CHAPTER

5

Data Mining for Business Intelligence

LEARNING OBJECTIVES

1. Define data mining as an enabling technology for business intelligence
2. Understand the objectives and benefits of business analytics and data mining
3. Recognize the wide range of applications of data mining
4. Learn the standardized data mining processes
5. Understand the steps involved in data preprocessing for data mining
6. Learn different methods and algorithms of data mining
7. Build awareness of the existing data mining software tools
8. Understand the pitfalls and myths of data mining

Generally speaking, data mining is a way to develop business intelligence from data that an organization collects, organizes, and stores. A wide range of data mining techniques are being used by organizations to gain a better understanding of their customers and their own operations and to solve complex organizational problems. In this chapter, we study data mining as an enabling technology for business intelligence, learn about the standard processes of conducting data mining projects, understand and build expertise in the use of major data mining techniques, develop awareness of the existing software tools, and explore common myths and pitfalls of data mining.

5.1 OPENING VIGNETTE: DATA MINING GOES TO HOLLYWOOD!

Predicting box-office receipts (i.e., financial success) of a particular motion picture is an interesting and challenging problem. According to some domain experts, the movie industry is the "land of hunches and wild guesses" due to the difficulty associated with forecasting product demand, making the movie business in Hollywood a risky endeavor. In support of such observations, Jack Valenti (the longtime president and CEO of the Motion Picture Association of America) once mentioned that ". . . no one can tell you how a movie is going to do in the marketplace . . . not until the film opens in darkened theatre and sparks fly up between the screen and the audience." Entertainment industry trade journals and magazines have been full of examples, statements, and experiences that support such a claim.

Like many other researchers who have attempted to shed light on this challenging real-world problem, Ramesh Sharda and Dursun Delen have been exploring the use of data mining to predict the financial performance of a motion picture at the box office before it even enters production (while the movie is nothing more than a conceptual idea). In their highly publicized prediction models, they convert the forecasting (or regression) problem into a classification problem; that is, rather than forecasting the point estimate of box-office receipts, they classify a movie based on its box-office receipts in one of nine categories, ranging from "flop" to "blockbuster," making the problem a multinomial classification problem. Table 5.1 illustrates the definition of the nine classes in terms of the range of box-office receipts.

Data

Data was collected from variety of movie-related databases (e.g., ShowBiz, IMDb, IMSDb, AllMovie, etc.) and consolidated into a single dataset. The dataset for the most recently developed models contained 2,632 movies released between 1998 and 2006. A summary of the independent variables along with their specifications is provided in Table 5.2. For more descriptive details and justification for inclusion of these independent variables, the reader is referred to Sharda and Delen (2007).

TABLE 5.1 Movie Classification Based on Receipts

Class No.	1	2	3	4	5	6	7	8	9
Range (in millions of dollars)	<1 (Flop)	>1 <10	>10 <20	>20 <40	>40 <65	>65 <100	>100 <150	>150 <200	>200 (Blockbuster)

TABLE 5.2 Summary of Independent Variables

Independent Variable	Number of Values	Possible Values
MPAA Rating	5	G, PG, PG-13, R, NR
Competition	3	High, Medium, Low
Star value	3	High, Medium, Low
Genre	10	Sci-Fi, Historic Epic Drama, Modern Drama, Politically Related, Thriller, Horror, Comedy, Cartoon, Action, Documentary
Special effects	3	High, Medium, Low
Sequel	1	Yes, No
Number of screens	1	Positive integer

Methodology

Using a variety of data mining methods, including neural networks, decision trees, support vector machines, and three types of ensembles, Sharda and Delen developed the prediction models. The data from 1998 to 2005 were used as training data to build the prediction models, and the data from 2006 was used as the test data to assess and compare the models' prediction accuracy. Figure 5.1 shows a screenshot of SPSS's PASW Modeler (formerly Clementine data mining tool) depicting the process map employed for the prediction problem. The upper-left side of the process map shows the model development process, and the lower-right corner of the process map shows the model assessment (i.e., testing or scoring) process (more details on PASW Modeler tool and its usage can be found on the book's Web site).

Results

Table 5.3 provides the prediction results of all three data mining methods as well as the results of the three different ensembles. The first performance measure is the percent correct classification rate, which is called *bingo*. Also reported in the table is the *1-Away* correct classification rate (i.e., within one category). The results indicate that SVM performed the best among the individual prediction models, followed by ANN; the worst of the three was the CART decision tree algorithm. In general, the ensemble models performed better than the individual predictions models, of which the fusion algorithm performed the best. What is probably more important to decision makers, and standing out in the results table, is the significantly low standard deviation obtained from the ensembles compared to the individual models.

Conclusion

The researchers claim that these prediction results are better than any reported in the published literature for this problem domain. Beyond the attractive accuracy of their prediction results of the box-office receipts, these models could also be used to further analyze (and potentially

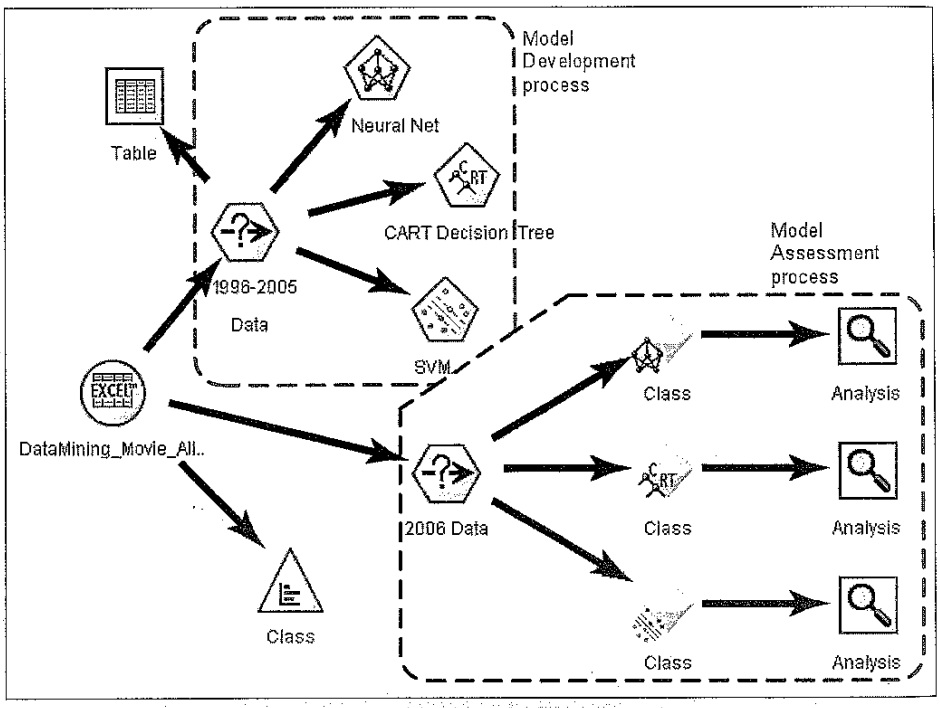


FIGURE 5.1 Process-Flow Screenshot for the Box-Office Prediction System *Source:* Used with permission from SPSS.

TABLE 5.3 Tabulated Prediction Results for Individual and Ensemble Models

	Prediction Models					
	Individual Models			Ensemble Models		
Performance Measure	SVM	ANN	C&RT	Random Forest	Boosted Tree	Fusion (Average)
Count (Bingo)	192	182	140	189	187	**194**
Count (1-Away)	104	120	126	121	104	**120**
Accuracy (% Bingo)	55.49%	52.60%	40.46%	54.62%	54.05%	**56.07%**
Accuracy (% 1-Away)	85.55%	87.28%	76.88%	89.60%	84.10%	**90.75%**
Standard deviation	0.93	0.87	1.05	0.76	0.84	**0.63**

optimize) the decision variables in order to maximize the financial return. Specifically, the parameters used for modeling could be altered using the already trained prediction models in order to better understand the impact of different parameters on the end results. During this process, which is commonly referred to as *sensitivity analysis*, the decision maker of a given entertainment firm could find out, with a fairly high accuracy level, how much value a specific actor (or a specific release date, or the addition of more technical effects, etc.) brings to the financial success of a film, making the underlying system an invaluable decision aid.

Questions for the Opening Vignette

1. Why should Hollywood decision makers use data mining?
2. What are the top challenges for Hollywood managers? Can you think of other industry segments that face similar problems?
3. Do you think the researchers used all of the relevant data to build prediction models?
4. Why do you think the researchers chose to convert a regression problem into a classification problem?
5. How do you think these prediction models can be used? Can you think of a good production system for such models?
6. Do you think the decision makers would easily adapt to such an information system?
7. What can be done to further improve the prediction models explained in this case?

What We Can Learn from This Vignette

The entertainment industry is full of interesting and challenging problems for decision makers. Making the right decisions to manage large amounts of money is critical to success (or mere survival) of many companies in this marketplace. Data mining is a prime candidate for better management of this data-rich, knowledge-poor business environment. The study described in the opening vignette clearly illustrates the power of data mining in predicting and explaining the financial outlook of a motion picture, which most still think is a form of art and hence cannot be forecasted. In this chapter, you will see a wide variety of data mining applications solving complex problems in a variety of industries where the data can be used to leverage competitive business advantage.

Sources: R. Sharda and D. Delen, "Predicting Box-Office Success of Motion Pictures with Neural Networks," *Expert Systems with Applications,* Vol. 30, 2006, pp. 243–254; D. Delen, R. Sharda, and P. Kumar, "Movie Forecast Guru: A Web-based DSS for Hollywood Managers," *Decision Support Systems,* Vol. 43, No. 4, 2007, pp. 1151–1170.

5.2 DATA MINING CONCEPTS AND APPLICATIONS

In an interview with *Computerworld* magazine in January 1999, Dr. Arno Penzias (Nobel laureate and former chief scientist of Bell Labs) identified data mining from organizational databases as a key application for corporations of the near future. In response to *Computerworld*'s age-old question of "What will be the killer applications in the corporation?" Dr. Penzias replied: "Data mining." He then added, "Data mining will become much more important and companies will throw away nothing about their customers because it will be so valuable. If you're not doing this, you're out of business." Similarly, in an article in *Harvard Business Review* Thomas Davenport (2006) argued that the latest strategic weapon for companies is analytical decision making, providing examples of companies such as Amazon.com, Capital One, Marriott International, and others that have used analytics to better understand their customers and optimize their extended supply chains to maximize their returns on investment while providing the best customer service. This level of success is highly dependent on a company understanding its customers, vendors, business processes, and the extended supply chain very well.

A large component of this understanding comes from analyzing the vast amount of data that a company collects. The cost of storing and processing data has decreased dramatically in the recent past, and, as a result, the amount of data stored in electronic form has grown at an explosive rate. With the creation of large databases, the possibility of analyzing the data stored in them has emerged. The term *data mining* was originally used to describe the process through which previously unknown patterns in data were discovered. This definition has since been stretched beyond those limits by some software vendors to include most forms of data analysis in order to increase sales with the popularity of the data mining label. In this chapter, we accept the original definition of data mining.

Although the term *data mining* is relatively new, the ideas behind it are not. Many of the techniques used in data mining have their roots in traditional statistical analysis and artificial intelligence work done since the early part of 1980s. Why, then, has it suddenly gained the attention of the business world? Following are some of most pronounced reasons:

- More intense competition at the global scale driven by customers' ever-changing needs and wants in an increasingly saturated marketplace.
- General recognition of the untapped value hidden in large data sources.
- Consolidation and integration of database records, which enables a single view of customers, vendors, transactions, etc.
- Consolidation of databases and other data repositories into a single location in the form of a data warehouse.
- The exponential increase in data processing and storage technologies.
- Significant reduction in the cost of hardware and software for data storage and processing.
- Movement toward the de-massification (conversion of information resources into nonphysical form) of business practices.

Data generated by the Internet is increasing rapidly in both volume and complexity. Large amounts of genomic data are being generated and accumulated all over the world. Disciplines such as astronomy and nuclear physics create huge quantities of data on a regular basis. Medical and pharmaceutical researchers constantly generate and store data that can then be used in data mining applications to identify better ways to accurately diagnose and treat illnesses and to discover new and improved drugs.

On the commercial side, perhaps the most common use of data mining has been in the finance, retail, and health care sectors. Data mining is used to detect and reduce fraudulent activities, especially in insurance claims and credit card use (Chan et al., 1999); to identify customer buying patterns (Hoffman, 1999); to reclaim profitable customers

(Hoffman, 1998); to identify trading rules from historical data; and to aid in increased profitability using market-basket analysis. Data mining is already widely used to better target clients, and with the widespread development of e-commerce, this can only become more imperative with time. See Application Case 5.1 for information on how 1-800-Flowers has used business analytics and data mining to excel in business.

APPLICATION CASE 5.1

Business Analytics and Data Mining Help 1-800-Flowers Excel in Business

1-800-Flowers is one of the best-known and most-successful brands in the gift-retailing business. For more than 30 years, the New York–based company has been providing customers around the world with the freshest flowers and finest selection of plants, gift baskets, gourmet foods, confections, and plush stuffed animals for every occasion. Founded by Jim McCann in 1976, 1-800-Flowers has quickly become the leader in direct-order e-commerce after opening its own Web site more than 14 years ago.

Problem

As successful as it has been, like many other companies involved in e-commerce, 1-800-Flowers needed to make decisions in real time to increase retention, reduce costs, and keep its best customers coming back for more again and again. As the business has grown from one flower shop to an online gift retailer serving more than 30 million customers, it has needed to stay ahead of the competition by being the best that it can be.

Solution

Strongly believing in the value of close customer relationships, 1-800-Flowers wanted to better understand its customers' needs and wants by analyzing every piece of data that it had about them. 1-800-Flowers decided to use SAS data mining tools to dig deep into its data assets to discover novel patterns about its customers and turn that knowledge into business transactions.

Results

According to McCann, business analytics and data mining tools from SAS have enabled 1-800-Flowers to grow its business regardless of the conditions of the larger economy. At a time when other retailers are struggling to survive, 1-800-Flowers has seen revenues grow, nearly doubling in the last 5 years.

Specific benefits of the analysis were as follows:

- ***More efficient marketing campaigns.*** 1-800-Flowers has drastically reduced the time it takes to segment customers for direct mailing. "It used to take 2 or 3 weeks—now it takes 2 or 3 days," says Aaron Cano, Vice President of Customer Knowledge Management. "That leaves us time to do more analysis and make sure we're sending relevant offers."
- ***Reduced mailings, increased response rates.*** The company has been able to significantly reduce marketing mailings while increasing response rates and be much more selective about TV and radio advertisements.
- ***Better customer experience.*** When a repeat customer logs on to 1-800-Flowers.com, the Web site immediately shows selections that are related to that customer's interests. "If a customer usually buys tulips for his wife, we show him our newest and best tulip selections," Cano says.
- ***Increased repeat sales.*** The company's best customers are returning more often because 1-800-Flowers knows who the customer is and what he or she likes. The company makes the shopping experience easy and relevant and markets to customers at the point of contact.

As a result of using business analytics and data mining, 1-800-Flowers reduced its operating expenses, increased the retention rate of its best customer segment to more than 80 percent, attracted 20 million new customers, and grew the overall repeat business from less than 40 percent to greater than 50 percent (a 10-basis-point increase in repeat sales across all brands translates into $40 million additional revenue for the business).

Sources: "SAS Helps 1-800-Flowers.com Grow Deep Roots with Customers," **sas.com/success/1800flowers.html** (accessed on May 23, 2009); "Data Mining at 1-800-Flowers," **kdnuggets.com/news/2009/n10/3i.html** (accessed on May 26, 2009).

Definitions, Characteristics, and Benefits

Simply defined, **data mining** is a term used to describe discovering or "mining" knowledge from large amounts of data. When considered by analogy, one can easily realize that the term *data mining* is a misnomer; that is, mining of gold from within rocks or dirt is referred to as "gold" mining rather than "rock" or "dirt" mining. Therefore, data mining perhaps should have been named "knowledge mining" or "knowledge discovery." Despite the mismatch between the term and its meaning, *data mining* has become the choice of the community. Many other names that are associated with data mining include *knowledge extraction, pattern analysis, data archaeology, information harvesting, pattern searching,* and *data dredging.*

Technically speaking, data mining is a process that uses statistical, mathematical, and artificial intelligence techniques to extract and identify useful information and subsequent knowledge (or patterns) from large sets of data. These patterns can be in the form of business rules, affinities, correlations, trends, or prediction models (see Nemati and Barko, 2001). Most literature defines data mining as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data stored in structured databases," where the data are organized in records structured by categorical, ordinal, and continuous variables (Fayyad et al., 1996). In this definition, the meanings of the key term are as follows:

- *Process* implies that data mining comprises many iterative steps.
- *Nontrivial* means that some experimentation-type search or inference is involved; that is, it is not as straightforward as a computation of predefined quantities.
- *Valid* means that the discovered patterns should hold true on new data with sufficient degree of certainty.
- *Novel* means that the patterns are not previously known to the user within the context of the system being analyzed.
- *Potentially useful* means that the discovered patterns should lead to some benefit to the user or task.
- *Ultimately understandable* means that the pattern should make business sense that leads to user saying "mmm! It makes sense; why didn't I think of that" if not immediately, at least after some post processing.

Data mining is not a new discipline, but rather a new definition for the use of many disciplines. Data mining is tightly positioned at the intersection of many disciplines, including statistics, artificial intelligence, machine learning, management science, information systems, and databases (see Figure 5.2). Using advances in all of these disciplines, data mining strives to make progress in extracting useful information and knowledge from large databases. It is an emerging field that has attracted much attention in a very short time.

The following are the major characteristics and objectives of data mining:

- Data are often buried deep within very large databases, which sometimes contain data from several years. In many cases, the data are cleansed and consolidated into a data warehouse.
- The data mining environment is usually a client-server architecture or a Web-based information systems architecture.
- Sophisticated new tools, including advanced visualization tools, help to remove the information ore buried in corporate files or archival public records. Finding it involves massaging and synchronizing the data to get the right results. Cutting-edge data miners are also exploring the usefulness of soft data (i.e., unstructured text stored in such places as Lotus Notes databases, text files on the Internet, or enterprise-wide intranets).
- The miner is often an end user, empowered by data drills and other power query tools to ask ad hoc questions and obtain answers quickly, with little or no programming skill.
- Striking it rich often involves finding an unexpected result and requires end users to think creatively throughout the process, including the interpretation of the findings.

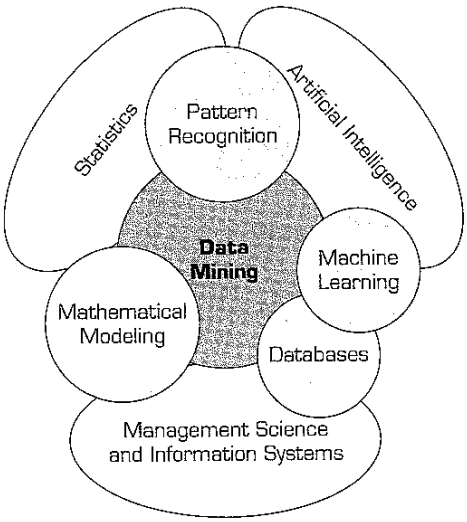


FIGURE 5.2 Data Mining as a Blend of Multiple Disciplines

- Data mining tools are readily combined with spreadsheets and other software development tools. Thus, the mined data can be analyzed and deployed quickly and easily.
- Because of the large amounts of data and massive search efforts, it is sometimes necessary to use parallel processing for data mining.

A company that effectively leverages data mining tools and technologies can acquire and maintain a strategic competitive advantage. Data mining offers organizations an indispensable decision-enhancing environment to exploit new opportunities by transforming data into a strategic weapon. See Nemati and Barko (2001) for a more detailed discussion on the strategic benefits of data mining.

TECHNOLOGY INSIGHTS 5.1 Data in Data Mining

Data refers to a collection of facts usually obtained as the result of experiences, observations, or experiments. Data may consist of numbers, words, images, and so on as measurements of a set of variables. Data are often viewed as the lowest level of abstraction from which information and knowledge are derived.

At the highest level of abstraction, one can classify data as categorical or numeric. The categorical data can be subdivided into nominal or ordinal data, whereas numeric data can be subdivided into interval or ratio. Figure 5.3 shows a simple taxonomy of data in data mining.

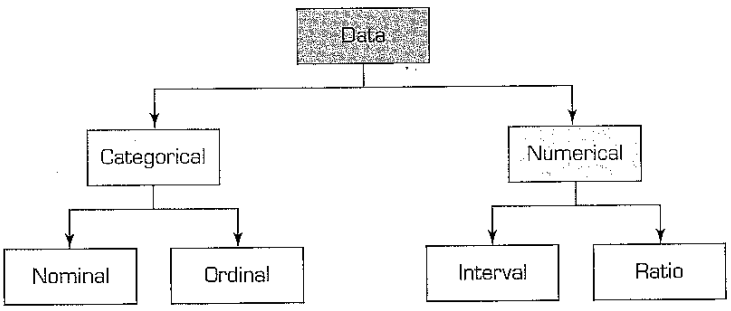


FIGURE 5.3 A Simple Taxonomy of Data in Data Mining

- **Categorical data** represent the labels of multiple classes used to divide a variable into specific groups. Examples of categorical variables include race, sex, age group, and educational level. Although the latter two variables may also be considered in a numerical manner by using exact values for age and highest grade completed, it is often more informative to categorize such variables into a relatively small number of ordered classes. The categorical data may also be called discrete data implying that it represents a finite number of values with no continuum between them. Even if the values used for the categorical (or discrete) variables are numeric, these numbers are nothing more than symbols and do not imply the possibility of calculating fractional values.
- **Nominal data** contain measurements of simple codes assigned to objects as labels, which are not measurements. For example, the variable *marital status* can be generally categorized as (1) single, (2) married, and (3) divorced. Nominal data can be represented with binomial values having two possible values (e.g., yes/no, true/false, good/bad), or multinomial values having three or more possible values (e.g., brown/green/blue, white/black/Latino/Asian, single/married/divorced).
- **Ordinal data** contain codes assigned to objects or events as labels that also represent the rank order among them. For example, the variable *credit score* can be generally categorized as (1) low, (2) medium, or (3) high. Similar ordered relationships can be seen in variables such as age group (i.e., child, young, middle-aged, elderly) and educational level (i.e., high school, college, graduate school). Some data mining algorithms, such as *ordinal multiple logistic regression*, take into account this additional rank order information to build a better classification model.
- **Numeric data** represent the numeric values of specific variables. Examples of numerically valued variables include age, number of children, total household income (in US dollars), travel distance (in miles), and temperature (in Fahrenheit degrees). Numeric values representing a variable can be integer (taking only whole numbers) or real (taking also the fractional number). The numeric data may also be called continuous data, implying that the variable contains continuous measures on a specific scale that allows insertion of interim values. Unlike a discrete variable, which represents finite, countable data, a continuous variable represents scalable measurements, and it is possible for the data to contain an infinite number of fractional values.
- **Interval data** are variables that can be measured on interval scales. A common example of interval scale measurement is temperature on the Celsius scale. In this particular scale, the unit of measurement is 1/100 of the difference between the melting temperature and the boiling temperature of water in atmospheric pressure; that is, there is not an absolute zero value.
- **Ratio data** include measurement variables commonly found in the physical sciences and engineering. Mass, length, time, plane angle, energy, and electric charge are examples of physical measures that are ratio scales. The scale type takes its name from the fact that measurement is the estimation of the ratio between a magnitude of a continuous quantity and a unit magnitude of the same kind. Informally, the distinguishing feature of a ratio scale is the possession of a nonarbitrary zero value. For example, the Kelvin temperature scale has a nonarbitrary zero point of absolute zero, which is equal to –273.15 degrees Celsius. This zero point is nonarbitrary, because the particles that comprise matter at this temperature have zero kinetic energy.
- Other data types include date/time, unstructured text, image, and audio. These data types need to be converted into some form of categorical or numeric representation before they can be processed by data mining algorithms. Data can also be classified as static or dynamic (i.e., temporal or time-series).

Some data mining methods are particular about the data types they can handle. Providing them with incompatible data types may lead to incorrect models or (more often) halt the model development process. For example, some data mining methods need all of the variables (both input as well as output) represented as numerically valued variables (e.g., neural networks, support vector machines, logistic regression). The nominal or ordinal variables are converted into numeric representations using some type of *1-of-N* pseudo variables (e.g., a categorical variable with three unique values can be transformed into three pseudo variables with binary

values—1 or 0). Because this process may increase the number of variables, one should be cautious about the effect of such representations, especially for the categorical variables that have large numbers of unique values.

Similarly, some data mining methods, such as ID3 (a classic decision tree algorithm) and rough sets (a relatively new rule induction algorithm), need all of the variables represented as categorically valued variables. Early versions of these methods required the user to discretize numeric variables into categorical representations before they could be processed by the algorithm. The good news is that most implementations of these algorithms in widely available software tools accept a mix of numeric and nominal variables and internally make the necessary conversions before processing the data.

APPLICATION CASE 5.2

Law Enforcement Organizations Use Data Mining to Better Fight Crime

In the midst of these unfavorable economic conditions, police departments all over the world are facing difficult times in fighting crimes with continually shrinking resources along with fewer leads, a larger number of cases, and increasingly more complicated crimes. At a police department in the United Kingdom, investigators find that these challenges limit the cases they can tackle. A high volume of cases without definite leads—such as house burglaries and vehicle thefts that lack clear evidence—are often filed away until new evidence is found. Therefore, the challenge for the police department was to determine a way to quickly and easily find patterns and trends in unsolved criminal cases.

Each electronic case file at the police department contains physical descriptions of the thieves as well as their modus operandi (MO). Whereas many cases lacking evidence were previously filed away, the department is now re-examining them and doing it more quickly than ever before. In PASW Modeler (formerly Clementine), the data modeler uses two Kohonen neural network models to cluster similar physical descriptions and MOs and then combines clusters to see whether groups of similar physical descriptions coincide with groups of similar MOs. If a good match is found and the perpetrators are known for one or more of the offenses, it is possible that the unsolved cases were committed by the same individuals.

The analytical team further investigates the clusters using statistical methods to verify the similarities' importance. If clusters indicate that the same criminal may be at work, the department is likely to reopen and investigate the other crimes. Or, if the criminal is unknown but a large cluster indicates the same offender, the leads from these cases can be combined and the case reprioritized. The department is also investigating the behavior of prolific repeat offenders with the goal of identifying crimes that seem to fit their behavioral pattern. The department hopes that the PASW Modeler will enable it to reopen old cases and make connections with known perpetrators.

Another police department in the United States is facing similar challenges: lack of sufficient resources coupled with an increasing number of criminal cases. In order to produce sustainable solutions to a wide range of crime and community disorders, the department is pursuing a community-oriented policing philosophy, which is a holistic approach requiring collaborative partnerships between citizens and community agencies and careful analysis of information surrounding the criminal cases. The underlying process aims to find long-term solutions to crimes by identifying their root causes, educating the community on the extent of the problems, and then working with the community to develop collaborative solutions that effectively address these causes. The main challenge was to convince the community that their involvement is necessary for any solution to be effective.

Using PASW statistical analysis and a data mining software tool, the police department conducted extensive data analysis to discover the variables strongly associated with the criminal cases, as well as assess citizen satisfaction with community policing. The results of this analysis presented compelling

evidence that community involvement coupled with intelligent data analysis are necessary ingredients in developing effective long-term solutions in the midst of economic difficulties.

Police departments around the globe are enhancing their crime-fighting techniques with innovative twenty-first-century approaches of applying data mining technology to prevent criminal activity. Success stories can be found on Web sites of major data mining tool and solution providers (e.g., SPSS, SAS, StatSoft, Salford Systems). as well as the major consultancy companies.

Sources: Based on C. McCue "Connecting the Dots: Data Mining and Predictive Analytics in Law Enforcement and Intelligence Analysis," *Police Chief Magazine,* Vol. 70, No. 10, October 2003; "Police Department Fights Crime with SPSS Inc. Technology," **spss.com/success/pdf/WMPCS-1208.pdf** (accessed July 25, 2009); "North Carolina Law Enforcement Agency Identifies Crime Areas and Secures Community Involvement," **spssshowcase.co.uk/success/pdf/CMPDCS-0109.pdf** (accessed on September 14, 2009).

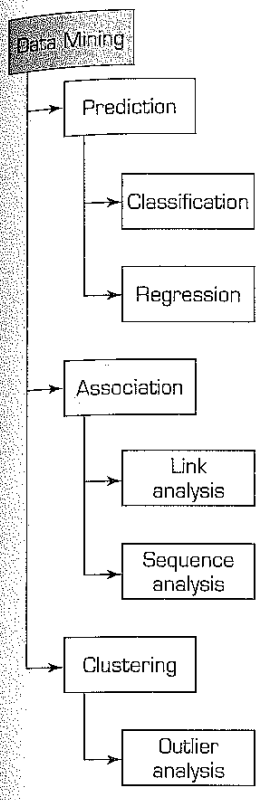
How Data Mining Works

Using existing and relevant data, data mining builds models to identify patterns among the attributes presented in the dataset. Models are the mathematical representations (simple linear relationships and/or complex highly nonlinear relationships) that identify the patterns among the attributes of the objects (e.g., customers) described in the dataset. Some of these patterns are explanatory (explaining the interrelationships and affinities among the attributes), whereas others are predictive (foretelling future values of certain attributes). In general, data mining seeks to identify four major types of patterns:

1. *Associations* find the commonly co-occurring groupings of things, such as beer and diapers going together in market-basket analysis.
2. *Predictions* tell the nature of future occurrences of certain events based on what has happened in the past, such as predicting the winner of the Super Bowl or forecasting the absolute temperature of a particular day.
3. *Clusters* identify natural groupings of things based on their known characteristics, such as assigning customers in different segments based on their demographics and past purchase behaviors.
4. *Sequential relationships* discover time-ordered events, such as predicting that an existing banking customer who already has a checking account will open a savings account followed by an investment account within a year.

These types of patterns have been *manually* extracted from data by humans for centuries, but the increasing volume of data in modern times has created a need for more automatic approaches. As datasets have grown in size and complexity, direct manual data analysis has increasingly been augmented with indirect, automatic data processing tools that use sophisticated methodologies, methods, and algorithms. The manifestation of such evolution of automated and semiautomated means of processing large datasets is now commonly referred to as *data mining.*

Generally speaking, data mining tasks can be classified into three main categories: prediction, association, and clustering. Based on the way in which the patterns are extracted from the historical data, the learning algorithms of data mining methods can be classified as either supervised or unsupervised. With supervised learning algorithms, the training data includes both the descriptive attributes (i.e., independent variables or decision variables) as well as the class attribute (i.e., output variable or result variable). In contrast, with unsupervised learning the training data includes only the descriptive attributes. Figure 5.4 shows a simple taxonomy for data mining tasks, along with the learning methods, and popular algorithms for each of the data mining tasks.



Learning Method	Popular Algorithms
Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
Supervised	Decision Trees, ANN/MLP, SVM, Rough Sets, Genetic Algorithms
Supervised	Linear/Nonlinear Regression, Regression Trees, ANN/MLP, SVM
Unsupervised	Apriori, Oner, ZeroR, Eclat
Unsupervised	Expectation Maximization Apriori Algorithm, Graph-Based Matching
Unsupervised	Apriori Algorithm, FP-Growth technique
Unsupervised	K-means, ANN/SOM
Unsupervised	K-means, Expectation Maximization (EM)

FIGURE 5.4 A Simple Taxonomy for Data Mining Tasks

PREDICTION **Prediction** is commonly referred to as the act of telling about the future. It differs from simple guessing by taking into account the experiences, opinions, and other relevant information in conducting the task of foretelling. A term that is commonly associated with prediction is *forecasting*. Even though many believe that these two terms are synonymous, there is a subtle but critical difference between the two. Whereas prediction is largely experience and opinion based, forecasting is data and model based. That is, in order of increasing reliability, one might list the relevant terms as *guessing, predicting,* and *forecasting,* respectively. In data mining terminology, *prediction* and *forecasting* are used synonymously, and the term *prediction* is used as the common representation of the act. Depending on the nature of what is being predicted, prediction can be named more specifically as classification (where the predicted thing, such as tomorrow's forecast, is a class label such as "rainy" or "sunny") or regression (where the predicted thing, such as tomorrow's temperature, is a real number, such as "65°F").

CLASSIFICATION **Classification**, or supervised induction, is perhaps the most common of all data mining tasks. The objective of classification is to analyze the historical data stored in a database and automatically generate a model that can predict future behavior. This induced model consists of generalizations over the records of a training dataset, which help distinguish predefined classes. The hope is that the model can then be used to predict the classes of other unclassified records and, more important, to accurately predict actual future events.

Common classification tools include neural networks and decision trees (from machine learning), logistic regression and discriminant analysis (from traditional statistics), and

emerging tools such as rough sets, support vector machines, and genetic algorithms. Statistics-based classification techniques (e.g., logistic regression and discriminant analysis) have received their share of criticism—that they make unrealistic assumptions about the data, such as independence and normality—which limit their use in classification-type data mining projects.

Neural networks (see Chapter 6 for a more detailed coverage of this popular machine learning algorithm) involve the development of mathematical structures (somewhat resembling the biological neural networks in the human brain) that have the capability to learn from past experiences presented in the form of well-structured datasets. They tend to be more effective when the number of variables involved is rather large and the relationships among them are complex and imprecise. Neural networks have disadvantages as well as advantages. For example, it is usually very difficult to provide a good rationale for the predictions made by a neural network. Also, neural networks tend to need considerable training. Unfortunately, the time needed for training tends to increase exponentially as the volume of data increases, and, in general, neural networks cannot be trained on very large databases. These and other factors have limited the applicability of neural networks in data-rich domains.

Decision trees classify data into a finite number of classes based on the values of the input variables. Decision trees are essentially a hierarchy of if-then statements and are thus significantly faster than neural networks. They are most appropriate for categorical and interval data. Therefore, incorporating continuous variables into a decision tree framework requires *discritization*; that is, converting continuous valued numerical variables to ranges and categories.

A related category of classification tools is rule induction. Unlike with a decision tree, with rule induction the if-then statements are induced from the training data directly, and they need not be hierarchical in nature. Other, more recent techniques such as SVM, rough sets, and genetic algorithms are gradually finding their way into the arsenal of classification algorithms and are covered in more detail in Chapter 13 as part of the discussion on advanced intelligent systems.

CLUSTERING **Clustering** partitions a collection of things (e.g., objects, events, etc. presented in a structured dataset) into segments (or natural groupings) whose members share similar characteristics. Unlike in classification, in clustering the class labels are unknown. As the selected algorithm goes through the dataset, identifying the commonalities of things based on their characteristics, the clusters are established. Because the clusters are determined using a heuristic-type algorithm, and because different algorithms may end up with different sets of clusters for the same dataset, before the results of clustering techniques are put to actual use it may be necessary for an expert to interpret, and potentially modify, the suggested clusters. After reasonable clusters have been identified, they can be used to classify and interpret new data.

Not surprisingly, clustering techniques include optimization. The goal of clustering is to create groups so that the members within each group have maximum similarity and the members across groups have minimum similarity. The most commonly used clustering techniques include *k*-means (from statistics) and self-organizing maps (from machine learning), which is a unique neural network architecture developed by Kohonen (1982).

Firms often effectively use their data mining systems to perform market segmentation with cluster analysis. Cluster analysis is a means of identifying classes of items so that items in a cluster have more in common with each other than with items in other clusters. It can be used in segmenting customers and directing appropriate marketing products to the segments at the right time in the right format at the right price. Cluster analysis is also used to identify natural groupings of events or objects so that a common set of characteristics of these groups can be identified to describe them. Application Case 5.3 describes how cluster analysis was combined with other data mining techniques to identify the causes of accidents.

APPLICATION CASE 5.3

Motor Vehicle Accidents and Driver Distractions

Driver distraction is at center stage in highway safety. A study published in 1996 by the National Highway Traffic Safety Administration (NHTSA) concluded that roughly 25 to 30 percent of the injuries caused by car crashes were due to driver distraction. In 1999, according to the Fatality Analysis Reporting System (FARS) developed by the National Center for Statistics and Analysis (NCSA), 11 percent of fatal crashes (i.e., 4,462 fatalities) were due to driver inattention.

A study was conducted to extract the patterns of distraction factors at traffic accidents. Data mining was used to draw the correlations and associations of factors from the crash datasets provided by FARS. Three data mining techniques (Kohonen-type neural networks, decision trees, and multilayer perceptron-type neural networks) were used to find different combinations of distraction factors that correlated with and potentially explained the high accident rates. The Kohonen-type neural network identified natural clusters and revealed patterns of input variables in the collection of data. Decision trees explored and classified the effect of each incident on successive events and also suggested the relationship between inattentive drivers and physical/mental conditions. Finally, a multilayer perceptron-type neural network model was trained and tested to discover the relationships between inattention and other driver-related factors in these traffic crashes. Clementine from SPSS was used to mine the data obtained from the FARS database for all three model types.

The prediction and exploration model identified 1,255 drivers who were involved in accidents in which inattention was one of the leading driver factors that led to a crash. Rear, head-on, and angled collisions, among other various output variables, were among the factors that had significant impact on the occurrence of crashes and their severity.

Sources: W. S. Tseng, H. Nguyen, J. Liebowitz, and W. Agresti, "Distractions and Motor Vehicle Accidents: Data Mining Application on Fatality Analysis Reporting System (FARS) Data Files," *Industrial Management & Data Systems,* Vol. 105, No. 9, January 2005, pp. 1188–1205; and J. Liebowitz, "New Trends in Intelligent Systems," Presentation made at University of Granada, **docto-si.ugr.es/seminario2006/presentaciones/jay.ppt** (accessed May 2009).

ASSOCIATIONS **Associations**, or *association rule learning in data mining,* is a popular and well-researched technique for discovering interesting relationships among variables in large databases. Thanks to automated data-gathering technologies such as bar code scanners, the use of association rules for discovering regularities among products in large-scale transactions recorded by point-of-sale systems in supermarkets has become a common knowledge-discovery task in the retail industry. In the context of the retail industry, association rule mining is often called *market-basket analysis.*

Two commonly used derivatives of association rule mining are **link analysis** and **sequence mining**. With link analysis, the linkage among many objects of interest is discovered automatically, such as the link between Web pages and referential relationships among groups of academic publication authors. With sequence mining, relationships are examined in terms of their order of occurrence to identify associations over time. Algorithms used in association rule mining include the popular Apriori (where frequent itemsets are identified) and FP-Growth, OneR, ZeroR, and Eclat.

VISUALIZATION AND TIME-SERIES FORECASTING Two techniques often associated with data mining are *visualization* and *time-series forecasting*. Visualization can be used in conjunction with other data mining techniques to gain a clearer understanding of underlying relationships. With time-series forecasting, the data are a series of values of the same variable that is captured and stored over time. These data are then used to develop models to extrapolate the future values of the same phenomenon.

HYPOTHESIS- OR DISCOVERY-DRIVEN DATA MINING Data mining can be hypothesis driven or discovery driven. **Hypothesis-driven data mining** begins with a proposition by the user, who then seeks to validate the truthfulness of the proposition. For example, a marketing manager may begin with the following proposition: "Are DVD player sales related to sales of television sets?"

Discovery-driven data mining finds patterns, associations, and other relationships hidden within datasets. It can uncover facts that an organization had not previously known or even contemplated.

Section 5.2 Review Questions

1. Define *data mining*. Why are there many different names and definitions for data mining?
2. What recent factors have increased the popularity of data mining?
3. Is data mining a new discipline? Explain.
4. What are some major data mining methods and algorithms?
5. What are the key differences between the major data mining methods?

5.3 DATA MINING APPLICATIONS

Data mining has become a popular tool in addressing many complex businesses issues. It has been proven to be very successful and helpful in many areas, some of which are shown by the following representative examples. The goal of many of these business data mining applications is to solve a pressing problem or to explore an emerging business opportunity in order to create a sustainable competitive advantage.

- ***Customer relationship management.*** Customer relationship management (CRM) is the new and emerging extension of traditional marketing. The goal of CRM is to create one-on-one relationships with customers by developing an intimate understanding of their needs and wants. As businesses build relationships with their customers over time through a variety of transactions (e.g., product inquiries, sales, service requests, warranty calls), they accumulate tremendous amounts of data. When combined with demographic and socioeconomic attributes, this information-rich data can be used to (1) identify most likely responders/buyers of new products/services (i.e., customer profiling); (2) understand the roots causes of customer attrition in order to improve customer retention (i.e., churn analysis); (3) discover time-variant associations between products and services to maximize sales and customer value; and (4) identify the most profitable customers and their preferential needs to strengthen relationships and to maximize sales.
- ***Banking.*** Data mining can help banks with the following: (1) automating the loan application process by accurately predicting the most probable defaulters; (2) detecting fraudulent credit card and online-banking transactions; (3) identifying ways to maximize customer value by selling them products and services that they are most likely to buy; and (4) optimizing the cash return by accurately forecasting the cash flow on banking entities (e.g., ATM machines, banking branches).
- ***Retailing and logistics.*** In the retailing industry, data mining can be used to (1) predict accurate sales volumes at specific retail locations in order to determine correct inventory levels; (2) identify sales relationships between different products (with market-basket analysis) to improve the store layout and optimize sales promotions; (3) forecast consumption levels of different product types (based on seasonal and environmental conditions) to optimize logistics and hence maximize sales; and (4) discover interesting patterns in the movement of products (especially for the

products that have a limited shelf life because they are prone to expiration, perishability, and contamination) in a supply chain by analyzing sensory and RFID data.

- ***Manufacturing and production.*** Manufacturers can use data mining to (1) predict machinery failures before they occur through the use of sensory data (enabling what is called *condition-based maintenance*); (2) identify anomalies and commonalities in production systems to optimize manufacturing capacity; and (3) discover novel patterns to identify and improve product quality.
- ***Brokerage and securities trading.*** Brokers and traders use data mining to (1) predict when and how much certain bond prices will change; (2) forecast the range and direction of stock fluctuations; (3) assess the effect of particular issues and events on overall market movements; and (4) identify and prevent fraudulent activities in securities trading.
- ***Insurance.*** The insurance industry uses data mining techniques to (1) forecast claim amounts for property and medical coverage costs for better business planning; (2) determine optimal rate plans based on the analysis of claims and customer data; (3) predict which customers are more likely to buy new policies with special features; and (4) identify and prevent incorrect claim payments and fraudulent activities.
- ***Computer hardware and software.*** Data mining can be used to (1) predict disk drive failures well before they actually occur; (2) identify and filter unwanted Web content and e-mail messages; (3) detect and prevent computer network security bridges; and (4) identify potentially unsecure software products.
- ***Government and defense.*** Data mining also has a number of military applications. It can be used to (1) forecast the cost of moving military personnel and equipment; (2) predict an adversary's moves and hence develop more successful strategies for military engagements; (3) predict resource consumption for better planning and budgeting; and (4) identify classes of unique experiences, strategies, and lessons learned from military operations for better knowledge sharing throughout the organization.
- ***Travel industry (airlines, hotels/resorts, rental car companies).*** Data mining has a variety of uses in the travel industry. It is successfully used to (1) predict sales of different services (seat types in airplanes, room types in hotels/resorts, car types in rental car companies) in order to optimally price services to maximize revenues as a function of time-varying transactions (commonly referred to as *yield management*); (2) forecast demand at different locations to better allocate limited organizational resources; (3) identify the most profitable customers and provide them with personalized services to maintain their repeat business; and (4) retain valuable employees by identifying and acting on the root causes for attrition.
- ***Health care.*** Data mining has a number of health care applications. It can be used to (1) identify people without health insurance and the factors underlying this undesired phenomenon; (2) identify novel cost-benefit relationships between different treatments to develop more effective strategies; (3) forecast the level and the time of demand at different service locations to optimally allocate organizational resources; and (4) understand the underlying reasons for customer and employee attrition.
- ***Medicine.*** Use of data mining in medicine should be viewed as an invaluable complement to traditional medical research, which is mainly clinical and biological in nature. Data mining analyses can (1) identify novel patterns to improve survivability of patients with cancer; (2) predict success rates of organ transplantation patients to develop better donor-organ matching policies; (3) identify the functions of different genes in the human chromosome (known as genomics); and (4) discover the relationships between symptoms and illnesses (as well as illnesses and successful treatments) to help medical professionals make informed and correct decisions in a timely manner.

- ***Entertainment industry.*** Data mining is successfully used by the entertainment industry to (1) analyze viewer data to decide what programs to show during prime time and how to maximize returns by knowing where to insert advertisements; (2) predict the financial success of movies before they are produced to make investment decisions and to optimize the returns; (3) forecast the demand at different locations and different times to better schedule entertainment events and to optimally allocate resources; and (4) develop optimal pricing policies to maximize revenues.
- ***Homeland security and law enforcement.*** Data mining has a number of homeland security and law enforcement applications. Data mining is often used to (1) identify patterns of terrorist behaviors (see Application Case 5.4 for a recent example of use of data mining to track funding of terrorists' activities); (2) discover crime patterns (e.g., locations, timings, criminal behaviors, and other related attributes) to help solve criminal cases in a timely manner; (3) predict and eliminate potential biological and chemical attacks to the nation's critical infrastructure by analyzing special-purpose sensory data; and (4) identify and stop malicious attacks on critical information infrastructures (often called *information warfare*).
- ***Sports.*** Data mining was used to improve the performance of National Basketball Association (NBA) teams in the United States. The NBA developed Advanced Scout, a PC-based data mining application that coaching staff use to discover interesting patterns in basketball game data. The pattern interpretation is facilitated by allowing the user to relate patterns to videotape. See Bhandari et al. (1997) for details.

APPLICATION CASE 5.4

A Mine on Terrorist Funding

The terrorist attack on the World Trade Center on September 11, 2001, underlined the importance of open source intelligence. The USA PATRIOT Act and the creation of the U.S. Department of Homeland Security (DHS) heralded the potential application of information technology and data mining techniques to detect money laundering and other forms of terrorist financing. Law enforcement agencies have been focusing on money laundering activities via normal transactions through banks and other financial service organizations.

Law enforcement agencies are now focusing on international trade pricing as a terrorism funding tool. International trade has been used by money launderers to move money silently out of a country without attracting government attention. This transfer is achieved by overvaluing imports and undervaluing exports. For example, a domestic importer and foreign exporter could form a partnership and overvalue imports, thereby transferring money from the home country, resulting in crimes related to customs fraud, income tax evasion, and money laundering. The foreign exporter could be a member of a terrorist organization.

Data mining techniques focus on analysis of data on import and export transactions from the U.S. Department of Commerce and commerce-related entities. Import prices that exceed the upper quartile import prices and export prices that are lower than the lower quartile export prices are tracked. The focus is on abnormal transfer prices between corporations that may result in shifting taxable income and taxes out of the United States. An observed price deviation may be related to income tax avoidance/evasion, money laundering, or terrorist financing. The observed price deviation may also be due to an error in the U.S. trade database.

Data mining will result in efficient evaluation of data, which, in turn, will aid in the fight against terrorism. The application of information technology and data mining techniques to financial transactions can contribute to better intelligence information.

Sources: J. S. Zdanowic, "Detecting Money Laundering and Terrorist Financing via Data Mining," *Communications of the ACM,* Vol. 47, No. 5, May 2004, p. 53; and R. J. Bolton, "Statistical Fraud Detection: A Review," *Statistical Science,* Vol. 17, No. 3, January 2002, p. 235.

Section 5.3 Review Questions

1. What are the major application areas for data mining?
2. Identify at least five specific applications of data mining and list five common characteristics of these applications.
3. What do you think is the most prominent application area for data mining? Why?
4. Can you think of other application areas for data mining not discussed in this section? Explain.

5.4 DATA MINING PROCESS

In order to systematically carry out data mining projects, a general process is usually followed. Based on best practices, data mining researchers and practitioners have proposed several processes (workflows or simple step-by-step approaches) to maximize the chances of success in conducting data mining projects. These efforts have led to several standardized processes, some of which (a few of the most popular ones) are described in this section.

One such standardized process, arguably the most popular one, Cross-Industry Standard Process for Data Mining—**CRISP-DM**—was proposed in the mid-1990s by a European consortium of companies to serve as a nonproprietary standard methodology for data mining (CRISP-DM, 2009). Figure 5.5 illustrates this proposed process, which is a sequence of six steps that starts with a good understanding of the business and the need for the data mining project (i.e., the application domain) and ends with the deployment of the solution that satisfied the specific business need. Even though these steps are sequential in nature, there is usually a great deal of backtracking. Because the data mining is driven by experience and experimentation, depending on the problem situation and the knowledge/experience of the analyst, the whole process can be very iterative

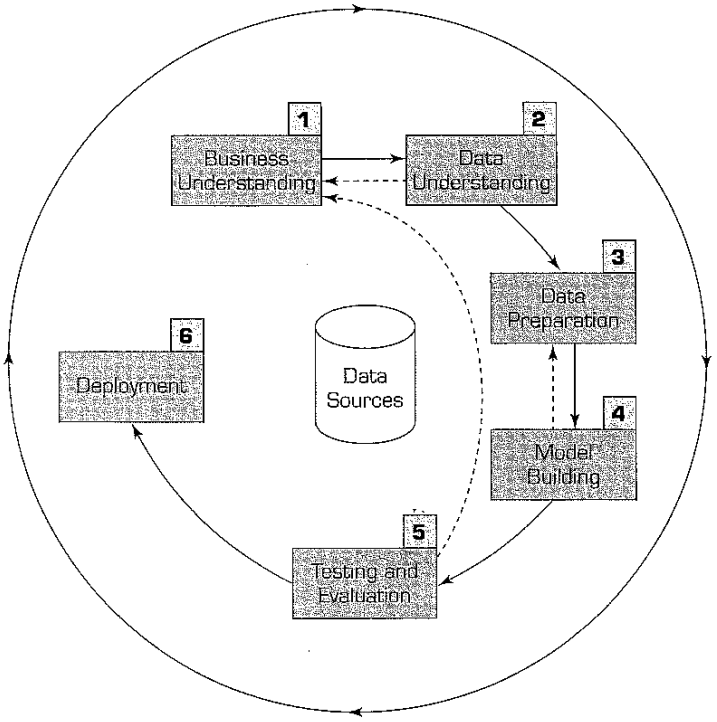


FIGURE 5.5 The Six-Step CRISP-DM Data Mining Process *Source:* Adapted from CRISP-DM.org.

(i.e., one should expect to go back and forth through the steps quite a few times) and time consuming. Because latter steps are built on the outcome of the former ones, one should pay extra attention to the earlier steps in order not to put the whole study on an incorrect path from the onset.

Step 1: Business Understanding

The key element of any data mining study is to know what the study is for. Answering such a question begins with a thorough understanding of the managerial need for new knowledge and an explicit specification of the business objective regarding the study to be conducted. Specific goals such as "What are the common characteristics of the customers we have lost to our competitors recently?" or "What are typical profiles of our customers, and how much value does each of them provide to us?" are needed. Then a project plan for finding such knowledge is developed that specifies the people responsible for collecting the data, analyzing the data, and reporting the findings. At this early stage, a budget to support the study should also be established, at least at a high level with rough numbers.

Step 2: Data Understanding

A data mining study is specific to addressing a well-defined business task, and different business tasks require different sets of data. Following the business understanding, the main activity of the data mining process is to identify the relevant data from many available databases. Some key points must be considered in the data identification and selection phase. First and foremost, the analyst should be clear and concise about the description of the data mining task so that the most relevant data can be identified. For example, a retail data mining project may seek to identify spending behaviors of female shoppers who purchase seasonal clothes based on their demographics, credit card transactions, and socioeconomic attributes. Furthermore, the analyst should build an intimate understanding of the data sources (e.g., where the relevant data are stored and in what form; what the process of collecting the data is—automated versus manual; who the collectors of the data are and how often the data are updated) and the variables (e.g., What are the most relevant variables? Are there any synonymous and/or homonymous variables? Are the variables independent of each other—do they stand as a complete information source without overlapping or conflicting information?).

In order to better understand the data, the analyst often uses a variety of statistical and graphical techniques, such as simple statistical summaries of each variable (e.g., for numeric variables the average, minimum/maximum, median, standard deviation are among the calculated measures, whereas for categorical variables the mode and frequency tables are calculated), correlation analysis, scatterplots, histograms, and box plots. A careful identification and selection of data sources and the most relevant variables can make it easier for data mining algorithms to quickly discover useful knowledge patterns.

Data sources for data selection can vary. Normally, data sources for business applications include demographic data (such as income, education, number of households, and age), sociographic data (such as hobby, club membership, and entertainment), transactional data (sales record, credit card spending, issued checks), and so on.

Data can be categorized as quantitative and qualitative. Quantitative data is measured using numeric values. It can be discrete (such as integers) or continuous (such as real numbers). Qualitative data, also known as categorical data, contains both nominal and ordinal data. Nominal data has finite nonordered values (e.g., gender data, which has two values: male and female). Ordinal data has finite ordered values. For example, customer credit ratings are considered ordinal data because the ratings can be excellent, fair, and bad.

Quantitative data can be readily represented by some sort of probability distribution. A probability distribution describes how the data is dispersed and shaped. For instance, normally distributed data is symmetric and is commonly referred to as being a bell-shaped curve. Qualitative data may be coded to numbers and then described by frequency distributions. Once the relevant data are selected according to the data mining business objective, data preprocessing should be pursued.

Step 3: Data Preparation

The purpose of data preparation (or more commonly called *data preprocessing*) is to take the data indentified in the previous step and prepare it for analysis by data mining methods. Compared to the other steps in CRISP-DM, data preprocessing consumes the most time and effort; most believe that this step accounts for roughly 80 percent of the total time spent on a data mining project. The reason for such an enormous effort spent on this step is the fact that real-world data is generally incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), noisy (containing errors or outliers), and inconsistent (containing discrepancies in codes or names). Figure 5.6 shows the four main steps needed to convert the raw real-world data into minable datasets.

In the first phase of data preprocessing, the relevant data is collected from the identified sources (accomplished in the previous step—Data Understanding—of the CRISP-DM process), the necessary records and variables are selected (based on an intimate understanding of the data the unnecessary sections are filtered out), and the records coming from multiple data sources are integrated (again, using the intimate understanding of the data the synonyms and homonyms are to be handled properly).

In the second phase of data preprocessing, the data is cleaned (this step is also known as data scrubbing). In this step, the values in the dataset are identified and dealt

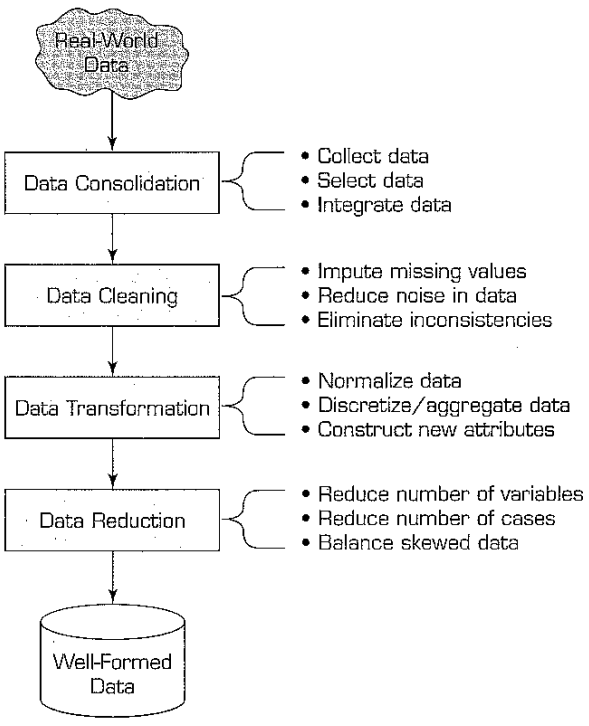


FIGURE 5.6 Data Preprocessing Steps

with. In some cases, missing values are an anomaly in the dataset, in which case they need to be imputed (filled with a most probable value) or ignored; in other cases, the missing values are a natural part of the dataset (e.g., the *household income* field is often left unanswered by people who are in the top income tier). In this step, the analyst should also identify noisy values in the data (i.e., the outliers) and smooth them out. Additionally, inconsistencies (unusual values within a variable) in the data should be handled using domain knowledge and/or expert opinion.

In the third phase of data preprocessing, the data is transformed for better processing. For instance, in many cases the data is normalized between a certain minimum and maximum for all variables in order to mitigate the potential bias of one variable (having large numeric values, such as for household income) dominating other variables (such as *number of dependents* or *years in service*, which may potentially be more important) having smaller values. Another transformation that takes place is discritization and/or aggregation. In some cases, the numeric variables are converted to categorical values (e.g., low, medium, high); in other cases a nominal variable's unique value range is reduced to a smaller set using concept hierarchies (e.g., as opposed to using the individual states with 50 different values, one may chose to use several regions for a variable that shows location) in order to have a dataset that is more amenable to computer processing. Still, in other cases one might choose to create new variables based on the existing ones in order to magnify the information found in a collection of variables in the dataset. For instance, in an organ transplantation dataset one might choose to use a single variable showing the blood-type match (1: match, 0: no-match) as opposed to separate multinominal values for the blood type of both the donor and the recipient. Such simplification may increase the information content while reducing the complexity of the relationships in the data.

The final phase of data preprocessing is data reduction. Even though data miners like to have large datasets, too much data is also a problem. In the simplest sense, one can visualize the data commonly used in data mining projects as a flat file consisting of two dimensions: variables (the number of columns) and cases/records (the number of rows). In some cases (e.g., image processing and genome projects with complex microarray data), the number of variables can be rather large, and the analyst must reduce the number down to a manageable size. Because the variables are treated as different dimensions that describe the phenomenon from different perspectives, in data mining this process is commonly called *dimensional reduction*. Even though there is not a single best way to accomplish this task, one can use the findings from previously published literature; consult domain experts; run appropriate statistical tests (e.g., principle component analysis or independent component analysis); and, more preferably, use a combination of these techniques to successfully reduce the dimensions in the data into a more manageable and most relevant subset.

With respect to the other dimension (i.e., the number of cases), some datasets may include millions or billions of records. Even though computing power is increasing exponentially, processing such a large number of records may not be practical or feasible. In such cases, one may need to sample a subset of the data for analysis. The underlying assumption of sampling is that the subset of the data will contain all relevant patterns of the complete dataset. In a homogenous dataset, such an assumption may hold well, but real-world data is hardly ever homogenous. The analyst should be extremely careful in selecting a subset of the data that reflects the essence of the complete dataset and is not specific to a subgroup or subcategory. The data is usually sorted on some variable, and taking a section of the data from the top or bottom may lead to a biased dataset on specific values of the indexed variable; therefore, one should always try to randomly select the records on the sample set. For skewed data, straightforward random sampling may not be sufficient, and stratified sampling (a proportional representation of different subgroups in the data is represented in the sample dataset) may be required. Speaking of skewed data; it is a good practice to balance the highly skewed data by either

oversampling the less represented or under sampling the more represented classes. Research has shown that balanced datasets tends to produce better prediction models than unbalanced ones (Wilson and Sharda, 1994).

The essence of data preprocessing is summarized in Table 5.4, which maps the main phases (along with their problem descriptions) to a representative list of tasks and algorithms.

TABLE 5.4 A Summary of Data Preprocessing Tasks and Potential Methods

Main Task	Subtasks	Popular Methods
Data consolidation	Access and collect the data	SQL queries, software agents, Web services.
	Select and filter the data	Domain expertise, SQL queries, statistical tests.
	Integrate and unify the data	SQL queries, domain expertise, ontology-driven data mapping.
Data cleaning	Handle missing values in the data	Fill-in missing values (imputations) with most appropriate values (mean, median, min/max, mode, etc.); recode the missing values with a constant such as "ML"; remove the record of the missing value; do nothing.
	Identify and reduce noise in the data	Identify the outliers in data with simple statistical techniques (such as averages and standard deviations) or with cluster analysis; once identified either remove the outliers or smooth them by using binning, regression, or simple averages.
	Find and eliminate erroneous data	Identify the erroneous values in data (other than outliers), such as odd values, inconsistent class labels, odd distributions; once identified, use domain expertise to correct the values or remove the records holding the erroneous values.
Data transformation	Normalize the data	Reduce the range of values in each numerically valued variable to a standard range (e.g., 0 to 1 or −1 to +1) by using a variety of normalization or scaling techniques.
	Discretize or aggregate the data	If needed, convert the numeric variables into discrete representations using range or frequency-based binning techniques; for categorical variables reduce the number of values by applying proper concept hierarchies.
	Construct new attributes	Derive new and more informative variables from the existing ones using a wide range of mathematical functions (as simple as addition and multiplication or as complex as a hybrid combination of log transformations).
Data reduction	Reduce number of attributes	Principle component analysis, independent component analysis, Chi-square testing, correlation analysis, and decision tree induction.
	Reduce number of records	Random sampling, stratified sampling, expert–knowledge-driven purposeful sampling.
	Balance skewed data	Oversample the less represented or under sample the more represented classes.

Step 4: Model Building

In this step, various modeling techniques are selected and applied to an already prepared dataset in order to address the specific business need. The model-building step also encompasses the assessment and comparative analysis of the various models built. Because there is not a universally known *best* method or algorithm for a data mining task, one should use a variety of viable model types along with a well-defined experimentation and assessment strategy to identify the "best" method for a given purpose. Even for a single method or algorithm, a number of parameters need to be calibrated to obtain optimal results. Some methods may have specific requirements on the way that the data is to be formatted; thus stepping back to the data preparation step is often necessary.

Depending on the business need, the data mining task can be of a prediction (either classification or regression), an association, or a clustering type. Each of these data mining tasks can use a variety of data mining methods and algorithms. Some of these data mining methods were explained earlier in this chapter, and some of the most popular algorithms, including decision trees for classification, *k*-means for clustering, and the Apriori algorithm for association rule mining, are described later in this chapter.

Step 5: Testing and Evaluation

In step 5, the developed models are assessed and evaluated for their accuracy and generality. This step assesses the degree to which the selected model (or models) meets the business objectives and, if so, to what extent (i.e., do more models need to be developed and assessed). Another option is to test the developed model(s) in a real-world scenario if time and budget constraints permit. Even though the outcome of the developed models is expected to relate to the original business objectives, other findings that are not necessarily related to the original business objectives but that might also unveil additional information or hints for future directions often are discovered.

The testing and evaluation step is a critical and challenging task. No value is added by the data mining task until the business value obtained from discovered knowledge patterns is identified and recognized. Determining the business value from discovered knowledge patterns is somewhat similar to playing with puzzles. The extracted knowledge patterns are pieces of the puzzle that need to be put together in the context of the specific business purpose. The success of this identification operation depends on the interaction among data analysts, business analysts, and decision makers (such as business managers). Because data analysts may not have the full understanding of the data mining objectives and what they mean to the business and the business analysts and decision makers may not have the technical knowledge to interpret the results of sophisticated mathematical solutions, interaction among them is necessary. In order to properly interpret knowledge patterns, it is often necessary to use a variety of tabulation and visualization techniques (e.g., pivot tables, cross tabulation of findings, pie charts, histograms, box plots, scatterplots).

Step 6: Deployment

Development and assessment of the models is not the end of the data mining project. Even if the purpose of the model is to have a simple exploration of the data, the knowledge gained from such exploration will need to be organized and presented in a way that the end user can understand and benefit from. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise. In many cases, it is the customer, not

the data analyst, who carries out the deployment steps. However, even if the analyst will not carry out the deployment effort, it is important for the customer to understand up front what actions need to be carried out in order to actually make use of the created models.

The deployment step may also include maintenance activities for the deployed models. Because everything about the business is constantly changing, the data that reflect the business activities also are changing. Over time, the models (and the patterns embedded within them) built on the old data may become obsolete, irrelevant, or misleading. Therefore, monitoring and maintenance of the models are important if the data mining results are to become a part of the day-to-day business and its environment. A careful preparation of a maintenance strategy helps to avoid unnecessarily long periods of incorrect usage of data mining results. In order to monitor the deployment of the data mining result(s), the project needs a detailed plan on the monitoring process, which may not be a trivial task for complex data mining models.

APPLICATION CASE 5.5

Data Mining in Cancer Research

According to the American Cancer Society, approximately 1.5 million new cancer cases will be diagnosed in 2009. Cancer is the second most common cause of death in the United States and in the world, exceeded only by cardiovascular disease. This year, 562,340 Americans are expected to die of cancer—more than 1,500 people a day—accounting for nearly 1 of every 4 deaths.

Cancer is a group of diseases generally characterized by uncontrolled growth and spread of abnormal cells. If the growth and/or spread is not controlled, it can result in death. Even though the exact reasons are not known, cancer is believed to be caused by both external factors (e.g., tobacco, infectious organisms, chemicals, and radiation) and internal factors (e.g., inherited mutations, hormones, immune conditions, and mutations that occur from metabolism). These causal factors may act together or in sequence to initiate or promote carcinogenesis. Cancer is treated with surgery, radiation, chemotherapy, hormone therapy, biological therapy, and targeted therapy. Survival statistics vary greatly by cancer type and stage at diagnosis.

The 5-year relative survival rate for all cancers diagnosed in 1996–2004 is 66 percent, up from 50 percent in 1975–1977. The improvement in survival reflects progress in diagnosing certain cancers at an earlier stage and improvements in treatment. Further improvements are needed to prevent and treat cancer.

Even though cancer research has traditionally been clinical and biological in nature, in recent years data-driven analytic studies have become a common complement. In medical domains where data- and analytics-driven research have been applied successfully, novel research directions have been identified to further advance the clinical and biological studies. Using various types of data, including molecular, clinical, literature-based, and clinical-trial data, along with suitable data mining tools and techniques, researchers have been able to identify novel patterns, paving the road toward a cancer-free society.

In one study, Delen (2009) used three popular data mining techniques (decision trees, artificial neural networks, and support vector machines) in conjunction with logistic regression to develop prediction models for prostate cancer survivability. The dataset contained around 120,000 records and 77 variables. A *k*-fold cross-validation methodology was used in model building, evaluation, and comparison. The results showed that support vector models are the most accurate predictor (with a test set accuracy of 92.85%) for this domain, followed by artificial neural networks and decision trees. Furthermore, using a sensitivity–analysis-based evaluation method, the study also revealed novel patterns related to prognostic factors of prostate cancer.

In a related study, Delen et al. (2006) used two data mining algorithms (artificial neural networks and decision trees) and logistic regression to develop prediction models for breast cancer survival using a large dataset (more than 200,000 cases). Using a *10*-fold cross-validation method to

measure the unbiased estimate of the prediction models for performance comparison purposes, the results indicated that the decision tree (C5 algorithm) was the best predictor, with 93.6 percent accuracy on the holdout sample (which was the best prediction accuracy reported in the literature); followed by artificial neural networks, with 91.2 percent accuracy; and logistic regression, with 89.2 percent accuracy. Further analysis of prediction models revealed prioritized importance of the prognostic factors, which can then be used as basis for further clinical and biological research studies.

These examples (among many others in the medical literature) show that advanced data mining techniques can be used to develop models that possess a high degree of predictive as well as explanatory power. Although data mining methods are capable of extracting patterns and relationships hidden deep in large and complex medical databases, without the cooperation and feedback from the medical experts their results are not of much use. The patterns found via data mining methods should be evaluated by medical professionals who have years of experience in the problem domain to decide whether they are logical, actionable, and novel to warrant new research directions. In short, data mining is not to replace medical professionals and researchers, but to complement their invaluable efforts to provide data-driven new research directions and to ultimately save more human lives.

Sources: D. Delen, "Analysis of Cancer Data: A Data Mining Approach," *Expert Systems*, Vol. 26, No. 1, 2009, pp. 100–112; J. Thongkam, G. Xu, Y. Zhang, and F. Huang, "Toward Breast Cancer Survivability Prediction Models Through Improving Training Space," *Expert Systems with Applications*, 2009, *in press*; D. Delen, G. Walker, and A. Kadam, "Predicting Breast Cancer Survivability: A Comparison of Three Data Mining Methods," *Artificial Intelligence in Medicine*, Vol. 34, No. 2, 2005, pp. 113–127.

Other Data Mining Standardized Processes and Methodologies

In order to be applied successfully, a data mining study must be viewed as a process that follows a standardized methodology rather than as a set of automated software tools and techniques. In addition to CRISP-DM, there is another well-known methodology developed by the SAS Institute, called SEMMA (2009). The acronym **SEMMA** stands for "sample, explore, modify, model, and assess."

Beginning with a statistically representative sample of the data, SEMMA makes it easy to apply exploratory statistical and visualization techniques, select and transform the most significant predictive variables, model the variables to predict outcomes, and confirm a model's accuracy. A pictorial representation of SEMMA is given in Figure 5.7.

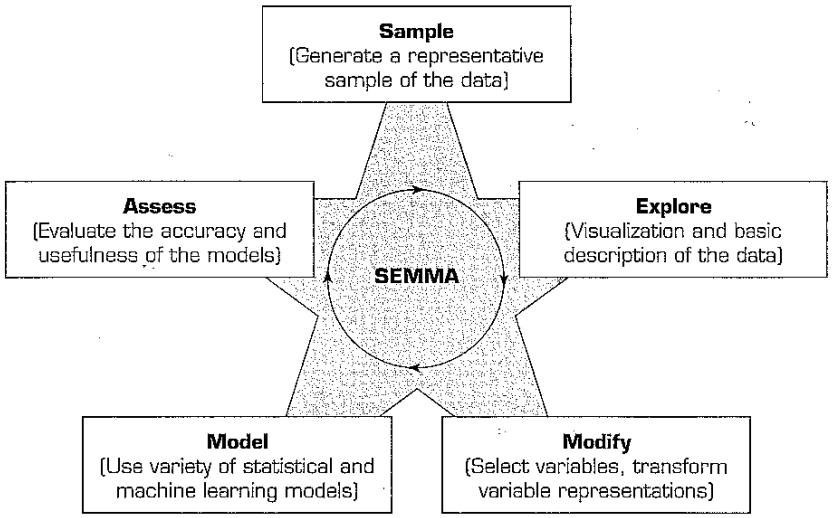


FIGURE 5.7 SEMMA Data Mining Process

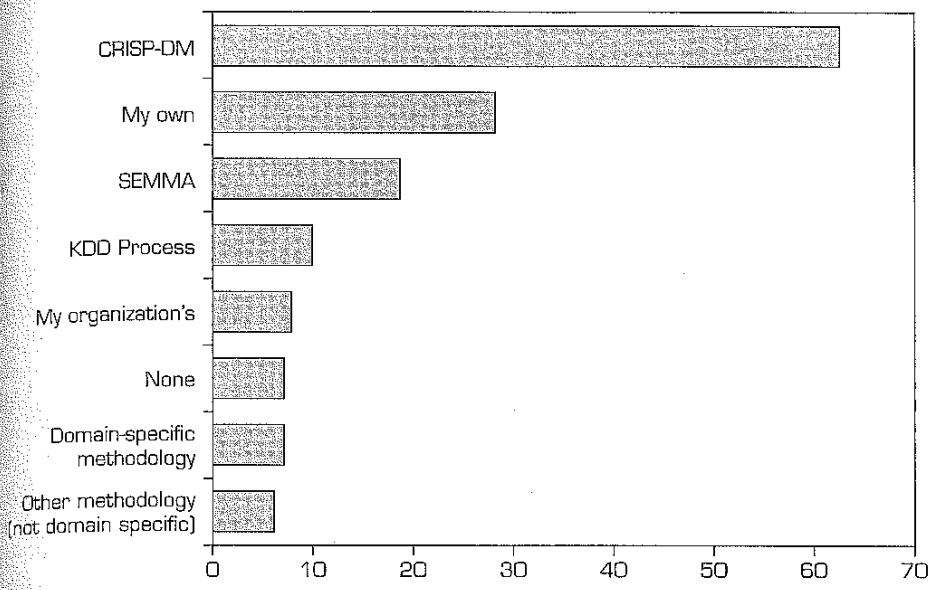


FIGURE 5.8 Ranking of Data Mining Methodologies/Processes *Source:* Used with permission from **kdnuggets.com.**

By assessing the outcome of each stage in the SEMMA process, the model developer can determine how to model new questions raised by the previous results, and thus proceed back to the exploration phase for additional refinement of the data; that is, as with CRISP-DM, SEMMA is driven by a highly iterative experimentation cycle. The main difference between CRISP-DM and SEMMA is that CRISP-DM takes a more comprehensive approach—including understanding of the business and the relevant data—to data mining projects, whereas SEMMA implicitly assumes that the data mining project's goals and objectives along with the appropriate data sources have been identified and understood.

Some practitioners commonly use the term **knowledge discovery in databases (KDD)** as a synonym for data mining. Fayyad et al. (1996) defined *knowledge discovery in databases* as a process of using data mining methods to find useful information and patterns in the data, as opposed to data mining, which involves using algorithms to identify patterns in data derived through the KDD process. KDD is a comprehensive process that encompasses data mining. The input to the KDD process consists of organizational data. The enterprise data warehouse enables KDD to be implemented efficiently because it provides a single source for data to be mined. Dunham (2003) summarized the KDD process as consisting of the following steps: data selection, data preprocessing, data transformation, data mining, and interpretation/evaluation. Figure 5.8 shows the polling results for the question of "What main methodology are you using for data mining?" (conducted by **kdnuggets.com** in August 2007).

Section 5.4 Review Questions

1. What are the major data mining processes?
2. Why do you think the early phases (understanding of the business and understanding of the data) take the longest in data mining projects?
3. List and briefly define the phases in the CRISP-DM process.
4. What are the main data preprocessing steps? Briefly describe each step and provide relevant examples.
5. How does CRISP-DM differ from SEMMA?

5.5 DATA MINING METHODS

A variety of methods are available for performing data mining studies, including classification, regression, clustering, and association. Most data mining software tools employ more than one technique (or algorithm) for each of these methods. This section describes the most popular data mining methods and explains their representative techniques.

Classification

Classification is perhaps the most frequently used data mining method for real-world problems. As a popular member of the machine-learning family of techniques, classification learns patterns from past data (a set of information—traits, variables, features—on characteristics of the previously labeled items, objects, or events) in order to place new instances (with unknown labels) into their respective groups or classes. For example, one could use classification to predict whether the weather on a particular day will be "sunny," "rainy," or "cloudy." Popular classification tasks include credit approval (i.e., good or bad credit risk), store location (e.g., good, moderate, bad), target marketing (e.g., likely customer, no hope), fraud detection (i.e., yes, no), and telecommunication (e.g., likely to turn to another phone company, yes/no). If what is being predicted is a class label (e.g., "sunny," "rainy," or "cloudy") the prediction problem is called a classification, whereas if it is a numeric value (e.g., temperature such as 68°F), the prediction problem is called a **regression**.

Even though clustering (another popular data mining method) can also be used to determine groups (or class memberships) of things, there is a significant difference between the two. Classification learns the function between the characteristics of things (i.e., independent variables) and their membership (i.e., output variable) through a supervised learning process where both types (input and output) of variables are presented to the algorithm; in clustering, the membership of the objects is learned through an unsupervised learning process where only the input variables are presented to the algorithm. Unlike classification, clustering does not have a supervising (or controlling) mechanism that enforces the learning process; instead, clustering algorithms use one or more heuristics (e.g., multidimensional distance measure) to discover natural groupings of objects.

The most common two-step methodology of classification-type prediction involves model development/training and model testing/deployment. In the model development phase, a collection of input data, including the actual class labels, is used. After a model has been trained, the model is tested against the holdout sample for accuracy assessment and eventually deployed for actual use where it is to predict classes of new data instances (where the class label is unknown). Several factors are considered in assessing the model, including the following:

- ***Predictive accuracy.*** The model's ability to correctly predict the class label of new or previously unseen data. Prediction accuracy is the most commonly used assessment factor for classification models. To compute this measure, actual class labels of a test dataset are matched against the class labels predicted by the model. The accuracy can then be computed as the *accuracy rate,* which is the percentage of test dataset samples correctly classified by the model (more on this topic is provided later in the chapter).
- ***Speed.*** The computational costs involved in generating and using the model, where faster is deemed to be better.
- ***Robustness.*** The model's ability to make reasonably accurate predictions, given noisy data or data with missing and erroneous values.

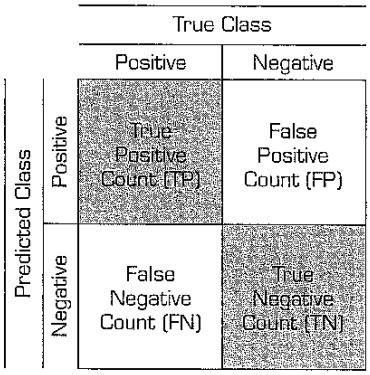


FIGURE 5.9 A Simple Confusion Matrix for Tabulation of Two-Class Classification Results

- ***Scalability.*** The ability to construct a prediction model efficiently given a rather large amount of data.
- ***Interpretability.*** The level of understanding and insight provided by the model (e.g., how and/or what the model concludes on certain predictions).

Estimating the True Accuracy of Classification Models

In classification problems, the primary source for accuracy estimation is the *confusion matrix* (also called a *classification matrix* or a *contingency table*). Figure 5.9 shows a confusion matrix for a two-class classification problem. The numbers along the diagonal from the upper left to the lower right represent correct decisions, and the numbers outside this diagonal represent the errors.

Table 5.5 provides equations for common accuracy metrics for classification models.

When the classification problem is not binary, the confusion matrix gets bigger (a square matrix with the size of the unique number of class labels), and accuracy metrics become limited to *per class accuracy rates* and the *overall classifier accuracy*.

$$(True\ Classification\ Rate)_i = \frac{(True\ Classification)_i}{\sum_{i=1}^{n}(False\ Classification)_i}$$

TABLE 5.5 Common Accuracy Metrics for Classification Models

Metric	Description
$True\ Positive\ Rate = \frac{TP}{TP + FN}$	The ratio of correctly classified positives divided by the total positive count (i.e., hit rate or recall)
$True\ Negative\ Rate = \frac{TN}{TN + FP}$	The ratio of correctly classified negatives divided by the total negative count (i.e., false alarm rate)
$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$	The ratio of correctly classified instances (positives and negatives) divided by the total number of instances
$Precision = \frac{TP}{TP + FP}$	The ratio of correctly classified positives divided by the sum of correctly classified positives and incorrectly classified positives
$Recall = \frac{TP}{TP + FN}$	Ratio of correctly classified positives divided by the sum of correctly classified positives and incorrectly classified negatives

$$(Overall\ Classifier\ Accuracy)_i = \frac{\sum_{i=1}^{n}(True\ Classification)_i}{Total\ Number\ of\ Cases}$$

Estimating the accuracy of a classification model (or classifier) induced by a supervised learning algorithm is important for the following two reasons: First, it can be used to estimate its future prediction accuracy, which could imply the level of confidence one should have in the classifier's output in the prediction system. Second, it can be used for choosing a classifier from a given set (identifying the "best" classification model among the many trained). The following are among the most popular estimation methodologies used for classification-type data mining models.

SIMPLE SPLIT The **simple split** (or holdout or test sample estimation) partitions the data into two mutually exclusive subsets called a *training set* and a *test set* (or *holdout set*). It is common to designate two-thirds of the data as the training set and the remaining one-third as the test set. The training set is used by the inducer (model builder), and the built classifier is then tested on the test set. An exception to this rule occurs when the classifier is an artificial neural network. In this case, the data is partitioned into three mutually exclusive subsets: training, validation, and testing. The validation set is used during model building to prevent overfitting (more on artificial neural networks can be found in Chapter 6). Figure 5.10 shows the simple split methodology.

The main criticism of this method is that it makes the assumption that the data in the two subsets are of the same kind (i.e., have the exact same properties). Because this is a simple random partitioning, in most realistic datasets where the data are skewed on the classification variable, such an assumption may not hold true. In order to improve this situation, stratified sampling is suggested, where the strata become the output variable. Even though this is an improvement over the simple split, it still has a bias associated from the single random partitioning.

***K*-FOLD CROSS-VALIDATION** In order to minimize the bias associated with the random sampling of the training and holdout data samples in comparing the predictive accuracy of two or more methods, one can use a methodology called ***k*-fold cross-validation**. In *k*-fold cross-validation, also called *rotation estimation,* the complete dataset is randomly split into *k* mutually exclusive subsets of approximately equal size. The classification model is trained and tested *k* times. Each time it is trained on all but one fold and then tested on the remaining single fold. The cross-validation estimate of the overall accuracy

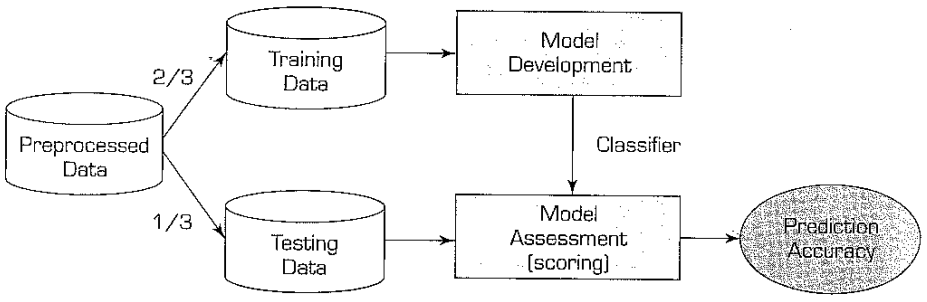


FIGURE 5.10 Simple Random Data Splitting

of a model is calculated by simply averaging the k individual accuracy measures, as shown in the following equation:

$$CVA = \frac{1}{k}\sum_{i=1}^{k} A_i$$

where CVA stands for cross-validation accuracy, k is the number of folds used, and A is the accuracy measure (e.g., hit-rate, sensitivity, specificity) of each fold.

ADDITIONAL CLASSIFICATION ASSESSMENT METHODOLOGIES Other popular assessment methodologies include the following:

- ***Leave-one-out.*** The leave-one-out method is similar to the k-fold cross-validation where the k takes the value of 1; that is, every data point is used for testing once on as many models developed as there are number of data points. This is a time-consuming methodology, but sometimes for small datasets it is a viable option.
- ***Bootstrapping.*** With **bootstrapping,** a fixed number of instances from the original data is sampled (with replacement) for training and the rest of the dataset is used for testing. This process is repeated as many times as desired.
- ***Jackknifing.*** Similar to the leave-one-out methodology; with jackknifing the accuracy is calculated by leaving one sample out at each iteration of the estimation process.
- ***Area under the ROC curve.*** The **area under the ROC curve** is a graphical assessment technique where the true positive rate is plotted on the Y-axis and false positive rate is plotted on the X-axis. The area under the ROC curve determines the accuracy measure of a classifier: A value of 1 indicates a perfect classifier whereas 0.5 indicates no better than random chance; in reality, the values would range between the two extreme cases. For example, in Figure 5.11 A has a better classification performance than B, while C is not any better than random chance of flipping a coin.

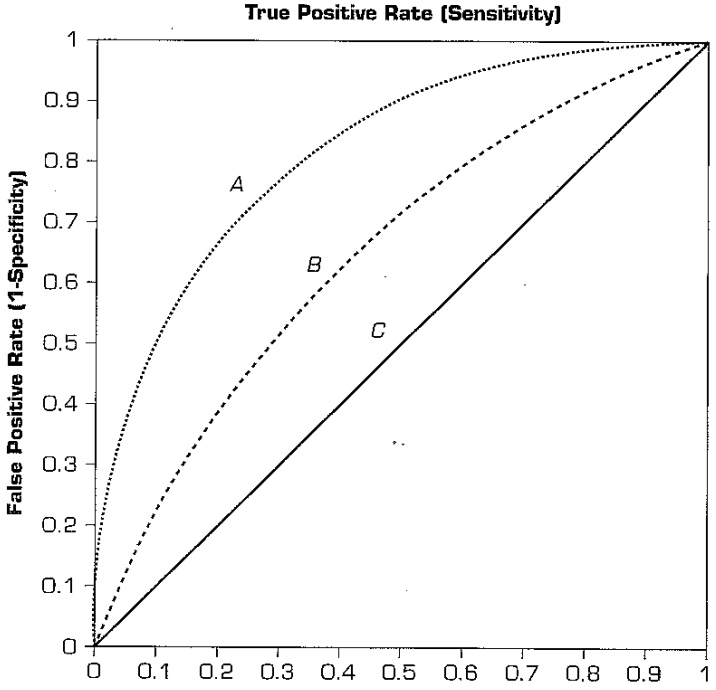


FIGURE 5.11 Sample ROC Curve

CLASSIFICATION TECHNIQUES A number of techniques (or algorithms) are used for classification modeling, including the following:

- ***Decision tree analysis.*** Decision tree analysis (a machine-learning technique) is arguably the most popular classification technique in the data mining arena. A detailed description of this technique is given in the following section.
- ***Statistical analysis.*** Statistical techniques were the primary classification algorithm for many years until the emergence of machine-learning techniques. Statistical classification techniques include logistic regression and discriminant analysis, both of which make the assumptions that the relationships between the input and output variables are linear in nature, the data is normally distributed, and the variables are not correlated and are independent of each other. The questionable nature of these assumptions has led to the shift toward machine-learning techniques.
- ***Neural networks.*** These are among the most popular machine-learning techniques that can be used for classification-type problems. A detailed description of this technique is presented in Chapter 6.
- ***Case-based reasoning.*** This approach uses historical cases to recognize commonalities in order to assign a new case into the most probable category.
- ***Bayesian classifiers.*** This approach uses probability theory to build classification models based on the past occurrences that are capable of placing a new instance into a most probable class (or category).
- ***Genetic algorithms.*** The use of the analogy of natural evolution to build directed-search-based mechanisms to classify data samples.
- ***Rough sets.*** This method takes into account the partial membership of class labels to predefined categories in building models (collection of rules) for classification problems.

A complete description of all of these classification techniques is beyond the scope of this book, thus only several of the most popular ones are presented here.

DECISION TREES Before describing the details of **decision trees,** we need to discuss some simple terminology. First, decision trees include many input variables that may have an impact on the classification of different patterns. These input variables are usually called *attributes.* For example, if we were to build a model to classify loan risks on the basis of just two characteristics—income and a credit rating—these two characteristics would be the attributes and the resulting output would be the *class label* (e.g., low, medium, or high risk). Second, a tree consists of branches and nodes. A *branch* represents the outcome of a test to classify a pattern (on the basis of a test) using one of the attributes. A *leaf node* at the end represents the final class choice for a pattern (a chain of branches from the root node to the leaf node which can be represented as a complex if-then statement).

The basic idea behind a decision tree is that it recursively divides a training set until each division consists entirely or primarily of examples from one class. Each nonleaf node of the tree contains a *split point,* which is a test on one or more attributes and determines how the data are to be divided further. Decision tree algorithms, in general, build an initial tree from the training data such that each leaf node is pure, and they then prune the tree to increase its generalization, and hence the prediction accuracy on test data.

In the growth phase, the tree is built by recursively dividing the data until each division is either pure (i.e., contains members of the same class) or relatively small. The basic idea is to ask questions whose answers would provide the most information, similar to what we may do when playing the game "Twenty Questions."

The split used to partition the data depends on the type of the attribute used in the split. For a continuous attribute A, splits are of the form value(A) $< x$, where x is

some "optimal" split value of A. For example, the split based on income could be "Income < 50000." For the categorical attribute A, splits are of the form value(A) belongs to x, where x is a subset of A. As an example, the split could be on the basis of gender: "Male versus Female."

A general algorithm for building a decision tree is as follows:

1. Create a root node and assign all of the training data to it.
2. Select the *best* splitting attribute.
3. Add a branch to the root node for each value of the split. Split the data into mutually exclusive (nonoverlapping) subsets along the lines of the specific split and mode to the branches.
4. Repeat the steps 2 and 3 for each and every leaf node until the stopping criteria is reached (e.g., the node is dominated by a single class label).

Many different algorithms have been proposed for creating decision trees. These algorithms differ primarily in terms of the way in which they determine the splitting attribute (and its split values), the order of splitting the attributes (splitting the same attribute only once or many times), the number of splits at each node (binary versus ternary), the stopping criteria, and the pruning of the tree (pre- versus postpruning). Some of the most well-known algorithms are ID3 (followed by C4.5 and C5 as the improved versions of ID3) from machine learning, classification and regression trees (CART) from statistics, and the chi-squared automatic interaction detector (CHAID) from pattern recognition.

When building a decision tree, the goal at each node is to determine the attribute and the split point of that attribute that best divides the training records in order to purify the class representation at that node. To evaluate the goodness of the split, some splitting indices have been proposed. Two of the most common ones are the Gini index and information gain. The Gini index is used in CART and SPRINT (Scalable PaRallelizable Induction of Decision Trees) algorithms. Versions of information gain are used in ID3 (and its newer versions, C4.5 and C5).

The **Gini index** has been used in economics to measure the diversity of a population. The same concept can be used to determine the purity of a specific class as a result of a decision to branch along a particular attribute or variable. The best split is the one that increases the purity of the sets resulting from a proposed split. Let us briefly look into a simple calculation of Gini index:

If a dataset S contains examples from n classes, the Gini index is defined as

$$gini(S) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is a relative frequency of class j in S. If a dataset S is split into two subsets, S_1 and S_2, with sizes N_1 and N_2, respectively, the Gini index of the split data contains examples from n classes, and the Gini index is defined as

$$gini_{split}(S) = \frac{N_1}{N} gini(S_1) + \frac{N_2}{N} gini(S_2)$$

The attribute/split combination that provides the smallest $gini_{split}(S)$ is chosen to split the node. In such a determination, one should enumerate all possible splitting points for each attribute.

Information gain is the splitting mechanism used in ID3, which is perhaps the most widely known decision tree algorithm. It was developed by Ross Quinlan in 1986, and since then he has evolved this algorithm into the C4.5 and C5 algorithms. The basic

idea behind ID3 (and its variants) is to use a concept called *entropy* in place of the Gini index. **Entropy** measures the extent of uncertainty or randomness in a dataset. If all the data in a subset belong to just one class, there is no uncertainty or randomness in that dataset; so the entropy is zero. The objective of this approach is to build subtrees so that the entropy of each final subset is zero (or close to zero). Let us also look at the calculation of the information gain.

Assume that there are two classes, P (positive) and N (negative). Let the set of examples S contain p counts of class P and n counts of class N. The amount of information needed to decide if an arbitrary example in S belongs to P or N is defined as

$$I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Assume that using attribute A a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$. If S_i contains p_i examples of P and n_i examples of N, the entropy, or the expected information needed to classify objects in all subtrees S_i, is

$$E(A) = \sum_{i=1}^{v}\frac{p_i + n_i}{p+n} I(p_i, n_i)$$

Then, the information that would be gained by branching on attribute A would be

$$Gain(A) = I(p, n) - E(A)$$

These calculations are repeated for each and every attribute, and the one with the highest information gain is selected as the splitting attribute. The basic ideas behind these splitting indices are rather similar to each other but the specific algorithmic details vary. A detailed definition of ID3 algorithm and its splitting mechanism can be found in Quinlan (1986).

APPLICATION CASE 5.6

Highmark, Inc., Employs Data Mining to Manage Insurance Costs

Highmark, Inc., based in Pittsburgh, Pennsylvania, has a long tradition of providing access to affordable, quality health care to its members and communities. Highmark was formed in 1996 by the merger of two Pennsylvania licensees of the Blue Cross and Blue Shield Association: Pennsylvania Blue Shield (now Highmark Blue Shield) and a Blue Cross plan in western Pennsylvania (now Highmark Blue Cross Blue Shield). Highmark is currently one of the largest health insurers in the United States.

Data in Managed Care Organizations

The amount of data floating around in managed care organizations such as Highmark is vast. These data, often considered to be occupying storage space and viewed as a menace to be dealt with, have recently been recognized as a source of new knowledge. Data mining tools and techniques provide practical means for analyzing patient data and unraveling mysteries that can lead to better managed care at lower costs—a mission that most managed care companies are trying to achieve.

Each day, managed care companies receive millions of data items about their customers, and each piece of information updates the case history of each member. Companies have become aware of the usefulness of the data at their disposal and use analytic software tools to extract patient clusters that are more costly than average to treat. Earlier efforts at using computer technology in order to extract patient-related actionable information were limited in establishing a connection between two different

diseases. For example, the software tool could scan through the data and report that diabetics or people suffering from coronary heart diseases were the most expensive to treat. However, these reporting-based software tools were inefficient in finding why these patients were getting sick or why some patients were more adversely affected by certain diseases than others. Data mining tools can solve some of these problems by analyzing multidimensional information and generating succinct relationships and correlations among different diseases and patient profiles.

Managed care organizations are inundated with data, and some of the companies do not want to add to the complexity by adding data mining applications. They may want to scan data for various reasons but are unable to decide why or how to analyze their data. Things are becoming brighter for patients as well as companies, however, because health insurance regulations are clearing the way for efficient data and structuring analysis.

The Need for Data Mining

Market pressures are driving managed care organizations to become more efficient, and hence to take data mining seriously. Customers are demanding more and better service, and competitors are becoming relentless, all of which are leading to the design and delivery of more customized products in a timely manner.

This customization brings us to the originating point of why and where the major portions of medical costs are occurring. Many organizations have started to use data mining software to predict who is more likely to fall sick and who is more likely to be the most expensive to treat. A look into the future has enabled organizations to filter out their costly patients and lower their Medicare costs by using preventive measures. Another important application of predictive studies is the management of premiums. An employer group that has a large number of employees falling in a higher cost bracket would see its rates increase.

Based on the historical data, predictive modeling might be able to foretell which patients are more likely to become a financial burden for the company. For example, a predictive modeling application might rate a diabetic patient as a high risk of increased medical costs, which by itself might not be actionable information. However, data mining implementation at Highmark draws a relationship between a diabetic patient and other patient- and environment-related parameters; that is, a patient with a specific cardiac condition might be at high risk of contracting diabetes. This relationship is drawn because the cardiac medication could lead the patient to developing diabetes later in life. Highmark officials testify to this fact by saying that they would not have monitored the patients for the cardiac medication and might not have drawn a relationship between the cardiac medication and diabetes. Medical research has been successful in codifying many of the complexities associated with patient conditions. Data mining has laid the foundation for better detection and proper intervention programs.

Sources: Condensed from G. Gillespie, "Data Mining: Solving Care, Cost Capers," *Health Data Management,* November 2004, **findarticles.com/p/articles/mi_km2925/is_200411/ai_n8622737** (accessed May 2009); and "Highmark Enhances Patient Care, Keeps Medical Costs Down with SAS," **sas.com/success/highmark.html** (accessed April 2006).

Cluster Analysis for Data Mining

Cluster analysis is an essential data mining method for classifying items, events, or concepts into common groupings called *clusters*. The method is commonly used in biology, medicine, genetics, social network analysis, anthropology, archaeology, astronomy, character recognition, and even in MIS development. As data mining has increased in popularity, the underlying techniques have been applied to business, especially to marketing. Cluster analysis has been used extensively for fraud detection (both credit card and e-commerce fraud) and market segmentation of customers in contemporary CRM systems. More applications in business continue to be developed as the strength of cluster analysis is recognized and used.

Cluster analysis is an exploratory data analysis tool for solving classification problems. The objective is to sort cases (e.g., people, things, events) into groups, or clusters,

so that the degree of association is strong among members of the same cluster and weak among members of different clusters. Each cluster describes the class to which its members belong. An obvious one-dimensional example of cluster analysis is to establish score ranges into which to assign class grades for a college class. This is similar to the cluster analysis problem that the U.S. Treasury faced when establishing new tax brackets in the 1980s. A fictional example of clustering occurs in J. K. Rowling's *Harry Potter* books. The Sorting Hat determines to which House (e.g., dormitory) to assign first-year students at the Hogwarts School. Another example involves determining how to seat guests at a wedding. As far as data mining goes, the importance of cluster analysis is that it may reveal associations and structures in data that were not previously apparent but are sensible and useful once found.

Cluster analysis results may be used to:

- Identify a classification scheme (e.g., types of customers)
- Suggest statistical models to describe populations
- Indicate rules for assigning new cases to classes for identification, targeting, and diagnostic purposes
- Provide measures of definition, size, and change in what were previously broad concepts
- Find typical cases to label and represent classes
- Decrease the size and complexity of the problem space for other data mining methods
- Identify outliers in a specific domain (e.g., rare-event detection)

DETERMINING THE OPTIMAL NUMBER OF CLUSTERS Clustering algorithms usually require one to specify the number of clusters to find. If this number is not known from prior knowledge, it should be chosen in some way. Unfortunately, there is not an optimal way of calculating what this number is supposed to be. Therefore, several different heuristic methods have been proposed. The following are among the most commonly referenced ones:

- Look at the percentage of variance explained as a function of the number of clusters; that is, choose a number of clusters so that adding another cluster would not give much better modeling of the data. Specifically, if one graphs the percentage of variance explained by the clusters, there is a point at which the marginal gain will drop (giving an angle in the graph), indicating the number of clusters to be chosen.
- Set the number of clusters to $(n/2)^{1/2}$, where n is the number of data points.
- Use the Akaike Information Criterion (AIC), which is a measure of the goodness of fit (based on the concept of entropy) to determine the number of clusters.
- Use Bayesian information criterion (BIC), which is a model-selection criterion (based on maximum likelihood estimation) to determine the number of clusters.

ANALYSIS METHODS Cluster analysis may be based on one or more of the following general methods:

- Statistical methods (including both hierarchical and nonhierarchical), such as *k*-means, *k*-modes, and so on.
- Neural networks (with the architecture called self-organizing map, or SOM)
- Fuzzy logic (e.g., fuzzy *c*-means algorithm)
- Genetic algorithms

Each of these methods generally works with one of two general method classes:

- ***Divisive.*** With divisive classes, all items start in one cluster and are broken apart.
- ***Agglomerative.*** With agglomerative classes, all items start in individual clusters, and the clusters are joined together.

Most cluster analysis methods involve the use of a **distance measure** to calculate the closeness between pairs of items. Popular distance measures include Euclidian distance (the ordinary distance between two points that one would measure with a ruler) and Manhattan distance (also called the rectilinear distance, or taxicab distance, between two points). Often, they are based on true distances that are measured, but this need not be so, as is typically the case in IS development. Weighted averages may be used to establish these distances. For example, in an IS development project, individual modules of the system may be related by the similarity between their inputs, outputs, processes, and the specific data used. These factors are then aggregated, pairwise by item, into a single distance measure.

K-MEANS CLUSTERING ALGORITHM The *k*-means algorithm (where *k* stands for the predetermined number of clusters) is arguably the most referenced clustering algorithm. It has its roots in traditional statistical analysis. As the name implies, the algorithm assigns each data point (customer, event, object, etc.) to the cluster whose center (also called *centroid*) is the nearest. The center is calculated as the average of all the points in the cluster; that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster. The algorithm steps are listed below and shown graphically in Figure 5.12:

Initialization step: Choose the number of clusters (i.e., the value of *k*).

Step 1 Randomly generate *k* random points as initial cluster centers.
Step 2 Assign each point to the nearest cluster center.
Step 3 Recompute the new cluster centers.

Repetition step: Repeat steps 2 and 3 until some convergence criterion is met (usually that the assignment of points to clusters becomes stable).

Association Rule Mining

Association rule mining is a popular data mining method that is commonly used as an example to explain what data mining is and what it can do to a technologically less savvy audience. Most of you might have heard the famous (or infamous, depending on how to look at it) relationship discovered between the sales of beer and diapers at grocery stores. As the story goes, a large supermarket chain (maybe Wal-Mart, maybe not; there is no consensus on which supermarket chain it was) did an analysis of customers'

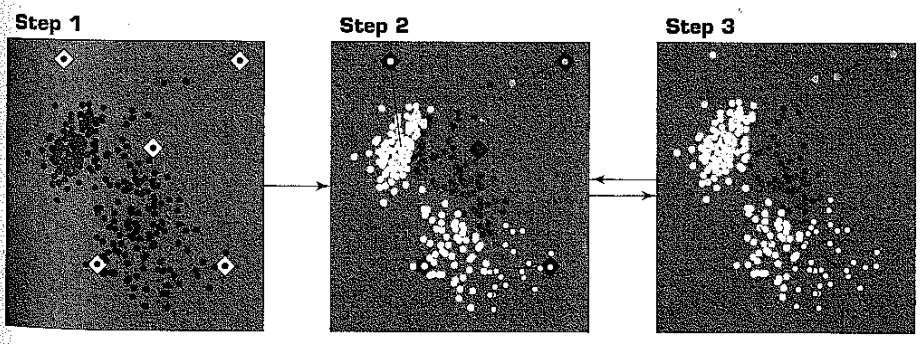


FIGURE 5.12 Graphical Illustration of the Steps in *k*-means Algorithm

buying habits and found a statistically significant correlation between purchases of beer and purchases of diapers. It was theorized that the reason for this was that fathers (presumably young men) were stopping off at the supermarket to buy diapers for their babies (especially on Thursdays), and since they could no longer go down to the sports bar as often, would buy beer as well. As a result of this finding, the supermarket chain is alleged to have placed the diapers next to the beer, resulting in increased sales of both.

In essence, association rule mining aims to find interesting relationships (affinities) between variables (items) in large databases. Because of its successful application to business problems, it is commonly called a *market-basket analysis*. The main idea in market-basket analysis is to identify strong relationships among different products (or services) that are usually purchased together (show up in the same basket together, either a physical basket at a grocery store or a virtual basket at an e-commerce Web site). For instance, market-basket analysis may find a pattern like, "If a customer buys lap-top computer and virus protection software, he/she also buys extended service plan 70 percent of the time." The input to market-basket analysis is the simple point-of-sale transaction data, where a number of products and/or services purchased together (just like the content of a purchase receipt) are tabulated under a single transaction instance. The outcome of the analysis is invaluable information that can be used to better understand customer-purchase behavior in order to maximize the profit from business transactions. A business can take advantage of such knowledge by (1) putting the items next to each other to make it more convenient for the customers to pick them up together and not forget to buy one when buying the others (increasing sales volume); (2) promoting the items as a package (do not put one on sale if the other(s) are on sale); and (3) placing them apart from each other so that the customer has to walk the aisles to search for it, and by doing so potentially seeing and buying other items.

Applications of market-basket analysis include cross-marketing, cross-selling, store design, catalog design, e-commerce site design, optimization of online advertising, product pricing, and sales/promotion configuration. In essence, market-basket analysis helps businesses infer customer needs and preferences from their purchase patterns. Outside the business realm, association rules are successfully used to discover relationships between symptoms and illnesses, diagnosis and patient characteristics and treatments (to be used in medical DSS), and genes and their functions (to be used in genomics projects), among others.

A good question to ask with respect to the patterns/relationships that association rule mining can discover is "Are all association rules interesting and useful?" In order to answer such a question, association rule mining uses two common metrics: **support** and **confidence**. Before defining these terms, let's get a little technical by showing what an association rule looks like:

$$X \Rightarrow Y\ [S\%,\ C\%]$$

{Laptop Computer, Antivirus Software} ⇒ {Extended Service Plan} [30%,70%]

Here, X (products and/or service; called the *left-hand side, LHS,* or the antecedent) is associated with Y (products and/or service; called the *right-hand side, RHS,* or *consequent*). S is the support, and C is the confidence for this particular rule. The support (S) of a rule is the measure of how often these products and/or services (i.e., LHS + RHS = Laptop Computer, Antivirus Software, and Extended Service Plan) appear together in the same transaction; that is, the proportion of transactions in the dataset that contain all of the products and/or services mentioned in a specific rule. In this

example, 30 percent of all transactions in the hypothetical store database had all three products present in a single sales ticket. The confidence of a rule is the measure of how often the products and/or services on the RHS (consequent) go together with the products and/or services on the LHS (antecedent); that is, the proportion of transactions that include LHS while also including the RHS. In other words, it is the conditional probability of finding the RHS of the rule present in transactions where the LHS of the rule already exists.

Several algorithms are available for generating association rules. Some well-known algorithms include Apriori, Eclat, and FP-Growth. These algorithms only do half the job, which is to identify the frequent itemsets in the database. Once the frequent itemsets are identified, they need to be converted into rules with antecedent and consequent parts. Determination of the rules from frequent itemsets is a straightforward matching process, but the process may be time consuming with large transaction databases. Even though there can be many items on each section of the rule, in practice the consequent part usually contains a single item. In the following section, one of the most popular algorithms for identification of frequent itemsets is explained.

APRIORI ALGORITHM The **Apriori algorithm** is the most commonly used algorithm to discover association rules. Given a set of itemsets (e.g., sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets that are common to at least a minimum number of the itemsets (i.e., complies with a minimum support). Apriori uses a bottom-up approach, where frequent subsets are extended one item at a time (a method known as *candidate generation,* whereby the size of frequent subsets increases from one-item subsets to two-item subsets, then three-item subsets, etc.), and groups of candidates at each level are tested against the data for minimum support. The algorithm terminates when no further successful extensions are found.

As an illustrative example, consider the following. A grocery store tracks sales transactions by SKU (stock-keeping unit) and thus knows which items are typically purchased together. The database of transactions, along with the subsequent steps in identifying the frequent itemsets, is shown in Figure 5.13. Each SKU in the transaction database corresponds to a product, such as "1 = butter," "2 = bread," "3 = water," and so on. The first step in Apriori is to count up the frequencies (i.e., the supports)

Raw Transaction Data

Transaction No	SKUs (Item No)
1	1, 2, 3, 4
1	2, 3, 4
1	2, 3
1	1, 2, 4
1	1, 2, 3, 4
1	2, 4

One-Item Itemsets

Itemset (SKUs)	Support
1	3
2	6
3	4
4	5

Two-Item Itemsets

Itemset (SKUs)	Support
1, 2	3
1, 3	2
1, 4	3
2, 3	4
2, 4	5
3, 4	3

Three-Item Itemsets

Itemset (SKUs)	Support
1, 2, 4	3
2, 3, 4	3

FIGURE 5.13 Identification of Frequent Itemsets in Apriori Algorithm

of each item (one-item itemsets). For this overly simplified example, let us set the minimum support to 3 (or 50%; meaning an itemset is considered to be a frequent itemset if it shows up in at least 3 out of 6 transactions in the database). Because all of the one-item itemsets have at least 3 in the support column, they are all considered frequent itemsets. However, had any of the one-item itemsets not been frequent, they would not have been included as a possible member of possible two-item pairs. In this way, Apriori *prunes* the tree of all possible itemsets. As Figure 5.13 shows, using one-item itemsets, all possible two-item itemsets are generated and the transaction database is used to calculate their support values. Because the two-item itemset {1, 3} has a support less than 3, it should not be included in the frequent itemsets that will be used to generate the next-level itemsets (three-item itemsets). The algorithm seems deceivingly simple, but only for small datasets. In much larger datasets, especially those with huge amounts of items present in low quantities and small amounts of items present in big quantities, the search and calculation become a computationally intensive process.

Section 5.5 Review Questions

1. Identify at least three of the main data mining methods.
2. Give examples of situations in which classification would be an appropriate data mining technique. Give examples of situations in which regression would be an appropriate data mining technique.
3. List and briefly define at least two classification techniques.
4. What are some of the criteria for comparing and selecting the best classification technique?
5. Briefly describe the general algorithm used in decision trees.
6. Define *Gini index*. What does it measure?
7. Give examples of situations in which cluster analysis would be an appropriate data mining technique.
8. What is the major difference between cluster analysis and classification?
9. What are some of the methods for cluster analysis?
10. Give examples of situations in which association would be an appropriate data mining technique.

5.6 DATA MINING SOFTWARE TOOLS

Many software vendors provide powerful data mining tools. Examples of these vendors include SPSS (PASW Modeler, formerly known as Clementine), SAS (Enterprise Miner), StatSoft (Statistica Data Miner), Salford (CART, MARS, TreeNet, RandomForest), Angoss (KnowledgeSTUDIO, KnowledgeSeeker), and Megaputer (PolyAnalyst). As can be seen, most of the more popular tools are developed by the largest statistical software companies (SPSS, SAS, and StatSoft). Most of the business intelligence tool vendors (e.g., IBM Cognos, Oracle Hyperion, SAP Business Objects, Microstrategy, Teradata, and Microsoft) also have some level of data mining capabilities integrated into their software offerings. These BI tools are still primarily focused on multidimensional modeling and data visualization and are not considered to be direct competitors of the data mining tool vendors.

In addition to these commercial tools, several open source and/or free data mining software tools are available online. Probably the most popular free (and open source) data mining tool is **Weka**, which is developed by a number of researchers from the University Waikato in New Zealand (the tool can be downloaded from **cs.waikato.ac.nz/ml/weka/**). Weka includes a large number of algorithms for different data mining tasks and has an intuitive user interface. Another recently released,

TABLE 5.6 Selected Data Mining Software

Product Name	Web Site (URL)
Clementine	spss.com/Clementine
Enterprise Miner	sas.com/technologies/bi/analytics/index.html
Statistica	statsoft.com/products/dataminer.htm
Intelligent Miner	ibm.com/software/data/iminer
PolyAnalyst	megaputer.com/polyanalyst.php
CART, MARS, TreeNet, RandomForest	salford-systems.com
Insightful Miner	insightful.com
XLMiner	xlminer.net
KXEN (Knowledge eXtraction ENgines)	kxen.com
GhostMiner	fqs.pl/ghostminer
Microsoft SQL Server Data Mining	microsoft.com/sqlserver/2008/data-mining.aspx
Knowledge Miner	knowledgeminer.net
Teradata Warehouse Miner	ncr.com/products/software/teradata_mining.htm
Oracle Data Mining (ODM)	otn.oracle.com/products/bi/9idmining.html
Fair Isaac Business Science	fairisaac.com/edm
DeltaMaster	bissantz.de
iData Analyzer	infoacumen.com
Orange Data Mining Tool	ailab.si/orange/
Zementis Predictive Analytics	zementis.com

free (for noncommercial use) data mining tool is **RapidMiner** (developed by Rapid-I; it can be downloaded from **rapid-i.com**). Its graphically enhanced user interface, employment of a rather large number of algorithms, and incorporation of a variety of data visualization features set it apart from the rest of the free tools. The main difference between commercial tools, such as Enterprise Miner, PASW, and Statistica, and free tools, such as Weka and RapidMiner, is computational efficiency. The same data mining task involving a rather large dataset may take a whole lot longer to complete with the free software, and in some cases it may not even be feasible (i.e., crashing due to the inefficient use of computer memory). Table 5.6 lists a few of the major products and their Web sites.

A suite of business intelligence capabilities that has become increasingly more popular for data mining studies is **Microsoft's SQL Server**, where data and the models are stored in the same relational database environment, making model management a considerably easier task. The **Microsoft Enterprise Consortium** serves as the worldwide source for access to Microsoft's SQL Server 2008 software suite for academic purposes—teaching and research. The consortium has been established to enable universities around the world to access enterprise technology without having to maintain the necessary hardware and software on their own campus. The consortium provides a wide range of business intelligence development tools (e.g., data mining, cube building, business reporting) as well as a number of large, realistic datasets from Sam's Club, Dillard's, and Tyson Foods. A screenshot that shows development of a decision tree for churn analysis in SQL Server 2008 Business Intelligence Development Suite is shown in Figure 5.14. The Microsoft Enterprise

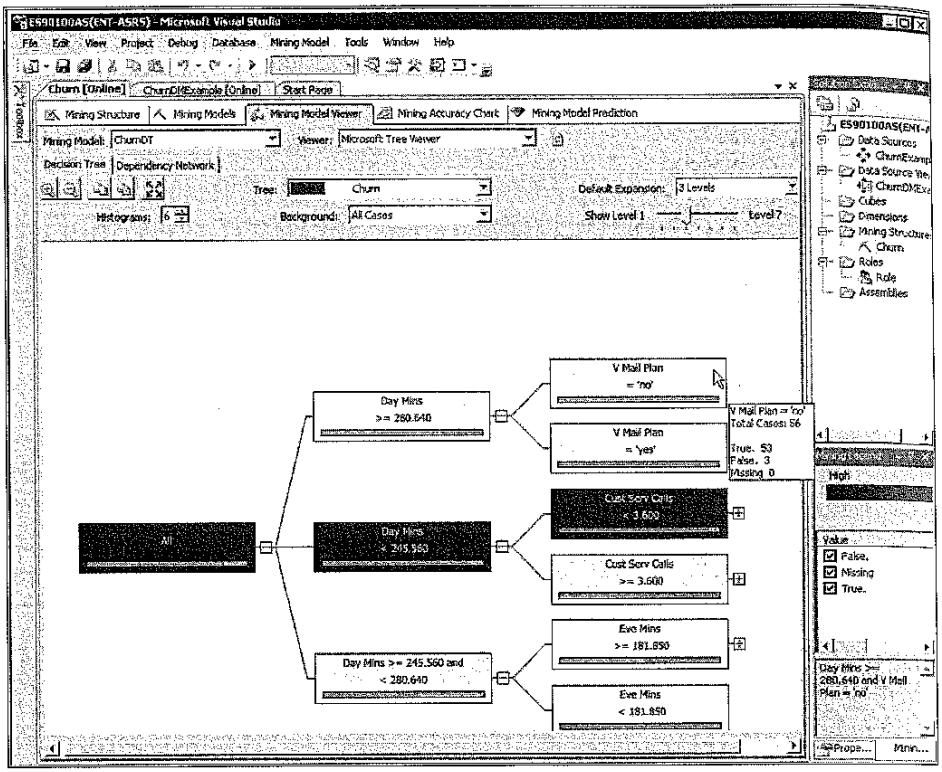


FIGURE 5.14 A Screenshot of a Decision Tree Development in SQL Server 2008 *Source:* Microsoft Enterprise Consortium and Microsoft SQL Server 2008.

Consortium is free of charge and can only be used for academic purposes. The Sam M. Walton College of Business at the University of Arkansas hosts the enterprise system and allows consortium members and their students to access these resources using a simple remote desktop connection. The details about becoming a part of the consortium along with easy-to-follow tutorials and examples can be found at **enterprise waltoncollege.uark.edu/mec/**.

A May 2009 survey by **kdnuggets.com** polled the data mining community on the following question: "What data mining tools have you used for a real project (not just for evaluation) in the past 6 months?" In order to make the results more representative, votes from tool vendors were removed. In previous years, there was a very strong correlation between the use of SPSS Clementine and SPSS Statistics as well as **SAS Enterprise Miner** and SAS Statistics, thus the votes for these two tool families were grouped together. In total, 364 unique votes were counted toward the rankings. The most popular tools were **SPSS PASW Modeler** (formerly Clementine), RapidMiner, SAS Enterprise Miner, and Microsoft Excel. Compared to poll results in previous years (see 2008 data at **kdnuggets.com/polls/2008/data-mining-software-tools-used.htm**), among commercial tools SPSS PASW Modeler, StatSoft Statistica, and SAS Enterprise Miner showed the most growth; among the free tools, RapidMiner and Orange showed the most growth. The results are shown in Figure 5.15.

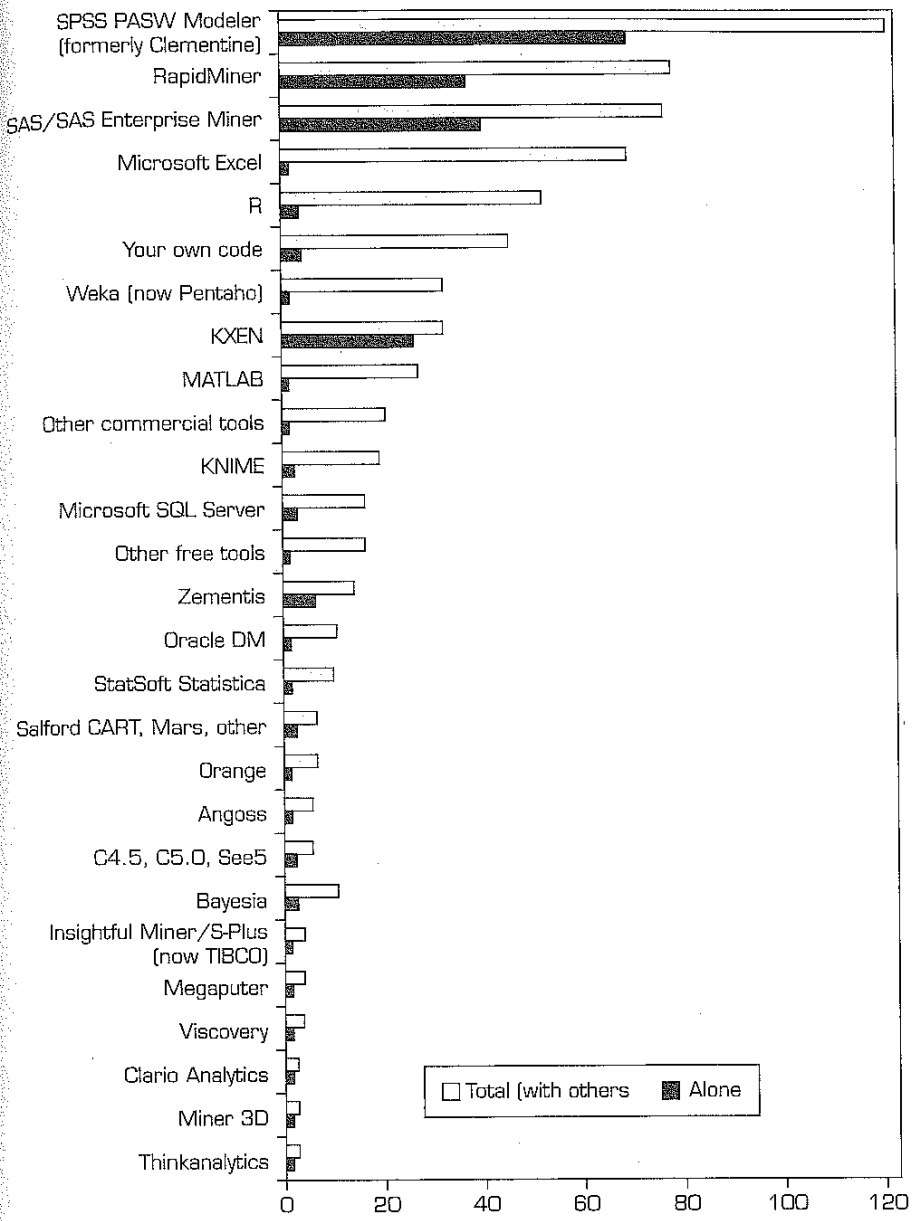


FIGURE 5.15 Popular Data Mining Software Tools (Poll Results) *Source:* Used with permission of **kdnuggets.com.**

APPLICATION CASE 5.7

Predicting Customer Churn—A Competition of Different Tools

In 2003, the Duke University/NCR Teradata Center sought to identify the best predictive modeling techniques to help manage a vexing problem for wireless telecommunications providers: customer churn. Although other industries are also faced with customers who defect to competitors, at the

retail level, wireless customers switch service providers at a rate of about 25 percent per year, or 25 per month. In the early 1990s when new subscriber growth rates were in the 50 percent range, telecommunications companies were tempted to focus on new customer acquisition rather than on customer retention. However, in a new era of slower growth rates—as low as 10 percent—it is becoming clear that customer retention is vital to overall profitability.

The key to customer retention is predicting which customers are most at risk of defecting to a competitor and offering the most valuable incentives to stay. To execute such a strategy effectively, one must be able to develop highly accurate predictions—churn scorecards—so that the retention effort is focused on the relevant customers.

The Data

The data were provided by a major wireless telecommunications company using its own customer records for the second half of 2001. Account summary data was provided for 100,000 customers who had been with the company for at least 6 months. To assist in the modeling process, churners (those who left the company by the end of the following 60 days) were oversampled so that one-half of the sample consisted of churners and the other half were customers remaining with the company at least another 60 days. A broad range of 171 potential predictors was made available, spanning all the types of data a typical service provider would routinely have. Predictor data included:

- ***Demographics:*** Age, location, number and ages of children, etc.
- ***Financials:*** Credit score, credit card ownership
- ***Product details:*** Handset price, handset capabilities, etc.
- ***Phone usage:*** Number and duration of various categories of calls, etc.

Evaluation Criteria

The data were provided to support predictive modeling development. Participants (a mix of data mining software companies, university research centers, other non-profits and consultancy companies) were asked to use their best models to predict the probability of churn for two different groups of customers: a "current" sample of 51,306 drawn from the latter half of 2001 and a "future" sample of 100,462 customers drawn from the first quarter of 2002. Predicting "future" data is generally considered more difficult because external factors and behavioral patterns may change over time. In the real world, predictive models are always applied to future data, and the tournament organizers wanted to reproduce a similar context.

Each contestant in the tournament was asked to rank the current and future score samples in descending order by probability of churn. Using the actual churn status available to the tournament organizers, two performance measures were calculated for each predictive model: the overall Gini index and the lift in the top decile. The two measures were calculated for the two samples, current and future, so that there were four performance scores available for every contestant. Evaluation criteria are described in detail in a number of locations, including the tournament Web site. The top-decile lift is the easiest to explain: It measures the number of actual churners captured among the customers ranked most likely to churn by a model.

The Results

Contestants were free to develop a separate model for each measure if they wished to try to optimize their models both to either the time period, the evaluation criterion, or both. Salford Systems was declared the winner in all categories. Salford Systems used its TreeNet software to create the model. TreeNet is an innovative form of boosted decision tree analysis that is well known for building accurate classification models. Across all the entries, the judges found that decision trees and logistic regression methods were generally the best at predicting churn, though they acknowledged that not all methodologies were adequately represented in the competition.

Salford's TreeNet models captured the most churners across the board and discovered which of the 171 possible variables were most important for predicting churn. In the top 10 percent of customers, TreeNet found 35 to 45 percent more churners than the competition average, and three times more than

would be found in a random sample. For companies with large subscriber bases, this could translate to the identification of thousands more potential churners each month. Targeting these customers with an appropriate retention campaign could save a company millions of dollars each year.

Sources: Salford Systems, "The Duke/NCR Teradata Churn Modeling Tournament," **salford-systems.com/churn.php** (accessed April 20, 2009); and W. Yu, D. N. Jutla, and S. C. Sivakumar, "A Churn-Strategy Alignment Model for Managers in Mobile Telecom," *Proceedings of the Communication Networks and Services Research Conference*, IEEE Publications, 2005, pp. 48–53.

Section 5.6 Review Questions

1. What are the most popular commercial data mining tools?
2. Why do you think the most popular tools are developed by statistics companies?
3. What are the most popular free data mining tools?
4. What are the main differences between commercial and free data mining software tools?
5. What would be your top five selection criteria for a data mining tool? Explain.

5.7 DATA MINING MYTHS AND BLUNDERS

Data mining is a powerful analytical tool that enables business executives to advance from describing the nature of the past to predicting the future. It helps marketers find patterns that unlock the mysteries of customer behavior. The results of data mining can be used to increase revenue, reduce expenses, identify fraud, and locate business opportunities, offering a whole new realm of competitive advantage. As an evolving and maturing field, data mining is often associated with a number of myths, including the following (Zaima, 2003):

Myth	Reality
Data mining provides instant, crystal-ball-like predictions.	Data mining is a multistep process that requires deliberate, proactive design and use.
Data mining is not yet viable for business applications.	The current state-of-the-art is ready to go for almost any business.
Data mining requires a separate, dedicated database.	Because of advances in database technology, a dedicated database is not required; even though it may be desirable.
Only those with advanced degrees can do data mining.	Newer Web-based tools enable managers of all educational levels to do data mining.
Data mining is only for large firms that have lots of customer data.	If the data accurately reflect the business or its customers, a company can use data mining.

Data mining visionaries have gained enormous competitive advantage by understanding that these myths are just that: myths.

The following 10 data mining mistakes are often made in practice (Skalak, 2001; Shultz, 2004), and you should try to avoid them:

1. Selecting the wrong problem for data mining.
2. Ignoring what your sponsor thinks data mining is and what it really can and cannot do.

3. Leaving insufficient time for data preparation. It takes more effort than is generally understood.
4. Looking only at aggregated results and not at individual records. IBM's DB2 IMS can highlight individual records of interest.
5. Being sloppy about keeping track of the data mining procedure and results.
6. Ignoring suspicious findings and quickly moving on.
7. Running mining algorithms repeatedly and blindly. It is important to think hard about the next stage of data analysis. Data mining is a very hands-on activity.
8. Believing everything you are told about the data.
9. Believing everything you are told about your own data mining analysis.
10. Measuring your results differently from the way your sponsor measures them.

Section 5.7 Review Questions

1. What are the most common myths about data mining?
2. What do you think are the reasons for these myths about data mining?
3. What are the most common data mining mistakes? How can they be minimized and/or eliminated?

Chapter Highlights

- Data mining is the process of discovering new knowledge from databases.
- Data mining can use simple flat files as data sources or it can be performed on data in data warehouses.
- There are many alternative names and definitions for data mining.
- Data mining is at the intersection of many disciplines, including statistics, artificial intelligence, and mathematical modeling.
- Companies use data mining to better understand their customers and optimize their operations.
- Data mining applications can be found in virtually every area of business and government, including health care, finance, marketing, and homeland security.
- Three broad categories of data mining tasks are prediction (classification or regression), clustering, and association.
- Similar to other information systems initiatives, a data mining project must follow a systematic project management process to be successful.
- Several data mining processes have been proposed: CRISP-DM, SEMMA, KDD, etc.
- CRISP-DM provides a systematic and orderly way to conduct data mining projects.
- The earlier steps in data mining projects (i.e., understanding the domain and the relevant data) consume most of the total project time (often more than 80% of the total time).
- Data preprocessing is essential to any successful data mining study. Good data leads to good information; good information leads to good decisions.
- Data preprocessing includes four main steps: data consolidation, data cleaning, data transformation, and data reduction.
- Classification methods learn from previous examples containing inputs and the resulting class labels, and once properly trained they are able to classify future cases.
- Clustering partitions pattern records into natural segments or clusters. Each segment's members share similar characteristics.
- Data mining can be hypothesis driven or discovery driven. Hypothesis-driven data mining begins with a proposition by the user. Discovery-driven data mining is a more open-ended expedition.
- A number of different algorithms are commonly used for classification. Commercial implementations include ID3, C4.5, C5, CART, and SPRINT.
- Decision trees partition data by branching along different attributes so that each leaf node has all the patterns of one class.
- The Gini index and information gain (entropy) are two popular ways to determine branching choices in a decision tree.

- The Gini index measures the purity of a sample. If everything in a sample belongs to one class, the Gini index value is zero.
- Several assessment techniques can measure the prediction accuracy of classification models, including simple split, *k*-fold cross-validation, bootstrapping, and area under the ROC curve.
- Cluster algorithms are used when the data records do not have predefined class identifiers (i.e., it is not known to what class a particular record belongs).
- Cluster algorithms compute measures of similarity in order to group similar cases into clusters.
- The most commonly used similarity measure in cluster analysis is a distance measure.
- The most commonly used clustering algorithms are *k*-means and self-organizing maps.
- Association rule mining is used to discover two or more items (or events or concepts) that go together.
- Association rule mining is commonly referred to as market-basket analysis.
- The most commonly used association algorithm is Apriori, whereby frequent itemsets are identified through a bottom-up approach.
- Association rules are assessed based on their support and confidence measures.
- Many commercial and free data mining tools are available.
- The most popular commercial data mining tools are SPSS PASW and SAS Enterprise Miner.
- The most popular free data mining tools are Weka and RapidMiner.

Key Terms

Apriori algorithm *227*
area under the ROC curve *219*
association *203*
bootstrapping *219*
categorical data *198*
classification *201*
clustering *202*
confidence *226*
CRISP-DM *207*
data mining *196*
decision tree *220*
discovery-driven data mining *204*
distance measure *225*
entropy *222*
Gini index *221*
hypothesis-driven data mining *204*
information gain *221*
interval data *198*
k-fold cross-validation *218*
knowledge discovery in databases (KDD) *215*
link analysis *203*
Microsoft Enterprise Consortium 229
Microsoft SQL Server 229
nominal data *198*
numeric data *198*
ordinal data *198*
prediction *201*
RapidMiner *229*
ratio data *198*
regression *216*
SAS Enterprise Miner *230*
SEMMA *214*
sequence mining *203*
simple split *217*
SPSS PASW Modeler *230*
support *226*
Weka *228*

Questions for Discussion

1. Define *data mining*. Why are there many names and definitions for data mining?
2. What are the main reasons for the recent popularity of data mining?
3. Discuss what an organization should consider before making a decision to purchase data mining software.
4. Distinguish data mining from other analytical tools and techniques.
5. Discuss the main data mining methods. What are the fundamental differences among them?
6. What are the main data mining application areas? Discuss the commonalities of these areas that make them a prospect for data mining studies.
7. Why do we need a standardized data mining process? What are the most commonly used data mining processes?
8. Discuss the differences between the two most commonly used data mining process.
9. Are data mining processes a mere sequential set of activities?
10. Why do we need data preprocessing? What are the main tasks and relevant techniques used in data preprocessing?
11. Discuss the reasoning behind the assessment of classification models.
12. What is the main difference between classification and clustering? Explain using concrete examples.
13. Moving beyond the chapter discussion, where else can association be used?
14. What are the most common myths and mistakes about data mining?

Exercises

TERADATA STUDENT NETWORK (TSN) AND OTHER HANDS-ON EXERCISES

1. Visit **teradatastudentnetwork.com**. Identify cases about data mining. Describe recent developments in the field.
2. Go to **teradatastudentnetwork.com** or a URL provided by your instructor. Locate Web seminars related to data mining. In particular, locate a seminar given by C. Imhoff and T. Zouqes. Watch the Web seminar. Then answer the following questions:
   a. What are some of the interesting applications of data mining?
   b. What types of payoffs and costs can organizations expect from data mining initiatives?
3. For this exercise, your goal is to build a model to identify inputs or predictors that differentiate risky customers from others (based on patterns pertaining to previous customers) and then use those inputs to predict new risky customers. This sample case is typical for this domain.

   The sample data to be used in this exercise are in Online File W5.1 in the file **CreditRisk.xlsx**. The dataset has 425 cases and 15 variables pertaining to past and current customers who have borrowed from a bank for various reasons. The dataset contains customer-related information such as financial standing, reason for the loan, employment, demographic information, and the outcome or dependent variable for credit standing, classifying each case as good or bad, based on the institution's past experience.

   Take 400 of the cases as training cases and set aside the other 25 for testing. Build a decision tree model to learn the characteristics of the problem. Test its performance on the other 25 cases. Report on your model's learning and testing performance. Prepare a report that identifies the decision tree model and training parameters, as well as the resulting performance on the test set. Use any decision tree software. (This exercise is courtesy of StatSoft, Inc., based on a German dataset from **ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german** renamed CreditRisk and altered.)
4. For this exercise, you will replicate (on a smaller scale) the box-office prediction modeling explained in the opening vignette. Download the training dataset from Online File W5.2, **MovieTrain.xlsx**, which has 184 records and is in Microsoft Excel format. Use the data description given in the opening vignette to understand the domain and the problem you are trying to solve. Pick and choose your independent variables. Develop at least three classification models (e.g., decision tree, logistic regression, neural networks). Compare the accuracy results using 10-fold cross-validation and percentage split techniques, use confusion matrices, and comment on the outcome. Test the models you have developed on the test set (see Online File W5.3, **MovieTest.xlsx**, 29 records). Analyze the results with different models and come up with the best classification model, supporting it with your results.

TEAM ASSIGNMENTS AND ROLE-PLAYING

1. Examine how new data-capture devices such as radio frequency identification (RFID) tags help organizations accurately identify and segment their customers for activities such as targeted marketing. Many of these applications involve data mining. Scan the literature and the Web and then propose five potential new data mining applications of RFID technology. What issues could arise if a country's laws required such devices to be embedded in everyone's body for a national identification system?
2. Interview administrators in your college or executives in your organization to determine how data warehousing, data mining, OLAP, and visualization BI/DSS tools could assist them in their work. Write a proposal describing your findings. Include cost estimates and benefits in your report.
3. A very good repository of data that has been used to test the performance of many machine-learning algorithms is available at **ics.uci.edu/~mlearn/MLRepository.html**. Some of the datasets are meant to test the limits of current machine-learning algorithms and to compare their performance with new approaches to learning. However, some of the smaller datasets can be useful for exploring the functionality of any data mining software or the software that is available as companion software with this book, such as Statistica Data Miner. Download at least one dataset from this repository (e.g., Credit Screening Databases, Housing Database) and apply decision tree or clustering methods, as appropriate. Prepare a report based on your results. (Some of these exercises may even be proposed as semester-long projects for term papers, for example.)
4. Consider the following dataset, which includes three attributes and a classification for admission decisions into an MBA program:

GMAT	GPA	Quantitative GMAT Score (percentile)	Decision
650	2.75	35	No
580	3.50	70	No
600	3.50	75	Yes
450	2.95	80	No
700	3.25	90	Yes
590	3.50	80	Yes
400	3.85	45	No
640	3.50	75	Yes
540	3.00	60	?
690	2.85	80	?
490	4.00	65	?

a. Using the data shown, develop your own manual expert rules for decision making.

b. Use the Gini index to build a decision tree. You can use manual calculations or a spreadsheet to perform the basic calculations.

c. Use an automated decision tree software program to build a tree for the same data.

INTERNET EXERCISES

1. Visit the AI Exploratorium at **cs.ualberta.ca/~aixplore/**. Click the Decision Tree link. Read the narrative on basketball game statistics. Examine the data and then build a decision tree. Report your impressions of the accuracy of this decision tree. Also, explore the effects of different algorithms.
2. Survey some data mining tools and vendors. Start with **fairisaac.com** and **egain.com**. Consult **dmreview.com** and identify some data mining products and service providers that are not mentioned in this chapter.
3. Find recent cases of successful data mining applications. Visit the Web sites of some data mining vendors and look for cases or success stories. Prepare a report summarizing five new case studies.
4. Go to vendor Web sites (especially those of SAS, SPSS, Cognos, Teradata, StatSoft, and Fair Isaac) and look at success stories for BI (OLAP and data mining) tools. What do the various success stories have in common? How do they differ?
5. Go to **statsoft.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
6. Go to **sas.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
7. Go to **spss.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
8. Go to **teradata.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
9. Go to **fairisaac.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
10. Go to **salfordsystems.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
11. Go to **rulequest.com**. Download at least three white papers on applications. Which of these applications may have used the data/text/Web mining techniques discussed in this chapter?
12. Go to **kdnuggets.com**. Explore the sections on applications as well as software. Find names of at least three additional packages for data mining and text mining.

END OF CHAPTER APPLICATION CASE

Data Mining Helps Develop Custom-Tailored Product Portfolios for Telecommunication Companies

Background

The consulting group argonauten360° helps businesses build and improve successful strategies for customer relationship management (CRM). The company uses Relevanz-Marketing to create value by facilitating dialogue with relevant customers. Its clients include, among many others, BMW, Allianz, Deutsche Bank, Gerling, and Coca-Cola.

The Problem

As a leading consulting company to the telecommunications industry (as well as others), argonauten360° applies effective advanced analytic technologies for client scoring, clustering, and life-time-value computations as a routine part of its daily work. The requirements for flexible and powerful analytic tools are demanding, because each project typically presents a new and specific set of circumstances, data scenarios, obstacles, and analytic challenges. Therefore, the existing toolset needed to be augmented with effective, cutting-edge, yet flexible, data mining capabilities. Another critical consideration was for the solution to yield quick return on investment. The solution had to be easy to apply, with a fast learning curve, so that analysts could quickly take ownership of even the most advanced analytic procedures.

The Solution

The company needed a unified, easy-to-use set of analytical tools with a wide range of modeling capabilities and straightforward deployment options. Having to learn different tools for different modeling tasks has significantly hindered the efficiency and effectiveness of the company's consultants, causing it to lean toward a unified solution environment with capabilities ranging from data access on any medium (e.g., databases, online data repositories, text documents, XML files) to deployment of sophisticated data mining solutions on a wide range of BI systems.

After 12 months of evaluating a wide range of data mining tools, the company chose Statistica Data Miner (by StatSoft, Inc.) because (according to company executives) it provided the ideal combination of features to satisfy most every analyst's needs and requirements with user-friendly interfaces.

An Example of an Innovative Project

In Europe, so-called "call-by-call" services are very popular with cell phone users as well as with regular phone users. Such plans have no (or very low) charges for basic service, but bill for the actual air time that is used. It is a very competitive business, and the success of the call-by-call telecommunications provider depends greatly on attractive per-minute calling rates. Rankings of those rates are widely published, and the key is to be ranked somewhere in the top-five lowest-cost providers while maintaining the best possible margins. Because of the competitive environment created by this situation, popular wisdom holds that "there is virtually no price elasticity in this market (to allow providers to charge even the smallest extra margin without losing customers); and even if such price elasticity existed, it certainly could not be predicted." However, the argonauten360° consultants analyzed the available data with Statistica's data mining tool and proved that popular wisdom is wrong! Indeed, their successful analyses won argonauten360° the business of a leading provider of call-by-call services.

The Analysis

The analysis was based on data describing minute-by-minute phone traffic. Specifically, the sale of minutes of airtime over a 1-year period was analyzed. To obtain the best possible discrimination, 20 ensembles of different types of models were developed for estimation purposes. Each model employed a regression-type mathematical representation function for predicting the long-term trends; individual models were

then combined at a higher level meta-model. All specific time intervals (time "zones") were carefully modeled, identifying each zone with particular price sensitivity and competitive pressures.

Results After 2 Months

Prior to the application of the models derived via data mining, heuristic "expert-opinions" were used to forecast the expected volume of minutes (of airtime) for the following 2 months. By using Statistica Data Miner, the accuracy of these prognoses improved significantly, while the error rate was cut in half. Given the enormous volume of minute-to-minute calling traffic (airtime), this was deemed to be a dramatically pleasing result, thus providing clear proof for the efficacy and potential benefits of advanced analytic strategies when applied to problems of this type.

Implementing the Solution at the Customer Site

The call-by-call provider now uses this solution for predicting and simulating optimal cellular (airtime) rates. The system was installed by argonauten360° as a complete turn-key ("push-of-the-button") solution. Using this solution, the call-by-call provider can now predict with much greater accuracy the demand (for airtime) in a highly price-sensitive and competitive market and offer the "correct" rates, thus enjoying a key competitive advantage.

In a second phase, this system will be further improved with a "dashboard-like" system that automatically compares predictions with observed data. This system will ensure that, when necessary, argonauten360° can update the estimates of model parameters to adjust to the dynamic marketplace. Hence, without acquiring any analytic know-how, the call-by-call provider now has access to a reliable implementation of a sophisticated demand-forecasting and rate-simulation system—something previously considered impossible. This is an excellent example of a successful application of data mining technologies to help the company gain competitive advantage in a highly competitive business environment.

Questions for the Case

1. Why do you think that consulting companies are more likely to use data mining tools and techniques? What specific value proposition do they offer?
2. Why was it important for argonauten360° to employ a comprehensive tool that has all modeling capabilities?
3. What was the problem that argonauten360° helped solve for a call-by-call provider?
4. Can you think of other problems for telecommunication companies that are likely to be solved with data mining?

Source: StatSoft, "The German Consulting Company argonauten360° Uses Statistica Data Miner to Develop Effective Product Portfolios Custom-Tailored to Their Customers," **statsoft.com/company/success_stories/pdf/argonauten360.pdf** (accessed on May 25,2009).

References

Bhandari, I., E. Colet, J. Parker, Z. Pines, R. Pratap, and K. Ramanujam. (1997). "Advanced Scout: Data Mining and Knowledge Discovery in NBA Data." *Data Mining and Knowledge Discovery,* Vol. 1, No. 1, pp. 121–125.

Buck, N. (December 2000/January 2001). "Eureka! Knowledge Discovery." *Software Magazine.*

Chan, P. K., W. Phan, A. Prodromidis, and S. Stolfo. (1999). "Distributed Data Mining in Credit Card Fraud Detection." *IEEE Intelligent Systems,* Vol. 14, No. 6, pp. 67–74.

CRISP-DM. (2009). Cross-Industry Standard Process for Data Mining (CRISP-DM). **crisp-dm.org**.

Davenport, T. H. (2006, January). "Competing on Analytics." *Harvard Business Review.*

Delen, D., R. Sharda, and P. Kumar. (2007). "Movie Forecast Guru: A Web-based DSS for Hollywood Managers." *Decision Support Systems,* Vol. 43, No. 4, pp. 1151–1170.

Dunham, M. (2003). *Data Mining: Introductory and Advanced Topics.* Upper Saddle River, NJ: Prentice Hall.

Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth. (1996). "From Knowledge Discovery in Databases." *AI Magazine,* Vol. 17, No. 3, pp. 37–54.

Hoffman, T. (1998, December 7). "Banks Turn to IT to Reclaim Most Profitable Customers." *Computerworld.*

Hoffman, T. (1999, April 19). "Insurers Mine for Age-Appropriate Offering." *Computerworld.*

Kohonen, T. (1982). "Self-organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics,* Vol. 43, No. 1, pp. 59–69.

Nemati, H. R., and C. D. Barko. (2001). "Issues in Organizational Data Mining: A Survey of Current Practices." *Journal of Data Warehousing,* Vol. 6, No. 1, pp. 25–36.

Quinlan, J. R. (1986). "Induction of Decision Trees." *Machine Learning,* Vol. 1, pp. 81–106.

SEMMA. (2009). "SAS's Data Mining Process: Sample, Explore, Modify, Model, Assess." **sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html** (accessed August 2009).

Sharda, R., and Delen, D. (2006). "Predicting Box-office Success of Motion Pictures with Neural Networks." *Expert Systems with Applications,* Vol. 30, pp. 243–254.

Shultz, R. (2004, December 7). "Live from NCDM: Tales of Database Buffoonery." **directmag.com/news/ncdm-12-07-04/index.html** (accessed April 2009).

Skalak, D. (2001). "Data Mining Blunders Exposed!" *DB2 Magazine,* Vol. 6, No. 2, pp. 10–13.

StatSoft. (2006). "Data Mining Techniques." **statsoft.com/textbook/stdatmin.html** (accessed August 2006).

Wilson, R., and R. Sharda. (1994). "Bankruptcy Prediction Using Neural Networks." *Decision Support Systems,* Vol. 11, pp. 545–557.

Zaima, A. (2003). "The Five Myths of Data Mining." *What Works: Best Practices in Business Intelligence and Data Warehousing,* Vol. 15, the Data Warehousing Institute, Chatsworth, CA, pp. 42–43.

CHAPTER

6

Artificial Neural Networks for Data Mining

LEARNING OBJECTIVES

1 Understand the concept and definitions of artificial neural networks (ANN)
2 Know the similarities and differences between biological and artificial neural networks
3 Learn the different types of neural network architectures
4 Learn the advantages and limitations of ANN
5 Understand how backpropagation learning works in feedforward neural networks
6 Understand the step-by-step process of how to use neural networks
7 Appreciate the wide variety of applications of neural networks

Neural networks have emerged as advanced data mining tools in cases where other techniques may not produce satisfactory predictive models. As the term implies, neural networks have a biologically inspired modeling capability but are essentially statistical modeling tools. In this chapter, we study the basics of neural networks, different types of neural network architectures, some specific applications, and the process of implementing a neural network project.

6.1 OPENING VIGNETTE: PREDICTING GAMBLING REFERENDA WITH NEURAL NETWORKS

Desperate for new income and employment opportunities, as well as the need to introduce resilience into the local economy, many communities now offer a variety of incentives for new tourism businesses. One example can be found in the strong support for the abolition of laws that prohibit gambling. A vast majority of the states have placed ballots to legalize different types of gambling, some of them more than once. Proponents of legalized gambling argue that the expansion of gambling-tourism can create significant positive long-term sociocultural (e.g., better living conditions, more leisure opportunities, stronger cultural identity) and economic benefits (improved job opportunities, more disposable income, increased tax revenue, and more) for many communities.

Although many consider legalized gambling to be vital to the regeneration and revitalization of inner cities and economically depressed areas, attempts to legalize gambling have generally been met with caution or resistance. The lack of support for legalization of gambling originates from both perceived and actual ethical concerns. Opponents of gambling argue that such activities defy religious beliefs and work ethics; invite political corruption, swindling, money laundering, and organized crime; erode traditional family and societal values and responsibilities; and instigate irresponsible behavior. In addition, gambling may produce negative fiscal externalities, such as increased state expenditures on public welfare and police protection. Local communities react to gambling via three methods: (1) by judicially prohibiting the activity through the court of law, (2) by judicially legalizing the activity and controlling it through regulatory licensing laws, and (3) by overlooking the politically controversial issue.

Despite the existence of substantial literature on gambling and lottery adoption, either from a behavioral standpoint or from a socioeconomic viewpoint, the literature on the prediction of gambling-ballot outcomes seems to be deficient. In order to fill this gap, Sirakaya et al. (2005) used artificial neural networks (ANN) to gain an in-depth understanding of the factors affecting both legalization and prohibition of gambling. Their results have been shown to be superior when compared to other forecasting techniques used in analyzing gambling-related datasets.

In order to identify factors that may have an effect on determining a gambling ballot outcome, the researchers studied previously conducted and published studies, interviewed experts in the gaming industry, and investigated the theoretical foundation developed with behavioral studies. After much deliberation, the variables potentially affecting voting behavior were consolidated and synthesized.

Data

The data for the study was collected using both primary and secondary data collection techniques. Primary data related to gambling ballot outcomes (yes/no votes on gambling propositions) were obtained from all 50 state-election offices. Secondary data was compiled from a variety of sources: county-level religious data containing information about the number of churches and church members; other county-level data, such as population estimates, age, personal income, ethnicity, gender, poverty level, and education, were extracted from the U.S. Census Bureau and state data centers. The dataset included 1,287 records, each representing a county's voting outcome from a past ballot. After going through variable identification and dimensional reduction analysis, the researchers settled on the following list of variables to be used in their ANN models:

- Ballot Type I (Gambling versus Wagering, a binary variable)
- Percent population voted (real-valued numeric variable)
- Medium family income (integer-valued numeric variable)
- Percent population church members (real-valued numeric variable)
- Percent population male (real-valued numeric variable)
- Poverty level (real-valued numeric variable)
- Unemployment rate (real-valued numeric variable)
- Percent population minority—non-White percentage (real-valued numeric variable)
- Percent population older than 45 (real-valued numeric variable)
- Metropolitan statistical area (yes/no, a binary variable)

The dependent variable was the *ballot outcome* having the values of yes (i.e., majority of the people in county said "yes" to legalized gambling) and no (i.e., majority of the people in county said "no" to legalized gambling).

Solution

The researchers chose to use a multilayered perceptron (MLP) neural network architecture (i.e., feedforward neural network with backpropagation learning algorithm) because of its reputation as an excellent predictor for this type of classification problem. Figure 6.1 shows the schematic representation of the neural network model structure used for the study. As Figure 6.1 illustrates, information flows from left to right in a feedforward neural network, starting from the input data and going through the weights of the hidden neurons, ultimately creating the output. At that moment, the output is compared against the actual outcome of the same event and the difference (error or delta) is propagated back to the network to adjust the neural weights (backpropagation learning) so that the next time the same or a similar event is presented to the network, the error will be smaller.

The following procedure was followed to develop the ANN model:

1. The data was validated for missing and null values. Records with missing and/or null values were removed from the dataset.
2. Records (rows) in the dataset were randomized among themselves in order to have a truly random dataset where any part of the dataset would represent the behavior of the whole dataset.
3. The randomized dataset (1,287 records) was split into three separate data files: (1) training data, (2) cross-validation data, and (3) testing data. Because the records were randomized before the splitting, it is safe to say that each dataset represented the general behavior of the model individually. A common practice is to split a dataset into three parts using the following percentages: 60 percent for training (773 observations), 20 percent for cross-validation (257 observations), and 20 percent for testing (257 observations). To optimize the predictive power of the neural network model, both the cross-validation and the training

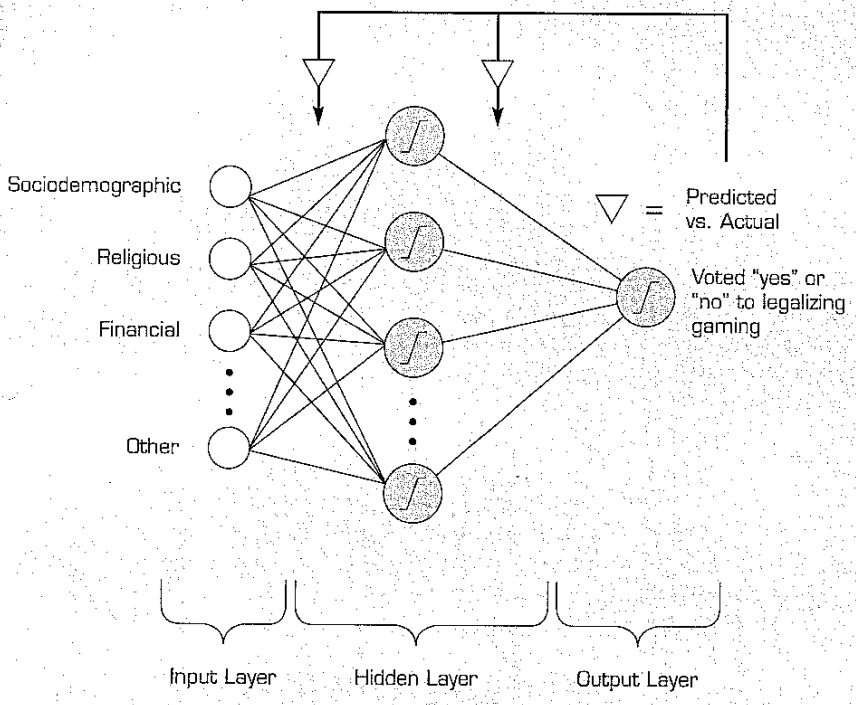


FIGURE 6.1 Schematic Representation of the Neural Network Model

datasets were used simultaneously. Once the predictive power of the training model reached the optimal level, the neural network weights were saved for the testing data.
4. Sensitivity analysis was performed to determine the cause-and-effect relationship between the inputs and outputs of a trained neural network model.

Results

The purpose of this study was to develop and test models that can be used as predictors of community support, or lack thereof, for commercial gaming. Specifically, the study examined the role of the factors that contribute to legalization and/or probation of gambling activities using artificial neural networks. Model-1 (predicting "no" votes) correctly predicted 201 out of 257 counties that would vote against gaming. Model-2 (predicting "yes" votes) correctly predicted 198 out of 257 counties that would vote for gaming.

On average, the ANN model predicted the voting outcome with 82 percent accuracy (correctly predicting four out of every five counties) on the test dataset (data that the ANN has not seen during model building process). Using sensitivity analysis on the trained neural network model, the researchers identified the most important variables in predicting gaming ballot outcomes. The most dominant variables were the county's religious inclination (i.e., percent church membership), the county's ethnic diversity (i.e., percent minority), and whether the county was classified as a Metropolitan Statistical Area (MSA) by the U.S. Census. Contrary to conventional wisdom, a county's financial characteristics (i.e., medium family income, poverty level, unemployment rate) and age distribution (i.e., percentage over 45) were not found to be significant factors in determining ballot outcomes.

Policy makers and the gaming industry could use the findings of this study (and/or similar studies) to predict which communities will pass a gambling initiative and which will strongly oppose it. The factors identified can be used as predictors for targeting those communities with high acceptance probabilities so as to effectively utilize resources to promote gambling and to avoid potential conflicts that may arise between the gaming industry and communities.

Questions for the Opening Vignette

1. Why is it important to study public opinion toward legalized gambling?
2. What factors might be used to predict public opinion toward gaming/gambling activities? Can you think of factors that are not mentioned in this case study?
3. What are the potential benefits and shortcomings of gaming/gambling for a county?
4. Why do you think ANN excels in analyzing this type of social choice problem?
5. What were the outcomes of the study? Who can use these results? How can the results be used?
6. Search the Internet to locate two additional cases that use ANN to predict public opinion.

What We Can Learn from This Vignette

As you will see in this chapter, neural networks can be applied in a wide range of areas, from standard business problems of assessing customer needs to understanding and enhancing security to improving health care and medicine. This vignette illustrates an innovative application of neural networks to predict public opinion, which most experts believe is an unpredictable phenomenon. In fact, conventional wisdom suggests that predicting the outcome of a public opinion poll is a hopeless effort, and the accuracy of such a model would not be any better than flipping a coin (i.e., random chance). However, the vignette shows that if ample effort is put forth to identify the potential factors there is very little (if anything) that cannot be predicted and analyzed with data mining techniques in general and neural networks in particular. As illustrated in the vignette, artificial neural network are not only good at predicting the outcome of complex social events, but they also are capable of revealing the underlying dynamics.

Sources: E. Sirakaya, D. Delen, and H-S. Choi, "Forecasting Gaming Referenda," *Annals of Tourism Research,* Vol. 32, No. 1, 2005, pp. 127–149; D. Delen and E. Sirakaya, "Determining the Efficacy of Data-Mining Methods in Predicting Gaming Ballot Outcomes," *Journal of Hospitality & Tourism Research,* Vol. 30, No. 3, 2006, pp. 313–332.

6.2 BASIC CONCEPTS OF NEURAL NETWORKS

Neural networks represent a brain metaphor for information processing. These models are biologically inspired rather than an exact replica of how the brain actually functions. Neural networks have been shown to be very promising systems in many forecasting and business classification applications due to their ability to "learn" from the data, their nonparametric nature (i.e., no rigid assumptions), and their ability to generalize. **Neural computing** refers to a pattern-recognition methodology for machine learning. The resulting model from neural computing is often called an **artificial neural network (ANN)** or a **neural network**. Neural networks have been used in many business applications for pattern recognition, forecasting, prediction, and classification. Neural network computing is a key component of any data mining tool kit. Applications of neural networks abound in finance, marketing, manufacturing, operations, information systems, and so on. Therefore, we devote this chapter to developing a better understanding of neural network models, methods, and applications.

The human brain possesses bewildering capabilities for information processing and problem solving that modern computers cannot compete with in many aspects. It has been postulated that a model or a system that is enlightened and supported by the results from brain research, with a structure similar to that of biological neural networks, could exhibit similar intelligent functionality. Based on this bottom-up approach, ANN (also known as *connectionist models, parallel distributed processing models, neuromorphic systems,* or simply *neural networks*) have been developed as biologically inspired and plausible models for various tasks.

Biological neural networks are composed of many massively interconnected **neurons**. Each neuron possesses **axons** and **dendrites**, fingerlike projections that enable the neuron to communicate with its neighboring neurons by transmitting and receiving electrical and chemical signals. More or less resembling the structure of their biological counterparts, ANN are composed of interconnected, simple processing elements called artificial neurons. When processing information, the processing elements in an ANN operate concurrently and collectively, similar to biological neurons. ANN possess some desirable traits similar to those of biological neural networks, such as the abilities to learn, to self-organize, and to support fault tolerance.

Coming along a winding journey, ANN have been investigated by researchers for more than half a century. The formal study of ANN began with the pioneering work of McCulloch and Pitts in 1943. Inspired by the results of biological experiments and observations, McCulloch and Pitts (1943) introduced a simple model of a binary artificial neuron that captured some of the functions of biological neurons. Using information-processing machines to model the brain, McCulloch and Pitts built their neural network model using a large number of interconnected artificial binary neurons. From these beginnings, neural network research became quite popular in the late 1950s and early 1960s. After a thorough analysis of an early neural network model (called the **perceptron**, which used no hidden layer) as well as a pessimistic evaluation of the research potential by Minsky and Papert in 1969, interest in neural networks diminished.

During the past two decades, there has been an exciting resurgence in ANN studies due to the introduction of new network topologies, new activation functions, and new learning algorithms, as well as progress in neuroscience and cognitive science. Advances in theory and methodology have overcome many of the obstacles that hindered neural network research a few decades ago. Evidenced by the appealing results of numerous studies, neural networks are gaining in acceptance and popularity. In addition, the desirable features in neural information processing make neural networks attractive for solving complex problems. ANN have been applied to numerous complex problems in a variety of application settings. The successful use of neural network applications has inspired renewed interest from industry and business.

Biological and Artificial Neural Networks

The human brain is composed of special cells called *neurons*. These cells do not die and replenish when a person is injured (all other cells reproduce to replace themselves and then die). This phenomenon may explain why humans retain information for an extended period of time and start to lose it when they get old—as the brain cells gradually start to die. Information storage spans sets of neurons. The brain has anywhere from 50 billion to 150 billion neurons, of which there are more than 100 different kinds. Neurons are partitioned into groups called *networks*. Each network contains several thousand highly interconnected neurons. Thus, the brain can be viewed as a collection of neural networks.

The ability to learn and to react to changes in our environment requires intelligence. The brain and the central nervous system control thinking and intelligent behavior. People who suffer brain damage have difficulty learning and reacting to changing environments. Even so, undamaged parts of the brain can often compensate with new learning.

A portion of a network composed of two cells is shown in Figure 6.2. The cell itself includes a **nucleus** (the central processing portion of the neuron). To the left of cell 1, the dendrites provide input signals to the cell. To the right, the axon sends output signals to cell 2 via the axon terminals. These axon terminals merge with the dendrites of cell 2. Signals can be transmitted unchanged, or they can be altered by synapses. A **synapse** is able to increase or decrease the strength of the connection between neurons and cause excitation or inhibition of a subsequent neuron. This is how information is stored in the neural networks.

An ANN emulates a biological neural network. Neural computing actually uses a very limited set of concepts from biological neural systems (see Technology Insights 6.1). It is more of an analogy to the human brain than an accurate model of it. Neural concepts usually are implemented as software simulations of the massively parallel processes involved in processing interconnected elements (also called artificial neurons, or *neurodes*) in a network architecture. The artificial neuron receives inputs analogous to the electrochemical impulses that dendrites of biological neurons receive from other neurons. The output of the artificial neuron corresponds to signals sent from a biological neuron over its axon. These artificial signals can be changed by weights in a manner similar to the physical changes that occur in the synapses (see Figure 6.3).

Several ANN paradigms have been proposed for applications in a variety of problem domains. Perhaps the easiest way to differentiate among the various neural models is on the basis of how they structurally emulate the human brain, the way they process information, and how they learn to perform their designated tasks.

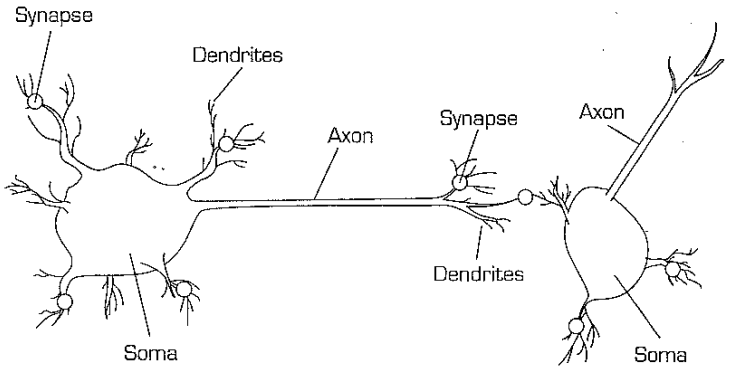


FIGURE 6.2 Portion of a Biological Neural Network: Two Interconnected Cells/Neurons

TECHNOLOGY INSIGHTS 6.1 The Relationship Between Biological and Artificial Neural Networks

The following list shows some of the relationships between biological and artificial networks.

Biological	Artificial
Soma	Node
Dendrites	Input
Axon	Output
Synapse	Weight
Slow	Fast
Many neurons (10^9)	Few neurons (a dozen to hundreds of thousands)

Sources: L. Medsker and J. Liebowitz, *Design and Development of Expert Systems and Neural Networks*, Macmillan, New York, 1994, p. 163; and F. Zahedi, *Intelligent Systems for Business: Expert Systems with Neural Networks*, Wadsworth, Belmont, CA, 1993.

Because they are biologically inspired, the main processing elements of a neural network are individual neurons, analogous to the brain's neurons. These artificial neurons receive the information from other neurons or external input stimuli, perform a transformation on the inputs, and then pass on the transformed information to other neurons or external outputs. This is similar to how it is presently thought that the human brain works. Passing information from neuron to neuron can be thought of as a way to activate, or trigger, a response from certain neurons based on the information or stimulus received.

Zahedi (1993) explored a dual role for ANN. One role is to borrow concepts from the biological world to improve the design of computers. ANN technology is used for complex information processing and machine intelligence. A second role is for neural networks to be used as simple biological models to test hypotheses about "real" biological neuronal information-processing systems. Of course, in the context of data mining and business analytics, we are interested in the use of neural networks for machine learning and information processing.

How information is processed by a neural network is inherently a function of its structure. Neural networks can have one or more layers of neurons. These neurons can be highly or fully interconnected, or only certain layers can be connected. Connections between neurons have an associated weight. In essence, the "knowledge" possessed by the network

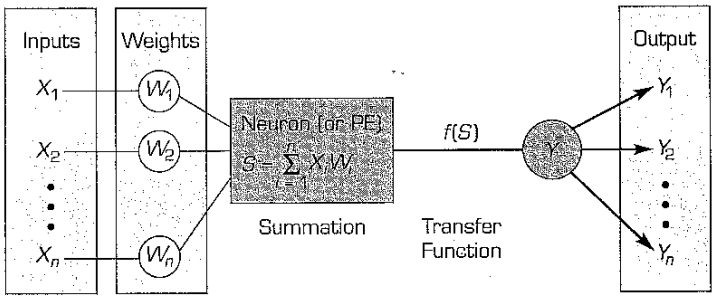


FIGURE 6.3 Processing Information in an Artificial Neuron

is encapsulated in these interconnection weights. Each neuron calculates a weighted sum of the incoming neuron values, transforms this input, and passes on its neural value as the input to subsequent neurons. Typically, although not always, this input/output transformation process at the individual neuron level is performed in a nonlinear fashion.

APPLICATION CASE 6.1

Neural Networks Help Reduce Telecommunications Fraud

The Forum of International Irregular Network Access (FIINA) estimates that telecommunications fraud results in a loss of $55 billion per year worldwide. South Africa's largest telecommunications operator was losing over $37 million per year to fraud. Subscription fraud—in which a customer provides fraudulent details or gives valid details and then disappears—was the company's biggest cause of revenue leakage. By the time the telecommunications provider is alerted to the fraud, the fraudster has already moved on to other victims. Other types of fraud include phone card manipulation, which involves tampering and cloning phone cards, and clip-on fraud, whereby a fraudster clips on to customers' telephone lines and then sell calls to overseas destinations for a fraction of normal rates.

Minotaur, developed by Neural Technologies (**neuralt.com**), was implemented to prevent fraud. Minotaur uses a hybrid mixture of intelligent systems and traditional computing techniques to provide customer subscription and real-time call-monitoring fraud detection. It processes data from numerous fields, such as event data records (e.g., switch/CDR, SS#7, IPDRs, PIN/authentication) and customer data (e.g., billing and payment, point of sale, provisioning), using a multistream analysis capacity. Frauds are detected on several levels, such as on an individual basis by using specific knowledge about the subscriber's usage, and on a global basis, using generic knowledge about subscriber usage and known fraud patterns. Minotaur's neural capability means it learns from experience, making use of adaptive feedback to keep up-to-date with changing fraud patterns. A combination of call/network data and subscriber information is profiled and then processed using intelligent neural, rule-based, and case-based techniques. Probable frauds are identified, collected into cases, and tracked to completion by means of a powerful and flexible workflow-based operational process.

In the first 3 months following installation of Minotaur:

- The average fraud loss per case was reduced by 40 percent.
- Fraud detection time was reduced by 83 percent.
- The average time taken to analyze suspected fraud cases was reduced by 75 percent.
- The average detection hit rate was improved by 74 percent.

The combination of neural, rule-based, and case-based technologies provides a fraud detection rate superior to that of conventional systems. Furthermore, the multistream analysis capability makes it extremely accurate.

Sources: "Combating Fraud: How a Leading Telecom Company Solved a Growing Problem," **neuralt.com/iqs/dlsfa.list/dlcpti.7/downloads.html** (accessed February 2009); P. A. Estévez, M. H. Claudio, and C. A. Perez, "Prevention in Telecommunications Using Fuzzy Rules and Neural Networks," **cec.uchile.cl/~pestevez/RI0.pdf** (accessed May 2009).

Elements of ANN

A neural network is composed of **processing elements** that are organized in different ways to form the network's structure. The basic processing unit is the neuron. A number of neurons are then organized into a network. Neurons can be organized in a number of different ways; these various network patterns are referred to as *topologies*. One popular approach, known as the feedforward-backpropagation paradigm (or simply **backpropagation**), allows all neurons to link the output in one layer to the input of the next layer, but it does not allow any feedback linkage (Haykin, 2009). Backpropagation is the most commonly used network paradigm.

PROCESSING ELEMENTS The processing elements (PE) of an ANN are artificial neurons. Each neuron receives inputs, processes them, and delivers a single output, as shown in Figure 6.3. The input can be raw input data or the output of other processing elements. The output can be the final result (e.g., 1 means yes, 0 means no), or it can be input to other neurons.

NETWORK STRUCTURE Each ANN is composed of a collection of neurons that are grouped into layers. A typical structure is shown in Figure 6.4. Note the three layers: input, intermediate (called the hidden layer), and output. A **hidden layer** is a layer of neurons that takes input from the previous layer and converts those inputs into outputs for further processing. Several hidden layers can be placed between the input and output layers, although it is common to use only one hidden layer. In that case, the hidden layer simply converts inputs into a nonlinear combination and passes the transformed inputs to the output layer. The most common interpretation of the hidden layer is as a feature-extraction mechanism; that is, the hidden layer converts the original inputs in the problem into a higher-level combination of such inputs.

Like a biological network, an ANN can be organized in several different ways (i.e., topologies or architectures); that is, the neurons can be interconnected in different ways. When information is processed, many of the processing elements perform their computations at the same time. This **parallel processing** resembles the way the brain works, and it differs from the serial processing of conventional computing.

Network Information Processing

Once the structure of a neural network is determined, information can be processed. We now present the major concepts related to network information processing.

INPUT Each input corresponds to a single attribute. For example, if the problem is to decide on approval or disapproval of a loan, attributes could include the applicant's income level, age, and home ownership status. The numeric value, or representation, of an attribute is the input to the network. Several types of data, such as text, pictures, and voice, can be used as inputs. Preprocessing may be needed to convert the data into meaningful inputs from symbolic data or to scale the data.

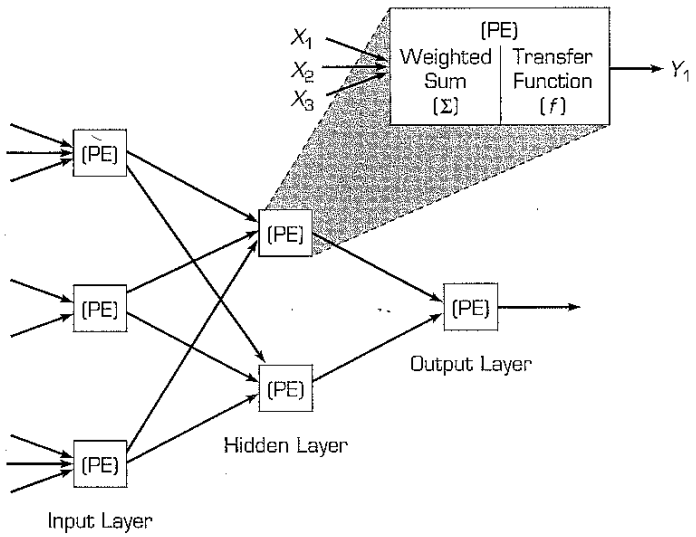


FIGURE 6.4 Neural Network with One Hidden Layer

OUTPUTS The output of a network contains the solution to a problem. For example, in the case of a loan application, the output can be *yes* or *no*. The ANN assigns numeric values to the output, such as 1 for "yes" and 0 for "no." The purpose of the network is to compute the output values. Often, postprocessing of the output is required because some networks use two outputs: one for "yes" and another for "no." It is common to round the outputs to the nearest 0 or 1.

CONNECTION WEIGHTS **Connection weights** are the key elements of an ANN. They express the relative strength (or mathematical value) of the input data or the many connections that transfer data from layer to layer. In other words, weights express the relative importance of each input to a processing element and, ultimately, the output. Weights are crucial in that they store learned patterns of information. It is through repeated adjustments of weights that a network learns.

SUMMATION FUNCTION The **summation function** computes the weighted sums of all the input elements entering each processing element. A summation function multiplies each input value by its weight and totals the values for a weighted sum Y. The formula for n inputs in one processing element (see Figure 6.5a) is:

$$Y = \sum_{i=1}^{n} X_i W_i$$

For the jth neuron of several processing neurons in a layer (see Figure 6.5b), the formula is:

$$Y_j = \sum_{i=1}^{n} X_i W_{ij}$$

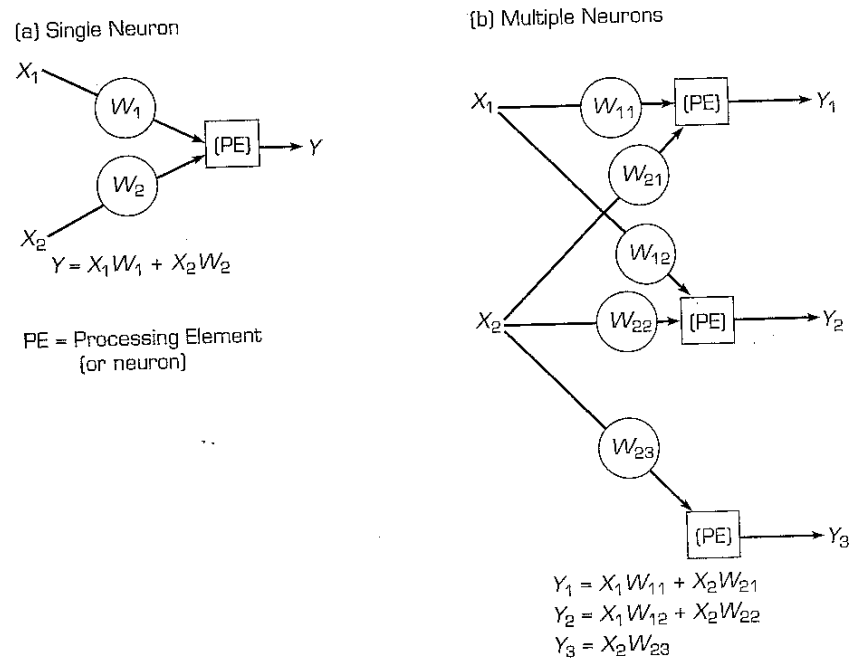


FIGURE 6.5 Summation Function for (a) a Single Neuron and (b) Several Neurons

TRANSFORMATION (TRANSFER) FUNCTION The summation function computes the internal stimulation, or activation level, of the neuron. Based on this level, the neuron may or may not produce an output. The relationship between the internal activation level and the output can be linear or nonlinear. The relationship is expressed by one of several types of **transformation (transfer) functions**. The transformation function combines (i.e., adds up) the inputs coming into a neuron from other neurons/sources and then produces an output based on the transformation function. Selection of the specific function affects the network's operation. The **sigmoid (logical activation) function** (or *sigmoid transfer function*) is an S-shaped transfer function in the range of 0 to 1, and it is a popular as well as useful nonlinear transfer function:

$$Y_T = \frac{1}{\left(1 + e^{-Y}\right)}$$

where Y_T is the transformed (i.e., normalized) value of Y (see Figure 6.6).

The transformation modifies the output levels to reasonable values (typically between 0 and 1). This transformation is performed before the output reaches the next level. Without such a transformation, the value of the output becomes very large, especially when there are several layers of neurons. Sometimes a threshold value is used instead of a transformation function. A **threshold value** is a hurdle value for the output of a neuron to trigger the next level of neurons. If an output value is smaller than the threshold value, it will not be passed to the next level of neurons. For example, any value of 0.5 or less becomes 0, and any value above 0.5 becomes 1. A transformation can occur at the output of each processing element, or it can be performed only at the final output nodes.

HIDDEN LAYERS Complex practical applications require one or more hidden layers between the input and output neurons and a correspondingly large number of weights. Many commercial ANN include three and sometimes up to five layers, with each containing 10 to 1,000 processing elements. Some experimental ANN use millions of processing elements. Because each layer increases the training effort exponentially and also increases the computation required, the use of more than three hidden layers is rare in most commercial systems.

Neural Network Architectures

There are several neural network architectures (models and/or algorithms; see Haykin, 2009). The most common ones include feedforward (with backpropagation), associative

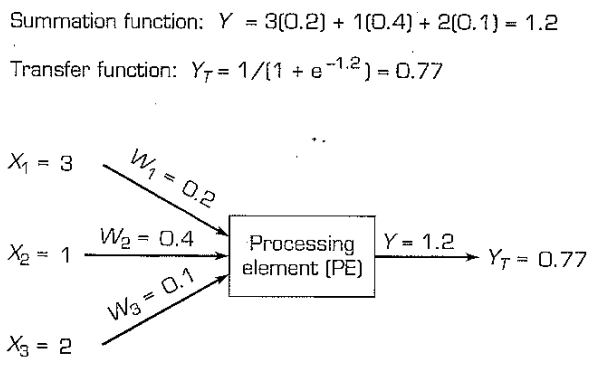


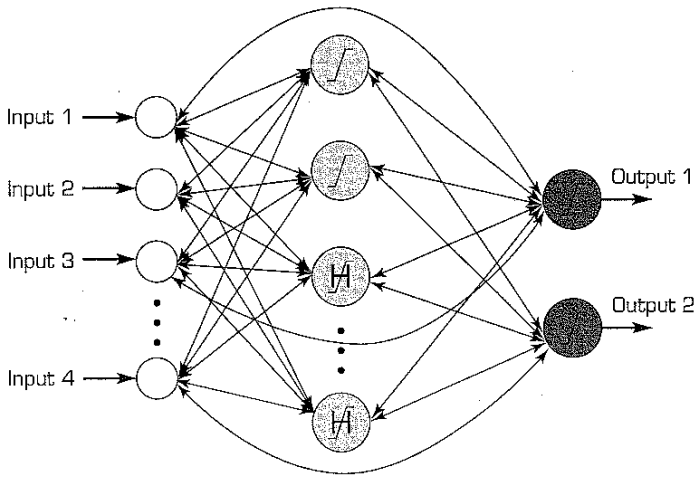
FIGURE 6.6 Example of ANN Transfer Function

memory, recurrent networks, Kohonen's self-organizing feature maps, and Hopfield networks. The feedforward network architecture (with backpropagation) is shown in Figure 6.4. Figure 6.7 shows a pictorial representation of a recurrent neural network architecture. Notice that in this architecture the connections are not unidirectional; there are many connections in every direction between the neurons, creating a chaotic-looking connection structure, which some experts believe better mimics the way biological neurons are structured in the human brain. Some of the other network architectures will be shown and briefly explained later in the chapter.

Ultimately, the architecture of a neural network model is driven by the task it is intended to address. For instance, neural network models have been used as classifiers, as forecasting tools, and as general optimizers. As shown later in this chapter, neural network classifiers are typically multilayer models in which information is passed from one layer to the next, with the ultimate goal of mapping an input to the network to a specific category, as identified by an output of the network. A neural model used as an optimizer, in contrast, can be a single layer of neurons, highly interconnected, and can compute neuron values iteratively until the model converges to a stable state. This stable state represents an optimal solution to the problem under analysis.

Finally, how a network is trained to perform its desired task is another identifying model characteristic. Neural network learning can occur in either a supervised or an unsupervised mode. With **supervised learning**, a sample training set is used to "teach" the network about its problem domain. This training set of exemplar cases (input and the desired output) is iteratively presented to the neural network. The output of the network in its present form is calculated and compared to the desired output. The **learning algorithm** is the training procedure that an ANN uses. The learning algorithm used determines how the neural interconnection weights are corrected due to differences in the actual and desired output for a member of the training set. Updating of the network's interconnection weights continues until the training algorithm's stopping criteria are met (e.g., all cases must be correctly classified within a certain tolerance level).

Alternatively, with **unsupervised learning** the network does not try to learn a target answer. Instead, the neural network learns a pattern through repeated exposures.



*H indicates a "hidden" neuron without a target output

FIGURE 6.7 A Recurrent Neural Network Architecture

This kind of learning can be envisioned as a neural network self-organizing or clustering its neurons related to the specific task.

Multilayer, feedforward neural networks are a class of models that show promise in classification and forecasting problems. As the name implies, these models structurally consist of multiple layers of neurons. Information is passed through the network in one direction, from the input layers of the network, through one or more hidden layers, toward the output layer of neurons. Neurons of each layer are connected only to the neurons of the subsequent layer.

Section 6.2 Review Questions

1. What is an ANN?
2. Explain the following terms: *neuron, axon,* and *synapse.*
3. How do weights function in an ANN?
4. What is the role of the summation function?
5. What is the role of the transformation function?

6.3 LEARNING IN ARTIFICIAL NEURAL NETWORKS

An important consideration in an ANN is the use of an appropriate learning algorithm (or training algorithm). Learning algorithms specify the process by which a neural network learns the underlying relationship between inputs and outputs, or just among the inputs. Hundreds of learning algorithms have been developed. ANN learning algorithms can also be classified as supervised and unsupervised (see Figure 6.8).

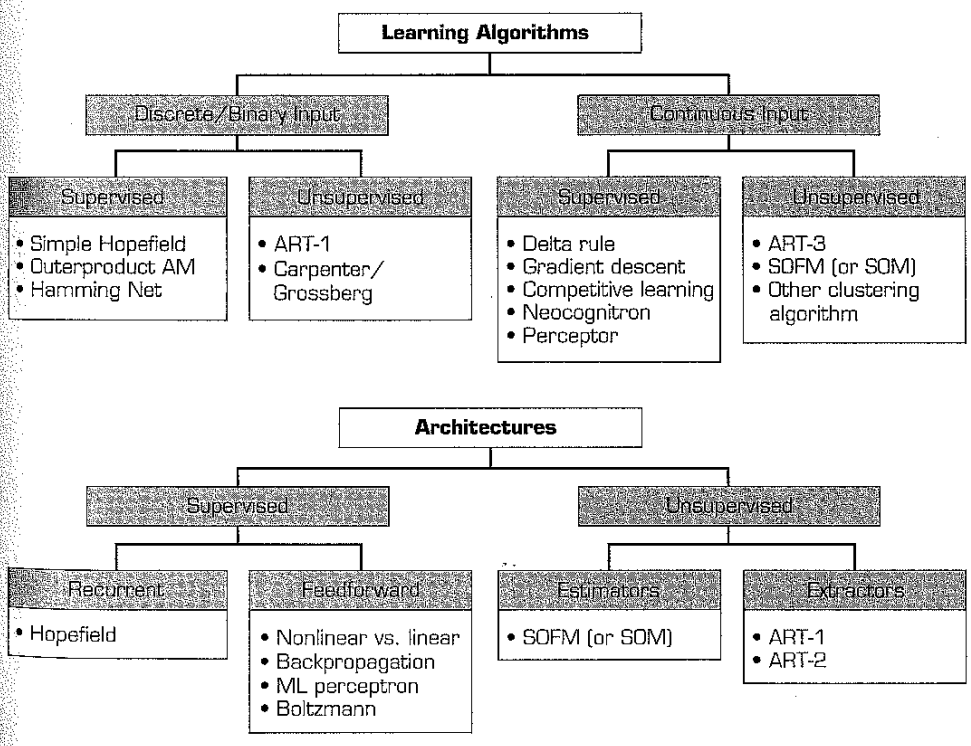


FIGURE 6.8 Taxonomy of ANN Learning Algorithms and Architectures *Source:* Based on L. Medsker and J. Liebowitz, *Design and Development of Expert Systems and Neural Computing,* Macmillan, New York, 1994, p. 166.

Supervised learning uses a set of inputs for which the appropriate (i.e., desired) outputs are known. For example, a dataset of loan applications with the success or failure of borrowers to repay their loans has a set of input parameters and presumed known outputs. With one type of supervised learning, the difference between the desired and actual outputs is used to correct the weights of the neural network. A variation of this approach simply acknowledges for each input trial whether the output is correct as the network adjusts weights in an attempt to achieve the correct results. Examples of this type of learning are backpropagation and the Hopfield network (Hopfield, 1982). Application Case 6.2 describes an application of supervised learning at Microsoft for improving the response rate of target mailings to potential customers.

APPLICATION CASE 6.2

Neural Networks Help Deliver Microsoft's Mail to the Intended Audience

Microsoft, the world leader in computer software, used BrainMaker neural network software from California Scientific (**calsci.com**) to maximize returns on direct mail. Every year, Microsoft sends approximately 40 million pieces of direct mail to 8.5 million registered customers, encouraging them to upgrade their software or to buy related products. Generally, the first mailing includes everyone in the database. The key is to direct the second mailing only to those who are most likely to respond.

Several variables were fed into the BrainMaker neural network to get productive results. The first step was to identify the variables that were relevant and to eliminate the variables that did not cause any effect. The following were some of the significant variables:

- *Recency.* Calculated in number of days, this is the last time a customer bought and registered a product. It is likely that the more recently a customer has bought something, the more likely it is that he or she will buy again the same or similar product.
- *First date to file.* This is the date of a customer's initial purchase and is a measure of loyalty. Chances are high that a customer will buy again if he or she has been loyal over time.
- *The number of products bought and registered.* This is the total number of products a customer has bought and registered.
- *The value of the products bought and registered.* This is calculated at the standard reselling price.
- *The number of days between the time the product came out and when it was purchased.* Research has shown that people who tend to buy things as soon as they are available are the key individuals to be reached.

Several other personal characteristics were also added and scored with yes/no responses.

Data was collected from seven or eight campaigns so that it was varied and represented all aspects of the business, including both Mac and Windows and high- and low-priced products. The customer-response information was converted to a format that the network could use, and yes/no responses were transformed to numeric data. Minimums and maximums were set on certain variables. Initially, the network was trained with 25 variables.

The neural network was tested on data from 20 different campaigns with known results not used during training. The results showed repeated and consistent savings. The use of BrainMaker to target customers on an average mailing resulted in a 35 percent cost savings for Microsoft. Before Microsoft began using BrainMaker, an average mailing had a response rate of 4.9 percent. With BrainMaker, the response rate to direct mailings increased to 8.2 percent.

Sources: California Scientific, "Maximize Returns on Direct Mail with BrainMaker Neural Networks Software," **calsci.com/DirectMail.html** (accessed August 2009); and G. Piatesky-Shapiro, "ISR: Microsoft Success Using Neural Network for Direct Marketing," **kdnuggets.com/news/94/n9.txt** (accessed May 2009).

With unsupervised learning, only input stimuli are shown to the network. The network is **self-organizing**; that is, it organizes itself internally so that each hidden processing element responds strategically to a different set of input stimuli (or groups of stimuli). No knowledge is supplied about which classifications (i.e., outputs) are correct, and those that the network derives may or may not be meaningful to the network developer (this is useful for cluster analysis). However, the number of categories into which a network classifies the inputs can be controlled by setting model parameters. A person must examine the final categories to assign meaning and determine the usefulness of the results. Examples of this type of learning are **adaptive resonance theory (ART)** (i.e., a neural network architecture that is aimed at being brainlike in unsupervised mode) and Kohonen's self-organizing feature maps (i.e., neural network models for machine learning).

As mentioned earlier, many different and distinct neural network paradigms have been proposed for various decision-making domains. A neural model that has been shown appropriate for classification problems (e.g., bankruptcy prediction) is the feedforward multilayered perceptron. Multilayered networks have continuously valued neurons (i.e., processing elements), are trained in a supervised manner, and consist of one or more layers of nodes (i.e., hidden nodes) between the input and output nodes. A typical feedforward neural network is shown in Figure 6.4. Input nodes represent where information is presented to the network, output nodes provide the neural network's "decision," and the hidden nodes via the interconnection weights contain the proper mapping of inputs to outputs (i.e., decisions).

The backpropagation learning algorithm is the standard way of implementing supervised training of feedforward neural networks. It is an iterative gradient-descent technique designed to minimize an error function between the actual output of the network and its desired output, as specified in the training dataset. Adjustment of the interconnection weights, which contain the mapping function per se, starts at the output node where the error measure is initially calculated and is then propagated back through the layers of the network, toward the input layer. More details are included in the following section.

The General ANN Learning Process

In supervised learning, the learning process is inductive; that is, connection weights are derived from existing cases. The usual process of learning involves three tasks (see Figure 6.9):

1. Compute temporary outputs.
2. Compare outputs with desired targets.
3. Adjust the weights and repeat the process.

When existing outputs are available for comparison, the learning process starts by setting the connection weights. These are set via rules or at random. The difference between the actual output (Y or Y_T) and the desired output (Z) for a given set of inputs is an error called delta (in calculus, the Greek symbol delta, Δ, means "difference").

The objective is to minimize delta (i.e., reduce it to 0 if possible), which is done by adjusting the network's weights. The key is to change the weights in the right direction, making changes that reduce delta (i.e., error). We will show how this is done later.

Information processing with an ANN consists of attempting to recognize patterns of activities (i.e., pattern recognition). During the learning stages, the interconnection weights change in response to training data presented to the system.

Different ANN compute delta in different ways, depending on the learning algorithm being used. Hundreds of learning algorithms are available for various situations and configurations, some of which are discussed later in this chapter.

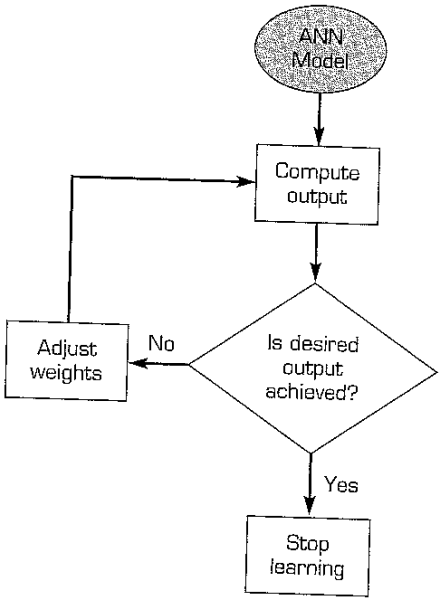


FIGURE 6.9 Supervised Learning Process of an ANN

How a Network Learns

Consider a single neuron that learns the inclusive OR operation—a classic problem in symbolic logic. The two input elements are X_1 and X_2. If either or both of them have a positive value, the result is also positive. This can be shown as follows:

	Inputs		
Case	X_1	X_2	Desired Results
1	0	0	0
2	0	1	1 (positive)
3	1	0	1 (positive)
4	1	1	1 (positive)

The neuron must be trained to recognize the input patterns and classify them to give the corresponding outputs. The procedure is to present the sequence of the four input patterns to the neuron so that the weights are adjusted after each iteration (using feedback of the error found by comparing the estimate to the desired result). This step is repeated until the weights converge to a uniform set of values that allows the neuron to classify each of the four inputs correctly. The results shown in Table 6.1 were produced in Excel. In this simple example, a threshold function is used to evaluate the summation of input values. After calculating outputs, a measure of the error (i.e., delta) between the output and the desired values is used to update the weights, subsequently reinforcing the correct results. At any step in the process, for a neuron j we have:

$$\text{delta} = Z_j - Y_j$$

TABLE 6.1 Example of Supervised Learning[a]

				Initial Weights				Final Weights	
Step	X_1	X_2	Z	W_1	W_2	Y	Delta	W_1	W_2
1	0	0	0	0.1	0.3	0	0.0	0.1	0.3
	0	1	1	0.1	0.3	0	1.0	0.1	0.5
	1	0	1	0.1	0.5	0	1.0	0.3	0.5
	1	1	1	0.3	0.5	1	0.0	0.3	0.5
2	0	0	0	0.3	0.5	0	0.0	0.3	0.5
	0	1	1	0.3	0.5	0	0.0	0.3	0.7
	1	0	1	0.3	0.7	0	1.0	0.5	0.7
	1	1	1	0.5	0.7	1	0.0	0.5	0.7
3	0	0	0	0.5	0.7	0	0.0	0.5	0.7
	0	1	1	0.5	0.7	1	0.0	0.5	0.7
	1	0	1	0.5	0.7	0	1.0	0.7	0.7
	1	1	1	0.7	0.7	1	0.0	0.7	0.7
4	0	0	0	0.7	0.7	0	0.0	0.7	0.7
	0	1	1	0.7	0.7	1	0.0	0.7	0.7
	1	0	1	0.7	0.7	1	0.0	0.7	0.7
	1	1	1	0.7	0.7	1	0.0	0.7	0.7

[a] Parameters: alpha = 0.2; threshold = 0.5; output is zero if the sum ($W_1 * X_1 + W_2 * X_2$) is not greater than 0.5.

where Z and Y are the desired and actual outputs, respectively. Then, the updated weights are:

$$W_i(\text{final}) = W_i(\text{initial}) + \text{alpha} \times \text{delta} \times X_i$$

where alpha is a parameter that controls how fast the learning takes place. This is called a **learning rate**. The choice of the learning rate parameter can have an impact on how fast (and how correctly) a neural network learns. A high value for the learning rate can lead to too much correction in the weight values, which causes the algorithm to just go back and forth among possible weight values, never reaching the optimal values, which may lie somewhere in between the endpoints. Too low a learning rate may slow the learning process and may lead to sub-optimal weight values. In practice, a neural network analyst will try many different learning rates to achieve the optimal learning.

Most implementations of the learning process also include a counterbalancing parameter called **momentum** to balance the learning rate. Essentially, whereas the purpose of the learning rate is to correct for the error, momentum is aimed at slowing the learning process. Many of the software programs available for neural networks can automatically select these parameters for the user or let the user experiment with many different combinations of such parameters.

As shown in Table 6.1, each calculation uses one of the X_1 and X_2 pairs and the corresponding value for the OR operation, along with the initial values, W_1 and W_2, of the neuron's weights. Initially, the weights are assigned random values, and the learning rate, alpha, is set low. Delta is used to derive the final weights, which then become the initial weights in the next iteration (i.e., row).

The initial values of weights for each input are transformed using the equation shown earlier to assign the values to the next input (i.e., row). The threshold value (0.5) sets the output Y to 1 in the next row if the weighted sum of inputs is greater than 0.5; otherwise, Y is set to 0. In the first step, two of the four outputs are incorrect (delta = 1), and a consistent set of weights has not been found. In subsequent steps, the learning algorithm improves the results until it finally produces a set of weights that give the correct results ($W_1 = W_2 = 0.7$ in step 4 of Table 6.1). Once determined, a neuron with these weights can quickly perform the OR operation.

In developing an ANN, an attempt is made to fit the problem characteristic to one of the known learning algorithms. Many variants of learning algorithms exist, but the core concepts behind all of them are similar.

Backpropagation

Backpropagation (short for *back-error propagation*) is the most widely used supervised learning algorithm in neural computing (Principe et al., 2000). It is very easy to implement. A backpropagation network includes one or more hidden layers. This type of network is considered feedforward because there are no interconnections between the output of a processing element and the input of a node in the same layer or in a preceding layer. Externally provided correct patterns are compared with the neural network's output during (supervised) training, and feedback is used to adjust the weights until the network has categorized all the training patterns as correctly as possible (the error tolerance is set in advance).

Starting with the output layer, errors between the actual and desired outputs are used to correct the weights for the connections to the previous layer (see Figure 6.10). For any output neuron j, the error (delta) = $(Z_j - Y_j)$ (df/dx), where Z and Y are the desired and actual outputs, respectively. Using the sigmoid function, $f = [1 + \exp(-x)]^{-1}$, where x is proportional to the sum of the weighted inputs to the neuron, is an effective way to compute the output of a neuron in practice. With this function, the derivative of the sigmoid function $df/dx = f(1 - f)$ and the error is a simple function of the desired and actual outputs. The factor $f(1 - f)$ is the logistic function, which serves to keep the error correction well bounded. The weights of each input to the j^{th} neuron are then changed in proportion to this calculated error. A more complicated expression can be derived to work backward in a similar way from the output neurons through the hidden layers to calculate the corrections to the associated weights of the inner neurons. This complicated method is an iterative approach to solving a nonlinear optimization problem that is very similar in meaning to the one characterizing multiple-linear regression.

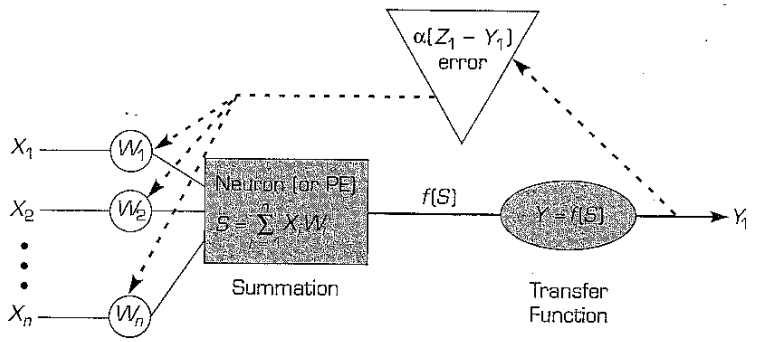


FIGURE 6.10 Backpropagation of Error for a Single Neuron

The learning algorithm includes the following procedures:

1. Initialize weights with random values and set other parameters.
2. Read in the input vector and the desired output.
3. Compute the actual output via the calculations, working forward through the layers.
4. Compute the error.
5. Change the weights by working backward from the output layer through the hidden layers.

This procedure is repeated for the entire set of input vectors until the desired output and the actual output agree within some predetermined tolerance. Given the calculation requirements for one iteration, a large network can take a very long time to train; therefore, in one variation, a set of cases is run forward and an aggregated error is fed backward to speed up learning. Sometimes, depending on the initial random weights and network parameters, the network does not converge to a satisfactory performance level. When this is the case, new random weights must be generated, and the network parameters, or even its structure, may have to be modified before another attempt is made. Current research is aimed at developing algorithms and using parallel computers to improve this process. For example, genetic algorithms (described in Chapter 13) can be used to guide the selection of the network parameters in order to maximize the desired output. In fact, most commercial ANN software tools are now using GA to help users "optimize" the network parameters.

Section 6.3 Review Questions

1. Briefly describe backpropagation.
2. What is the purpose of a threshold value in a learning algorithm?
3. What is the purpose of a learning rate and momentum?
4. How does error between actual and predicted outcomes affect the value of weights in neural networks?
5. Search the Internet to identify other learning algorithms for feedforward neural networks.

6.4 DEVELOPING NEURAL NETWORK–BASED SYSTEMS

Although the development process of ANN is similar to the structured design methodologies of traditional computer-based information systems, some phases are unique or have some unique aspects. In the process described here, we assume that the preliminary steps of system development, such as determining information requirements, conducting a feasibility analysis, and gaining a champion in top management for the project, have been completed successfully. Such steps are generic to any information system.

As shown in Figure 6.11, the development process for an ANN application includes nine steps. In step 1, the data to be used for training and testing the network are collected. Important considerations are that the particular problem is amenable to a neural network solution and that adequate data exist and can be obtained. In step 2, training data must be identified, and a plan must be made for testing the performance of the network.

In steps 3 and 4, a network architecture and a learning method are selected. The availability of a particular development tool or the capabilities of the development personnel may determine the type of neural network to be constructed. Also, certain problem types have demonstrated high success rates with certain configurations (e.g., multilayer feedforward neural networks for bankruptcy prediction, as described in the next section). Important considerations are the exact number of neurons and the number of layers. Some packages use genetic algorithms to select the network design.

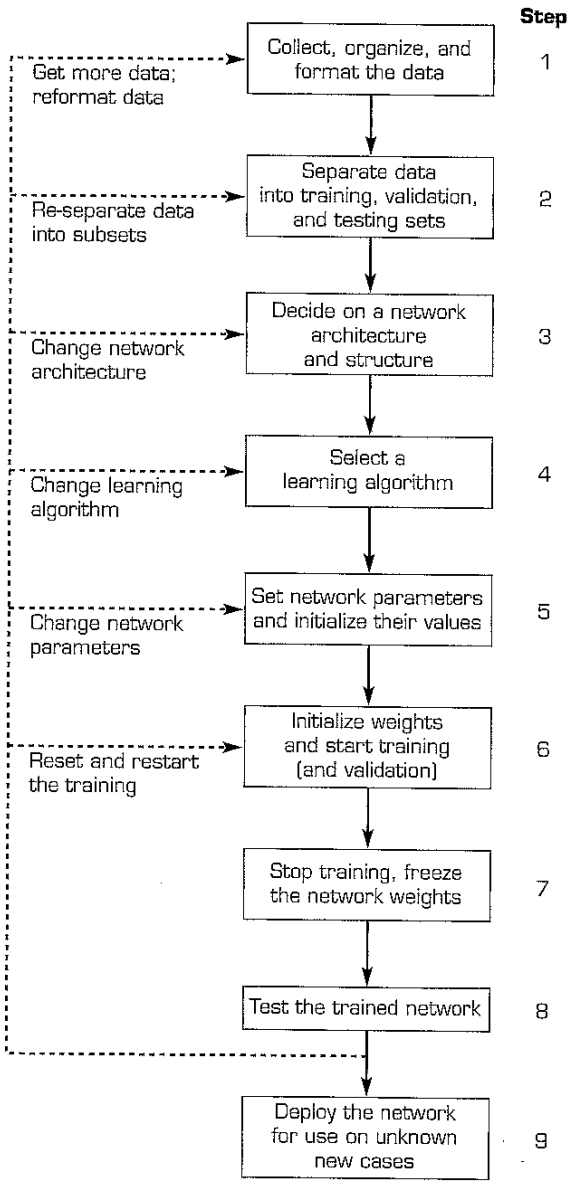


FIGURE 6.11 ANN Development Process

There are several parameters for tuning the network to the desired learning-performance level. Part of the process in step 5 is the initialization of the network weights and parameters, followed by the modification of the parameters as training-performance feedback is received. Often, the initial values are important in determining the efficiency and length of training. Some methods change the parameters during training to enhance performance.

Step 6 transforms the application data into the type and format required by the neural network. This may require writing software to preprocess the data or performing these operations directly in an ANN package. Data storage and manipulation techniques and processes must be designed for conveniently and efficiently retraining the neural

network, when needed. The application data representation and ordering often influence the efficiency and possibly the accuracy of the results.

In steps 7 and 8, training and testing are conducted iteratively by presenting input and desired or known output data to the network. The network computes the outputs and adjusts the weights until the computed outputs are within an acceptable tolerance of the known outputs for the input cases. The desired outputs and their relationships to input data are derived from historical data (i.e., a portion of the data collected in step 1).

In step 9, a stable set of weights is obtained. Now the network can reproduce the desired outputs, given inputs such as those in the training set. The network is ready for use as a stand-alone system or as part of another software system where new input data will be presented to it and its output will be a recommended decision.

In the following sections, we examine these steps in more detail.

Data Collection and Preparation

The first two steps in the ANN development process involve collecting data and separating them into a training set and a testing set. The training cases are used to adjust the weights, and the testing cases are used for network validation. The data used for training and testing must include all the attributes that are useful for solving the problem. The system can only learn as much as the data can tell. Therefore, collection and preparation of data are the most critical steps in building a good system.

In general, the more data used the better. Larger datasets increase processing time during training but improve the accuracy of the training and often lead to faster convergence to a good set of weights. For a moderately sized dataset, typically 80 percent of the data are randomly selected for training and 20 percent are selected for testing. For small datasets, typically all the data are used for training and testing. For large datasets, a sufficiently large sample is taken and treated like a moderately sized dataset.

For example, say a bank wants to build a neural network–based system in order to use clients' financial data to determine whether they may go bankrupt. The bank needs to first identify what financial data may be used as inputs and how to obtain them. Five attributes may be useful inputs: (1) working capital/total assets, (2) retained earnings/total assets, (3) earnings before interest and taxes/total assets, (4) market value of equity/total debt, and (5) sales/total sales. The output is a binary variable: bankruptcy or not.

Selection of Network Structure

After the training and testing datasets are identified, the next step is to design the structure of the neural networks. This includes the selection of a **topology** and determination of (1) input nodes, (2) output nodes, (3) number of hidden layers, and (4) number of hidden nodes. The multilayer feedforward topology is often used in business applications, although other network models are beginning to find some business use as well.

The design of input nodes must be based on the attributes of the dataset. In the example of predicting bankruptcy, for example, the bank might choose a three-layer structure that includes one input layer, one output layer, and one hidden layer. The input layer contains five nodes, each of which is a variable, and the output layer contains a node with 0 for bankrupt and 1 for safe. Determining the number of hidden nodes is tricky. A few heuristics have been proposed, but none of them is unquestionably the best. A typical approach is to choose the average number of input and output nodes. In the previous case, the hidden node may be set to $(5 + 1)/2 = 3$. Figure 6.12 shows an MLP ANN structure for the box-office prediction problem.

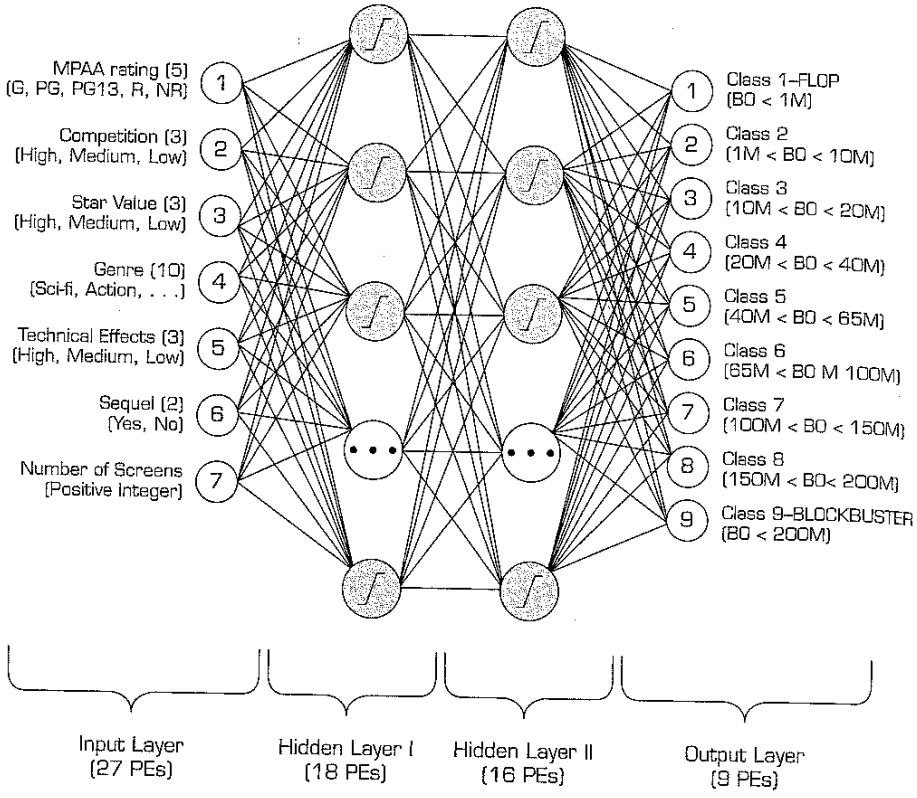


FIGURE 6.12 MLP ANN Structure for the Box-Office Prediction Problem

Learning Algorithm Selection

After the network structure is chosen, we need to find a learning algorithm to identify a set of connection weights that best cover the training data and have the best predictive accuracy. For the feedforward topology we chose for the bankruptcy-prediction problem, a typical approach is to use the backpropagation algorithm. Because many commercial packages are available on the market, there is no need to implement the learning algorithm by ourselves. Instead, we can choose a suitable commercial package to analyze the data. Technology Insights 6.2 summarizes information on the different types of neural network software packages that are available.

Network Training

Training of ANN is an iterative process that starts from a random set of weights and gradually enhances the fitness of the network model and the known dataset. The iteration continues until the error sum is converged to below a preset acceptable level. In the backpropagation algorithm, two parameters, learning rate and momentum, can be adjusted to control the speed of reaching a solution. These determine the ratio of the difference between the calculated value and the actual value of the training cases. Some software packages may have their own parameters in their learning heuristics to speed up the learning process. It is important to read carefully when using this type of software.

How are neural networks implemented in practice? After the analyst/developer has conducted enough tests to ascertain that a neural network can do a good job for the application, the network needs to be implemented in the existing systems. A number of

TECHNOLOGY INSIGHTS 6.2 ANN Software

Many tools are available for developing neural networks (see this book's Web site and the resource lists at PC AI, **pcai.com**). Some of these tools function like expert system shells. They provide a set of standard architectures, learning algorithms, and parameters, along with the ability to manipulate the data. Some development tools can support up to several dozen network paradigms and learning algorithms.

Neural network implementations are also available in most of the comprehensive data mining tools, such as the SAS Enterprise Miner, PASW Modeler (formerly Clementine), and Statistica Data Miner. Weka is an open source collection of machine-learning algorithms for data mining tasks, and it includes neural network capabilities. Weka can be downloaded from **cs.waikato.ac.nz/~ml/weka**. Statistica is available on a trial basis to adopters of this book.

Many specialized neural network tools enable the building and deployment of a neural network model in practice. Any listing of such tools would be incomplete. Online resources such as Wikipedia (**en.wikipedia.org/wiki/Artificial_neural_network**), Google's or Yahoo!'s software directory, and the vendor listings on **pcai.com** are good places to locate the latest information on neural network software vendors. Some of the vendors that have been around for a while and have reported industrial applications of their neural network software include California Scientific (BrainMaker), NeuralWare, NeuroDimension Inc., Ward Systems Group (Neuroshell), and Megaputer. Again, the list can never be complete.

Some ANN development tools are spreadsheet add-ins. Most can read spreadsheet, database, and text files. Some are freeware or shareware. Some ANN systems have been developed in Java to run directly on the Web and are accessible through a Web browser interface. Other ANN products are designed to interface with expert systems as hybrid development products.

Developers may instead prefer to use more general programming languages, such as C++, or a spreadsheet to program the model and perform the calculations. A variation on this is to use a library of ANN routines. For example, hav.Software (**hav.com**) provides a library of C++ classes for implementing stand-alone or embedded feedforward, simple recurrent, and random-order recurrent neural networks. Computational software such as MATLAB also includes neural network–specific libraries.

neural network shells can generate code in C++, Java, or Visual Basic that can be embedded in another system that can access source data or is called directly by a graphical user interface for deployment, independently of the development system. Or, after training an ANN in a development tool, given the weights, network structure, and transfer function, one can easily develop one's own implementation in a third-generation programming language such as C++. Most of the ANN development packages as well as data mining tools can generate such code. The code can then be embedded in a stand-alone application or in a Web server application.

Some data conversion may be necessary in the training process. This includes (1) changing the data format to meet the requirements of the software, (2) normalizing the data scale to make the data more comparable, and (3) removing problematic data. When the training dataset is ready, it is loaded into the package, and the learning procedure is executed. Depending on the number of nodes and the size of the training dataset, reaching a solution may take from a few thousand to millions of iterations.

Testing

Recall that in step 2 of the development process shown in Figure 6.11 the available data are divided into training and testing datasets. When the training has been completed, it is necessary to test the network. Testing (step 8) examines the performance of the derived

network model by measuring its ability to classify the testing data correctly. **Black-box testing** (i.e., comparing test results to historical results) is the primary approach for verifying that inputs produce the appropriate outputs. Error terms can be used to compare results against known benchmark methods.

The network is generally not expected to perform perfectly (zero error is difficult, if not impossible, to attain), and only a certain level of accuracy is really required. For example, if 1 means nonbankrupt and 0 means bankrupt, then any output between 0.1 and 1 might indicate a certain likelihood of nonbankrupty. The neural network application is usually an alternative to another method that can be used as a benchmark against which to compare accuracy. For example, a statistical technique such as multiple regression or another quantitative method may be known to classify inputs correctly 50 percent of the time.

The neural network implementation often improves on this. For example, Liang (1992) reported that ANN performance was superior to the performance of multiple discriminant analysis and rule induction. Ainscough and Aronson (1999) investigated the application of neural network models in predicting retail sales, given a set of several inputs (e.g., regular price, various promotions). They compared their results to those of multiple regression and improved the adjusted R^2 (correlation coefficient) from .5 to .7. If the neural network is replacing manual operations, performance levels and speed of human processing can be the standard for deciding whether the testing phase is successful.

The test plan should include routine cases as well as potentially problematic situations. If the testing reveals large deviations, the training set must be reexamined, and the training process may have to be repeated (some "bad" data may have to be omitted from the input set).

Note that we cannot equate neural network results exactly with those found using statistical methods. For example, in stepwise linear regression, input variables are sometimes determined to be insignificant, but because of the nature of neural computing, a neural network uses them to attain higher levels of accuracy. When they are omitted from a neural network model, its performance typically suffers.

Implementation of an ANN

Implementation (i.e., deployment) of an ANN solution (step 9) often requires interfaces with other computer-based information systems and user training. Ongoing monitoring and feedback to the developers are recommended for system improvements and long-term success. It is also important to gain the confidence of users and management early in the deployment to ensure that the system is accepted and used properly.

Section 6.4 Review Questions

1. List the nine steps in conducting a neural network project.
2. What are some of the design parameters for developing a neural network?
3. Describe different types of neural network software available today.
4. How are neural networks implemented in practice when the training/testing is complete?
5. What parameters may need to be adjusted in the neural network training process?

6.5 ILLUMINATING THE BLACK BOX OF ANN WITH SENSITIVITY ANALYSIS

Neural networks have been used as an effective tool for solving highly complex real-world problems in a wide range of application areas. Even though ANN have been proven in many problem scenarios to be superior predictors and/or cluster identifiers (compared to their traditional counterparts), in some applications there exists an additional

need to know "how it does what it does." ANN are typically thought of as black boxes, capable of solving complex problems but lacking the explanation of their capabilities. This phenomenon is commonly referred to as the "black-box" syndrome.

It is important to be able to explain a model's "inner being"; such an explanation offers assurance that the network has been properly trained and will behave as desired once deployed in a business intelligence environment. Such a need to "look under the hood" might be attributable to a relatively small training set (as a result of the high cost of data acquisition) or a very high liability in case of a system error. One example of such an application is the deployment of airbags in automobiles. Here, both the cost of data acquisition (crashing cars) and the liability concerns (danger to human lives) are rather significant. Another representative example for the importance of explanation is loan-application processing. If an applicant is refused for a loan, he or she has the right to know why. Having a prediction system that does a good job on differentiating good and bad applications may not be sufficient if it does not also provide the justification of its predictions.

A variety of techniques has been proposed for analysis and evaluation of trained neural networks. These techniques provide a clear interpretation of how a neural network does what it does; that is, specifically how (and to what extent) the individual inputs factor into the generation of specific network output. Sensitivity analysis has been the front runner of the techniques proposed for shedding light into the "black-box" characterization of trained neural networks.

Sensitivity analysis is a method for extracting the cause-and-effect relationships among the inputs and the outputs of a trained neural network model. In the process of performing sensitivity analysis, the trained neural network's learning capability is disabled so that the network weights are not affected. The basic procedure behind sensitivity analysis is that the inputs to the network are systematically perturbed within the allowable value ranges and the corresponding change in the output is recorded for each and every input variable (Principe et al., 2000). Figure 6.13 shows a graphical illustration of this process. The first input is varied between its mean plus-and-minus a user-defined number of standard deviations (or for categorical variables, all of its possible values are used) while all other input variables are fixed at their respective means (or modes). The network output is computed for a user-defined number of steps above and below the mean. This process is repeated for each input. As a result, a report is generated to summarize the variation of each output with respect to the variation in each input. The generated report often contains a column plot (along with numeric values presented on the x-axis), reporting the relative sensitivity values for each input variable. A representative example of sensitivity analysis on ANN models is provided in Application Case 6.3.

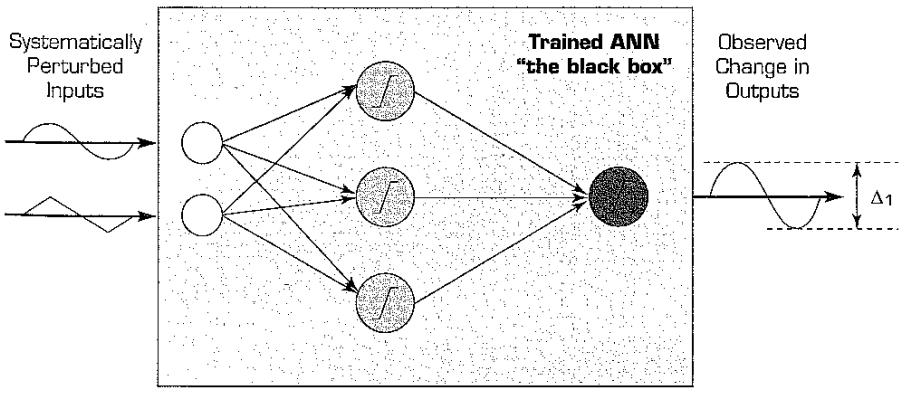


FIGURE 6.13 A Figurative Illustration of Sensitivity Analysis on an ANN Model

APPLICATION CASE 6.3

Sensitivity Analysis Reveals Injury Severity Factors in Traffic Accidents

According to the National Highway Traffic Safety Administration, over 6 million traffic accidents claim more than 41,000 lives each year in the United States. Causes of accidents and related injury severity are of special interest to traffic-safety researchers. Such research is aimed not only at reducing the number of accidents but also the severity of injury. One way to accomplish the latter is to identify the most profound factors that affect injury severity. Understanding the circumstances under which drivers and passengers are more likely to be severely injured (or killed) in an automobile accident can help improve the overall driving safety situation. Factors that potentially elevate the risk of injury severity of vehicle occupants in the event of an automotive accident include demographic and/or behavioral characteristics of the person (e.g., age, gender, seatbelt usage, use of drugs or alcohol while driving), environmental factors and/or roadway conditions at the time of the accident (e.g., surface conditions, weather or light conditions, the direction of impact, vehicle orientation in the crash, occurrence of a rollover), as well as technical characteristics of the vehicle itself (e.g., vehicle's age, body type).

In an exploratory data mining study, Delen et al. (2006) used a large sample of data—30,358 police-reported accident records obtained from the General Estimates System of the National Highway Traffic Safety Administration—to identify which factors become increasingly more important in escalating the probability of injury severity during a traffic crash. Accidents examined in this study included a geographically representative sample of multiple-vehicle collision accidents, single-vehicle fixed-object collisions, and single-vehicle noncollision (rollover) crashes.

Contrary to many of the previous studies conducted in this domain, which have primarily used regression-type generalized linear models where the functional relationships between injury severity and crash-related factors are assumed to be linear (which is an oversimplification of the reality in most real-world situations), Delen and his colleagues decided to go in a different direction. Because ANN are known to be superior in capturing highly nonlinear complex relationships between the predictor variables (crash factors) and the target variable (severity level of the injuries), they decided to use a series of ANN models to estimate the significance of the crash factors on the level of injury severity sustained by the driver.

From a methodological standpoint, they followed a two-step process. In the first step, they developed a series of prediction models (one for each injury severity level) to capture the in-depth relationships between the crash-related factors and a specific level of injury severity. In the second step, they conducted sensitivity analysis on the trained neural network models to identify the prioritized importance of crash-related factors as they relate to different injury severity levels. In the formulation of the study, the five-class prediction problem was decomposed into a number of binary classification models in order to obtain the granularity of information needed to identify the "true" cause-and-effect relationships between the crash-related factors and different levels of injury severity.

The results revealed considerable differences among the models built for different injury severity levels. This implies that the most influential factors in prediction models highly depend on the level of injury severity. For example, the study revealed that the variable seatbelt use was the most important determinant for predicting higher levels of injury severity (such as incapacitating injury or fatality), but it was one of the least significant predictors for lower levels of injury severity (such non-incapacitating injury and minor injury). Another interesting finding involved gender: The drivers' gender was among the significant predictors for lower levels of injury severity, but it was not among the significant factors for higher levels of injury severity, indicating that more serious injuries do not depend on the driver being a male or a female. Yet another interesting and somewhat intuitive finding of the study indicated that age becomes an increasingly more significant factor as the level of injury severity increases, implying that older people are more likely to incur severe injuries (and fatalities) in serious automobile crashes than younger people.

Source: D. Delen, R. Sharda, and M. Bessonov, "Identifying Significant Predictors of Injury Severity in Traffic Accidents Using a Series of Artificial Neural Networks," *Accident Analysis and Prevention,* Vol. 38, No. 3, 2006, pp. 434–444.

Review Questions for Section 6.5

1. What is the so-called "black-box" syndrome?
2. Why is it important to be able to explain an ANN's model structure?
3. How does sensitivity analysis work?
4. Search the Internet to find other ANN explanation methods.

6.6 A SAMPLE NEURAL NETWORK PROJECT

We next describe a typical application of neural networks to predict bankruptcy of companies using the same data and a similar experimental design as used by Wilson and Sharda (1994). For comparative purposes, the performance of neural networks is contrasted with logistic regression.

The Altman (1968) study has been used as the standard of comparison for many bankruptcy classification studies using discriminant analysis and logistic regression; follow-up studies have identified several other attributes to improve prediction performance. We use the same financial ratios as in Altman's study, realizing that more sophisticated inputs to the neural network model should only enhance its performance. These ratios are as follows:

X_1: Working capital/total assets

X_2: Retained earnings/total assets

X_3: Earnings before interest and taxes/total assets

X_4: Market value of equity/total debt

X_5: Sales/total assets

In step 1, we collected the relevant data. The sample, which was obtained from *Moody's Industrial Manuals,* consisted of firms that either were in operation or went bankrupt between 1975 and 1982. The sample included 129 firms, 65 of which went bankrupt during the period and 64 nonbankrupt firms matched on industry and year. Data for the bankrupt firms was obtained from the final financial statements issued before bankruptcy. Thus, the prediction of bankruptcy was to be made about 1 year in advance.

In step 2, we divided the dataset into a training set and a testing set. Because the determination of the split may affect experimental findings, a resampling procedure can be used to create many different pairs of training and testing sets, which also ensures that there is no overlap in the composition of the matched training and testing sets. For example, a training set of 20 patterns can be created by randomly setting 20 records from the collected set. A set of 20 other patterns/records can be created as a test set.

In addition, the results of this (and any other) study could be affected by the proportion of nonbankrupt firms to bankrupt firms in both the training and testing sets; that is, the population of all firms contains a certain proportion of firms on the verge of bankruptcy. This base rate may have an impact on a prediction technique's performance in two ways. First, a technique may not work well when the firms of interest (i.e., those that are bankrupt) constitute a very small percentage of the population (i.e., a low base rate). This would be due to a technique's inability to identify the features necessary for classification. Second, there are differences in base rates between training samples and testing samples. If a classification model is built using a training sample with a certain base rate, does the model still work when the base rate in the test population is different? This issue is important for one more reason: If a classification model based on a certain base rate works across other proportions, it may be possible to build a model using a higher proportion of cases of interest than actually occur in the population.

To study the effects of this proportion on the predictive performance of the two techniques, we created three proportions (or base rates) for the testing set composition while holding the composition of the training set fixed at a 50/50 base rate. The first factor level

(or base rate) could be a 50/50 proportion of bankrupt to nonbankrupt cases, the second level could be an 80/20 proportion (80% nonbankrupt, 20% bankrupt), and the third level could be an approximate 90/10 proportion. We did not know the actual proportion of firms going bankrupt, but believed the 80/20 and 90/10 cases would be close.

Within each of the three different testing set compositions, 20 different training–testing set pairs were generated via Monte Carlo resampling from the original 129 firms. Thus, a total of 60 distinct training–testing dataset pairs were generated from the original data. In each case, the training set and test set pairs contained unique firms (i.e., no overlap was allowed). This restriction provided a stronger test of a technique's performance.

To summarize, neural networks and logistic regression models were developed using training sets of equal proportions of firms to determine the classification function but were evaluated with test sets containing 50/50, 80/20, and 90/10 base rates. (The dataset used here is available at this book's Web site.)

Steps 3 through 6 involved preparing for the neural network experiment. We could have used any neural network software package that implements the aforementioned backpropagation training algorithm to construct and test trained neural network models. We had to decide on the size of the neural network, including the number of hidden layers and the number of neurons in the hidden layer. For example, one possible structure was to use 5 input neurons (1 for each financial ratio), 10 hidden neurons, and 2 output neurons (1 indicating a bankrupt firm and the other indicating a nonbankrupt firm). Figure 6.14 illustrates this network configuration. Neural output values ranged from 0 to 1. The output node BR indicates a firm classified as likely to go bankrupt, and the node NBR, not so.

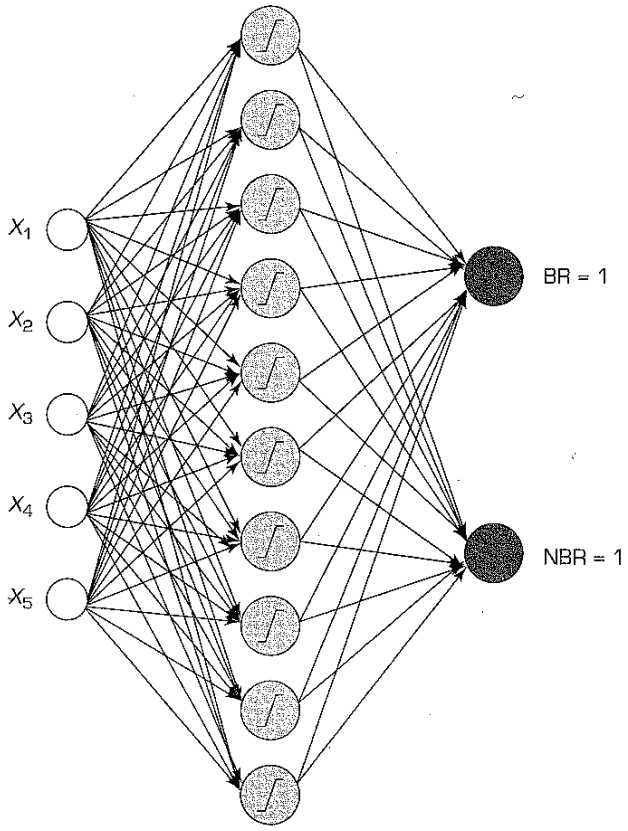


FIGURE 6.14 A Typical ANN Model for Bankruptcy Prediction

A user of a neural network has two difficult decisions to make in the training process (step 6): At what point has the neural network appropriately learned the relationships, and what is the threshold of error with regard to determining correct classifications of the test set? Typically, these issues are addressed by using training tolerances and testing tolerances that state the acceptable levels of variance for considering classifications as "correct."

Step 7 is the actual neural network training. In training the networks in this example, a heuristic backpropagation algorithm was used to ensure convergence (i.e., all firms in the training set are classified correctly). The training set was presented to the neural network software repeatedly until the software sufficiently learned the relationship between the attributes of the cases and whether a firm was distressed. Then, to accurately assess the prediction efficacy of the network, the holdout sample (i.e., test set) was presented to the network and the number of correct classifications was noted (step 8).

In determining correct classifications, a testing threshold of 0.49 was used. Thus, the output node with a value over 0.5 was used to assess whether the network provided a correct classification. Cases in which both output neurons provided output levels either less than 0.5 or greater than 0.5 were automatically treated as misclassifications.

To compare the performance of the neural network against classical statistical techniques, a logistic regression approach was implemented via SYSTAT, a statistical software package. Table 6.2 represents the average percentage of correct classifications provided by the two different techniques when evaluated by the 20 holdout samples for each of the three different test set base rates. When the testing sets contained an equal number of the two cases, neural networks correctly classified 97.5 percent of the holdout cases, whereas logistic regression was correct 93.25 percent of the time. Similarly, when the testing set comprised 20,070 bankrupt firms, neural networks classified at a 95.6 percent correct rate, whereas logistic regression correctly classified at a 92.2 percent rate.

A nonparametric test, the Wilcoxon test for paired observations, was undertaken to assess whether the correct classification percentages for the two techniques were significantly different. Those instances where statistically significant differences were found are indicated in Table 6.2 by footnotes. In general, neural networks performed significantly better than logistic regression.

Table 6.2 also illustrates the correct percentages of bankrupt firm predictions and nonbankrupt firm predictions. In the prediction of bankrupt cases, neural networks predicted significantly better than logistic regression for test sets of equal proportion, at the same percentage when the ratio was 80/20 and a little worse (although not significantly) for 90/10 test sets. The neural networks clearly outperformed the logistic regression model in the prediction of the nonbankrupt firms.

TABLE 6.2 Performance Comparison of Neural Networks and Logistic Regression

	Test Proportions					
	50/50		80/20		90/10	
Criteria	NN	LR	NN	LR	NN	LR
Overall percentage of correct classification	97.5[a]	93.25	95.6[a]	92.2	95.68[b]	90.23
Bankrupt firm classification success rate	97.0[a]	91.90	92.0	92.0	92.5	95.0 ($P = .282$)
Nonbankrupt firm classification success rate	98.0[a]	95.5	96.5[a]	92.25	96.0[b]	89.75

[a] $P < .01$; [b] $P < .05$.

A number of studies in the recent past have investigated the performance of neural networks in predicting business failure. Typically, these studies have compared neural network performance to that of traditional statistical techniques such as discriminant analysis and logistic regression. In addition, some recent studies have compared neural networks to other artificial intelligence techniques, such as decision trees, support vector machines, rough sets, and a variety of rule-induction systems. The purpose of this section was to illustrate how a neural network project can be carried out to predict bankruptcy, not necessarily to argue that neural networks do a better job at prediction in this problem domain.

Section 6.6 Review Questions

1. What parameters can be used to predict failure of a firm?
2. How were data divided between training and test sets for this experiment?
3. Explain what is meant by resampling in this context? How was resampling used for this problem?
4. What were the network parameters for this neural network experiment?
5. How was an output converted to bankrupt or nonbankrupt?
6. How did the neural network model compare with a logistic regression model in this experiment?

6.7 OTHER POPULAR NEURAL NETWORK PARADIGMS

The MLP-based neural networks thus far described in this chapter are just one specific type of neural network. Literally hundreds of different neural networks have been proposed. Many are variants of the MLP model; they just differ in their implementations of input representation, the learning process, output processing, and so on. However, other types of neural networks are quite different from the MLP model. Some of these are introduced later in this chapter. These other variants include radial basis function networks, probabilistic neural networks, generalized regression neural networks, and support vector machines. Many online resources describe the details of these types of neural networks. A good resource is the e-book at the StatSoft, Inc., Web site (**statsoft.com/textbook/stathome.html**). Even though the MLP is the most popular ANN architecture, it is helpful to examine some of the other varieties. The next subsection introduces two of the other popular neural network architectures: Kohonen's self-organizing feature maps and Hopfield networks.

Kohonen's Self-Organizing Feature Maps

First introduced by the Finnish professor Teuvo Kohonen, **Kohonen's self-organizing feature maps** (Kohonen networks or SOM, in short) are among the most popular data mining architectures. SOM provide a way to represent multidimensional data in much lower dimensional spaces, usually one or two dimensions. This process of reducing the dimensionality of vectors is essentially a data compression technique and is known as *vector quantisation*. Additionally, the Kohonen technique creates a network that stores information in such a way that any topological relationships within the training set are maintained.

One of the most interesting aspects of SOM is that they learn to classify data without supervision. In supervised training techniques, such as backpropagation, the training data consists of vector pairs—an input vector and a target vector. With this approach, an input vector is presented to the network (typically a multilayer feedforward network), and the output is compared with the target vector. If they differ, the weights of the network are altered slightly to reduce the error in the output. This is repeated many times and with many sets of vector pairs until the network gives the desired output. In contrast, a SOM requires no target vector. A SOM learns to classify the training data without any external supervision whatsoever.

Before getting into the details of the technique, it is often helpful to forget most everything you may already know about neural networks. If you try to think of SOM in

terms of neurons, activation functions, and feedforward/recurrent connections, you will likely become confused rather quickly. So, temporarily forget all that knowledge you have gained from the previous sections of this chapter and get ready to embark on a new neural network paradigm.

SOM NETWORK ARCHITECTURE In order to keep it simple and easy to understand, we will use a two-dimensional SOM. The network is created from a two-dimensional lattice of "nodes," each of which is fully connected to the input layer. Figure 6.15 illustrates a very small Kohonen network of 4 × 4 nodes connected to the input layer (with three inputs), representing a two-dimensional vector.

In the lattice, each node has a specific topological position (an x, y coordinate in the lattice) and has a vector of weights of the same dimension as the input vectors; that is, if the training data consists of vectors, $\boldsymbol{V}$, of n dimensions (e.g., V_1, V_2, V_3, . . ., V_n), then each node in the lattice will contain a corresponding weight vector, $\boldsymbol{W}$, of n dimensions (W_1, W_2, W_3, . . .,W_n).

THE SOM LEARNING ALGORITHM Unlike many other types of neural networks, a target output does not need to be specified. Instead, during the training process the node weights are matched against the input vector. The node on the lattice that most closely resembles the input vector (and its surrounding area of nodes, "the neighborhood") is selectively optimized to more closely resemble the data for the class of which the input vector is a member. From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones. Each zone is effectively a feature classifier, so you can think of the graphical output as a type of feature map of the input space. Any new, previously unseen input vectors presented to the network will stimulate nodes in the zone with similar weight vectors.

Training in SOM occurs in the following steps and over many iterations (details on the algorithm are also available at **ai-junkie.com/ann/som/som3.html**):

1. Each node's weights are initialized.
2. A vector is chosen from the set of training data and presented to the lattice.
3. Every node is examined to calculate which one's weight most resembles the input vector. The winning node, commonly known as the Best Matching Unit (BMU), is identified.

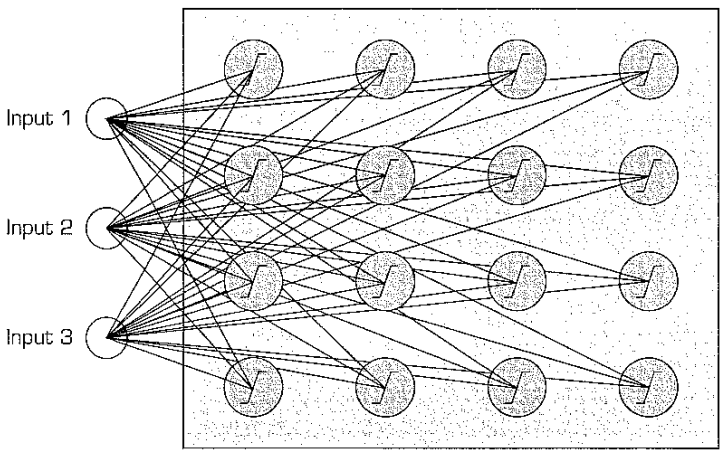


FIGURE 6.15 A 4 × 4 Kohonen (SOM) Network Structure

4. The radius of the neighborhood of the BMU is now calculated. This is a va that starts large, typically set to the "radius" of the lattice, but diminishes ea step. Any nodes found within this radius are deemed to be inside the BMI neighborhood.
5. Each neighboring node's (the nodes found in step 4) weights are adjusted to ma them more like the input vector. The closer a node is to the BMU, the more weights are altered.
6. Repeat steps 2 through 5 for N iterations or until other stopping criteria are reache

The mathematical details of these steps can be found in Haykin (2009), and will not l given herein.

APPLICATIONS OF SOM SOM are commonly used as *visualization identification*. The can make it easy for humans to "see" relationships among vast amounts of diverse da items (e.g., multidimensional structured or unstructured data such as images, audi recordings, and text documents). With regards to image recognition, SOM can help fin similar two- or three-dimensional pictures from a large collection of images in a database which can be very useful to law enforcement agencies in identifying criminals via autc mated face recognition. Furthermore, the technology may also help in identifying chil pornography from a collection of images on a confiscated computer drive or on a Wel site. Other applications of SOM include:

- Bibliographic classification (**edpsciences.org/articles/aas/pdf/1998/10/ds1464.pdf**
- Image-browsing systems (**cis.hut.fi/picsom/**)
- Medical diagnosis
- Interpretation of seismic activity
- Speech recognition (this is what Kohonen initially used this architecture for)
- Data compression
- Separating sound sources (**cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi**)
- Environmental modeling
- Vampire classification! (**hut.fi/~jslindst/vtes/**)

Hopfield Networks

The Hopfield network is another interesting neural network architecture, first introduced by John Hopfield (1982). He demonstrated in a series of research articles in the early 1980s how highly interconnected networks of nonlinear neurons can be extremely effective in solving complex computational problems. These networks were shown to provide novel and quick solutions to a family of problems stated in terms of a desired objective subject to a number of constraints.

One of the major advantages of Hopfield neural networks, which are gaining popularity in solving optimization or mathematical programming problems, is the fact that their structure can be realized on an electronic circuit board, possibly on a VLSI (very large-scale integration) circuit, to be used as an online solver with a parallel-distributed process. The structure of a Hopfield network utilizes three common methods—penalty functions, Lagrange multipliers, and primal and dual methods—to construct an energy function. When the energy function reaches a steady state, an optimal solution of the problem is believed to be obtained. Hopfield networks have been successfully used to solve all three types of mathematical problems: linear, nonlinear, and mixed integer (Wen et al., 2009).

Architecturally, a general Hopfield network is represented as a single large layer of neurons with total interconnectivity; that is, each neuron is connected to every other neuron within the network (see Figure 6.16). In addition, the output of each neuron may depend on its own previous values. One use of Hopfield networks has

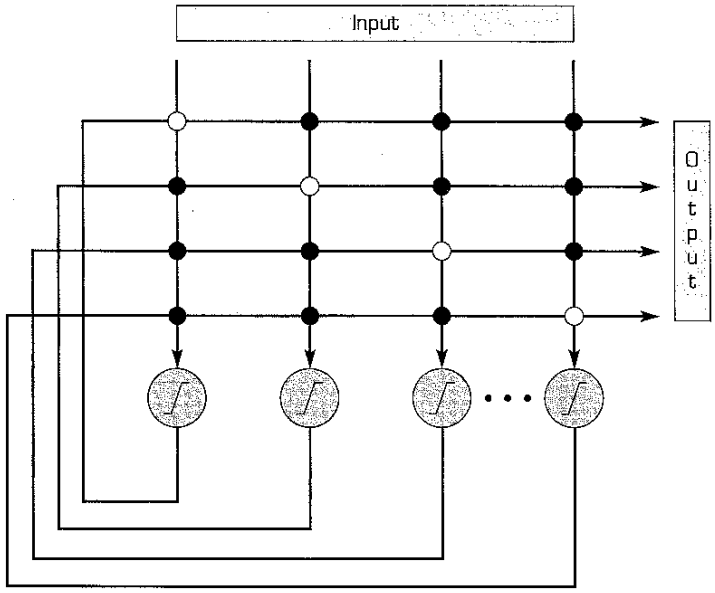


FIGURE 6.16 A Simple Hopfield Network

been in solving the classic traveling salesman problem (TSP). In this problem, each neuron in the network represents the desirability of a city n being visited in position m of a TSP tour. Interconnection weights are specified to represent the constraints of feasible solutions to the TSP (e.g., forcing a city to appear in a tour only once). An energy function is specified, which represents the objective of the model (e.g., minimize total distance in the TSP tour), and is used in determining when to stop the neural network evolution with the best possible solution. The network starts with random neuron values and, using the stated interconnection weights, the neuron values are updated iteratively over time. Gradually, the neuron values stabilize, evolving into a final state (as driven by the global energy function) that represents a solution to the problem. At this point in the network evolution, the value of neuron (n, m) represents whether city n should be in location m of the TSP tour. Although Hopfield and Tank (1985) and others have claimed great success in solving the TSP, further research has shown that those claims were somewhat premature in terms of finding "the global" optimum solution. Nonetheless, this novel approach to solving a classic optimization problem offers promise for solving a wide range of complex optimization problems, especially because Hopfield networks take advantage of the inherent parallelism of the neural structure.

Hopfield networks are distinct from feedforward networks because the neurons are highly interconnected, weights between neurons tend to be fixed, and there is no training, per se. The complexity and challenge in using a Hopfield network for optimization problems is in the correct specification of the interconnection weights and the identification of the proper global energy function to drive the network evolution process.

Section 6.7 Review Questions

1. List some of the different types of neural networks.
2. What is one key difference between an MLP network and a Kohonen network?
3. What is another name for a Kohonen network?
4. Briefly describe a Hopfield network.

6.8 APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS

Because of their ability to model real-world complex problems, researchers and practitioners have found many uses for ANN. Many of these uses have led to solutions for problems previously believed to be unsolvable. At the highest conceptual level, common uses of neural networks can be classified into four general classes (somewhat resembling the general groupings of tasks addressed by data mining):

1. ***Classification.*** A neural network can be trained to predict a categorical (i.e., class-label) output variable. In a mathematical sense, this involves dividing an n-dimensional space into various regions and given a point in the space one should be able to determine to which region it belongs. This idea has been used in many real-world applications of **pattern recognition**, where each pattern is transformed into a multidimensional point and classified into a certain group, each of which represents a known pattern. Types of ANN used for this task include feedforward networks (such as MLP with backpropagation learning), radial basis functions, and probabilistic neural networks.
2. ***Regression.*** A neural network can be trained to predict an output variable whose values are of numeric (i.e., real or integer numbers) type. If a network fits well in modeling a known sequence of values, it can be used to predict future results. An obvious example of the regression task is stock market index predictions. Types of ANN used for this task include feedforward networks (such as MLP with backpropagation learning) and radial basis functions.
3. ***Clustering.*** Sometimes a dataset is so complicated that there is no obvious way to classify the data into different categories. ANN can be used to identify special features of these data and classify them into different categories without prior knowledge of the data. This technique is useful in identifying natural grouping of things for commercial as well as scientific problems. Types of ANN used for this task include Adaptive Resonance Theory (ART) networks and SOM.
4. ***Association.*** A neural network can be trained to "remember" a number of unique patterns so that when a distorted version of a particular pattern is presented the network associates it with the closest one in its memory and returns the original version of that particular pattern. This can be useful for restoring noisy data and identifying objects and events where the data is noisy or incomplete. Types of ANN used for this task include Hopfield networks.

ANN have been applied in many domains. A survey of their applications in finance can be found in Wallace (2008) and Fadlalla and Lin (2001). Many attempts have been made to use neural networks to predict and understand financial markets. Collard (1990) stated that his neural network model for commodity training resulted in significantly higher profits over other trading strategies. Kamijo and Tanigawa (1990) used a neural network to chart and understand Tokyo Stock Exchange data. They found that the model beat a "buy and hold" strategy. Finally, Fishman et al. (1991) developed a neural model using a variety of economic indicators to predict percentage change in the S&P 500 five days ahead. The authors claim that the model was more accurate in its predictions than alleged experts in the field using the same indicators. Although some of these early neural network studies claimed some level of success in predicting certain features of financial markets, the notion of the unpredictable nature of the markets still stands.

Neural networks have been successfully trained to determine whether loan applications should be approved (Gallant, 1988). It has also been shown that neural networks can predict mortgage applicant solvency better than mortgage writers (Collins et al., 1988). Predicting rating of corporate bonds and attempting to forecast their profitability is another area where neural networks have been successfully applied (see Dutta and Shakhar, 1988; and Surkan and Singleton, 1990). Neural networks outperformed regression analysis and

other mathematical modeling tools in predicting bond rating and profitability. The main conclusion reached was that neural networks provided a more general framework for connecting financial information of a firm to the respective bond rating.

Another interesting area where neural networks have been successfully applied is in sports prediction. In a recent study, Loeffelholz et al. (2009) examined the use of neural networks as a tool for predicting the success of basketball teams in the National Basketball Association (NBA). They used statistics for 620 NBA games to train a number of different neural network models, including feedforward, radial basis, probabilistic, and generalized regression neural networks. They also investigated which subset of features used to train the neural networks were the most salient ones for prediction. They compared the results obtained from these networks to predictions made by numerous basketball experts. The best networks were able to correctly predict the winning team 74.33 percent of the time (on average), as compared to the experts who were correct only 68.67 percent of the time.

In another sports-related study, Iyer and Sharda (2009) explored the use of neural networks to rate and select the best combination cricket players for a competition. Using data from 1985 onward until the 2006–2007 season, they trained and tested numerous neural networks to construct a model that accurately predicted cricketers' near-future performance based on their recent-past accomplishments. They compared the predictions of their neural network models against the actual performance of the same cricketers in the World Cup 2007. Their results showed that neural networks can indeed provide accurate predictions and hence can be used as an invaluable decision support tool in the team selection process.

Fraud prevention is another successfully utilized area of neural network applications in business. Chase Manhattan Bank has successfully used neural networks in dealing with credit card fraud (Rochester, 1990), with the neural network models outperforming traditional regression approaches. Also, neural networks have been used in the validation of bank signatures to prevent fraudulent bank transactions (see Francett, 1989; and Mighell, 1989). These networks identified forgeries significantly better than any human expert.

Another important application of neural networks is in time-series forecasting. Several studies have attempted to use neural networks for time-series prediction. Examples include Fozzard et al. (1989), Tang et al. (1991), and Hill et al. (1994). The general conclusion is that neural networks appear to do at least as well as (if not better than) their statistical counterpart, the Box-Jenkins forecasting technique.

A new and prosperous area of application for neural networks is in the field of health care and medicine. Because of their ability to capture and represent highly complex relationships, neural networks are being used to discover patterns in large health care, medical, and biological datasets. In a recently published study, Das et al. (2009) claim to have developed an effective diagnosis system for heart diseases based on artificial neural networks. Using the Cleveland Heart Disease Database, they showed that neural networks can diagnose heart diseases with impressive 89 percent classification accuracy.

In another recent study, Güler et al. (2009) used artificial neural networks to develop a diagnostic system to detect the severity of traumatic brain injuries. They found a significant similarity between classifications made by neurologists and the ANN system output for normal, mild, moderate, and severe brain injuries. Automating such a high-level expertise-requiring classification problem could lead to more rapid decision making and corresponding actions to save human lives. See Application Case 6.4 for an example of how ANN are used to better diagnose breast cancer.

Because neural networks have been a subject of intense study since the late 1980s, there have been many interesting applications of them. You can do a simple Web search to find plenty of recent examples in addition to the ones listed in this chapter. Some other noteworthy applications include live intrusion tracking (see Thaler, 2002), Web content filtering (Lee et al., 2002), exchange rate prediction (Davis et al., 2001), and hospital bed allocation (Walczak et al., 2002).

APPLICATION CASE 6.4

Neural Networks for Breast Cancer Diagnosis

ANN have proven to be a useful tool in pattern recognition and classification tasks in diverse areas, including clinical medicine. Despite the wide applicability of ANN, the large amount of data required for training makes them an unsuitable classification technique when the available data are scarce. Magnetic resonance spectroscopy (MRS) plays a pivotal role in the investigation of cell biochemistry and provides a reliable method for detection of metabolic changes in breast tissue. The scarcity of MRS data and the complexity of interpretation of relevant physiological information impose extra demands that prohibit the applicability of most statistical and machine-learning techniques developed.

Knowledge-based artificial neural networks (KBANN) help to prevail over such difficulties and complexities. A KBANN combines knowledge from a domain, in the form of simple rules, with connectionist learning. This combination trains the network through the use of small sets of data (as is typical of medical diagnosis tasks). The primary structure is based on the dependencies of a set of known domain rules, and it is necessary to refine those rules through training.

The KBANN process consists of two algorithms. One is the Rules-to-Network algorithm, in which the main task is the translation process between a knowledge base containing information about a domain theory and the initial structure of a neural network. This algorithm maps the structure of an approximately correct domain theory, with all the rules and their dependencies, into a neural network structure. The defined network is then trained using the backpropagation learning algorithm.

Feedback mechanisms, which inhibit or stimulate the growth of normal cells, control the division and replacement of cells in normal tissues. In the case of tumors, that process is incapable of controlling the production of new cells, and the division is done without any regard to the need for replacement, disturbing the structure of normal tissue. Changes observed in phospholipid metabolite concentrations, which are associated with differences in cell proliferation in malignant tissues, have served as the basic inputs for the identification of relevant features present in malignant or cancerous tissues but not in normal tissues. The abnormal levels of certain phospholipid characteristics are considered indicators of tumors. These include several parameters, such as PDE, PME, Pi, PCr, γATP, αATP, and βATP. KBANN produced an accurate tumor classification of 87 percent from a set of 26, with an average pattern error of 0.0500 and a standard deviation of 0.0179.

Source: M. Sordo, H. Buxton, and D. Watson, "A Hybrid Approach to Breast Cancer Diagnosis," in *Practical Applications of Computational Intelligence Techniques,* Vol. 16, L. Jain and P. DeWilde (eds.), Kluwer, Norwell, MA, 2001. **acl.icnet.uk/PUBLICATIONS/sordo/chapter2001.pdf** (accessed May 2009).

In general, ANN are suitable for problems whose inputs are both categorical a numeric, and where the relationships between inputs and outputs are not linear or t input data are not normally distributed. In such cases, classical statistical methods m not be reliable enough. Because ANN do not make any assumptions about the d distribution, their power is less affected than traditional statistical methods when the d are not properly distributed. Finally, there are cases in which the neural networks sim provide one more way of building a predictive model for the situation at hand. Given t ease of experimentation using the available software tools, it is certainly worth explori the power of neural networks in any data modeling situation.

Section 6.8 Review Questions

1. List some applications of neural networks in accounting and finance.
2. What are some sports applications of neural networks?
3. How have neural networks been used in health care?
4. What are some applications of neural networks in information security?
5. Conduct a Web search to identify homeland security applications of neural networks.

Chapter Highlights

- Neural computing involves a set of methods that emulate the way the human brain works. The basic processing unit is a neuron. Multiple neurons are grouped into layers and linked together.
- In a neural network, the knowledge is stored in the weight associated with each connection between two neurons.
- Backpropagation is the most popular paradigm in business applications of neural networks. Most business applications are handled using this algorithm.
- A backpropagation-based neural network consists of an input layer, an output layer, and a certain number of hidden layers (usually one). The nodes in one layer are fully connected to the nodes in the next layer. Learning is done through a trial-and-error process of adjusting the connection weights.
- Each node at the input layer typically represents a single attribute that may affect the prediction.
- Neural network learning can occur in supervised or unsupervised mode.
- In supervised learning mode, the training patterns include a correct answer/classification/forecast.
- In unsupervised learning mode, there are no known answers. Thus, unsupervised learning is used for clustering or exploratory data analysis.
- The usual process of learning in a neural network involves three steps: (1) compute temporary outputs based on inputs and random weights, (2) compute outputs with desired targets, and (3) adjust the weights and repeat the process.
- The delta rule is commonly used to adjust the weights. It includes a learning rate and a momentum parameter.
- Developing neural network–based systems requires a step-by-step process. It includes data preparation and preprocessing, training and testing, and conversion of the trained model into a production system.
- Neural network software is available to allow easy experimentation with many models. Neural network modules are included in all major data mining software tools. Specific neural network packages are also available. Some neural network tools are available as spreadsheet add-ins.
- After a trained network has been created, it is usually implemented in end-user systems through programming languages such as C++, Java, and Visual Basic. Most neural network tools can generate code for the trained network in these languages.
- Many neural network models beyond backpropagation exist, including radial basis functions, support vector machines, Hopfield networks, and Kohonen's self-organizing maps.
- Neural network applications abound in almost all business disciplines as well as in virtually all other functional areas.
- Business applications of neural networks included finance, bankruptcy prediction, time-series forecasting, and so on.
- New applications of neural networks are emerging in health care, security, and so on.

Key Terms

adaptive resonance theory (ART) *255*
artificial neural network (ANN) *245*
axon *245*
backpropagation *248*
black-box testing *264*
connection weight *250*
dendrite *245*
hidden layer *249*
Kohonen's self-organizing feature map *270*
learning algorithm *252*
learning rate *257*
momentum *257*
neural computing *245*
neural network *245*
neuron *245*
nucleus *246*
parallel processing *249*
pattern recognition *274*
perceptron *245*
processing element (PE) *248*
self-organizing *255*
sigmoid (logical activation) function *251*
summation function *250*
supervised learning *252*
synapse *246*
threshold value *251*
topology *261*
transformation (transfer) function *251*
unsupervised learning *252*

Questions for Discussion

1. Compare artificial and biological neural networks. What aspects of biological networks are not mimicked by artificial ones? What aspects are similar?
2. The performance of ANN relies heavily on the summation and transformation functions. Explain the combined effects of the summation and transformation functions and how they differ from statistical regression analysis.
3. ANN can be used for both supervised and unsupervised learning. Explain how they learn in a supervised mode and in an unsupervised mode.
4. Explain the difference between a training set and a testing set. Why do we need to differentiate them? Can the same set be used for both purposes? Why or why not?
5. Say that a neural network has been constructed to predict the creditworthiness of applicants. There are two output nodes: one for yes (1 = yes, 0 = no) and one for no (1 = no, 0 = yes). An applicant receives a score of 0.83 for the "yes" output node and a 0.44 for the "no" output node. Discuss what may have happened and whether the applicant is a good credit risk.
6. Everyone would like to make a great deal of money on the stock market. Only a few are very successful. Why is using an ANN a promising approach? What can it do that other decision support technologies cannot do? How could it fail?

Exercises

TERADATA STUDENT NETWORK (TSN) AND OTHER HANDS-ON EXERCISES

1. Go to the Teradata Student Network Web site (**teradata studentnetwork.com**) or the URL given by your instructor. Locate Web seminars related to data mining and neural networks. Specifically, view the seminar given by Professor Hugh Watson at the SPIRIT2005 conference at Oklahoma State University, then answer the following questions:
   a. Which real-time application at Continental Airlines may have used a neural network?
   b. What inputs and outputs can be used in building a neural network application?
   c. Given that Continental's data mining applications are in real time, how might Continental implement a neural network in practice?
   d. What other neural network applications would you propose for the airline industry?
2. Go to the Teradata Student Network Web site (**teradata studentnetwork.com**) or the URL given by your instructor. Locate the Harrah's case. Read the case and answer the following questions:
   a. Which of the Harrah's data applications are most likely implemented using neural networks?
   b. What other applications could Harrah's develop using the data it is collecting from its customers?
   c. What are some concerns you might have as a customer at this casino?
3. This exercise relates to the bankruptcy-prediction problem discussed in this chapter. The bankruptcy-prediction problem can be viewed as a problem of classification. The dataset you will be using for this problem includes five ratios that have been computed from the financial statements of real-world firms. These five ratios have been used in studies involving bankruptcy prediction. The first sample includes data on firms that went bankrupt and firms that didn't. This will be your training sample for the neural network. The second sample of 10 firms also consists of some bankrupt firms and some nonbankrupt firms. Your goal is to train a neural network, using the first 20 data, and then test its performance on the other 10 data. (Try to analyze the new cases yourself manually before you run the neural network and see how well you do.) The following tables show the training sample and test data you should use for this exercise.

Training Sample						
Firm	WC/TA	RE/TA	EBIT/TA	MVE/TD	S/TA	BR/NB
1	0.1650	0.1192	0.2035	0.8130	1.6702	1
2	0.1415	0.3868	0.0681	0.5755	1.0579	1
3	0.5804	0.3331	0.0810	1.1964	1.3572	1
4	0.2304	0.2960	0.1225	0.4102	3.0809	1
5	0.3684	0.3913	0.0524	0.1658	1.1533	1
6	0.1527	0.3344	0.0783	0.7736	1.5046	1
7	0.1126	0.3071	0.0839	1.3429	1.5736	1

(continued)

8	0.0141	0.2366	0.0905	0.5863	1.4651	1
9	0.2220	0.1797	0.1526	0.3459	1.7237	1
10	0.2776	0.2567	0.1642	0.2968	1.8904	1
11	0.2689	0.1729	0.0287	0.1224	0.9277	0
12	0.2039	−0.0476	0.1263	0.8965	1.0457	0
13	0.5056	−0.1951	0.2026	0.5380	1.9514	0
14	0.1759	0.1343	0.0946	0.1955	1.9218	0
15	0.3579	0.1515	0.0812	0.1991	1.4582	0
16	0.2845	0.2038	0.0171	0.3357	1.3258	0
17	0.1209	0.2823	−0.0113	0.3157	2.3219	0
18	0.1254	0.1956	0.0079	0.2073	1.4890	0
19	0.1777	0.0891	0.0695	0.1924	1.6871	0
20	0.2409	0.1660	0.0746	0.2516	1.8524	0

Test Data

Firm	WC/TA	RE/TA	EBIT/TA	MVE/TD	S/TA	BR/NB
A	0.1759	0.1343	0.0946	0.1955	1.9218	?
B	0.3732	0.3483	−0.0013	0.3483	1.8223	?
C	0.1725	0.3238	0.1040	0.8847	0.5576	?
D	0.1630	0.3555	0.0110	0.3730	2.8307	?
E	0.1904	0.2011	0.1329	0.5580	1.6623	?
F	0.1123	0.2288	0.0100	0.1884	2.7186	?
G	0.0732	0.3526	0.0587	0.2349	1.7432	?
H	0.2653	0.2683	0.0235	0.5118	1.8350	?
I	0.1070	0.0787	0.0433	0.1083	1.2051	?
J	0.2921	0.2390	0.0673	0.3402	0.9277	?

Describe the results of the neural network prediction, including software, architecture, and training information. Submit the trained network file(s) so that your instructor can load and test your network.

4. The purpose of this exercise is to develop a model to predict forest cover type using a number of cartographic measures. The given dataset (Online File W6.1) includes four wilderness areas found in the Roosevelt National Forest of northern Colorado. A total of 12 cartographic measures were utilized as independent variables; seven major forest cover types were used as dependent variables. The following table provides a short description of these independent and dependent variables:

Number	Name	Description
	Independent Variables	
1	Elevation	Elevation in meters
2	Aspect	Aspect in degrees azimuth
3	Slope	Slope in degrees
4	Horizontal_Distance_To_Hydrology	Horizontal distance to nearest surface-water features

(continued)

5	Vertical_Distance_To_Hydrology	Vertical distance to nearest surface-water features
6	Horizontal_Distance_To_Roadways	Horizontal distance to nearest roadway
7	Hillshade_9am	Hill shade index at 9 A.M., summer solstice
8	Hillshade_Noon	Hill shade index at noon, summer solstice
9	Hillshade_3pm	Hill shade index at 3 P.M., summer solstice
10	Horizontal_Distance_To_Fire_Points	Horizontal distance to nearest wildfire ignition points
11	Wilderness_Area (4 binary variables)	Wilderness area designation
12	Soil_Type (40 binary variables)	Soil type designation

Number	Dependent Variable	
1	Cover_Type (7 unique types)	Forest cover type designation

Note: More details about the dataset (variables and observations) can be found in the online file.

This is an excellent example for a multiclass classification problem. The dataset is rather large (with 581,012 unique instances) and feature rich. As you will see, the data is also raw and skewed (unbalanced for different cover types). As a model builder, you are to make necessary decisions to preprocess the data and build the best possible predictor. Use your favorite tool to build the models and document the details of your actions and experiences in a written report. Use screenshots within your report to illustrate important and interesting findings. You are expected to discuss and justify any decision that you make along the way.

The reuse of this dataset is unlimited with retention of copyright notice for Jock A. Blackard and Colorado State University.

TEAM ASSIGNMENTS AND ROLE-PLAYING

1. Consider the following set of data that relates daily electricity usage as a function of outside high temperature (for the day):

Temperature, X	Kilowatts, Y
46.8	12,530
52.1	10,800
55.1	10,180
59.2	9,730
61.9	9,750
66.2	10,230
69.9	11,160
76.8	13,910
79.7	15,110
79.3	15,690
80.2	17,020
83.3	17,880

   a. Plot the raw data. What pattern do you see? What do you think is really affecting electricity usage?
   b. Solve this problem with linear regression $Y = a + bX$ (in a spreadsheet). How well does this work? Plot your results. What is wrong? Calculate the sum-of-the-squares error and R^2.
   c. Solve this problem by using nonlinear regression. We recommend a quadratic function, $Y = a + b_1X + b_2X^2$. How well does this work? Plot your results. Is anything wrong? Calculate the sum-of-the-squares error and R^2.
   d. Break up the problem into three sections (look at the plot). Solve it using three linear regression models—one for each section. How well does this work? Plot your results. Calculate the sum-of-the-squares error and R^2. Is this modeling approach appropriate? Why or why not?
   e. Build a neural network to solve the original problem. (You may have to scale the X and Y values to be between 0 and 1.) Train it (on the entire set of data) and solve the problem (i.e., make predictions for each of the original data items). How well does this work? Plot your results. Calculate the sum-of-the-squares error and R^2.
   f. Which method works best and why?
2. Build a real-world neural network. Using demo software downloaded from the Web (e.g., NeuroSolutions at **neurodimension.com** or another site), identify real-world data (e.g., start searching on the Web at **ics.uci.edu/~mlearn/MLRepository.html** or use data from an organization with which someone in your group has a contact) and build a neural network to make predictions. Topics might include sales forecasts, predicting success in an academic program (e.g., predict GPA from high school rating and SAT scores; being careful to look out for "bad" data, such as GPAs of 0.0), or housing prices; or survey the class for weight, gender, and height and try to predict height based on the other two factors. You could also use

U.S. Census data on this book's Web site or at **census.gov**, by state, to identify a relationship between education level and income. How good are your predictions? Compare the results to predictions generated using standard statistical methods (regression). Which method is better? How could your system be embedded in a DSS for real decision making?

3. For each of the following applications, would it be better to use a neural network or an expert system? Explain your answers, including possible exceptions or special conditions.
   a. Diagnosis of a well-established but complex disease
   b. Price-lookup subsystem for a high-volume merchandise seller
   c. Automated voice-inquiry processing system
   d. Training of new employees
   e. Handwriting recognition
4. Consider the following dataset, which includes three attributes and a classification for admission decisions into an MBA program:

GMAT	GPA	Quantitative GMAT	Decision
650	2.75	35	NO
580	3.50	70	NO
600	3.50	75	YES
450	2.95	80	NO
700	3.25	90	YES
590	3.50	80	YES
400	3.85	45	NO
640	3.50	75	YES
540	3.00	60	?
690	2.85	80	?
490	4.00	65	?

   a. Using the data given here as examples, develop your own manual expert rules for decision making.
   b. Build a decision tree using SPRINT (Gini index). You can build it by using manual calculations or use a spreadsheet to perform the basic calculations.
   c. Build another decision tree, using the entropy and information gain (ID3) approach. You can use a spreadsheet calculator for this exercise.
   d. Although the dataset here is extremely small, try to build a little neural network for it.
   e. Use automated decision tree software (e.g., See5; download a trial version from **rule-quest.com**) to build a tree for these data.
   f. Report the predictions on the last three observations from each of the five classification approaches.
   g. Comment on the similarity and differences of the approaches. What did you learn from this exercise?
5. You have worked on neural networks and other data mining techniques. Give examples of where each of these has been used. Based on your knowledge, how would you differentiate among these techniques? Assume that a few years from now you come across a situation in which neural network or other data mining techniques could be used to build an interesting application for your organization. You have an intern working with you to do the grunt work. How will you decide whether the application is appropriate for a neural network or for another data mining model? Based on your homework assignments, what specific software guidance can you provide to get your intern to be productive for you quickly? Your answer for this question might mention the specific software, describe how to go about setting up the model/neural network, and validate the application.

INTERNET EXERCISES

1. Explore the Web sites of several neural network vendors, such as California Scientific Software (**calsci.com**), NeuralWare (**neuralware.com**), and Ward Systems Group (**wardsystems.com**), and review some of their products. Download at least two demos and install, run, and compare them.
2. A very good repository of data that has been used to test the performance of neural network and other machine-learning algorithms can be accessed at **ics.uci.edu/~mlearn/MLRepository.html**. Some of the datasets are really meant to test the limits of current machine-learning algorithms and compare their performance against new approaches to learning. However, some of the smaller datasets can be useful for exploring the functionality of the software you might download in Internet Exercise 1 or the software that is available at StatSoft.com (i.e., Statistica Data Miner with extensive neural network capabilities). Download at least one dataset from the UCI repository (e.g., Credit Screening Databases, Housing Database). Then apply neural networks as well as decision tree methods, as appropriate. Prepare a report on your results. (Some of these exercises could also be completed in a group or may even be proposed as semester-long projects for term papers and so on.)
3. Go to **calsci.com** and read about the company's various business applications. Prepare a report that summarizes the applications.
4. Go to **nd.com**. Read about the company's applications in investment and trading. Prepare a report about them.
5. Go to **nd.com**. Download the trial version of Neurosolutions for Excel and experiment with it, using one of the datasets from the exercises in this chapter. Prepare a report about your experience with the tool.
6. Go to **neoxi.com**. Identify at least two software tools that have not been mentioned in this chapter. Visit Web

sites of those tools and prepare a brief report on the tools' capabilities.

7. Go to **neuroshell.com**. Look at Gee Whiz examples. Comment on the feasibility of achieving the results claimed by the developers of this neural network model.
8. Go to **easynn.com**. Download the trial version of the software. After the installation of the software, find the sample file called Houseprices.tvq. Retrain the neural network and test the model by supplying some data. Prepare a report about your experience with this software.
9. Visit **statsoft.com**. Download at least three white papers of applications. Which of these applications may have used neural networks?
10. Go to **neuralware.com**. Prepare a report about the products the company offers.

END OF CHAPTER APPLICATION CASE

Coors Improves Beer Flavors with Neural Networks

Coors Brewers Ltd., based in Burton-upon-Trent, Britain's brewing capital, is proud of having the United Kingdom's top beer brands, a 20 percent share of the market, years of experience, and some of the best people in the business. Popular brands include Carling (the country's bestselling lager), Grolsch, Coors Fine Light Beer, Sol, and Korenwolf.

Problem

Today's customer has a wide variety of options regarding what he or she drinks. A drinker's choice depends on various factors, including mood, venue, and occasion. Coors' goal is to ensure that the customer chooses a Coors brand no matter what the circumstances are.

According to Coors, creativity is the key to long-term success. To be the customer's choice brand, Coors needs to be creative and anticipate the customer's ever so rapidly changing moods. An important issue with beers is the flavor; each beer has a distinctive flavor. These flavors are mostly determined through panel tests. However, such tests take time. If Coors could understand the beer flavor based solely on its chemical composition, it would open up new avenues to create beer that would suit customer expectations.

The relationship between chemical analysis and beer flavor is not clearly understood yet. Substantial data exist on the chemical composition of a beer and sensory analysis. Coors needed a mechanism to link those two together. Neural networks were applied to create the link between chemical composition and sensory analysis.

Solution

Over the years, Coors Brewers Ltd. has accumulated a significant amount of data related to the final product analysis, which has been supplemented by sensory data provided by the trained in-house testing panel. Some of the analytical inputs and sensory outputs are shown in the following table:

Analytical Data: Inputs	Sensory Data: Outputs
Alcohol	Alcohol
Color	Estery
Calculated bitterness	Malty
Ethyl acetate	Grainy
Isobutyl acetate	Burnt
Ethyl butyrate	Hoppy
Isoamyl acetate	Toffee
Ethyl hexanoate	Sweet

A single neural network, restricted to a single quality and flavor, was first used to model the relationship between the analytical and sensory data. The neural network was based on a package solution supplied by NeuroDimension, Inc. (**nd.com**). The neural network consisted of an MLP architecture with two hidden layers. Data were normalized within the network, thereby enabling comparison between the results for the various sensory outputs. The neural network was trained (to learn the relationship between the inputs and outputs) through the presentation of many combinations of relevant input/output combinations. When there was no observed improvement in the network error in the

last 100 epochs, training was automatically terminated. Training was carried out 50 times to ensure that a considerable mean network error could be calculated for comparison purposes. Prior to each training run, a different training and cross-validation dataset was presented by randomizing the source data records, thereby removing any bias.

This technique produced poor results, due to two major factors. First, concentrating on a single product's quality meant that the variation in the data was pretty low. The neural network could not extract useful relationships from the data. Second, it was probable that only one subset of the provided inputs would have an impact on the selected beer flavor. Performance of the neural network was affected by "noise" created by inputs that had no impact on flavor.

A more diverse product range was included in the training range to address the first factor. It was more challenging to identify the most important analytical inputs. This challenge was addressed by using a software switch that enabled the neural network to be trained on all possible combinations of inputs. The switch was not used to disable a significant input; if the significant input were disabled, we could expect the network error to increase. If the disabled input was insignificant, then the network error would either remain unchanged or be reduced due to the removal of noise. This approach is called an *exhaustive search* because all possible combinations are evaluated. The technique, although conceptually simple, was computationally impractical with the numerous inputs; the number of possible combinations was 16.7 million per flavor.

A more efficient method of searching for the relevant inputs was required. A genetic algorithm (see Chapter 13 for a detailed description of genetic algorithms) was the solution to the problem. A genetic algorithm was able to manipulate the different input switches in response to the error term from the neural network. The objective of the genetic algorithm was to minimize the network error term. When this minimum was reached, the switch settings would identify the analytical inputs that were most likely to predict the flavor.

Results

After determining what inputs were relevant, it was possible to identify which flavors could be predicted more skillfully. The network was trained using the relevant inputs previously identified multiple times. Before each training run, the network data were randomized to ensure that a different training and cross-validation dataset was used. Network error was recorded after each training run. The testing set used for assessing the performance of the trained network contained approximately 80 records out of the sample data. The neural network accurately predicted a few flavors by using the chemical inputs. For example, "burnt" flavor was predicted with a correlation coefficient of 0.87.

Today, a limited number of flavors are being predicted by using the analytical data. Sensory response is extremely complex, with many potential interactions and hugely variable sensitivity thresholds. Standard instrumental analysis tends to be of gross parameters, and for practical and economical reasons, many flavor-active compounds are simply not measured. The relationship of flavor and analysis can be effectively modeled only if a large number of flavor-contributory analytes are considered. What is more, in addition to the obvious flavor-active materials, mouth-feel and physical contributors should also be considered in the overall sensory profile. With further development of the input parameters, the accuracy of the neural network models will improve.

Questions for the Case

1. Why is beer flavor important to Coors' profitability?
2. What is the objective of the neural network used at Coors?
3. Why were the results of Coors' neural network initially poor, and what was done to improve the results?
4. What benefits might Coors derive if this project is successful?
5. What modifications would you make to improve the results of beer flavor prediction?

Sources: C. I. Wilson and L. Threapleton, "Application of Artificial Intelligence for Predicting Beer Flavours from Chemical Analysis," *Proceedings of the 29th European Brewery Congress,* Dublin, Ireland, May 17–22, 2003, **neurosolutions.com/resources/apps/beer.html** (accessed May 2009); R. Nischwitz, M. Goldsmith, M. Lees, P. Rogers, and L. MacLeod, "Developing Functional Malt Specifications for Improved Brewing Performance," The Regional Institute Ltd., **regional.org.au/au/abts/1999/nischwitz.htm** (accessed May 2009).

References

Ainscough, T. L., and J. E. Aronson. (1999). "A Neural Networks Approach for the Analysis of Scanner Data." *Journal of Retailing and Consumer Services,* Vol. 6.

Altman, E. I. (1968). "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy." *Journal of Finance,* Vol. 23.

California Scientific. "Maximize Returns on Direct Mail with BrainMaker Neural Networks Software." **calsci.com/DirectMail.html** (accessed August 2009).

Collard, J. E. (1990). "Commodity Trading with a Neural Net." *Neural Network News,* Vol. 2, No. 10.

Collins, E., S. Ghosh, and C. L. Scofield. (1988). "An Application of a Multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgments." *IEEE International Conference on Neural Networks,* Vol. 2, pp. 459–466.

Das, R., I. Turkoglu, and A. Sengur. (2009). "Effective Diagnosis of Heart Disease Through Neural Networks Ensembles." *Expert Systems with Applications,* Vol. 36, pp. 7675–7680.

Davis, J. T., A. Episcopos, and S. Wettimuny. (2001). "Predicting Direction Shifts on Canadian–U.S. Exchange Rates with Artificial Neural Networks." *International Journal of Intelligent Systems in Accounting, Finance and Management,* Vol. 10, No. 2.

Delen, D., and E. Sirakaya. (2006). "Determining the Efficacy of Data-Mining Methods in Predicting Gaming Ballot Outcomes." *Journal of Hospitality & Tourism Research,* Vol. 30, No. 3, pp. 313–332.

Delen, D., R. Sharda, and M. Bessonov. (2006). "Identifying Significant Predictors of Injury Severity in Traffic Accidents Using a Series of Artificial Neural Networks." *Accident Analysis and Prevention,* Vol. 38, No. 3, pp. 434–444.

Dutta, S., and S. Shakhar. (1988, July 24–27). "Bond-Rating: A Non-Conservative Application of Neural Networks." *Proceedings of the IEEE International Conference on Neural Networks,* San Diego, CA.

Estévez, P. A., M. H. Claudio, and C. A. Perez. "Prevention in Telecommunications Using Fuzzy Rules and Neural Networks." **cec.uchile.cl/~pestevez/RI0.pdf** (accessed May 2009).

Fadlalla, A., and C. Lin. (2001). "An Analysis of the Applications of Neural Networks in Finance." *Interfaces,* Vol. 31, No. 4.

Fishman, M., D. Barr, and W. Loick. (1991, April). "Using Neural Networks in Market Analysis." *Technical Analysis of Stocks and Commodities.*

Fozzard, R., G. Bradshaw, and L. Ceci. (1989). "A Connectionist Expert System for Solar Flare Forecasting." In D. S. Touretsky (ed.), *Advances in Neural Information Processing Systems,* Vol. 1. San Mateo, CA: Kaufman.

Francett, B. (1989, January). "Neural Nets Arrive." *Computer Decisions.*

Gallant, S. (1988, February). "Connectionist Expert Systems." *Communications of the ACM,* Vol. 31, No. 2.

Güler, I., Z. Gökçil, and E. Gülbandilar. (2009). "Evaluating Traumatic Brain Injuries Using Artificial Neural Networks." *Expert Systems with Applications,* Vol. 36, pp. 10424–10427.

Haykin, S. S. (2009). *Neural Networks and Learning Machines,* 3rd ed. Upper Saddle River, NJ: Prentice Hall.

Hill, T., T. Marquez, M. O'Connor, and M. Remus. (1994). "Neural Network Models for Forecasting and Decision Making." *International Journal of Forecasting,* Vol. 10.

Hopfield, J. (1982, April). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of National Academy of Science,* Vol. 79, No. 8.

Hopfield, J. J., and D. W. Tank. (1985). "Neural Computation of Decisions in Optimization Problems." *Biological Cybernetics,* Vol. 52.

Iyer, S. R., and R. Sharda. (2009). "Prediction of Athletes' Performance Using Neural Networks: An Application in Cricket Team Selection." *Expert Systems with Applications,* Vol. 36, No. 3, pp. 5510–5522.

Kamijo, K., and T. Tanigawa. (1990, June 7–11). "Stock Price Pattern Recognition: A Recurrent Neural Network Approach." *International Joint Conference on Neural Networks,* San Diego.

Lee, P. Y., S. C. Hui, and A. C. M. Fong. (2002, September/October). "Neural Networks for Web Content Filtering." *IEEE Intelligent Systems.*

Liang, T. P. (1992). "A Composite Approach to Automated Knowledge Acquisition." *Management Science,* Vol. 38, No. 1.

Loeffelholz, B., E. Bednar, and K. W. Bauer. (2009). "Predicting NBA Games Using Neural Networks." *Journal of Quantitative Analysis in Sports,* Vol. 5, No. 1.

McCulloch, W. S., and W. H. Pitts. (1943). "A Logical Calculus of the Ideas Imminent in Nervous Activity." *Bulletin of Mathematical Biophysics,* Vol. 5.

Medsker, L., and J. Liebowitz. (1994). *Design and Development of Expert Systems and Neural Networks.* New York: Macmillan, p. 163.

Mighell, D. (1989). "Back-Propagation and Its Application to Handwritten Signature Verification." In D. S. Touretsky (ed.), *Advances in Neural Information Processing Systems.* San Mateo, CA: Kaufman.

Minsky, M., and S. Papert. (1969). *Perceptrons.* Cambridge, MA: MIT Press.

Neural Technologies. "Combating Fraud: How a Leading Telecom Company Solved a Growing Problem." **neuralt.com/iqs/dlsfa.list/dlcpti.7/downloads.html.**

Nischwitz, R., M. Goldsmith, M. Lees, P. Rogers, and L. MacLeod. "Developing Functional Malt Specifications for Improved Brewing Performance." The Regional Institute Ltd., **regional.org.au/au/abts/1999/nischwitz.htm** (accessed May 2009).

Piatesky-Shapiro, G. "ISR: Microsoft Success Using Neural Network for Direct Marketing," **kdnuggets.com/news/94/n9.txt** (accessed May 2009).

Principe, J. C., N. R. Euliano, and W. C. Lefebvre. (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. New York: Wiley.

Rochester, J. (ed.). (1990, February). "New Business Uses for Neurocomputing." *I/S Analyzer.*

Sirakaya, E., D. Delen, and H-S. Choi. (2005). "Forecasting Gaming Referenda." *Annals of Tourism Research,* Vol. 32, No. 1, pp. 127–149.

Sordo, M., H. Buxton, and D. Watson. (2001). "A Hybrid Approach to Breast Cancer Diagnosis." In L. Jain and P. DeWilde (eds.), *Practical Applications of Computational Intelligence Techniques,* Vol. 16. Norwell, MA: Kluwer.

Surkan, A., and J. Singleton. (1990). "Neural Networks for Bond Rating Improved by Multiple Hidden Layers." *Proceedings of the IEEE International Conference on Neural Networks,* Vol. 2.

Tang, Z., C. de Almieda, and P. Fishwick. (1991). "Time-Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology." *Simulation,* Vol. 57, No. 5.

Thaler, S. L. (2002, January/February). "AI for Network Protection: LITMUS:—Live Intrusion Tracking via Multiple Unsupervised STANNOs." *PC AI.*

Walczak, S., W. E. Pofahi, and R. J. Scorpio. (2002). "A Decision Support Tool for Allocating Hospital Bed Resources and Determining Required Acuity of Care." *Decision Support Systems,* Vol. 34, No. 4.

Wallace, M. P. (2008, July). "Neural Networks and Their Applications in Finance." *Business Intelligence Journal,* pp. 67–76.

Wen, U-P., K-M. Lan, and H-S. Shih. (2009). "A Review of Hopfield Neural Networks for Solving Mathematical Programming Problems." *European Journal of Operational Research,* Vol. 198, pp. 675–687.

Wilson, C. I., and L. Threapleton. (2003, May 17–22). "Application of Artificial Intelligence for Predicting Beer Flavours from Chemical Analysis." *Proceedings of the 29th European Brewery Congress,* Dublin, Ireland. **neurosolutions.com/resources/apps/beer.html** (accessed May 2009).

Wilson, R., and R. Sharda. (1994). "Bankruptcy Prediction Using Neural Networks." *Decision Support Systems,* Vol. 11.

Zahedi, F. (1993). *Intelligent Systems for Business: Expert Systems with Neural Networks.* Belmont, CA: Wadsworth.