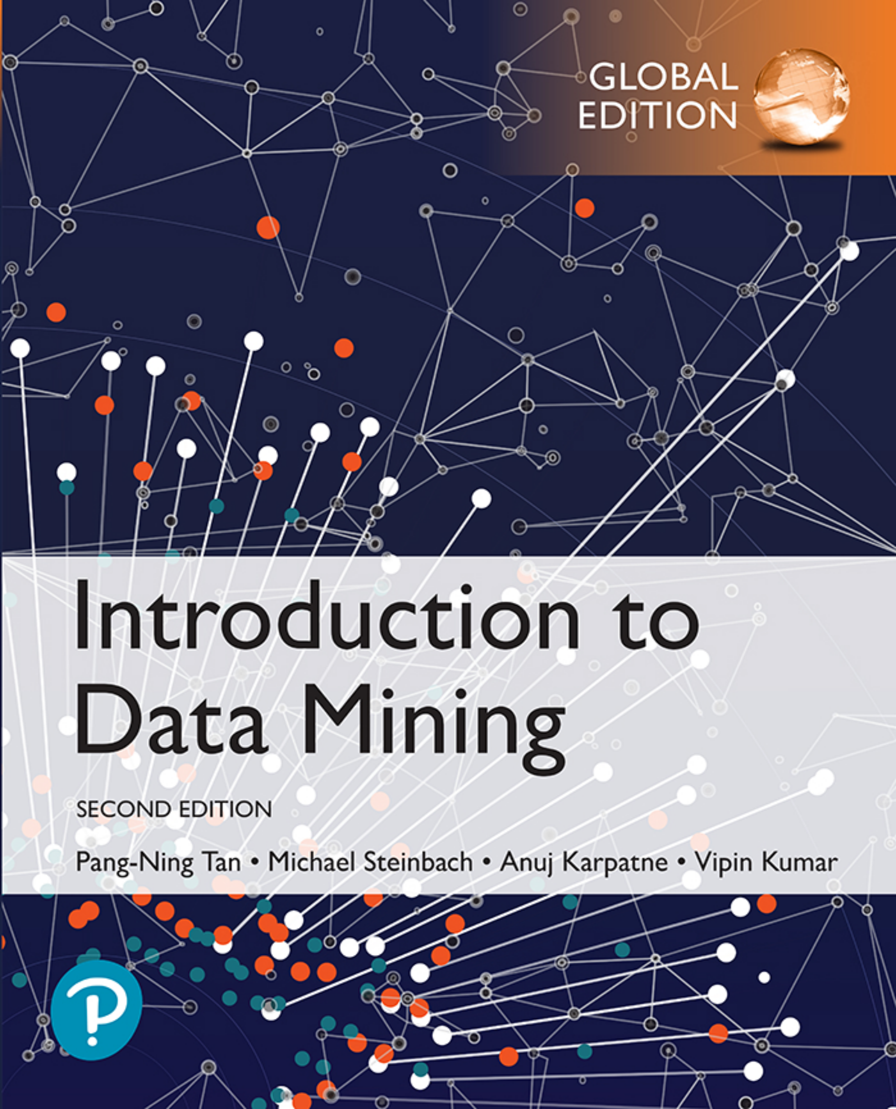


GLOBAL EDITION

Introduction to Data Mining

SECOND EDITION

Pang-Ning Tan • Michael Steinbach • Anuj Karpatne • Vipin Kumar

INTRODUCTION TO DATA MINING

INTRODUCTION TO DATA MINING

SECOND EDITION

GLOBAL EDITION

PANG-NING TAN
Michigan State University

MICHAEL STEINBACH
University of Minnesota

ANUJ KARPATNE
University of Minnesota

VIPIN KUMAR
University of Minnesota

Pearson

330 Hudson Street, NY NY 10013

Director, Portfolio Management: Engineering, Computer Science & Global Editions: Julian Partridge
Specialist, Higher Ed Portfolio Management: Matt Goldstein
Portfolio Management Assistant: Meghan Jacoby
Acquisitions Editor, Global Edition: Sourabh Maheshwari
Managing Content Producer: Scott Disanno
Content Producer: Carole Snyder
Senior Project Editor, Global Edition: K.K. Neelakantan
Web Developer: Steve Wright
Manager, Media Production, Global Edition: Vikram Kumar
Rights and Permissions Manager: Ben Ferrini
Manufacturing Buyer, Higher Ed, Lake Side Communications Inc (LSC): Maura Zaldivar-Garcia
Senior Manufacturing Controller, Global Edition: Caterina Pellegrino
Inventory Manager: Ann Lam
Product Marketing Manager: Yvonne Vannatta
Field Marketing Manager: Demetrius Hall
Marketing Assistant: Jon Bryant
Cover Designer: Lumina Datamatics
Full-Service Project Management: Ramya Radhakrishnan, Integra Software Services

Pearson Education Limited
KAO Two
KAO Park
Harlow
CM17 9NA
United Kingdom

and Associated Companies throughout the world

Visit us on the World Wide Web at: www.pearsonglobaleditions.com

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

ISBN 10: 0-273-76922-7
ISBN 13: 978-0-273-76922-4
eBook ISBN 13: 978-0-273-77532-4

eBook formatted by Integra Software Services.

To our families ...

Preface to the Second Edition

Since the first edition, roughly 12 years ago, much has changed in the field of data analysis. The volume and variety of data being collected continues to increase, as has the rate (velocity) at which it is being collected and used to make decisions. Indeed, the term Big Data has been used to refer to the massive and diverse data sets now available. In addition, the term data science has been coined to describe an emerging area that applies tools and techniques from various fields, such as data mining, machine learning, statistics, and many others, to extract actionable insights from data, often big data.

The growth in data has created numerous opportunities for all areas of data analysis. The most dramatic developments have been in the area of predictive modeling, across a wide range of application domains. For instance, recent advances in neural networks, known as deep learning, have shown impressive results in a number of challenging areas, such as image classification, speech recognition, as well as text categorization and understanding. While not as dramatic, other areas, e.g., clustering, association analysis, and anomaly detection have also continued to advance. This new edition is in response to those advances.

Overview As with the first edition, the second edition of the book provides a comprehensive introduction to data mining and is designed to be accessible and useful to students, instructors, researchers, and professionals. Areas covered include data preprocessing, predictive modeling, association analysis, cluster analysis, anomaly detection, and avoiding false discoveries. The goal is to present fundamental concepts and algorithms for each topic, thus providing the reader with the necessary background for the application of data mining to real problems. As before, classification, association analysis and cluster analysis, are each covered in a pair of chapters. The introductory chapter covers basic concepts, representative algorithms, and evaluation techniques, while the more following chapter discusses advanced concepts and algorithms. As before, our objective is to provide the reader with a sound understanding of the foundations of data mining, while still covering many important advanced topics. Because of this approach, the book is useful both as a learning tool and as a reference.

To help readers better understand the concepts that have been presented, we provide an extensive set of examples, figures, and exercises. The solutions to the original exercises, which are already circulating on the web, will be made public. The exercises are mostly unchanged from the last edition, with the exception of new exercises in the chapter on avoiding false discoveries. New exercises for the other chapters and their solutions will be available to instructors via the web. Bibliographic notes are included at the end of each chapter for readers who are interested in more advanced topics, historically important papers, and recent trends. These have also been significantly updated. The book also contains a comprehensive subject and author index.

What is New in the Second Edition? Some of the most significant improvements in the text have been in the two chapters on classification. The introductory chapter uses the decision tree classifier for illustration, but the discussion on many topics—those that apply across all classification approaches—has been greatly expanded and clarified, including topics such as overfitting, underfitting, the impact of training size, model complexity, model selection, and common pitfalls in model evaluation. Almost every section of the advanced classification chapter has been significantly updated. The material on Bayesian networks, support vector machines, and artificial neural networks has been significantly expanded. We have added a separate section on deep networks to address the current developments in this area. The discussion of evaluation, which occurs in the section on imbalanced classes, has also been updated and improved.

The changes in association analysis are more localized. We have completely reworked the section on the evaluation of association patterns (introductory chapter), as well as the sections on sequence and graph mining (advanced chapter). Changes to cluster analysis are also localized. The introductory chapter added the K-means initialization technique and an updated the discussion of cluster evaluation. The advanced clustering chapter adds a new section on spectral graph clustering. Anomaly detection has been greatly revised and expanded. Existing approaches—statistical, nearest neighbor/density-based, and clustering based—have been retained and updated, while new approaches have been added: reconstruction-based, one-class classification, and information-theoretic. The reconstruction-based approach is illustrated using autoencoder networks that are part of the deep learning paradigm. The data chapter has been updated to include discussions of mutual information and kernel-based techniques.

The last chapter, which discusses how to avoid false discoveries and produce valid results, is completely new, and is novel among other contemporary textbooks on data mining. It supplements the discussions in the other chapters with a discussion of the statistical concepts (statistical significance, p-values, false discovery rate, permutation testing, etc.) relevant to avoiding spurious results, and then illustrates these concepts in the context of data mining techniques. This chapter addresses the increasing concern over the validity and reproducibility of results obtained from data analysis. The addition of this last chapter is a recognition of the importance of this topic and an acknowledgment that a deeper understanding of this area is needed for those analyzing data.

The data exploration chapter has been deleted, as have the appendices, from the print edition of the book, but will remain available on the web. A new appendix provides a brief discussion of scalability in the context of big data.

To the Instructor As a textbook, this book is suitable for a wide range of students at the advanced undergraduate or graduate level. Since students come to this subject with diverse backgrounds that may not include extensive knowledge of statistics or databases, our book requires minimal prerequisites. No database knowledge is needed, and we assume only a modest background in statistics or mathematics, although such a background will make for easier going in some sections. As before, the book, and more specifically, the chapters covering major data mining topics, are designed to be as self-contained as possible. Thus, the order in which topics can be covered is quite flexible. The core material is covered in chapters 2 (data), 3 (classification), 4 (association analysis), 5 (clustering), and 9 (anomaly detection). We recommend at least a cursory coverage of Chapter 10 (Avoiding False Discoveries) to instill in students some caution when interpreting the results of their data analysis. Although the introductory data chapter (2) should be covered first, the basic classification (3), association analysis (4), and clustering chapters (5), can be covered in any order. Because of the relationship of anomaly detection (9) to classification (3) and clustering (5), these chapters should precede Chapter 9. Various topics can be selected from the advanced classification, association analysis, and clustering chapters (6, 7, and 8, respectively) to fit the schedule and interests of the instructor and students. We also advise that the lectures be augmented by projects or practical exercises in data mining. Although they are time consuming, such hands-on assignments greatly enhance the value of the course.

Support Materials Support materials available to all readers of this book are available on the book's website.

- PowerPoint lecture slides
- Suggestions for student projects
- Data mining resources, such as algorithms and data sets
- Online tutorials that give step-by-step examples for selected data mining techniques described in the book using actual data sets and data analysis software

Additional support materials, including solutions to exercises, are available only to instructors adopting this textbook for classroom use.

Acknowledgments Many people contributed to the first and second editions of the book. We begin by acknowledging our families to whom this book is dedicated. Without their patience and support, this project would have been impossible.

We would like to thank the current and former students of our data mining groups at the University of Minnesota and Michigan State for their contributions. Eui-Hong (Sam) Han and Mahesh Joshi helped with the initial data mining classes. Some of the exercises and presentation slides that they created can be found in the book and its accompanying slides. Students in our data mining groups who provided comments on drafts of the book or who contributed in other ways include Shyam Boriah, Haibin Cheng, Varun Chandola, Eric Eilertson, Levent Ertöz, Jing Gao, Rohit Gupta, Sridhar Iyer, Jung-Eun Lee, Benjamin Mayer, Aysel Ozgur, Uygar Oztekin, Gaurav Pandey, Kashif Riaz, Jerry Scripps, Gyorgy Simon, Hui Xiong, Jieping Ye, and Pusheng Zhang. We would also like to thank the students of our data mining classes at the University of Minnesota and Michigan State University who worked with early drafts of the book and provided invaluable feedback. We specifically note the helpful suggestions of Bernardo Craemer, Arifin Ruslim, Jamshid Vayghan, and Yu Wei.

Joydeep Ghosh (University of Texas) and Sanjay Ranka (University of Florida) class tested early versions of the book. We also received many useful suggestions directly from the following UT students: Pankaj Adhikari, Rajiv Bhatia, Frederic Bosche, Arindam Chakraborty, Meghana Deodhar, Chris Everson, David Gardner, Saad Godil, Todd Hay, Clint Jones, Ajay Joshi, Joonsoo Lee, Yue Luo, Anuj Nanavati, Tyler Olsen, Sunyoung Park, Aashish Phansalkar, Geoff Prewett, Michael Ryoo, Daryl Shannon, and Mei Yang.

Ronald Kostoff (ONR) read an early version of the clustering chapter and offered numerous suggestions. George Karypis provided invaluable LaTeX assistance in creating an author index. Irene Moulitsas also provided assistance with LaTeX and reviewed some of the appendices. Musetta Steinbach was very helpful in finding errors in the figures.

We would like to acknowledge our colleagues at the University of Minnesota and Michigan State who have helped create a positive environment for data mining research. They include Arindam Banerjee, Dan Boley, Joyce Chai, Anil Jain, Ravi Janardan, Rong Jin, George Karypis, Claudia Neuhauser, Haesun Park, William F. Punch, György Simon, Shashi Shekhar, and Jaideep Srivastava. The collaborators on our many data mining projects, who also have our gratitude, include Ramesh Agrawal, Maneesh Bhargava, Steve Cannon, Alok Choudhary, Imme Ebert-Uphoff, Auroop Ganguly, Piet C. de Groen, Fran Hill, Yongdae Kim, Steve Klooster, Kerry Long, Nihar Mahapatra, Rama Nemani, Nikunj Oza, Chris Potter, Lisiane Pruinelli, Nagiza Samatova, Jonathan Shapiro, Kevin Silverstein, Brian Van Ness, Bonnie Westra, Nevin Young, and Zhi-Li Zhang.

The departments of Computer Science and Engineering at the University of Minnesota and Michigan State University provided computing resources and a supportive environment for this project. ARDA, ARL, ARO, DOE, NASA, NOAA, and NSF provided research support for Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. In particular, Kamal Abdali, Mitra Basu, Dick Brackney, Jagdish Chandra, Joe Coughlan, Michael Coyle, Stephen Davis, Frederica Darema, Richard Hirsch, Chandrika Kamath, Tsengdar Lee, Raju Namburu, N. Radhakrishnan, James Sidoran, Sylvia Spengler, Bhavani Thuraisingham, Walt Tiernin, Maria Zemankova, Aidong Zhang, and Xiaodong Zhang have been supportive of our research in data mining and high-performance computing.

It was a pleasure working with the helpful staff at Pearson Education. In particular, we would like to thank Matt Goldstein, Kathy Smith, Carole Snyder, and Joyce Wells. We would also like to thank George Nichols, who helped with the art work and Paul Anagnostopoulos, who provided LaTeX support.

We are grateful to the following Pearson reviewers: Leman Akoglu (Carnegie Mellon University), Chien-Chung Chan (University of Akron), Zhengxin Chen (University of Nebraska at Omaha), Chris Clifton (Purdue University), Joydeep Ghosh (University of Texas, Austin), Nazli Goharian (Illinois Institute of Technology), J. Michael Hardin (University of Alabama), Jingrui He (Arizona

State University), James Hearne (Western Washington University), Hillol Kargupta (University of Maryland, Baltimore County and Agnik, LLC), Eamonn Keogh (University of California-Riverside), Bing Liu (University of Illinois at Chicago), Mariofanna Milanova (University of Arkansas at Little Rock), Srinivasan Parthasarathy (Ohio State University), Zbigniew W. Ras (University of North Carolina at Charlotte), Xintao Wu (University of North Carolina at Charlotte), and Mohammed J. Zaki (Rensselaer Polytechnic Institute).

Over the years since the first edition, we have also received numerous comments from readers and students who have pointed out typos and various other issues. We are unable to mention these individuals by name, but their input is much appreciated and has been taken into account for the second edition.

Acknowledgments for the Global Edition Pearson would like to thank and acknowledge Pramod Kumar Singh (Atal Bihari Vajpayee Indian Institute of Information Technology and Management) for contributing to the Global Edition, and Annappa (National Institute of Technology Surathkal), Komal Arora, and Soumen Mukherjee (RCC Institute of Technology) for reviewing the Global Edition.

Contents

This page is intentionally left blank

1

Introduction

Rapid advances in data collection and storage technology, coupled with the ease with which data can be generated and disseminated, have triggered the explosive growth of data, leading to the current age of **big data**. Deriving actionable insights from these large data sets is increasingly important in decision making across almost all areas of society, including business and industry; science and engineering; medicine and biotechnology; and government and individuals. However, the amount of data (volume), its complexity (variety), and the rate at which it is being collected and processed (velocity) have simply become too great for humans to analyze unaided. Thus, there is a great need for automated tools for extracting useful information from the big data despite the challenges posed by its enormity and diversity.

Data mining blends traditional data analysis methods with sophisticated algorithms for processing this abundance of data. 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 applications that require more advanced techniques for data analysis.

Business and Industry 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 server logs from e-commerce websites 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, fraud detection, and automated buying and selling. An example of the last application is high-speed stock trading, where decisions on buying and selling have to be made in less than a second using data about financial transactions. Data mining 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?” These questions have inspired the development of such data mining techniques as association analysis (Chapters 4 and 7).

As the Internet continues to revolutionize the way we interact and make decisions in our everyday lives, we are generating massive amounts of data about our online experiences, e.g., web browsing, messaging, and posting on social networking websites. This has opened several opportunities for business applications that use web data. For example, in the e-commerce sector, data about our online viewing or shopping preferences can be used to provide personalized recommendations of products. Data mining also plays a prominent role in supporting several other Internet-based services, such as filtering spam messages, answering search queries, and suggesting social updates and connections. The large corpus of text, images, and videos available on the Internet has enabled a number of advancements in data mining methods, including deep learning, which is discussed in Chapter 6. These developments have led to great advances in a number of applications, such as object recognition, natural language translation, and autonomous driving.

Another domain that has undergone a rapid big data transformation is the use of mobile sensors and devices, such as smart phones and wearable computing devices. With better sensor technologies, it has become possible to collect a variety of information about our physical world using low-cost sensors embedded on everyday objects that are connected to each other, termed the Internet of Things (IOT). This deep integration of physical sensors in digital systems is beginning to generate large amounts of diverse and distributed data about our environment, which can be used for designing convenient, safe, and energy-efficient home systems, as well as for urban planning of smart cities.

Medicine, Science, and Engineering Researchers in medicine, science, and engineering are rapidly accumulating data that is key to significant 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 spatiotemporal 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 the following: "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 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, high-dimensional nature of data requires new data analysis methods. In addition to analyzing gene expression 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.

Another example is the use of data mining techniques to analyze electronic health record (EHR) data, which has become increasingly available. Not very long ago, studies of patients required manually examining the physical records of individual patients and extracting very specific pieces of information pertinent to the particular question being investigated. EHRs allow for a faster and broader exploration of such data. However, there are significant challenges since the observations on any one patient typically occur during their visits to a doctor or hospital and only a small number of details about the health of the patient are measured during any particular visit.

Currently, EHR analysis focuses on simple types of data, e.g., a patient's blood pressure or the diagnosis code of a disease. However, large amounts of more complex types of medical data are also being collected, such as electrocardiograms (ECGs) and neuroimages from magnetic resonance imaging (MRI) or functional Magnetic Resonance Imaging (fMRI). Although challenging to analyze, this data also provides vital information about patients. Integrating and analyzing such data, with traditional EHR and genomic data is one of the capabilities needed to enable precision medicine, which aims to provide more personalized patient care.

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 data sets in order to find novel and useful patterns that might otherwise remain unknown. They also provide the capability to predict the outcome of a future observation, such as the amount a customer will spend at an online or a brick-and-mortar store.

Not all information discovery tasks are considered to be data mining. Examples include queries, e.g., looking up individual records in a database or finding web pages that contain a particular set of keywords. This is because such tasks can be accomplished through simple interactions with a database management system or an information retrieval system. These systems rely on traditional computer science techniques, which include sophisticated indexing structures and query processing algorithms, for efficiently organizing and retrieving information from large data repositories. Nonetheless, data mining techniques have been used to enhance the performance of such systems by improving the quality of the search results based on their relevance to the input queries.

Data Mining and Knowledge Discovery in Databases

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 steps, from data preprocessing to postprocessing of data mining results.

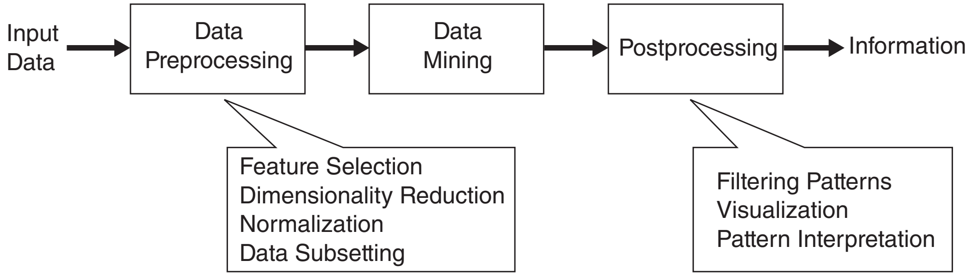


Figure 1.1. The process of knowledge discovery in databases (KDD).

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 a 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 to ensure that only valid and useful results are incorporated into the decision support system. An example of postprocessing is visualization, which allows analysts to explore the data and the data mining results from a variety of viewpoints. Hypothesis testing methods can also be applied during postprocessing to eliminate spurious data mining results. (See Chapter 10.)

1.2 Motivating Challenges

As mentioned earlier, traditional data analysis techniques have often encountered practical difficulties in meeting the challenges posed by big data applications. 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 terabytes, petabytes, or even exabytes are becoming common. If data mining algorithms are to handle these massive data sets, 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. A general overview of techniques for scaling up data mining algorithms is given in Appendix F.

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 due to issues such as the curse of dimensionality (to be discussed in Chapter 2). 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 web and social media data containing text, hyperlinks, images, audio, and videos; DNA data with sequential and three-dimensional structure; and climate data that consists of measurements (temperature, pressure, etc.) at various times and 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. The key challenges faced by distributed data mining algorithms include the following: (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 and privacy 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.

1.3 The Origins of Data Mining

While data mining has traditionally been viewed as an intermediate process within the KDD framework, as shown in Figure 1.1, it has emerged over the years as an academic field within computer science, focusing on all aspects of KDD, including data preprocessing, mining, and postprocessing. Its origin can be traced back to the late 1980s, following a series of workshops organized on the topic of knowledge discovery in databases. The workshops brought together researchers from different disciplines to discuss the challenges and opportunities in applying computational techniques to extract actionable knowledge from large databases. The workshops quickly grew into hugely popular conferences that were attended by researchers and practitioners from both the academia and industry. The success of these conferences, along with the interest shown by businesses and industry in recruiting new hires with a data mining background, have fueled the tremendous growth of this field.

The field was initially built upon the methodology and algorithms that researchers had previously used. In particular, data mining researchers draw 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, and extending them to solve the challenges of mining big data.

A number of other areas also play key supporting roles. In particular, database systems are needed to provide support for efficient storage, indexing,

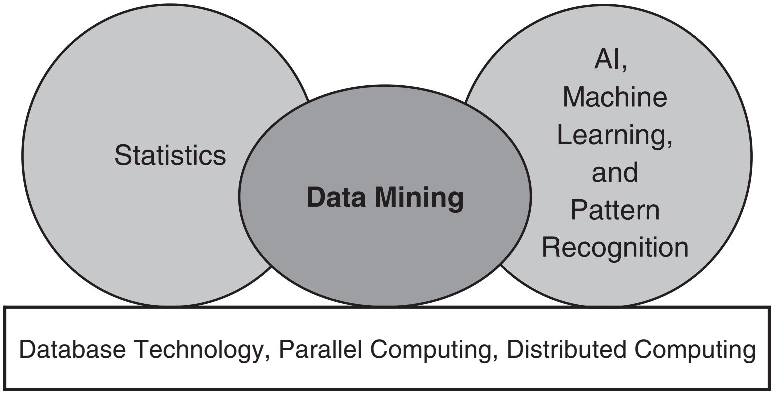

Statistics
Data Mining
AI, Machine Learning, and Pattern Recognition
Database Technology, Parallel Computing, Distributed Computing

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

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.

Data Science and Data-Driven Discovery

Data science is an interdisciplinary field that studies and applies tools and techniques for deriving useful insights from data. Although data science is regarded as an emerging field with a distinct identity of its own, the tools and techniques often come from many different areas of data analysis, such as data mining, statistics, AI, machine learning, pattern recognition, database technology, and distributed and parallel computing. (See Figure 1.2.)

The emergence of data science as a new field is a recognition that, often, none of the existing areas of data analysis provides a complete set of tools for the data analysis tasks that are often encountered in emerging applications. Instead, a broad range of computational, mathematical, and statistical skills is often required. To illustrate the challenges that arise in analyzing such data, consider the following example. Social media and the Web present new opportunities for social scientists to observe and quantitatively measure human behavior on a large scale. To conduct such a study, social scientists work with analysts who possess skills in areas such as web mining, natural language processing (NLP), network analysis, data mining, and statistics. Compared to more traditional research in social science, which is often based on surveys, this analysis requires a broader range of skills and tools, and involves far larger amounts of data. Thus, data science is, by necessity, a highly interdisciplinary field that builds on the continuing work of many fields.

The data-driven approach of data science emphasizes the direct discovery of patterns and relationships from data, especially in large quantities of data, often without the need for extensive domain knowledge. A notable example of the success of this approach is represented by advances in neural networks, i.e., deep learning, which have been particularly successful in areas which have long proved challenging, e.g., recognizing objects in photos or videos and words in speech, as well as in other application areas. However, note that this is just one example of the success of data-driven approaches, and dramatic improvements have also occurred in many other areas of data analysis. Many of these developments are topics described later in this book.

Some cautions on potential limitations of a purely data-driven approach are given in the Bibliographic Notes.

1.4 Data Mining Tasks

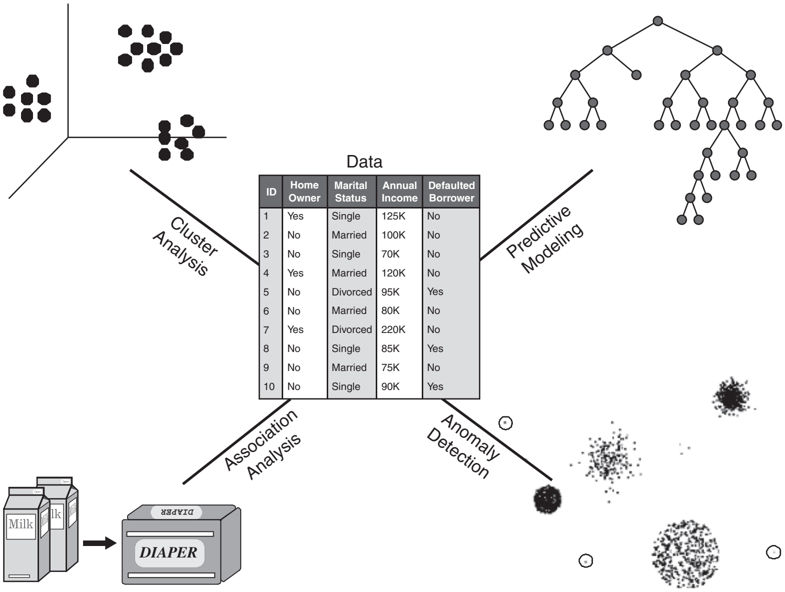
Data mining tasks are generally divided into two major categories:

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.

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.



ID	Home Owner	Marital Status	Annual 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	80K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 1.3. Four of the core data mining tasks.

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 who 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 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. 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. ■

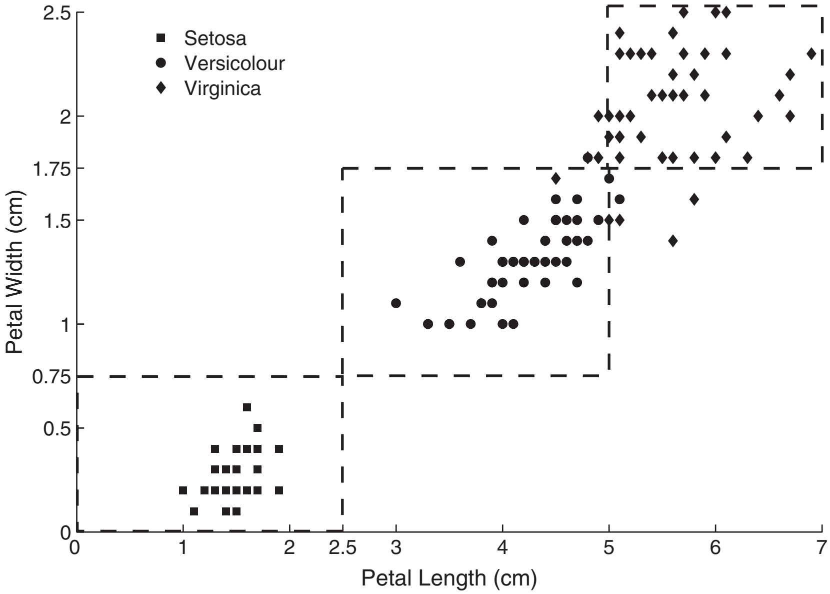


Figure 1.4. Petal width versus petal length for 150 Iris flowers.

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 {`Diapers`} $\longrightarrow$ {`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. ■

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}

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 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	Word-frequency pairs
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, such as droughts, floods, fires, hurricanes, etc.

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. Chapters 3 and 6 cover classification. Chapter 3 provides a foundation by discussing decision tree classifiers and several issues that are important to all classification: overfitting, underfitting, model selection, and performance evaluation. Using this foundation, Chapter 6 describes a number of other important classification techniques: rule-based systems, nearest neighbor classifiers, Bayesian classifiers, artificial neural networks, including deep learning, support vector machines, and ensemble classifiers, which are collections of classifiers. The multiclass and imbalanced class problems are also discussed. These topics can be covered independently.

Association analysis is explored in Chapters 4 and 7. Chapter 4 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 5 and 8. Chapter 5 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 8, including fuzzy and probabilistic clustering, Self-Organizing Maps (SOM), graph-based clustering, spectral clustering, and density-based clustering. There is also a discussion of scalability issues and factors to consider when selecting a clustering algorithm.

Chapter 9, is on anomaly detection. After some basic definitions, several different types of anomaly detection are considered: statistical, distance-based, density-based, clustering-based, reconstruction-based, one-class classification, and information theoretic. The last chapter, Chapter 10, supplements the discussions in the other chapters with a discussion of the statistical concepts important for avoiding spurious results, and then discusses those concepts in the context of data mining techniques studied in the previous chapters. These techniques include statistical hypothesis testing, p-values, the false discovery rate, and permutation testing. Appendices A through F give a brief review of important topics that are used in portions of the book: linear algebra, dimensionality reduction, statistics, regression, optimization, and scaling up data mining techniques for big data.

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 section of the appropriate chapter. References to topics not covered in this book, such as mining streaming data 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 [16], Han et al. [29], Hand et al. [31], Roiger and Geatz [50], Zaki and Meira [61], and Aggarwal [2]. Data mining books with a stronger emphasis on business applications include the works by Berry and Linoff [5], Pyle [47], and Parr Rud [45]. Books with an emphasis on statistical learning include those by Cherkassky and Mulier [11], and Hastie et al. [32]. Similar books with an emphasis on machine learning or pattern recognition are those by Duda et al. [15], Kantardzic [34], Mitchell [43], Webb [57], and Witten and Frank [58]. There are also some more specialized books: Chakrabarti [9] (web mining), Fayyad et al. [20] (collection of early articles on data mining), Fayyad et al. [18] (visualization), Grossman et al. [25] (science and engineering), Kargupta and Chan [35] (distributed data mining), Wang et al. [56] (bioinformatics), and Zaki and Ho [60] (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 Conference and Workshop on Neural Information Processing Systems

(NIPS),the International Conference on Machine Learning (ICML), 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 National Conference on Artificial Intelligence (AAAI), the IEEE International Conference on Big Data, and the IEEE International Conference on Data Science and Advanced Analytics (DSAA).

Journal publications on data mining include IEEE Transactions on Knowledge and Data Engineering, Data Mining and Knowledge Discovery, Knowledge and Information Systems, ACM Transactions on Knowledge Discovery from Data, Statistical Analysis and Data Mining, and Information Systems. There are various open-source data mining software available, including Weka [27] and Scikit-learn [46]. More recently, data mining software such as Apache Mahout and Apache Spark have been developed for large-scale problems on the distributed computing platform.

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. [19] describe data mining and how it fits into the total knowledge discovery process. Chen et al. [10] give a database perspective on data mining. Ramakrishnan and Grama [48] provide a general discussion of data mining and present several viewpoints. Hand [30] describes how data mining differs from statistics, as does Friedman [21]. Lambert [40] 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. [23] consider the lessons that statistics may have for data mining. Smyth et al. [53] 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. Han et al. [28] consider emerging applications in data mining and Smyth [52] describes some research challenges in data mining. Wu et al. [59] discuss how developments in data mining research can be turned into practical tools. Data mining standards are the subject of a paper by Grossman et al. [24]. Bradley [7] discusses how data mining algorithms can be scaled to large data sets.

The emergence of new data mining applications has produced 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 [3], Clifton et al. [12] and

Kargupta et al. [36]. Vassilios et al. [55] provide a survey. Another area of concern is the bias in predictive models that may be used for some applications, e.g., screening job applicants or deciding prison parole [39]. Assessing whether such applications are producing biased results is made more difficult by the fact that the predictive models used for such applications are often black box models, i.e., models that are not interpretable in any straightforward way.

Data science, its constituent fields, and more generally, the new paradigm of knowledge discovery they represent [33], have great potential, some of which has been realized. However, it is important to emphasize that data science works mostly with observational data, i.e., data that was collected by various organizations as part of their normal operation. The consequence of this is that sampling biases are common and the determination of causal factors becomes more problematic. For this and a number of other reasons, it is often hard to interpret the predictive models built from this data [42, 49]. Thus, theory, experimentation and computational simulations will continue to be the methods of choice in many areas, especially those related to science.

More importantly, a purely data-driven approach often ignores the existing knowledge in a particular field. Such models may perform poorly, for example, predicting impossible outcomes or failing to generalize to new situations. However, if the model does work well, e.g., has high predictive accuracy, then this approach may be sufficient for practical purposes in some fields. But in many areas, such as medicine and science, gaining insight into the underlying domain is often the goal. Some recent work attempts to address these issues in order to create theory-guided data science, which takes pre-existing domain knowledge into account [17, 37].

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 [14] (classification), Giannella et al. [22] (association analysis), Guha et al. [26] (clustering), Kifer et al. [38] (change detection), Papadimitriou et al. [44] (time series), and Law et al. [41] (dimensionality reduction).

Another area of interest is recommender and collaborative filtering systems [1, 6, 8, 13, 54], which suggest movies, television shows, books, products, etc. that a person might like. In many cases, this problem, or at least a component of it, is treated as a prediction problem and thus, data mining techniques can be applied [4, 51].

Bibliography

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[2] C. Aggarwal. *Data mining: The Textbook*. Springer, 2009.

[3] 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.

[4] X. Amatriain and J. M. Pujol. Data mining methods for recommender systems. In *Recommender Systems Handbook*, pages 227–262. Springer, 2015.

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

[6] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[7] 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.

[8] R. Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

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

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

[11] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley-IEEE Press, 2nd edition, 1998.

[12] 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.

[13] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pages 107–144, 2011.

[14] 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.

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

[16] M. H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2006.

[17] J. H. Faghmous, A. Banerjee, S. Shekhar, M. Steinbach, V. Kumar, A. R. Ganguly, and N. Samatova. Theory-guided data science for climate change. *Computer*, 47(11):74–78, 2014.

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

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

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

[26] 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.

[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009.

[28] 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.

[29] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, 3rd edition, 2011.

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

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

[32] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, Prediction*. Springer, 2nd edition, 2009.

[33] T. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

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

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

[36] 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.

[37] A. Karpatne, G. Atluri, J. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar. Theory-guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[38] 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.

[39] J. Kleinberg, J. Ludwig, and S. Mullainathan. A Guide to Solving Social Problems with Machine Learning. *Harvard Business Review*, December 2016.

[40] 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.

[41] 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.

[42] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

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

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

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

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

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

[49] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

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

[51] J. Schafer. The Application of Data-Mining to Recommender Systems. *Encyclopedia of data warehousing and mining*, 1:44–48, 2009.

[52] 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.

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

[54] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[55] 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.

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

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

[58] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 3rd edition, 2011.

[59] 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.

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

[61] M. J. Zaki and W. Meira Jr. *Data Mining and Analysis: Fundamental Concepts and Algorithms.* Cambridge University Press, New York, 2014.

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.

This page is intentionally left blank

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 often 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. Indeed, 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, sometimes needs 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*, *instance*, *observation*, or *entity*. In turn, data objects are described by a number of **attributes** that capture the 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 consider the types of attributes 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 can 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 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 is common to refer to the type of an attribute as the **type of a measurement scale**. It should be apparent from the previous discussion that an attribute can be described using different measurement scales and that the properties of an attribute need not be the same as the properties of the values used to measure it. In other words, the values used to represent an attribute can 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 because, 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 side of the figure, but not by those on the left side. More specifically, the measurement scale on the left side captures only the ordering of the length attribute, while the scale on the right 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. ■

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.

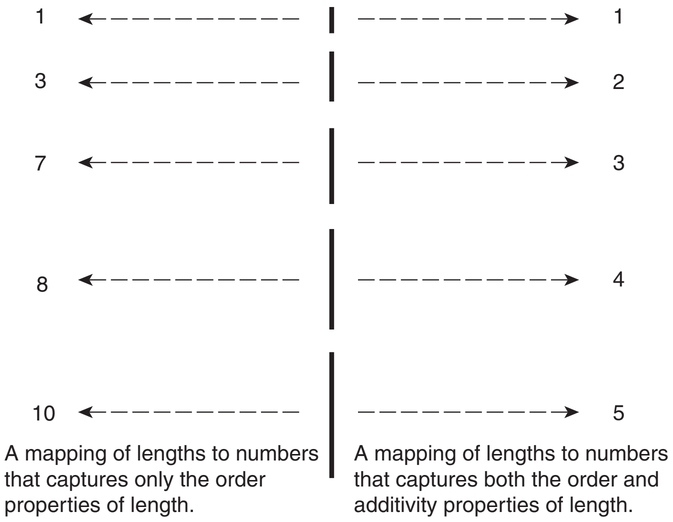

1
3
7
8
10
1
2
3
4
5
A mapping of lengths to numbers that captures only the order properties of length.
A mapping of lengths to numbers that captures both the order and additivity properties of length.

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** $\times$ 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. $(\times, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

this does not mean that the statistical 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

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	$new\_value = a \times old\_value + b$, 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	$new\_value = a \times old\_value$	Length can be measured in meters or feet.

psychologist who originally defined the types of attributes shown in Table 2.2, defined them in terms of these **permissible transformations**. For example, 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 by 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 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 be measured and represented only 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 in which each object is a student and each attribute records whether 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 4. 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**.

General Comments on Levels of Measurement

As described in the rest of this chapter, there are many diverse types of data. The previous discussion of measurement scales, while useful, is not complete and has some limitations. We provide the following comments and guidance.

- **Distinctness, order, and meaningful intervals and ratios are only four properties of data—many others are possible**. For instance, some data is inherently cyclical, e.g., position on the surface of the Earth or time. As another example, consider set valued attributes, where each attribute value is a set of elements, e.g., the set of movies seen in the last year. Define one set of elements (movies) to be greater (larger) than a second set if the second set is a subset of the first. However, such a relationship defines only a partial order that does not match any of the attribute types just defined.

- **The numbers or symbols used to capture attribute values may not capture all the properties of the attributes or may suggest properties that are not there**. An illustration of this for integers was presented in Example 2.3, i.e., averages of IDs and out of range ages.

- **Data is often transformed for the purpose of analysis**—see Section 2.3.7. This often changes the distribution of the observed variable to a distribution that is easier to analyze, e.g., a Gaussian (normal) distribution. Often, such transformations only preserve the order of the original values, and other properties are lost. Nonetheless, if the desired outcome is a statistical test of differences or a predictive model, such a transformation is justified.

- **The final evaluation of any data analysis, including operations on attributes, is whether the results make sense from a domain point of view**.

In summary, it can be challenging to determine which operations can be performed on a particular attribute or a collection of attributes without compromising the integrity of the analysis. Fortunately, established practice often serves as a reliable guide. Occasionally, however, standard practices are erroneous or have limitations.

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, distribution, and resolution.

Dimensionality The dimensionality of a data set is the number of attributes that the objects in the data set possess. Analyzing data with a small number of dimensions tends to be qualitatively different from analyzing moderate or high-dimensional data. Indeed, the difficulties associated with the analysis of 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.

Distribution The distribution of a data set is the frequency of occurrence of various values or sets of values for the attributes comprising data objects. Equivalently, the distribution of a data set can be considered as a description of the concentration of objects in various regions of the data space. Statisticians have enumerated many types of distributions, e.g., Gaussian (normal), and described their properties. (See Appendix C.) Although statistical approaches for describing distributions can yield powerful analysis techniques, many data sets have distributions that are not well captured by standard statistical distributions.

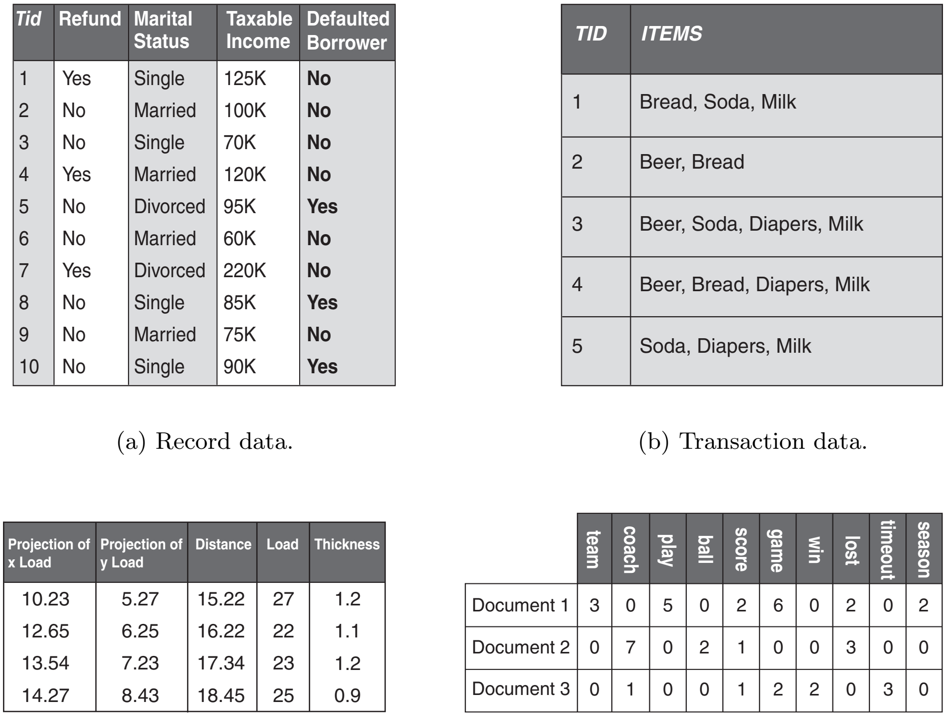
As a result, many data mining algorithms do not assume a particular statistical distribution for the data they analyze. However, some general aspects of distributions often have a strong impact. For example, suppose a categorical attribute is used as a class variable, where one of the categories occurs 95% of the time, while the other categories together occur only 5% of the time. This **skewness** in the distribution can make classification difficult as discussed in Section 6.11. (Skewness has other impacts on data analysis that are not discussed here.)

A special case of skewed data is **sparsity**. For sparse binary, count or continuous data, most attributes of an object have values of 0. In many cases, fewer than 1% of the values 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. Indeed, some data mining algorithms, such as the association rule mining algorithms described in Chapter 4, work well only for sparse data. Finally, note that often the attributes in sparse data sets are asymmetric attributes.

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 can 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.



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, Diapers, Milk
4	Beer, Bread, Diapers, Milk
5	Soda, Diapers, 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.

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 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 all the data objects in a collection of data 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, 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 where 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—the "bag of words" approach—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.

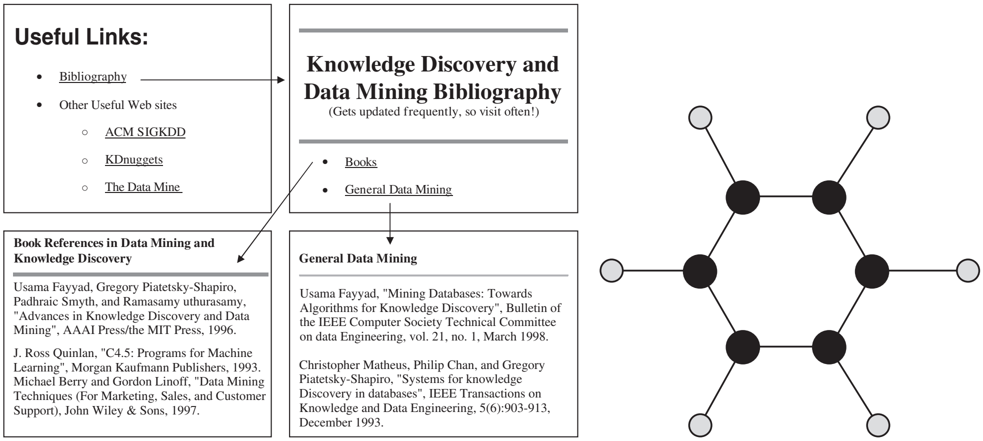
Data with Relationships among Objects The relationships among objects frequently convey important information. In such cases, the data is often represented as a graph. In particular, the data objects are mapped to nodes of the graph, while the relationships among objects are captured by the links between 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. Another important example of such graph data are the social networks, where data objects are people and the relationships among them are their interactions via social media.

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. Frequent graph mining, which is a branch of data mining that analyzes such data, is considered in Section 7.5.

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 Transaction Data Sequential transaction data can be thought of as an extension of transaction data, where each transaction 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



Useful Links:

- Bibliography
- Other Useful Web sites
  - ACM SIGKDD
  - KDnuggets
  - The Data Mine

Knowledge Discovery and Data Mining Bibliography

(Gets updated frequently, so visit often!)

- Books
- General Data Mining

Book References in Data Mining and Knowledge Discovery

Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy uthurasamy, "Advances in Knowledge Discovery and Data Mining", AAAI Press/the MIT Press, 1996.

J. Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993. Michael Berry and Gordon Linoff, "Data Mining Techniques (For Marketing, Sales, and Customer Support), John Wiley & Sons, 1997.

General Data Mining

Usama Fayyad, "Mining Databases: Towards Algorithms for Knowledge Discovery", Bulletin of the IEEE Computer Society Technical Committee on data Engineering, vol. 21, no. 1, March 1998.

Christopher Matheus, Philip Chan, and Gregory Piatetsky-Shapiro, "Systems for knowledge Discovery in databases", IEEE Transactions on Knowledge and Data Engineering, 5(6):903-913, December 1993.

(a) Linked web pages.

(b) Benzene molecule.

Figure 2.3. Different variations of graph data.

"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, 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 about 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*.

Time Series Data Time series data is a special type of ordered data where 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, such as time series, it is important to consider **temporal autocorrelation**; i.e., if

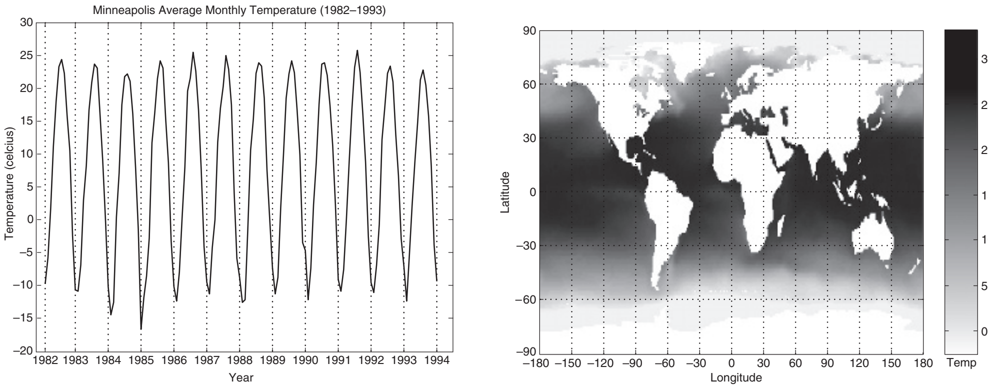
Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

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.

two measurements are close in time, then the values of those measurements are often very similar.

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.

Spatial and Spatio-Temporal Data Some objects have spatial attributes, such as positions or areas, in addition to other types of attributes. An example of spatial data is weather data (precipitation, temperature, pressure) that is collected for a variety of geographical locations. Often such measurements are collected over time, and thus, the data consists of time series at various locations. In that case, we refer to the data as spatio-temporal data. Although analysis can be conducted separately for each specific time or location, a more complete analysis of spatio-temporal data requires consideration of both the spatial and temporal aspects of the data.

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. Note that spatial autocorrelation is analogous to temporal autocorrelation.

Important examples of spatial and spatio-temporal 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 at various instants in time can be recorded for each grid point in the simulation. A different type of spatio-temporal data arises from tracking the trajectories of objects, e.g., vehicles, in time and space.

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 ignores the temporal autocorrelation of the attributes or the spatial autocorrelation of the data objects, i.e., the locations on the spatial grid.

2.2 Data Quality

Data mining algorithms 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 can be missing. In other cases, there can 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 involve both measurement and data collection problems: outliers, missing and inconsistent values, and duplicate data.

Measurement and Data Collection Errors

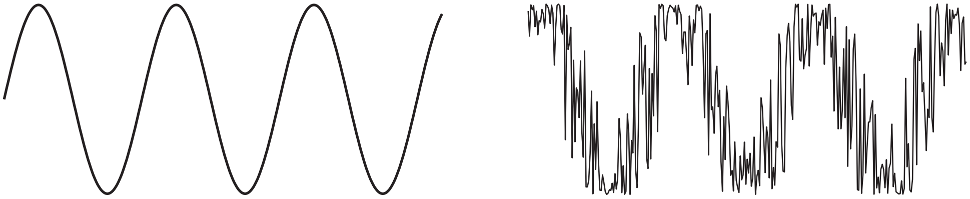
The term **measurement error** refers to any problem resulting from the measurement process. A common problem is that the value recorded differs from the true value to some extent. For continuous attributes, the numerical difference of the measured and true value is called the **error**. The term **data 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, certain types of data errors are commonplace, and well-developed techniques often exist 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 typically involves 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 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.

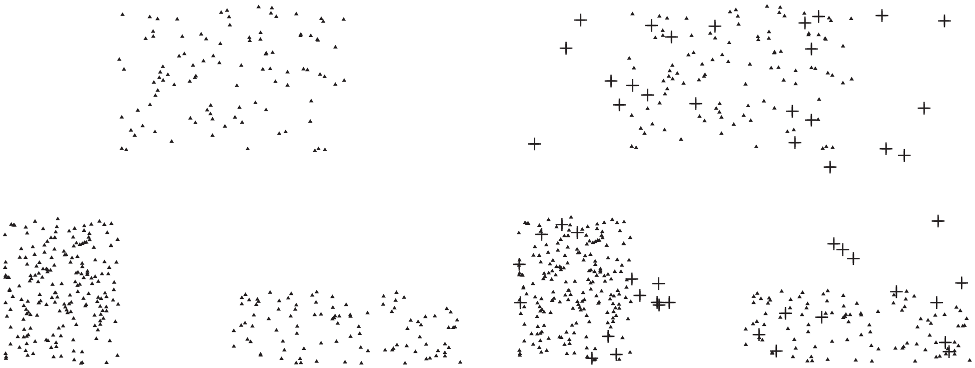
The term noise is often used in connection with data that has a spatial or temporal component. In such cases, techniques from signal or image processing



(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.

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 can 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.

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 be determined only 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 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 record the length of data only to the nearest millimeter. The precision of such a measurement would be ± 0.5mm. We do not review the details of working with significant digits because 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 about 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, they can be referred to as **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. Unlike noise, outliers can be legitimate data objects or values that we are interested in detecting. For instance, in fraud and network intrusion detection, the goal is to find unusual objects or events from among a large number of normal ones. Chapter 9 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 is 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, because 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 is possible 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 can 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 one or more values of corresponding attributes are usually different, and these inconsistent values must be 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 can 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 17 on page 128.

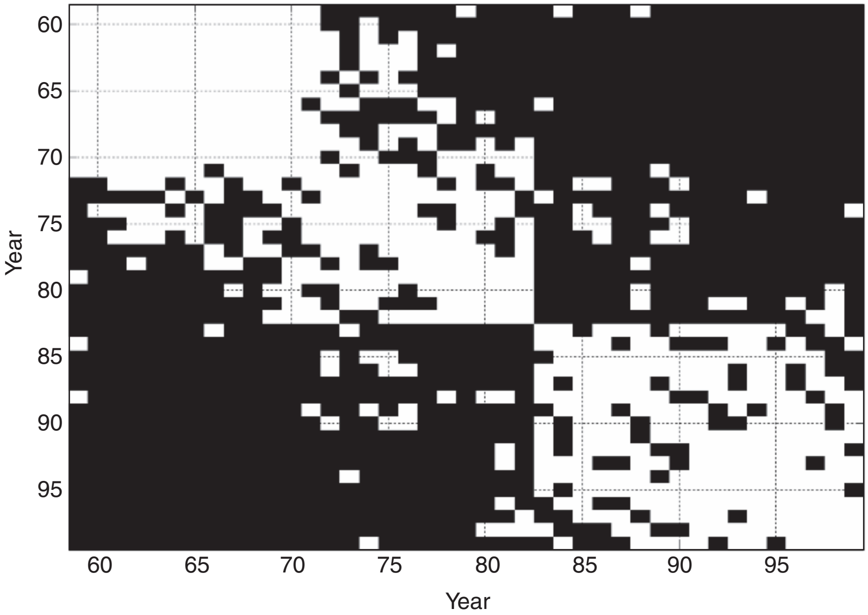


Figure 2.7. Correlation of SST data between pairs of years. White areas indicate positive correlation. Black areas indicate negative correlation.

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, many issues 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 lead to erroneous results when applied to the broader population.

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 usually 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 consider 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 topics fall into two categories: selecting data objects and attributes for the analysis or for 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 about 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 per day 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 in terms of a higher level category, e.g., televisions versus electronics.

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 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). References to OLAP are given in the Bibliographic Notes.

There are several motivations for aggregation. First, the smaller data sets resulting from data reduction require less memory and processing time,

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	...
⋮	⋮	⋮	⋮	⋮	

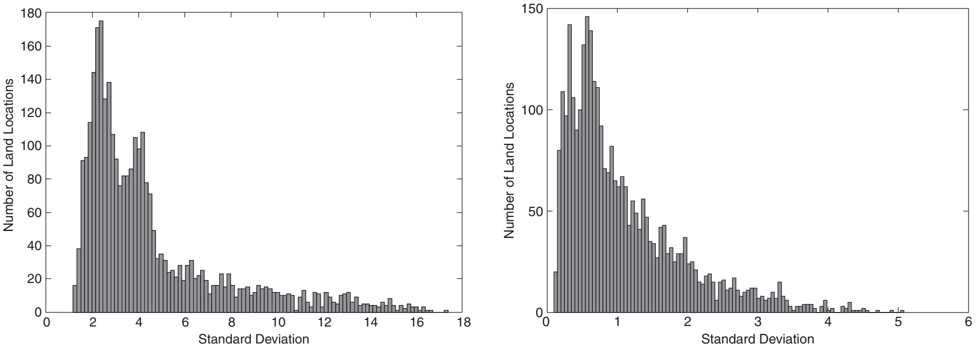
and hence, aggregation often enables the use of more expensive data mining algorithms. Second, aggregation can act as a change of scope or scale by providing a high-level view of the data instead of a low-level view. In the previous example, aggregating over store locations and months gives us a monthly, 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 values 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 1982–1993. 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.

■

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



(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 1982–1993.

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 usually sample because it is too computationally expensive in terms of the memory or time required 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 computationally 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 technique.

Sampling Approaches

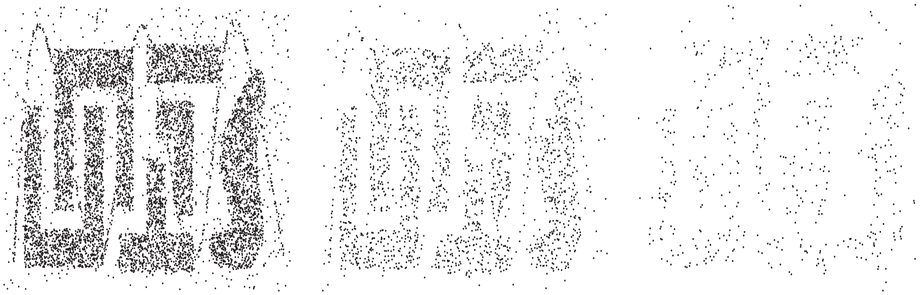
There are many sampling techniques, but only a few of the most basic ones and their variations will be covered here. The simplest type of sampling is **simple random sampling**. For this type of sampling, there is an equal probability of selecting any particular object. There are two variations on random sampling (and other sampling techniques as well): (1) **sampling without replacement**—as each object 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 because 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 object types 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 can 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. ■

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 consisting 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



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

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

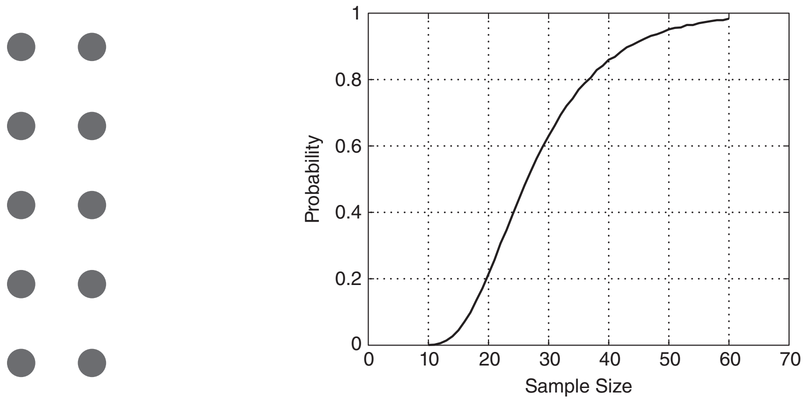
> to each other, but not very similar to objects in different groups. Figure 2.10(a) shows an idealized set of clusters (groups) from which these points might be drawn.

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 126.

■

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



(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.

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 of 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 usually involves 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 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. Thus, the data objects we observe are quite possibly not a representative sample of all possible objects. 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 differences in density and in the distances between points, which are critical for clustering, become less meaningful. (This is discussed further in Sections 8.1.2, 8.4.6, and 8.4.8.) As a result, many clustering and classification algorithms (and other data analysis algorithms) have trouble with high-dimensional data leading to 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 3, 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 so that the attributes are non-redundant.

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.

Because 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, trade-offs 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

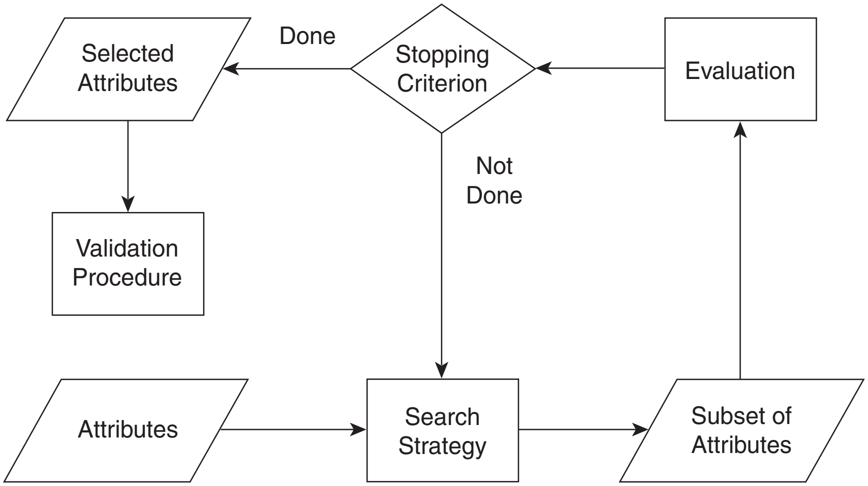


Figure 2.11. Flowchart of a feature subset selection process.

classification 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 algorithm, 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, 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 validation 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 can sometimes be determined automatically. For example, some classification schemes, such as support vector machines (Chapter 6), 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. Two related methodologies for creating new attributes are described next: feature extraction and mapping the data to a new space.

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 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.

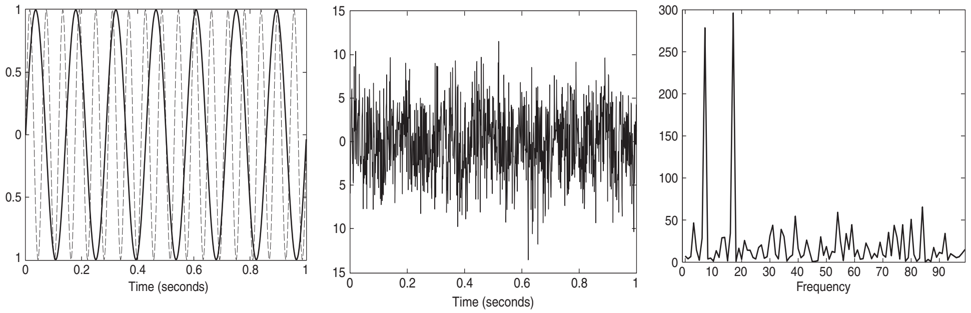
Although feature extraction is often complicated, Example 2.10 illustrates that it can be relatively straightforward.

Example 2.10 (Density)**.** 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 such simple feature extraction by exploring basic mathematical combinations of existing attributes, the most common approach is to construct features using domain expertise. ■

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, 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 Example 2.11, 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 2.11 (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. ■



(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.

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.

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 can 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 or 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 a way that satisfies a criterion that is thought to have a relationship to good performance for the data mining task being considered. In general, the best discretization depends on the algorithm being used, as well as the other

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

attributes being considered. Typically, however, the discretization of each attribute is considered in isolation.

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 asymmetric binary attribute for each categorical value, as shown in Table 2.6. If the number of resulting attributes is too large, then the techniques described in the following sections can be used to reduce the number of categorical values before binarization.

Likewise, for association problems, it can 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. Transformation of a continuous attribute to a categorical attribute involves two subtasks: deciding how many categories, n, 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 can 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 5), 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.

In this particular example, if we measure the performance of a discretization technique by the extent to which different objects that clump together have the same categorical value, then K-means performs best, followed by equal frequency, and finally, equal width. More generally, the best discretization will depend on the application and often involves domain-specific discretization. For example, the discretization of people into low income, middle income, and high income is based on economic factors.

■

Supervised Discretization If classification is our application and class labels are known for some data objects, then discretization approaches that use class labels often produce better classification. 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, i.e., the extent to which an interval contains a single class label. 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 in a separate interval and create larger intervals by merging adjacent intervals that are similar according to a statistical test. An alternative to this bottom-up approach is a top-down approach that 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.