Introduction to Data Mining

Pang-Ning Tan
Michael Steinbach
Vipin Kumar

Pearson

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England and Associated Companies throughout the world

Visit us on the World Wide Web at: www.pearsoned.co.uk

ISBN 10: 1-292-02615-4
ISBN 13: 978-1-292-02615-2

Printed in the United States of America

Contents

1

Introduction

Rapid advances in data collection and storage technology have enabled organizations to accumulate vast amounts of data. However, extracting useful information has proven extremely challenging. Often, traditional data analysis tools and techniques cannot be used because of the massive size of a data set. Sometimes, the non-traditional nature of the data means that traditional approaches cannot be applied even if the data set is relatively small. In other situations, the questions that need to be answered cannot be addressed using existing data analysis techniques, and thus, new methods need to be developed.

Data mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data. It has also opened up exciting opportunities for exploring and analyzing new types of data and for analyzing old types of data in new ways. In this introductory chapter, we present an overview of data mining and outline the key topics to be covered in this book. We start with a description of some well-known applications that require new techniques for data analysis.

Business Point-of-sale data collection (bar code scanners, radio frequency identification (RFID), and smart card technology) have allowed retailers to collect up-to-the-minute data about customer purchases at the checkout counters of their stores. Retailers can utilize this information, along with other business-critical data such as Web logs from e-commerce Web sites and customer service records from call centers, to help them better understand the needs of their customers and make more informed business decisions.

Data mining techniques can be used to support a wide range of business intelligence applications such as customer profiling, targeted marketing, workflow management, store layout, and fraud detection. It can also help retailers

answer important business questions such as "Who are the most profitable customers?" "What products can be cross-sold or up-sold?" and "What is the revenue outlook of the company for next year?" Some of these questions motivated the creation of association analysis (Chapters 6 and 7), a new data analysis technique.

Medicine, Science, and Engineering Researchers in medicine, science, and engineering are rapidly accumulating data that is key to important new discoveries. For example, as an important step toward improving our understanding of the Earth's climate system, NASA has deployed a series of Earth-orbiting satellites that continuously generate global observations of the land surface, oceans, and atmosphere. However, because of the size and spatio-temporal nature of the data, traditional methods are often not suitable for analyzing these data sets. Techniques developed in data mining can aid Earth scientists in answering questions such as "What is the relationship between the frequency and intensity of ecosystem disturbances such as droughts and hurricanes to global warming?" "How is land surface precipitation and temperature affected by ocean surface temperature?" and "How well can we predict the beginning and end of the growing season for a region?"

As another example, researchers in molecular biology hope to use the large amounts of genomic data currently being gathered to better understand the structure and function of genes. In the past, traditional methods in molecular biology allowed scientists to study only a few genes at a time in a given experiment. Recent breakthroughs in microarray technology have enabled scientists to compare the behavior of thousands of genes under various situations. Such comparisons can help determine the function of each gene and perhaps isolate the genes responsible for certain diseases. However, the noisy and high-dimensional nature of data requires new types of data analysis. In addition to analyzing gene array data, data mining can also be used to address other important biological challenges such as protein structure prediction, multiple sequence alignment, the modeling of biochemical pathways, and phylogenetics.

1.1 What Is Data Mining?

Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown. They also provide capabilities to predict the outcome of a

future observation, such as predicting whether a newly arrived customer will spend more than $100 at a department store.

Not all information discovery tasks are considered to be data mining. For example, looking up individual records using a database management system or finding particular Web pages via a query to an Internet search engine are tasks related to the area of **information retrieval**. Although such tasks are important and may involve the use of the sophisticated algorithms and data structures, they rely on traditional computer science techniques and obvious features of the data to create index structures for efficiently organizing and retrieving information. Nonetheless, data mining techniques have been used to enhance information retrieval systems.

Data Mining and Knowledge Discovery

Data mining is an integral part of **knowledge discovery in databases (KDD)**, which is the overall process of converting raw data into useful information, as shown in Figure 1.1. This process consists of a series of transformation steps, from data preprocessing to postprocessing of data mining results.

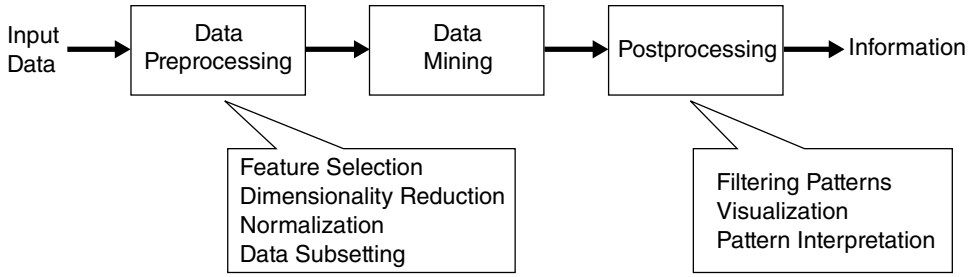


Figure 1.1. The process of knowledge discovery in databases (KDD).

The input data can be stored in a variety of formats (flat files, spreadsheets, or relational tables) and may reside in a centralized data repository or be distributed across multiple sites. The purpose of **preprocessing** is to transform the raw input data into an appropriate format for subsequent analysis. The steps involved in data preprocessing include fusing data from multiple sources, cleaning data to remove noise and duplicate observations, and selecting records and features that are relevant to the data mining task at hand. Because of the many ways data can be collected and stored, data

preprocessing is perhaps the most laborious and time-consuming step in the overall knowledge discovery process.

"Closing the loop" is the phrase often used to refer to the process of integrating data mining results into decision support systems. For example, in business applications, the insights offered by data mining results can be integrated with campaign management tools so that effective marketing promotions can be conducted and tested. Such integration requires a **postprocessing** step that ensures that only valid and useful results are incorporated into the decision support system. An example of postprocessing is visualization (see Chapter 3), which allows analysts to explore the data and the data mining results from a variety of viewpoints. Statistical measures or hypothesis testing methods can also be applied during postprocessing to eliminate spurious data mining results.

1.2 Motivating Challenges

As mentioned earlier, traditional data analysis techniques have often encountered practical difficulties in meeting the challenges posed by new data sets. The following are some of the specific challenges that motivated the development of data mining.

Scalability Because of advances in data generation and collection, data sets with sizes of gigabytes, terabytes, or even petabytes are becoming common. If data mining algorithms are to handle these massive data sets, then they must be scalable. Many data mining algorithms employ special search strategies to handle exponential search problems. Scalability may also require the implementation of novel data structures to access individual records in an efficient manner. For instance, out-of-core algorithms may be necessary when processing data sets that cannot fit into main memory. Scalability can also be improved by using sampling or developing parallel and distributed algorithms.

High Dimensionality It is now common to encounter data sets with hundreds or thousands of attributes instead of the handful common a few decades ago. In bioinformatics, progress in microarray technology has produced gene expression data involving thousands of features. Data sets with temporal or spatial components also tend to have high dimensionality. For example, consider a data set that contains measurements of temperature at various locations. If the temperature measurements are taken repeatedly for an extended period, the number of dimensions (features) increases in proportion to

the number of measurements taken. Traditional data analysis techniques that were developed for low-dimensional data often do not work well for such high-dimensional data. Also, for some data analysis algorithms, the computational complexity increases rapidly as the dimensionality (the number of features) increases.

Heterogeneous and Complex Data Traditional data analysis methods often deal with data sets containing attributes of the same type, either continuous or categorical. As the role of data mining in business, science, medicine, and other fields has grown, so has the need for techniques that can handle heterogeneous attributes. Recent years have also seen the emergence of more complex data objects. Examples of such non-traditional types of data include collections of Web pages containing semi-structured text and hyperlinks; DNA data with sequential and three-dimensional structure; and climate data that consists of time series measurements (temperature, pressure, etc.) at various locations on the Earth's surface. Techniques developed for mining such complex objects should take into consideration relationships in the data, such as temporal and spatial autocorrelation, graph connectivity, and parent-child relationships between the elements in semi-structured text and XML documents.

Data Ownership and Distribution Sometimes, the data needed for an analysis is not stored in one location or owned by one organization. Instead, the data is geographically distributed among resources belonging to multiple entities. This requires the development of distributed data mining techniques. Among the key challenges faced by distributed data mining algorithms include (1) how to reduce the amount of communication needed to perform the distributed computation, (2) how to effectively consolidate the data mining results obtained from multiple sources, and (3) how to address data security issues.

Non-traditional Analysis The traditional statistical approach is based on a hypothesize-and-test paradigm. In other words, a hypothesis is proposed, an experiment is designed to gather the data, and then the data is analyzed with respect to the hypothesis. Unfortunately, this process is extremely labor-intensive. Current data analysis tasks often require the generation and evaluation of thousands of hypotheses, and consequently, the development of some data mining techniques has been motivated by the desire to automate the process of hypothesis generation and evaluation. Furthermore, the data sets analyzed in data mining are typically not the result of a carefully designed

experiment and often represent opportunistic samples of the data, rather than random samples. Also, the data sets frequently involve non-traditional types of data and data distributions.

1.3 The Origins of Data Mining

Brought together by the goal of meeting the challenges of the previous section, researchers from different disciplines began to focus on developing more efficient and scalable tools that could handle diverse types of data. This work, which culminated in the field of data mining, built upon the methodology and algorithms that researchers had previously used. In particular, data mining draws upon ideas, such as (1) sampling, estimation, and hypothesis testing from statistics and (2) search algorithms, modeling techniques, and learning theories from artificial intelligence, pattern recognition, and machine learning. Data mining has also been quick to adopt ideas from other areas, including optimization, evolutionary computing, information theory, signal processing, visualization, and information retrieval.

A number of other areas also play key supporting roles. In particular, database systems are needed to provide support for efficient storage, indexing, and query processing. Techniques from high performance (parallel) computing are often important in addressing the massive size of some data sets. Distributed techniques can also help address the issue of size and are essential when the data cannot be gathered in one location.

Figure 1.2 shows the relationship of data mining to other areas.

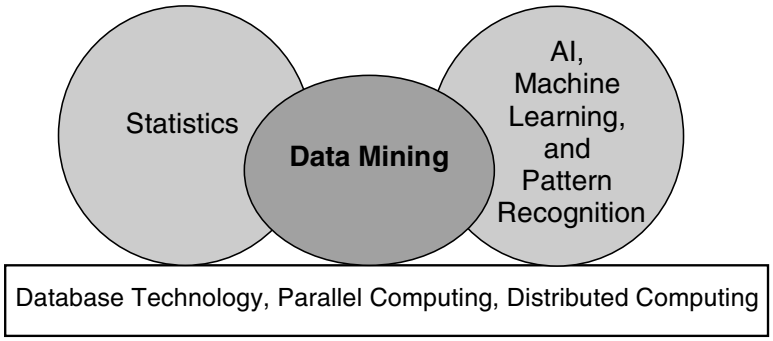


Figure 1.2. Data mining as a confluence of many disciplines.

1.4 Data Mining Tasks

Data mining tasks are generally divided into two major categories:

Predictive tasks. The objective of these tasks is to predict the value of a particular attribute based on the values of other attributes. The attribute to be predicted is commonly known as the **target** or **dependent variable**, while the attributes used for making the prediction are known as the **explanatory** or **independent variables**.

Descriptive tasks. Here, the objective is to derive patterns (correlations, trends, clusters, trajectories, and anomalies) that summarize the underlying relationships in data. Descriptive data mining tasks are often exploratory in nature and frequently require postprocessing techniques to validate and explain the results.

Figure 1.3 illustrates four of the core data mining tasks that are described in the remainder of this book.

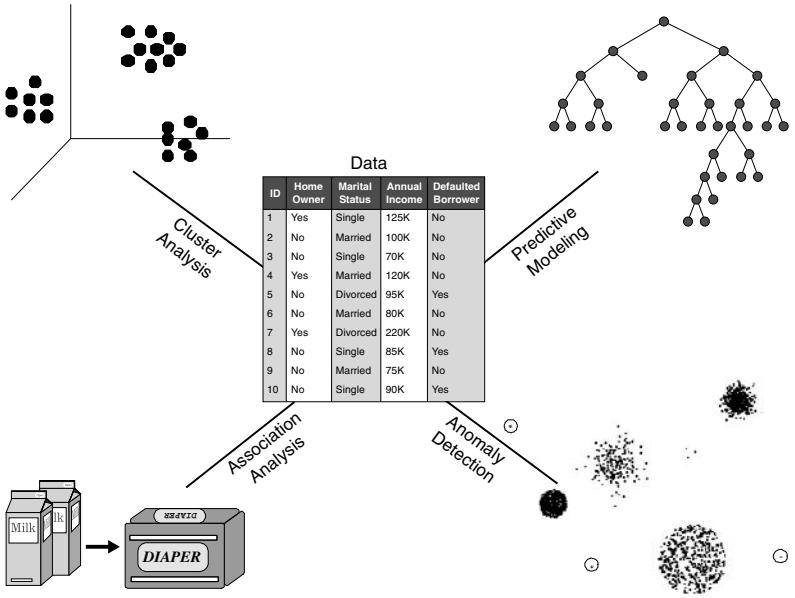


Figure 1.3. Four of the core data mining tasks.

Predictive modeling refers to the task of building a model for the target variable as a function of the explanatory variables. There are two types of predictive modeling tasks: **classification**, which is used for discrete target variables, and **regression**, which is used for continuous target variables. For example, predicting whether a Web user will make a purchase at an online bookstore is a classification task because the target variable is binary-valued. On the other hand, forecasting the future price of a stock is a regression task because price is a continuous-valued attribute. The goal of both tasks is to learn a model that minimizes the error between the predicted and true values of the target variable. Predictive modeling can be used to identify customers that will respond to a marketing campaign, predict disturbances in the Earth's ecosystem, or judge whether a patient has a particular disease based on the results of medical tests.

Example 1.1 (Predicting the Type of a Flower). Consider the task of predicting a species of flower based on the characteristics of the flower. In particular, consider classifying an Iris flower as to whether it belongs to one of the following three Iris species: Setosa, Versicolour, or Virginica. To perform this task, we need a data set containing the characteristics of various flowers of these three species. A data set with this type of information is the well-known Iris data set from the UCI Machine Learning Repository at `http://www.ics.uci.edu/~mlearn`. In addition to the species of a flower, this data set contains four other attributes: sepal width, sepal length, petal length, and petal width. (The Iris data set and its attributes are described further in Section 3.1.) Figure 1.4 shows a plot of petal width versus petal length for the 150 flowers in the Iris data set. Petal width is broken into the categories *low*, *medium*, and *high*, which correspond to the intervals [0, 0.75), [0.75, 1.75), [1.75, ∞), respectively. Also, petal length is broken into categories *low*, *medium*, and *high*, which correspond to the intervals [0, 2.5), [2.5, 5), [5, ∞), respectively. Based on these categories of petal width and length, the following rules can be derived:

Petal width low and petal length low implies Setosa.
Petal width medium and petal length medium implies Versicolour.
Petal width high and petal length high implies Virginica.

While these rules do not classify all the flowers, they do a good (but not perfect) job of classifying most of the flowers. Note that flowers from the Setosa species are well separated from the Versicolour and Virginica species with respect to petal width and length, but the latter two species overlap somewhat with respect to these attributes. ■

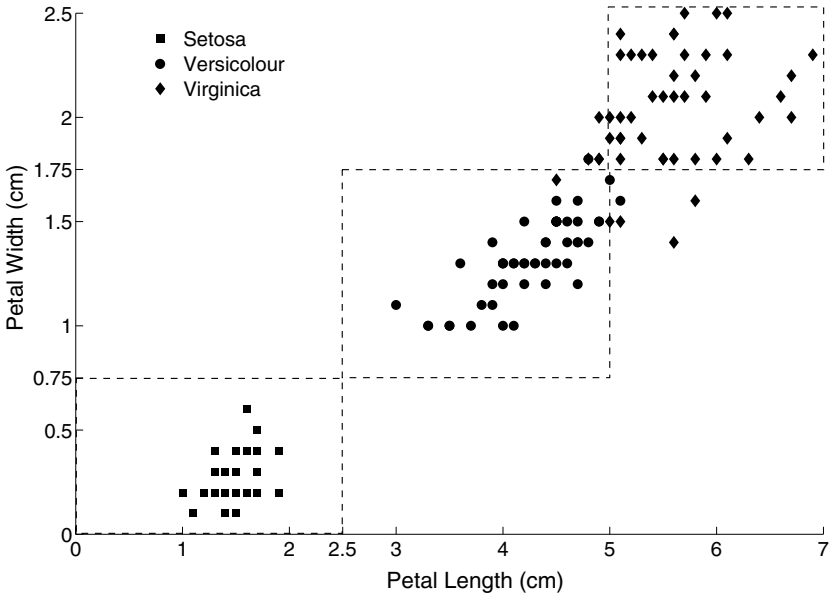


Figure 1.4. Petal width versus petal length for 150 Iris flowers.

Association analysis is used to discover patterns that describe strongly associated features in the data. The discovered patterns are typically represented in the form of implication rules or feature subsets. Because of the exponential size of its search space, the goal of association analysis is to extract the most interesting patterns in an efficient manner. Useful applications of association analysis include finding groups of genes that have related functionality, identifying Web pages that are accessed together, or understanding the relationships between different elements of Earth's climate system.

Example 1.2 (Market Basket Analysis). The transactions shown in Table 1.1 illustrate point-of-sale data collected at the checkout counters of a grocery store. Association analysis can be applied to find items that are frequently bought together by customers. For example, we may discover the rule $\{\texttt{Diapers}\} \longrightarrow \{\texttt{Milk}\}$, which suggests that customers who buy diapers also tend to buy milk. This type of rule can be used to identify potential cross-selling opportunities among related items. ■

Cluster analysis seeks to find groups of closely related observations so that observations that belong to the same cluster are more similar to each other

Table 1.1. Market basket data.

Transaction ID	Items
1	{Bread, Butter, Diapers, Milk}
2	{Coffee, Sugar, Cookies, Salmon}
3	{Bread, Butter, Coffee, Diapers, Milk, Eggs}
4	{Bread, Butter, Salmon, Chicken}
5	{Eggs, Bread, Butter}
6	{Salmon, Diapers, Milk}
7	{Bread, Tea, Sugar, Eggs}
8	{Coffee, Sugar, Chicken, Eggs}
9	{Bread, Diapers, Milk, Salt}
10	{Tea, Eggs, Cookies, Diapers, Milk}

than observations that belong to other clusters. Clustering has been used to group sets of related customers, find areas of the ocean that have a significant impact on the Earth's climate, and compress data.

Example 1.3 (Document Clustering). The collection of news articles shown in Table 1.2 can be grouped based on their respective topics. Each article is represented as a set of word-frequency pairs (w, c), where w is a word and c is the number of times the word appears in the article. There are two natural clusters in the data set. The first cluster consists of the first four articles, which correspond to news about the economy, while the second cluster contains the last four articles, which correspond to news about health care. A good clustering algorithm should be able to identify these two clusters based on the similarity between words that appear in the articles.

Table 1.2. Collection of news articles.

Article	Words
1	dollar: 1, industry: 4, country: 2, loan: 3, deal: 2, government: 2
2	machinery: 2, labor: 3, market: 4, industry: 2, work: 3, country: 1
3	job: 5, inflation: 3, rise: 2, jobless: 2, market: 3, country: 2, index: 3
4	domestic: 3, forecast: 2, gain: 1, market: 2, sale: 3, price: 2
5	patient: 4, symptom: 2, drug: 3, health: 2, clinic: 2, doctor: 2
6	pharmaceutical: 2, company: 3, drug: 2, vaccine: 1, flu: 3
7	death: 2, cancer: 4, drug: 3, public: 4, health: 3, director: 2
8	medical: 2, cost: 3, increase: 2, patient: 2, health: 3, care: 1

■

Anomaly detection is the task of identifying observations whose characteristics are significantly different from the rest of the data. Such observations are known as **anomalies** or **outliers**. The goal of an anomaly detection algorithm is to discover the real anomalies and avoid falsely labeling normal objects as anomalous. In other words, a good anomaly detector must have a high detection rate and a low false alarm rate. Applications of anomaly detection include the detection of fraud, network intrusions, unusual patterns of disease, and ecosystem disturbances.

Example 1.4 (Credit Card Fraud Detection). A credit card company records the transactions made by every credit card holder, along with personal information such as credit limit, age, annual income, and address. Since the number of fraudulent cases is relatively small compared to the number of legitimate transactions, anomaly detection techniques can be applied to build a profile of legitimate transactions for the users. When a new transaction arrives, it is compared against the profile of the user. If the characteristics of the transaction are very different from the previously created profile, then the transaction is flagged as potentially fraudulent. ■

1.5 Scope and Organization of the Book

This book introduces the major principles and techniques used in data mining from an algorithmic perspective. A study of these principles and techniques is essential for developing a better understanding of how data mining technology can be applied to various kinds of data. This book also serves as a starting point for readers who are interested in doing research in this field.

We begin the technical discussion of this book with a chapter on data (Chapter 2), which discusses the basic types of data, data quality, preprocessing techniques, and measures of similarity and dissimilarity. Although this material can be covered quickly, it provides an essential foundation for data analysis. Chapter 3, on data exploration, discusses summary statistics, visualization techniques, and On-Line Analytical Processing (OLAP). These techniques provide the means for quickly gaining insight into a data set.

Chapters 4 and 5 cover classification. Chapter 4 provides a foundation by discussing decision tree classifiers and several issues that are important to all classification: overfitting, performance evaluation, and the comparison of different classification models. Using this foundation, Chapter 5 describes a number of other important classification techniques: rule-based systems, nearest-neighbor classifiers, Bayesian classifiers, artificial neural networks, support vector machines, and ensemble classifiers, which are collections of classi-

fiers. The multiclass and imbalanced class problems are also discussed. These topics can be covered independently.

Association analysis is explored in Chapters 6 and 7. Chapter 6 describes the basics of association analysis: frequent itemsets, association rules, and some of the algorithms used to generate them. Specific types of frequent itemsets—maximal, closed, and hyperclique—that are important for data mining are also discussed, and the chapter concludes with a discussion of evaluation measures for association analysis. Chapter 7 considers a variety of more advanced topics, including how association analysis can be applied to categorical and continuous data or to data that has a concept hierarchy. (A concept hierarchy is a hierarchical categorization of objects, e.g., store items, clothing, shoes, sneakers.) This chapter also describes how association analysis can be extended to find sequential patterns (patterns involving order), patterns in graphs, and negative relationships (if one item is present, then the other is not).

Cluster analysis is discussed in Chapters 8 and 9. Chapter 8 first describes the different types of clusters and then presents three specific clustering techniques: K-means, agglomerative hierarchical clustering, and DBSCAN. This is followed by a discussion of techniques for validating the results of a clustering algorithm. Additional clustering concepts and techniques are explored in Chapter 9, including fuzzy and probabilistic clustering, Self-Organizing Maps (SOM), graph-based clustering, and density-based clustering. There is also a discussion of scalability issues and factors to consider when selecting a clustering algorithm.

The last chapter, Chapter 10, is on anomaly detection. After some basic definitions, several different types of anomaly detection are considered: statistical, distance-based, density-based, and clustering-based. Appendices A through E give a brief review of important topics that are used in portions of the book: linear algebra, dimensionality reduction, statistics, regression, and optimization.

The subject of data mining, while relatively young compared to statistics or machine learning, is already too large to cover in a single book. Selected references to topics that are only briefly covered, such as data quality, are provided in the bibliographic notes of the appropriate chapter. References to topics not covered in this book, such as data mining for streams and privacy-preserving data mining, are provided in the bibliographic notes of this chapter.

1.6 Bibliographic Notes

The topic of data mining has inspired many textbooks. Introductory textbooks include those by Dunham [10], Han and Kamber [21], Hand et al. [23], and Roiger and Geatz [36]. Data mining books with a stronger emphasis on business applications include the works by Berry and Linoff [2], Pyle [34], and Parr Rud [33]. Books with an emphasis on statistical learning include those by Cherkassky and Mulier [6], and Hastie et al. [24]. Some books with an emphasis on machine learning or pattern recognition are those by Duda et al. [9], Kantardzic [25], Mitchell [31], Webb [41], and Witten and Frank [42]. There are also some more specialized books: Chakrabarti [4] (web mining), Fayyad et al. [13] (collection of early articles on data mining), Fayyad et al. [11] (visualization), Grossman et al. [18] (science and engineering), Kargupta and Chan [26] (distributed data mining), Wang et al. [40] (bioinformatics), and Zaki and Ho [44] (parallel data mining).

There are several conferences related to data mining. Some of the main conferences dedicated to this field include the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), the IEEE International Conference on Data Mining (ICDM), the SIAM International Conference on Data Mining (SDM), the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), and the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Data mining papers can also be found in other major conferences such as the ACM SIGMOD/PODS conference, the International Conference on Very Large Data Bases (VLDB), the Conference on Information and Knowledge Management (CIKM), the International Conference on Data Engineering (ICDE), the International Conference on Machine Learning (ICML), and the National Conference on Artificial Intelligence (AAAI).

Journal publications on data mining include *IEEE Transactions on Knowledge and Data Engineering*, *Data Mining and Knowledge Discovery*, *Knowledge and Information Systems*, *Intelligent Data Analysis*, *Information Systems*, and the *Journal of Intelligent Information Systems*.

There have been a number of general articles on data mining that define the field or its relationship to other fields, particularly statistics. Fayyad et al. [12] describe data mining and how it fits into the total knowledge discovery process. Chen et al. [5] give a database perspective on data mining. Ramakrishnan and Grama [35] provide a general discussion of data mining and present several viewpoints. Hand [22] describes how data mining differs from statistics, as does Friedman [14]. Lambert [29] explores the use of statistics for large data sets and provides some comments on the respective roles of data mining and statistics.

Glymour et al. [16] consider the lessons that statistics may have for data mining. Smyth et al. [38] describe how the evolution of data mining is being driven by new types of data and applications, such as those involving streams, graphs, and text. Emerging applications in data mining are considered by Han et al. [20] and Smyth [37] describes some research challenges in data mining. A discussion of how developments in data mining research can be turned into practical tools is given by Wu et al. [43]. Data mining standards are the subject of a paper by Grossman et al. [17]. Bradley [3] discusses how data mining algorithms can be scaled to large data sets.

With the emergence of new data mining applications have come new challenges that need to be addressed. For instance, concerns about privacy breaches as a result of data mining have escalated in recent years, particularly in application domains such as Web commerce and health care. As a result, there is growing interest in developing data mining algorithms that maintain user privacy. Developing techniques for mining encrypted or randomized data is known as **privacy-preserving data mining**. Some general references in this area include papers by Agrawal and Srikant [1], Clifton et al. [7] and Kargupta et al. [27]. Vassilios et al. [39] provide a survey.

Recent years have witnessed a growing number of applications that rapidly generate continuous streams of data. Examples of stream data include network traffic, multimedia streams, and stock prices. Several issues must be considered when mining data streams, such as the limited amount of memory available, the need for online analysis, and the change of the data over time. Data mining for stream data has become an important area in data mining. Some selected publications are Domingos and Hulten [8] (classification), Giannella et al. [15] (association analysis), Guha et al. [19] (clustering), Kifer et al. [28] (change detection), Papadimitriou et al. [32] (time series), and Law et al. [30] (dimensionality reduction).

Bibliography

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of 2000 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 439–450, Dallas, Texas, 2000. ACM Press.

[2] M. J. A. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Wiley Computer Publishing, 2nd edition, 2004.

[3] P. S. Bradley, J. Gehrke, R. Ramakrishnan, and R. Srikant. Scaling mining algorithms to large databases. *Communications of the ACM*, 45(8):38–43, 2002.

[4] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco, CA, 2003.

[5] M.-S. Chen, J. Han, and P. S. Yu. Data Mining: An Overview from a Database Perspective. *IEEE Transactions on Knowledge abd Data Engineering*, 8(6):866–883, 1996.

[6] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley Interscience, 1998.

[7] C. Clifton, M. Kantarcioglu, and J. Vaidya. Defining privacy for data mining. In *National Science Foundation Workshop on Next Generation Data Mining*, pages 126–133, Baltimore, MD, November 2002.

[8] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proc. of the 6th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 71–80, Boston, Massachusetts, 2000. ACM Press.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., New York, 2nd edition, 2001.

[10] M. H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2002.

[11] U. M. Fayyad, G. G. Grinstein, and A. Wierse, editors. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, San Francisco, CA, September 2001.

[12] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press, 1996.

[13] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[14] J. H. Friedman. Data Mining and Statistics: What's the Connection? Unpublished. www-stat.stanford.edu/∼jhf/ftp/dm-stat.ps, 1997.

[15] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu. Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha, editors, *Next Generation Data Mining*, pages 191–212. AAAI/MIT, 2003.

[16] C. Glymour, D. Madigan, D. Pregibon, and P. Smyth. Statistical Themes and Lessons for Data Mining. *Data Mining and Knowledge Discovery*, 1(1):11–28, 1997.

[17] R. L. Grossman, M. F. Hornick, and G. Meyer. Data mining standards initiatives. *Communications of the ACM*, 45(8):59–61, 2002.

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

[19] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering Data Streams: Theory and Practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, May/June 2003.

[20] J. Han, R. B. Altman, V. Kumar, H. Mannila, and D. Pregibon. Emerging scientific applications in data mining. *Communications of the ACM*, 45(8):54–58, 2002.

[21] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, 2001.

[22] D. J. Hand. Data Mining: Statistics and More? *The American Statistician*, 52(2): 112–118, 1998.

[23] D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, 2001.

[24] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, Prediction*. Springer, New York, 2001.

[25] M. Kantardzic. *Data Mining: Concepts, Models, Methods, and Algorithms*. Wiley-IEEE Press, Piscataway, NJ, 2003.

[26] H. Kargupta and P. K. Chan, editors. *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press, September 2002.

[27] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the Privacy Preserving Properties of Random Data Perturbation Techniques. In *Proc. of the 2003 IEEE Intl. Conf. on Data Mining*, pages 99–106, Melbourne, Florida, December 2003. IEEE Computer Society.

[28] D. Kifer, S. Ben-David, and J. Gehrke. Detecting Change in Data Streams. In *Proc. of the 30th VLDB Conf.*, pages 180–191, Toronto, Canada, 2004. Morgan Kaufmann.

[29] D. Lambert. What Use is Statistics for Massive Data? In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 54–62, 2000.

[30] M. H. C. Law, N. Zhang, and A. K. Jain. Nonlinear Manifold Learning for Data Streams. In *Proc. of the SIAM Intl. Conf. on Data Mining*, Lake Buena Vista, Florida, April 2004. SIAM.

[31] T. Mitchell. *Machine Learning*. McGraw-Hill, Boston, MA, 1997.

[32] S. Papadimitriou, A. Brockwell, and C. Faloutsos. Adaptive, unsupervised stream mining. *VLDB Journal*, 13(3):222–239, 2004.

[33] O. Parr Rud. *Data Mining Cookbook: Modeling Data for Marketing, Risk and Customer Relationship Management*. John Wiley & Sons, New York, NY, 2001.

[34] D. Pyle. *Business Modeling and Data Mining*. Morgan Kaufmann, San Francisco, CA, 2003.

[35] N. Ramakrishnan and A. Grama. Data Mining: From Serendipity to Science—Guest Editors' Introduction. *IEEE Computer*, 32(8):34–37, 1999.

[36] R. Roiger and M. Geatz. *Data Mining: A Tutorial Based Primer*. Addison-Wesley, 2002.

[37] P. Smyth. Breaking out of the Black-Box: Research Challenges in Data Mining. In *Proc. of the 2001 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2001.

[38] P. Smyth, D. Pregibon, and C. Faloutsos. Data-driven evolution of data mining algorithms. *Communications of the ACM*, 45(8):33–37, 2002.

[39] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1):50–57, 2004.

[40] J. T. L. Wang, M. J. Zaki, H. Toivonen, and D. E. Shasha, editors. *Data Mining in Bioinformatics*. Springer, September 2004.

[41] A. R. Webb. *Statistical Pattern Recognition*. John Wiley & Sons, 2nd edition, 2002.

[42] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[43] X. Wu, P. S. Yu, and G. Piatetsky-Shapiro. Data Mining: How Research Meets Practical Development? *Knowledge and Information Systems*, 5(2):248–261, 2003.

[44] M. J. Zaki and C.-T. Ho, editors. *Large-Scale Parallel Data Mining*. Springer, September 2002.

1.7 Exercises

1. Discuss whether or not each of the following activities is a data mining task.

(a) Dividing the customers of a company according to their gender.

(b) Dividing the customers of a company according to their profitability.

(c) Computing the total sales of a company.

(d) Sorting a student database based on student identification numbers.

(e) Predicting the outcomes of tossing a (fair) pair of dice.

(f) Predicting the future stock price of a company using historical records.

(g) Monitoring the heart rate of a patient for abnormalities.

(h) Monitoring seismic waves for earthquake activities.

(i) Extracting the frequencies of a sound wave.

2. Suppose that you are employed as a data mining consultant for an Internet search engine company. Describe how data mining can help the company by giving specific examples of how techniques, such as clustering, classification, association rule mining, and anomaly detection can be applied.

3. For each of the following data sets, explain whether or not data privacy is an important issue.

   (a) Census data collected from 1900–1950.

   (b) IP addresses and visit times of Web users who visit your Website.

   (c) Images from Earth-orbiting satellites.

   (d) Names and addresses of people from the telephone book.

   (e) Names and email addresses collected from the Web.

2

Data

This chapter discusses several data-related issues that are important for successful data mining:

The Type of Data Data sets differ in a number of ways. For example, the attributes used to describe data objects can be of different types—quantitative or qualitative—and data sets may have special characteristics; e.g., some data sets contain time series or objects with explicit relationships to one another. Not surprisingly, the type of data determines which tools and techniques can be used to analyze the data. Furthermore, new research in data mining is often driven by the need to accommodate new application areas and their new types of data.

The Quality of the Data Data is often far from perfect. While most data mining techniques can tolerate some level of imperfection in the data, a focus on understanding and improving data quality typically improves the quality of the resulting analysis. Data quality issues that often need to be addressed include the presence of noise and outliers; missing, inconsistent, or duplicate data; and data that is biased or, in some other way, unrepresentative of the phenomenon or population that the data is supposed to describe.

Preprocessing Steps to Make the Data More Suitable for Data Mining Often, the raw data must be processed in order to make it suitable for analysis. While one objective may be to improve data quality, other goals focus on modifying the data so that it better fits a specified data mining technique or tool. For example, a continuous attribute, e.g., length, may need to be transformed into an attribute with discrete categories, e.g., *short*, *medium*, or *long*, in order to apply a particular technique. As another example, the

number of attributes in a data set is often reduced because many techniques are more effective when the data has a relatively small number of attributes.

Analyzing Data in Terms of Its Relationships One approach to data analysis is to find relationships among the data objects and then perform the remaining analysis using these relationships rather than the data objects themselves. For instance, we can compute the similarity or distance between pairs of objects and then perform the analysis—clustering, classification, or anomaly detection—based on these similarities or distances. There are many such similarity or distance measures, and the proper choice depends on the type of data and the particular application.

Example 2.1 (An Illustration of Data-Related Issues). To further illustrate the importance of these issues, consider the following hypothetical situation. You receive an email from a medical researcher concerning a project that you are eager to work on.

> Hi,
>
> I've attached the data file that I mentioned in my previous email. Each line contains the information for a single patient and consists of five fields. We want to predict the last field using the other fields. I don't have time to provide any more information about the data since I'm going out of town for a couple of days, but hopefully that won't slow you down too much. And if you don't mind, could we meet when I get back to discuss your preliminary results? I might invite a few other members of my team.
>
> Thanks and see you in a couple of days.

Despite some misgivings, you proceed to analyze the data. The first few rows of the file are as follows:

```
012  232  33.5  0   10.7
020  121  16.9  2  210.1
027  165  24.0  0  427.6
⋮
```

A brief look at the data reveals nothing strange. You put your doubts aside and start the analysis. There are only 1000 lines, a smaller data file than you had hoped for, but two days later, you feel that you have made some progress. You arrive for the meeting, and while waiting for others to arrive, you strike

up a conversation with a statistician who is working on the project. When she learns that you have also been analyzing the data from the project, she asks if you would mind giving her a brief overview of your results.

Statistician: So, you got the data for all the patients?

Data Miner: Yes. I haven't had much time for analysis, but I do have a few interesting results.

Statistician: Amazing. There were so many data issues with this set of patients that I couldn't do much.

Data Miner: Oh? I didn't hear about any possible problems.

Statistician: Well, first there is field 5, the variable we want to predict. It's common knowledge among people who analyze this type of data that results are better if you work with the log of the values, but I didn't discover this until later. Was it mentioned to you?

Data Miner: No.

Statistician: But surely you heard about what happened to field 4? It's supposed to be measured on a scale from 1 to 10, with 0 indicating a missing value, but because of a data entry error, all 10's were changed into 0's. Unfortunately, since some of the patients have missing values for this field, it's impossible to say whether a 0 in this field is a real 0 or a 10. Quite a few of the records have that problem.

Data Miner: Interesting. Were there any other problems?

Statistician: Yes, fields 2 and 3 are basically the same, but I assume that you probably noticed that.

Data Miner: Yes, but these fields were only weak predictors of field 5.

Statistician: Anyway, given all those problems, I'm surprised you were able to accomplish anything.

Data Miner: True, but my results are really quite good. Field 1 is a very strong predictor of field 5. I'm surprised that this wasn't noticed before.

Statistician: What? Field 1 is just an identification number.

Data Miner: Nonetheless, my results speak for themselves.

Statistician: Oh, no! I just remembered. We assigned ID numbers after we sorted the records based on field 5. There is a strong connection, but it's meaningless. Sorry.

■

Although this scenario represents an extreme situation, it emphasizes the importance of "knowing your data." To that end, this chapter will address each of the four issues mentioned above, outlining some of the basic challenges and standard approaches.

2.1 Types of Data

A **data set** can often be viewed as a collection of **data objects**. Other names for a data object are *record*, *point*, *vector*, *pattern*, *event*, *case*, *sample*, *observation*, or *entity*. In turn, data objects are described by a number of **attributes** that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred. Other names for an attribute are *variable*, *characteristic*, *field*, *feature*, or *dimension*.

Example 2.2 (Student Information). Often, a data set is a file, in which the objects are records (or rows) in the file and each field (or column) corresponds to an attribute. For example, Table 2.1 shows a data set that consists of student information. Each row corresponds to a student and each column is an attribute that describes some aspect of a student, such as grade point average (GPA) or identification number (ID).

Table 2.1. A sample data set containing student information.

Student ID	Year	Grade Point Average (GPA)	...
	⋮		
1034262	Senior	3.24	...
1052663	Sophomore	3.51	...
1082246	Freshman	3.62	...
	⋮		

■

Although record-based data sets are common, either in flat files or relational database systems, there are other important types of data sets and systems for storing data. In Section 2.1.2, we will discuss some of the types of data sets that are commonly encountered in data mining. However, we first consider attributes.

2.1.1 Attributes and Measurement

In this section we address the issue of describing data by considering what types of attributes are used to describe data objects. We first define an attribute, then consider what we mean by the type of an attribute, and finally describe the types of attributes that are commonly encountered.

What Is an attribute?

We start with a more detailed definition of an attribute.

Definition 2.1. An **attribute** is a property or characteristic of an object that may vary, either from one object to another or from one time to another.

For example, eye color varies from person to person, while the temperature of an object varies over time. Note that eye color is a symbolic attribute with a small number of possible values $\{brown, black, blue, green, hazel, etc.\}$, while temperature is a numerical attribute with a potentially unlimited number of values.

At the most basic level, attributes are not about numbers or symbols. However, to discuss and more precisely analyze the characteristics of objects, we assign numbers or symbols to them. To do this in a well-defined way, we need a measurement scale.

Definition 2.2. A **measurement scale** is a rule (function) that associates a numerical or symbolic value with an attribute of an object.

Formally, the process of **measurement** is the application of a measurement scale to associate a value with a particular attribute of a specific object. While this may seem a bit abstract, we engage in the process of measurement all the time. For instance, we step on a bathroom scale to determine our weight, we classify someone as male or female, or we count the number of chairs in a room to see if there will be enough to seat all the people coming to a meeting. In all these cases, the "physical value" of an attribute of an object is mapped to a numerical or symbolic value.

With this background, we can now discuss the type of an attribute, a concept that is important in determining if a particular data analysis technique is consistent with a specific type of attribute.

The Type of an Attribute

It should be apparent from the previous discussion that the properties of an attribute need not be the same as the properties of the values used to mea-

sure it. In other words, the values used to represent an attribute may have properties that are not properties of the attribute itself, and vice versa. This is illustrated with two examples.

Example 2.3 (Employee Age and ID Number). Two attributes that might be associated with an employee are *ID* and *age* (in years). Both of these attributes can be represented as integers. However, while it is reasonable to talk about the average age of an employee, it makes no sense to talk about the average employee ID. Indeed, the only aspect of employees that we want to capture with the ID attribute is that they are distinct. Consequently, the only valid operation for employee IDs is to test whether they are equal. There is no hint of this limitation, however, when integers are used to represent the employee ID attribute. For the age attribute, the properties of the integers used to represent age are very much the properties of the attribute. Even so, the correspondence is not complete since, for example, ages have a maximum, while integers do not. ■

Example 2.4 (Length of Line Segments). Consider Figure 2.1, which shows some objects—line segments—and how the length attribute of these objects can be mapped to numbers in two different ways. Each successive line segment, going from the top to the bottom, is formed by appending the topmost line segment to itself. Thus, the second line segment from the top is formed by appending the topmost line segment to itself twice, the third line segment from the top is formed by appending the topmost line segment to itself three times, and so forth. In a very real (physical) sense, all the line segments are multiples of the first. This fact is captured by the measurements on the right-hand side of the figure, but not by those on the left hand-side. More specifically, the measurement scale on the left-hand side captures only the ordering of the length attribute, while the scale on the right-hand side captures both the ordering and additivity properties. Thus, an attribute can be measured in a way that does not capture all the properties of the attribute. ■

The type of an attribute should tell us what properties of the attribute are reflected in the values used to measure it. Knowing the type of an attribute is important because it tells us which properties of the measured values are consistent with the underlying properties of the attribute, and therefore, it allows us to avoid foolish actions, such as computing the average employee ID. Note that it is common to refer to the type of an attribute as the **type of a measurement scale**.

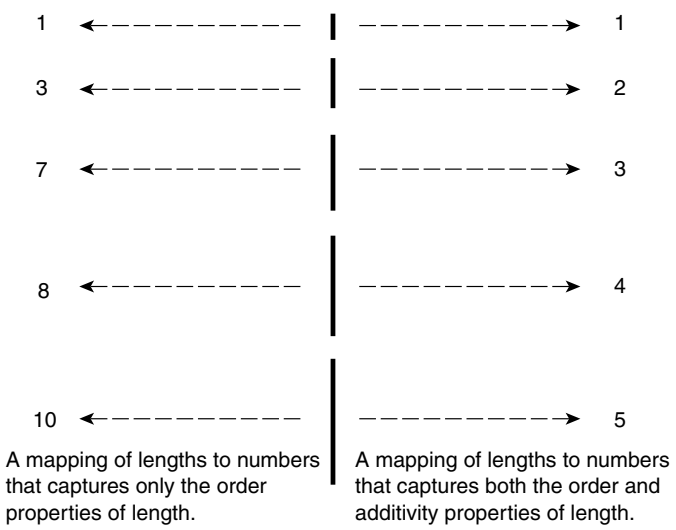


Figure 2.1. The measurement of the length of line segments on two different scales of measurement.

The Different Types of Attributes

A useful (and simple) way to specify the type of an attribute is to identify the properties of numbers that correspond to underlying properties of the attribute. For example, an attribute such as length has many of the properties of numbers. It makes sense to compare and order objects by length, as well as to talk about the differences and ratios of length. The following properties (operations) of numbers are typically used to describe attributes.

1. **Distinctness** $=$ and $\neq$
2. **Order** $<$, $\leq$, $>$, and $\geq$
3. **Addition** $+$ and $-$
4. **Multiplication** $*$ and $/$

Given these properties, we can define four types of attributes: **nominal**, **ordinal**, **interval**, and **ratio**. Table 2.2 gives the definitions of these types, along with information about the statistical operations that are valid for each type. Each attribute type possesses all of the properties and operations of the attribute types above it. Consequently, any property or operation that is valid for nominal, ordinal, and interval attributes is also valid for ratio attributes. In other words, the definition of the attribute types is cumulative. However,

Table 2.2. Different attribute types.

Attribute Type		Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, χ^2 test
	Ordinal	The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation Pearson's correlation, t and F tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

this does not mean that the operations appropriate for one attribute type are appropriate for the attribute types above it.

Nominal and ordinal attributes are collectively referred to as **categorical** or **qualitative** attributes. As the name suggests, qualitative attributes, such as employee ID, lack most of the properties of numbers. Even if they are represented by numbers, i.e., integers, they should be treated more like symbols. The remaining two types of attributes, interval and ratio, are collectively referred to as **quantitative** or **numeric** attributes. Quantitative attributes are represented by numbers and have most of the properties of numbers. Note that quantitative attributes can be integer-valued or continuous.

The types of attributes can also be described in terms of transformations that do not change the meaning of an attribute. Indeed, S. Smith Stevens, the psychologist who originally defined the types of attributes shown in Table 2.2, defined them in terms of these **permissible transformations**. For example,

Table 2.3. Transformations that define attribute levels.

Attribute Type		Transformation	Comment
Categorical (Qualitative)	Nominal	Any one-to-one mapping, e.g., a permutation of values	If all employee ID numbers are reassigned, it will not make any difference.
	Ordinal	An order-preserving change of values, i.e., $new\_value = f(old\_value)$, where f is a monotonic function.	An attribute encompassing the notion of good, better, best can be represented equally well by the values $\{1, 2, 3\}$ or by $\{0.5, 1, 10\}$.
Numeric (Quantitative)	Interval	$new\_value = a * old\_value + b$, a and b constants.	The Fahrenheit and Celsius temperature scales differ in the location of their zero value and the size of a degree (unit).
	Ratio	$new\_value = a * old\_value$	Length can be measured in meters or feet.

the meaning of a length attribute is unchanged if it is measured in meters instead of feet.

The statistical operations that make sense for a particular type of attribute are those that will yield the same results when the attribute is transformed using a transformation that preserves the attribute's meaning. To illustrate, the average length of a set of objects is different when measured in meters rather than in feet, but both averages represent the same length. Table 2.3 shows the permissible (meaning-preserving) transformations for the four attribute types of Table 2.2.

Example 2.5 (Temperature Scales). Temperature provides a good illustration of some of the concepts that have been described. First, temperature can be either an interval or a ratio attribute, depending on its measurement scale. When measured on the Kelvin scale, a temperature of 2° is, in a physically meaningful way, twice that of a temperature of 1°. This is not true when temperature is measured on either the Celsius or Fahrenheit scales, because, physically, a temperature of 1° Fahrenheit (Celsius) is not much different than a temperature of 2° Fahrenheit (Celsius). The problem is that the zero points of the Fahrenheit and Celsius scales are, in a physical sense, arbitrary, and therefore, the ratio of two Celsius or Fahrenheit temperatures is not physically meaningful. ■

Describing Attributes by the Number of Values

An independent way of distinguishing between attributes is by the number of values they can take.

Discrete A discrete attribute has a finite or countably infinite set of values. Such attributes can be categorical, such as zip codes or ID numbers, or numeric, such as counts. Discrete attributes are often represented using integer variables. **Binary attributes** are a special case of discrete attributes and assume only two values, e.g., true/false, yes/no, male/female, or 0/1. Binary attributes are often represented as Boolean variables, or as integer variables that only take the values 0 or 1.

Continuous A continuous attribute is one whose values are real numbers. Examples include attributes such as temperature, height, or weight. Continuous attributes are typically represented as floating-point variables. Practically, real values can only be measured and represented with limited precision.

In theory, any of the measurement scale types—nominal, ordinal, interval, and ratio—could be combined with any of the types based on the number of attribute values—binary, discrete, and continuous. However, some combinations occur only infrequently or do not make much sense. For instance, it is difficult to think of a realistic data set that contains a continuous binary attribute. Typically, nominal and ordinal attributes are binary or discrete, while interval and ratio attributes are continuous. However, **count attributes**, which are discrete, are also ratio attributes.

Asymmetric Attributes

For asymmetric attributes, only presence—a non-zero attribute value—is regarded as important. Consider a data set where each object is a student and each attribute records whether or not a student took a particular course at a university. For a specific student, an attribute has a value of 1 if the student took the course associated with that attribute and a value of 0 otherwise. Because students take only a small fraction of all available courses, most of the values in such a data set would be 0. Therefore, it is more meaningful and more efficient to focus on the non-zero values. To illustrate, if students are compared on the basis of the courses they don't take, then most students would seem very similar, at least if the number of courses is large. Binary attributes where only non-zero values are important are called **asymmetric**

binary attributes. This type of attribute is particularly important for association analysis, which is discussed in Chapter 6. It is also possible to have discrete or continuous asymmetric features. For instance, if the number of credits associated with each course is recorded, then the resulting data set will consist of **asymmetric discrete** or **continuous attributes**.

2.1.2 Types of Data Sets

There are many types of data sets, and as the field of data mining develops and matures, a greater variety of data sets become available for analysis. In this section, we describe some of the most common types. For convenience, we have grouped the types of data sets into three groups: record data, graph-based data, and ordered data. These categories do not cover all possibilities and other groupings are certainly possible.

General Characteristics of Data Sets

Before providing details of specific kinds of data sets, we discuss three characteristics that apply to many data sets and have a significant impact on the data mining techniques that are used: dimensionality, sparsity, and resolution.

Dimensionality The dimensionality of a data set is the number of attributes that the objects in the data set possess. Data with a small number of dimensions tends to be qualitatively different than moderate or high-dimensional data. Indeed, the difficulties associated with analyzing high-dimensional data are sometimes referred to as the **curse of dimensionality**. Because of this, an important motivation in preprocessing the data is **dimensionality reduction**. These issues are discussed in more depth later in this chapter and in Appendix B.

Sparsity For some data sets, such as those with asymmetric features, most attributes of an object have values of 0; in many cases, fewer than 1% of the entries are non-zero. In practical terms, sparsity is an advantage because usually only the non-zero values need to be stored and manipulated. This results in significant savings with respect to computation time and storage. Furthermore, some data mining algorithms work well only for sparse data.

Resolution It is frequently possible to obtain data at different levels of resolution, and often the properties of the data are different at different resolutions. For instance, the surface of the Earth seems very uneven at a resolution of a

few meters, but is relatively smooth at a resolution of tens of kilometers. The patterns in the data also depend on the level of resolution. If the resolution is too fine, a pattern may not be visible or may be buried in noise; if the resolution is too coarse, the pattern may disappear. For example, variations in atmospheric pressure on a scale of hours reflect the movement of storms and other weather systems. On a scale of months, such phenomena are not detectable.

Record Data

Much data mining work assumes that the data set is a collection of records (data objects), each of which consists of a fixed set of data fields (attributes). See Figure 2.2(a). For the most basic form of record data, there is no explicit relationship among records or data fields, and every record (object) has the same set of attributes. Record data is usually stored either in **flat** files or in relational databases. Relational databases are certainly more than a collection of records, but data mining often does not use any of the additional information available in a relational database. Rather, the database serves as a convenient place to find records. Different types of record data are described below and are illustrated in Figure 2.2.

Transaction or Market Basket Data Transaction data is a special type of record data, where each record (transaction) involves a set of items. Consider a grocery store. The set of products purchased by a customer during one shopping trip constitutes a transaction, while the individual products that were purchased are the items. This type of data is called **market basket data** because the items in each record are the products in a person's "market basket." Transaction data is a collection of sets of items, but it can be viewed as a set of records whose fields are asymmetric attributes. Most often, the attributes are binary, indicating whether or not an item was purchased, but more generally, the attributes can be discrete or continuous, such as the number of items purchased or the amount spent on those items. Figure 2.2(b) shows a sample transaction data set. Each row represents the purchases of a particular customer at a particular time.

The Data Matrix If the data objects in a collection of data all have the same fixed set of numeric attributes, then the data objects can be thought of as points (vectors) in a multidimensional space, where each dimension represents a distinct attribute describing the object. A set of such data objects can be interpreted as an m by n matrix, where there are m rows, one for each object,

Tid	Refund	Marital Status	Taxable Income	Defaulted Borrower
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

(a) Record data.

TID	*ITEMS*
1	Bread, Soda, Milk
2	Beer, Bread
3	Beer, Soda, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Soda, Diaper, Milk

(b) Transaction data.

Projection of x Load	Projection of y Load	Distance	Load	Thickness
10.23	5.27	15.22	27	1.2
12.65	6.25	16.22	22	1.1
13.54	7.23	17.34	23	1.2
14.27	8.43	18.45	25	0.9

(c) Data matrix.

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

(d) Document-term matrix.

Figure 2.2. Different variations of record data.

and n columns, one for each attribute. (A representation that has data objects as columns and attributes as rows is also fine.) This matrix is called a **data matrix** or a **pattern matrix**. A data matrix is a variation of record data, but because it consists of numeric attributes, standard matrix operation can be applied to transform and manipulate the data. Therefore, the data matrix is the standard data format for most statistical data. Figure 2.2(c) shows a sample data matrix.

The Sparse Data Matrix A sparse data matrix is a special case of a data matrix in which the attributes are of the same type and are asymmetric; i.e., only non-zero values are important. Transaction data is an example of a sparse data matrix that has only 0–1 entries. Another common example is document data. In particular, if the order of the terms (words) in a document is ignored,

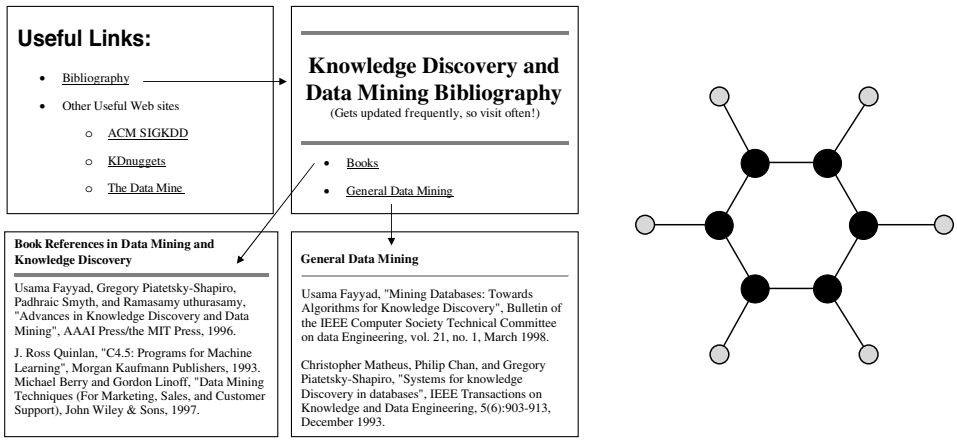
then a document can be represented as a term vector, where each term is a component (attribute) of the vector and the value of each component is the number of times the corresponding term occurs in the document. This representation of a collection of documents is often called a **document-term matrix**. Figure 2.2(d) shows a sample document-term matrix. The documents are the rows of this matrix, while the terms are the columns. In practice, only the non-zero entries of sparse data matrices are stored.

Graph-Based Data

A graph can sometimes be a convenient and powerful representation for data. We consider two specific cases: (1) the graph captures relationships among data objects and (2) the data objects themselves are represented as graphs.

Data with Relationships among Objects The relationships among objects frequently convey important information. In such cases, the data is often represented as a graph. In particular, the data objects are mapped to nodes of the graph, while the relationships among objects are captured by the links between objects and link properties, such as direction and weight. Consider Web pages on the World Wide Web, which contain both text and links to other pages. In order to process search queries, Web search engines collect and process Web pages to extract their contents. It is well known, however, that the links to and from each page provide a great deal of information about the relevance of a Web page to a query, and thus, must also be taken into consideration. Figure 2.3(a) shows a set of linked Web pages.

Data with Objects That Are Graphs If objects have structure, that is, the objects contain subobjects that have relationships, then such objects are frequently represented as graphs. For example, the structure of chemical compounds can be represented by a graph, where the nodes are atoms and the links between nodes are chemical bonds. Figure 2.3(b) shows a ball-and-stick diagram of the chemical compound benzene, which contains atoms of carbon (black) and hydrogen (gray). A graph representation makes it possible to determine which substructures occur frequently in a set of compounds and to ascertain whether the presence of any of these substructures is associated with the presence or absence of certain chemical properties, such as melting point or heat of formation. Substructure mining, which is a branch of data mining that analyzes such data, is considered in Section 7.5.



(a) Linked Web pages. (b) Benzene molecule.

Figure 2.3. Different variations of graph data.

Ordered Data

For some types of data, the attributes have relationships that involve order in time or space. Different types of ordered data are described next and are shown in Figure 2.4.

Sequential Data Sequential data, also referred to as **temporal data**, can be thought of as an extension of record data, where each record has a time associated with it. Consider a retail transaction data set that also stores the time at which the transaction took place. This time information makes it possible to find patterns such as "candy sales peak before Halloween." A time can also be associated with each attribute. For example, each record could be the purchase history of a customer, with a listing of items purchased at different times. Using this information, it is possible to find patterns such as "people who buy DVD players tend to buy DVDs in the period immediately following the purchase."

Figure 2.4(a) shows an example of sequential transaction data. There are five different times—*t1*, *t2*, *t3*, *t4*, and *t5*; three different customers—C1,

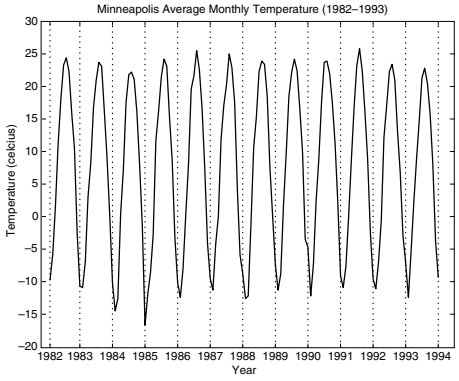
Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

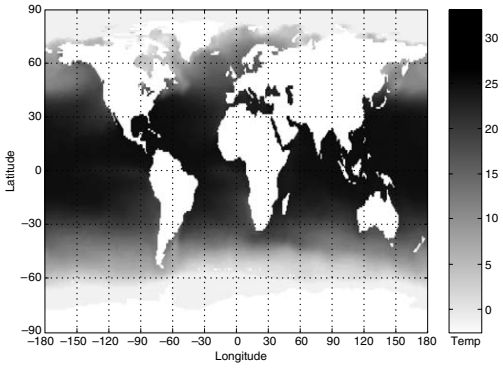
(a) Sequential transaction data.

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

(b) Genomic sequence data.



(c) Temperature time series.



(d) Spatial temperature data.

Figure 2.4. Different variations of ordered data.

C2, and C3; and five different items—A, B, C, D, and E. In the top table, each row corresponds to the items purchased at a particular time by each customer. For instance, at time *t3*, customer C2 purchased items A and D. In the bottom table, the same information is displayed, but each row corresponds to a particular customer. Each row contains information on each transaction involving the customer, where a transaction is considered to be a set of items and the time at which those items were purchased. For example, customer C3 bought items A and C at time *t2*.

Sequence Data Sequence data consists of a data set that is a sequence of individual entities, such as a sequence of words or letters. It is quite similar to sequential data, except that there are no time stamps; instead, there are positions in an ordered sequence. For example, the genetic information of plants and animals can be represented in the form of sequences of nucleotides that are known as genes. Many of the problems associated with genetic sequence data involve predicting similarities in the structure and function of genes from similarities in nucleotide sequences. Figure 2.4(b) shows a section of the human genetic code expressed using the four nucleotides from which all DNA is constructed: A, T, G, and C.

Time Series Data Time series data is a special type of sequential data in which each record is a **time series**, i.e., a series of measurements taken over time. For example, a financial data set might contain objects that are time series of the daily prices of various stocks. As another example, consider Figure 2.4(c), which shows a time series of the average monthly temperature for Minneapolis during the years 1982 to 1994. When working with temporal data, it is important to consider **temporal autocorrelation**; i.e., if two measurements are close in time, then the values of those measurements are often very similar.

Spatial Data Some objects have spatial attributes, such as positions or areas, as well as other types of attributes. An example of spatial data is weather data (precipitation, temperature, pressure) that is collected for a variety of geographical locations. An important aspect of spatial data is **spatial autocorrelation**; i.e., objects that are physically close tend to be similar in other ways as well. Thus, two points on the Earth that are close to each other usually have similar values for temperature and rainfall.

Important examples of spatial data are the science and engineering data sets that are the result of measurements or model output taken at regularly or irregularly distributed points on a two- or three-dimensional grid or mesh. For instance, Earth science data sets record the temperature or pressure measured at points (grid cells) on latitude–longitude spherical grids of various resolutions, e.g., 1° by 1°. (See Figure 2.4(d).) As another example, in the simulation of the flow of a gas, the speed and direction of flow can be recorded for each grid point in the simulation.

Handling Non-Record Data

Most data mining algorithms are designed for record data or its variations, such as transaction data and data matrices. Record-oriented techniques can be applied to non-record data by extracting features from data objects and using these features to create a record corresponding to each object. Consider the chemical structure data that was described earlier. Given a set of common substructures, each compound can be represented as a record with binary attributes that indicate whether a compound contains a specific substructure. Such a representation is actually a transaction data set, where the transactions are the compounds and the items are the substructures.

In some cases, it is easy to represent the data in a record format, but this type of representation does not capture all the information in the data. Consider spatio-temporal data consisting of a time series from each point on a spatial grid. This data is often stored in a data matrix, where each row represents a location and each column represents a particular point in time. However, such a representation does not explicitly capture the time relationships that are present among attributes and the spatial relationships that exist among objects. This does not mean that such a representation is inappropriate, but rather that these relationships must be taken into consideration during the analysis. For example, it would not be a good idea to use a data mining technique that assumes the attributes are statistically independent of one another.

2.2 Data Quality

Data mining applications are often applied to data that was collected for another purpose, or for future, but unspecified applications. For that reason, data mining cannot usually take advantage of the significant benefits of "addressing quality issues at the source." In contrast, much of statistics deals with the design of experiments or surveys that achieve a prespecified level of data quality. Because preventing data quality problems is typically not an option, data mining focuses on (1) the detection and correction of data quality problems and (2) the use of algorithms that can tolerate poor data quality. The first step, detection and correction, is often called **data cleaning**.

The following sections discuss specific aspects of data quality. The focus is on measurement and data collection issues, although some application-related issues are also discussed.

2.2.1 Measurement and Data Collection Issues

It is unrealistic to expect that data will be perfect. There may be problems due to human error, limitations of measuring devices, or flaws in the data collection process. Values or even entire data objects may be missing. In other cases, there may be spurious or duplicate objects; i.e., multiple data objects that all correspond to a single "real" object. For example, there might be two different records for a person who has recently lived at two different addresses. Even if all the data is present and "looks fine," there may be inconsistencies—a person has a height of 2 meters, but weighs only 2 kilograms.

In the next few sections, we focus on aspects of data quality that are related to data measurement and collection. We begin with a definition of measurement and data collection errors and then consider a variety of problems that involve measurement error: noise, artifacts, bias, precision, and accuracy. We conclude by discussing data quality issues that may involve both measurement and data collection problems: outliers, missing and inconsistent values, and duplicate data.

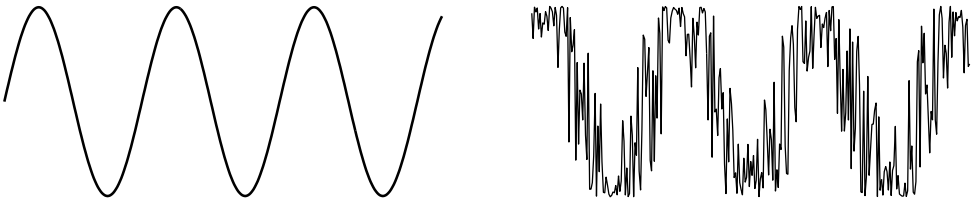
Measurement and Data Collection Errors

The term **measurement error** refers to any problem resulting from the measurement process. A common problem is that the value recorded differs from the true value to some extent. For continuous attributes, the numerical difference of the measured and true value is called the **error**. The term **data collection error** refers to errors such as omitting data objects or attribute values, or inappropriately including a data object. For example, a study of animals of a certain species might include animals of a related species that are similar in appearance to the species of interest. Both measurement errors and data collection errors can be either systematic or random.

We will only consider general types of errors. Within particular domains, there are certain types of data errors that are commonplace, and there often exist well-developed techniques for detecting and/or correcting these errors. For example, keyboard errors are common when data is entered manually, and as a result, many data entry programs have techniques for detecting and, with human intervention, correcting such errors.

Noise and Artifacts

Noise is the random component of a measurement error. It may involve the distortion of a value or the addition of spurious objects. Figure 2.5 shows a time series before and after it has been disrupted by random noise. If a bit



(a) Time series. (b) Time series with noise.

Figure 2.5. Noise in a time series context.

(a) Three groups of points. (b) With noise points (+) added.

Figure 2.6. Noise in a spatial context.

more noise were added to the time series, its shape would be lost. Figure 2.6 shows a set of data points before and after some noise points (indicated by '+'s) have been added. Notice that some of the noise points are intermixed with the non-noise points.

The term noise is often used in connection with data that has a spatial or temporal component. In such cases, techniques from signal or image processing can frequently be used to reduce noise and thus, help to discover patterns (signals) that might be "lost in the noise." Nonetheless, the elimination of noise is frequently difficult, and much work in data mining focuses on devising **robust algorithms** that produce acceptable results even when noise is present.

Data errors may be the result of a more deterministic phenomenon, such as a streak in the same place on a set of photographs. Such deterministic distortions of the data are often referred to as **artifacts**.

Precision, Bias, and Accuracy

In statistics and experimental science, the quality of the measurement process and the resulting data are measured by precision and bias. We provide the standard definitions, followed by a brief discussion. For the following definitions, we assume that we make repeated measurements of the same underlying quantity and use this set of values to calculate a mean (average) value that serves as our estimate of the true value.

Definition 2.3 (Precision). The closeness of repeated measurements (of the same quantity) to one another.

Definition 2.4 (Bias). A systematic variation of measurements from the quantity being measured.

Precision is often measured by the standard deviation of a set of values, while bias is measured by taking the difference between the mean of the set of values and the known value of the quantity being measured. Bias can only be determined for objects whose measured quantity is known by means external to the current situation. Suppose that we have a standard laboratory weight with a mass of 1g and want to assess the precision and bias of our new laboratory scale. We weigh the mass five times, and obtain the following five values: $\{1.015, 0.990, 1.013, 1.001, 0.986\}$. The mean of these values is 1.001, and hence, the bias is 0.001. The precision, as measured by the standard deviation, is 0.013.

It is common to use the more general term, **accuracy**, to refer to the degree of measurement error in data.

Definition 2.5 (Accuracy). The closeness of measurements to the true value of the quantity being measured.

Accuracy depends on precision and bias, but since it is a general concept, there is no specific formula for accuracy in terms of these two quantities.

One important aspect of accuracy is the use of **significant digits**. The goal is to use only as many digits to represent the result of a measurement or calculation as are justified by the precision of the data. For example, if the length of an object is measured with a meter stick whose smallest markings are millimeters, then we should only record the length of data to the nearest millimeter. The precision of such a measurement would be $\pm$ 0.5mm. We do not

review the details of working with significant digits, as most readers will have encountered them in previous courses, and they are covered in considerable depth in science, engineering, and statistics textbooks.

Issues such as significant digits, precision, bias, and accuracy are sometimes overlooked, but they are important for data mining as well as statistics and science. Many times, data sets do not come with information on the precision of the data, and furthermore, the programs used for analysis return results without any such information. Nonetheless, without some understanding of the accuracy of the data and the results, an analyst runs the risk of committing serious data analysis blunders.

Outliers

Outliers are either (1) data objects that, in some sense, have characteristics that are different from most of the other data objects in the data set, or (2) values of an attribute that are unusual with respect to the typical values for that attribute. Alternatively, we can speak of **anomalous** objects or values. There is considerable leeway in the definition of an outlier, and many different definitions have been proposed by the statistics and data mining communities. Furthermore, it is important to distinguish between the notions of noise and outliers. Outliers can be legitimate data objects or values. Thus, unlike noise, outliers may sometimes be of interest. In fraud and network intrusion detection, for example, the goal is to find unusual objects or events from among a large number of normal ones. Chapter 10 discusses anomaly detection in more detail.

Missing Values

It is not unusual for an object to be missing one or more attribute values. In some cases, the information was not collected; e.g., some people decline to give their age or weight. In other cases, some attributes are not applicable to all objects; e.g., often, forms have conditional parts that are filled out only when a person answers a previous question in a certain way, but for simplicity, all fields are stored. Regardless, missing values should be taken into account during the data analysis.

There are several strategies (and variations on these strategies) for dealing with missing data, each of which may be appropriate in certain circumstances. These strategies are listed next, along with an indication of their advantages and disadvantages.

Eliminate Data Objects or Attributes A simple and effective strategy is to eliminate objects with missing values. However, even a partially specified data object contains some information, and if many objects have missing values, then a reliable analysis can be difficult or impossible. Nonetheless, if a data set has only a few objects that have missing values, then it may be expedient to omit them. A related strategy is to eliminate attributes that have missing values. This should be done with caution, however, since the eliminated attributes may be the ones that are critical to the analysis.

Estimate Missing Values Sometimes missing data can be reliably estimated. For example, consider a time series that changes in a reasonably smooth fashion, but has a few, widely scattered missing values. In such cases, the missing values can be estimated (interpolated) by using the remaining values. As another example, consider a data set that has many similar data points. In this situation, the attribute values of the points closest to the point with the missing value are often used to estimate the missing value. If the attribute is continuous, then the average attribute value of the nearest neighbors is used; if the attribute is categorical, then the most commonly occurring attribute value can be taken. For a concrete illustration, consider precipitation measurements that are recorded by ground stations. For areas not containing a ground station, the precipitation can be estimated using values observed at nearby ground stations.

Ignore the Missing Value during Analysis Many data mining approaches can be modified to ignore missing values. For example, suppose that objects are being clustered and the similarity between pairs of data objects needs to be calculated. If one or both objects of a pair have missing values for some attributes, then the similarity can be calculated by using only the attributes that do not have missing values. It is true that the similarity will only be approximate, but unless the total number of attributes is small or the number of missing values is high, this degree of inaccuracy may not matter much. Likewise, many classification schemes can be modified to work with missing values.

Inconsistent Values

Data can contain inconsistent values. Consider an address field, where both a zip code and city are listed, but the specified zip code area is not contained in that city. It may be that the individual entering this information transposed two digits, or perhaps a digit was misread when the information was scanned

from a handwritten form. Regardless of the cause of the inconsistent values, it is important to detect and, if possible, correct such problems.

Some types of inconsistences are easy to detect. For instance, a person's height should not be negative. In other cases, it can be necessary to consult an external source of information. For example, when an insurance company processes claims for reimbursement, it checks the names and addresses on the reimbursement forms against a database of its customers.

Once an inconsistency has been detected, it is sometimes possible to correct the data. A product code may have "check" digits, or it may be possible to double-check a product code against a list of known product codes, and then correct the code if it is incorrect, but close to a known code. The correction of an inconsistency requires additional or redundant information.

Example 2.6 (Inconsistent Sea Surface Temperature). This example illustrates an inconsistency in actual time series data that measures the sea surface temperature (SST) at various points on the ocean. SST data was originally collected using ocean-based measurements from ships or buoys, but more recently, satellites have been used to gather the data. To create a long-term data set, both sources of data must be used. However, because the data comes from different sources, the two parts of the data are subtly different. This discrepancy is visually displayed in Figure 2.7, which shows the correlation of SST values between pairs of years. If a pair of years has a positive correlation, then the location corresponding to the pair of years is colored white; otherwise it is colored black. (Seasonal variations were removed from the data since, otherwise, all the years would be highly correlated.) There is a distinct change in behavior where the data has been put together in 1983. Years within each of the two groups, 1958–1982 and 1983–1999, tend to have a positive correlation with one another, but a negative correlation with years in the other group. This does not mean that this data should not be used, only that the analyst should consider the potential impact of such discrepancies on the data mining analysis. ■

Duplicate Data

A data set may include data objects that are duplicates, or almost duplicates, of one another. Many people receive duplicate mailings because they appear in a database multiple times under slightly different names. To detect and eliminate such duplicates, two main issues must be addressed. First, if there are two objects that actually represent a single object, then the values of corresponding attributes may differ, and these inconsistent values must be

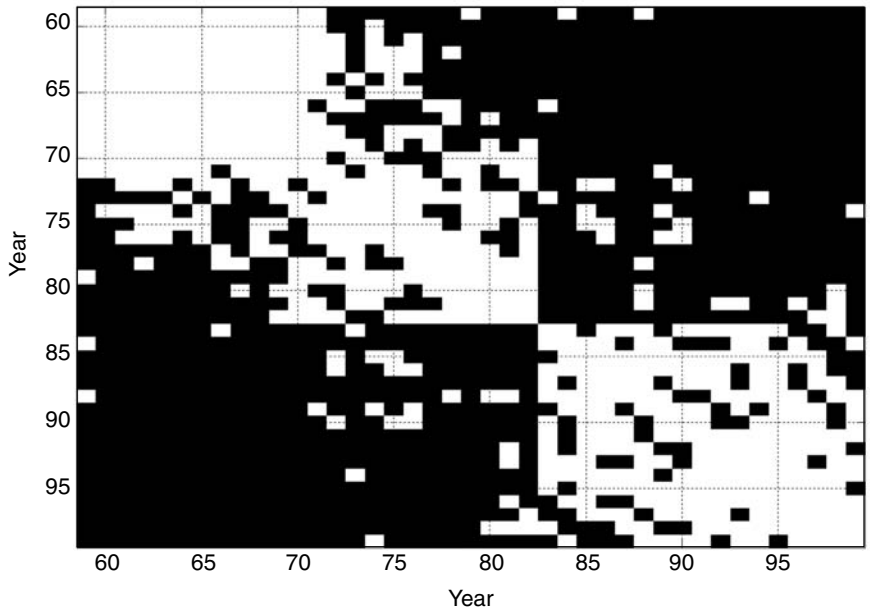


Figure 2.7. Correlation of SST data between pairs of years. White areas indicate positive correlation. Black areas indicate negative correlation.

resolved. Second, care needs to be taken to avoid accidentally combining data objects that are similar, but not duplicates, such as two distinct people with identical names. The term **deduplication** is often used to refer to the process of dealing with these issues.

In some cases, two or more objects are identical with respect to the attributes measured by the database, but they still represent different objects. Here, the duplicates are legitimate, but may still cause problems for some algorithms if the possibility of identical objects is not specifically accounted for in their design. An example of this is given in Exercise 13 on page 91.

2.2.2 Issues Related to Applications

Data quality issues can also be considered from an application viewpoint as expressed by the statement "data is of high quality if it is suitable for its intended use." This approach to data quality has proven quite useful, particularly in business and industry. A similar viewpoint is also present in statistics and the experimental sciences, with their emphasis on the careful design of experiments to collect the data relevant to a specific hypothesis. As with quality

issues at the measurement and data collection level, there are many issues that are specific to particular applications and fields. Again, we consider only a few of the general issues.

Timeliness Some data starts to age as soon as it has been collected. In particular, if the data provides a snapshot of some ongoing phenomenon or process, such as the purchasing behavior of customers or Web browsing patterns, then this snapshot represents reality for only a limited time. If the data is out of date, then so are the models and patterns that are based on it.

Relevance The available data must contain the information necessary for the application. Consider the task of building a model that predicts the accident rate for drivers. If information about the age and gender of the driver is omitted, then it is likely that the model will have limited accuracy unless this information is indirectly available through other attributes.

Making sure that the objects in a data set are relevant is also challenging. A common problem is **sampling bias**, which occurs when a sample does not contain different types of objects in proportion to their actual occurrence in the population. For example, survey data describes only those who respond to the survey. (Other aspects of sampling are discussed further in Section 2.3.2.) Because the results of a data analysis can reflect only the data that is present, sampling bias will typically result in an erroneous analysis.

Knowledge about the Data Ideally, data sets are accompanied by documentation that describes different aspects of the data; the quality of this documentation can either aid or hinder the subsequent analysis. For example, if the documentation identifies several attributes as being strongly related, these attributes are likely to provide highly redundant information, and we may decide to keep just one. (Consider sales tax and purchase price.) If the documentation is poor, however, and fails to tell us, for example, that the missing values for a particular field are indicated with a -9999, then our analysis of the data may be faulty. Other important characteristics are the precision of the data, the type of features (nominal, ordinal, interval, ratio), the scale of measurement (e.g., meters or feet for length), and the origin of the data.

2.3 Data Preprocessing

In this section, we address the issue of which preprocessing steps should be applied to make the data more suitable for data mining. Data preprocessing

is a broad area and consists of a number of different strategies and techniques that are interrelated in complex ways. We will present some of the most important ideas and approaches, and try to point out the interrelationships among them. Specifically, we will discuss the following topics:

- Aggregation
- Sampling
- Dimensionality reduction
- Feature subset selection
- Feature creation
- Discretization and binarization
- Variable transformation

Roughly speaking, these items fall into two categories: selecting data objects and attributes for the analysis or creating/changing the attributes. In both cases the goal is to improve the data mining analysis with respect to time, cost, and quality. Details are provided in the following sections.

A quick note on terminology: In the following, we sometimes use synonyms for attribute, such as feature or variable, in order to follow common usage.

2.3.1 Aggregation

Sometimes "less is more" and this is the case with **aggregation**, the combining of two or more objects into a single object. Consider a data set consisting of transactions (data objects) recording the daily sales of products in various store locations (Minneapolis, Chicago, Paris, ...) for different days over the course of a year. See Table 2.4. One way to aggregate transactions for this data set is to replace all the transactions of a single store with a single storewide transaction. This reduces the hundreds or thousands of transactions that occur daily at a specific store to a single daily transaction, and the number of data objects is reduced to the number of stores.

An obvious issue is how an aggregate transaction is created; i.e., how the values of each attribute are combined across all the records corresponding to a particular location to create the aggregate transaction that represents the sales of a single store or date. Quantitative attributes, such as price, are typically aggregated by taking a sum or an average. A qualitative attribute, such as item, can either be omitted or summarized as the set of all the items that were sold at that location.

The data in Table 2.4 can also be viewed as a multidimensional array, where each attribute is a dimension. From this viewpoint, aggregation is the

Table 2.4. Data set containing information about customer purchases.

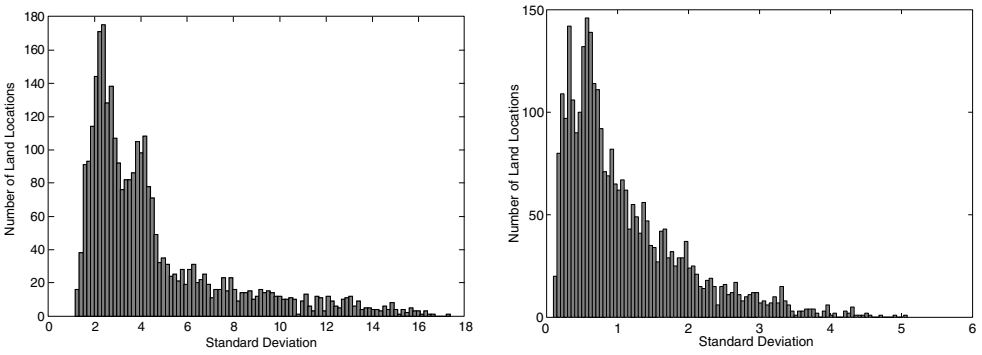
Transaction ID	Item	Store Location	Date	Price	...
⋮	⋮	⋮	⋮	⋮	
101123	Watch	Chicago	09/06/04	$25.99	...
101123	Battery	Chicago	09/06/04	$5.99	...
101124	Shoes	Minneapolis	09/06/04	$75.00	...
⋮	⋮	⋮	⋮	⋮	

process of eliminating attributes, such as the type of item, or reducing the number of values for a particular attribute; e.g., reducing the possible values for date from 365 days to 12 months. This type of aggregation is commonly used in Online Analytical Processing (OLAP), which is discussed further in Chapter 3.

There are several motivations for aggregation. First, the smaller data sets resulting from data reduction require less memory and processing time, and hence, aggregation may permit the use of more expensive data mining algorithms. Second, aggregation can act as a change of scope or scale by providing a high-level view of the data instead of a low-level view. In the previous example, aggregating over store locations and months gives us a monthly, per store view of the data instead of a daily, per item view. Finally, the behavior of groups of objects or attributes is often more stable than that of individual objects or attributes. This statement reflects the statistical fact that aggregate quantities, such as averages or totals, have less variability than the individual objects being aggregated. For totals, the actual amount of variation is larger than that of individual objects (on average), but the percentage of the variation is smaller, while for means, the actual amount of variation is less than that of individual objects (on average). A disadvantage of aggregation is the potential loss of interesting details. In the store example aggregating over months loses information about which day of the week has the highest sales.

Example 2.7 (Australian Precipitation). This example is based on precipitation in Australia from the period 1982 to 1993. Figure 2.8(a) shows a histogram for the standard deviation of average monthly precipitation for 3,030 0.5° by 0.5° grid cells in Australia, while Figure 2.8(b) shows a histogram for the standard deviation of the average yearly precipitation for the same locations. The average yearly precipitation has less variability than the average monthly precipitation. All precipitation measurements (and their standard deviations) are in centimeters.

■



(a) Histogram of standard deviation of average monthly precipitation

(b) Histogram of standard deviation of average yearly precipitation

Figure 2.8. Histograms of standard deviation for monthly and yearly precipitation in Australia for the period 1982 to 1993.

2.3.2 Sampling

Sampling is a commonly used approach for selecting a subset of the data objects to be analyzed. In statistics, it has long been used for both the preliminary investigation of the data and the final data analysis. Sampling can also be very useful in data mining. However, the motivations for sampling in statistics and data mining are often different. Statisticians use sampling because obtaining the entire set of data of interest is too expensive or time consuming, while data miners sample because it is too expensive or time consuming to process all the data. In some cases, using a sampling algorithm can reduce the data size to the point where a better, but more expensive algorithm can be used.

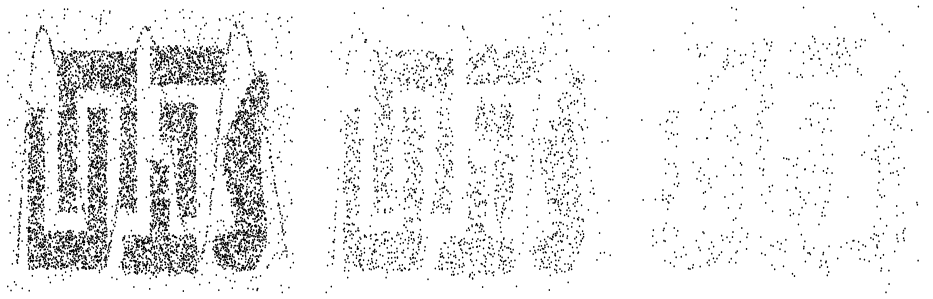
The key principle for effective sampling is the following: Using a sample will work almost as well as using the entire data set if the sample is representative. In turn, **a sample is representative** if it has approximately the same property (of interest) as the original set of data. If the mean (average) of the data objects is the property of interest, then a sample is representative if it has a mean that is close to that of the original data. Because sampling is a statistical process, the representativeness of any particular sample will vary, and the best that we can do is choose a sampling scheme that guarantees a high probability of getting a representative sample. As discussed next, this involves choosing the appropriate sample size and sampling techniques.

Sampling Approaches

There are many sampling techniques, but only a few of the most basic ones and their variations will be covered here. The simplest type of sampling is **simple random sampling**. For this type of sampling, there is an equal probability of selecting any particular item. There are two variations on random sampling (and other sampling techniques as well): (1) **sampling without replacement**—as each item is selected, it is removed from the set of all objects that together constitute the **population**, and (2) **sampling with replacement**—objects are not removed from the population as they are selected for the sample. In sampling with replacement, the same object can be picked more than once. The samples produced by the two methods are not much different when samples are relatively small compared to the data set size, but sampling with replacement is simpler to analyze since the probability of selecting any object remains constant during the sampling process.

When the population consists of different types of objects, with widely different numbers of objects, simple random sampling can fail to adequately represent those types of objects that are less frequent. This can cause problems when the analysis requires proper representation of all object types. For example, when building classification models for rare classes, it is critical that the rare classes be adequately represented in the sample. Hence, a sampling scheme that can accommodate differing frequencies for the items of interest is needed. **Stratified sampling**, which starts with prespecified groups of objects, is such an approach. In the simplest version, equal numbers of objects are drawn from each group even though the groups are of different sizes. In another variation, the number of objects drawn from each group is proportional to the size of that group.

Example 2.8 (Sampling and Loss of Information). Once a sampling technique has been selected, it is still necessary to choose the sample size. Larger sample sizes increase the probability that a sample will be representative, but they also eliminate much of the advantage of sampling. Conversely, with smaller sample sizes, patterns may be missed or erroneous patterns can be detected. Figure 2.9(a) shows a data set that contains 8000 two-dimensional points, while Figures 2.9(b) and 2.9(c) show samples from this data set of size 2000 and 500, respectively. Although most of the structure of this data set is present in the sample of 2000 points, much of the structure is missing in the sample of 500 points. ■



(a) 8000 points (b) 2000 points (c) 500 points

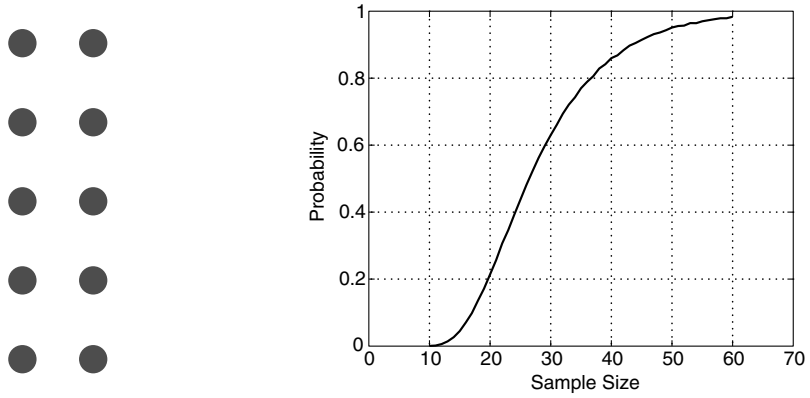
Figure 2.9. Example of the loss of structure with sampling.

Example 2.9 (Determining the Proper Sample Size). To illustrate that determining the proper sample size requires a methodical approach, consider the following task.

> Given a set of data that consists of a small number of almost equal-sized groups, find at least one representative point for each of the groups. Assume that the objects in each group are highly similar to each other, but not very similar to objects in different groups. Also assume that there are a relatively small number of groups, e.g., 10. Figure 2.10(a) shows an idealized set of clusters (groups) from which these points might be drawn.

This problem can be efficiently solved using sampling. One approach is to take a small sample of data points, compute the pairwise similarities between points, and then form groups of points that are highly similar. The desired set of representative points is then obtained by taking one point from each of these groups. To follow this approach, however, we need to determine a sample size that would guarantee, with a high probability, the desired outcome; that is, that at least one point will be obtained from each cluster. Figure 2.10(b) shows the probability of getting one object from each of the 10 groups as the sample size runs from 10 to 60. Interestingly, with a sample size of 20, there is little chance (20%) of getting a sample that includes all 10 clusters. Even with a sample size of 30, there is still a moderate chance (almost 40%) of getting a sample that doesn't contain objects from all 10 clusters. This issue is further explored in the context of clustering by Exercise 4 on page 559.

■



(a) Ten groups of points.

(b) Probability a sample contains points from each of 10 groups.

Figure 2.10. Finding representative points from 10 groups.

Progressive Sampling

The proper sample size can be difficult to determine, so **adaptive** or **progressive sampling** schemes are sometimes used. These approaches start with a small sample, and then increase the sample size until a sample of sufficient size has been obtained. While this technique eliminates the need to determine the correct sample size initially, it requires that there be a way to evaluate the sample to judge if it is large enough.

Suppose, for instance, that progressive sampling is used to learn a predictive model. Although the accuracy of predictive models increases as the sample size increases, at some point the increase in accuracy levels off. We want to stop increasing the sample size at this leveling-off point. By keeping track of the change in accuracy of the model as we take progressively larger samples, and by taking other samples close to the size of the current one, we can get an estimate as to how close we are to this leveling-off point, and thus, stop sampling.

2.3.3 Dimensionality Reduction

Data sets can have a large number of features. Consider a set of documents, where each document is represented by a vector whose components are the frequencies with which each word occurs in the document. In such cases,

there are typically thousands or tens of thousands of attributes (components), one for each word in the vocabulary. As another example, consider a set of time series consisting of the daily closing price of various stocks over a period of 30 years. In this case, the attributes, which are the prices on specific days, again number in the thousands.

There are a variety of benefits to dimensionality reduction. A key benefit is that many data mining algorithms work better if the dimensionality—the number of attributes in the data—is lower. This is partly because dimensionality reduction can eliminate irrelevant features and reduce noise and partly because of the curse of dimensionality, which is explained below. Another benefit is that a reduction of dimensionality can lead to a more understandable model because the model may involve fewer attributes. Also, dimensionality reduction may allow the data to be more easily visualized. Even if dimensionality reduction doesn't reduce the data to two or three dimensions, data is often visualized by looking at pairs or triplets of attributes, and the number of such combinations is greatly reduced. Finally, the amount of time and memory required by the data mining algorithm is reduced with a reduction in dimensionality.

The term dimensionality reduction is often reserved for those techniques that reduce the dimensionality of a data set by creating new attributes that are a combination of the old attributes. The reduction of dimensionality by selecting new attributes that are a subset of the old is known as feature subset selection or feature selection. It will be discussed in Section 2.3.4.

In the remainder of this section, we briefly introduce two important topics: the curse of dimensionality and dimensionality reduction techniques based on linear algebra approaches such as principal components analysis (PCA). More details on dimensionality reduction can be found in Appendix B.

The Curse of Dimensionality

The curse of dimensionality refers to the phenomenon that many types of data analysis become significantly harder as the dimensionality of the data increases. Specifically, as dimensionality increases, the data becomes increasingly sparse in the space that it occupies. For classification, this can mean that there are not enough data objects to allow the creation of a model that reliably assigns a class to all possible objects. For clustering, the definitions of density and the distance between points, which are critical for clustering, become less meaningful. (This is discussed further in Sections 9.1.2, 9.4.5, and 9.4.7.) As a result, many clustering and classification algorithms (and other

data analysis algorithms) have trouble with high-dimensional data—reduced classification accuracy and poor quality clusters.

Linear Algebra Techniques for Dimensionality Reduction

Some of the most common approaches for dimensionality reduction, particularly for continuous data, use techniques from linear algebra to project the data from a high-dimensional space into a lower-dimensional space. **Principal Components Analysis (PCA)** is a linear algebra technique for continuous attributes that finds new attributes (principal components) that (1) are linear combinations of the original attributes, (2) are **orthogonal** (perpendicular) to each other, and (3) capture the maximum amount of variation in the data. For example, the first two principal components capture as much of the variation in the data as is possible with two orthogonal attributes that are linear combinations of the original attributes. **Singular Value Decomposition (SVD)** is a linear algebra technique that is related to PCA and is also commonly used for dimensionality reduction. For additional details, see Appendices A and B.

2.3.4 Feature Subset Selection

Another way to reduce the dimensionality is to use only a subset of the features. While it might seem that such an approach would lose information, this is not the case if redundant and irrelevant features are present. **Redundant features** duplicate much or all of the information contained in one or more other attributes. For example, the purchase price of a product and the amount of sales tax paid contain much of the same information. **Irrelevant features** contain almost no useful information for the data mining task at hand. For instance, students' ID numbers are irrelevant to the task of predicting students' grade point averages. Redundant and irrelevant features can reduce classification accuracy and the quality of the clusters that are found.

While some irrelevant and redundant attributes can be eliminated immediately by using common sense or domain knowledge, selecting the best subset of features frequently requires a systematic approach. The ideal approach to feature selection is to try all possible subsets of features as input to the data mining algorithm of interest, and then take the subset that produces the best results. This method has the advantage of reflecting the objective and bias of the data mining algorithm that will eventually be used. Unfortunately, since the number of subsets involving n attributes is 2^n, such an approach is impractical in most situations and alternative strategies are needed. There are three standard approaches to feature selection: embedded, filter, and wrapper.

Embedded approaches Feature selection occurs naturally as part of the data mining algorithm. Specifically, during the operation of the data mining algorithm, the algorithm itself decides which attributes to use and which to ignore. Algorithms for building decision tree classifiers, which are discussed in Chapter 4, often operate in this manner.

Filter approaches Features are selected before the data mining algorithm is run, using some approach that is independent of the data mining task. For example, we might select sets of attributes whose pairwise correlation is as low as possible.

Wrapper approaches These methods use the target data mining algorithm as a black box to find the best subset of attributes, in a way similar to that of the ideal algorithm described above, but typically without enumerating all possible subsets.

Since the embedded approaches are algorithm-specific, only the filter and wrapper approaches will be discussed further here.

An Architecture for Feature Subset Selection

It is possible to encompass both the filter and wrapper approaches within a common architecture. The feature selection process is viewed as consisting of four parts: a measure for evaluating a subset, a search strategy that controls the generation of a new subset of features, a stopping criterion, and a validation procedure. Filter methods and wrapper methods differ only in the way in which they evaluate a subset of features. For a wrapper method, subset evaluation uses the target data mining algorithm, while for a filter approach, the evaluation technique is distinct from the target data mining algorithm. The following discussion provides some details of this approach, which is summarized in Figure 2.11.

Conceptually, feature subset selection is a search over all possible subsets of features. Many different types of search strategies can be used, but the search strategy should be computationally inexpensive and should find optimal or near optimal sets of features. It is usually not possible to satisfy both requirements, and thus, tradeoffs are necessary.

An integral part of the search is an evaluation step to judge how the current subset of features compares to others that have been considered. This requires an evaluation measure that attempts to determine the goodness of a subset of attributes with respect to a particular data mining task, such as classification

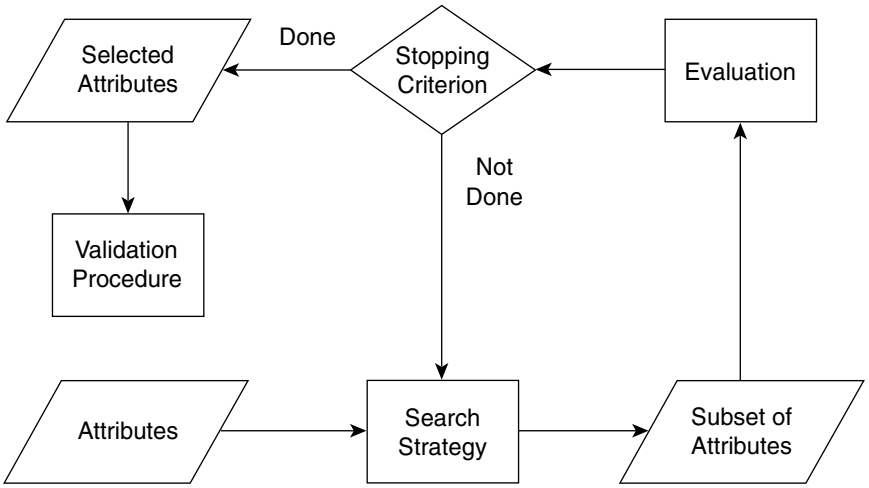


Figure 2.11. Flowchart of a feature subset selection process.

or clustering. For the filter approach, such measures attempt to predict how well the actual data mining algorithm will perform on a given set of attributes. For the wrapper approach, where evaluation consists of actually running the target data mining application, the subset evaluation function is simply the criterion normally used to measure the result of the data mining.

Because the number of subsets can be enormous and it is impractical to examine them all, some sort of stopping criterion is necessary. This strategy is usually based on one or more conditions involving the following: the number of iterations, whether the value of the subset evaluation measure is optimal or exceeds a certain threshold, whether a subset of a certain size has been obtained, whether simultaneous size and evaluation criteria have been achieved, and whether any improvement can be achieved by the options available to the search strategy.

Finally, once a subset of features has been selected, the results of the target data mining algorithm on the selected subset should be validated. A straightforward evaluation approach is to run the algorithm with the full set of features and compare the full results to results obtained using the subset of features. Hopefully, the subset of features will produce results that are better than or almost as good as those produced when using all features. Another validation approach is to use a number of different feature selection algorithms to obtain subsets of features and then compare the results of running the data mining algorithm on each subset.

Feature Weighting

Feature weighting is an alternative to keeping or eliminating features. More important features are assigned a higher weight, while less important features are given a lower weight. These weights are sometimes assigned based on domain knowledge about the relative importance of features. Alternatively, they may be determined automatically. For example, some classification schemes, such as support vector machines (Chapter 5), produce classification models in which each feature is given a weight. Features with larger weights play a more important role in the model. The normalization of objects that takes place when computing the cosine similarity (Section 2.4.5) can also be regarded as a type of feature weighting.

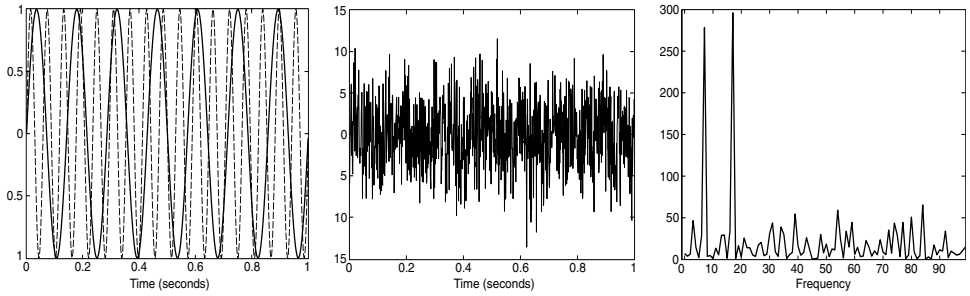
2.3.5 Feature Creation

It is frequently possible to create, from the original attributes, a new set of attributes that captures the important information in a data set much more effectively. Furthermore, the number of new attributes can be smaller than the number of original attributes, allowing us to reap all the previously described benefits of dimensionality reduction. Three related methodologies for creating new attributes are described next: feature extraction, mapping the data to a new space, and feature construction.

Feature Extraction

The creation of a new set of features from the original raw data is known as **feature extraction**. Consider a set of photographs, where each photograph is to be classified according to whether or not it contains a human face. The raw data is a set of pixels, and as such, is not suitable for many types of classification algorithms. However, if the data is processed to provide higher-level features, such as the presence or absence of certain types of edges and areas that are highly correlated with the presence of human faces, then a much broader set of classification techniques can be applied to this problem.

Unfortunately, in the sense in which it is most commonly used, feature extraction is highly domain-specific. For a particular field, such as image processing, various features and the techniques to extract them have been developed over a period of time, and often these techniques have limited applicability to other fields. Consequently, whenever data mining is applied to a relatively new area, a key task is the development of new features and feature extraction methods.



(a) Two time series. (b) Noisy time series. (c) Power spectrum

Figure 2.12. Application of the Fourier transform to identify the underlying frequencies in time series data.

Mapping the Data to a New Space

A totally different view of the data can reveal important and interesting features. Consider, for example, time series data, which often contains periodic patterns. If there is only a single periodic pattern and not much noise, then the pattern is easily detected. If, on the other hand, there are a number of periodic patterns and a significant amount of noise is present, then these patterns are hard to detect. Such patterns can, nonetheless, often be detected by applying a **Fourier transform** to the time series in order to change to a representation in which frequency information is explicit. In the example that follows, it will not be necessary to know the details of the Fourier transform. It is enough to know that, for each time series, the Fourier transform produces a new data object whose attributes are related to frequencies.

Example 2.10 (Fourier Analysis). The time series presented in Figure 2.12(b) is the sum of three other time series, two of which are shown in Figure 2.12(a) and have frequencies of 7 and 17 cycles per second, respectively. The third time series is random noise. Figure 2.12(c) shows the power spectrum that can be computed after applying a Fourier transform to the original time series. (Informally, the power spectrum is proportional to the square of each frequency attribute.) In spite of the noise, there are two peaks that correspond to the periods of the two original, non-noisy time series. Again, the main point is that better features can reveal important aspects of the data. ■

Many other sorts of transformations are also possible. Besides the Fourier transform, the **wavelet transform** has also proven very useful for time series and other types of data.

Feature Construction

Sometimes the features in the original data sets have the necessary information, but it is not in a form suitable for the data mining algorithm. In this situation, one or more new features constructed out of the original features can be more useful than the original features.

Example 2.11 (Density). To illustrate this, consider a data set consisting of information about historical artifacts, which, along with other information, contains the volume and mass of each artifact. For simplicity, assume that these artifacts are made of a small number of materials (wood, clay, bronze, gold) and that we want to classify the artifacts with respect to the material of which they are made. In this case, a density feature constructed from the mass and volume features, i.e., $density = mass/volume$, would most directly yield an accurate classification. Although there have been some attempts to automatically perform feature construction by exploring simple mathematical combinations of existing attributes, the most common approach is to construct features using domain expertise. ■

2.3.6 Discretization and Binarization

Some data mining algorithms, especially certain classification algorithms, require that the data be in the form of categorical attributes. Algorithms that find association patterns require that the data be in the form of binary attributes. Thus, it is often necessary to transform a continuous attribute into a categorical attribute (**discretization**), and both continuous and discrete attributes may need to be transformed into one or more binary attributes (**binarization**). Additionally, if a categorical attribute has a large number of values (categories), or some values occur infrequently, then it may be beneficial for certain data mining tasks to reduce the number of categories by combining some of the values.

As with feature selection, the best discretization and binarization approach is the one that "produces the best result for the data mining algorithm that will be used to analyze the data." It is typically not practical to apply such a criterion directly. Consequently, discretization or binarization is performed in

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

a way that satisfies a criterion that is thought to have a relationship to good performance for the data mining task being considered.

Binarization

A simple technique to binarize a categorical attribute is the following: If there are m categorical values, then uniquely assign each original value to an integer in the interval $[0, m-1]$. If the attribute is ordinal, then order must be maintained by the assignment. (Note that even if the attribute is originally represented using integers, this process is necessary if the integers are not in the interval $[0, m-1]$.) Next, convert each of these m integers to a binary number. Since $n = \lceil \log_2(m) \rceil$ binary digits are required to represent these integers, represent these binary numbers using n binary attributes. To illustrate, a categorical variable with 5 values {*awful*, *poor*, *OK*, *good*, *great*} would require three binary variables x_1, x_2, and x_3. The conversion is shown in Table 2.5.

Such a transformation can cause complications, such as creating unintended relationships among the transformed attributes. For example, in Table 2.5, attributes x_2 and x_3 are correlated because information about the *good* value is encoded using both attributes. Furthermore, association analysis requires asymmetric binary attributes, where only the presence of the attribute (value = 1) is important. For association problems, it is therefore necessary to introduce one binary attribute for each categorical value, as in Table 2.6. If the

number of resulting attributes is too large, then the techniques described below can be used to reduce the number of categorical values before binarization.

Likewise, for association problems, it may be necessary to replace a single binary attribute with two asymmetric binary attributes. Consider a binary attribute that records a person's gender, male or female. For traditional association rule algorithms, this information needs to be transformed into two asymmetric binary attributes, one that is a 1 only when the person is male and one that is a 1 only when the person is female. (For asymmetric binary attributes, the information representation is somewhat inefficient in that two bits of storage are required to represent each bit of information.)

Discretization of Continuous Attributes

Discretization is typically applied to attributes that are used in classification or association analysis. In general, the best discretization depends on the algorithm being used, as well as the other attributes being considered. Typically, however, the discretization of an attribute is considered in isolation.

Transformation of a continuous attribute to a categorical attribute involves two subtasks: deciding how many categories to have and determining how to map the values of the continuous attribute to these categories. In the first step, after the values of the continuous attribute are sorted, they are then divided into n intervals by specifying $n-1$ **split points**. In the second, rather trivial step, all the values in one interval are mapped to the same categorical value. Therefore, the problem of discretization is one of deciding how many split points to choose and where to place them. The result can be represented either as a set of intervals $\{(x_0, x_1], (x_1, x_2], \ldots, (x_{n-1}, x_n)\}$, where x_0 and x_n may be $+\infty$ or $-\infty$, respectively, or equivalently, as a series of inequalities $x_0 < x \leq x_1, \ldots, x_{n-1} < x < x_n$.

Unsupervised Discretization A basic distinction between discretization methods for classification is whether class information is used (supervised) or not (unsupervised). If class information is not used, then relatively simple approaches are common. For instance, the **equal width** approach divides the range of the attribute into a user-specified number of intervals each having the same width. Such an approach can be badly affected by outliers, and for that reason, an **equal frequency (equal depth)** approach, which tries to put the same number of objects into each interval, is often preferred. As another example of unsupervised discretization, a clustering method, such as K-means (see Chapter 8), can also be used. Finally, visually inspecting the data can sometimes be an effective approach.

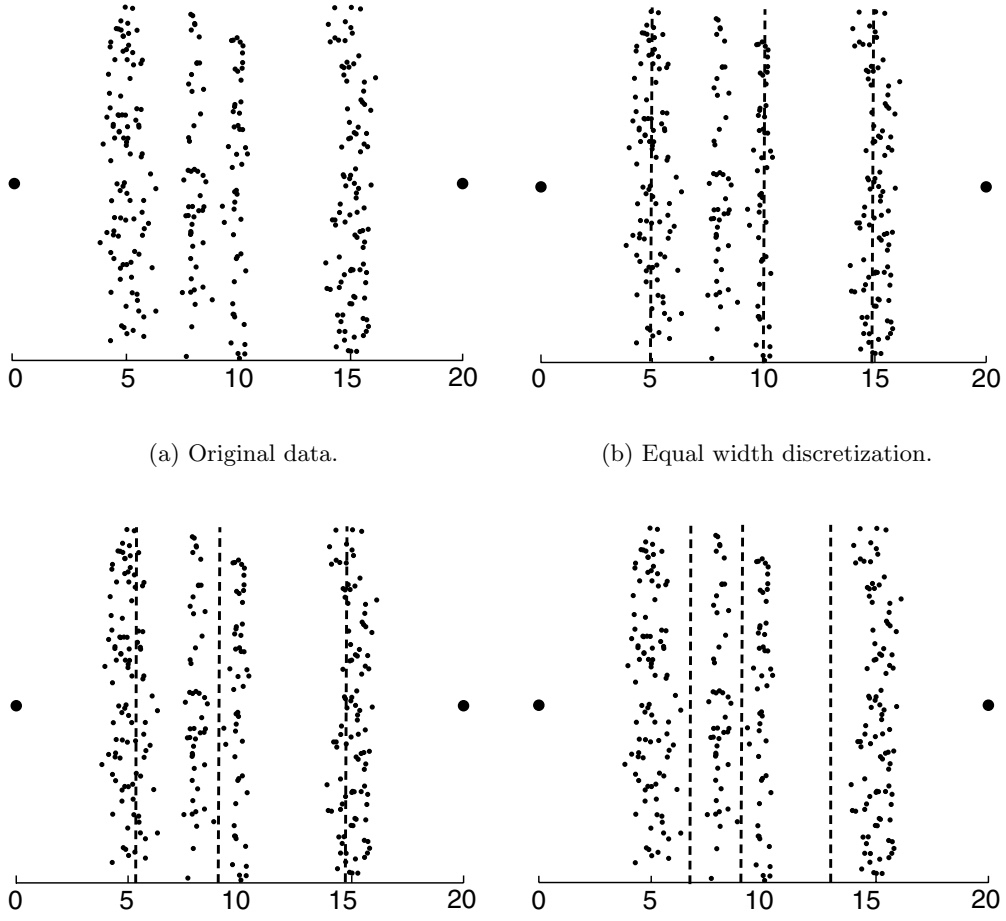
Example 2.12 (Discretization Techniques). This example demonstrates how these approaches work on an actual data set. Figure 2.13(a) shows data points belonging to four different groups, along with two outliers—the large dots on either end. The techniques of the previous paragraph were applied to discretize the x values of these data points into four categorical values. (Points in the data set have a random y component to make it easy to see how many points are in each group.) Visually inspecting the data works quite well, but is not automatic, and thus, we focus on the other three approaches. The split points produced by the techniques equal width, equal frequency, and K-means are shown in Figures 2.13(b), 2.13(c), and 2.13(d), respectively. The split points are represented as dashed lines. If we measure the performance of a discretization technique by the extent to which different objects in different groups are assigned the same categorical value, then K-means performs best, followed by equal frequency, and finally, equal width. ■

Supervised Discretization The discretization methods described above are usually better than no discretization, but keeping the end purpose in mind and using additional information (class labels) often produces better results. This should not be surprising, since an interval constructed with no knowledge of class labels often contains a mixture of class labels. A conceptually simple approach is to place the splits in a way that maximizes the purity of the intervals. In practice, however, such an approach requires potentially arbitrary decisions about the purity of an interval and the minimum size of an interval. To overcome such concerns, some statistically based approaches start with each attribute value as a separate interval and create larger intervals by merging adjacent intervals that are similar according to a statistical test. Entropy-based approaches are one of the most promising approaches to discretization, and a simple approach based on entropy will be presented.

First, it is necessary to define **entropy**. Let k be the number of different class labels, m_i be the number of values in the i^{th} interval of a partition, and m_{ij} be the number of values of class j in interval i. Then the entropy e_i of the i^{th} interval is given by the equation

$$e_i = \sum_{i=1}^{k} p_{ij} \log_2 p_{ij},$$

where $p_{ij} = m_{ij}/m_i$ is the probability (fraction of values) of class j in the i^{th} interval. The total entropy, e, of the partition is the weighted average of the individual interval entropies, i.e.,



(a) Original data.

(b) Equal width discretization.

(c) Equal frequency discretization.

(d) K-means discretization.

Figure 2.13. Different discretization techniques.

$$e = \sum_{i=1}^{n} w_i e_i,$$

where m is the number of values, $w_i = m_i/m$ is the fraction of values in the i^{th} interval, and n is the number of intervals. Intuitively, the entropy of an interval is a measure of the purity of an interval. If an interval contains only values of one class (is perfectly pure), then the entropy is 0 and it contributes

nothing to the overall entropy. If the classes of values in an interval occur equally often (the interval is as impure as possible), then the entropy is a maximum.

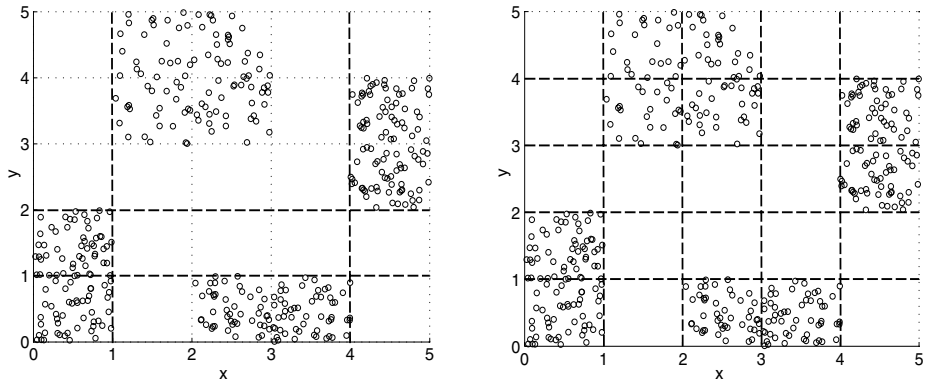
A simple approach for partitioning a continuous attribute starts by bisecting the initial values so that the resulting two intervals give minimum entropy. This technique only needs to consider each value as a possible split point, because it is assumed that intervals contain ordered sets of values. The splitting process is then repeated with another interval, typically choosing the interval with the worst (highest) entropy, until a user-specified number of intervals is reached, or a stopping criterion is satisfied.

Example 2.13 (Discretization of Two Attributes). This method was used to independently discretize both the x and y attributes of the two-dimensional data shown in Figure 2.14. In the first discretization, shown in Figure 2.14(a), the x and y attributes were both split into three intervals. (The dashed lines indicate the split points.) In the second discretization, shown in Figure 2.14(b), the x and y attributes were both split into five intervals. ■

This simple example illustrates two aspects of discretization. First, in two dimensions, the classes of points are well separated, but in one dimension, this is not so. In general, discretizing each attribute separately often guarantees suboptimal results. Second, five intervals work better than three, but six intervals do not improve the discretization much, at least in terms of entropy. (Entropy values and results for six intervals are not shown.) Consequently, it is desirable to have a stopping criterion that automatically finds the right number of partitions.

Categorical Attributes with Too Many Values

Categorical attributes can sometimes have too many values. If the categorical attribute is an ordinal attribute, then techniques similar to those for continuous attributes can be used to reduce the number of categories. If the categorical attribute is nominal, however, then other approaches are needed. Consider a university that has a large number of departments. Consequently, a *department name* attribute might have dozens of different values. In this situation, we could use our knowledge of the relationships among different departments to combine departments into larger groups, such as *engineering*, *social sciences*, or *biological sciences*. If domain knowledge does not serve as a useful guide or such an approach results in poor classification performance, then it is necessary to use a more empirical approach, such as grouping values



(a) Three intervals (b) Five intervals

Figure 2.14. Discretizing x and y attributes for four groups (classes) of points.

together only if such a grouping results in improved classification accuracy or achieves some other data mining objective.

2.3.7 Variable Transformation

A **variable transformation** refers to a transformation that is applied to all the values of a variable. (We use the term variable instead of attribute to adhere to common usage, although we will also refer to attribute transformation on occasion.) In other words, for each object, the transformation is applied to the value of the variable for that object. For example, if only the magnitude of a variable is important, then the values of the variable can be transformed by taking the absolute value. In the following section, we discuss two important types of variable transformations: simple functional transformations and normalization.

Simple Functions

For this type of variable transformation, a simple mathematical function is applied to each value individually. If x is a variable, then examples of such transformations include x^k, $\log x$, e^x, $\sqrt{x}$, $1/x$, $\sin x$, or $|x|$. In statistics, variable transformations, especially $sqrt$, log, and $1/x$, are often used to transform data that does not have a Gaussian (normal) distribution into data that does. While this can be important, other reasons often take precedence in data min-

ing. Suppose the variable of interest is the number of data bytes in a session, and the number of bytes ranges from 1 to 1 billion. This is a huge range, and it may be advantageous to compress it by using a $\log_{10}$ transformation. In this case, sessions that transferred 10^8 and 10^9 bytes would be more similar to each other than sessions that transferred 10 and 1000 bytes ($9 - 8 = 1$ versus $3 - 1 = 2$). For some applications, such as network intrusion detection, this may be what is desired, since the first two sessions most likely represent transfers of large files, while the latter two sessions could be two quite distinct types of sessions.

Variable transformations should be applied with caution since they change the nature of the data. While this is what is desired, there can be problems if the nature of the transformation is not fully appreciated. For instance, the transformation $1/x$ reduces the magnitude of values that are 1 or larger, but increases the magnitude of values between 0 and 1. To illustrate, the values $\{1, 2, 3\}$ go to $\{1, \frac{1}{2}, \frac{1}{3}\}$, but the values $\{1, \frac{1}{2}, \frac{1}{3}\}$ go to $\{1, 2, 3\}$. Thus, for all sets of values, the transformation $1/x$ reverses the order. To help clarify the effect of a transformation, it is important to ask questions such as the following: Does the order need to be maintained? Does the transformation apply to all values, especially negative values and 0? What is the effect of the transformation on the values between 0 and 1? Exercise 17 on page 92 explores other aspects of variable transformation.

Normalization or Standardization

Another common type of variable transformation is the **standardization** or **normalization** of a variable. (In the data mining community the terms are often used interchangeably. In statistics, however, the term normalization can be confused with the transformations used for making a variable **normal**, i.e., **Gaussian**.) The goal of standardization or normalization is to make an entire set of values have a particular property. A traditional example is that of "standardizing a variable" in statistics. If $\overline{x}$ is the mean (average) of the attribute values and s_x is their standard deviation, then the transformation $x' = (x - \overline{x})/s_x$ creates a new variable that has a mean of 0 and a standard deviation of 1. If different variables are to be combined in some way, then such a transformation is often necessary to avoid having a variable with large values dominate the results of the calculation. To illustrate, consider comparing people based on two variables: age and income. For any two people, the difference in income will likely be much higher in absolute terms (hundreds or thousands of dollars) than the difference in age (less than 150). If the differences in the range of values of age and income are not taken into account, then

the comparison between people will be dominated by differences in income. In particular, if the similarity or dissimilarity of two people is calculated using the similarity or dissimilarity measures defined later in this chapter, then in many cases, such as that of Euclidean distance, the income values will dominate the calculation.

The mean and standard deviation are strongly affected by outliers, so the above transformation is often modified. First, the mean is replaced by the **median**, i.e., the middle value. Second, the standard deviation is replaced by the **absolute standard deviation**. Specifically, if x is a variable, then the absolute standard deviation of x is given by $\sigma_A = \sum_{i=1}^{m} |x_i - \mu|$, where x_i is the i^{th} value of the variable, m is the number of objects, and μ is either the mean or median. Other approaches for computing estimates of the location (center) and spread of a set of values in the presence of outliers are described in Sections 3.2.3 and 3.2.4, respectively. These measures can also be used to define a standardization transformation.

2.4 Measures of Similarity and Dissimilarity

Similarity and dissimilarity are important because they are used by a number of data mining techniques, such as clustering, nearest neighbor classification, and anomaly detection. In many cases, the initial data set is not needed once these similarities or dissimilarities have been computed. Such approaches can be viewed as transforming the data to a similarity (dissimilarity) space and then performing the analysis.

We begin with a discussion of the basics: high-level definitions of similarity and dissimilarity, and a discussion of how they are related. For convenience, the term **proximity** is used to refer to either similarity or dissimilarity. Since the proximity between two objects is a function of the proximity between the corresponding attributes of the two objects, we first describe how to measure the proximity between objects having only one simple attribute, and then consider proximity measures for objects with multiple attributes. This includes measures such as correlation and Euclidean distance, which are useful for dense data such as time series or two-dimensional points, as well as the Jaccard and cosine similarity measures, which are useful for sparse data like documents. Next, we consider several important issues concerning proximity measures. The section concludes with a brief discussion of how to select the right proximity measure.

2.4.1 Basics

Definitions

Informally, the **similarity** between two objects is a numerical measure of the degree to which the two objects are alike. Consequently, similarities are *higher* for pairs of objects that are more alike. Similarities are usually non-negative and are often between 0 (no similarity) and 1 (complete similarity).

The **dissimilarity** between two objects is a numerical measure of the degree to which the two objects are different. Dissimilarities are *lower* for more similar pairs of objects. Frequently, the term **distance** is used as a synonym for dissimilarity, although, as we shall see, distance is often used to refer to a special class of dissimilarities. Dissimilarities sometimes fall in the interval $[0, 1]$, but it is also common for them to range from 0 to ∞.

Transformations

Transformations are often applied to convert a similarity to a dissimilarity, or vice versa, or to transform a proximity measure to fall within a particular range, such as [0,1]. For instance, we may have similarities that range from 1 to 10, but the particular algorithm or software package that we want to use may be designed to only work with dissimilarities, or it may only work with similarities in the interval [0,1]. We discuss these issues here because we will employ such transformations later in our discussion of proximity. In addition, these issues are relatively independent of the details of specific proximity measures.

Frequently, proximity measures, especially similarities, are defined or transformed to have values in the interval [0,1]. Informally, the motivation for this is to use a scale in which a proximity value indicates the fraction of similarity (or dissimilarity) between two objects. Such a transformation is often relatively straightforward. For example, if the similarities between objects range from 1 (not at all similar) to 10 (completely similar), we can make them fall within the range $[0, 1]$ by using the transformation $s' = (s - 1)/9$, where s and s' are the original and new similarity values, respectively. In the more general case, the transformation of similarities to the interval $[0, 1]$ is given by the expression $s' = (s - min\_s)/(max\_s - min\_s)$, where max_s and min_s are the maximum and minimum similarity values, respectively. Likewise, dissimilarity measures with a finite range can be mapped to the interval [0,1] by using the formula $d' = (d - min\_d)/(max\_d - min\_d)$.

There can be various complications in mapping proximity measures to the interval $[0, 1]$, however. If, for example, the proximity measure originally takes

values in the interval [0,∞], then a non-linear transformation is needed and values will not have the same relationship to one another on the new scale. Consider the transformation $d' = d/(1 + d)$ for a dissimilarity measure that ranges from 0 to ∞. The dissimilarities 0, 0.5, 2, 10, 100, and 1000 will be transformed into the new dissimilarities 0, 0.33, 0.67, 0.90, 0.99, and 0.999, respectively. Larger values on the original dissimilarity scale are compressed into the range of values near 1, but whether or not this is desirable depends on the application. Another complication is that the meaning of the proximity measure may be changed. For example, correlation, which is discussed later, is a measure of similarity that takes values in the interval [-1,1]. Mapping these values to the interval [0,1] by taking the absolute value loses information about the sign, which can be important in some applications. See Exercise 22 on page 94.

Transforming similarities to dissimilarities and vice versa is also relatively straightforward, although we again face the issues of preserving meaning and changing a linear scale into a non-linear scale. If the similarity (or dissimilarity) falls in the interval [0,1], then the dissimilarity can be defined as $d = 1 - s$ ($s = 1 - d$). Another simple approach is to define similarity as the negative of the dissimilarity (or vice versa). To illustrate, the dissimilarities 0, 1, 10, and 100 can be transformed into the similarities 0, -1, -10, and -100, respectively.

The similarities resulting from the negation transformation are not restricted to the range $[0, 1]$, but if that is desired, then transformations such as $s = \frac{1}{d+1}$, $s = e^{-d}$, or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$ can be used. For the transformation $s = \frac{1}{d+1}$, the dissimilarities 0, 1, 10, 100 are transformed into 1, 0.5, 0.09, 0.01, respectively. For $s = e^{-d}$, they become 1.00, 0.37, 0.00, 0.00, respectively, while for $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$ they become 1.00, 0.99, 0.00, 0.00, respectively. In this discussion, we have focused on converting dissimilarities to similarities. Conversion in the opposite direction is considered in Exercise 23 on page 94.

In general, any monotonic decreasing function can be used to convert dissimilarities to similarities, or vice versa. Of course, other factors also must be considered when transforming similarities to dissimilarities, or vice versa, or when transforming the values of a proximity measure to a new scale. We have mentioned issues related to preserving meaning, distortion of scale, and requirements of data analysis tools, but this list is certainly not exhaustive.

2.4.2 Similarity and Dissimilarity between Simple Attributes

The proximity of objects with a number of attributes is typically defined by combining the proximities of individual attributes, and thus, we first discuss

proximity between objects having a single attribute. Consider objects described by one nominal attribute. What would it mean for two such objects to be similar? Since nominal attributes only convey information about the distinctness of objects, all we can say is that two objects either have the same value or they do not. Hence, in this case similarity is traditionally defined as 1 if attribute values match, and as 0 otherwise. A dissimilarity would be defined in the opposite way: 0 if the attribute values match, and 1 if they do not.

For objects with a single ordinal attribute, the situation is more complicated because information about order should be taken into account. Consider an attribute that measures the quality of a product, e.g., a candy bar, on the scale {*poor*, *fair*, *OK*, *good*, *wonderful*}. It would seem reasonable that a product, P1, which is rated *wonderful*, would be closer to a product P2, which is rated *good*, than it would be to a product P3, which is rated *OK*. To make this observation quantitative, the values of the ordinal attribute are often mapped to successive integers, beginning at 0 or 1, e.g., {*poor*=0, *fair*=1, *OK*=2, good=3, *wonderful*=4}. Then, $d(\text{P1}, \text{P2}) = 3 - 2 = 1$ or, if we want the dissimilarity to fall between 0 and 1, $d(\text{P1}, \text{P2}) = \frac{3-2}{4} = 0.25$. A similarity for ordinal attributes can then be defined as $s = 1 - d$.

This definition of similarity (dissimilarity) for an ordinal attribute should make the reader a bit uneasy since this assumes equal intervals, and this is not so. Otherwise, we would have an interval or ratio attribute. Is the difference between the values *fair* and *good* really the same as that between the values *OK* and *wonderful*? Probably not, but in practice, our options are limited, and in the absence of more information, this is the standard approach for defining proximity between ordinal attributes.

For interval or ratio attributes, the natural measure of dissimilarity between two objects is the absolute difference of their values. For example, we might compare our current weight and our weight a year ago by saying "I am ten pounds heavier." In cases such as these, the dissimilarities typically range from 0 to ∞, rather than from 0 to 1. The similarity of interval or ratio attributes is typically expressed by transforming a similarity into a dissimilarity, as previously described.

Table 2.7 summarizes this discussion. In this table, x and y are two objects that have one attribute of the indicated type. Also, $d(x, y)$ and $s(x, y)$ are the dissimilarity and similarity between x and y, respectively. Other approaches are possible; these are the most common ones.

The following two sections consider more complicated measures of proximity between objects that involve multiple attributes: (1) dissimilarities between data objects and (2) similarities between data objects. This division

Table 2.7. Similarity and dissimilarity for simple attributes

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

allows us to more naturally display the underlying motivations for employing various proximity measures. We emphasize, however, that similarities can be transformed into dissimilarities and vice versa using the approaches described earlier.

2.4.3 Dissimilarities between Data Objects

In this section, we discuss various kinds of dissimilarities. We begin with a discussion of distances, which are dissimilarities with certain properties, and then provide examples of more general kinds of dissimilarities.

Distances

We first present some examples, and then offer a more formal description of distances in terms of the properties common to all distances. The **Euclidean distance**, d, between two points, $\mathbf{x}$ and $\mathbf{y}$, in one-, two-, three-, or higher-dimensional space, is given by the following familiar formula:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}, \tag{2.1}$$

where n is the number of dimensions and x_k and y_k are, respectively, the k^{th} attributes (components) of x and y. We illustrate this formula with Figure 2.15 and Tables 2.8 and 2.9, which show a set of points, the x and y coordinates of these points, and the **distance matrix** containing the pairwise distances of these points.

The Euclidean distance measure given in Equation 2.1 is generalized by the **Minkowski** distance metric shown in Equation 2.2,

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}, \tag{2.2}$$

where r is a parameter. The following are the three most common examples of Minkowski distances.

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance. A common example is the **Hamming distance**, which is the number of bits that are different between two objects that have only binary attributes, i.e., between two binary vectors.
- $r = 2$. Euclidean distance (L_2 norm).
- $r = \infty$. Supremum (L_{max} or L_∞ norm) distance. This is the maximum difference between any attribute of the objects. More formally, the L_∞ distance is defined by Equation 2.3

$$d(\mathbf{x}, \mathbf{y}) = \lim_{r \to \infty} \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}. \tag{2.3}$$

The r parameter should not be confused with the number of dimensions (attributes) n. The Euclidean, Manhattan, and supremum distances are defined for all values of n: $1, 2, 3, \ldots$, and specify different ways of combining the differences in each dimension (attribute) into an overall distance.

Tables 2.10 and 2.11, respectively, give the proximity matrices for the L_1 and L_∞ distances using data from Table 2.8. Notice that all these distance matrices are symmetric; i.e., the ij^{th} entry is the same as the ji^{th} entry. In Table 2.9, for instance, the fourth row of the first column and the fourth column of the first row both contain the value 5.1.

Distances, such as the Euclidean distance, have some well-known properties. If $d(\mathbf{x}, \mathbf{y})$ is the distance between two points, $\mathbf{x}$ and $\mathbf{y}$, then the following properties hold.

1. **Positivity**
   (a) $d(\mathbf{x}, \mathbf{x}) \geq 0$ for all $\mathbf{x}$ and $\mathbf{y}$,
   (b) $d(\mathbf{x}, \mathbf{y}) = 0$ only if $\mathbf{x} = \mathbf{y}$.

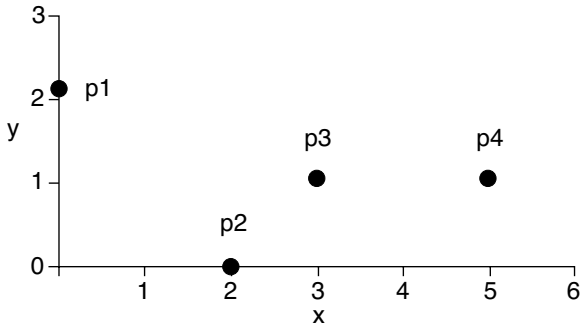


Figure 2.15. Four two-dimensional points.

Table 2.8. x and y coordinates of four points.

point	x coordinate	y coordinate
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Table 2.9. Euclidean distance matrix for Table 2.8.

	p1	p2	p3	p4
p1	0.0	2.8	3.2	5.1
p2	2.8	0.0	1.4	3.2
p3	3.2	1.4	0.0	2.0
p4	5.1	3.2	2.0	0.0

Table 2.10. L_1 distance matrix for Table 2.8.

L_1	p1	p2	p3	p4
p1	0.0	4.0	4.0	6.0
p2	4.0	0.0	2.0	4.0
p3	4.0	2.0	0.0	2.0
p4	6.0	4.0	2.0	0.0

Table 2.11. L_∞ distance matrix for Table 2.8.

L_∞	p1	p2	p3	p4
p1	0.0	2.0	3.0	5.0
p2	2.0	0.0	1.0	3.0
p3	3.0	1.0	0.0	2.0
p4	5.0	3.0	2.0	0.0

2. **Symmetry**
   $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$.

3. **Triangle Inequality**
   $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ for all points $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$.

Measures that satisfy all three properties are known as **metrics**. Some people only use the term distance for dissimilarity measures that satisfy these properties, but that practice is often violated. The three properties described here are useful, as well as mathematically pleasing. Also, if the triangle inequality holds, then this property can be used to increase the efficiency of techniques (including clustering) that depend on distances possessing this property. (See Exercise 25.) Nonetheless, many dissimilarities do not satisfy one or more of the metric properties. We give two examples of such measures.

Example 2.14 (Non-metric Dissimilarities: Set Differences). This example is based on the notion of the difference of two sets, as defined in set theory. Given two sets A and B, $A - B$ is the set of elements of A that are not in B. For example, if $A = \{1, 2, 3, 4\}$ and $B = \{2, 3, 4\}$, then $A - B = \{1\}$ and $B - A = \emptyset$, the empty set. We can define the distance d between two sets A and B as $d(A, B) = size(A - B)$, where $size$ is a function returning the number of elements in a set. This distance measure, which is an integer value greater than or equal to 0, does not satisfy the second part of the positivity property, the symmetry property, or the triangle inequality. However, these properties can be made to hold if the dissimilarity measure is modified as follows: $d(A, B) = size(A - B) + size(B - A)$. See Exercise 21 on page 94. ■

Example 2.15 (Non-metric Dissimilarities: Time). This example gives a more everyday example of a dissimilarity measure that is not a metric, but that is still useful. Define a measure of the distance between times of the day as follows:

$$d(t_1, t_2) = \left\{ \begin{array}{ll} t_2 - t_1 & \text{if } t_1 \leq t_2 \\ 24 + (t_2 - t_1) & \text{if } t_1 \geq t_2 \end{array} \right\}. \tag{2.4}$$

To illustrate, d(1PM, 2PM) = 1 hour, while d(2PM, 1PM) = 23 hours. Such a definition would make sense, for example, when answering the question: "If an event occurs at 1PM every day, and it is now 2PM, how long do I have to wait for that event to occur again?" ■

2.4.4 Similarities between Data Objects

For similarities, the triangle inequality (or the analogous property) typically does not hold, but symmetry and positivity typically do. To be explicit, if $s(\mathbf{x}, \mathbf{y})$ is the similarity between points $\mathbf{x}$ and $\mathbf{y}$, then the typical properties of similarities are the following:

1. $s(\mathbf{x}, \mathbf{y}) = 1$ only if $\mathbf{x} = \mathbf{y}$. $(0 \leq s \leq 1)$
2. $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$. (Symmetry)

There is no general analog of the triangle inequality for similarity measures. It is sometimes possible, however, to show that a similarity measure can easily be converted to a metric distance. The cosine and Jaccard similarity measures, which are discussed shortly, are two examples. Also, for specific similarity measures, it is possible to derive mathematical bounds on the similarity between two objects that are similar in spirit to the triangle inequality.

Example 2.16 (A Non-symmetric Similarity Measure). Consider an experiment in which people are asked to classify a small set of characters as they flash on a screen. The **confusion matrix** for this experiment records how often each character is classified as itself, and how often each is classified as another character. For instance, suppose that "0" appeared 200 times and was classified as a "0" 160 times, but as an "o" 40 times. Likewise, suppose that 'o' appeared 200 times and was classified as an "o" 170 times, but as "0" only 30 times. If we take these counts as a measure of the similarity between two characters, then we have a similarity measure, but one that is not symmetric. In such situations, the similarity measure is often made symmetric by setting $s'(\mathbf{x}, \mathbf{y}) = s'(\mathbf{y}, \mathbf{x}) = (s(\mathbf{x}, \mathbf{y}) + s(\mathbf{y}, \mathbf{x}))/2$, where s' indicates the new similarity measure. ■

2.4.5 Examples of Proximity Measures

This section provides specific examples of some similarity and dissimilarity measures.

Similarity Measures for Binary Data

Similarity measures between objects that contain only binary attributes are called **similarity coefficients**, and typically have values between 0 and 1. A value of 1 indicates that the two objects are completely similar, while a value of 0 indicates that the objects are not at all similar. There are many rationales for why one coefficient is better than another in specific instances.

Let $\mathbf{x}$ and $\mathbf{y}$ be two objects that consist of n binary attributes. The comparison of two such objects, i.e., two binary vectors, leads to the following four quantities (frequencies):

f_{00} = the number of attributes where $\mathbf{x}$ is 0 and $\mathbf{y}$ is 0
f_{01} = the number of attributes where $\mathbf{x}$ is 0 and $\mathbf{y}$ is 1
f_{10} = the number of attributes where $\mathbf{x}$ is 1 and $\mathbf{y}$ is 0
f_{11} = the number of attributes where $\mathbf{x}$ is 1 and $\mathbf{y}$ is 1

Simple Matching Coefficient One commonly used similarity coefficient is the **simple matching coefficient** (SMC), which is defined as

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}. \qquad (2.5)$$

This measure counts both presences and absences equally. Consequently, the SMC could be used to find students who had answered questions similarly on a test that consisted only of true/false questions.

Jaccard Coefficient Suppose that $\mathbf{x}$ and $\mathbf{y}$ are data objects that represent two rows (two transactions) of a transaction matrix (see Section 2.1.2). If each asymmetric binary attribute corresponds to an item in a store, then a 1 indicates that the item was purchased, while a 0 indicates that the product was not purchased. Since the number of products not purchased by any customer far outnumbers the number of products that were purchased, a similarity measure such as SMC would say that all transactions are very similar. As a result, the Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes. The **Jaccard coefficient**, which is often symbolized by J, is given by the following equation:

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}. \qquad (2.6)$$

Example 2.17 (The SMC and Jaccard Similarity Coefficients). To illustrate the difference between these two similarity measures, we calculate SMC and J for the following two binary vectors.

$\mathbf{x} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
$\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$

$f_{01} = 2$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 1
$f_{10} = 1$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 0
$f_{00} = 7$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 0
$f_{11} = 0$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 1

$SMC = \frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}} = \frac{0+7}{2+1+0+7} = 0.7$

$J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}} = \frac{0}{2+1+0} = 0$ ■

Cosine Similarity

Documents are often represented as vectors, where each attribute represents the frequency with which a particular term (word) occurs in the document. It is more complicated than this, of course, since certain common words are ig-

nored and various processing techniques are used to account for different forms of the same word, differing document lengths, and different word frequencies.

Even though documents have thousands or tens of thousands of attributes (terms), each document is sparse since it has relatively few non-zero attributes. (The normalizations used for documents do not create a non-zero entry where there was a zero entry; i.e., they preserve sparsity.) Thus, as with transaction data, similarity should not depend on the number of shared 0 values since any two documents are likely to "not contain" many of the same words, and therefore, if 0–0 matches are counted, most documents will be highly similar to most other documents. Therefore, a similarity measure for documents needs to ignores 0–0 matches like the Jaccard measure, but also must be able to handle non-binary vectors. The **cosine similarity**, defined next, is one of the most common measure of document similarity. If $\mathbf{x}$ and $\mathbf{y}$ are two document vectors, then

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|}, \tag{2.7}$$

where $\cdot$ indicates the vector dot product, $\mathbf{x} \cdot \mathbf{y} = \sum_{k=1}^{n} x_k y_k$, and $\|\mathbf{x}\|$ is the length of vector $\mathbf{x}$, $\|\mathbf{x}\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$.

Example 2.18 (Cosine Similarity of Two Document Vectors). This example calculates the cosine similarity for the following two data objects, which might represent document vectors:

$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$
$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$

$\mathbf{x} \cdot \mathbf{y} = 3 * 1 + 2 * 0 + 0 * 0 + 5 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 1 + 0 * 0 + 0 * 2 = 5$
$\|\mathbf{x}\| = \sqrt{3 * 3 + 2 * 2 + 0 * 0 + 5 * 5 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 2 + 0 * 0 + 0 * 0} = 6.48$
$\|\mathbf{y}\| = \sqrt{1 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 2 * 2} = 2.24$
$\mathbf{cos}(\mathbf{x}, \mathbf{y}) = \mathbf{0.31}$

■

As indicated by Figure 2.16, cosine similarity really is a measure of the (cosine of the) angle between $\mathbf{x}$ and $\mathbf{y}$. Thus, if the cosine similarity is 1, the angle between $\mathbf{x}$ and $\mathbf{y}$ is 0°, and $\mathbf{x}$ and $\mathbf{y}$ are the same except for magnitude (length). If the cosine similarity is 0, then the angle between $\mathbf{x}$ and $\mathbf{y}$ is 90°, and they do not share any terms (words).

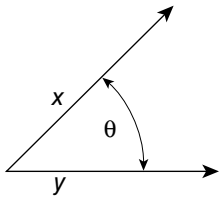


Figure 2.16. Geometric illustration of the cosine measure.

Equation 2.7 can be written as Equation 2.8.

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}}{\|\mathbf{x}\|} \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|} = \mathbf{x}' \cdot \mathbf{y}', \tag{2.8}$$

where $\mathbf{x}' = \mathbf{x}/\|\mathbf{x}\|$ and $\mathbf{y}' = \mathbf{y}/\|\mathbf{y}\|$. Dividing $\mathbf{x}$ and $\mathbf{y}$ by their lengths normalizes them to have a length of 1. This means that cosine similarity does not take the *magnitude* of the two data objects into account when computing similarity. (Euclidean distance might be a better choice when magnitude is important.) For vectors with a length of 1, the cosine measure can be calculated by taking a simple dot product. Consequently, when many cosine similarities between objects are being computed, normalizing the objects to have unit length can reduce the time required.

Extended Jaccard Coefficient (Tanimoto Coefficient)

The extended Jaccard coefficient can be used for document data and that reduces to the Jaccard coefficient in the case of binary attributes. The extended Jaccard coefficient is also known as the Tanimoto coefficient. (However, there is another coefficient that is also known as the Tanimoto coefficient.) This coefficient, which we shall represent as EJ, is defined by the following equation:

$$EJ(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x} \cdot \mathbf{y}}. \tag{2.9}$$

Correlation

The correlation between two data objects that have binary or continuous variables is a measure of the linear relationship between the attributes of the objects. (The calculation of correlation between attributes, which is more common, can be defined similarly.) More precisely, **Pearson's correlation**

coefficient between two data objects, $\mathbf{x}$ and $\mathbf{y}$, is defined by the following equation:

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) * \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x \; s_y}, \quad (2.10)$$

where we are using the following standard statistical notation and definitions:

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \overline{x})(y_k - \overline{y}) \quad (2.11)$$

$$\begin{aligned} \text{standard\_deviation}(\mathbf{x}) &= s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \overline{x})^2} \\ \text{standard\_deviation}(\mathbf{y}) &= s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (y_k - \overline{y})^2} \end{aligned}$$

$$\begin{aligned} \overline{x} &= \frac{1}{n} \sum_{k=1}^{n} x_k \text{ is the mean of } \mathbf{x} \\ \overline{y} &= \frac{1}{n} \sum_{k=1}^{n} y_k \text{ is the mean of } \mathbf{y} \end{aligned}$$

Example 2.19 (Perfect Correlation). Correlation is always in the range -1 to 1. A correlation of 1 (-1) means that $\mathbf{x}$ and $\mathbf{y}$ have a perfect positive (negative) linear relationship; that is, $x_k = ay_k + b$, where a and b are constants. The following two sets of values for $\mathbf{x}$ and $\mathbf{y}$ indicate cases where the correlation is -1 and $+1$, respectively. In the first case, the means of $\mathbf{x}$ and $\mathbf{y}$ were chosen to be 0, for simplicity.

$\mathbf{x} = (-3, \;\; 6, \; 0, \;\; 3, -6)$
$\mathbf{y} = (\;\; 1, -2, \; 0, -1, \;\; 2)$

$\mathbf{x} = (3, 6, 0, 3, 6)$
$\mathbf{y} = (1, 2, 0, 1, 2)$

■

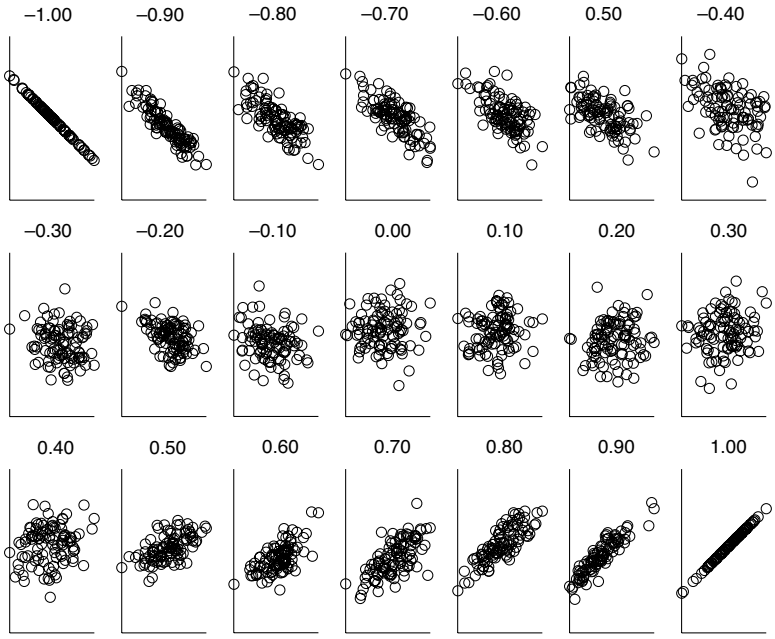


Figure 2.17. Scatter plots illustrating correlations from -1 to 1.

Example 2.20 (Non-linear Relationships). If the correlation is 0, then there is no linear relationship between the attributes of the two data objects. However, non-linear relationships may still exist. In the following example, $x_k = y_k^2$, but their correlation is 0.

$\mathbf{x} = (-3, -2, -1,\ 0,\ 1,\ 2,\ 3)$
$\mathbf{y} = (\ 9,\ \ 4,\ \ \ 1,\ 0,\ 1,\ 4,\ 9)$

■

Example 2.21 (Visualizing Correlation). It is also easy to judge the correlation between two data objects $\mathbf{x}$ and $\mathbf{y}$ by plotting pairs of corresponding attribute values. Figure 2.17 shows a number of these plots when $\mathbf{x}$ and $\mathbf{y}$ have 30 attributes and the values of these attributes are randomly generated (with a normal distribution) so that the correlation of $\mathbf{x}$ and $\mathbf{y}$ ranges from -1 to 1. Each circle in a plot represents one of the 30 attributes; its x coordinate is the value of one of the attributes for $\mathbf{x}$, while its y coordinate is the value of the same attribute for $\mathbf{y}$. ■

If we transform $\mathbf{x}$ and $\mathbf{y}$ by subtracting off their means and then normalizing them so that their lengths are 1, then their correlation can be calculated by

taking the dot product. Notice that this is not the same as the standardization used in other contexts, where we make the transformations, $x'_k = (x_k - \overline{x})/s_x$ and $y'_k = (y_k - \overline{y})/s_y$.

Bregman Divergence* This section provides a brief description of Bregman divergences, which are a family of proximity functions that share some common properties. As a result, it is possible to construct general data mining algorithms, such as clustering algorithms, that work with any Bregman divergence. A concrete example is the K-means clustering algorithm (Section 8.2). Note that this section requires knowledge of vector calculus.

Bregman divergences are loss or distortion functions. To understand the idea of a loss function, consider the following. Let $\mathbf{x}$ and $\mathbf{y}$ be two points, where $\mathbf{y}$ is regarded as the original point and $\mathbf{x}$ is some distortion or approximation of it. For example, $\mathbf{x}$ may be a point that was generated, for example, by adding random noise to $\mathbf{y}$. The goal is to measure the resulting distortion or loss that results if $\mathbf{y}$ is approximated by $\mathbf{x}$. Of course, the more similar $\mathbf{x}$ and $\mathbf{y}$ are, the smaller the loss or distortion. Thus, Bregman divergences can be used as dissimilarity functions.

More formally, we have the following definition.

Definition 2.6 (Bregman Divergence). Given a strictly convex function ϕ (with a few modest restrictions that are generally satisfied), the Bregman divergence (loss function) $D(\mathbf{x}, \mathbf{y})$ generated by that function is given by the following equation:

$$D(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \nabla\phi(\mathbf{y}), (\mathbf{x} - \mathbf{y}) \rangle \tag{2.12}$$

where $\nabla\phi(\mathbf{y})$ is the gradient of ϕ evaluated at $\mathbf{y}$, $\mathbf{x} - \mathbf{y}$, is the vector difference between $\mathbf{x}$ and $\mathbf{y}$, and $\langle \nabla\phi(\mathbf{y}), (\mathbf{x} - \mathbf{y}) \rangle$ is the inner product between $\nabla\phi(\mathbf{x})$ and $(\mathbf{x} - \mathbf{y})$. For points in Euclidean space, the inner product is just the dot product.

$D(\mathbf{x}, \mathbf{y})$ can be written as $D(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - L(\mathbf{x})$, where $L(\mathbf{x}) = \phi(\mathbf{y}) + \langle \nabla\phi(\mathbf{y}), (\mathbf{x} - \mathbf{y}) \rangle$ and represents the equation of a plane that is tangent to the function ϕ at $\mathbf{y}$. Using calculus terminology, $L(\mathbf{x})$ is the linearization of ϕ around the point $\mathbf{y}$ and the Bregman divergence is just the difference between a function and a linear approximation to that function. Different Bregman divergences are obtained by using different choices for ϕ.

Example 2.22. We provide a concrete example using squared Euclidean distance, but restrict ourselves to one dimension to simplify the mathematics. Let

x and y be real numbers and $\phi(t)$ be the real valued function, $\phi(t) = t^2$. In that case, the gradient reduces to the derivative and the dot product reduces to multiplication. Specifically, Equation 2.12 becomes Equation 2.13.

$$D(x, y) = x^2 - y^2 - 2y(x - y) = (x - y)^2 \tag{2.13}$$

The graph for this example, with $y = 1$, is shown in Figure 2.18. The Bregman divergence is shown for two values of x: $x = 2$ and $x = 3$. ■

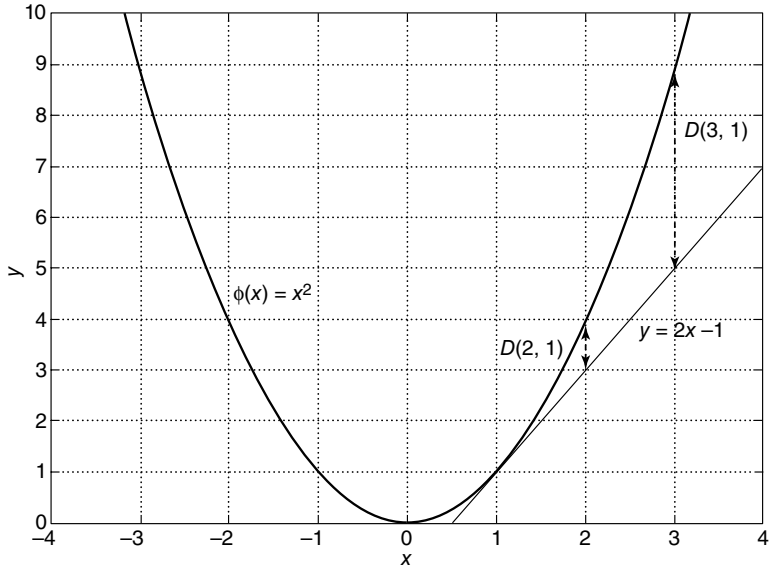


Figure 2.18. Illustration of Bregman divergence.

2.4.6 Issues in Proximity Calculation

This section discusses several important issues related to proximity measures: (1) how to handle the case in which attributes have different scales and/or are correlated, (2) how to calculate proximity between objects that are composed of different types of attributes, e.g., quantitative and qualitative, (3) and how to handle proximity calculation when attributes have different weights; i.e., when not all attributes contribute equally to the proximity of objects.

Standardization and Correlation for Distance Measures

An important issue with distance measures is how to handle the situation when attributes do not have the same range of values. (This situation is often described by saying that "the variables have different scales.") Earlier, Euclidean distance was used to measure the distance between people based on two attributes: age and income. Unless these two attributes are standardized, the distance between two people will be dominated by income.

A related issue is how to compute distance when there is correlation between some of the attributes, perhaps in addition to differences in the ranges of values. A generalization of Euclidean distance, the **Mahalanobis distance**, is useful when attributes are correlated, have different ranges of values (different variances), and the distribution of the data is approximately Gaussian (normal). Specifically, the Mahalanobis distance between two objects (vectors) $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\text{mahalanobis}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{y})^T, \tag{2.14}$$

where $\boldsymbol{\Sigma}^{-1}$ is the inverse of the covariance matrix of the data. Note that the covariance matrix Σ is the matrix whose ij^{th} entry is the covariance of the i^{th} and j^{th} attributes as defined by Equation 2.11.

Example 2.23. In Figure 2.19, there are 1000 points, whose x and y attributes have a correlation of 0.6. The distance between the two large points at the opposite ends of the long axis of the ellipse is 14.7 in terms of Euclidean distance, but only 6 with respect to Mahalanobis distance. In practice, computing the Mahalanobis distance is expensive, but can be worthwhile for data whose attributes are correlated. If the attributes are relatively uncorrelated, but have different ranges, then standardizing the variables is sufficient.

■

Combining Similarities for Heterogeneous Attributes

The previous definitions of similarity were based on approaches that assumed all the attributes were of the same type. A general approach is needed when the attributes are of different types. One straightforward approach is to compute the similarity between each attribute separately using Table 2.7, and then combine these similarities using a method that results in a similarity between 0 and 1. Typically, the overall similarity is defined as the average of all the individual attribute similarities.

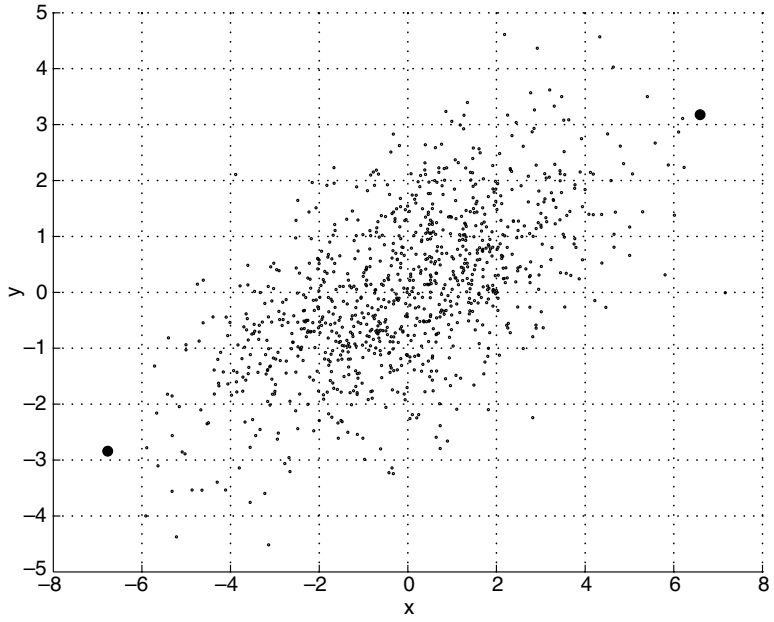


Figure 2.19. Set of two-dimensional points. The Mahalanobis distance between the two points represented by large dots is 6; their Euclidean distance is 14.7.

Unfortunately, this approach does not work well if some of the attributes are asymmetric attributes. For example, if all the attributes are asymmetric binary attributes, then the similarity measure suggested previously reduces to the simple matching coefficient, a measure that is not appropriate for asymmetric binary attributes. The easiest way to fix this problem is to omit asymmetric attributes from the similarity calculation when their values are 0 for both of the objects whose similarity is being computed. A similar approach also works well for handling missing values.

In summary, Algorithm 2.1 is effective for computing an overall similarity between two objects, $\mathbf{x}$ and $\mathbf{y}$, with different types of attributes. This procedure can be easily modified to work with dissimilarities.

Using Weights

In much of the previous discussion, all attributes were treated equally when computing proximity. This is not desirable when some attributes are more important to the definition of proximity than others. To address these situations,

Algorithm 2.1 Similarities of heterogeneous objects.

1: For the k^{th} attribute, compute a similarity, $s_k(\mathbf{x}, \mathbf{y})$, in the range $[0, 1]$.
2: Define an indicator variable, δ_k, for the k^{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is an asymmetric attribute and} \\ & \text{both objects have a value of 0, or if one of the objects} \\ & \text{has a missing value for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3: Compute the overall similarity between the two objects using the following formula:

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^{n} \delta_k} \tag{2.15}$$

the formulas for proximity can be modified by weighting the contribution of each attribute.

If the weights w_k sum to 1, then (2.15) becomes

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^{n} \delta_k}. \tag{2.16}$$

The definition of the Minkowski distance can also be modified as follows:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} w_k |x_k - y_k|^r \right)^{1/r}. \tag{2.17}$$

2.4.7 Selecting the Right Proximity Measure

The following are a few general observations that may be helpful. First, the type of proximity measure should fit the type of data. For many types of dense, continuous data, metric distance measures such as Euclidean distance are often used. Proximity between continuous attributes is most often expressed in terms of differences, and distance measures provide a well-defined way of combining these differences into an overall proximity measure. Although attributes can have different scales and be of differing importance, these issues can often be dealt with as described earlier.

For sparse data, which often consists of asymmetric attributes, we typically employ similarity measures that ignore 0–0 matches. Conceptually, this reflects the fact that, for a pair of complex objects, similarity depends on the number of characteristics they both share, rather than the number of characteristics they both lack. More specifically, for sparse, asymmetric data, most

objects have only a few of the characteristics described by the attributes, and thus, are highly similar in terms of the characteristics they do not have. The cosine, Jaccard, and extended Jaccard measures are appropriate for such data.

There are other characteristics of data vectors that may need to be considered. Suppose, for example, that we are interested in comparing time series. If the magnitude of the time series is important (for example, each time series represent total sales of the same organization for a different year), then we could use Euclidean distance. If the time series represent different quantities (for example, blood pressure and oxygen consumption), then we usually want to determine if the time series have the same shape, not the same magnitude. Correlation, which uses a built-in normalization that accounts for differences in magnitude and level, would be more appropriate.

In some cases, transformation or normalization of the data is important for obtaining a proper similarity measure since such transformations are not always present in proximity measures. For instance, time series may have trends or periodic patterns that significantly impact similarity. Also, a proper computation of similarity may require that time lags be taken into account. Finally, two time series may only be similar over specific periods of time. For example, there is a strong relationship between temperature and the use of natural gas, but only during the heating season.

Practical consideration can also be important. Sometimes, a one or more proximity measures are already in use in a particular field, and thus, others will have answered the question of which proximity measures should be used. Other times, the software package or clustering algorithm being used may drastically limit the choices. If efficiency is a concern, then we may want to choose a proximity measure that has a property, such as the triangle inequality, that can be used to reduce the number of proximity calculations. (See Exercise 25.)

However, if common practice or practical restrictions do not dictate a choice, then the proper choice of a proximity measure can be a time-consuming task that requires careful consideration of both domain knowledge and the purpose for which the measure is being used. A number of different similarity measures may need to be evaluated to see which ones produce results that make the most sense.

2.5 Bibliographic Notes

It is essential to understand the nature of the data that is being analyzed, and at a fundamental level, this is the subject of measurement theory. In

particular, one of the initial motivations for defining types of attributes was to be precise about which statistical operations were valid for what sorts of data. We have presented the view of measurement theory that was initially described in a classic paper by S. S. Stevens [79]. (Tables 2.2 and 2.3 are derived from those presented by Stevens [80].) While this is the most common view and is reasonably easy to understand and apply, there is, of course, much more to measurement theory. An authoritative discussion can be found in a three-volume series on the foundations of measurement theory [63, 69, 81]. Also of interest is a wide-ranging article by Hand [55], which discusses measurement theory and statistics, and is accompanied by comments from other researchers in the field. Finally, there are many books and articles that describe measurement issues for particular areas of science and engineering.

Data quality is a broad subject that spans every discipline that uses data. Discussions of precision, bias, accuracy, and significant figures can be found in many introductory science, engineering, and statistics textbooks. The view of data quality as "fitness for use" is explained in more detail in the book by Redman [76]. Those interested in data quality may also be interested in MIT's Total Data Quality Management program [70, 84]. However, the knowledge needed to deal with specific data quality issues in a particular domain is often best obtained by investigating the data quality practices of researchers in that field.

Aggregation is a less well-defined subject than many other preprocessing tasks. However, aggregation is one of the main techniques used by the database area of Online Analytical Processing (OLAP), which is discussed in Chapter 3. There has also been relevant work in the area of symbolic data analysis (Bock and Diday [47]). One of the goals in this area is to summarize traditional record data in terms of symbolic data objects whose attributes are more complex than traditional attributes. Specifically, these attributes can have values that are sets of values (categories), intervals, or sets of values with weights (histograms). Another goal of symbolic data analysis is to be able to perform clustering, classification, and other kinds of data analysis on data that consists of symbolic data objects.

Sampling is a subject that has been well studied in statistics and related fields. Many introductory statistics books, such as the one by Lindgren [65], have some discussion on sampling, and there are entire books devoted to the subject, such as the classic text by Cochran [49]. A survey of sampling for data mining is provided by Gu and Liu [54], while a survey of sampling for databases is provided by Olken and Rotem [72]. There are a number of other data mining and database-related sampling references that may be of interest,

including papers by Palmer and Faloutsos [74], Provost et al. [75], Toivonen [82], and Zaki et al. [85].

In statistics, the traditional techniques that have been used for dimensionality reduction are multidimensional scaling (MDS) (Borg and Groenen [48], Kruskal and Uslaner [64]) and principal component analysis (PCA) (Jolliffe [58]), which is similar to singular value decomposition (SVD) (Demmel [50]). Dimensionality reduction is discussed in more detail in Appendix B.

Discretization is a topic that has been extensively investigated in data mining. Some classification algorithms only work with categorical data, and association analysis requires binary data, and thus, there is a significant motivation to investigate how to best binarize or discretize continuous attributes. For association analysis, we refer the reader to work by Srikant and Agrawal [78], while some useful references for discretization in the area of classification include work by Dougherty et al. [51], Elomaa and Rousu [52], Fayyad and Irani [53], and Hussain et al. [56].

Feature selection is another topic well investigated in data mining. A broad coverage of this topic is provided in a survey by Molina et al. [71] and two books by Liu and Motada [66, 67]. Other useful papers include those by Blum and Langley [46], Kohavi and John [62], and Liu et al. [68].

It is difficult to provide references for the subject of feature transformations because practices vary from one discipline to another. Many statistics books have a discussion of transformations, but typically the discussion is restricted to a particular purpose, such as ensuring the normality of a variable or making sure that variables have equal variance. We offer two references: Osborne [73] and Tukey [83].

While we have covered some of the most commonly used distance and similarity measures, there are hundreds of such measures and more are being created all the time. As with so many other topics in this chapter, many of these measures are specific to particular fields; e.g., in the area of time series see papers by Kalpakis et al. [59] and Keogh and Pazzani [61]. Clustering books provide the best general discussions. In particular, see the books by Anderberg [45], Jain and Dubes [57], Kaufman and Rousseeuw [60], and Sneath and Sokal [77].

Bibliography

[45] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, December 1973.

[46] A. Blum and P. Langley. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, 97(1–2):245–271, 1997.

[47] H. H. Bock and E. Diday. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data (Studies in Classification, Data Analysis, and Knowledge Organization)*. Springer-Verlag Telos, January 2000.

[48] I. Borg and P. Groenen. *Modern Multidimensional Scaling—Theory and Applications*. Springer-Verlag, February 1997.

[49] W. G. Cochran. *Sampling Techniques*. John Wiley & Sons, 3rd edition, July 1977.

[50] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial & Applied Mathematics, September 1997.

[51] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and Unsupervised Discretization of Continuous Features. In *Proc. of the 12th Intl. Conf. on Machine Learning*, pages 194–202, 1995.

[52] T. Elomaa and J. Rousu. General and Efficient Multisplitting of Numerical Attributes. *Machine Learning*, 36(3):201–244, 1999.

[53] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuousvalued attributes for classification learning. In *Proc. 13th Int. Joint Conf. on Artificial Intelligence*, pages 1022–1027. Morgan Kaufman, 1993.

[54] F. H. Gaohua Gu and H. Liu. Sampling and Its Application in Data Mining: A Survey. Technical Report TRA6/00, National University of Singapore, Singapore, 2000.

[55] D. J. Hand. Statistics and the Theory of Measurement. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 159(3):445–492, 1996.

[56] F. Hussain, H. Liu, C. L. Tan, and M. Dash. TRC6/99: Discretization: an enabling technique. Technical report, National University of Singapore, Singapore, 1999.

[57] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall Advanced Reference Series. Prentice Hall, March 1988. Book available online at http://www.cse.msu.edu/∼jain/Clustering_Jain_Dubes.pdf.

[58] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 2nd edition, October 2002.

[59] K. Kalpakis, D. Gada, and V. Puttagunta. Distance Measures for Effective Clustering of ARIMA Time-Series. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 273–280. IEEE Computer Society, 2001.

[60] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics. John Wiley and Sons, New York, November 1990.

[61] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD*, pages 285–289, 2000.

[62] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1–2):273–324, 1997.

[63] D. Krantz, R. D. Luce, P. Suppes, and A. Tversky. *Foundations of Measurements: Volume 1: Additive and polynomial representations*. Academic Press, New York, 1971.

[64] J. B. Kruskal and E. M. Uslaner. *Multidimensional Scaling*. Sage Publications, August 1978.

[65] B. W. Lindgren. *Statistical Theory*. CRC Press, January 1993.

[66] H. Liu and H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer International Series in Engineering and Computer Science, 453. Kluwer Academic Publishers, July 1998.

[67] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer International Series in Engineering and Computer Science, 454. Kluwer Academic Publishers, July 1998.

[68] H. Liu, H. Motoda, and L. Yu. Feature Extraction, Selection, and Construction. In N. Ye, editor, *The Handbook of Data Mining*, pages 22–41. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2003.

[69] R. D. Luce, D. Krantz, P. Suppes, and A. Tversky. *Foundations of Measurements: Volume 3: Representation, Axiomatization, and Invariance*. Academic Press, New York, 1990.

[70] MIT Total Data Quality Management Program. web.mit.edu/tdqm/www/index.shtml, 2003.

[71] L. C. Molina, L. Belanche, and A. Nebot. Feature Selection Algorithms: A Survey and Experimental Evaluation. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, 2002.

[72] F. Olken and D. Rotem. Random Sampling from Databases—A Survey. *Statistics & Computing*, 5(1):25–42, March 1995.

[73] J. Osborne. Notes on the Use of Data Transformations. *Practical Assessment, Research & Evaluation*, 28(6), 2002.

[74] C. R. Palmer and C. Faloutsos. Density biased sampling: An improved method for data mining and clustering. *ACM SIGMOD Record*, 29(2):82–92, 2000.

[75] F. J. Provost, D. Jensen, and T. Oates. Efficient Progressive Sampling. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 23–32, 1999.

[76] T. C. Redman. *Data Quality: The Field Guide*. Digital Press, January 2001.

[77] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, San Francisco, 1971.

[78] R. Srikant and R. Agrawal. Mining Quantitative Association Rules in Large Relational Tables. In *Proc. of 1996 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 1–12, Montreal, Quebec, Canada, August 1996.

[79] S. S. Stevens. On the Theory of Scales of Measurement. *Science*, 103(2684):677–680, June 1946.

[80] S. S. Stevens. Measurement. In G. M. Maranell, editor, *Scaling: A Sourcebook for Behavioral Scientists*, pages 22–41. Aldine Publishing Co., Chicago, 1974.

[81] P. Suppes, D. Krantz, R. D. Luce, and A. Tversky. *Foundations of Measurements: Volume 2: Geometrical, Threshold, and Probabilistic Representations*. Academic Press, New York, 1989.

[82] H. Toivonen. Sampling Large Databases for Association Rules. In *VLDB96*, pages 134–145. Morgan Kaufman, September 1996.

[83] J. W. Tukey. On the Comparative Anatomy of Transformations. *Annals of Mathematical Statistics*, 28(3):602–632, September 1957.

[84] R. Y. Wang, M. Ziad, Y. W. Lee, and Y. R. Wang. *Data Quality*. The Kluwer International Series on Advances in Database Systems, Volume 23. Kluwer Academic Publishers, January 2001.

[85] M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of Sampling for Data Mining of Association Rules. Technical Report TR617, Rensselaer Polytechnic Institute, 1996.

2.6 Exercises

1. In the initial example of Chapter 2, the statistician says, "Yes, fields 2 and 3 are basically the same." Can you tell from the three lines of sample data that are shown why she says that?

2. Classify the following attributes as binary, discrete, or continuous. Also classify them as qualitative (nominal or ordinal) or quantitative (interval or ratio). Some cases may have more than one interpretation, so briefly indicate your reasoning if you think there may be some ambiguity.

   Example: Age in years. **Answer:** Discrete, quantitative, ratio

   (a) Time in terms of AM or PM.

   (b) Brightness as measured by a light meter.

   (c) Brightness as measured by people's judgments.

   (d) Angles as measured in degrees between 0 and 360.

   (e) Bronze, Silver, and Gold medals as awarded at the Olympics.

   (f) Height above sea level.

   (g) Number of patients in a hospital.

   (h) ISBN numbers for books. (Look up the format on the Web.)

   (i) Ability to pass light in terms of the following values: opaque, translucent, transparent.

   (j) Military rank.

   (k) Distance from the center of campus.

   (l) Density of a substance in grams per cubic centimeter.

   (m) Coat check number. (When you attend an event, you can often give your coat to someone who, in turn, gives you a number that you can use to claim your coat when you leave.)

3. You are approached by the marketing director of a local company, who believes that he has devised a foolproof way to measure customer satisfaction. He explains his scheme as follows: "It's so simple that I can't believe that no one has thought of it before. I just keep track of the number of customer complaints for each product. I read in a data mining book that counts are ratio attributes, and so, my measure of product satisfaction must be a ratio attribute. But when I rated the products based on my new customer satisfaction measure and showed them to my boss, he told me that I had overlooked the obvious, and that my measure was worthless. I think that he was just mad because our best-selling product had the worst satisfaction since it had the most complaints. Could you help me set him straight?"

   (a) Who is right, the marketing director or his boss? If you answered, his boss, what would you do to fix the measure of satisfaction?

   (b) What can you say about the attribute type of the original product satisfaction attribute?

4. A few months later, you are again approached by the same marketing director as in Exercise 3. This time, he has devised a better approach to measure the extent to which a customer prefers one product over other, similar products. He explains, "When we develop new products, we typically create several variations and evaluate which one customers prefer. Our standard procedure is to give our test subjects all of the product variations at one time and then ask them to rank the product variations in order of preference. However, our test subjects are very indecisive, especially when there are more than two products. As a result, testing takes forever. I suggested that we perform the comparisons in pairs and then use these comparisons to get the rankings. Thus, if we have three product variations, we have the customers compare variations 1 and 2, then 2 and 3, and finally 3 and 1. Our testing time with my new procedure is a third of what it was for the old procedure, but the employees conducting the tests complain that they cannot come up with a consistent ranking from the results. And my boss wants the latest product evaluations, yesterday. I should also mention that he was the person who came up with the old product evaluation approach. Can you help me?"

   (a) Is the marketing director in trouble? Will his approach work for generating an ordinal ranking of the product variations in terms of customer preference? Explain.

   (b) Is there a way to fix the marketing director's approach? More generally, what can you say about trying to create an ordinal measurement scale based on pairwise comparisons?

   (c) For the original product evaluation scheme, the overall rankings of each product variation are found by computing its average over all test subjects. Comment on whether you think that this is a reasonable approach. What other approaches might you take?

5. Can you think of a situation in which identification numbers would be useful for prediction?

6. An educational psychologist wants to use association analysis to analyze test results. The test consists of 100 questions with four possible answers each.

   (a) How would you convert this data into a form suitable for association analysis?

   (b) In particular, what type of attributes would you have and how many of them are there?

7. Which of the following quantities is likely to show more temporal autocorrelation: daily rainfall or daily temperature? Why?

8. Discuss why a document-term matrix is an example of a data set that has asymmetric discrete or asymmetric continuous features.

9. Many sciences rely on observation instead of (or in addition to) designed experiments. Compare the data quality issues involved in observational science with those of experimental science and data mining.

10. Discuss the difference between the precision of a measurement and the terms single and double precision, as they are used in computer science, typically to represent floating-point numbers that require 32 and 64 bits, respectively.

11. Give at least two advantages to working with data stored in text files instead of in a binary format.

12. Distinguish between noise and outliers. Be sure to consider the following questions.

    (a) Is noise ever interesting or desirable? Outliers?

    (b) Can noise objects be outliers?

    (c) Are noise objects always outliers?

    (d) Are outliers always noise objects?

    (e) Can noise make a typical value into an unusual one, or vice versa?

13. Consider the problem of finding the K nearest neighbors of a data object. A programmer designs Algorithm 2.2 for this task.

Algorithm 2.2 Algorithm for finding K nearest neighbors.

1: **for** $i = 1$ to *number of data objects* **do**
2: Find the distances of the i^{th} object to all other objects.
3: Sort these distances in decreasing order.
(Keep track of which object is associated with each distance.)
4: **return** the objects associated with the first K distances of the sorted list
5: **end for**

    (a) Describe the potential problems with this algorithm if there are duplicate objects in the data set. Assume the distance function will only return a distance of 0 for objects that are the same.

    (b) How would you fix this problem?

14. The following attributes are measured for members of a herd of Asian elephants: *weight, height, tusk length, trunk length,* and *ear area*. Based on these measurements, what sort of similarity measure from Section 2.4 would you use to compare or group these elephants? Justify your answer and explain any special circumstances.

15. You are given a set of m objects that is divided into K groups, where the i^{th} group is of size m_i. If the goal is to obtain a sample of size $n < m$, what is the difference between the following two sampling schemes? (Assume sampling with replacement.)

    (a) We randomly select $n * m_i/m$ elements from each group.

    (b) We randomly select n elements from the data set, without regard for the group to which an object belongs.

16. Consider a document-term matrix, where tf_{ij} is the frequency of the i^{th} word (term) in the j^{th} document and m is the number of documents. Consider the variable transformation that is defined by

$$tf'_{ij} = tf_{ij} * \log \frac{m}{df_i}, \tag{2.18}$$

    where df_i is the number of documents in which the i^{th} term appears, which is known as the **document frequency** of the term. This transformation is known as the **inverse document frequency** transformation.

    (a) What is the effect of this transformation if a term occurs in one document? In every document?

    (b) What might be the purpose of this transformation?

17. Assume that we apply a square root transformation to a ratio attribute x to obtain the new attribute x^*. As part of your analysis, you identify an interval (a, b) in which x^* has a linear relationship to another attribute y.

    (a) What is the corresponding interval (a, b) in terms of x?

    (b) Give an equation that relates y to x.

18. This exercise compares and contrasts some similarity and distance measures.

    (a) For binary data, the L1 distance corresponds to the Hamming distance; that is, the number of bits that are different between two binary vectors. The Jaccard similarity is a measure of the similarity between two binary vectors. Compute the Hamming distance and the Jaccard similarity between the following two binary vectors.

    $\mathbf{x} = 0101010001$
    $\mathbf{y} = 0100011000$

    (b) Which approach, Jaccard or Hamming distance, is more similar to the Simple Matching Coefficient, and which approach is more similar to the cosine measure? Explain. (Note: The Hamming measure is a distance, while the other three measures are similarities, but don't let this confuse you.)

(c) Suppose that you are comparing how similar two organisms of different species are in terms of the number of genes they share. Describe which measure, Hamming or Jaccard, you think would be more appropriate for comparing the genetic makeup of two organisms. Explain. (Assume that each animal is represented as a binary vector, where each attribute is 1 if a particular gene is present in the organism and 0 otherwise.)

(d) If you wanted to compare the genetic makeup of two organisms of the same species, e.g., two human beings, would you use the Hamming distance, the Jaccard coefficient, or a different measure of similarity or distance? Explain. (Note that two human beings share $> 99.9\%$ of the same genes.)

19. For the following vectors, $\mathbf{x}$ and $\mathbf{y}$, calculate the indicated similarity or distance measures.

(a) $\mathbf{x} = (1, 1, 1, 1),\ \mathbf{y} = (2, 2, 2, 2)$ cosine, correlation, Euclidean

(b) $\mathbf{x} = (0, 1, 0, 1),\ \mathbf{y} = (1, 0, 1, 0)$ cosine, correlation, Euclidean, Jaccard

(c) $\mathbf{x} = (0, -1, 0, 1),\ \mathbf{y} = (1, 0, -1, 0)$ cosine, correlation, Euclidean

(d) $\mathbf{x} = (1, 1, 0, 1, 0, 1),\ \mathbf{y} = (1, 1, 1, 0, 0, 1)$ cosine, correlation, Jaccard

(e) $\mathbf{x} = (2, -1, 0, 2, 0, -3),\ \mathbf{y} = (-1, 1, -1, 0, 0, -1)$ cosine, correlation

20. Here, we further explore the cosine and correlation measures.

(a) What is the range of values that are possible for the cosine measure?

(b) If two objects have a cosine measure of 1, are they identical? Explain.

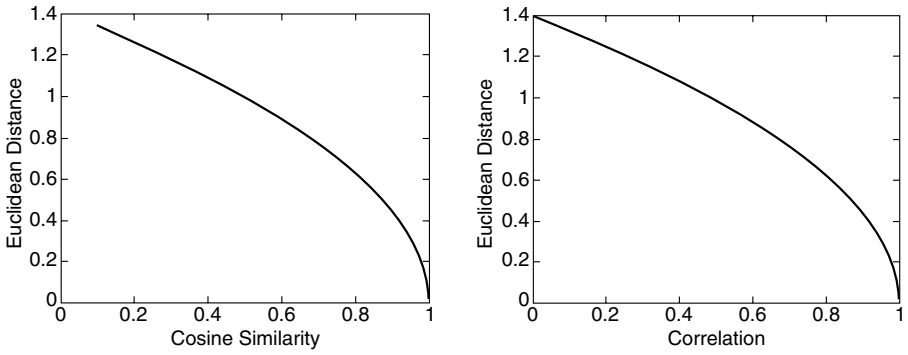
(c) What is the relationship of the cosine measure to correlation, if any? (Hint: Look at statistical measures such as mean and standard deviation in cases where cosine and correlation are the same and different.)

(d) Figure 2.20(a) shows the relationship of the cosine measure to Euclidean distance for 100,000 randomly generated points that have been normalized to have an L2 length of 1. What general observation can you make about the relationship between Euclidean distance and cosine similarity when vectors have an L2 norm of 1?

(e) Figure 2.20(b) shows the relationship of correlation to Euclidean distance for 100,000 randomly generated points that have been standardized to have a mean of 0 and a standard deviation of 1. What general observation can you make about the relationship between Euclidean distance and correlation when the vectors have been standardized to have a mean of 0 and a standard deviation of 1?

(f) Derive the mathematical relationship between cosine similarity and Euclidean distance when each data object has an L_2 length of 1.

(g) Derive the mathematical relationship between correlation and Euclidean distance when each data point has been been standardized by subtracting its mean and dividing by its standard deviation.



(a) Relationship between Euclidean distance and the cosine measure.

(b) Relationship between Euclidean distance and correlation.

Figure 2.20. Graphs for Exercise 20.

21. Show that the set difference metric given by

$$d(A, B) = size(A - B) + size(B - A) \tag{2.19}$$

satisfies the metric axioms given on page 70. A and B are sets and $A - B$ is the set difference.

22. Discuss how you might map correlation values from the interval [−1,1] to the interval [0,1]. Note that the type of transformation that you use might depend on the application that you have in mind. Thus, consider two applications: clustering time series and predicting the behavior of one time series given another.

23. Given a similarity measure with values in the interval [0,1] describe two ways to transform this similarity value into a dissimilarity value in the interval [0,∞].

24. Proximity is typically defined between a pair of objects.

    (a) Define two ways in which you might define the proximity among a group of objects.

    (b) How might you define the distance between two sets of points in Euclidean space?

    (c) How might you define the proximity between two sets of data objects? (Make no assumption about the data objects, except that a proximity measure is defined between any pair of objects.)

25. You are given a set of points S in Euclidean space, as well as the distance of each point in S to a point $\mathbf{x}$. (It does not matter if $\mathbf{x} \in S$.)

(a) If the goal is to find all points within a specified distance ε of point $\mathbf{y}$, $\mathbf{y} \neq \mathbf{x}$, explain how you could use the triangle inequality and the already calculated distances to $\mathbf{x}$ to potentially reduce the number of distance calculations necessary? Hint: The triangle inequality, $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{x})$, can be rewritten as $d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$.

(b) In general, how would the distance between $\mathbf{x}$ and $\mathbf{y}$ affect the number of distance calculations?

(c) Suppose that you can find a small subset of points S', from the original data set, such that every point in the data set is within a specified distance ε of at least one of the points in S', and that you also have the pairwise distance matrix for S'. Describe a technique that uses this information to compute, with a minimum of distance calculations, the set of all points within a distance of β of a specified point from the data set.

26. Show that 1 minus the Jaccard similarity is a distance measure between two data objects, $\mathbf{x}$ and $\mathbf{y}$, that satisfies the metric axioms given on page 70. Specifically, $d(\mathbf{x}, \mathbf{y}) = 1 - \mathrm{J}(\mathbf{x}, \mathbf{y})$.

27. Show that the distance measure defined as the angle between two data vectors, $\mathbf{x}$ and $\mathbf{y}$, satisfies the metric axioms given on page 70. Specifically, $d(\mathbf{x}, \mathbf{y}) = \arccos(\cos(\mathbf{x}, \mathbf{y}))$.

28. Explain why computing the proximity between two attributes is often simpler than computing the similarity between two objects.

3

Exploring Data

The previous chapter addressed high-level data issues that are important in the knowledge discovery process. This chapter provides an introduction to **data exploration**, which is a preliminary investigation of the data in order to better understand its specific characteristics. Data exploration can aid in selecting the appropriate preprocessing and data analysis techniques. It can even address some of the questions typically answered by data mining. For example, patterns can sometimes be found by visually inspecting the data. Also, some of the techniques used in data exploration, such as visualization, can be used to understand and interpret data mining results.

This chapter covers three major topics: summary statistics, visualization, and On-Line Analytical Processing (OLAP). Summary statistics, such as the mean and standard deviation of a set of values, and visualization techniques, such as histograms and scatter plots, are standard methods that are widely employed for data exploration. OLAP, which is a more recent development, consists of a set of techniques for exploring multidimensional arrays of values. OLAP-related analysis functions focus on various ways to create summary data tables from a multidimensional data array. These techniques include aggregating data either across various dimensions or across various attribute values. For instance, if we are given sales information reported according to product, location, and date, OLAP techniques can be used to create a summary that describes the sales activity at a particular location by month and product category.

The topics covered in this chapter have considerable overlap with the area known as **Exploratory Data Analysis** (EDA), which was created in the 1970s by the prominent statistician, John Tukey. This chapter, like EDA, places a heavy emphasis on visualization. Unlike EDA, this chapter does not include topics such as cluster analysis or anomaly detection. There are two

reasons for this. First, data mining views descriptive data analysis techniques as an end in themselves, whereas statistics, from which EDA originated, tends to view hypothesis-based testing as the final goal. Second, cluster analysis and anomaly detection are large areas and require full chapters for an in-depth discussion. Hence, cluster analysis is covered in Chapters 8 and 9, while anomaly detection is discussed in Chapter 10.

3.1 The Iris Data Set

In the following discussion, we will often refer to the Iris data set that is available from the University of California at Irvine (UCI) Machine Learning Repository. It consists of information on 150 Iris flowers, 50 each from one of three Iris species: Setosa, Versicolour, and Virginica. Each flower is characterized by five attributes:

1. sepal length in centimeters
2. sepal width in centimeters
3. petal length in centimeters
4. petal width in centimeters
5. class (Setosa, Versicolour, Virginica)

The sepals of a flower are the outer structures that protect the more fragile parts of the flower, such as the petals. In many flowers, the sepals are green, and only the petals are colorful. For Irises, however, the sepals are also colorful. As illustrated by the picture of a Virginica Iris in Figure 3.1, the sepals of an Iris are larger than the petals and are drooping, while the petals are upright.

3.2 Summary Statistics

Summary statistics are quantities, such as the mean and standard deviation, that capture various characteristics of a potentially large set of values with a single number or a small set of numbers. Everyday examples of summary statistics are the average household income or the fraction of college students who complete an undergraduate degree in four years. Indeed, for many people, summary statistics are the most visible manifestation of statistics. We will concentrate on summary statistics for the values of a single attribute, but will provide a brief description of some multivariate summary statistics.

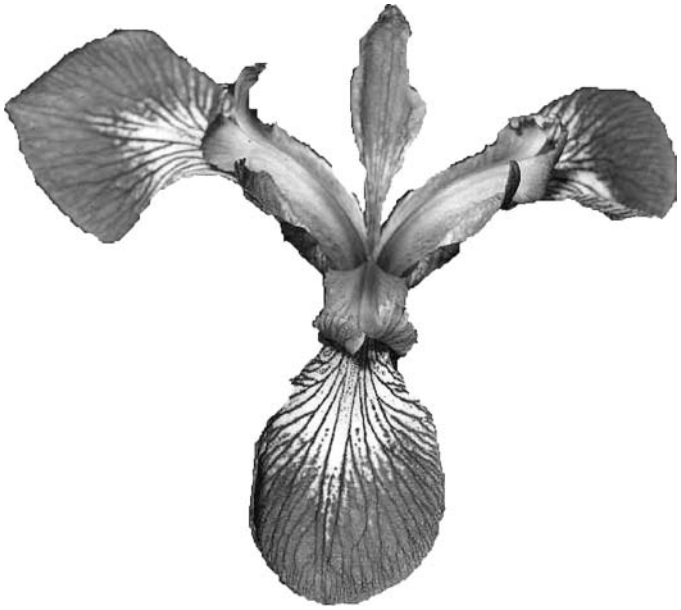


Figure 3.1. Picture of Iris Virginica. Robert H. Mohlenbrock @ USDA-NRCS PLANTS Database/ USDA NRCS. 1995. Northeast wetland flora: Field office guide to plant species. Northeast National Technical Center, Chester, PA. Background removed.

This section considers only the descriptive nature of summary statistics. However, as described in Appendix C, statistics views data as arising from an underlying statistical process that is characterized by various parameters, and some of the summary statistics discussed here can be viewed as estimates of statistical parameters of the underlying distribution that generated the data.

3.2.1 Frequencies and the Mode

Given a set of unordered categorical values, there is not much that can be done to further characterize the values except to compute the frequency with which each value occurs for a particular set of data. Given a categorical attribute x, which can take values $\{v_1, \ldots, v_i, \ldots v_k\}$ and a set of m objects, the frequency of a value v_i is defined as

$$\text{frequency}(v_i) = \frac{\text{number of objects with attribute value } v_i}{m}. \tag{3.1}$$

The **mode** of a categorical attribute is the value that has the highest frequency.

Example 3.1. Consider a set of students who have an attribute, *class*, which can take values from the set $\{freshman, sophomore, junior, senior\}$. Table 3.1 shows the number of students for each value of the *class* attribute. The mode of the *class* attribute is $freshman$, with a frequency of 0.33. This may indicate dropouts due to attrition or a larger than usual freshman class.

Table 3.1. Class size for students in a hypothetical college.

Class	Size	Frequency
freshman	140	0.33
sophomore	160	0.27
junior	130	0.22
senior	170	0.18

■

Categorical attributes often, but not always, have a small number of values, and consequently, the mode and frequencies of these values can be interesting and useful. Notice, though, that for the Iris data set and the *class* attribute, the three types of flower all have the same frequency, and therefore, the notion of a mode is not interesting.

For continuous data, the mode, as currently defined, is often not useful because a single value may not occur more than once. Nonetheless, in some cases, the mode may indicate important information about the nature of the values or the presence of missing values. For example, the heights of 20 people measured to the nearest millimeter will typically not repeat, but if the heights are measured to the nearest tenth of a meter, then some people may have the same height. Also, if a unique value is used to indicate a missing value, then this value will often show up as the mode.

3.2.2 Percentiles

For ordered data, it is more useful to consider the **percentiles** of a set of values. In particular, given an ordinal or continuous attribute x and a number p between 0 and 100, the p^{th} percentile x_p is a value of x such that $p\%$ of the observed values of x are less than x_p. For instance, the 50^{th} percentile is the value $x_{50\%}$ such that 50% of all values of x are less than $x_{50\%}$. Table 3.2 shows the percentiles for the four quantitative attributes of the Iris data set.

Table 3.2. Percentiles for sepal length, sepal width, petal length, and petal width. (All values are in centimeters.)

Percentile	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	2.0	1.0	0.1
10	4.8	2.5	1.4	0.2
20	5.0	2.7	1.5	0.2
30	5.2	2.8	1.7	0.4
40	5.6	3.0	3.9	1.2
50	5.8	3.0	4.4	1.3
60	6.1	3.1	4.6	1.5
70	6.3	3.2	5.0	1.8
80	6.6	3.4	5.4	1.9
90	6.9	3.6	5.8	2.2
100	7.9	4.4	6.9	2.5

Example 3.2. The percentiles, $x_{0\%}, x_{10\%}, \ldots, x_{90\%}, x_{100\%}$ of the integers from 1 to 10 are, in order, the following: 1.0, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.0. By tradition, $\min(x) = x_{0\%}$ and $\max(x) = x_{100\%}$. ■

3.2.3 Measures of Location: Mean and Median

For continuous data, two of the most widely used summary statistics are the **mean** and **median**, which are measures of the *location* of a set of values. Consider a set of m objects and an attribute x. Let $\{x_1, \ldots, x_m\}$ be the attribute values of x for these m objects. As a concrete example, these values might be the heights of m children. Let $\{x_{(1)}, \ldots, x_{(m)}\}$ represent the values of x after they have been sorted in non-decreasing order. Thus, $x_{(1)} = \min(x)$ and $x_{(m)} = \max(x)$. Then, the mean and median are defined as follows:

$$\text{mean}(x) = \overline{x} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{3.2}$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r + 1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases} \tag{3.3}$$

To summarize, the median is the middle value if there are an odd number of values, and the average of the two middle values if the number of values is even. Thus, for seven values, the median is $x_{(4)}$, while for ten values, the median is $\frac{1}{2}(x_{(5)} + x_{(6)})$.

Although the mean is sometimes interpreted as the middle of a set of values, this is only correct if the values are distributed in a symmetric manner. If the distribution of values is skewed, then the median is a better indicator of the middle. Also, the mean is sensitive to the presence of outliers. For data with outliers, the median again provides a more robust estimate of the middle of a set of values.

To overcome problems with the traditional definition of a mean, the notion of a **trimmed mean** is sometimes used. A percentage p between 0 and 100 is specified, the top and bottom $(p/2)\%$ of the data is thrown out, and the mean is then calculated in the normal way. The median is a trimmed mean with $p = 100\%$, while the standard mean corresponds to $p = 0\%$.

Example 3.3. Consider the set of values $\{1, 2, 3, 4, 5, 90\}$. The mean of these values is 17.5, while the median is 3.5. The trimmed mean with $p = 40\%$ is also 3.5. ■

Example 3.4. The means, medians, and trimmed means ($p = 20\%$) of the four quantitative attributes of the Iris data are given in Table 3.3. The three measures of location have similar values except for the attribute *petal length*.

Table 3.3. Means and medians for sepal length, sepal width, petal length, and petal width. (All values are in centimeters.)

Measure	Sepal Length	Sepal Width	Petal Length	Petal Width
mean	5.84	3.05	3.76	1.20
median	5.80	3.00	4.35	1.30
trimmed mean (20%)	5.79	3.02	3.72	1.12

■

3.2.4 Measures of Spread: Range and Variance

Another set of commonly used summary statistics for continuous data are those that measure the dispersion or spread of a set of values. Such measures indicate if the attribute values are widely spread out or if they are relatively concentrated around a single point such as the mean.

The simplest measure of spread is the **range**, which, given an attribute x with a set of m values $\{x_1, \ldots, x_m\}$, is defined as

$$\text{range}(x) = \max(x) - \min(x) = x_{(m)} - x_{(1)}. \tag{3.4}$$

Table 3.4. Range, standard deviation (std), absolute average difference (AAD), median absolute difference (MAD), and interquartile range (IQR) for sepal length, sepal width, petal length, and petal width. (All values are in centimeters.)

Measure	Sepal Length	Sepal Width	Petal Length	Petal Width
range	3.6	2.4	5.9	2.4
std	0.8	0.4	1.8	0.8
AAD	0.7	0.3	1.6	0.6
MAD	0.7	0.3	1.2	0.7
IQR	1.3	0.5	3.5	1.5

Although the range identifies the maximum spread, it can be misleading if most of the values are concentrated in a narrow band of values, but there are also a relatively small number of more extreme values. Hence, the **variance** is preferred as a measure of spread. The variance of the (observed) values of an attribute x is typically written as s_x^2 and is defined below. The **standard deviation**, which is the square root of the variance, is written as s_x and has the same units as x.

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2 \tag{3.5}$$

The mean can be distorted by outliers, and since the variance is computed using the mean, it is also sensitive to outliers. Indeed, the variance is particularly sensitive to outliers since it uses the squared difference between the mean and other values. As a result, more robust estimates of the spread of a set of values are often used. Following are the definitions of three such measures: the **absolute average deviation** (AAD), the **median absolute deviation** (MAD), and the **interquartile range**(IQR). Table 3.4 shows these measures for the Iris data set.

$$\text{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \overline{x}| \tag{3.6}$$

$$\text{MAD}(x) = median\Big(\{|x_1 - \overline{x}|, \ldots, |x_m - \overline{x}|\}\Big) \tag{3.7}$$

$$\text{interquartile range}(x) = x_{75\%} - x_{25\%} \tag{3.8}$$

3.2.5 Multivariate Summary Statistics

Measures of location for data that consists of several attributes (multivariate data) can be obtained by computing the mean or median separately for each attribute. Thus, given a data set the mean of the data objects, $\overline{\mathbf{x}}$, is given by

$$\overline{\mathbf{x}} = (\overline{x_1}, \ldots, \overline{x_n}), \tag{3.9}$$

where $\overline{x_i}$ is the mean of the i^{th} attribute x_i.

For multivariate data, the spread of each attribute can be computed independently of the other attributes using any of the approaches described in Section 3.2.4. However, for data with continuous variables, the spread of the data is most commonly captured by the **covariance matrix S**, whose ij^{th} entry s_{ij} is the covariance of the i^{th} and j^{th} attributes of the data. Thus, if x_i and x_j are the i^{th} and j^{th} attributes, then

$$s_{ij} = \text{covariance}(x_i, x_j). \tag{3.10}$$

In turn, $covariance(x_i, x_j)$ is given by

$$\text{covariance}(x_i, x_j) = \frac{1}{m-1}\sum_{k=1}^{m}(x_{ki} - \overline{x_i})(x_{kj} - \overline{x_j}), \tag{3.11}$$

where x_{ki} and x_{kj} are the values of the i^{th} and j^{th} attributes for the k^{th} object. Notice that $\text{covariance}(x_i, x_i) = \text{variance}(x_i)$. Thus, the covariance matrix has the variances of the attributes along the diagonal.

The covariance of two attributes is a measure of the degree to which two attributes vary together and depends on the magnitudes of the variables. A value near 0 indicates that two attributes do not have a (linear) relationship, but it is not possible to judge the degree of relationship between two variables by looking only at the value of the covariance. Because the correlation of two attributes immediately gives an indication of how strongly two attributes are (linearly) related, correlation is preferred to covariance for data exploration. (Also see the discussion of correlation in Section 2.4.5.) The ij^{th} entry of the **correlation matrix R**, is the correlation between the i^{th} and j^{th} attributes of the data. If x_i and x_j are the i^{th} and j^{th} attributes, then

$$r_{ij} = \text{correlation}(x_i, x_j) = \frac{\text{covariance}(x_i, x_j)}{s_i s_j}, \tag{3.12}$$

where s_i and s_j are the variances of x_i and x_j, respectively. The diagonal entries of $\mathbf{R}$ are correlation$(x_i, x_i) = 1$, while the other entries are between -1 and 1. It is also useful to consider correlation matrices that contain the pairwise correlations of objects instead of attributes.

3.2.6 Other Ways to Summarize the Data

There are, of course, other types of summary statistics. For instance, the **skewness** of a set of values measures the degree to which the values are symmetrically distributed around the mean. There are also other characteristics of the data that are not easy to measure quantitatively, such as whether the distribution of values is multimodal; i.e., the data has multiple "bumps" where most of the values are concentrated. In many cases, however, the most effective approach to understanding the more complicated or subtle aspects of how the values of an attribute are distributed, is to view the values graphically in the form of a histogram. (Histograms are discussed in the next section.)

3.3 Visualization

Data visualization is the display of information in a graphic or tabular format. Successful visualization requires that the data (information) be converted into a visual format so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported. The goal of visualization is the interpretation of the visualized information by a person and the formation of a mental model of the information.

In everyday life, visual techniques such as graphs and tables are often the preferred approach used to explain the weather, the economy, and the results of political elections. Likewise, while algorithmic or mathematical approaches are often emphasized in most technical disciplines—data mining included—visual techniques can play a key role in data analysis. In fact, sometimes the use of visualization techniques in data mining is referred to as **visual data mining**.

3.3.1 Motivations for Visualization

The overriding motivation for using visualization is that people can quickly absorb large amounts of visual information and find patterns in it. Consider Figure 3.2, which shows the Sea Surface Temperature (SST) in degrees Celsius for July, 1982. This picture summarizes the information from approximately 250,000 numbers and is readily interpreted in a few seconds. For example, it

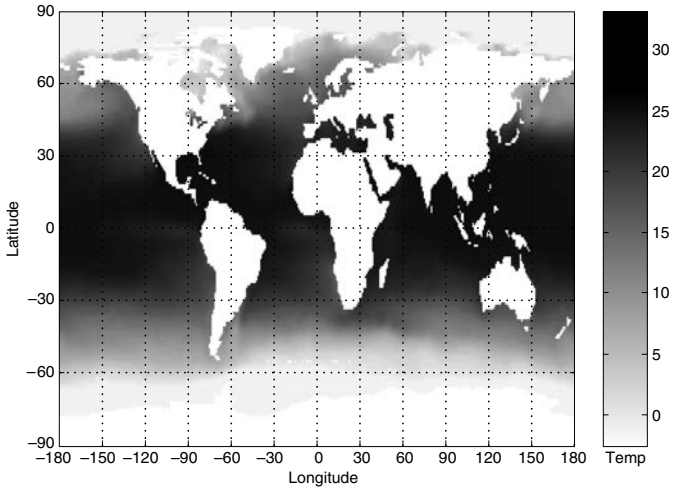


Figure 3.2. Sea Surface Temperature (SST) for July, 1982.

is easy to see that the ocean temperature is highest at the equator and lowest at the poles.

Another general motivation for visualization is to make use of the domain knowledge that is "locked up in people's heads." While the use of domain knowledge is an important task in data mining, it is often difficult or impossible to fully utilize such knowledge in statistical or algorithmic tools. In some cases, an analysis can be performed using non-visual tools, and then the results presented visually for evaluation by the domain expert. In other cases, having a domain specialist examine visualizations of the data may be the best way of finding patterns of interest since, by using domain knowledge, a person can often quickly eliminate many uninteresting patterns and direct the focus to the patterns that are important.

3.3.2 General Concepts

This section explores some of the general concepts related to visualization, in particular, general approaches for visualizing the data and its attributes. A number of visualization techniques are mentioned briefly and will be described in more detail when we discuss specific approaches later on. We assume that the reader is familiar with line graphs, bar charts, and scatter plots.

Representation: Mapping Data to Graphical Elements

The first step in visualization is the mapping of information to a visual format; i.e., mapping the objects, attributes, and relationships in a set of information to visual objects, attributes, and relationships. That is, data objects, their attributes, and the relationships among data objects are translated into graphical elements such as points, lines, shapes, and colors.

Objects are usually represented in one of three ways. First, if only a single categorical attribute of the object is being considered, then objects are often lumped into categories based on the value of that attribute, and these categories are displayed as an entry in a table or an area on a screen. (Examples shown later in this chapter are a cross-tabulation table and a bar chart.) Second, if an object has multiple attributes, then the object can be displayed as a row (or column) of a table or as a line on a graph. Finally, an object is often interpreted as a point in two- or three-dimensional space, where graphically, the point might be represented by a geometric figure, such as a circle, cross, or box.

For attributes, the representation depends on the type of attribute, i.e., nominal, ordinal, or continuous (interval or ratio). Ordinal and continuous attributes can be mapped to continuous, ordered graphical features such as location along the x, y, or z axes; intensity; color; or size (diameter, width, height, etc.). For categorical attributes, each category can be mapped to a distinct position, color, shape, orientation, embellishment, or column in a table. However, for nominal attributes, whose values are unordered, care should be taken when using graphical features, such as color and position that have an inherent ordering associated with their values. In other words, the graphical elements used to represent the ordinal values often have an order, but ordinal values do not.

The representation of relationships via graphical elements occurs either explicitly or implicitly. For graph data, the standard graph representation—a set of nodes with links between the nodes—is normally used. If the nodes (data objects) or links (relationships) have attributes or characteristics of their own, then this is represented graphically. To illustrate, if the nodes are cities and the links are highways, then the diameter of the nodes might represent population, while the width of the links might represent the volume of traffic.

In most cases, though, mapping objects and attributes to graphical elements implicitly maps the relationships in the data to relationships among graphical elements. To illustrate, if the data object represents a physical object that has a location, such as a city, then the relative positions of the graphical objects corresponding to the data objects tend to naturally preserve the actual

relative positions of the objects. Likewise, if there are two or three continuous attributes that are taken as the coordinates of the data points, then the resulting plot often gives considerable insight into the relationships of the attributes and the data points because data points that are visually close to each other have similar values for their attributes.

In general, it is difficult to ensure that a mapping of objects and attributes will result in the relationships being mapped to easily observed relationships among graphical elements. Indeed, this is one of the most challenging aspects of visualization. In any given set of data, there are many implicit relationships, and hence, a key challenge of visualization is to choose a technique that makes the relationships of interest easily observable.

Arrangement

As discussed earlier, the proper choice of visual representation of objects and attributes is essential for good visualization. The arrangement of items within the visual display is also crucial. We illustrate this with two examples.

Example 3.5. This example illustrates the importance of rearranging a table of data. In Table 3.5, which shows nine objects with six binary attributes, there is no clear relationship between objects and attributes, at least at first glance. If the rows and columns of this table are permuted, however, as shown in Table 3.6, then it is clear that there are really only two types of objects in the table—one that has all ones for the first three attributes and one that has only ones for the last three attributes. ■

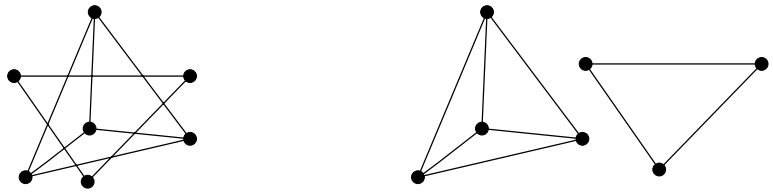
Table 3.5. A table of nine objects (rows) with six binary attributes (columns).

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

Table 3.6. A table of nine objects (rows) with six binary attributes (columns) permuted so that the relationships of the rows and columns are clear.

	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

Example 3.6. Consider Figure 3.3(a), which shows a visualization of a graph. If the connected components of the graph are separated, as in Figure 3.3(b), then the relationships between nodes and graphs become much simpler to understand. ■



(a) Original view of a graph.

(b) Uncoupled view of connected components of the graph.

Figure 3.3. Two visualizations of a graph.

Selection

Another key concept in visualization is **selection**, which is the elimination or the de-emphasis of certain objects and attributes. Specifically, while data objects that only have a few dimensions can often be mapped to a two- or three-dimensional graphical representation in a straightforward way, there is no completely satisfactory and general approach to represent data with many attributes. Likewise, if there are many data objects, then visualizing all the objects can result in a display that is too crowded. If there are many attributes and many objects, then the situation is even more challenging.

The most common approach to handling many attributes is to choose a subset of attributes—usually two—for display. If the dimensionality is not too high, a matrix of bivariate (two-attribute) plots can be constructed for simultaneous viewing. (Figure 3.16 shows a matrix of scatter plots for the pairs of attributes of the Iris data set.) Alternatively, a visualization program can automatically show a series of two-dimensional plots, in which the sequence is user directed or based on some predefined strategy. The hope is that visualizing a collection of two-dimensional plots will provide a more complete view of the data.

The technique of selecting a pair (or small number) of attributes is a type of dimensionality reduction, and there are many more sophisticated dimensionality reduction techniques that can be employed, e.g., principal components analysis (PCA). Consult Appendices A (Linear Algebra) and B (Dimensionality Reduction) for more information.

When the number of data points is high, e.g., more than a few hundred, or if the range of the data is large, it is difficult to display enough information about each object. Some data points can obscure other data points, or a data object may not occupy enough pixels to allow its features to be clearly displayed. For example, the shape of an object cannot be used to encode a characteristic of that object if there is only one pixel available to display it. In these situations, it is useful to be able to eliminate some of the objects, either by zooming in on a particular region of the data or by taking a sample of the data points.

3.3.3 Techniques

Visualization techniques are often specialized to the type of data being analyzed. Indeed, new visualization techniques and approaches, as well as specialized variations of existing approaches, are being continuously created, typically in response to new kinds of data and visualization tasks.

Despite this specialization and the ad hoc nature of visualization, there are some generic ways to classify visualization techniques. One such classification is based on the number of attributes involved (1, 2, 3, or many) or whether the data has some special characteristic, such as a hierarchical or graph structure. Visualization methods can also be classified according to the type of attributes involved. Yet another classification is based on the type of application: scientific, statistical, or information visualization. The following discussion will use three categories: visualization of a small number of attributes, visualization of data with spatial and/or temporal attributes, and visualization of data with many attributes.

Most of the visualization techniques discussed here can be found in a wide variety of mathematical and statistical packages, some of which are freely available. There are also a number of data sets that are freely available on the World Wide Web. Readers are encouraged to try these visualization techniques as they proceed through the following sections.

Visualizing Small Numbers of Attributes

This section examines techniques for visualizing data with respect to a small number of attributes. Some of these techniques, such as histograms, give insight into the distribution of the observed values for a single attribute. Other techniques, such as scatter plots, are intended to display the relationships between the values of two attributes.

Stem and Leaf Plots Stem and leaf plots can be used to provide insight into the distribution of one-dimensional integer or continuous data. (We will assume integer data initially, and then explain how stem and leaf plots can be applied to continuous data.) For the simplest type of stem and leaf plot, we split the values into groups, where each group contains those values that are the same except for the last digit. Each group becomes a stem, while the last digits of a group are the leaves. Hence, if the values are two-digit integers, e.g., 35, 36, 42, and 51, then the stems will be the high-order digits, e.g., 3, 4, and 5, while the leaves are the low-order digits, e.g., 1, 2, 5, and 6. By plotting the stems vertically and leaves horizontally, we can provide a visual representation of the distribution of the data.

Example 3.7. The set of integers shown in Figure 3.4 is the sepal length in centimeters (multiplied by 10 to make the values integers) taken from the Iris data set. For convenience, the values have also been sorted.

The stem and leaf plot for this data is shown in Figure 3.5. Each number in Figure 3.4 is first put into one of the vertical groups—4, 5, 6, or 7—according to its ten's digit. Its last digit is then placed to the right of the colon. Often, especially if the amount of data is larger, it is desirable to split the stems. For example, instead of placing all values whose ten's digit is 4 in the same "bucket," the stem 4 is repeated twice; all values 40–44 are put in the bucket corresponding to the first stem and all values 45–49 are put in the bucket corresponding to the second stem. This approach is shown in the stem and leaf plot of Figure 3.6. Other variations are also possible. ■

Histograms Stem and leaf plots are a type of **histogram**, a plot that displays the distribution of values for attributes by dividing the possible values into bins and showing the number of objects that fall into each bin. For categorical data, each value is a bin. If this results in too many values, then values are combined in some way. For continuous attributes, the range of values is divided into bins—typically, but not necessarily, of equal width—and the values in each bin are counted.

```
43 44 44 44 45 46 46 46 46 47 47 48 48 48 48 48 49 49 49 49 49 49 50
50 50 50 50 50 50 50 50 50 51 51 51 51 51 51 51 51 51 52 52 52 52 53
54 54 54 54 54 54 55 55 55 55 55 55 55 56 56 56 56 56 56 57 57 57 57
57 57 57 57 58 58 58 58 58 58 58 59 59 59 60 60 60 60 60 60 61 61 61
61 61 61 62 62 62 62 63 63 63 63 63 63 63 63 63 64 64 64 64 64 64 64
65 65 65 65 65 66 66 67 67 67 67 67 67 67 67 68 68 68 69 69 69 69 70
71 72 72 72 73 74 76 77 77 77 77 79
```

Figure 3.4. Sepal length data from the Iris data set.

```
4 :  34444566667788888999999
5 :  000000000011111111122223444444555555566666677777778888888999
6 :  00000011111122223333333334444444555556677777778889999
7 :  0122234677779
```

Figure 3.5. Stem and leaf plot for the sepal length from the Iris data set.

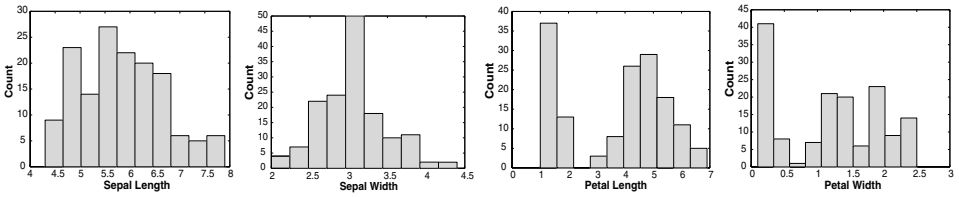
```
4 :  3444
4 :  566667788888999999
5 :  00000000001111111122223444444
5 :  55555556666667777777788888888999
6 :  000000111111222233333333344444444
6 :  555556677777777889999
7 :  0122234
7 :  677779
```

Figure 3.6. Stem and leaf plot for the sepal length from the Iris data set when buckets corresponding to digits are split.

Once the counts are available for each bin, a **bar plot** is constructed such that each bin is represented by one bar and the area of each bar is proportional to the number of values (objects) that fall into the corresponding range. If all intervals are of equal width, then all bars are the same width and the height of a bar is proportional to the number of values in the corresponding bin.

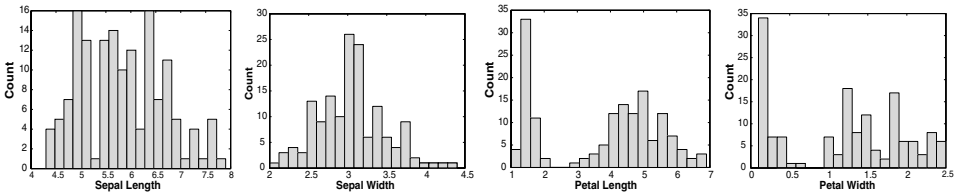
Example 3.8. Figure 3.7 shows histograms (with 10 bins) for sepal length, sepal width, petal length, and petal width. Since the shape of a histogram can depend on the number of bins, histograms for the same data, but with 20 bins, are shown in Figure 3.8. ■

There are variations of the histogram plot. A **relative (frequency) histogram** replaces the count by the relative frequency. However, this is just a



(a) Sepal length. (b) Sepal width. (c) Petal length. (d) Petal width.

Figure 3.7. Histograms of four Iris attributes (10 bins).



(a) Sepal length. (b) Sepal width. (c) Petal length. (d) Petal width.

Figure 3.8. Histograms of four Iris attributes (20 bins).

change in scale of the y axis, and the shape of the histogram does not change. Another common variation, especially for unordered categorical data, is the **Pareto histogram**, which is the same as a normal histogram except that the categories are sorted by count so that the count is decreasing from left to right.

Two-Dimensional Histograms Two-dimensional histograms are also possible. Each attribute is divided into intervals and the two sets of intervals define two-dimensional rectangles of values.

Example 3.9. Figure 3.9 shows a two-dimensional histogram of petal length and petal width. Because each attribute is split into three bins, there are nine rectangular two-dimensional bins. The height of each rectangular bar indicates the number of objects (flowers in this case) that fall into each bin. Most of the flowers fall into only three of the bins—those along the diagonal. It is not possible to see this by looking at the one-dimensional distributions. ■

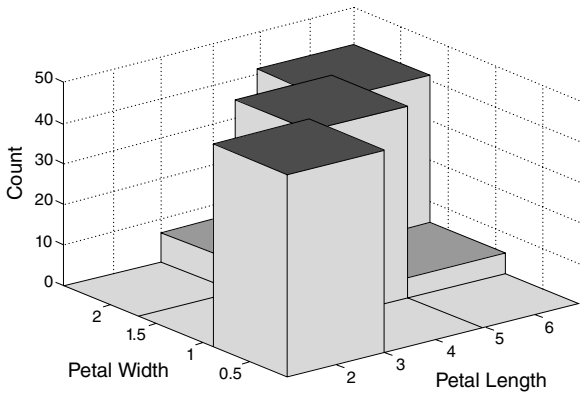


Figure 3.9. Two-dimensional histogram of petal length and width in the Iris data set.

While two-dimensional histograms can be used to discover interesting facts about how the values of two attributes co-occur, they are visually more complicated. For instance, it is easy to imagine a situation in which some of the columns are hidden by others.

Box Plots Box plots are another method for showing the distribution of the values of a single numerical attribute. Figure 3.10 shows a labeled box plot for sepal length. The lower and upper ends of the box indicate the 25^{th} and 75^{th} percentiles, respectively, while the line inside the box indicates the value of the 50^{th} percentile. The top and bottom lines of the **tails** indicate the 10^{th} and 90^{th} percentiles. Outliers are shown by "+" marks. Box plots are relatively compact, and thus, many of them can be shown on the same plot. Simplified versions of the box plot, which take less space, can also be used.

Example 3.10. The box plots for the first four attributes of the Iris data set are shown in Figure 3.11. Box plots can also be used to compare how attributes vary between different classes of objects, as shown in Figure 3.12. ■

Pie Chart A pie chart is similar to a histogram, but is typically used with categorical attributes that have a relatively small number of values. Instead of showing the relative frequency of different values with the area or height of a bar, as in a histogram, a pie chart uses the relative area of a circle to indicate relative frequency. Although pie charts are common in popular articles, they

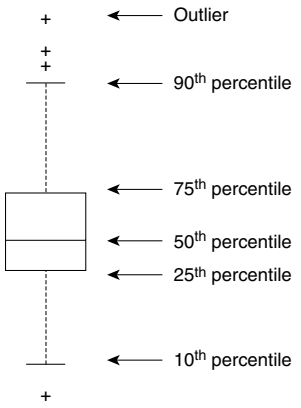


Figure 3.10. Description of box plot for sepal length.

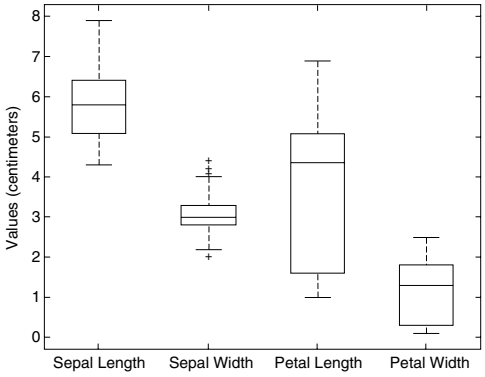
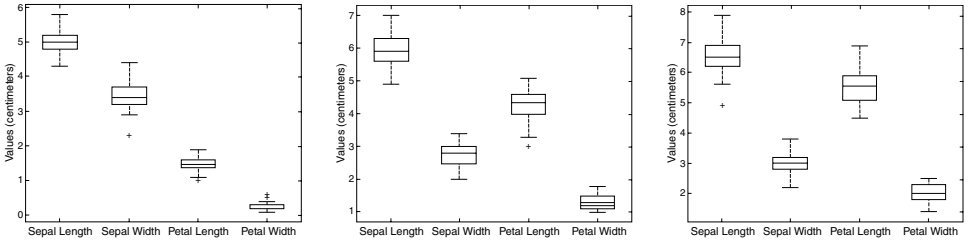


Figure 3.11. Box plot for Iris attributes.



(a) Setosa. (b) Versicolour. (c) Virginica.

Figure 3.12. Box plots of attributes by Iris species.

are used less frequently in technical publications because the size of relative areas can be hard to judge. Histograms are preferred for technical work.

Example 3.11. Figure 3.13 displays a pie chart that shows the distribution of Iris species in the Iris data set. In this case, all three flower types have the same frequency. ■

Percentile Plots and Empirical Cumulative Distribution Functions A type of diagram that shows the distribution of the data more quantitatively is the plot of an empirical cumulative distribution function. While this type of plot may sound complicated, the concept is straightforward. For each value of a statistical distribution, a **cumulative distribution function** (CDF) shows

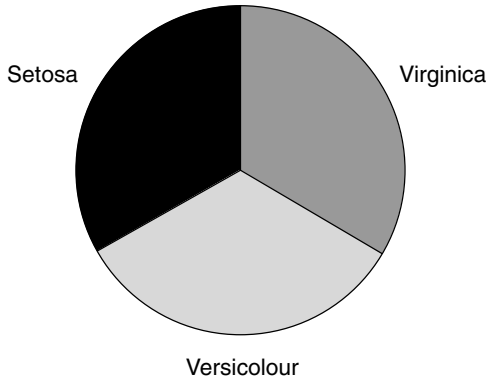


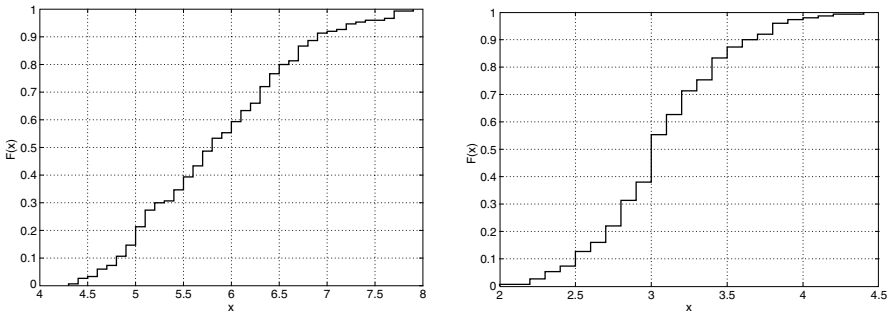
Figure 3.13. Distribution of the types of Iris flowers.

the probability that a point is less than that value. For each observed value, an **empirical cumulative distribution function** (ECDF) shows the fraction of points that are less than this value. Since the number of points is finite, the empirical cumulative distribution function is a step function.

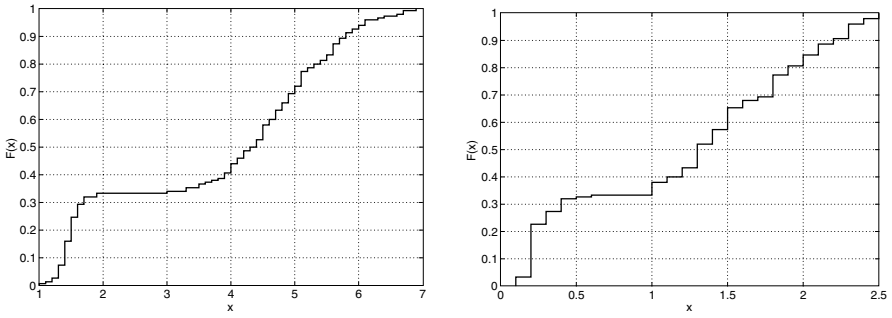
Example 3.12. Figure 3.14 shows the ECDFs of the Iris attributes. The percentiles of an attribute provide similar information. Figure 3.15 shows the **percentile plots** of the four continuous attributes of the Iris data set from Table 3.2. The reader should compare these figures with the histograms given in Figures 3.7 and 3.8. ■

Scatter Plots Most people are familiar with scatter plots to some extent, and they were used in Section 2.4.5 to illustrate linear correlation. Each data object is plotted as a point in the plane using the values of the two attributes as x and y coordinates. It is assumed that the attributes are either integer- or real-valued.

Example 3.13. Figure 3.16 shows a scatter plot for each pair of attributes of the Iris data set. The different species of Iris are indicated by different markers. The arrangement of the scatter plots of pairs of attributes in this type of tabular format, which is known as a **scatter plot matrix**, provides an organized way to examine a number of scatter plots simultaneously. ■



(a) Sepal Length. (b) Sepal Width.



(c) Petal Length. (d) Petal Width.

Figure 3.14. Empirical CDFs of four Iris attributes.

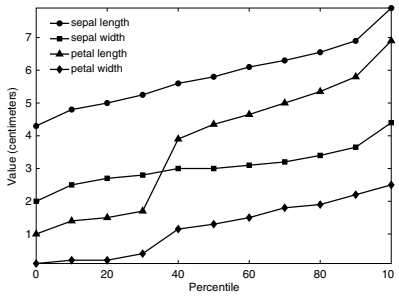


Figure 3.15. Percentile plots for sepal length, sepal width, petal length, and petal width.

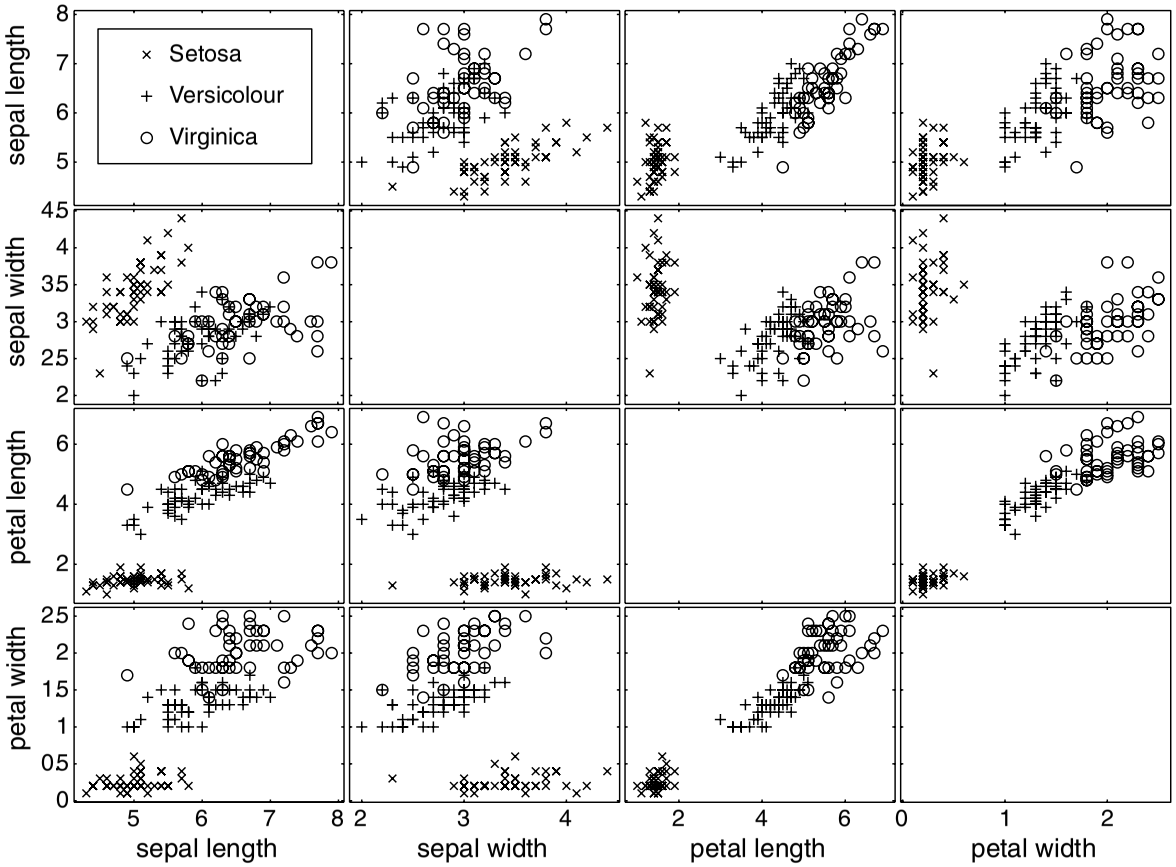


Figure 3.16. Matrix of scatter plots for the Iris data set.

There are two main uses for scatter plots. First, they graphically show the relationship between two attributes. In Section 2.4.5, we saw how scatter plots could be used to judge the degree of linear correlation. (See Figure 2.17.) Scatter plots can also be used to detect non-linear relationships, either directly or by using a scatter plot of the transformed attributes.

Second, when class labels are available, they can be used to investigate the degree to which two attributes separate the classes. If is possible to draw a line (or a more complicated curve) that divides the plane defined by the two attributes into separate regions that contain mostly objects of one class, then it is possible to construct an accurate classifier based on the specified pair of attributes. If not, then more attributes or more sophisticated methods are needed to build a classifier. In Figure 3.16, many of the pairs of attributes (for example, petal width and petal length) provide a moderate separation of the Iris species.

Example 3.14. There are two separate approaches for displaying three attributes of a data set with a scatter plot. First, each object can be displayed according to the values of three, instead of two attributes. Figure 3.17 shows a three-dimensional scatter plot for three attributes in the Iris data set. Second, one of the attributes can be associated with some characteristic of the marker, such as its size, color, or shape. Figure 3.18 shows a plot of three attributes of the Iris data set, where one of the attributes, sepal width, is mapped to the size of the marker. ■

Extending Two- and Three-Dimensional Plots As illustrated by Figure 3.18, two- or three-dimensional plots can be extended to represent a few additional attributes. For example, scatter plots can display up to three additional attributes using color or shading, size, and shape, allowing five or six dimensions to be represented. There is a need for caution, however. As the complexity of a visual representation of the data increases, it becomes harder for the intended audience to interpret the information. There is no benefit in packing six dimensions' worth of information into a two- or three-dimensional plot, if doing so makes it impossible to understand.

Visualizing Spatio-temporal Data

Data often has spatial or temporal attributes. For instance, the data may consist of a set of observations on a spatial grid, such as observations of pressure on the surface of the Earth or the modeled temperature at various grid points in the simulation of a physical object. These observations can also be

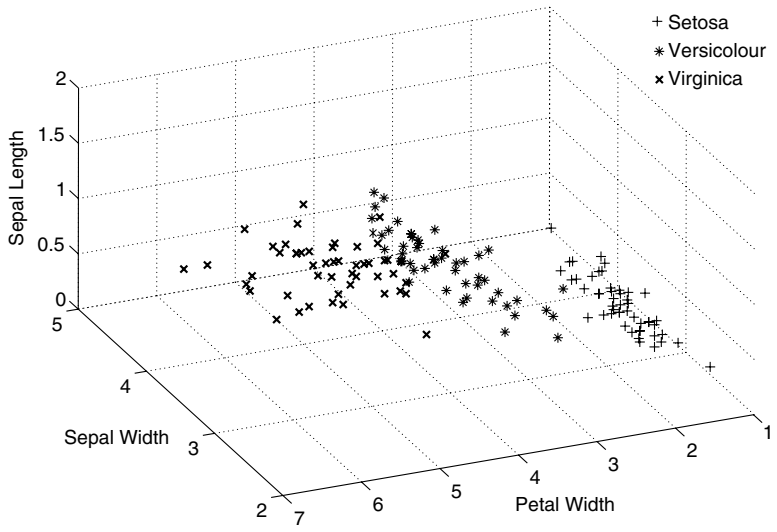


Figure 3.17. Three-dimensional scatter plot of sepal width, sepal length, and petal width.

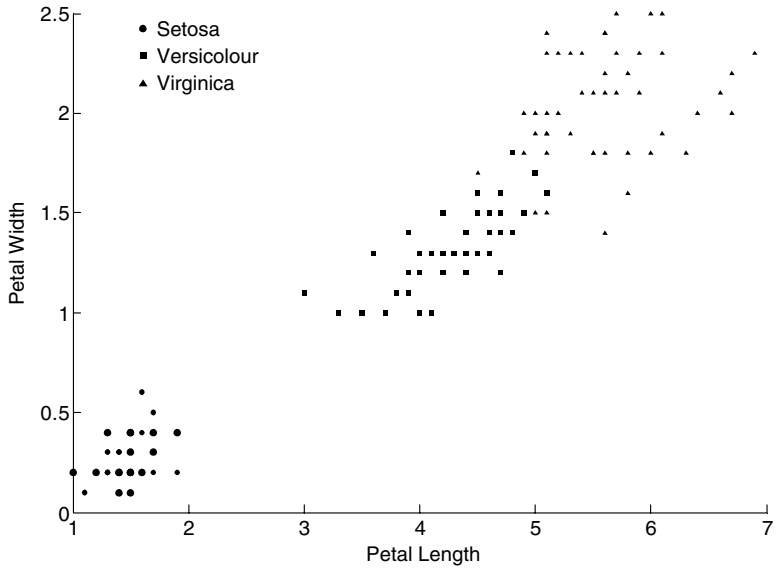


Figure 3.18. Scatter plot of petal length versus petal width, with the size of the marker indicating sepal width.

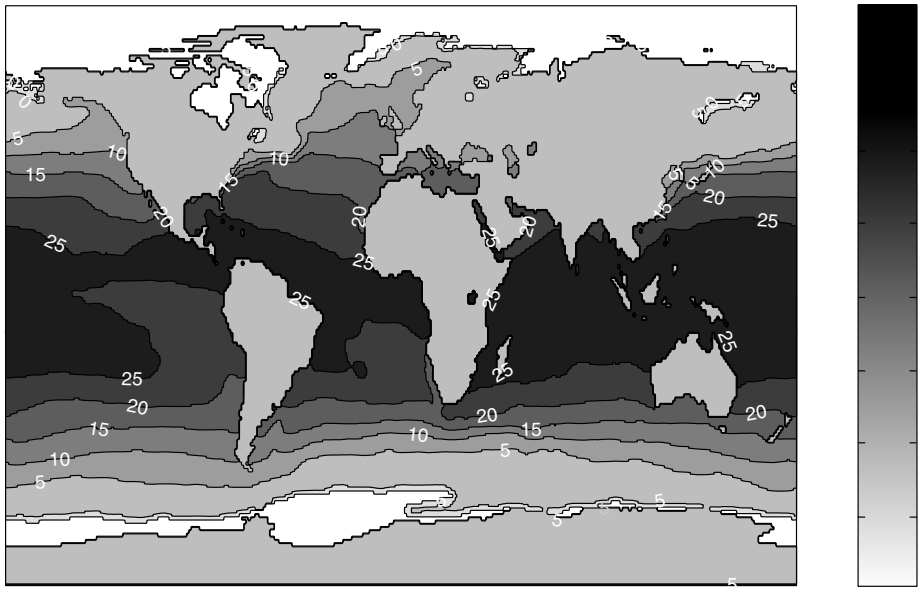


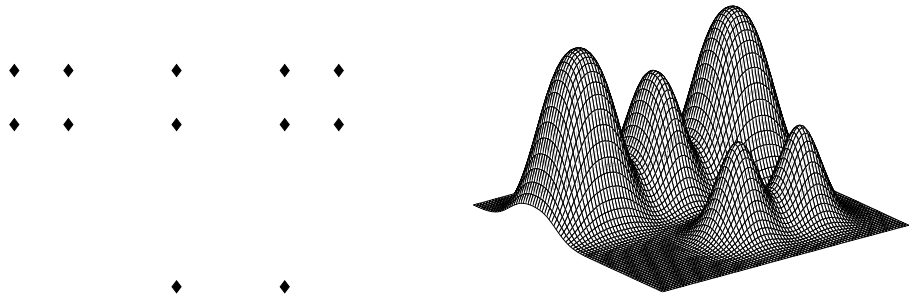
Figure 3.19. Contour plot of SST for December 1998.

made at various points in time. In addition, data may have only a temporal component, such as time series data that gives the daily prices of stocks.

Contour Plots For some three-dimensional data, two attributes specify a position in a plane, while the third has a continuous value, such as temperature or elevation. A useful visualization for such data is a **contour plot**, which breaks the plane into separate regions where the values of the third attribute (temperature, elevation) are roughly the same. A common example of a contour plot is a contour map that shows the elevation of land locations.

Example 3.15. Figure 3.19 shows a contour plot of the average sea surface temperature (SST) for December 1998. The land is arbitrarily set to have a temperature of 0°C. In many contour maps, such as that of Figure 3.19, the **contour lines** that separate two regions are labeled with the value used to separate the regions. For clarity, some of these labels have been deleted. ■

Surface Plots Like contour plots, **surface plots** use two attributes for the x and y coordinates. The third attribute is used to indicate the height above



(a) Set of 12 points.

(b) Overall density function—surface plot.

Figure 3.20. Density of a set of 12 points.

the plane defined by the first two attributes. While such graphs can be useful, they require that a value of the third attribute be defined for all combinations of values for the first two attributes, at least over some range. Also, if the surface is too irregular, then it can be difficult to see all the information, unless the plot is viewed interactively. Thus, surface plots are often used to describe mathematical functions or physical surfaces that vary in a relatively smooth manner.

Example 3.16. Figure 3.20 shows a surface plot of the density around a set of 12 points. This example is further discussed in Section 9.3.3. ■

Vector Field Plots In some data, a characteristic may have both a magnitude and a direction associated with it. For example, consider the flow of a substance or the change of density with location. In these situations, it can be useful to have a plot that displays both direction and magnitude. This type of plot is known as a **vector plot**.

Example 3.17. Figure 3.21 shows a contour plot of the density of the two smaller density peaks from Figure 3.20(b), annotated with the density gradient vectors. ■

Lower-Dimensional Slices Consider a spatio-temporal data set that records some quantity, such as temperature or pressure, at various locations over time. Such a data set has four dimensions and cannot be easily displayed by the types

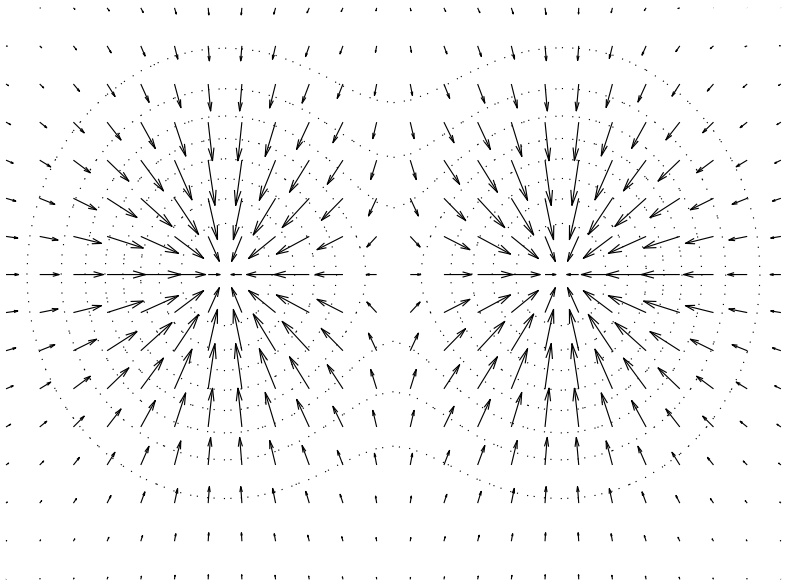


Figure 3.21. Vector plot of the gradient (change) in density for the bottom two density peaks of Figure 3.20.

of plots that we have described so far. However, separate "slices" of the data can be displayed by showing a set of plots, one for each month. By examining the change in a particular area from one month to another, it is possible to notice changes that occur, including those that may be due to seasonal factors.

Example 3.18. The underlying data set for this example consists of the average monthly sea level pressure (SLP) from 1982 to 1999 on a 2.5° by 2.5° latitude-longitude grid. The twelve monthly plots of pressure for one year are shown in Figure 3.22. In this example, we are interested in slices for a particular month in the year 1982. More generally, we can consider slices of the data along any arbitrary dimension. ■

Animation Another approach to dealing with slices of data, whether or not time is involved, is to employ animation. The idea is to display successive two-dimensional slices of the data. The human visual system is well suited to detecting visual changes and can often notice changes that might be difficult to detect in another manner. Despite the visual appeal of animation, a set of still plots, such as those of Figure 3.22, can be more useful since this type of visualization allows the information to be studied in arbitrary order and for arbitrary amounts of time.

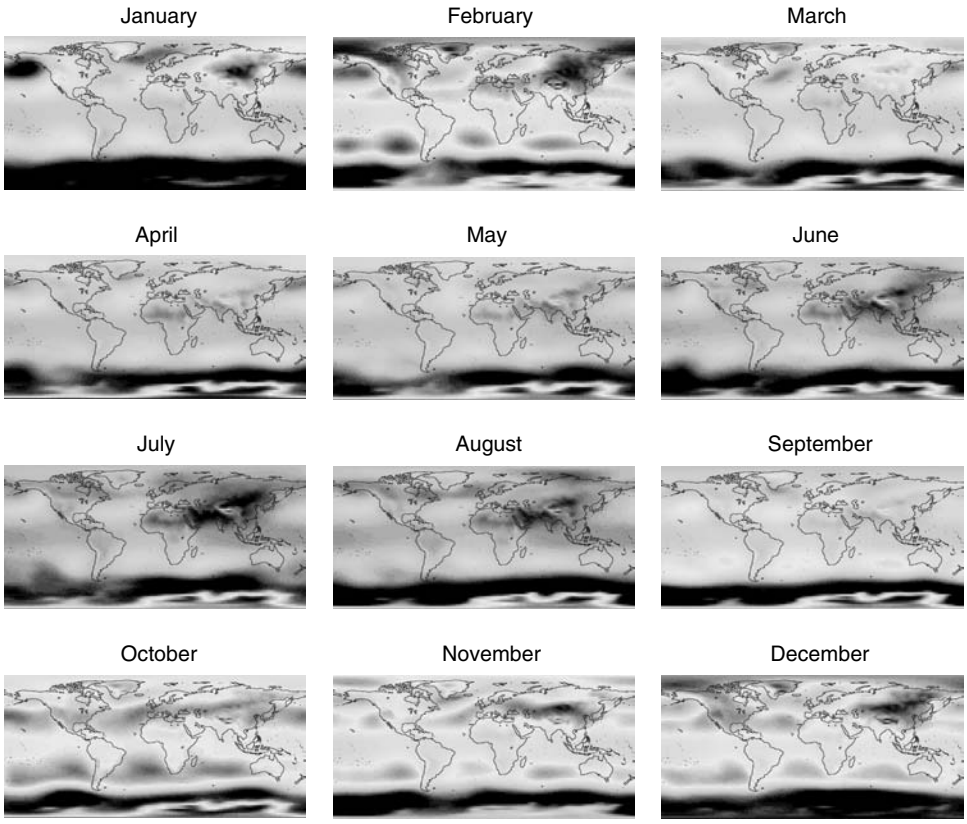


Figure 3.22. Monthly plots of sea level pressure over the 12 months of 1982.

3.3.4 Visualizing Higher-Dimensional Data

This section considers visualization techniques that can display more than the handful of dimensions that can be observed with the techniques just discussed. However, even these techniques are somewhat limited in that they only show some aspects of the data.

Matrices An image can be regarded as a rectangular array of pixels, where each pixel is characterized by its color and brightness. A data matrix is a rectangular array of values. Thus, a data matrix can be visualized as an image by associating each entry of the data matrix with a pixel in the image. The brightness or color of the pixel is determined by the value of the corresponding entry of the matrix.

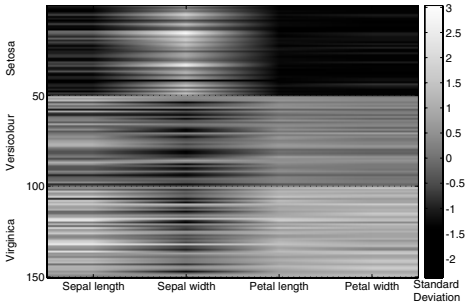


Figure 3.23. Plot of the Iris data matrix where columns have been standardized to have a mean of 0 and standard deviation of 1.

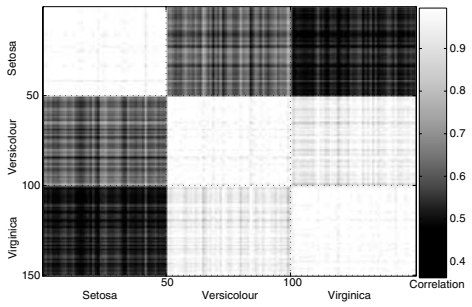


Figure 3.24. Plot of the Iris correlation matrix.

There are some important practical considerations when visualizing a data matrix. If class labels are known, then it is useful to reorder the data matrix so that all objects of a class are together. This makes it easier, for example, to detect if all objects in a class have similar attribute values for some attributes. If different attributes have different ranges, then the attributes are often standardized to have a mean of zero and a standard deviation of 1. This prevents the attribute with the largest magnitude values from visually dominating the plot.

Example 3.19. Figure 3.23 shows the standardized data matrix for the Iris data set. The first 50 rows represent Iris flowers of the species Setosa, the next 50 Versicolour, and the last 50 Virginica. The Setosa flowers have petal width and length well below the average, while the Versicolour flowers have petal width and length around average. The Virginica flowers have petal width and length above average. ■

It can also be useful to look for structure in the plot of a proximity matrix for a set of data objects. Again, it is useful to sort the rows and columns of the similarity matrix (when class labels are known) so that all the objects of a class are together. This allows a visual evaluation of the cohesiveness of each class and its separation from other classes.

Example 3.20. Figure 3.24 shows the correlation matrix for the Iris data set. Again, the rows and columns are organized so that all the flowers of a particular species are together. The flowers in each group are most similar

to each other, but Versicolour and Virginica are more similar to one another than to Setosa. ■

If class labels are not known, various techniques (matrix reordering and seriation) can be used to rearrange the rows and columns of the similarity matrix so that groups of highly similar objects and attributes are together and can be visually identified. Effectively, this is a simple kind of clustering. See Section 8.5.3 for a discussion of how a proximity matrix can be used to investigate the cluster structure of data.

Parallel Coordinates Parallel coordinates have one coordinate axis for each attribute, but the different axes are parallel to one other instead of perpendicular, as is traditional. Furthermore, an object is represented as a line instead of as a point. Specifically, the value of each attribute of an object is mapped to a point on the coordinate axis associated with that attribute, and these points are then connected to form the line that represents the object.

It might be feared that this would yield quite a mess. However, in many cases, objects tend to fall into a small number of groups, where the points in each group have similar values for their attributes. If so, and if the number of data objects is not too large, then the resulting parallel coordinates plot can reveal interesting patterns.

Example 3.21. Figure 3.25 shows a parallel coordinates plot of the four numerical attributes of the Iris data set. The lines representing objects of different classes are distinguished by their shading and the use of three different line styles—solid, dotted, and dashed. The parallel coordinates plot shows that the classes are reasonably well separated for petal width and petal length, but less well separated for sepal length and sepal width. Figure 3.25 is another parallel coordinates plot of the same data, but with a different ordering of the axes. ■

One of the drawbacks of parallel coordinates is that the detection of patterns in such a plot may depend on the order. For instance, if lines cross a lot, the picture can become confusing, and thus, it can be desirable to order the coordinate axes to obtain sequences of axes with less crossover. Compare Figure 3.26, where sepal width (the attribute that is most mixed) is at the left of the figure, to Figure 3.25, where this attribute is in the middle.

Star Coordinates and Chernoff Faces

Another approach to displaying multidimensional data is to encode objects as **glyphs** or **icons**—symbols that impart information non-verbally. More

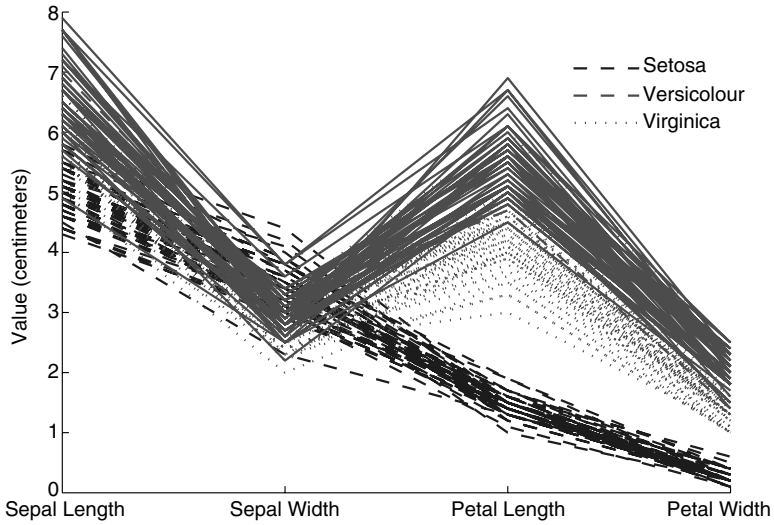


Figure 3.25. A parallel coordinates plot of the four Iris attributes.

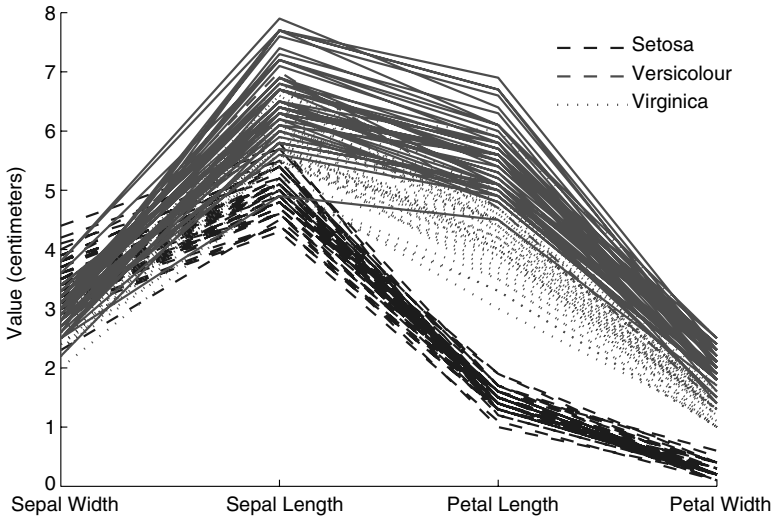


Figure 3.26. A parallel coordinates plot of the four Iris attributes with the attributes reordered to emphasize similarities and dissimilarities of groups.

specifically, each attribute of an object is mapped to a particular feature of a glyph, so that the value of the attribute determines the exact nature of the feature. Thus, at a glance, we can distinguish how two objects differ.

Star coordinates are one example of this approach. This technique uses one axis for each attribute. These axes all radiate from a center point, like the spokes of a wheel, and are evenly spaced. Typically, all the attribute values are mapped to the range [0,1].

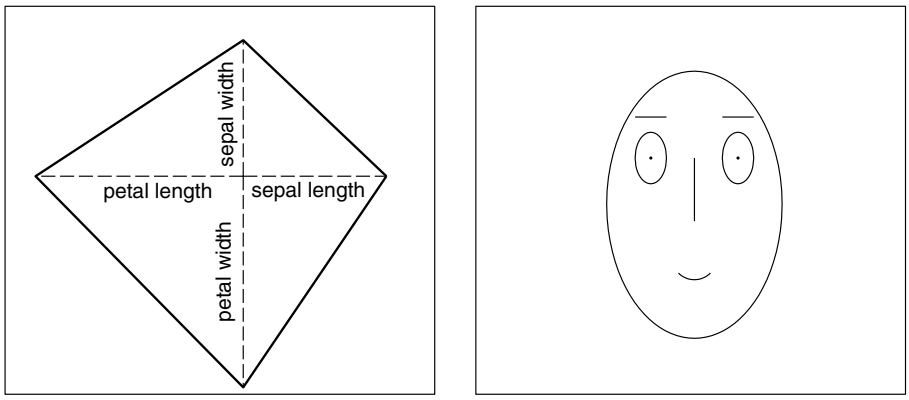
An object is mapped onto this star-shaped set of axes using the following process: Each attribute value of the object is converted to a fraction that represents its distance between the minimum and maximum values of the attribute. This fraction is mapped to a point on the axis corresponding to this attribute. Each point is connected with a line segment to the point on the axis preceding or following its own axis; this forms a polygon. The size and shape of this polygon gives a visual description of the attribute values of the object. For ease of interpretation, a separate set of axes is used for each object. In other words, each object is mapped to a polygon. An example of a star coordinates plot of flower 150 is given in Figure 3.27(a).

It is also possible to map the values of features to those of more familiar objects, such as faces. This technique is named **Chernoff faces** for its creator, Herman Chernoff. In this technique, each attribute is associated with a specific feature of a face, and the attribute value is used to determine the way that the facial feature is expressed. Thus, the shape of the face may become more elongated as the value of the corresponding data feature increases. An example of a Chernoff face for flower 150 is given in Figure 3.27(b).

The program that we used to make this face mapped the features to the four features listed below. Other features of the face, such as width between the eyes and length of the mouth, are given default values.

Data Feature	Facial Feature
sepal length	size of face
sepal width	forehead/jaw relative arc length
petal length	shape of forehead
petal width	shape of jaw

Example 3.22. A more extensive illustration of these two approaches to viewing multidimensional data is provided by Figures 3.28 and 3.29, which shows the star and face plots, respectively, of 15 flowers from the Iris data set. The first 5 flowers are of species Setosa, the second 5 are Versicolour, and the last 5 are Virginica. ■



(a) Star graph of Iris 150.

(b) Chernoff face of Iris 150.

Figure 3.27. Star coordinates graph and Chernoff face of the 150^{th} flower of the Iris data set.

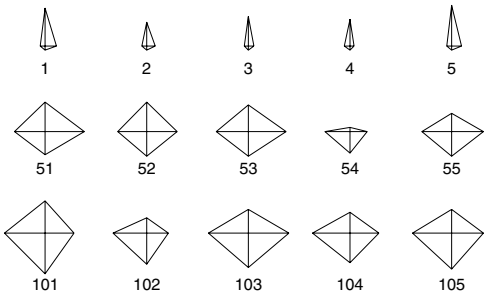


Figure 3.28. Plot of 15 Iris flowers using star coordinates.

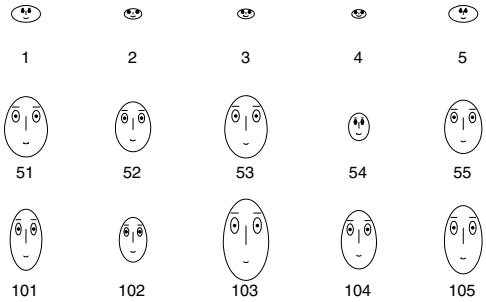


Figure 3.29. A plot of 15 Iris flowers using Chernoff faces.

Despite the visual appeal of these sorts of diagrams, they do not scale well, and thus, they are of limited use for many data mining problems. Nonetheless, they may still be of use as a means to quickly compare small sets of objects that have been selected by other techniques.

3.3.5 Do's and Don'ts

To conclude this section on visualization, we provide a short list of visualization do's and don'ts. While these guidelines incorporate a lot of visualization wisdom, they should not be followed blindly. As always, guidelines are no substitute for thoughtful consideration of the problem at hand.

ACCENT Principles The following are the *ACCENT* principles for effective graphical display put forth by D. A. Burn (as adapted by Michael Friendly):

Apprehension Ability to correctly perceive relations among variables. Does the graph maximize apprehension of the relations among variables?

Clarity Ability to visually distinguish all the elements of a graph. Are the most important elements or relations visually most prominent?

Consistency Ability to interpret a graph based on similarity to previous graphs. Are the elements, symbol shapes, and colors consistent with their use in previous graphs?

Efficiency Ability to portray a possibly complex relation in as simple a way as possible. Are the elements of the graph economically used? Is the graph easy to interpret?

Necessity The need for the graph, and the graphical elements. Is the graph a more useful way to represent the data than alternatives (table, text)? Are all the graph elements necessary to convey the relations?

Truthfulness Ability to determine the true value represented by any graphical element by its magnitude relative to the implicit or explicit scale. Are the graph elements accurately positioned and scaled?

Tufte's Guidelines Edward R. Tufte has also enumerated the following principles for graphical excellence:

- Graphical excellence is the well-designed presentation of interesting data—a matter of *substance*, of *statistics*, and of *design*.
- Graphical excellence consists of complex ideas communicated with clarity, precision, and efficiency.
- Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.
- Graphical excellence is nearly always multivariate.
- And graphical excellence requires telling the truth about the data.

3.4 OLAP and Multidimensional Data Analysis

In this section, we investigate the techniques and insights that come from viewing data sets as multidimensional arrays. A number of database systems support such a viewpoint, most notably, On-Line Analytical Processing (OLAP) systems. Indeed, some of the terminology and capabilities of OLAP systems have made their way into spreadsheet programs that are used by millions of people. OLAP systems also have a strong focus on the interactive analysis of data and typically provide extensive capabilities for visualizing the data and generating summary statistics. For these reasons, our approach to multidimensional data analysis will be based on the terminology and concepts common to OLAP systems.

3.4.1 Representing Iris Data as a Multidimensional Array

Most data sets can be represented as a table, where each row is an object and each column is an attribute. In many cases, it is also possible to view the data as a multidimensional array. We illustrate this approach by representing the Iris data set as a multidimensional array.

Table 3.7 was created by discretizing the petal length and petal width attributes to have values of *low*, *medium*, and *high* and then counting the number of flowers from the Iris data set that have particular combinations of petal width, petal length, and species type. (For petal width, the categories *low*, *medium*, and *high* correspond to the intervals [0, 0.75), [0.75, 1.75), [1.75, ∞), respectively. For petal length, the categories *low*, *medium*, and *high* correspond to the intervals [0, 2.5), [2.5, 5), [5, ∞), respectively.)

Table 3.7. Number of flowers having a particular combination of petal width, petal length, and species type.

Petal Length	Petal Width	Species Type	Count
low	low	Setosa	46
low	medium	Setosa	2
medium	low	Setosa	2
medium	medium	Versicolour	43
medium	high	Versicolour	3
medium	high	Virginica	3
high	medium	Versicolour	2
high	medium	Virginica	3
high	high	Versicolour	2
high	high	Virginica	44

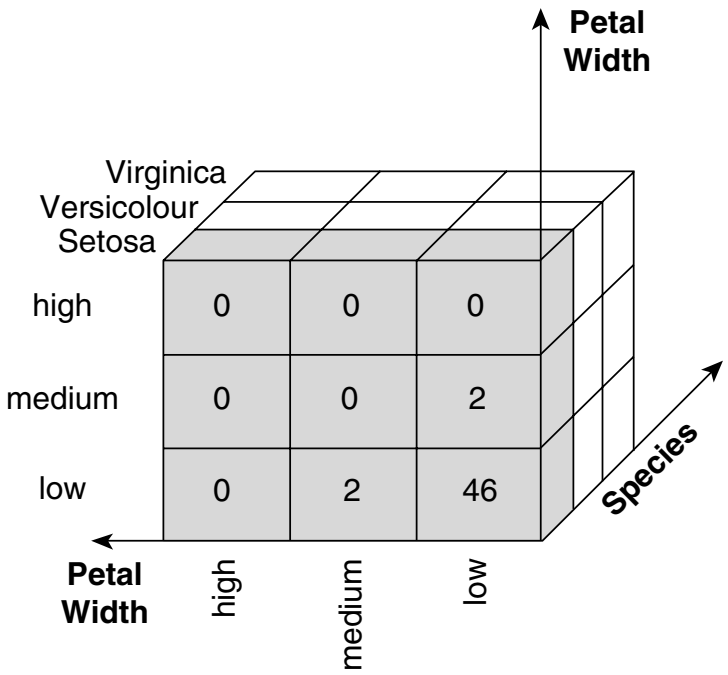


Figure 3.30. A multidimensional data representation for the Iris data set.

Table 3.8. Cross-tabulation of flowers according to petal length and width for flowers of the Setosa species.

		Width		
		low	medium	high
Length	low	46	2	0
	medium	2	0	0
	high	0	0	0

Table 3.9. Cross-tabulation of flowers according to petal length and width for flowers of the Versicolour species.

		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	43	3
	high	0	2	2

Table 3.10. Cross-tabulation of flowers according to petal length and width for flowers of the Virginica species.

		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	0	3
	high	0	3	44

Empty combinations—those combinations that do not correspond to at least one flower—are not shown.

The data can be organized as a multidimensional array with three dimensions corresponding to petal width, petal length, and species type, as illustrated in Figure 3.30. For clarity, slices of this array are shown as a set of three two-dimensional tables, one for each species—see Tables 3.8, 3.9, and 3.10. The information contained in both Table 3.7 and Figure 3.30 is the same. However, in the multidimensional representation shown in Figure 3.30 (and Tables 3.8, 3.9, and 3.10), the values of the attributes—petal width, petal length, and species type—are array indices.

What is important are the insights can be gained by looking at data from a multidimensional viewpoint. Tables 3.8, 3.9, and 3.10 show that each species of Iris is characterized by a different combination of values of petal length and width. Setosa flowers have low width and length, Versicolour flowers have medium width and length, and Virginica flowers have high width and length.

3.4.2 Multidimensional Data: The General Case

The previous section gave a specific example of using a multidimensional approach to represent and analyze a familiar data set. Here we describe the general approach in more detail.

The starting point is usually a tabular representation of the data, such as that of Table 3.7, which is called a **fact table**. Two steps are necessary in order to represent data as a multidimensional array: identification of the dimensions and identification of an attribute that is the focus of the analysis. The dimensions are categorical attributes or, as in the previous example, continuous attributes that have been converted to categorical attributes. The values of an attribute serve as indices into the array for the dimension corresponding to the attribute, and the number of attribute values is the size of that dimension. In the previous example, each attribute had three possible values, and thus, each dimension was of size three and could be indexed by three values. This produced a $3 \times 3 \times 3$ multidimensional array.

Each combination of attribute values (one value for each different attribute) defines a cell of the multidimensional array. To illustrate using the previous example, if petal length = *low*, petal width = *medium*, and species = Setosa, a specific cell containing the value 2 is identified. That is, there are only two flowers in the data set that have the specified attribute values. Notice that each row (object) of the data set in Table 3.7 corresponds to a cell in the multidimensional array.

The contents of each cell represents the value of a **target quantity** (target variable or attribute) that we are interested in analyzing. In the Iris example, the target quantity is the *number of flowers* whose petal width and length fall within certain limits. The target attribute is quantitative because a key goal of multidimensional data analysis is to look aggregate quantities, such as totals or averages.

The following summarizes the procedure for creating a multidimensional data representation from a data set represented in tabular form. First, identify the categorical attributes to be used as the dimensions and a quantitative attribute to be used as the target of the analysis. Each row (object) in the table is mapped to a cell of the multidimensional array. The indices of the cell are specified by the values of the attributes that were selected as dimensions, while the value of the cell is the value of the target attribute. Cells not defined by the data are assumed to have a value of 0.

Example 3.23. To further illustrate the ideas just discussed, we present a more traditional example involving the sale of products.The fact table for this example is given by Table 3.11. The dimensions of the multidimensional representation are the *product ID*, *location*, and *date* attributes, while the target attribute is the *revenue*. Figure 3.31 shows the multidimensional representation of this data set. This larger and more complicated data set will be used to illustrate additional concepts of multidimensional data analysis. ■

3.4.3 Analyzing Multidimensional Data

In this section, we describe different multidimensional analysis techniques. In particular, we discuss the creation of data cubes, and related operations, such as slicing, dicing, dimensionality reduction, roll-up, and drill down.

Data Cubes: Computing Aggregate Quantities

A key motivation for taking a multidimensional viewpoint of data is the importance of aggregating data in various ways. In the sales example, we might wish to find the total sales revenue for a specific year and a specific product. Or we might wish to see the yearly sales revenue for each location across all products. Computing aggregate totals involves fixing specific values for some of the attributes that are being used as dimensions and then summing over all possible values for the attributes that make up the remaining dimensions. There are other types of aggregate quantities that are also of interest, but for simplicity, this discussion will use totals (sums).

Table 3.12 shows the result of summing over all locations for various combinations of date and product. For simplicity, assume that all the dates are within one year. If there are 365 days in a year and 1000 products, then Table 3.12 has 365,000 entries (totals), one for each product-data pair. We could also specify the store location and date and sum over products, or specify the location and product and sum over all dates.

Table 3.13 shows the **marginal totals** of Table 3.12. These totals are the result of further summing over either dates or products. In Table 3.13, the total sales revenue due to product 1, which is obtained by summing across row 1 (over all dates), is \$370,000. The total sales revenue on January 1, 2004, which is obtained by summing down column 1 (over all products), is \$527,362. The total sales revenue, which is obtained by summing over all rows and columns (all times and products) is \$227,352,127. All of these totals are for all locations because the entries of Table 3.13 include all locations.

A key point of this example is that there are a number of different totals (aggregates) that can be computed for a multidimensional array, depending on how many attributes we sum over. Assume that there are n dimensions and that the i^{th} dimension (attribute) has s_i possible values. There are n different ways to sum only over a single attribute. If we sum over dimension j, then we obtain $s_1 * \cdots * s_{j-1} * s_{j+1} * \cdots * s_n$ totals, one for each possible combination of attribute values of the $n-1$ other attributes (dimensions). The totals that result from summing over one attribute form a multidimensional array of $n-1$ dimensions and there are n such arrays of totals. In the sales example, there

Table 3.11. Sales revenue of products (in dollars) for various locations and times.

Product ID	Location	Date	Revenue
⋮	⋮	⋮	⋮
1	Minneapolis	Oct. 18, 2004	$250
1	Chicago	Oct. 18, 2004	$79
⋮	⋮	⋮	
1	Paris	Oct. 18, 2004	301
⋮	⋮	⋮	⋮
27	Minneapolis	Oct. 18, 2004	$2,321
27	Chicago	Oct. 18, 2004	$3,278
⋮	⋮	⋮	
27	Paris	Oct. 18, 2004	$1,325
⋮	⋮	⋮	⋮

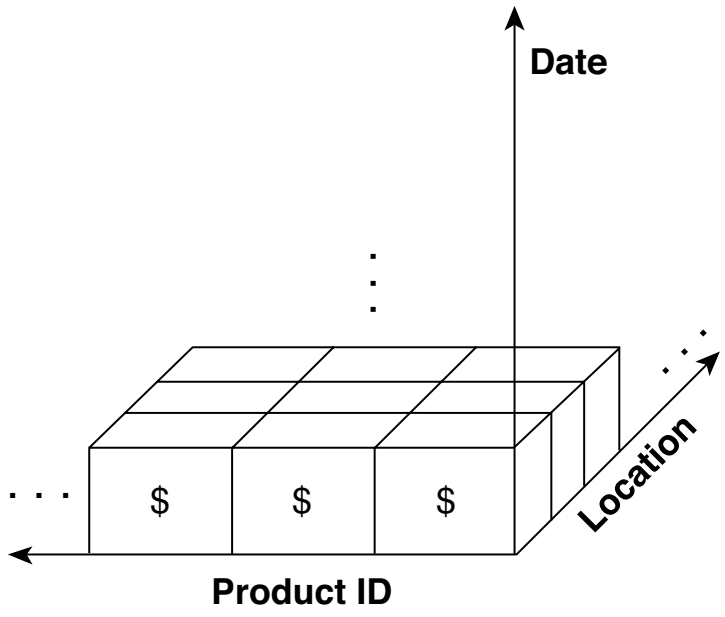


Figure 3.31. Multidimensional data representation for sales data.

Table 3.12. Totals that result from summing over all locations for a fixed time and product.

		date			
		Jan 1, 2004	Jan 2, 2004	...	Dec 31, 2004
product ID	1	$1,001	$987	...	$891
	⋮	⋮			⋮
	27	$10,265	$10,225	...	$9,325
	⋮	⋮			⋮

Table 3.13. Table 3.12 with marginal totals.

		date				
		Jan 1, 2004	Jan 2, 2004	...	Dec 31, 2004	total
product ID	1	$1,001	$987	...	$891	$370,000
	⋮	⋮			⋮	⋮
	27	$10,265	$10,225	...	$9,325	$3,800,020
	⋮	⋮			⋮	⋮
	total	$527,362	$532,953	...	$631,221	$227,352,127

are three sets of totals that result from summing over only one dimension and each set of totals can be displayed as a two-dimensional table.

If we sum over two dimensions (perhaps starting with one of the arrays of totals obtained by summing over one dimension), then we will obtain a multidimensional array of totals with $n - 2$ dimensions. There will be $\binom{n}{2}$ distinct arrays of such totals. For the sales examples, there will be $\binom{3}{2} = 3$ arrays of totals that result from summing over location and product, location and time, or product and time. In general, summing over k dimensions yields $\binom{n}{k}$ arrays of totals, each with dimension $n - k$.

A multidimensional representation of the data, together with all possible totals (aggregates), is known as a **data cube**. Despite the name, the size of each dimension—the number of attribute values—does not need to be equal. Also, a data cube may have either more or fewer than three dimensions. More importantly, a data cube is a generalization of what is known in statistical terminology as a **cross-tabulation**. If marginal totals were added, Tables 3.8, 3.9, or 3.10 would be typical examples of cross tabulations.

Dimensionality Reduction and Pivoting

The aggregation described in the last section can be viewed as a form of **dimensionality reduction**. Specifically, the j^{th} dimension is eliminated by summing over it. Conceptually, this collapses each "column" of cells in the j^{th} dimension into a single cell. For both the sales and Iris examples, aggregating over one dimension reduces the dimensionality of the data from 3 to 2. If s_j is the number of possible values of the j^{th} dimension, the number of cells is reduced by a factor of s_j. Exercise 17 on page 143 asks the reader to explore the difference between this type of dimensionality reduction and that of PCA.

Pivoting refers to aggregating over all dimensions except two. The result is a two-dimensional cross tabulation with the two specified dimensions as the only remaining dimensions. Table 3.13 is an example of pivoting on date and product.

Slicing and Dicing

These two colorful names refer to rather straightforward operations. **Slicing** is selecting a group of cells from the entire multidimensional array by specifying a specific value for one or more dimensions. Tables 3.8, 3.9, and 3.10 are three slices from the Iris set that were obtained by specifying three separate values for the species dimension. **Dicing** involves selecting a subset of cells by specifying a range of attribute values. This is equivalent to defining a subarray from the complete array. In practice, both operations can also be accompanied by aggregation over some dimensions.

Roll-Up and Drill-Down

In Chapter 2, attribute values were regarded as being "atomic" in some sense. However, this is not always the case. In particular, each date has a number of properties associated with it such as the year, month, and week. The data can also be identified as belonging to a particular business quarter, or if the application relates to education, a school quarter or semester. A location also has various properties: continent, country, state (province, etc.), and city. Products can also be divided into various categories, such as clothing, electronics, and furniture.

Often these categories can be organized as a hierarchical tree or lattice. For instance, years consist of months or weeks, both of which consist of days. Locations can be divided into nations, which contain states (or other units of local government), which in turn contain cities. Likewise, any category

of products can be further subdivided. For example, the product category, furniture, can be subdivided into the subcategories, chairs, tables, sofas, etc.

This hierarchical structure gives rise to the roll-up and drill-down operations. To illustrate, starting with the original sales data, which is a multidimensional array with entries for each date, we can aggregate (**roll up**) the sales across all the dates in a month. Conversely, given a representation of the data where the time dimension is broken into months, we might want to split the monthly sales totals (**drill down**) into daily sales totals. Of course, this requires that the underlying sales data be available at a daily granularity.

Thus, roll-up and drill-down operations are related to aggregation. Notice, however, that they differ from the aggregation operations discussed until now in that they aggregate cells within a dimension, not across the entire dimension.

3.4.4 Final Comments on Multidimensional Data Analysis

Multidimensional data analysis, in the sense implied by OLAP and related systems, consists of viewing the data as a multidimensional array and aggregating data in order to better analyze the structure of the data. For the Iris data, the differences in petal width and length are clearly shown by such an analysis. The analysis of business data, such as sales data, can also reveal many interesting patterns, such as profitable (or unprofitable) stores or products.

As mentioned, there are various types of database systems that support the analysis of multidimensional data. Some of these systems are based on relational databases and are known as ROLAP systems. More specialized database systems that specifically employ a multidimensional data representation as their fundamental data model have also been designed. Such systems are known as MOLAP systems. In addition to these types of systems, statistical databases (SDBs) have been developed to store and analyze various types of statistical data, e.g., census and public health data, that are collected by governments or other large organizations. References to OLAP and SDBs are provided in the bibliographic notes.

3.5 Bibliographic Notes

Summary statistics are discussed in detail in most introductory statistics books, such as [92]. References for exploratory data analysis are the classic text by Tukey [104] and the book by Velleman and Hoaglin [105].

The basic visualization techniques are readily available, being an integral part of most spreadsheets (Microsoft EXCEL [95]), statistics programs (SAS

[99], SPSS [102], R [96], and S-PLUS [98]), and mathematics software (MATLAB [94] and Mathematica [93]). Most of the graphics in this chapter were generated using MATLAB. The statistics package R is freely available as an open source software package from the R project.

The literature on visualization is extensive, covering many fields and many decades. One of the classics of the field is the book by Tufte [103]. The book by Spence [101], which strongly influenced the visualization portion of this chapter, is a useful reference for information visualization—both principles and techniques. This book also provides a thorough discussion of many dynamic visualization techniques that were not covered in this chapter. Two other books on visualization that may also be of interest are those by Card et al. [87] and Fayyad et al. [89].

Finally, there is a great deal of information available about data visualization on the World Wide Web. Since Web sites come and go frequently, the best strategy is a search using "information visualization," "data visualization," or "statistical graphics." However, we do want to single out for attention "The Gallery of Data Visualization," by Friendly [90]. The ACCENT Principles for effective graphical display as stated in this chapter can be found there, or as originally presented in the article by Burn [86].

There are a variety of graphical techniques that can be used to explore whether the distribution of the data is Gaussian or some other specified distribution. Also, there are plots that display whether the observed values are statistically significant in some sense. We have not covered any of these techniques here and refer the reader to the previously mentioned statistical and mathematical packages.

Multidimensional analysis has been around in a variety of forms for some time. One of the original papers was a white paper by Codd [88], the father of relational databases. The data cube was introduced by Gray et al. [91], who described various operations for creating and manipulating data cubes within a relational database framework. A comparison of statistical databases and OLAP is given by Shoshani [100]. Specific information on OLAP can be found in documentation from database vendors and many popular books. Many database textbooks also have general discussions of OLAP, often in the context of data warehousing. For example, see the text by Ramakrishnan and Gehrke [97].

Bibliography

[86] D. A. Burn. Designing Effective Statistical Graphs. In C. R. Rao, editor, *Handbook of Statistics 9*. Elsevier/North-Holland, Amsterdam, The Netherlands, September 1993.

[87] S. K. Card, J. D. MacKinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA, January 1999.

[88] E. F. Codd, S. B. Codd, and C. T. Smalley. Providing OLAP (On-line Analytical Processing) to User- Analysts: An IT Mandate. White Paper, E.F. Codd and Associates, 1993.

[89] U. M. Fayyad, G. G. Grinstein, and A. Wierse, editors. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, San Francisco, CA, September 2001.

[90] M. Friendly. Gallery of Data Visualization. http://www.math.yorku.ca/SCS/Gallery/, 2005.

[91] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Journal Data Mining and Knowledge Discovery*, 1(1): 29–53, 1997.

[92] B. W. Lindgren. *Statistical Theory*. CRC Press, January 1993.

[93] Mathematica 5.1. Wolfram Research, Inc. http://www.wolfram.com/, 2005.

[94] MATLAB 7.0. The MathWorks, Inc. http://www.mathworks.com, 2005.

[95] Microsoft Excel 2003. Microsoft, Inc. http://www.microsoft.com/, 2003.

[96] R: A language and environment for statistical computing and graphics. The R Project for Statistical Computing. http://www.r-project.org/, 2005.

[97] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill, 3rd edition, August 2002.

[98] S-PLUS. Insightful Corporation. http://www.insightful.com, 2005.

[99] SAS: Statistical Analysis System. SAS Institute Inc. http://www.sas.com/, 2005.

[100] A. Shoshani. OLAP and statistical databases: similarities and differences. In *Proc. of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, pages 185–196. ACM Press, 1997.

[101] R. Spence. *Information Visualization*. ACM Press, New York, December 2000.

[102] SPSS: Statistical Package for the Social Sciences. SPSS, Inc. http://www.spss.com/, 2005.

[103] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, March 1986.

[104] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977.

[105] P. Velleman and D. Hoaglin. *The ABC's of EDA: Applications, Basics, and Computing of Exploratory Data Analysis*. Duxbury, 1981.

3.6 Exercises

1. Obtain one of the data sets available at the UCI Machine Learning Repository and apply as many of the different visualization techniques described in the chapter as possible. The bibliographic notes and book Web site provide pointers to visualization software.

2. Identify at least two advantages and two disadvantages of using color to visually represent information.

3. What are the arrangement issues that arise with respect to three-dimensional plots?

4. Discuss the advantages and disadvantages of using sampling to reduce the number of data objects that need to be displayed. Would simple random sampling (without replacement) be a good approach to sampling? Why or why not?

5. Describe how you would create visualizations to display information that describes the following types of systems.

   (a) Computer networks. Be sure to include both the static aspects of the network, such as connectivity, and the dynamic aspects, such as traffic.

   (b) The distribution of specific plant and animal species around the world for a specific moment in time.

   (c) The use of computer resources, such as processor time, main memory, and disk, for a set of benchmark database programs.

   (d) The change in occupation of workers in a particular country over the last thirty years. Assume that you have yearly information about each person that also includes gender and level of education.

   Be sure to address the following issues:

   - **Representation.** How will you map objects, attributes, and relationships to visual elements?
   - **Arrangement.** Are there any special considerations that need to be taken into account with respect to how visual elements are displayed? Specific examples might be the choice of viewpoint, the use of transparency, or the separation of certain groups of objects.
   - **Selection.** How will you handle a large number of attributes and data objects?

6. Describe one advantage and one disadvantage of a stem and leaf plot with respect to a standard histogram.

7. How might you address the problem that a histogram depends on the number and location of the bins?

8. Describe how a box plot can give information about whether the value of an attribute is symmetrically distributed. What can you say about the symmetry of the distributions of the attributes shown in Figure 3.11?

9. Compare sepal length, sepal width, petal length, and petal width, using Figure 3.12.

10. Comment on the use of a box plot to explore a data set with four attributes: age, weight, height, and income.

11. Give a possible explanation as to why most of the values of petal length and width fall in the buckets along the diagonal in Figure 3.9.

12. Use Figures 3.14 and 3.15 to identify a characteristic shared by the petal width and petal length attributes.

13. Simple line plots, such as that displayed in Figure 2.12 on page 56, which shows two time series, can be used to effectively display high-dimensional data. For example, in Figure 2.12 it is easy to tell that the frequencies of the two time series are different. What characteristic of time series allows the effective visualization of high-dimensional data?

14. Describe the types of situations that produce sparse or dense data cubes. Illustrate with examples other than those used in the book.

15. How might you extend the notion of multidimensional data analysis so that the target variable is a qualitative variable? In other words, what sorts of summary statistics or data visualizations would be of interest?

16. Construct a data cube from Table 3.14. Is this a dense or sparse data cube? If it is sparse, identify the cells that empty.

Table 3.14. Fact table for Exercise 16.

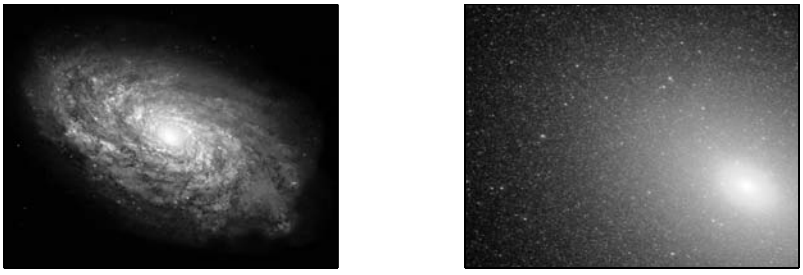
Product ID	Location ID	Number Sold
1	1	10
1	3	6
2	1	5
2	2	22

17. Discuss the differences between dimensionality reduction based on aggregation and dimensionality reduction based on techniques such as PCA and SVD.

4

Classification: Basic Concepts, Decision Trees, and Model Evaluation

Classification, which is the task of assigning objects to one of several predefined categories, is a pervasive problem that encompasses many diverse applications. Examples include detecting spam email messages based upon the message header and content, categorizing cells as malignant or benign based upon the results of MRI scans, and classifying galaxies based upon their shapes (see Figure 4.1).



(a) A spiral galaxy. (b) An elliptical galaxy.

Figure 4.1. Classification of galaxies. The images are from the NASA website.

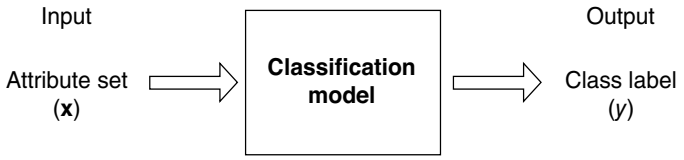


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

This chapter introduces the basic concepts of classification, describes some of the key issues such as model overfitting, and presents methods for evaluating and comparing the performance of a classification technique. While it focuses mainly on a technique known as decision tree induction, most of the discussion in this chapter is also applicable to other classification techniques, many of which are covered in Chapter 5.

4.1 Preliminaries

The input data for a classification task is a collection of records. Each record, also known as an instance or example, is characterized by a tuple $(\mathbf{x}, y)$, where $\mathbf{x}$ is the attribute set and y is a special attribute, designated as the class label (also known as category or target attribute). Table 4.1 shows a sample data set used for classifying vertebrates into one of the following categories: mammal, bird, fish, reptile, or amphibian. The attribute set includes properties of a vertebrate such as its body temperature, skin cover, method of reproduction, ability to fly, and ability to live in water. Although the attributes presented in Table 4.1 are mostly discrete, the attribute set can also contain continuous features. The class label, on the other hand, must be a discrete attribute. This is a key characteristic that distinguishes classification from **regression**, a predictive modeling task in which y is a continuous attribute. Regression techniques are covered in Appendix D.

Definition 4.1 (Classification). Classification is the task of learning a **target function** f that maps each attribute set $\mathbf{x}$ to one of the predefined class labels y.

The target function is also known informally as a **classification model**. A classification model is useful for the following purposes.

Descriptive Modeling A classification model can serve as an explanatory tool to distinguish between objects of different classes. For example, it would be useful—for both biologists and others—to have a descriptive model that

Table 4.1. The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

summarizes the data shown in Table 4.1 and explains what features define a vertebrate as a mammal, reptile, bird, fish, or amphibian.

Predictive Modeling A classification model can also be used to predict the class label of unknown records. As shown in Figure 4.2, a classification model can be treated as a black box that automatically assigns a class label when presented with the attribute set of an unknown record. Suppose we are given the following characteristics of a creature known as a gila monster:

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

We can use a classification model built from the data set shown in Table 4.1 to determine the class to which the creature belongs.

Classification techniques are most suited for predicting or describing data sets with binary or nominal categories. They are less effective for ordinal categories (e.g., to classify a person as a member of high-, medium-, or low-income group) because they do not consider the implicit order among the categories. Other forms of relationships, such as the subclass–superclass relationships among categories (e.g., humans and apes are primates, which in

turn, is a subclass of mammals) are also ignored. The remainder of this chapter focuses only on binary or nominal class labels.

4.2 General Approach to Solving a Classification Problem

A classification technique (or classifier) is a systematic approach to building classification models from an input data set. Examples include decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naïve Bayes classifiers. Each technique employs a **learning algorithm** to identify a model that best fits the relationship between the attribute set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.

Figure 4.3 shows a general approach for solving classification problems. First, a **training set** consisting of records whose class labels are known must

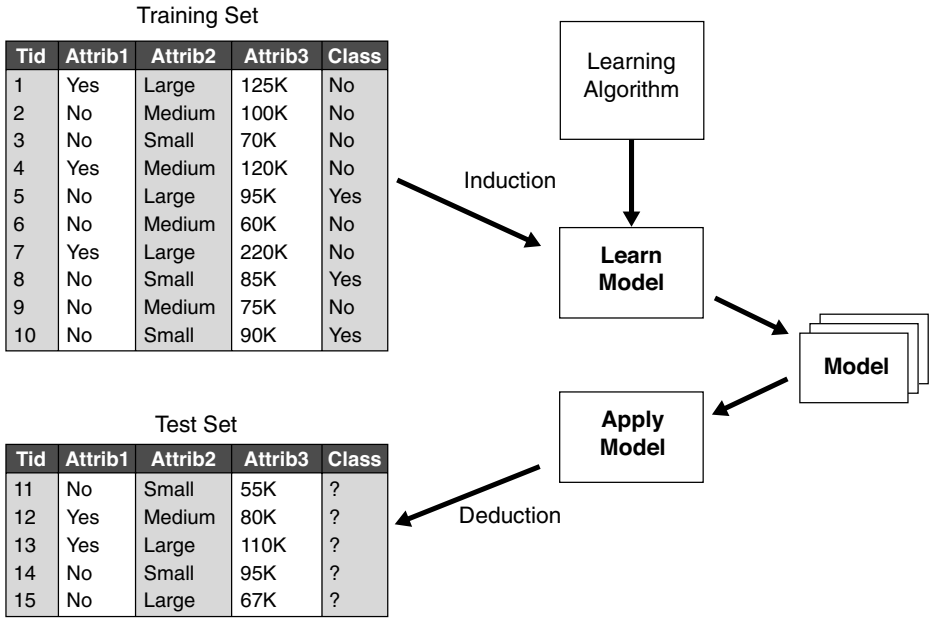


Figure 4.3. General approach for building a classification model.

Table 4.2. Confusion matrix for a 2-class problem.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

be provided. The training set is used to build a classification model, which is subsequently applied to the **test set**, which consists of records with unknown class labels.

Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a **confusion matrix**. Table 4.2 depicts the confusion matrix for a binary classification problem. Each entry f_{ij} in this table denotes the number of records from class i predicted to be of class j. For instance, f_{01} is the number of records from class 0 incorrectly predicted as class 1. Based on the entries in the confusion matrix, the total number of correct predictions made by the model is $(f_{11} + f_{00})$ and the total number of incorrect predictions is $(f_{10} + f_{01})$.

Although a confusion matrix provides the information needed to determine how well a classification model performs, summarizing this information with a single number would make it more convenient to compare the performance of different models. This can be done using a **performance metric** such as **accuracy**, which is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.1)$$

Equivalently, the performance of a model can be expressed in terms of its **error rate**, which is given by the following equation:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.2)$$

Most classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set. We will revisit the topic of model evaluation in Section 4.5.

4.3 Decision Tree Induction

This section introduces a **decision tree** classifier, which is a simple yet widely used classification technique.

4.3.1 How a Decision Tree Works

To illustrate how classification with a decision tree works, consider a simpler version of the vertebrate classification problem described in the previous section. Instead of classifying the vertebrates into five distinct groups of species, we assign them to two categories: mammals and non-mammals.

Suppose a new species is discovered by scientists. How can we tell whether it is a mammal or a non-mammal? One approach is to pose a series of questions about the characteristics of the species. The first question we may ask is whether the species is cold- or warm-blooded. If it is cold-blooded, then it is definitely not a mammal. Otherwise, it is either a bird or a mammal. In the latter case, we need to ask a follow-up question: Do the females of the species give birth to their young? Those that do give birth are definitely mammals, while those that do not are likely to be non-mammals (with the exception of egg-laying mammals such as the platypus and spiny anteater).

The previous example illustrates how we can solve a classification problem by asking a series of carefully crafted questions about the attributes of the test record. Each time we receive an answer, a follow-up question is asked until we reach a conclusion about the class label of the record. The series of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges. Figure 4.4 shows the decision tree for the mammal classification problem. The tree has three types of nodes:

- A **root node** that has no incoming edges and zero or more outgoing edges.
- **Internal nodes**, each of which has exactly one incoming edge and two or more outgoing edges.
- **Leaf** or **terminal** nodes, each of which has exactly one incoming edge and no outgoing edges.

In a decision tree, each leaf node is assigned a class label. The **non-terminal** nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. For example, the root node shown in Figure 4.4 uses the attribute `Body`