"/>
        <Value value="may play"/>
        <Value value="no play"/>
      </DataField>
    </DataDictionary>
    <TreeModel modelName="golfing" functionName="classification">
      <MiningSchema>
        <MiningField name="temperature"/>
        <MiningField name="humidity"/>
        <MiningField name="windy"/>
        <MiningField name="outlook"/>
        <MiningField name="whatIdo" usageType="predicted"/>
      </MiningSchema>
      <Node score="will play">
        <True/>
        <Node score="will play">
          <SimplePredicate field="outlook" operator="equal"
value="sunny"/>
          <Node score="will play">
            <CompoundPredicate booleanOperator="and" >
```

Listing 1.2 Extraction of a PMML document for a decision tree model

```
              <SimplePredicate field="temperature" operator="lessThan"
value="90" />
              <SimplePredicate field="temperature"
operator="greaterThan" value="50" />
            </CompoundPredicate>
            <Node score="will play" >
              <SimplePredicate field="humidity" operator="lessThan"
value="80" />
            </Node>
            <Node score="no play" >
              <SimplePredicate field="humidity"
operator="greaterOrEqual" value="80" />
            </Node>
          </Node>
          <Node score="no play" >
            <CompoundPredicate booleanOperator="or" >
              <SimplePredicate field="temperature"
operator="greaterOrEqual" value="90"/>
              <SimplePredicate field="temperature"
operator="lessOrEqual" value="50" />
            </CompoundPredicate>
          </Node>
        </Node>
        <Node score="may play" >
          <CompoundPredicate booleanOperator="or" >
            <SimplePredicate field="outlook" operator="equal"
value="overcast" />
            <SimplePredicate field="outlook" operator="equal"
value="rain" />
          </CompoundPredicate>
          <Node score="may play" >
            <CompoundPredicate booleanOperator="and" >
              <SimplePredicate field="temperature"
operator="greaterThan" value="60" />
              <SimplePredicate field="temperature" operator="lessThan"
value="100" />
              <SimplePredicate field="outlook" operator="equal"
value="overcast" />
              <SimplePredicate field="humidity" operator="lessThan"
value="70" />
              <SimplePredicate field="windy" operator="equal"
value="false" />
            </CompoundPredicate>
          </Node>
          <Node score="no play" >
            <CompoundPredicate booleanOperator="and" >
              <SimplePredicate field="outlook" operator="equal"
value="rain" />
```

Listing 1.2 *(continued)*

```
              <SimplePredicate field="humidity" operator="lessThan"
value="70" />
            </CompoundPredicate>
          </Node>
        </Node>
      </Node>
    </TreeModel>
  </PMML>
```

Listing 1.2 *(continued)*

Crisp-DM

The Crisp-DM data mining methodology was initialized by three companies: SPSS (ISL by then), NCR, and DaimlerChrysler in 1996. It was later sponsored by the European Community research fund. By August 2000, version 1.0 of Crisp-DM was published.

Crisp-DM does not describe a particular data mining technique; rather it focuses on the process of a data mining project's life cycle.

The Crisp-DM methodology can be described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction: phase, generic task, specialized task, and process instance.

Phase: The topic level of the process is called phase. For example, business understanding is the first phase of the data mining process.

Generic task: Generic task is the general description of tasks under each phase. This level is still quite abstract, for example, data cleaning is a generic task.

Specialized task: Specialized task describes how generic tasks can be carried out in certain specific situation. For example, the data cleansing task has special tasks such as cleaning numeric values and cleaning categorical values.

Process instances: Process instances is the lowest level of task, and contains records of actions, decisions, and the results of an actual data mining engagement. Figure 1.9 displays the life cycle of major phases of a data mining project. The project consists of six phases. The sequence of the phases is not always ordered. Moving back and forth between different phases is often required in a data mining project.

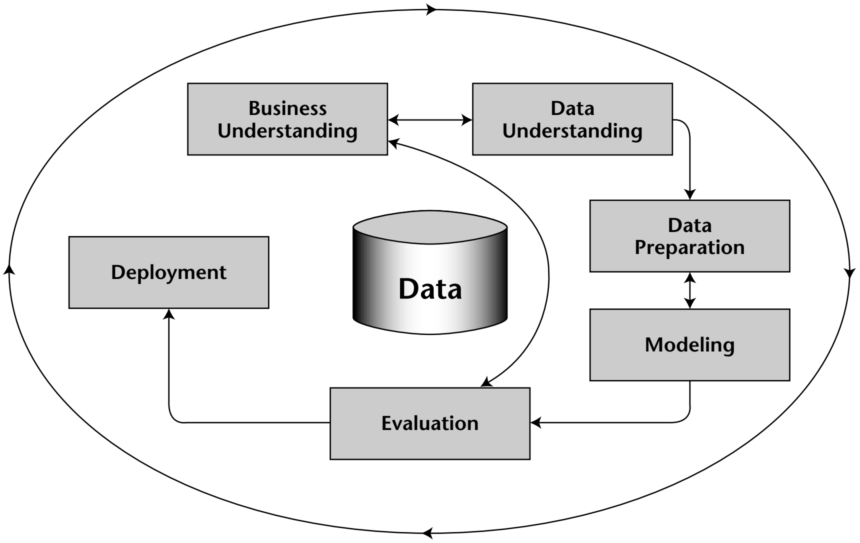


Figure 1.9 Phases of CRISP-DM reference model

Figure 1.10 describes an outline of the phases accompanied by generic tasks (bold) and outputs (italic). These tasks are quite straightforward. For example, the first phase is Business Understanding. In this phase of the project, there are four generic tasks: determine business objectives, assess situation, determine data mining goals, and produce the project plan.

In the determine business objectives task, there are three outputs: background, business objectives, and business success criteria. Crisp-DM further defines the detail for each output.

Common Warehouse Metadata

CWM stands for Common Warehouse Metadata (CWM) and it is led by the OMG CWM Working Group (IBM, Unisys, NCR, and a few other vendors). It addresses the metadata definition issue for the business intelligence field, including OALP, data mining, transformation, and so on. The goal of CWM is to solve the metadata management and integration problem for data warehouses, thus allowing different applications to be easily integrated in a heterogeneous environment.

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives *Background* *Business Objectives* *Business Success Criteria* **Assess Situation** *Inventory of Resources* *Requirements, Assumptions and Constraints* *Risks and Contingencies* *Terminology* *Costs and Benefits* **Determine Data Mining Goals** *Data Mining Goals* *Data Mining Success Criteria* **Produce Project Plan** *Project Plan* *Initial Assessment of Tools and Techniques*	**Collect Initial Data** *Initial Data Collection Report* **Describe Data** *Data Description Report* **Explore Data** *Data Exploration Report* **Verify Data Quality** *Data Quality Report*	**Data Set** *Data Set Description* **Select Data** *Rationale for Inclusion/Exclusion* **Clean Data** *Data Cleaning Report* **Construct Data** *Derived Attributes* *Generated Records* **Integrate Data** *Merged Data* **Format Data** *Reformatted Data*	**Select Modeling Techniques** *Modeling Techniques* *Modeling Assumptions* **Generate Text Design** *Text Design* **Build Model** *Parameter Settings* *Models* *Model Description* **Assess Model** *Model Assessment* *Revised Parameter Settings*	**Evaluate Results** *Assessment of Data Mining Results u.r.l. Business Success Criteria* *Approved Models* **Review Process** *Review of Process* **Determine Next Steps** *List of Possible Actions* *Decisions*	**Plan Deployment** *Deployment Plan* **Plan Monitoring and Maintenance** *Monitoring and Maintenance Plan* **Produce Final Report** *Final Report* *Final Presentation* **Review Project** *Experience Documentation*

Figure 1.10 Generic tasks and outputs of the CRISP-DM reference model

CWM is a complete specification of the syntax and semantics needed to export/import data warehouse metadata and meta models. It mainly includes:

- CWM Meta model (defined in UML)
- Interchange format for shared warehouse metadata (CWM DTD)
- Interchange format for the CWM Meta model (CWM XML)
- Access API for shared warehouse metadata (CWM IDL)

In the CWM specification, there is a data mining package that defines meta models for data mining.

Figure 1.11 presents the meta model related to the Model conceptual area. It consists of a representation of the `MiningModel`; the `MiningSettings`, which drive the construction of the model; the `ApplicationInputSpecification`, which specifies the set of input attributes for the model; and the `MiningModelResult`, which represents the result set produced by the testing or application of a generated model.

Apart from the Model conceptual area, there are other two conceptual areas: Settings and Attributes. The Settings conceptual area mainly focuses on the data mining algorithm parameter settings. There are four subclasses of mining settings: `StatisticsSettings`, `ClusteringSettings`, `SupervisedMiningSettings`, and `AssociationrulesSettings`. The `SupervisedMiningSettings` subclass has two subclasses: `ClassificationSettings` and `RegressionSettings`.

The Attributes conceptual area defines two subclasses of the Mining attribute: `NumericAttribute` and `CategoricalAttribute`.

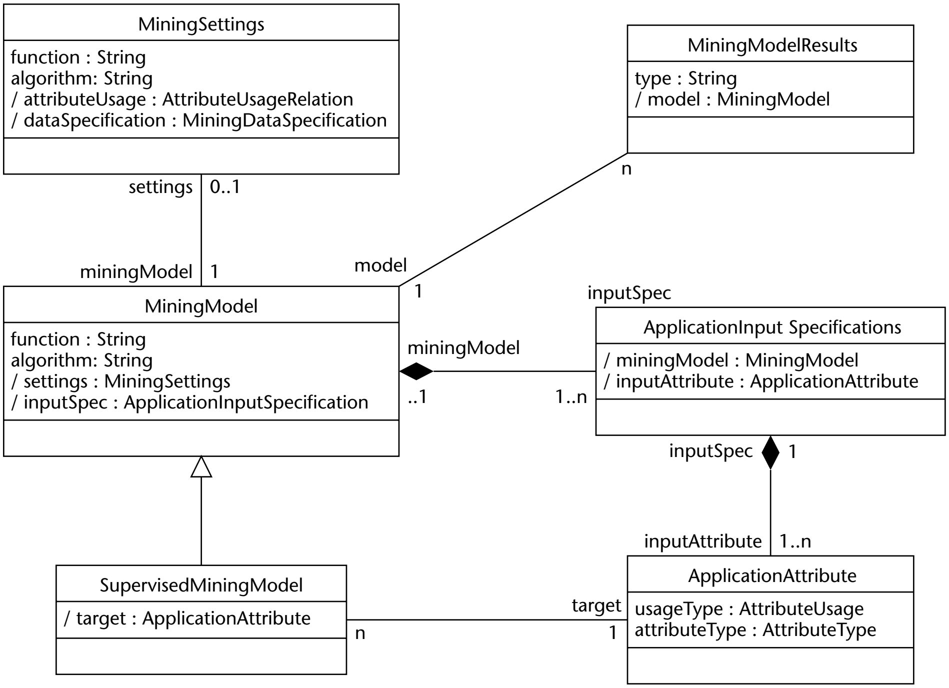

MiningSettings
function : String
algorithm: String
/ attributeUsage : AttributeUsageRelation
/ dataSpecification : MiningDataSpecification
settings
0..1
miningModel
1
MiningModelResults
type : String
/ model : MiningModel
n
model
1
MiningModel
function : String
algorithm: String
/ settings : MiningSettings
/ inputSpec : ApplicationInputSpecification
miningModel
..1
inputSpec
ApplicationInput Specifications
/ miningModel : MiningModel
/ inputAttribute : ApplicationAttribute
1..n
inputSpec
1
inputAttribute
1..n
SupervisedMiningModel
/ target : ApplicationAttribute
n
target
1
ApplicationAttribute
usageType : AttributeUsage
attributeType : AttributeType

Figure 1.11 The CWM data mining meta model: Model conceptual area

New Trends in Data Mining

Data mining is relatively young compared to database technology. It is still considered a niche and emerging market. One of the reasons for this designation is that most of the data mining packages are targeted toward statisticians or data miners. Application developers find it too challenging to master these technologies. More recently a number of data mining vendors, including Microsoft, realized this and initialized data mining APIs that are directed toward the developer communities. We believe that in the next few years there will be more and more developers who will be able to build mining models, and as a consequence a large number of applications will include data mining features.

We foresee the following trends in data mining area over the next few years:

Embedded data mining: More and more business applications will include data mining features, in particular the prediction feature, for added value. For example, the CRM application will allow users to forecast product sales. Online retailers will recommend products to customers for cross-selling purposes. This will be due largely to the fact that industrial data mining APIs, such as OLE DB for Data Mining, enable database and application developers to use data mining features and embed them in a line of business applications. Embedded data mining will increase the overall size of data mining market.

Data mining packages for vertical applications: Data mining is becoming popular as major database vendors add it to their database management system (DBMS) packages. Data mining can be applied to almost every sector. Today, major data mining markets exist in finance, insurance, and telecom. There is a growing need for specialized data mining techniques to solve business problems in many vertical sectors. For example, in the health care field, we need special data mining techniques to analyze DNA sequences. In network security applications, we need real time training algorithms to detect network intrusion. We need nontraditional data mining techniques to analyze the unstructured data in the World Wide Web. Text mining is yet another vertical sector to which we need to apply data mining. Traditional horizontal data mining packages are too general to solve these problems. We foresee there will be more new data mining packages specialized for these vertical sectors.

Products consolidation: Hundreds of software vendors are providing horizontal data mining packages. Many packages include only one or two algorithms. The data mining market is still very fragmented. Just as with other software sectors, consolidation is inevitable. Small vendors will find more competitive pressure in the horizontal market, especially when major database vendors add data mining features to DBMSs.

PMML: Although big vendors, such as Microsoft, Oracle, IBM, and SAS, are competing on various data mining APIs, they are all member of the same club: the DMG (Data Mining Group). They all support PMML as the model persistence format. PMML offers many advantages in terms of model exchange and model deployment. Because it is an XML document, it is also editable by advanced users. PMML will become more popular in the near future.

Summary

In this chapter, we have given you an extended introduction to data mining. By now, you should know the basics of data mining. There's nothing magic about it; it is about discovering hidden patterns from historical datasets and applying these patterns for predictions. There are a handful of data mining tasks, including classification, regression, association, clustering, forecasting, fraud detection, and visualization. These tasks cover hundreds of business scenarios. You learned the basic concepts of the set of data mining techniques and the typical life cycle of a data mining project.

The chapter also told you about the current data mining market and major product vendors. You learned about new standards in this field and the trends for data mining over the next few years.