Data

Mining

Third

Edition

The Morgan Kaufmann Series in Data Management Systems (Selected Titles)

Joe Celko's Data, Measurements, and Standards in SQL
Joe Celko

Information Modeling and Relational Databases, 2nd *Edition*
Terry Halpin, Tony Morgan

Joe Celko's Thinking in Sets
Joe Celko

Business Metadata
Bill Inmon, Bonnie O'Neil, Lowell Fryman

Unleashing Web 2.0
Gottfried Vossen, Stephan Hagemann

Enterprise Knowledge Management
David Loshin

The Practitioner's Guide to Data Quality Improvement
David Loshin

Business Process Change, 2nd *Edition*
Paul Harmon

IT Manager's Handbook, 2nd *Edition*
Bill Holtsnider, Brian Jaffe

Joe Celko's Puzzles and Answers, 2nd *Edition*
Joe Celko

Architecture and Patterns for IT Service Management, 2nd *Edition,*
Resource Planning and Governance
Charles Betz

Joe Celko's Analytics and OLAP in SQL
Joe Celko

Data Preparation for Data Mining Using SAS
Mamdouh Refaat

Querying XML: XQuery, XPath, and SQL/ XML in Context
Jim Melton, Stephen Buxton

Data Mining: Concepts and Techniques, 3rd *Edition*
Jiawei Han, Micheline Kamber, Jian Pei

Database Modeling and Design: Logical Design, 5th *Edition*
Toby J. Teorey, Sam S. Lightstone, Thomas P. Nadeau, H. V.
Jagadish *Foundations of Multidimensional and Metric Data*
Structures
Hanan Samet

Joe Celko's SQL for Smarties: Advanced SQL Programming,
4*th Edition* Joe Celko

Moving Objects Databases
Ralf Hartmut Guting, Markus Schneider ¨

Joe Celko's SQL Programming Style
Joe Celko

Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration Earl Cox

Data Modeling Essentials, 3*rd Edition*
Graeme C. Simsion, Graham C. Witt

Developing High Quality Data Models
Matthew West

Location-Based Services
Jochen Schiller, Agnes Voisard

Managing Time in Relational Databases: How to Design, Update, and Query Temporal Data Tom Johnston, Randall Weis

Database Modeling with Microsoft^{R} *Visio for Enterprise Architects*
Terry Halpin, Ken Evans, Patrick Hallock, Bill Maclean

Designing Data-Intensive Web Applications
Stephano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai,
Maristella Matera *Mining the Web: Discovering Knowledge from Hypertext Data*
Soumen Chakrabarti

Advanced SQL: 1999—Understanding Object-Relational and Other Advanced Features Jim Melton

Database Tuning: Principles, Experiments, and Troubleshooting Techniques Dennis Shasha, Philippe Bonnet

SQL: 1999—Understanding Relational Language Components
Jim Melton, Alan R. Simon

Information Visualization in Data Mining and Knowledge Discovery
Edited by Usama Fayyad, Georges G. Grinstein, Andreas Wierse

Transactional Information Systems
Gerhard Weikum, Gottfried Vossen

Spatial Databases
Philippe Rigaux, Michel Scholl, and Agnes Voisard

Managing Reference Data in Enterprise Databases
Malcolm Chisholm

Understanding SQL and Java Together
Jim Melton, Andrew Eisenberg

Database: Principles, Programming, and Performance, 2*nd Edition*
Patrick and Elizabeth O'Neil

The Object Data Standard
Edited by R. G. G. Cattell, Douglas Barry

Data on the Web: From Relations to Semistructured Data and XML
Serge Abiteboul, Peter Buneman, Dan Suciu

Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 3*rd Edition*
Ian Witten, Eibe Frank, Mark A. Hall

Joe Celko's Data and Databases: Concepts in Practice
Joe Celko

Developing Time-Oriented Database Applications in SQL
Richard T. Snodgrass

Web Farming for the Data Warehouse
Richard D. Hackathorn

Management of Heterogeneous and Autonomous Database Systems

Edited by Ahmed Elmagarmid, Marek Rusinkiewicz, Amit Sheth

Object-Relational DBMSs, 2nd *Edition*
Michael Stonebraker, Paul Brown, with Dorothy Moore

Universal Database Management: A Guide to Object/Relational Technology Cynthia Maro Saracco

Readings in Database Systems, 3rd *Edition*
Edited by Michael Stonebraker, Joseph M. Hellerstein

Understanding SQL's Stored Procedures: A Complete Guide to SQL/PSM Jim Melton

Principles of Multimedia Database Systems
V. S. Subrahmanian

Principles of Database Query Processing for Advanced Applications
Clement T. Yu, Weiyi Meng

Advanced Database Systems
Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T. Snodgrass, V. S. Subrahmanian, Roberto Zicari

Principles of Transaction Processing, 2nd *Edition*
Philip A. Bernstein, Eric Newcomer

Using the New DB2: IBM's Object-Relational Database System
Don Chamberlin

Distributed Algorithms
Nancy A. Lynch

Active Database Systems: Triggers and Rules for Advanced Database Processing Edited by Jennifer Widom, Stefano Ceri

Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach Michael L. Brodie, Michael Stonebraker

Atomic Transactions
Nancy Lynch, Michael Merritt, William Weihl, Alan Fekete

Query Processing for Advanced Database Systems
Edited by Johann Christoph Freytag, David Maier, Gottfried Vossen

Transaction Processing
Jim Gray, Andreas Reuter

Database Transaction Models for Advanced Applications
Edited by Ahmed K. Elmagarmid

A Guide to Developing Client/Server SQL Applications
Setrag Khoshafian, Arvola Chan, Anna Wong, Harry K. T. Wong

Data Mining Concepts and Techni ques

Third

Edition

Jiawei Han

University of Illinois at

Urbana–Champaign

Micheline

Kamber

Jian Pei

Simon Fraser University

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier

Morgan Kaufmann Publishers is an imprint of Elsevier.
225 Wyman Street, Waltham, MA 02451, USA

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods or professional practices, may become necessary. Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information or methods described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

Han, Jiawei.
Data mining : concepts and techniques / Jiawei Han, Micheline Kamber, Jian Pei.
– 3rd ed. p. cm.
ISBN 978-0-12-381479-1
1. Data mining. I. Kamber, Micheline. II. Pei, Jian. III. Title.
QA76.9.D343H36 2011
006.3^{0}12–dc22 2011010635

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

For information on all Morgan Kaufmann publications, visit our
Web site at *www.mkp.com* or *www.elsevierdirect.com*

Printed in the United States of America
11 12 13 14 15 10 9 8 7 6 5 4 3 2 1

To Y. Dora and Lawrence for your love and encouragement

J.H.

To Erik, Kevan, Kian, and Mikael for your love and inspiration

M.K.

To my wife, Jennifer, and daughter, Jacqueline

This page intentionally left blank

Contents

Foreword

Analyzing large amounts of data is a necessity. Even popular science books, like “super crunchers,” give compelling cases where large amounts of data yield discoveries and intuitions that surprise even experts. Every enterprise benefits from collecting and ana lyzing its data: Hospitals can spot trends and anomalies in their patient records, search engines can do better ranking and ad placement, and environmental and public health agencies can spot patterns and abnormalities in their data. The list continues, with cybersecurity and computer network intrusion detection; monitoring of the energy consumption of household appliances; pattern analysis in bioinformatics and pharma ceutical data; financial and business intelligence data; spotting trends in blogs, Twitter, and many more. Storage is inexpensive and getting even less so, as are data sensors. Thus, collecting and storing data is easier than ever before.

The problem then becomes *how to analyze* the data. This is exactly the focus of this Third Edition of the book. Jiawei, Micheline, and Jian give encyclopedic coverage of all the related methods, from the classic topics of clustering and classification, to database methods (e.g., association rules, data cubes) to more recent and advanced topics (e.g., SVD/PCA, wavelets, support vector machines).

The exposition is extremely accessible to beginners and advanced readers alike. The book gives the fundamental material first and the more advanced material in follow-up chapters. It also has numerous rhetorical questions, which I found extremely helpful for maintaining focus.

We have used the first two editions as textbooks in data mining courses at Carnegie Mellon and plan to continue to do so with this Third Edition. The new version has significant additions: Notably, it has more than 100 citations to works from 2006 onward, focusing on more recent material such as graphs and social networks, sen sor networks, and outlier detection. This book has a new section for visualization, has expanded outlier detection into a whole chapter, and has separate chapters for advanced methods—for example, pattern mining with top-*k* patterns and more and clustering methods with biclustering and graph clustering.

Overall, it is an excellent book on classic and modern data mining methods, and it is ideal not only for teaching but also as a reference book.

Christos Faloutsos
Carnegie Mellon University

Foreword to Second Edition

We are deluged by data—scientific data, medical data, demographic data, financial data, and marketing data. People have no time to look at this data. Human attention has become the precious resource. So, we must find ways to automatically analyze the data, to automatically classify it, to automatically summarize it, to automatically dis cover and characterize trends in it, and to automatically flag anomalies. This is one of the most active and exciting areas of the database research community. Researchers in areas including statistics, visualization, artificial intelligence, and machine learning are contributing to this field. The breadth of the field makes it difficult to grasp the extraordinary progress over the last few decades.

Six years ago, Jiawei Han's and Micheline Kamber's seminal textbook organized and presented Data Mining. It heralded a golden age of innovation in the field. This revision of their book reflects that progress; more than half of the references and historical notes are to recent work. The field has matured with many new and improved algorithms, and has broadened to include many more datatypes: streams, sequences, graphs, time-series, geospatial, audio, images, and video. We are certainly not at the end of the golden age— indeed research and commercial interest in data mining continues to grow—but we are all fortunate to have this modern compendium.

The book gives quick introductions to database and data mining concepts with particular emphasis on data analysis. It then covers in a chapter-by-chapter tour the concepts and techniques that underlie classification, prediction, association, and clus tering. These topics are presented with examples, a tour of the best algorithms for each problem class, and with pragmatic rules of thumb about when to apply each technique. The Socratic presentation style is both very readable and very informative. I certainly learned a lot from reading the first edition and got re-educated and updated in reading the second edition.

Jiawei Han and Micheline Kamber have been leading contributors to data mining research. This is the text they use with their students to bring them up to speed on the field. The field is evolving very rapidly, but this book is a quick way to learn the basic ideas, and to understand where the field is today. I found it very informative and stimulating, and believe you will too.

Jim Gray
In his memory

Preface

The computerization of our society has substantially enhanced our capabilities for both generating and collecting data from diverse sources. A tremendous amount of data has flooded almost every aspect of our lives. This explosive growth in stored or transient data has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amounts of data into useful information and knowledge. This has led to the generation of a promising and flourishing frontier in computer science called *data mining*, and its various applications. Data mining, also popularly referred to as *knowledge discovery from data (KDD)*, is the automated or con venient extraction of patterns representing knowledge implicitly stored or captured in large databases, data warehouses, the Web, other massive information repositories, or data streams.

This book explores the concepts and techniques of *knowledge discovery* and *data min ing*.As amultidisciplinary field, datamining draws onworkfrom areasincluding statistics, machine learning, pattern recognition, database technology, information retrieval, network science, knowledge-based systems, artificial intelligence, high-performance computing, and data visualization. We focus on issues relating to the feasibility, use fulness, effectiveness, and scalability of techniques for the discovery of patterns hidden in *large data sets*. As a result, this book is not intended as an introduction to statis tics, machine learning, database systems, or other such areas, although we do provide some background knowledge to facilitate the reader’s comprehension of their respective roles in data mining. Rather, the book is a comprehensive introduction to data mining. It is useful for computing science students, application developers, and business professionals, as well as researchers involved in any of the disciplines previously listed.

Data mining emerged during the late 1980s, made great strides during the 1990s, and continues to flourish into the new millennium. This book presents an overall picture of the field, introducing interesting data mining techniques and systems and discussing applications and research directions. An important motivation for writing this book was the need to build an organized framework for the study of data mining—a challenging task, owing to the extensive multidisciplinary nature of this fast-developing field. We hope that this book will encourage people with different backgrounds and experiences to exchange their views regarding data mining so as to contribute toward the further promotion and shaping of this exciting and dynamic field.

Organization of the Book

Since the publication of the first two editions of this book, great progress has been made in the field of data mining. Many new data mining methodologies, systems, and applications have been developed, especially for handling new kinds of data, includ ing information networks, graphs, complex structures, and data streams, as well as text, Web, multimedia, time-series, and spatiotemporal data. Such fast development and rich, new technical contents make it difficult to cover the full spectrum of the field in a single book. Instead of continuously expanding the coverage of this book, we have decided to cover the core material in sufficient scope and depth, and leave the handling of complex data types to a separate forthcoming book.

The third edition substantially revises the first two editions of the book, with numer ous enhancements and a reorganization of the technical contents. The core technical material, which handles mining on general data types, is expanded and substantially enhanced. Several individual chapters for topics from the second edition (e.g., data pre processing, frequent pattern mining, classification, and clustering) are now augmented and each split into two chapters for this new edition. For these topics, one chapter encap sulates the basic concepts and techniques while the other presents advanced concepts and methods.

Chapters from the second edition on mining complex data types (e.g., stream data, sequence data, graph-structured data, social network data, and multirelational data, as well as text, Web, multimedia, and spatiotemporal data) are now reserved for a new book that will be dedicated to *advanced topics in data mining*. Still, to support readers in learning such advanced topics, we have placed an electronic version of the relevant chapters from the second edition onto the book's web site as companion material for the third edition.

The chapters of the third edition are described briefly as follows, with emphasis on the new material.

Chapter 1 provides an *introduction* to the multidisciplinary field of data mining. It discusses the evolutionary path of information technology, which has led to the need for data mining, and the importance of its applications. It examines the data types to be mined, including relational, transactional, and data warehouse data, as well as complex data types such as time-series, sequences, data streams, spatiotemporal data, multimedia data, text data, graphs, social networks, and Web data. The chapter presents a general classification of data mining tasks, based on the kinds of knowledge to be mined, the kinds of technologies used, and the kinds of applications that are targeted. Finally, major challenges in the field are discussed.

Chapter 2 introduces the *general data features*. It first discusses data objects and attribute types and then introduces typical measures for basic statistical data descriptions. It overviews data visualization techniques for various kinds of data. In addition to methods of numeric data visualization, methods for visualizing text, tags, graphs, and multidimensional data are introduced. Chapter 2 also introduces ways to measure similarity and dissimilarity for various kinds of data.

Chapter 3 introduces *techniques for data preprocessing*. It first introduces the concept of data quality and then discusses methods for data cleaning, data integration, data reduction, data transformation, and data discretization.

Chapters 4 and 5 provide a solid introduction to *data warehouses*, *OLAP* (online analytical processing), and *data cube technology*. **Chapter 4** introduces the basic concepts, modeling, design architectures, and general implementations of data warehouses and OLAP, as well as the relationship between data warehousing and other data generalization methods. **Chapter 5** takes an in-depth look at data cube technology, presenting a detailed study of methods of data cube computation, including Star-Cubing and high dimensional OLAP methods. Further explorations of data cube and OLAP technologies are discussed, such as sampling cubes, ranking cubes, prediction cubes, multifeature cubes for complex analysis queries, and discovery-driven cube exploration.

Chapters 6 and 7 present methods for *mining frequent patterns, associations*, and *correlations* in large data sets. **Chapter 6** introduces fundamental concepts, such as market basket analysis, with many techniques for frequent itemset mining presented in an organized way. These range from the basic Apriori algorithm and its variations to more advanced methods that improve efficiency, including the frequent pattern growth approach, frequent pattern mining with vertical data format, and mining closed and max frequent itemsets. The chapter also discusses pattern evaluation methods and introduces measures for mining correlated patterns. **Chapter 7** is on advanced pattern mining methods. It discusses methods for pattern mining in multilevel and multidimensional space, mining rare and negative patterns, mining colossal patterns and high-dimensional data, constraint-based pattern mining, and mining compressed or approximate patterns. It also introduces methods for pattern exploration and application, including semantic annotation of frequent patterns.

Chapters 8 and 9 describe methods for *data classification*. Due to the importance and diversity of classification methods, the contents are partitioned into two chapters. **Chapter 8** introduces basic concepts and methods for classification, including decision tree induction, Bayes classification, and rule-based classification. It also discusses model evaluation and selection methods and methods for improving classification accuracy, including ensemble methods and how to handle imbalanced data. **Chapter 9** discusses advanced methods for classification, including Bayesian belief networks, the neural network technique of backpropagation, support vector machines, classification using frequent patterns, *k*-nearest-neighbor classifiers, case-based reasoning, genetic algo rithms, rough set theory, and fuzzy set approaches. Additional topics include multiclass classification, semi-supervised classification, active learning, and transfer learning.

Cluster analysis forms the topic of Chapters 10 and 11. **Chapter 10** introduces the basic concepts and methods for data clustering, including an overview of basic cluster analysis methods, partitioning methods, hierarchical methods, density-based methods, and grid-based methods. It also introduces methods for the evaluation of clustering. **Chapter 11** discusses advanced methods for clustering, including probabilistic model based clustering, clustering high-dimensional data, clustering graph and network data, and clustering with constraints.

Chapter 12 is dedicated to *outlier detection*. It introduces the basic concepts of out liers and outlier analysis and discusses various outlier detection methods from the view of degree of supervision (i.e., supervised, semi-supervised, and unsupervised meth ods), as well as from the view of approaches (i.e., statistical methods, proximity-based methods, clustering-based methods, and classification-based methods). It also discusses methods for mining contextual and collective outliers, and for outlier detection in high-dimensional data.

Finally, in **Chapter 13**, we discuss *trends, applications*, and *research frontiers* in data mining. We briefly cover mining complex data types, including mining sequence data (e.g., time series, symbolic sequences, and biological sequences), mining graphs and networks, and mining spatial, multimedia, text, and Web data. In-depth treatment of data mining methods for such data is left to a book on advanced topics in data mining, the writing of which is in progress. The chapter then moves ahead to cover other data mining methodologies, including statistical data mining, foundations of data mining, visual and audio data mining, as well as data mining applications. It discusses data mining for financial data analysis, for industries like retail and telecommunication, for use in science and engineering, and for intrusion detection and prevention. It also dis cusses the relationship between data mining and recommender systems. Because data mining is present in many aspects of daily life, we discuss issues regarding data mining and society, including ubiquitous and invisible data mining, as well as privacy, security, and the social impacts of data mining. We conclude our study by looking at data mining trends.

Throughout the text, *italic* font is used to emphasize terms that are defined, while **bold** font is used to highlight or summarize main ideas. Sans serif font is used for reserved words. Bold italic font is used to represent multidimensional quantities.

This book has several strong features that set it apart from other texts on data mining. It presents a very broad yet in-depth coverage of the principles of data mining. The chapters are written to be as self-contained as possible, so they may be read in order of interest by the reader. Advanced chapters offer a larger-scale view and may be considered optional for interested readers. All of the major methods of data mining are presented. The book presents important topics in data mining regarding multidimensional OLAP analysis, which is often overlooked or minimally treated in other data mining books. The book also maintains web sites with a number of online resources to aid instructors, students, and professionals in the field. These are described further in the following.

To the Instructor

This book is designed to give a broad, yet detailed overview of the data mining field. It can be used to teach an introductory course on data mining at an advanced undergrad uate level or at the first-year graduate level. Sample course syllabi are provided on the book's web sites (*www.cs.uiuc.edu/~hanj/bk3* and *www.booksite.mkp.com/datamining3e*) in addition to extensive teaching resources such as lecture slides, instructors' manuals, and reading lists (see p. xxix).

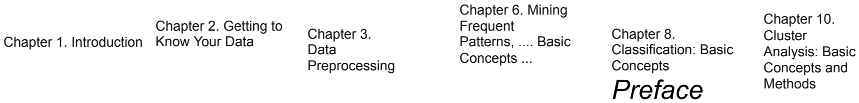
Chapter 1. Introduction
Chapter 2. Getting to Know Your Data
Chapter 3. Data Preprocessing
Chapter 6. Mining Frequent Patterns, ... Basic Concepts ...
Chapter 8. Classification: Basic Concepts
Chapter 10. Cluster Analysis: Basic Concepts and Methods

Figure P.1 A suggested sequence of chapters for a short introductory course.

Depending on the length of the instruction period, the background of students, and your interests, you may select subsets of chapters to teach in various sequential order ings. For example, if you would like to give only a short introduction to students on data mining, you may follow the suggested sequence in Figure P.1. Notice that depending on the need, you can also omit some sections or subsections in a chapter if desired.

Depending on the length of the course and its technical scope, you may choose to selectively add more chapters to this preliminary sequence. For example, instructors who are more interested in advanced classification methods may first add "Chapter 9. Classification: Advanced Methods"; those more interested in pattern mining may choose to include "Chapter 7. Advanced Pattern Mining"; whereas those interested in OLAP and data cube technology may like to add "Chapter 4. Data Warehousing and Online Analytical Processing" and "Chapter 5. Data Cube Technology."

Alternatively, you may choose to teach the whole book in a two-course sequence that covers all of the chapters in the book, plus, when time permits, some advanced topics such as graph and network mining. Material for such advanced topics may be selected from the companion chapters available from the book's web site, accompanied with a set of selected research papers.

Individual chapters in this book can also be used for tutorials or for special topics in related courses, such as machine learning, pattern recognition, data warehousing, and intelligent data analysis.

Each chapter ends with a set of exercises, suitable as assigned homework. The exer cises are either short questions that test basic mastery of the material covered, longer questions that require analytical thinking, or implementation projects. Some exercises can also be used as research discussion topics. The bibliographic notes at the end of each chapter can be used to find the research literature that contains the origin of the concepts and methods presented, in-depth treatment of related topics, and possible extensions.

To the Student

We hope that this textbook will spark your interest in the young yet fast-evolving field of data mining. We have attempted to present the material in a clear manner, with careful explanation of the topics covered. Each chapter ends with a summary describing the main points. We have included many figures and illustrations throughout the text to make the book more enjoyable and reader-friendly. Although this book was designed as a textbook, we have tried to organize it so that it will also be useful to you as a reference

book or handbook, should you later decide to perform in-depth research in the related fields or pursue a career in data mining.

What do you need to know to read this book?

You should have some knowledge of the concepts and terminology associated with statistics, database systems, and machine learning. However, we do try to provide enough background of the basics, so that if you are not so familiar with these fields or your memory is a bit rusty, you will not have trouble following the discussions in the book.

You should have some programming experience. In particular, you should be able to read pseudocode and understand simple data structures such as multidimensional arrays.

To the Professional

This book was designed to cover a wide range of topics in the data mining field. As a result, it is an excellent handbook on the subject. Because each chapter is designed to be as standalone as possible, you can focus on the topics that most interest you. The book can be used by application programmers and information service managers who wish to learn about the key ideas of data mining on their own. The book would also be useful for technical data analysis staff in banking, insurance, medicine, and retailing industries who are interested in applying data mining solutions to their businesses. Moreover, the book may serve as a comprehensive survey of the data mining field, which may also benefit researchers who would like to advance the state-of-the-art in data mining and extend the scope of data mining applications.

The techniques and algorithms presented are of practical utility. Rather than selecting algorithms that perform well on small “toy” data sets, the algorithms described in the book are geared for the discovery of patterns and knowledge hidden in large, real data sets. Algorithms presented in the book are illustrated in pseudocode. The pseudocode is similar to the C programming language, yet is designed so that it should be easy to follow by programmers unfamiliar with C or C++. If you wish to implement any of the algorithms, you should find the translation of our pseudocode into the programming language of your choice to be a fairly straightforward task.

Book Web Sites with Resources

The book has a web site at *www.cs.uiuc.edu/~hanj/bk3* and another with Morgan Kauf mann Publishers at *www.booksite.mkp.com/datamining3e*. These web sites contain many supplemental materials for readers of this book or anyone else with an interest in data mining. The resources include the following:

Slide presentations for each chapter. Lecture notes in Microsoft PowerPoint slides are available for each chapter.

Companion chapters on advanced data mining. Chapters 8 to 10 of the second edition of the book, which cover mining complex data types, are available on the book’s web sites for readers who are interested in learning more about such advanced topics, beyond the themes covered in this book.

Instructors’ manual. This complete set of answers to the exercises in the book is available only to instructors from the publisher’s web site.

Course syllabi and lecture plans. These are given for undergraduate and graduate versions of introductory and advanced courses on data mining, which use the text and slides.

Supplemental reading lists with hyperlinks. Seminal papers for supplemental read ing are organized per chapter.

Links to data mining data sets and software. We provide a set of links to data mining data sets and sites that contain interesting data mining software packages, such as IlliMine from the University of Illinois at Urbana-Champaign (*http://illimine.cs.uiuc.edu*).

Sample assignments, exams, and course projects. A set of sample assignments, exams, and course projects is available to instructors from the publisher’s web site.

Figures from the book. This may help you to make your own slides for your classroom teaching.

Contents of the book in PDF format.

Errata on the different printings of the book. We encourage you to point out any errors in this book. Once the error is confirmed, we will update the errata list and include acknowledgment of your contribution.

Comments or suggestions can be sent to *hanj@cs.uiuc.edu*. We would be happy to hear from you.

This page intentionally left blank

Acknowledgments

Third Edition of the Book

We would like to express our grateful thanks to all of the previous and current mem bers of the Data Mining Group at UIUC, the faculty and students in the Data and Information Systems (DAIS) Laboratory in the Department of Computer Science at the University of Illinois at Urbana-Champaign, and many friends and colleagues, whose constant support and encouragement have made our work on this edition a rewarding experience. We would also like to thank students in CS412 and CS512 classes at UIUC of the 2010–2011 academic year, who carefully went through the early drafts of this book, identified many errors, and suggested various improvements.

We also wish to thank David Bevans and Rick Adams at Morgan Kaufmann Publish ers, for their enthusiasm, patience, and support during our writing of this edition of the book. We thank Marilyn Rash, the Project Manager, and her team members, for keeping us on schedule.

We are also grateful for the invaluable feedback from all of the reviewers. Moreover, we would like to thank U.S. National Science Foundation, NASA, U.S. Air Force Office of Scientific Research, U.S. Army Research Laboratory, and Natural Science and Engineer ing Research Council of Canada (NSERC), as well as IBM Research, Microsoft Research, Google, Yahoo! Research, Boeing, HP Labs, and other industry research labs for their support of our research in the form of research grants, contracts, and gifts. Such research support deepens our understanding of the subjects discussed in this book. Finally, we thank our families for their wholehearted support throughout this project.

Second Edition of the Book

We would like to express our grateful thanks to all of the previous and current mem bers of the Data Mining Group at UIUC, the faculty and students in the Data and

Information Systems (DAIS) Laboratory in the Department of Computer Science at the University of Illinois at Urbana-Champaign, and many friends and colleagues, whose constant support and encouragement have made our work on this edition a rewarding experience. These include Gul Agha, Rakesh Agrawal, Loretta Auvil, Peter Bajcsy, Geneva Belford, Deng Cai, Y. Dora Cai, Roy Cambell, Kevin C.-C. Chang, Surajit Chaudhuri, Chen Chen, Yixin Chen, Yuguo Chen, Hong Cheng, David Cheung, Shengnan Cong, Gerald DeJong, AnHai Doan, Guozhu Dong, Charios Ermopoulos, Martin Ester, Chris tos Faloutsos, Wei Fan, Jack C. Feng, Ada Fu, Michael Garland, Johannes Gehrke, Hector Gonzalez, Mehdi Harandi, Thomas Huang, Wen Jin, Chulyun Kim, Sangkyum Kim, Won Kim, Won-Young Kim, David Kuck, Young-Koo Lee, Harris Lewin, Xiaolei Li, Yifan Li, Chao Liu, Han Liu, Huan Liu, Hongyan Liu, Lei Liu, Ying Lu, Klara Nahrstedt, David Padua, Jian Pei, Lenny Pitt, Daniel Reed, Dan Roth, Bruce Schatz, Zheng Shao, Marc Snir, Zhaohui Tang, Bhavani M. Thuraisingham, Josep Torrellas, Peter Tzvetkov, Benjamin W. Wah, Haixun Wang, Jianyong Wang, Ke Wang, Muyuan Wang, Wei Wang, Michael Welge, Marianne Winslett, Ouri Wolfson, Andrew Wu, Tianyi Wu, Dong Xin, Xifeng Yan, Jiong Yang, Xiaoxin Yin, Hwanjo Yu, Jeffrey X. Yu, Philip S. Yu, Maria Zemankova, ChengXiang Zhai, Yuanyuan Zhou, and Wei Zou.

Deng Cai and ChengXiang Zhai have contributed to the text mining and Web mining sections, Xifeng Yan to the graph mining section, and Xiaoxin Yin to the multirela tional data mining section. Hong Cheng, Charios Ermopoulos, Hector Gonzalez, David J. Hill, Chulyun Kim, Sangkyum Kim, Chao Liu, Hongyan Liu, Kasif Manzoor, Tianyi Wu, Xifeng Yan, and Xiaoxin Yin have contributed to the proofreading of the individual chapters of the manuscript.

We also wish to thank Diane Cerra, our Publisher at Morgan Kaufmann Publishers, for her constant enthusiasm, patience, and support during our writing of this book. We are indebted to Alan Rose, the book Production Project Manager, for his tireless and ever-prompt communications with us to sort out all details of the production process. We are grateful for the invaluable feedback from all of the reviewers. Finally, we thank our families for their wholehearted support throughout this project.

First Edition of the Book

We would like to express our sincere thanks to all those who have worked or are currently working with us on data mining–related research and/or the DBMiner project, or have provided us with various support in data mining. These include Rakesh Agrawal, Stella Atkins, Yvan Bedard, Binay Bhattacharya, (Yandong) Dora Cai, Nick Cercone, Surajit Chaudhuri, Sonny H. S. Chee, Jianping Chen, Ming-Syan Chen, Qing Chen, Qiming Chen, Shan Cheng, David Cheung, Shi Cong, Son Dao, Umeshwar Dayal, James Delgrande, Guozhu Dong, Carole Edwards, Max Egenhofer, Martin Ester, Usama Fayyad, Ling Feng, Ada Fu, Yongjian Fu, Daphne Gelbart, Randy Goebel, Jim Gray, Robert Grossman, Wan Gong, Yike Guo, Eli Hagen, Howard Hamilton, Jing He, Larry Henschen, Jean Hou, Mei-Chun Hsu, Kan Hu, Haiming Huang, Yue Huang, Julia Itske vitch, Wen Jin, Tiko Kameda, Hiroyuki Kawano, Rizwan Kheraj, Eddie Kim, Won Kim, Krzysztof Koperski, Hans-Peter Kriegel, Vipin Kumar, Laks V. S. Lakshmanan, Joyce Man Lam, James Lau, Deyi Li, George (Wenmin) Li, Jin Li, Ze-Nian Li, Nancy Liao, Gang Liu, Junqiang Liu, Ling Liu, Alan (Yijun) Lu, Hongjun Lu, Tong Lu, Wei Lu, Xuebin Lu, Wo-Shun Luk, Heikki Mannila, Runying Mao, Abhay Mehta, Gabor Melli, Alberto Mendelzon, Tim Merrett, Harvey Miller, Drew Miners, Behzad Mortazavi-Asl, Richard Muntz, Raymond T. Ng, Vicent Ng, Shojiro Nishio, Beng-Chin Ooi, Tamer Ozsu, Jian Pei, Gregory Piatetsky-Shapiro, Helen Pinto, Fred Popowich, Amynmohamed Rajan, Peter Scheuermann, Shashi Shekhar, Wei-Min Shen, Avi Silberschatz, Evangelos Simoudis, Nebojsa Stefanovic, Yin Jenny Tam, Simon Tang, Zhaohui Tang, Dick Tsur, Anthony K. H. Tung, Ke Wang, Wei Wang, Zhaoxia Wang, Tony Wind, Lara Winstone, Ju Wu, Betty (Bin) Xia, Cindy M. Xin, Xiaowei Xu, Qiang Yang, Yiwen Yin, Clement Yu, Jeffrey Yu, Philip S. Yu, Osmar R. Zaiane, Carlo Zaniolo, Shuhua Zhang, Zhong Zhang, Yvonne Zheng, Xiaofang Zhou, and Hua Zhu.

We are also grateful to Jean Hou, Helen Pinto, Lara Winstone, and Hua Zhu for their help with some of the original figures in this book, and to Eugene Belchev for his careful proofreading of each chapter.

We also wish to thank Diane Cerra, our Executive Editor at Morgan Kaufmann Pub lishers, for her enthusiasm, patience, and support during our writing of this book, as well as Howard Severson, our Production Editor, and his staff for their conscientious efforts regarding production. We are indebted to all of the reviewers for their invaluable feedback. Finally, we thank our families for their wholehearted support throughout this project.

This page intentionally left blank

About the Authors

Jiawei Han is a Bliss Professor of Engineering in the Department of Computer Science at the University of Illinois at Urbana-Champaign. He has received numerous awards for his contributions on research into knowledge discovery and data mining, including ACM SIGKDD Innovation Award (2004), IEEE Computer Society Technical Achieve ment Award (2005), and IEEE W. Wallace McDowell Award (2009). He is a Fellow of ACM and IEEE. He served as founding Editor-in-Chief of *ACM Transactions on Know ledge Discovery from Data* (2006–2011) and as an editorial board member of several jour nals, including *IEEE Transactions on Knowledge and Data Engineering* and *Data*

Mining and Knowledge Discovery.

Micheline Kamber has a master's degree in computer science (specializing in artificial intelligence) from Concordia University in Montreal, Quebec. She was an NSERC Scholar and has worked as a researcher at McGill University, Simon Fraser University, and in Switzerland. Her background in data mining and passion for writing in easy to-understand terms help make this text a favorite of professionals, instructors, and students.

Jian Pei is currently an associate professor at the School of Computing Science, Simon Fraser University in British Columbia. He received a Ph.D. degree in computing science from Simon Fraser University in 2002 under Dr. Jiawei Han's supervision. He has published prolifically in the premier academic forums on data mining, databases, Web searching, and information retrieval and actively served the academic community. His publications have received thousands of citations and several prestigious awards. He is an associate editor of several data mining and data analytics journals.

This page intentionally left blank

Introduction

This book is an introduction to the young and fast-growing field of *data mining* (also known as *knowledge discovery from data*, or *KDD* for short). The book focuses on fundamental data mining concepts and techniques for discovering interesting patterns from data in various applications. In particular, we emphasize prominent techniques for developing effective, efficient, and scalable data mining tools.

This chapter is organized as follows. In Section 1.1, you will learn why data mining is in high demand and how it is part of the natural evolution of information technology. Section 1.2 defines data mining with respect to the knowledge discovery process. Next, you will learn about data mining from many aspects, such as the kinds of data that can be mined (Section 1.3), the kinds of knowledge to be mined (Section 1.4), the kinds of technologies to be used (Section 1.5), and targeted applications (Section 1.6). In this way, you will gain a multidimensional view of data mining. Finally, Section 1.7 outlines major data mining research and development issues.

1.1 Why Data Mining?

Necessity, who is the mother of invention. – Plato

We live in a world where vast amounts of data are collected daily. Analyzing such data is an important need. Section 1.1.1 looks at how data mining can meet this need by providing tools to discover knowledge from data. In Section 1.1.2, we observe how data mining can be viewed as a result of the natural evolution of information technology.

1.1.1 Moving toward the Information Age

"*We are living in the information age*" is a popular saying; however, *we are actually living in the data age*. Terabytes or petabytes[1] of data pour into our computer networks, the World Wide Web (WWW), and various data storage devices every day from business,

[1] A petabyte is a unit of information or computer storage equal to 1 quadrillion bytes, or a thousand terabytes, or 1 million gigabytes.

society, science and engineering, medicine, and almost every other aspect of daily life. This explosive growth of available data volume is a result of the computerization of our society and the fast development of powerful data collection and storage tools. Businesses worldwide generate gigantic data sets, including sales transactions, stock trading records, product descriptions, sales promotions, company profiles and perfor mance, and customer feedback. For example, large stores, such as Wal-Mart, handle hundreds of millions of transactions per week at thousands of branches around the world. Scientific and engineering practices generate high orders of petabytes of data in a continuous manner, from remote sensing, process measuring, scientific experiments, system performance, engineering observations, and environment surveillance.

Global backbone telecommunication networks carry tens of petabytes of data traffic every day. The medical and health industry generates tremendous amounts of data from medical records, patient monitoring, and medical imaging. Billions of Web searches supported by search engines process tens of petabytes of data daily. Communities and social media have become increasingly important data sources, producing digital pic tures and videos, blogs, Web communities, and various kinds of social networks. The list of sources that generate huge amounts of data is endless.

This explosively growing, widely available, and gigantic body of data makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining. The field is young, dynamic, and promising. Data mining has and will continue to make great strides in our journey from the data age toward the coming information age.

Example 1.1 Data mining turns a large collection of data into knowledge. A search engine (e.g., Google) receives hundreds of millions of queries every day. Each query can be viewed as a transaction where the user describes her or his information need. What novel and useful knowledge can a search engine learn from such a huge collection of queries col lected from users over time? Interestingly, some patterns found in user search queries can disclose invaluable knowledge that cannot be obtained by reading individual data items alone. For example, Google's *Flu Trends* uses specific search terms as indicators of flu activity. It found a close relationship between the number of people who search for flu-related information and the number of people who actually have flu symptoms. A pattern emerges when all of the search queries related to flu are aggregated. Using aggre gated Google search data, *Flu Trends* can estimate flu activity up to two weeks faster than traditional systems can.[2] This example shows how data mining can turn a large collection of data into knowledge that can help meet a current global challenge.

1.1.2 Data Mining as the Evolution of Information Technology

Data mining can be viewed as a result of the natural evolution of information tech nology. The database and data management industry evolved in the development of

[2] This is reported in [GMP$^+$09].

Data Collection and Database Creation
(1960s and earlier)
Primitive file processing

Database Management Systems
(1970s to early 1980s)
Hierarchical and network database systems
Relational database systems
Data modeling: entity-relationship models, etc.
Indexing and accessing methods
Query languages: SQL, etc.
User interfaces, forms, and reports
Query processing and optimization
Transactions, concurrency control, and recovery
Online transaction processing (OLTP)

Advanced Database Systems
(mid-1980sto present)
Advanced data models: extended-relational, object relational, deductive, etc.
Managing complex data: spatial, temporal, multimedia, sequence and structured, scientific, engineering, moving objects, etc.
Data streams and cyber-physical data systems
Web-based databases (XML, semantic web)
Managing uncertain data and data cleaning
Integration of heterogeneous sources Text database systems and integration with information retrieval
Extremely large data management
Database system tuning and adaptive systems
Advanced queries: ranking, skyline, etc.
Cloud computing and parallel data processing
Issues of data privacy and security

Advanced Data Analysis
(late- 1980sto present)

Data warehouse and OLAP

Data mining and knowledge discovery: classification, clustering, outlier analysis, association and correlation, comparative summary, discrimination analysis, pattern discovery, trend and deviation analysis, etc.

Mining complex types of data: streams, sequence, text, spatial, temporal, multimedia, Web, networks, etc.

Data mining applications: business, society, retail, banking, telecommunications, science and engineering, blogs, daily life, etc. Data mining and society: invisible data mining, privacy-preserving data mining, mining social and information networks, recommendersystems, etc.

Future Generation of Information Systems
(Present to future)

Figure 1.1 The evolution of database system technology.

several critical functionalities (Figure 1.1): *data collection and database creation, data management* (including data storage and retrieval and database transaction processing), and *advanced data analysis* (involving data warehousing and data mining). The early development of data collection and database creation mechanisms served as a prerequi site for the later development of effective mechanisms for data storage and retrieval, as well as query and transaction processing. Nowadays numerous database systems offer query and transaction processing as common practice. Advanced data analysis has naturally become the next step.

Since the 1960s, database and information technology has evolved systematically from primitive file processing systems to sophisticated and powerful database systems. The research and development in database systems since the 1970s progressed from early hierarchical and network database systems to relational database systems (where data are stored in relational table structures; see Section 1.3.1), data modeling tools, and indexing and accessing methods. In addition, users gained convenient and flexible data access through query languages, user interfaces, query optimization, and transac tion management. Efficient methods for online transaction processing (OLTP), where a query is viewed as a read-only transaction, contributed substantially to the evolution and wide acceptance of relational technology as a major tool for efficient storage, retrieval, and management of large amounts of data.

After the establishment of database management systems, database technology moved toward the development of *advanced database systems*, *data warehousing*, and *data mining* for advanced data analysis and *web-based databases*. Advanced database systems, for example, resulted from an upsurge of research from the mid-1980s onward. These systems incorporate new and powerful data models such as extended-relational, object-oriented, object-relational, and deductive models. Application-oriented database systems have flourished, including spatial, temporal, multimedia, active, stream and sensor, scientific and engineering databases, knowledge bases, and office information bases. Issues related to the distribution, diversification, and sharing of data have been studied extensively.

Advanced data analysis sprang up from the late 1980s onward. The steady and dazzling progress of computer hardware technology in the past three decades led to large supplies of powerful and affordable computers, data collection equipment, and storage media. This technology provides a great boost to the database and information industry, and it enables a huge number of databases and information repositories to be available for transaction management, information retrieval, and data analysis. Data can now be stored in many different kinds of databases and information repositories.

One emerging data repository architecture is the **data warehouse** (Section 1.3.2). This is a repository of multiple heterogeneous data sources organized under a unified schema at a single site to facilitate management decision making. Data warehouse technology includes data cleaning, data integration, and online analytical processing (OLAP)—that is, analysis techniques with functionalities such as summarization, con solidation, and aggregation, as well as the ability to view information from different angles. Although OLAP tools support multidimensional analysis and decision making, additional data analysis tools are required for in-depth analysis—for example, data min ing tools that provide data classification, clustering, outlier/anomaly detection, and the characterization of changes in data over time.

Huge volumes of data have been accumulated beyond databases and data ware houses. During the 1990s, the World Wide Web and web-based databases (e.g., XML databases) began to appear. Internet-based global information bases, such as the WWW and various kinds of interconnected, heterogeneous databases, have emerged and play a vital role in the information industry. The effective and efficient analysis of data from such different forms of data by integration of information retrieval, data mining, and information network analysis technologies is a challenging task.

How can I analyze these data?

Figure 1.2 The world is data rich but information poor.

In summary, the abundance of data, coupled with the need for powerful data analysis tools, has been described as a *data rich but information poor* situation (Figure 1.2). The fast-growing, tremendous amount of data, collected and stored in large and numerous data repositories, has far exceeded our human ability for comprehension without power ful tools. As a result, data collected in large data repositories become “data tombs”—data archives that are seldom visited. Consequently, important decisions are often made based not on the information-rich data stored in data repositories but rather on a decision maker’s intuition, simply because the decision maker does not have the tools to extract the valuable knowledge embedded in the vast amounts of data. Efforts have been made to develop expert system and knowledge-based technologies, which typically rely on users or domain experts to *manually* input knowledge into knowledge bases. Unfortunately, however, the manual knowledge input procedure is prone to biases and errors and is extremely costly and time consuming. The widening gap between data and information calls for the systematic development of *data mining tools* that can turn data tombs into “golden nuggets” of knowledge.

1.2 What Is Data Mining?

It is no surprise that data mining, as a truly interdisciplinary subject, can be defined in many different ways. Even the term *data mining* does not really present all the major components in the picture. To refer to the mining of gold from rocks or sand, we say *gold mining* instead of rock or sand mining. Analogously, data mining should have been more

Knowledge

Figure 1.3 Data mining—searching for knowledge (interesting patterns) in data.

appropriately named "knowledge mining from data," which is unfortunately somewhat long. However, the shorter term, *knowledge mining* may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material (Figure 1.3). Thus, such a misnomer carrying both "data" and "mining" became a pop ular choice. In addition, many other terms have a similar meaning to data mining—for example, *knowledge mining from data*, *knowledge extraction*, *data/pattern analysis*, *data archaeology*, and *data dredging*.

Many people treat data mining as a synonym for another popularly used term, **knowledge discovery from data**, or **KDD**, while others view data mining as merely an essential step in the process of knowledge discovery. The knowledge discovery process is shown in Figure 1.4 as an iterative sequence of the following steps:

1. **Data cleaning** (to remove noise and inconsistent data)
2. **Data integration** (where multiple data sources may be combined)[3]

[3] A popular trend in the information industry is to perform data cleaning and data integration as a preprocessing step, where the resulting data are stored in a data warehouse.

Data mining

Selection and transformation

Data Warehouse

Knowledge

Evaluation and presentation

Cleaning and integration

Patterns

Figure 1.4 Data mining as a step in the process of knowledge discovery.

3. **Data selection** (where data relevant to the analysis task are retrieved from the database)
4. **Data transformation** (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)[4]
5. **Data mining** (an essential process where intelligent methods are applied to extract data patterns)
6. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on *interestingness measures*—see Section 1.4.6)
7. **Knowledge presentation** (where visualization and knowledge representation tech niques are used to present mined knowledge to users)

Steps 1 through 4 are different forms of data preprocessing, where data are prepared for mining. The data mining step may interact with the user or a knowledge base. The interesting patterns are presented to the user and may be stored as new knowledge in the knowledge base.

The preceding view shows data mining as one step in the knowledge discovery pro cess, albeit an essential one because it uncovers hidden patterns for evaluation. However, in industry, in media, and in the research milieu, the term *data mining* is often used to refer to the entire knowledge discovery process (perhaps because the term is shorter than *knowledge discovery from data*). Therefore, we adopt a broad view of data min ing functionality: **Data mining** is the *process* of discovering interesting patterns and knowledge from *large* amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically.

1.3 What Kinds of Data Can Be Mined?

As a general technology, data mining can be applied to any kind of data as long as the data are meaningful for a target application. The most basic forms of data for mining applications are database data (Section 1.3.1), data warehouse data (Section 1.3.2), and transactional data (Section 1.3.3). The concepts and techniques presented in this book focus on such data. Data mining can also be applied to other forms of data (e.g., data streams, ordered/sequence data, graph or networked data, spatial data, text data, multimedia data, and the WWW). We present an overview of such data in Section 1.3.4. Techniques for mining of these kinds of data are briefly introduced in Chapter 13. In depth treatment is considered an advanced topic. Data mining will certainly continue to embrace new data types as they emerge.

[4] Sometimes data transformation and consolidation are performed before the data selection process, particularly in the case of data warehousing. *Data reduction* may also be performed to obtain a smaller representation of the original data without sacrificing its integrity.

1.3.1 Database Data

A database system, also called a **database management system** (**DBMS**), consists of a collection of interrelated data, known as a **database**, and a set of software programs to manage and access the data. The software programs provide mechanisms for defining database structures and data storage; for specifying and managing concurrent, shared, or distributed data access; and for ensuring consistency and security of the information stored despite system crashes or attempts at unauthorized access.

A **relational database** is a collection of **tables**, each of which is assigned a unique name. Each table consists of a set of **attributes** (*columns* or *fields*) and usually stores a large set of **tuples** (*records* or *rows*). Each tuple in a relational table represents an object identified by a unique *key* and described by a set of attribute values. A semantic data model, such as an **entity-relationship (ER)** data model, is often constructed for relational databases. An ER data model represents the database as a set of entities and their relationships.

Example 1.2 A relational database for *AllElectronics*. The fictitious *AllElectronics* store is used to illustrate concepts throughout this book. The company is described by the following relation tables: *customer, item, employee*, and *branch*. The headers of the tables described here are shown in Figure 1.5. (A header is also called the *schema* of a relation.)

The relation *customer* consists of a set of attributes describing the customer infor mation, including a unique customer identity number (*cust ID*), customer name, address, age, occupation, annual income, credit information, and category.

Similarly, each of the relations *item*, *employee*, and *branch* consists of a set of attri butes describing the properties of these entities.

Tables can also be used to represent the relationships between or among multiple entities. In our example, these include *purchases* (customer purchases items, creating a sales transaction handled by an employee), *items sold* (lists items sold in a given transaction), and *works at* (employee works at a branch of *AllElectronics*).

customer (cust ID, name, address, age, occupation, annual income, *credit information, category*, . . .*)*
item (item ID, brand, category, type, price, place made, supplier, cost, . . .*)*
employee (empl ID, name, category, group, salary, commission, . . .*)*
branch (branch ID, name, address, . . .*)*
purchases (trans ID, cust ID, empl ID, date, time, method paid, amount)
items sold (trans ID, item ID, qty)
works at (empl ID, branch ID)

Figure 1.5 Relational schema for a relational database, *AllElectronics*.

Relational data can be accessed by **database queries** written in a relational query language (e.g., SQL) or with the assistance of graphical user interfaces. A given query is transformed into a set of relational operations, such as join, selection, and projection, and is then optimized for efficient processing. A query allows retrieval of specified sub sets of the data. Suppose that your job is to analyze the *AllElectronics* data. Through the use of relational queries, you can ask things like, *“Show me a list of all items that were sold in the last quarter.”* Relational languages also use aggregate functions such as sum, avg (average), count, max (maximum), and min (minimum). Using aggregates allows you to ask: *“Show me the total sales of the last month, grouped by branch,”* or *“How many sales transactions occurred in the month of December?”* or *“Which salesperson had the highest sales?”*

When **mining relational databases**, we can go further by *searching for trends* or *data patterns*. For example, data mining systems can analyze customer data to predict the credit risk of new customers based on their income, age, and previous credit information. Data mining systems may also detect deviations—that is, items with sales that are far from those expected in comparison with the previous year. Such deviations can then be further investigated. For example, data mining may discover that there has been a change in packaging of an item or a significant increase in price.

Relational databases are one of the most commonly available and richest information repositories, and thus they are a major data form in the study of data mining.

1.3.2 Data Warehouses

Suppose that *AllElectronics* is a successful international company with branches around the world. Each branch has its own set of databases. The president of *AllElectronics* has asked you to provide an analysis of the company's sales per item type per branch for the third quarter. This is a difficult task, particularly since the relevant data are spread out over several databases physically located at numerous sites.

If *AllElectronics* had a data warehouse, this task would be easy. A **data warehouse** is a repository of information collected from multiple sources, stored under a unified schema, and usually residing at a single site. Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and peri odic data refreshing. This process is discussed in Chapters 3 and 4. Figure 1.6 shows the typical framework for construction and use of a data warehouse for *AllElectronics*.

To facilitate decision making, the data in a data warehouse are organized around *major subjects* (e.g., customer, item, supplier, and activity). The data are stored to pro vide information from a *historical perspective*, such as in the past 6 to 12 months, and are typically *summarized*. For example, rather than storing the details of each sales transac tion, the data warehouse may store a summary of the transactions per item type for each store or, summarized to a higher level, for each sales region.

A data warehouse is usually modeled by a multidimensional data structure, called a **data cube**, in which each **dimension** corresponds to an attribute or a set of attributes in the schema, and each **cell** stores the value of some aggregate measure such as *count*

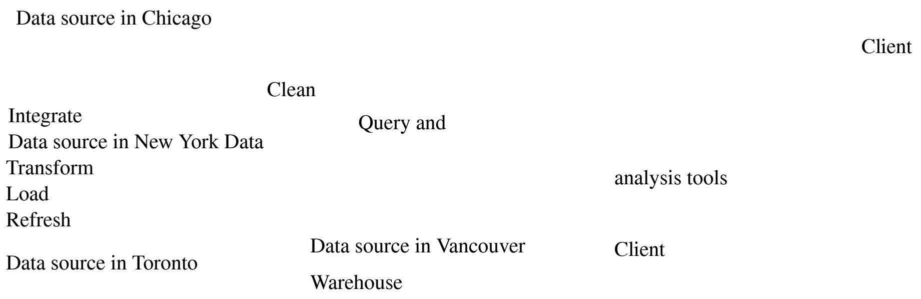

Figure 1.6 Typical framework of a data warehouse for *AllElectronics*.

or *sum(sales amount)*. A data cube provides a multidimensional view of data and allows the precomputation and fast access of summarized data.

Example 1.3 A data cube for *AllElectronics*. A data cube for summarized sales data of *AllElectronics* is presented in Figure 1.7(a). The cube has three dimensions: *address* (with city values *Chicago, New York, Toronto, Vancouver*), *time* (with quarter values *Q1, Q2, Q3, Q4*), and *item*(with item type values *home entertainment, computer, phone, security*). The aggregate value stored in each cell of the cube is*sales amount* (in thousands). For example, the total sales for the first quarter, *Q1*, for the items related to security systems in Vancouver is $400,000, as stored in cellh*Vancouver, Q1, security*i. Additional cubes may be used to store aggregate sumsovereachdimension,corresponding to theaggregatevaluesobtainedusing different SQL group-bys (e.g., the total sales amount per city and quarter, or per city and item, or per quarter and item, or per each individual dimension).

By providing multidimensional data views and the precomputation of summarized data, data warehouse systems can provide inherent support for OLAP. Online analyti cal processing operations make use of background knowledge regarding the domain of the data being studied to allow the presentation of data at *different levels of abstraction*. Such operations accommodate different user viewpoints. Examples of OLAP opera tions include **drill-down** and **roll-up**, which allow the user to view the data at differing degrees of summarization, as illustrated in Figure 1.7(b). For instance, we can drill down on sales data summarized by *quarter* to see data summarized by *month*. Sim ilarly, we can roll up on sales data summarized by *city* to view data summarized by *country*.

Although data warehouse tools help support data analysis, additional tools for data mining are often needed for in-depth analysis. **Multidimensional data mining** (also called **exploratory multidimensional data mining**) performs data mining in

440

address **(cities)**

Chicago

New York

1560

Toronto

Vancouver

395

<Vancouver,

security

home entertainm ent

phone

Q1 605 825 14 400 Q2

Q3

Q4

Q1, security>

computer

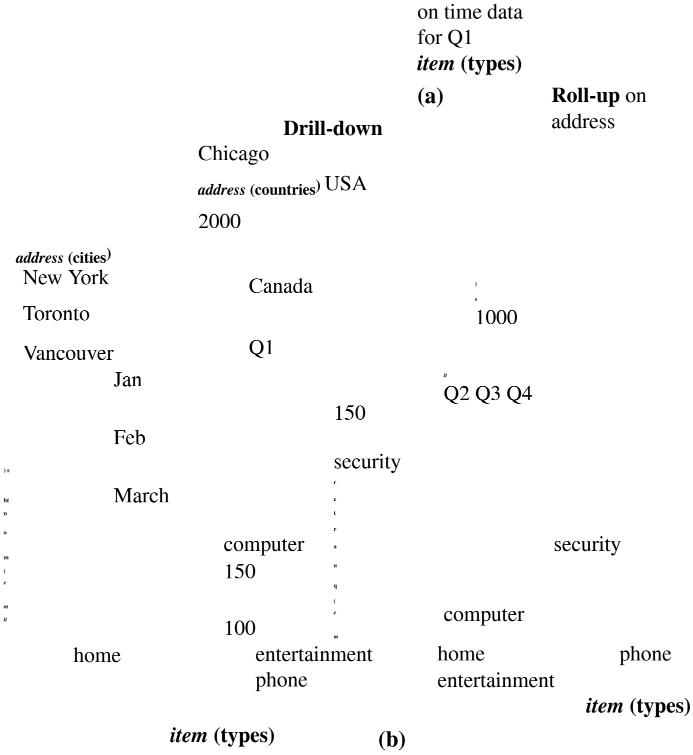

Figure 1.7 A multidimensional data cube, commonly used for data warehousing, (a) showing summarized data for *AllElectronics* and (b) showing summarized data resulting from drill-down and roll-up operations on the cube in (a). For improved readability, only some of the cube cell values are shown.

multidimensional space in an OLAP style. That is, it allows the exploration of multiple combinations of dimensions at varying levels of granularity in data mining, and thus has greater potential for discovering interesting patterns representing knowledge. An overview of data warehouse and OLAP technology is provided in Chapter 4. Advanced issues regarding data cube computation and multidimensional data mining are discussed in Chapter 5.

1.3.3 Transactional Data

In general, each record in a **transactional database** captures a transaction, such as a customer's purchase, a flight booking, or a user's clicks on a web page. A transaction typically includes a unique transaction identity number (*trans_ID*) and a list of the **items** making up the transaction, such as the items purchased in the transaction. A transactional database may have additional tables, which contain other information related to the transactions, such as item description, information about the salesperson or the branch, and so on.

Example 1.4 A transactional database for *AllElectronics*. Transactions can be stored in a table, with one record per transaction. A fragment of a transactional database for *AllElectronics* is shown in Figure 1.8. From the relational database point of view, the *sales* table in the figure is a nested relation because the attribute *list of item IDs* contains a set of *items*. Because most relational database systems do not support nested relational structures, the transactional database is usually either stored in a flat file in a format similar to the table in Figure 1.8 or unfolded into a standard relation in a format similar to the *items sold* table in Figure 1.5.

As an analyst of *AllElectronics*, you may ask,*"Which items sold well together?"* This kind of *market basket data analysis* would enable you to bundle groups of items together as a strategy for boosting sales. For example, given the knowledge that printers are commonly purchased together with computers, you could offer certain printers at a steep discount (or even for free) to customers buying selected computers, in the hopes of selling more computers (which are often more expensive than printers). A tradi tional database system is not able to perform market basket data analysis. Fortunately, data mining on transactional data can do so by mining *frequent itemsets*, that is, sets of items that are frequently sold together. The mining of such frequent patterns from transactional data is discussed in Chapters 6 and 7.

trans ID	*list of item IDs*
T100	I1, I3, I8, I16
T200	I2, I8
...	...

Figure 1.8 Fragment of a transactional database for sales at *AllElectronics*.

1.3.4 Other Kinds of Data

Besides relational database data, data warehouse data, and transaction data, there are many other kinds of data that have versatile forms and structures and rather different semantic meanings. Such kinds of data can be seen in many applications: time-related or sequence data (e.g., historical records, stock exchange data, and time-series and bio logical sequence data), data streams (e.g., video surveillance and sensor data, which are continuously transmitted), spatial data (e.g., maps), engineering design data (e.g., the design of buildings, system components, or integrated circuits), hypertext and multi media data (including text, image, video, and audio data), graph and networked data (e.g., social and information networks), and the Web (a huge, widely distributed infor mation repository made available by the Internet).

These applications bring about new challenges, like how to handle data carrying special structures (e.g., sequences, trees, graphs, and networks) and specific semantics (such as ordering, image, audio and video contents, and connectivity), and how to mine patterns that carry rich structures and semantics.

Various kinds of knowledge can be mined from these kinds of data. Here, we list just a few. Regarding temporal data, for instance, we can mine banking data for chang ing trends, which may aid in the scheduling of bank tellers according to the volume of customer traffic. Stock exchange data can be mined to uncover trends that could help you plan investment strategies (e.g., the best time to purchase *AllElectronics* stock). We could mine computer network data streams to detect intrusions based on the anomaly of message flows, which may be discovered by clustering, dynamic construction of stream models or by comparing the current frequent patterns with those at a previous time. With spatial data, we may look for patterns that describe changes in metropolitan poverty rates based on city distances from major highways. The relationships among a set of spatial objects can be examined in order to discover which subsets of objects are spatially autocorrelated or associated. By mining text data, such as literature on data mining from the past ten years, we can identify the evolution of hot topics in the field. By mining user comments on products (which are often submitted as short text messages), we can assess customer sentiments and understand how well a product is embraced by a market. From multimedia data, we can mine images to identify objects and classify them by assigning semantic labels or tags. By mining video data of a hockey game, we can detect video sequences corresponding to goals. *Web mining* can help us learn about the distribution of information on the WWW in general, characterize and classify web pages, and uncover web dynamics and the association and other relationships among different web pages, users, communities, and web-based activities.

It is important to keep in mind that, in many applications, multiple types of data are present. For example, in web mining, there often exist text data and multimedia data (e.g., pictures and videos) on web pages, graph data like web graphs, and map data on some web sites. In bioinformatics, genomic sequences, biological networks, and 3-D spatial structures of genomes may coexist for certain biological objects. Mining multiple data sources of complex data often leads to fruitful findings due to the mutual enhancement and consolidation of such multiple sources. On the other hand, it is also challenging because of the difficulties in data cleaning and data integration, as well as the complex interactions among the multiple sources of such data.

While such data require sophisticated facilities for efficient storage, retrieval, and updating, they also provide fertile ground and raise challenging research and imple mentation issues for data mining. Data mining on such data is an advanced topic. The methods involved are extensions of the basic techniques presented in this book.

1.4 What Kinds of Patterns Can Be Mined?

We have observed various types of data and information repositories on which data mining can be performed. Let us now examine the kinds of patterns that can be mined. There are a number of ***data mining functionalities***. These include characterization and discrimination (Section 1.4.1); the mining of frequent patterns, associations, and correlations (Section 1.4.2); classification and regression (Section 1.4.3); clustering anal ysis (Section 1.4.4); and outlier analysis (Section 1.4.5). Data mining functionalities are used to specify the kinds of patterns to be found in data mining tasks. In general, such tasks can be classified into two categories: **descriptive** and **predictive**. Descriptive min ing tasks characterize properties of the data in a target data set. Predictive mining tasks perform induction on the current data in order to make predictions.

Data mining functionalities, and the kinds of patterns they can discover, are described below. In addition, Section 1.4.6 looks at what makes a pattern interesting. Interesting patterns represent *knowledge*.

1.4.1 Class/Concept Description: Characterization and Discrimination

Data entries can be associated with classes or concepts. For example, in the *AllElectronics* store, classes of items for sale include *computers* and *printers*, and concepts of customers include *bigSpenders* and *budgetSpenders*. It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called **class/concept descriptions**. These descriptions can be derived using (1) *data characterization*, by summarizing the data of the class under study (often called the **target class**) in general terms, or (2) *data discrimination*, by comparison of the target class with one or a set of comparative classes (often called the **contrasting classes**), or (3) both data characterization and discrimination.

Data characterization is a summarization of the general characteristics or features of a target class of data. The data corresponding to the user-specified class are typically collected by a query. For example, to study the characteristics of software products with sales that increased by 10% in the previous year, the data related to such products can be collected by executing an SQL query on the sales database.

There are several methods for effective data summarization and characterization. Simple data summaries based on statistical measures and plots are described in Chapter 2. The data cube-based OLAP roll-up operation (Section 1.3.2) can be used to perform user-controlled data summarization along a specified dimension. This pro cess is further detailed in Chapters 4 and 5, which discuss data warehousing. An *attribute-oriented induction* technique can be used to perform data generalization and characterization without step-by-step user interaction. This technique is also described in

Chapter 4.

The output of data characterization can be presented in various forms. Examples include **pie charts**, **bar charts**, **curves**, **multidimensional data cubes**, and **multidimen sional tables**, including crosstabs. The resulting descriptions can also be presented as **generalized relations** or in rule form (called **characteristic rules**).

Example 1.5 Data characterization. A customer relationship manager at *AllElectronics* may order the following data mining task: *Summarize the characteristics of customers who spend more than $5000 a year at AllElectronics*. The result is a general profile of these customers, such as that they are 40 to 50 years old, employed, and have excellent credit ratings. The data mining system should allow the customer relationship manager to drill down on any dimension, such as on *occupation* to view these customers according to their type of employment.

Data discrimination is a comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes. The target and contrasting classes can be specified by a user, and the corresponding data objects can be retrieved through database queries. For example, a user may want to compare the general features of software products with sales that increased by 10% last year against those with sales that decreased by at least 30% during the same period. The methods used for data discrimination are similar to those used for data characterization.

"How are discrimination descriptions output?" The forms of output presentation are similar to those for characteristic descriptions, although discrimination descrip tions should include comparative measures that help to distinguish between the target and contrasting classes. Discrimination descriptions expressed in the form of rules are referred to as **discriminant rules**.

Example 1.6 Data discrimination. A customer relationship manager at *AllElectronics* may want to compare two groups of customers—those who shop for computer products regularly (e.g., more than twice a month) and those who rarely shop for such products (e.g., less than three times a year). The resulting description provides a general comparative profile of these customers, such as that 80% of the customers who frequently purchase computer products are between 20 and 40 years old and have a university education, whereas 60% of the customers who infrequently buy such products are either seniors or youths, and have no university degree. Drilling down on a dimension like *occupation*, or adding a new dimension like *income level*, may help to find even more discriminative features between the two classes.

Concept description, including characterization and discrimination, is described in Chapter 4.

1.4.2 Mining Frequent Patterns, Associations, and Correlations

Frequent patterns, as the name suggests, are patterns that occur frequently in data. There are many kinds of frequent patterns, including frequent itemsets, frequent sub sequences (also known as sequential patterns), and frequent substructures. A *frequent itemset* typically refers to a set of items that often appear together in a transactional data set—for example, milk and bread, which are frequently bought together in gro cery stores by many customers. A frequently occurring subsequence, such as the pattern that customers, tend to purchase first a laptop, followed by a digital camera, and then a memory card, is a (*frequent*) *sequential pattern*. A substructure can refer to different structural forms (e.g., graphs, trees, or lattices) that may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (*frequent*) *structured pattern*. Mining frequent patterns leads to the discovery of interesting associations and correlations within data.

Example 1.7 Association analysis. Suppose that, as a marketing manager at *AllElectronics*, you want to know which items are frequently purchased together (i.e., within the same transac tion). An example of such a rule, mined from the *AllElectronics* transactional database, is

$$buys(X, \text{“}computer\text{”}) \Rightarrow buys(X, \text{“}software\text{”})\ [support = 1\%, confidence = 50\%],$$

where X is a variable representing a customer. A **confidence**, or certainty, of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well. A 1% **support** means that 1% of all the transactions under analysis show that computer and software are purchased together. This association rule involves a single attribute or predicate (i.e., *buys*) that repeats. Association rules that contain a single predicate are referred to as **single-dimensional association rules**. Dropping the predicate notation, the rule can be written simply as “*computer* ⇒ *software* [1%, 50%].”

Suppose, instead, that we are given the *AllElectronics* relational database related to purchases. A data mining system may find association rules like

$$age(X, \text{“20..29”}) \wedge income(X, \text{“40K..49K”}) \Rightarrow buys(X, \text{“laptop”})$$
$$[support = 2\%, confidence = 60\%].$$

The rule indicates that of the *AllElectronics* customers under study, 2% are 20 to 29 years old with an income of $40,000 to $49,000 and have purchased a laptop (computer) at *AllElectronics*. There is a 60% probability that a customer in this age and income group will purchase a laptop. Note that this is an association involving more than one attribute or predicate (i.e., *age, income*, and *buys*). Adopting the terminology used in multidimensional databases, where each attribute is referred to as a dimension, the above rule can be referred to as a **multidimensional association rule**.

Typically, association rules are discarded as uninteresting if they do not satisfy both a **minimum support threshold** and a **minimum confidence threshold**. Additional anal ysis can be performed to uncover interesting statistical **correlations** between associated attribute–value pairs.

Frequent itemset mining is a fundamental form of frequent pattern mining.

The min ing of frequent patterns, associations, and correlations is discussed in Chapters 6 and 7, where particular emphasis is placed on efficient algorithms for frequent itemset min ing. Sequential pattern mining and structured pattern mining are considered advanced topics.

1.4.3 Classification and Regression for Predictive Analysis

Classification is the process of finding a **model** (or function) that describes and distin guishes data classes or concepts. The model are derived based on the analysis of a set of **training data** (i.e., data objects for which the class labels are known). The model is used to predict the class label of objects for which the the class label is unknown.

“How is the derived model presented?” The derived model may be represented in var ious forms, such as *classification rules* (i.e., *IF-THEN rules*), *decision trees*, *mathematical formulae*, or *neural networks*(Figure 1.9). A**decision tree** is a flowchart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily

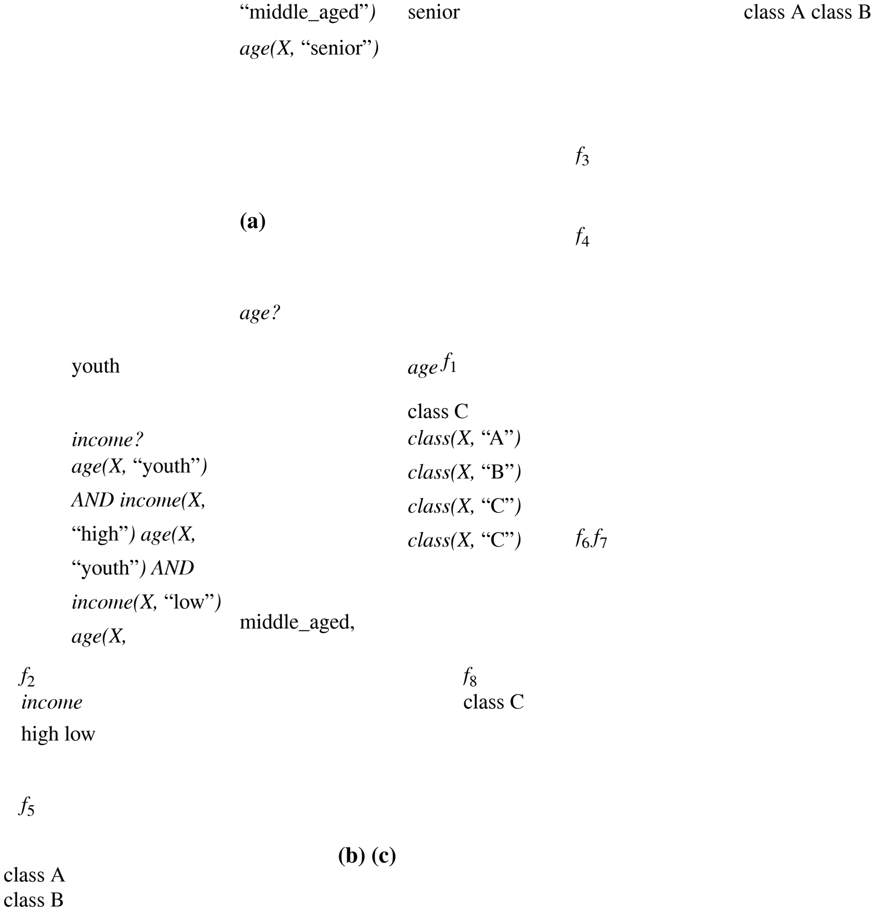

Figure 1.9 A classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

be converted to classification rules. A**neural network**, when used for classification, is typ ically a collection of neuron-like processing units with weighted connections between the units. There are many other methods for constructing classification models, such as na¨ıve Bayesian classification, support vector machines, and *k*-nearest-neighbor classification.

Whereas classification predicts categorical (discrete, unordered) labels, **regression** models continuous-valued functions. That is, regression is used to predict missing or unavailable *numerical data values* rather than (discrete) class labels. The term *prediction* refers to both numeric prediction and class label prediction. **Regression analysis** is a statistical methodology that is most often used for numeric prediction, although other methods exist as well. Regression also encompasses the identification of distribution *trends* based on the available data.

Classification and regression may need to be preceded by **relevance analysis**, which attempts to identify attributes that are significantly relevant to the classification and regression process. Such attributes will be selected for the classification and regression process. Other attributes, which are irrelevant, can then be excluded from consideration.

Example 1.8 Classification and regression. Suppose as a sales manager of *AllElectronics* you want to classify a large set of items in the store, based on three kinds of responses to a sales cam paign: *good response*, *mild response* and *no response*. You want to derive a model for each of these three classes based on the descriptive features of the items, such as *price*, *brand, place made, type*, and *category*. The resulting classification should maximally distinguish each class from the others, presenting an organized picture of the data set.

Suppose that the resulting classification is expressed as a decision tree. The decision tree, for instance, may identify *price* as being the single factor that best distinguishes the three classes. The tree may reveal that, in addition to *price*, other features that help to further distinguish objects of each class from one another include *brand* and *place made*. Such a decision tree may help you understand the impact of the given sales campaign and design a more effective campaign in the future.

Suppose instead, that rather than predicting categorical response labels for each store item, you would like to predict the amount of revenue that each item will generate during an upcoming sale at *AllElectronics*, based on the previous sales data. This is an example of regression analysis because the regression model constructed will predict a continuous function (or ordered value.)

Chapters 8 and 9 discuss classification in further detail. Regression analysis is beyond the scope of this book. Sources for further information are given in the bibliographic notes.

1.4.4 Cluster Analysis

Unlike classification and regression, which analyze class-labeled (training) data sets, **clustering** analyzes data objects without consulting class labels. In many cases, class labeled data may simply not exist at the beginning. Clustering can be used to generate

Figure 1.10 A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters.

class labels for a group of data. The objects are clustered or grouped based on the princi ple of *maximizing the intraclass similarity and minimizing the interclass similarity*. That is, clusters of objects are formed so that objects within a cluster have high similarity in com parison to one another, but are rather dissimilar to objects in other clusters. Each cluster so formed can be viewed as a class of objects, from which rules can be derived. Clus tering can also facilitate **taxonomy formation**, that is, the organization of observations into a hierarchy of classes that group similar events together.

Example 1.9 Cluster analysis. Cluster analysis can be performed on *AllElectronics* customer data to identify homogeneous subpopulations of customers. These clusters may represent indi vidual target groups for marketing. Figure 1.10 shows a 2-D plot of customers with respect to customer locations in a city. Three clusters of data points are evident.

Cluster analysis forms the topic of Chapters 10 and 11.

1.4.5 Outlier Analysis

A data set may contain objects that do not comply with the general behavior or model of the data. These data objects are **outliers**. Many data mining methods discard outliers as noise or exceptions. However, in some applications (e.g., fraud detection) the rare

events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as **outlier analysis** or **anomaly mining**.

Outliers may be detected using statistical tests that assume a distribution or proba bility model for the data, or using distance measures where objects that are remote from any other cluster are considered outliers. Rather than using statistical or distance mea sures, density-based methods may identify outliers in a local region, although they look normal from a global statistical distribution view.

Example 1.10 Outlier analysis. Outlier analysis may uncover fraudulent usage of credit cards by detecting purchases of unusually large amounts for a given account number in compari son to regular charges incurred by the same account. Outlier values may also be detected with respect to the locations and types of purchase, or the purchase frequency.

Outlier analysis is discussed in Chapter 12.

1.4.6 Are All Patterns Interesting?

A data mining system has the potential to generate thousands or even millions of patterns, or rules.

You may ask, *"Are all of the patterns interesting?"* Typically, the answer is no—only a small fraction of the patterns potentially generated would actually be of interest to a given user.

This raises some serious questions for data mining. You may wonder, *"What makes a pattern interesting? Can a data mining system generate all of the interesting patterns? Or, Can the system generate only the interesting ones?"*

To answer the first question, a pattern is **interesting** if it is (1) *easily understood* by humans, (2) *valid* on new or test data with some degree of *certainty*, (3) potentially *useful*, and (4) *novel*. A pattern is also interesting if it validates a hypothesis that the user *sought to confirm*. An interesting pattern represents **knowledge**.

Several **objective measures of pattern interestingness** exist. These are based on the structure of discovered patterns and the statistics underlying them. An objective measure for association rules of the form $X \Rightarrow Y$ is rule **support**, representing the per centage of transactions from a transaction database that the given rule satisfies. This is taken to be the probability $P(X \cup Y)$, where $X \cup Y$ indicates that a transaction contains both X and Y, that is, the union of itemsets X and Y. Another objective measure for association rules is **confidence**, which assesses the degree of certainty of the detected association. This is taken to be the conditional probability $P(Y|X)$, that is, the prob ability that a transaction containing X also contains Y. More formally, support and confidence are defined as

$$support(X \Rightarrow Y) = P(X \cup Y),$$

$$confidence(X \Rightarrow Y) = P(Y|X).$$

In general, each interestingness measure is associated with a threshold, which may be controlled by the user. For example, rules that do not satisfy a confidence threshold of,

say, 50% can be considered uninteresting. Rules below the threshold likely reflect noise, exceptions, or minority cases and are probably of less value.

Other objective interestingness measures include *accuracy* and *coverage* for classification (IF-THEN) rules. In general terms, accuracy tells us the percentage of data that are correctly classified by a rule. Coverage is similar to support, in that it tells us the percentage of data to which a rule applies. Regarding understandability, we may use simple objective measures that assess the complexity or length in bits of the patterns mined.

Although objective measures help identify interesting patterns, they are often insufficient unless combined with subjective measures that reflect a particular user's needs and interests. For example, patterns describing the characteristics of customers who shop frequently at *AllElectronics* should be interesting to the marketing manager, but may be of little interest to other analysts studying the same database for patterns on employee performance. Furthermore, many patterns that are interesting by objective standards may represent common sense and, therefore, are actually uninteresting.

Subjective interestingness measures are based on user beliefs in the data. These measures find patterns interesting if the patterns are **unexpected** (contradicting a user's belief) or offer strategic information on which the user can act. In the latter case, such patterns are referred to as **actionable**. For example, patterns like "a large earthquake often follows a cluster of small quakes" may be highly actionable if users can act on the information to save lives. Patterns that are **expected** can be interesting if they confirm a hypothesis that the user wishes to validate or they resemble a user's hunch.

The second question—"*Can a data mining system generate* all *of the interesting patterns?*"—refers to the **completeness** of a data mining algorithm. It is often unrealistic and inefficient for data mining systems to generate all possible patterns. Instead, user provided constraints and interestingness measures should be used to focus the search. For some mining tasks, such as association, this is often sufficient to ensure the completeness of the algorithm. Association rule mining is an example where the use of constraints and interestingness measures can ensure the completeness of mining. The methods involved are examined in detail in Chapter 6.

Finally, the third question—*"Can a data mining system generate only interesting patterns?"*—is an optimization problem in data mining. It is highly desirable for data mining systems to generate only interesting patterns. This would be efficient for users and data mining systems because neither would have to search through the patterns generated to identify the truly interesting ones. Progress has been made in this direction; however, such optimization remains a challenging issue in data mining.

Measures of pattern interestingness are essential for the efficient discovery of patterns by target users. Such measures can be used after the data mining step to rank the discovered patterns according to their interestingness, filtering out the uninteresting ones. More important, such measures can be used to guide and constrain the discovery process, improving the search efficiency by pruning away subsets of the pattern space that do not satisfy prespecified interestingness constraints. Examples of such a constraint based mining process are described in Chapter 7 (with respect to pattern discovery) and Chapter 11 (with respect to clustering).

Methods to assess pattern interestingness, and their use to improve data mining efficiency, are discussed throughout the book with respect to each kind of pattern that can be mined.

1.5 Which Technologies Are Used?

As a highly application-driven domain, data mining has incorporated many techniques from other domains such as statistics, machine learning, pattern recognition, database and data warehouse systems, information retrieval, visualization, algorithms, high performance computing, and many application domains (Figure 1.11). The interdisciplinary nature of data mining research and development contributes significantly to the success of data mining and its extensive applications. In this section, we give examples of several disciplines that strongly influence the development of data mining methods.

1.5.1 Statistics

Statisticsstudies the collection, analysis, interpretation or explanation, and presentation of data. Data mining has an inherent connection with statistics.

A **statistical model** is a set of mathematical functions that describe the behavior of the objects in a target class in terms of random variables and their associated probability distributions. Statistical models are widely used to model data and data classes. For example, in data mining tasks like data characterization and classification, statistical

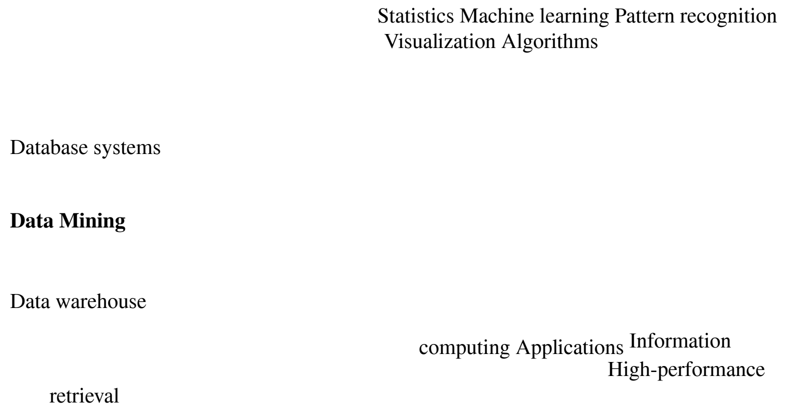

Figure 1.11 Data mining adopts techniques from many domains.

models of target classes can be built. In other words, such statistical models can be the outcome of a data mining task. Alternatively, data mining tasks can be built on top of statistical models. For example, we can use statistics to model noise and missing data values. Then, when mining patterns in a large data set, the data mining process can use the model to help identify and handle noisy or missing values in the data.

Statistics research develops tools for prediction and forecasting using data and sta tistical models. Statistical methods can be used to summarize or describe a collection of data. Basic **statistical descriptions** of data are introduced in Chapter 2. Statistics is useful for mining various patterns from data as well as for understanding the underlying mechanisms generating and affecting the patterns. **Inferential statistics** (or **predictive statistics**) models data in a way that accounts for randomness and uncertainty in the observations and is used to draw inferences about the process or population under investigation.

Statistical methods can also be used to verify data mining results. For example, after a classification or prediction model is mined, the model should be verified by statisti cal hypothesis testing. A **statistical hypothesis test** (sometimes called *confirmatory data analysis*) makes statistical decisions using experimental data. A result is called *statistically significant* if it is unlikely to have occurred by chance. If the classification or prediction model holds true, then the descriptive statistics of the model increases the soundness of the model.

Applying statistical methods in data mining is far from trivial. Often, a serious chal lenge is how to scale up a statistical method over a large data set. Many statistical methods have high complexity in computation. When such methods are applied on large data sets that are also distributed on multiple logical or physical sites, algorithms should be carefully designed and tuned to reduce the computational cost. This challenge becomes even tougher for online applications, such as online query suggestions in search engines, where data mining is required to continuously handle fast, real-time data streams.

1.5.2 Machine Learning

Machine learning investigates how computers can learn (or improve their performance) based on data. A main research area is for computer programs to *automatically* learn to recognize complex patterns and make intelligent decisions based on data. For example, a typical machine learning problem is to program a computer so that it can automatically recognize handwritten postal codes on mail after learning from a set of examples.

Machine learning is a fast-growing discipline. Here, we illustrate classic problems in machine learning that are highly related to data mining.

Supervised learning is basically a synonym for classification. The supervision in the learning comes from the labeled examples in the training data set. For example, in the postal code recognition problem, a set of handwritten postal code images and their corresponding machine-readable translations are used as the training examples, which supervise the learning of the classification model.

Unsupervised learning is essentially a synonym for clustering. The learning process is unsupervised since the input examples are not class labeled. Typically, we may use clustering to discover classes within the data. For example, an unsupervised learning method can take, as input, a set of images of handwritten digits. Suppose that it finds 10 clusters of data. These clusters may correspond to the 10 distinct digits of 0 to 9, respectively. However, since the training data are not labeled, the learned model cannot tell us the semantic meaning of the clusters found.

Semi-supervised learning is a class of machine learning techniques that make use of both labeled and unlabeled examples when learning a model. In one approach, labeled examples are used to learn class models and unlabeled examples are used to refine the boundaries between classes. For a two-class problem, we can think of the set of examples belonging to one class as the *positive examples* and those belonging to the other class as the *negative examples*. In Figure 1.12, if we do not consider the unlabeled examples, the dashed line is the decision boundary that best partitions the positive examples from the negative examples. Using the unlabeled examples, we can refine the decision boundary to the solid line. Moreover, we can detect that the two positive examples at the top right corner, though labeled, are likely noise or outliers.

Active learning is a machine learning approach that lets users play an active role in the learning process. An active learning approach can ask a user (e.g., a domain expert) to label an example, which may be from a set of unlabeled examples or synthesized by the learning program. The goal is to optimize the model quality by actively acquiring knowledge from human users, given a constraint on how many examples they can be asked to label.

Noise/outliers

Negative example

Unlabeled example

Figure 1.12 Semi-supervised learning.

Decision boundary without unlabeled examples Decision boundary with unlabeled examples

Positive example

You can see there are many similarities between data mining and machine learning. For classification and clustering tasks, machine learning research often focuses on the accuracy of the model. In addition to accuracy, data mining research places strong emphasis on the efficiency and scalability of mining methods on large data sets, as well as on ways to handle complex types of data and explore new, alternative methods.

1.5.3 Database Systems and Data Warehouses

Database systems research focuses on the creation, maintenance, and use of databases for organizations and end-users. Particularly, database systems researchers have estab lished highly recognized principles in data models, query languages, query processing and optimization methods, data storage, and indexing and accessing methods. Database systems are often well known for their high scalability in processing very large, relatively structured data sets.

Many data mining tasks need to handle large data sets or even real-time, fast stream ing data. Therefore, data mining can make good use of scalable database technologies to achieve high efficiency and scalability on large data sets. Moreover, data mining tasks can be used to extend the capability of existing database systems to satisfy advanced users' sophisticated data analysis requirements.

Recent database systems have built systematic data analysis capabilities on database data using data warehousing and data mining facilities. A **data warehouse** integrates data originating from multiple sources and various timeframes. It consolidates data in multidimensional space to form partially materialized data cubes. The data cube model not only facilitates OLAP in multidimensional databases but also promotes *multidimensional data mining* (see Section 1.3.2).

1.5.4 Information Retrieval

Information retrieval (**IR**) is the science of searching for documents or information in documents. Documents can be text or multimedia, and may reside on the Web. The differences between traditional information retrieval and database systems are twofold: Information retrieval assumes that (1) the data under search are unstructured; and (2) the queries are formed mainly by keywords, which do not have complex structures (unlike SQL queries in database systems).

The typical approaches in information retrieval adopt probabilistic models. For example, a text document can be regarded as a bag of words, that is, a multiset of words appearing in the document. The document's **language model** is the probability density function that generates the bag of words in the document. The similarity between two documents can be measured by the similarity between their corresponding language models.

Furthermore, a topic in a set of text documents can be modeled as a probability dis tribution over the vocabulary, which is called a **topic model**. A text document, which may involve one or multiple topics, can be regarded as a mixture of multiple topic mod els. By integrating information retrieval models and data mining techniques, we can find

the major topics in a collection of documents and, for each document in the collection, the major topics involved.

Increasingly large amounts of text and multimedia data have been accumulated and made available online due to the fast growth of the Web and applications such as dig ital libraries, digital governments, and health care information systems. Their effective search and analysis have raised many challenging issues in data mining. Therefore, text mining and multimedia data mining, integrated with information retrieval methods, have become increasingly important.

1.6 Which Kinds of Applications Are Targeted?

Where there are data, there are data mining applications

As a highly application-driven discipline, data mining has seen great successes in many applications. It is impossible to enumerate all applications where data mining plays a critical role. Presentations of data mining in knowledge-intensive application domains, such as bioinformatics and software engineering, require more in-depth treatment and are beyond the scope of this book. To demonstrate the importance of applications as a major dimension in data mining research and development, we briefly discuss two highly successful and popular application examples of data mining: *business intelligence* and *search engines*.

1.6.1 Business Intelligence

It is critical for businesses to acquire a better understanding of the commercial context of their organization, such as their customers, the market, supply and resources, and competitors. **Business intelligence** (**BI**) technologies provide historical, current, and predictive views of business operations. Examples include reporting, online analytical processing, business performance management, competitive intelligence, benchmark ing, and predictive analytics.

"How important is business intelligence?" Without data mining, many businesses may not be able to perform effective market analysis, compare customer feedback on simi lar products, discover the strengths and weaknesses of their competitors, retain highly valuable customers, and make smart business decisions.

Clearly, data mining is the core of business intelligence. Online analytical process ing tools in business intelligence rely on data warehousing and multidimensional data mining. Classification and prediction techniques are the core of predictive analytics in business intelligence, for which there are many applications in analyzing markets, supplies, and sales. Moreover, clustering plays a central role in customer relationship management, which groups customers based on their similarities. Using characteriza tion mining techniques, we can better understand features of each customer group and develop customized customer reward programs.

1.6.2 Web Search Engines

A **Web search engine** is a specialized computer server that searches for information on the Web. The search results of a user query are often returned as a list (sometimes called *hits*). The hits may consist of web pages, images, and other types of files. Some search engines also search and return data available in public databases or open directo ries. Search engines differ from **web directories** in that web directories are maintained by human editors whereas search engines operate algorithmically or by a mixture of algorithmic and human input.

Web search engines are essentially very large data mining applications. Various data mining techniques are used in all aspects of search engines, ranging from *crawling* [5] (e.g., deciding which pages should be crawled and the crawling frequencies), indexing (e.g., selecting pages to be indexed and deciding to which extent the index should be constructed), and searching (e.g., deciding how pages should be ranked, which adver tisements should be added, and how the search results can be personalized or made "context aware").

Search engines pose grand challenges to data mining. First, they have to handle a huge and ever-growing amount of data. Typically, such data cannot be processed using one or a few machines. Instead, search engines often need to use *computer clouds*, which consist of thousands or even hundreds of thousands of computers that collaboratively mine the huge amount of data. Scaling up data mining methods over computer clouds and large distributed data sets is an area for further research.

Second, Web search engines often have to deal with online data. A search engine may be able to afford constructing a model offline on huge data sets. To do this, it may construct a query classifier that assigns a search query to predefined categories based on the query topic (i.e., whether the search query "apple" is meant to retrieve information about a fruit or a brand of computers). Whether a model is constructed offline, the application of the model online must be fast enough to answer user queries in real time.

Another challenge is maintaining and incrementally updating a model on fast growing data streams. For example, a query classifier may need to be incrementally maintained continuously since new queries keep emerging and predefined categories and the data distribution may change. Most of the existing model training methods are offline and static and thus cannot be used in such a scenario.

Third, Web search engines often have to deal with queries that are asked only a very small number of times. Suppose a search engine wants to provide *context-aware* query recommendations. That is, when a user poses a query, the search engine tries to infer the context of the query using the user's profile and his query history in order to return more customized answers within a small fraction of a second. However, although the total number of queries asked can be huge, most of the queries may be asked only once or a few times. Such severely skewed data are challenging for many data mining and machine learning methods.

[5] A Web crawler is a computer program that browses the Web in a methodical, automated manner.

1.7 Major Issues in Data Mining

Life is short but art is long. – Hippocrates

Data mining is a dynamic and fast-expanding field with great strengths. In this section, we briefly outline the major issues in data mining research, partitioning them into five groups: *mining methodology, user interaction, efficiency and scalability, diversity of data types*, and *data mining and society*. Many of these issues have been addressed in recent data mining research and development *to a certain extent* and are now consid ered *data mining requirements*; others are still at the research stage. The issues continue to stimulate further investigation and improvement in data mining.

1.7.1 Mining Methodology

Researchers have been vigorously developing new data mining methodologies. This involves the investigation of new kinds of knowledge, mining in multidimensional space, integrating methods from other disciplines, and the consideration of semantic ties among data objects. In addition, mining methodologies should consider issues such as data uncertainty, noise, and incompleteness. Some mining methods explore how user specified measures can be used to assess the interestingness of discovered patterns as well as guide the discovery process. Let's have a look at these various aspects of mining methodology.

Mining various and new kinds of knowledge: Data mining covers a wide spectrum of data analysis and knowledge discovery tasks, from data characterization and discrim ination to association and correlation analysis, classification, regression, clustering, outlier analysis, sequence analysis, and trend and evolution analysis. These tasks may use the same database in different ways and require the development of numerous data mining techniques. Due to the diversity of applications, new mining tasks con tinue to emerge, making data mining a dynamic and fast-growing field. For example, for effective knowledge discovery in information networks, integrated clustering and ranking may lead to the discovery of high-quality clusters and object ranks in large networks.

Mining knowledge in multidimensional space: When searching for knowledge in large data sets, we can explore the data in multidimensional space. That is, we can search for interesting patterns among combinations of dimensions (attributes) at varying levels of abstraction. Such mining is known as *(exploratory) multidimensional data mining*. In many cases, data can be aggregated or viewed as a multidimensional data cube. Mining knowledge in cube space can substantially enhance the power and flexibility of data mining.

Data mining—an interdisciplinary effort: The power of data mining can be substan tially enhanced by integrating new methods from multiple disciplines. For example, to mine data with natural language text, it makes sense to fuse data mining methods with methods of information retrieval and natural language processing. As another example, consider the mining of software bugs in large programs. This form of min ing, known as *bug mining*, benefits from the incorporation of software engineering knowledge into the data mining process.

Boosting the power of discovery in a networked environment: Most data objects reside in a linked or interconnected environment, whether it be the Web, database rela tions, files, or documents. Semantic links across multiple data objects can be used to advantage in data mining. Knowledge derived in one set of objects can be used to boost the discovery of knowledge in a "related" or semantically linked set of objects.

Handling uncertainty, noise, or incompleteness of data: Data often contain noise, errors, exceptions, or uncertainty, or are incomplete. Errors and noise may confuse the data mining process, leading to the derivation of erroneous patterns. Data clean ing, data preprocessing, outlier detection and removal, and uncertainty reasoning are examples of techniques that need to be integrated with the data mining process.

Pattern evaluation and pattern- or constraint-guided mining: Not all the patterns gen erated by data mining processes are interesting. What makes a pattern interesting may vary from user to user. Therefore, techniques are needed to assess the inter estingness of discovered patterns based on subjective measures. These estimate the value of patterns with respect to a given user class, based on user beliefs or expec tations. Moreover, by using interestingness measures or user-specified constraints to *guide* the discovery process, we may generate more interesting patterns and reduce the search space.

1.7.2 User Interaction

The user plays an important role in the data mining process. Interesting areas of research include *how to interact with a data mining system*, *how to incorporate a user's back ground knowledge in mining*, and *how to visualize and comprehend data mining results*. We introduce each of these here.

Interactive mining: The data mining process should be highly *interactive*. Thus, it is important to build flexible user interfaces and an exploratory mining environment, facilitating the user's interaction with the system. A user may like to first sample a set of data, explore general characteristics of the data, and estimate potential min ing results. Interactive mining should allow users to dynamically change the focus of a search, to refine mining requests based on returned results, and to drill, dice, and pivot through the data and knowledge space interactively, dynamically exploring "cube space" while mining.

Incorporation of background knowledge: Background knowledge, constraints, rules, and other information regarding the domain under study should be incorporated into the knowledge discovery process. Such knowledge can be used for pattern evaluation as well as to guide the search toward interesting patterns.

Ad hoc data mining and data mining query languages: Query languages (e.g., SQL) have played an important role in flexible searching because they allow users to pose ad hoc queries. Similarly, high-level data mining query languages or other high-level flexible user interfaces will give users the freedom to define ad hoc data mining tasks. This should facilitate specification of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and constraints to be enforced on the discovered patterns. Optimization of the processing of such flexible mining requests is another promising area of study.

Presentation and visualization of data mining results: How can a data mining system present data mining results, vividly and flexibly, so that the discovered knowledge can be easily understood and directly usable by humans? This is especially crucial if the data mining process is interactive. It requires the system to adopt expressive knowledge representations, user-friendly interfaces, and visualization techniques.

1.7.3 Efficiency and Scalability

Efficiency and scalability are always considered when comparing data mining algo rithms. As data amounts continue to multiply, these two factors are especially critical.

Efficiency and scalability of data mining algorithms: Data mining algorithms must be efficient and scalable in order to effectively extract information from huge amounts of data in many data repositories or in dynamic data streams. In other words, the running time of a data mining algorithm must be predictable, short, and acceptable by applications. *Efficiency, scalability, performance, optimization*, and the ability to execute in *real time* are key criteria that drive the development of many new data mining algorithms.

Parallel, distributed, and incremental mining algorithms: The humongous size of many data sets, the wide distribution of data, and the computational complexity of some data mining methods are factors that motivate the development of **parallel and dis tributed data-intensive mining algorithms**. Such algorithms first partition the data into "pieces." Each piece is processed, in parallel, by searching for patterns. The par allel processes may interact with one another. The patterns from each partition are eventually merged.

Cloud computing and *cluster computing*, which use computers in a distributed and collaborative way to tackle very large-scale computational tasks, are also active research themes in parallel data mining. In addition, the high cost of some data min ing processes and the incremental nature of input promote **incremental** data mining, which incorporates new data updates without having to mine the entire data “from scratch.” Such methods perform knowledge modification incrementally to amend and strengthen what was previously discovered.

1.7.4 Diversity of Database Types

The wide diversity of database types brings about challenges to data mining. These include

Handling complex types of data: Diverse applications generate a wide spectrum of new data types, from structured data such as relational and data warehouse data to semi-structured and unstructured data; from stable data repositories to dynamic data streams; from simple data objects to temporal data, biological sequences, sensor data, spatial data, hypertext data, multimedia data, software program code, Web data, and social network data. It is unrealistic to expect one data mining system to mine all kinds of data, given the diversity of data types and the different goals of data mining. Domain- or application-dedicated data mining systems are being constructed for in depth mining of specific kinds of data. The construction of effective and efficient data mining tools for diverse applications remains a challenging and active area of research.

Mining dynamic, networked, and global data repositories: Multiple sources of data are connected by the Internet and various kinds of networks, forming gigantic, dis tributed, and heterogeneous global information systems and networks. The discovery of knowledge from different sources of structured, semi-structured, or unstructured yet interconnected data with diverse data semantics poses great challenges to data mining. Mining such gigantic, interconnected information networks may help dis close many more patterns and knowledge in heterogeneous data sets than can be dis covered from a small set of isolated data repositories. Web mining, multisource data mining, and information network mining have become challenging and fast-evolving data mining fields.

1.7.5 Data Mining and Society

How does data mining impact society? What steps can data mining take to preserve the privacy of individuals? Do we use data mining in our daily lives without even knowing that we do? These questions raise the following issues:

Social impacts of data mining: With data mining penetrating our everyday lives, it is important to study the impact of data mining on society. How can we use data mining technology to benefit society? How can we guard against its misuse? The improper disclosure or use of data and the potential violation of individual privacy and data protection rights are areas of concern that need to be addressed.

Privacy-preserving data mining: Data mining will help scientific discovery, business management, economy recovery, and security protection (e.g., the real-time dis covery of intruders and cyberattacks). However, it poses the risk of disclosing an individual’s personal information. Studies on privacy-preserving data publishing and data mining are ongoing. The philosophy is to observe data sensitivity and preserve people’s privacy while performing successful data mining.

Invisible data mining: We cannot expect everyone in society to learn and master data mining techniques. More and more systems should have data mining functions built within so that people can perform data mining or use data mining results simply by mouse clicking, without any knowledge of data mining algorithms. Intelligent search engines and Internet-based stores perform such *invisible data mining* by incorporating data mining into their components to improve their functionality and performance. This is done often unbeknownst to the user. For example, when purchasing items online, users may be unaware that the store is likely collecting data on the buying patterns of its customers, which may be used to recommend other items for purchase in the future.

These issues and many additional ones relating to the research, development, and application of data mining are discussed throughout the book.

1.8 Summary

Necessity is the mother of invention. With the mounting growth of data in every application, data mining meets the imminent need for effective, scalable, and flexible data analysis in our society. Data mining can be considered as a natural evolution of information technology and a confluence of several related disciplines and application domains.

Data mining is the process of discovering interesting patterns from massive amounts of data. As a *knowledge discovery process*, it typically involves data cleaning, data integration, data selection, data transformation, pattern discovery, pattern evaluation, and knowledge presentation.

A pattern is *interesting* if it is valid on test data with some degree of certainty, novel, potentially useful (e.g., can be acted on or validates a hunch about which the user was curious), and easily understood by humans. Interesting patterns represent **knowledge**. Measures of **pattern interestingness**, either *objective* or *subjective*, can be used to guide the discovery process.

We present a **multidimensional view** of data mining. The major dimensions are **data**, **knowledge**, **technologies**, and **applications**.

Data mining can be conducted on any kind of **data** as long as the data are meaningful for a target application, such as database data, data warehouse data, transactional data, and advanced data types. Advanced data types include time-related or sequence data, data streams, spatial and spatiotemporal data, text and multimedia data, graph and networked data, and Web data.

A **data warehouse** is a repository for long-term storage of data from multiple sources, organized so as to facilitate management decision making. The data are stored under a unified schema and are typically summarized. Data warehouse systems pro vide multidimensional data analysis capabilities, collectively referred to as **online analytical processing**.

Multidimensional data mining (also called **exploratory multidimensional data mining**) integrates core data mining techniques with OLAP-based multidimen sional analysis. It searches for interesting patterns among multiple combinations of dimensions (attributes) at varying levels of abstraction, thereby exploring multi dimensional data space.

Data mining functionalities are used to specify the kinds of patterns or **knowledge** to be found in data mining tasks. The functionalities include characterization and discrimination; the mining of frequent patterns, associations, and correlations; clas sification and regression; cluster analysis; and outlier detection. As new types of data, new applications, and new analysis demands continue to emerge, there is no doubt we will see more and more novel data mining tasks in the future.

Data mining, as a highly application-driven domain, has incorporated **technologies** from many other domains. These include statistics, machine learning, database and data warehouse systems, and information retrieval. The **interdisciplinary nature of data mining research and development** contributes significantly to the success of data mining and its extensive applications.

Data mining has many successful **applications**, such as business intelligence, Web search, bioinformatics, health informatics, finance, digital libraries, and digital governments.

There are many challenging **issues in data mining research**. Areas include mining methodology, user interaction, efficiency and scalability, and dealing with diverse data types. Data mining research has strongly impacted society and will continue to do so in the future.

1.9 Exercises

1.1 What is *data mining*? In your answer, address the following:

(a) Is it another hype?

(b) Is it a simple transformation or application of technology developed from *databases*, *statistics*, *machine learning*, and *pattern recognition*?

(c) We have presented a view that data mining is the result of the evolution of *database technology*. Do you think that data mining is also the result of the evolution of *machine learning research*? Can you present such views based on the historical progress of this discipline? Address the same for the fields of *statistics* and *pattern recognition*.

(d) Describe the steps involved in data mining when viewed as a process of knowledge discovery.

1.2 How is a *data warehouse* different from a *database*? How are they similar?

1.3 Define each of the following *data mining functionalities*: characterization, discrimi nation, association and correlation analysis, classification, regression, clustering, and

outlier analysis. Give examples of each data mining functionality, using a real-life database that you are familiar with.

1.4 Present an example where data mining is crucial to the success of a business. What *data mining functionalities* does this business need (e.g., think of the kinds of patterns that could be mined)? Can such patterns be generated alternatively by data query processing or simple statistical analysis?

1.5 Explain the difference and similarity between discrimination and classification, between characterization and clustering, and between classification and regression.

1.6 Based on your observations, describe another possible kind of knowledge that needs to be discovered by data mining methods but has not been listed in this chapter. Does it require a mining methodology that is quite different from those outlined in this chapter?

1.7 *Outliers* are often discarded as noise. However, one person's garbage could be another's treasure. For example, exceptions in credit card transactions can help us detect the fraudulent use of credit cards. Using fraudulence detection as an example, propose two methods that can be used to detect outliers and discuss which one is more reliable.

1.8 Describe three challenges to data mining regarding *data mining methodology* and *user interaction issues.*

1.9 What are the major challenges of mining a huge amount of data (e.g., billions of tuples) in comparison with mining a small amount of data (e.g., data set of a few hundred tuple)?

1.10 Outline the major research challenges of data mining in one specific application domain, such as stream/sensor data analysis, spatiotemporal data analysis, or bioinformatics.

1.10 Bibliographic Notes

The book *Knowledge Discovery in Databases*, edited by Piatetsky-Shapiro and Frawley [P-SF91], is an early collection of research papers on knowledge discovery from data. The book *Advances in Knowledge Discovery and Data Mining*, edited by Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy [FPSS+96], is a collection of later research results on knowledge discovery and data mining. There have been many data min ing books published in recent years, including *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman [HTF09]; *Introduction to Data Mining* by Tan, Steinbach, and Kumar [TSK05]; *Data Mining: Practical Machine Learning Tools* *and Techniques with Java Implementations* by Witten, Frank, and Hall [WFH11]; *Predictive Data Mining* by Weiss and Indurkhya [WI98]; *Mastering Data Mining: The Art and Science of Customer Relationship Management* by Berry and Linoff [BL99]; *Principles of Data Mining (Adaptive Computation and Machine Learning)* by Hand, Mannila, and Smyth [HMS01]; *Mining the Web: Discovering Knowledge from Hypertext Data* by Chakrabarti [Cha03a]; *Web Data Mining: Exploring Hyperlinks, Contents, and Usage*

Data by Liu [Liu06]; *Data Mining: Introductory and Advanced Topics* by Dunham [Dun03]; and *Data Mining: Multimedia, Soft Computing, and Bioinformatics* by Mitra and Acharya [MA03].

There are also books that contain collections of papers or chapters on particular aspects of knowledge discovery—for example, *Relational Data Mining* edited by Dzeroski and Lavrac [De01]; *Mining Graph Data* edited by Cook and Holder [CH07]; *Data Streams: Models and Algorithms* edited by Aggarwal [Agg06]; *Next Generation of Data Mining* edited by Kargupta, Han, Yu, et al. [KHY$^+$08]; *Multimedia Data Mining: A Systematic Introduction to Concepts and Theory* edited by Z. Zhang and R. Zhang [ZZ09]; *Geographic Data Mining and Knowledge Discovery* edited by Miller and Han [MH09]; and *Link Mining: Models, Algorithms and Applications* edited by Yu, Han, and Faloutsos [YHF10]. There are many tutorial notes on data mining in major databases, data mining, machine learning, statistics, and Web technology conferences.

KDNuggets is a regular electronic newsletter containing information relevant to knowledge discovery and data mining, moderated by Piatetsky-Shapiro since 1991. The Internet site *KDNuggets* (*www.kdnuggets.com*) contains a good collection of KDD related information.

The data mining community started its first international conference on knowledge discovery and data mining in 1995. The conference evolved from the four international workshops on knowledge discovery in databases, held from 1989 to 1994. ACM-SIGKDD, a Special Interest Group on Knowledge Discovery in Databases was set up under ACM in 1998 and has been organizing the international conferences on knowledge discovery and data mining since 1999. IEEE Computer Science Society has organized its annual data mining conference, International Conference on Data Mining (ICDM), since 2001. SIAM (Society on Industrial and Applied Mathematics) has organized its annual data mining conference, SIAM Data Mining Conference (SDM), since 2002. A dedicated journal, *Data Mining and Knowledge Discovery*, published by Kluwers Publishers, has been available since 1997. An ACM journal, *ACM Transactions on Knowledge Discovery from Data*, published its first volume in 2007.

ACM-SIGKDD also publishes a bi-annual newsletter, *SIGKDD Explorations*. There are a few other international or regional conferences on data mining, such as the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), and the International Conference on Data Warehousing and Knowledge Discovery (DaWaK).

Research in data mining has also been published in books, conferences, and journals on databases, statistics, machine learning, and data visualization. References to such sources are listed at the end of the book.

Popular textbooks on database systems include *Database Systems: The Complete Book* by Garcia-Molina, Ullman, and Widom [GMUW08]; *Database Management Systems* by Ramakrishnan and Gehrke [RG03]; *Database System Concepts* by Silberschatz, Korth, and Sudarshan [SKS10]; and *Fundamentals of Database Systems* by Elmasri and Navathe [EN10]. For an edited collection of seminal articles on database systems, see *Readings in Database Systems* by Hellerstein and Stonebraker [HS05].

There are also many books on data warehouse technology, systems, and applications, such as *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* by Kimball and Ross [KR02]; *The Data Warehouse Lifecycle Toolkit* by Kimball, Ross, Thornthwaite, and Mundy [KRTM08]; *Mastering Data Warehouse Design: Relational and Dimensional Techniques* by Imhoff, Galemmo, and Geiger [IGG03]; and *Building the Data Warehouse* by Inmon [Inm96]. A set of research papers on materialized views and data warehouse implementations were collected in *Materialized Views: Techniques, Implementations, and Applications* by Gupta and Mumick [GM99]. Chaudhuri and Dayal [CD97] present an early comprehensive overview of data warehouse technology.

Research results relating to data mining and data warehousing have been published in the proceedings of many international database conferences, including the ACM-SIGMOD International Conference on Management of Data (SIGMOD), the International Conference on Very Large Data Bases (VLDB), the ACM SIGACT SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), the International Conference on Data Engineering (ICDE), the International Conference on Extending Database Technology (EDBT), the International Conference on Database Theory (ICDT), the International Conference on Information and Knowledge Management (CIKM), the International Conference on Database and Expert Systems Applications (DEXA), and the International Symposium on Database Systems for Advanced Applications (DASFAA). Research in data mining is also published in major database journals, such as *IEEE Transactions on Knowledge and Data Engineering (TKDE), ACM Transactions on Database Systems (TODS), Information Systems, The VLDB Journal, Data and Knowledge Engineering*, *International Journal of Intelligent Information Systems (JIIS)*, and *Knowledge and Information Systems (KAIS)*.

Many effective data mining methods have been developed by statisticians and introduced in a rich set of textbooks. An overview of classification from a statistical pattern recognition perspective can be found in *Pattern Classification* by Duda, Hart, and Stork [DHS01]. There are also many textbooks covering regression and other topics in statistical analysis, such as *Mathematical Statistics: Basic Ideas and Selected Topics* by Bickel and Doksum [BD01]; *The Statistical Sleuth: A Course in Methods of Data Analysis* by Ramsey and Schafer [RS01]; *Applied Linear Statistical Models* by Neter, Kutner, Nachtsheim, and Wasserman [NKNW96]; *An Introduction to Generalized Linear Models* by Dobson [Dob90]; *Applied Statistical Time Series Analysis* by Shumway [Shu88]; and *Applied Multivariate Statistical Analysis* by

Johnson and Wichern [JW92].

Research in statistics is published in the proceedings of several major statistical con ferences, including Joint Statistical Meetings, International Conference of the Royal Statistical Society and Symposium on the Interface: Computing Science and Statistics. Other sources of publication include the *Journal of the Royal Statistical Society*, *The Annals of Statistics*, the *Journal of American Statistical Association*, *Technometrics*, and *Biometrika*.

Textbooks and reference books on machine learning and pattern recognition include *Machine Learning* by Mitchell [Mit97]; *Pattern Recognition and Machine Learning* by Bishop [Bis06]; *Pattern Recognition* by Theodoridis and Koutroumbas [TK08]; *Introduc tion to Machine Learning* by Alpaydin [Alp11]; *Probabilistic Graphical Models: Principles and Techniques* by Koller and Friedman [KF09]; and *Machine Learning: An Algorithmic Perspective* by Marsland [Mar09]. For an edited collection of seminal articles on machine learning, see *Machine Learning, An Artificial Intelligence Approach*, Volumes 1 through 4, edited by Michalski et al. [MCM83, MCM86, KM90, MT94], and *Readings in Machine Learning* by Shavlik and Dietterich [SD90].

Machine learning and pattern recognition research is published in the proceed ings of several major machine learning, artificial intelligence, and pattern recognition conferences, including the International Conference on Machine Learning (ML), the ACM Conference on Computational Learning Theory (COLT), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Pattern Recognition (ICPR), the International Joint Conference on Artificial Intel ligence (IJCAI), and the American Association of Artificial Intelligence Conference (AAAI). Other sources of publication include major machine learning, artificial intelligence, pattern recognition, and knowledge system journals, some of which have been mentioned before. Others include *Machine Learning (ML), Pattern Recognition (PR), Artificial Intelligence Journal (AI)*, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, and *Cognitive Science*.

Textbooks and reference books on information retrieval include *Introduction to Information Retrieval* by Manning, Raghavan, and Schutz [MRS08]; *Information Retrieval: Implementing and Evaluating Search Engines* by Buttcher, Clarke, and Cormack ¨ [BCC10]; *Search Engines: Information Retrieval in Practice* by Croft, Metzler, and Strohman [CMS09]; *Modern Information Retrieval: The Concepts and Technology Behind Search* by Baeza-Yates and Ribeiro-Neto [BYRN11]; and *Information Retrieval: Algorithms and Heuristics* by Grossman and Frieder [GR04].

Information retrieval research is published in the proceedings of several informa tion retrieval and Web search and mining conferences, including the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), the International World Wide Web Conference (WWW), the ACM Interna tional Conference on Web Search and Data Mining (WSDM), the ACM Conference on Information and Knowledge Management (CIKM), the European Conference on Infor mation Retrieval (ECIR), the Text Retrieval Conference (TREC), and the ACM/IEEE Joint Conference on Digital Libraries (JCDL). Other sources of publication include major information retrieval, information systems, and Web journals, such as *Journal of Information Retrieval, ACM Transactions on Information Systems (TOIS)*, *Informa tion Processing and Management*, *Knowledge and Information Systems (KAIS)*, and *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Getting to Know Your Data 2

It's tempting to jump straight into mining, but first, we need to get the data ready. This involves having a closer look at attributes and data values. Real-world data are typically noisy, enormous in volume (often several gigabytes or more), and may originate from a hodge podge of heterogenous sources. This chapter is about getting familiar with your data. Knowledge about your data is useful for data preprocessing (see Chapter 3), the first major task of the data mining process. You will want to know the following: What are the types of *attributes* or fields that make up your data? What kind of values does each attribute have? Which attributes are discrete, and which are continuous-valued? What do the data *look like*? How are the values distributed? Are there ways we can visualize the data to get a better sense of it all? Can we spot any outliers? Can we measure the similarity of some data objects with respect to others? Gaining such insight into the data will help with the subsequent analysis.

"So what can we learn about our data that's helpful in data preprocessing?" We begin in Section 2.1 by studying the various attribute types. These include nominal attributes, binary attributes, ordinal attributes, and numeric attributes. Basic *statistical descriptions* can be used to learn more about each attribute's values, as described in Section 2.2. Given a *temperature* attribute, for example, we can determine its **mean** (average value), **median** (middle value), and **mode** (most common value). These are **measures of central tendency**, which give us an idea of the "middle" or center of distribution.

Knowing such basic statistics regarding each attribute makes it easier to fill in missing values, smooth noisy values, and spot outliers during data preprocessing. Knowledge of the attributes and attribute values can also help in fixing inconsistencies incurred dur ing data integration. Plotting the measures of central tendency shows us if the data are symmetric or skewed. Quantile plots, histograms, and scatter plots are other graphic dis plays of basic statistical descriptions. These can all be useful during data preprocessing and can provide insight into areas for mining.

The field of data visualization provides many additional techniques for viewing data through graphical means. These can help identify relations, trends, and biases "hidden" in unstructured data sets. Techniques may be as simple as scatter-plot matrices (where

two attributes are mapped onto a 2-D grid) to more sophisticated methods such as tree maps (where a hierarchical partitioning of the screen is displayed based on the attribute values). Data visualization techniques are described in Section 2.3.

Finally, we may want to examine how similar (or dissimilar) data objects are. For example, suppose we have a database where the data objects are patients, described by their symptoms. We may want to find the similarity or dissimilarity between individ ual patients. Such information can allow us to find clusters of like patients within the data set. The similarity/dissimilarity between objects may also be used to detect out liers in the data, or to perform nearest-neighbor classification. (Clustering is the topic of Chapters 10 and 11, while nearest-neighbor classification is discussed in Chapter 9.) There are many measures for assessing similarity and dissimilarity. In general, such mea sures are referred to as proximity measures. Think of the proximity of two objects as a function of the *distance* between their attribute values, although proximity can also be calculated based on probabilities rather than actual distance. Measures of data proximity are described in Section 2.4.

In summary, by the end of this chapter, you will know the different attribute types and basic statistical measures to describe the central tendency and dispersion (spread) of attribute data. You will also know techniques to visualize attribute distributions and how to compute the similarity or dissimilarity between objects.

2.1 Data Objects and Attribute Types

Data sets are made up of data objects. A **data object** represents an entity—in a sales database, the objects may be customers, store items, and sales; in a medical database, the objects may be patients; in a university database, the objects may be students, professors, and courses. Data objects are typically described by attributes. Data objects can also be referred to as *samples, examples, instances, data points*, or *objects*. If the data objects are stored in a database, they are *data tuples*. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes. In this section, we define attributes and look at the various attribute types.

2.1.1 What Is an Attribute?

An **attribute** is a data field, representing a characteristic or feature of a data object. The nouns *attribute*, *dimension*, *feature*, and *variable* are often used interchangeably in the literature. The term *dimension* is commonly used in data warehousing. Machine learning literature tends to use the term *feature*, while statisticians prefer the term *variable*. Data mining and database professionals commonly use the term *attribute*, and we do here as well. Attributes describing a customer object can include, for example, *customer ID*, *name*, and *address*. Observed values for a given attribute are known as *observations*. A set of attributes used to describe a given object is called an *attribute vector* (or *feature vec tor*). The distribution of data involving one attribute (or variable) is called *univariate*. A *bivariate* distribution involves two attributes, and so on.

The **type** of an attribute is determined by the set of possible values—nominal, binary, ordinal, or numeric—the attribute can have. In the following subsections, we introduce each type.

2.1.2 Nominal Attributes

Nominal means “relating to names.” The values of a **nominal attribute** are symbols or *names of things*. Each value represents some kind of category, code, or state, and so nomi nal attributes are also referred to as **categorical**. The values do not have any meaningful order. In computer science, the values are also known as *enumerations*.

Example 2.1 Nominal attributes. Suppose that *hair color* and *marital status* are two attributes describing *person* objects. In our application, possible values for *hair color* are *black*, *brown*, *blond*, *red*, *auburn*, *gray*, and *white*. The attribute *marital status* can take on the values *single, married, divorced*, and *widowed*. Both *hair color* and *marital status* are nominal attributes. Another example of a nominal attribute is *occupation*, with the values *teacher, dentist, programmer, farmer*, and so on.

Although we said that the values of a nominal attribute are symbols or “names of things,” it is possible to represent such symbols or “names” with numbers. With *hair color*, for instance, we can assign a code of 0 for *black*, 1 for *brown*, and so on. Another example is *customor ID*, with possible values that are all numeric. However, in such cases, the numbers are not intended to be used quantitatively. That is, mathe matical operations on values of nominal attributes are not meaningful. It makes no sense to subtract one customer ID number from another, unlike, say, subtracting an age value from another (where *age* is a numeric attribute). Even though a nominal attribute may have integers as values, it is not considered a numeric attribute because the inte gers are not meant to be used quantitatively. We will say more on numeric attributes in Section 2.1.5.

Because nominal attribute values do not have any meaningful order about them and are not quantitative, it makes no sense to find the mean (average) value or median (middle) value for such an attribute, given a set of objects. One thing that is of inter est, however, is the attribute’s most commonly occurring value. This value, known as the *mode*, is one of the measures of central tendency. You will learn about measures of central tendency in Section 2.2.

2.1.3 Binary Attributes

A **binary attribute** is a nominal attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is absent, and 1 means that it is present. Binary attributes are referred to as **Boolean** if the two states correspond to *true* and *false*.

Example 2.2 Binary attributes. Given the attribute *smoker* describing a *patient* object, 1 indicates that the patient smokes, while 0 indicates that the patient does not. Similarly, suppose

the patient undergoes a medical test that has two possible outcomes. The attribute *medical test* is binary, where a value of 1 means the result of the test for the patient is positive, while 0 means the result is negative.

A binary attribute is **symmetric** if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1. One such example could be the attribute *gender* having the states *male* and *female*.

A binary attribute is **asymmetric** if the outcomes of the states are not equally impor tant, such as the *positive* and *negative* outcomes of a medical test for HIV. By convention, we code the most important outcome, which is usually the rarest one, by 1 (e.g., *HIV positive*) and the other by 0 (e.g., *HIV negative*).

2.1.4 Ordinal Attributes

An **ordinal attribute** is an attribute with possible values that have a meaningful order or *ranking* among them, but the magnitude between successive values is not known.

Example 2.3 Ordinal attributes. Suppose that *drink size* corresponds to the size of drinks available at a fast-food restaurant. This nominal attribute has three possible values: *small, medium*, and *large*. The values have a meaningful sequence (which corresponds to increasing drink size); however, we cannot tell from the values *how much* bigger, say, a medium is than a large. Other examples of ordinal attributes include *grade* (e.g., *A+, A, A−, B+*, and so on) and *professional rank*. Professional ranks can be enumerated in a sequential order: for example, *assistant*, *associate*, and *full* for professors, and *private, private first class, specialist, corporal, and sergeant* for army ranks.

Ordinal attributes are useful for registering subjective assessments of qualities that cannot be measured objectively; thus ordinal attributes are often used in surveys for ratings. In one survey, participants were asked to rate how satisfied they were as cus tomers. Customer satisfaction had the following ordinal categories: *0: very dissatisfied, 1: somewhat dissatisfied, 2: neutral, 3: satisfied*, and *4: very satisfied.*

Ordinal attributes may also be obtained from the discretization of numeric quantities by splitting the value range into a finite number of ordered categories as described in Chapter 3 on data reduction.

The central tendency of an ordinal attribute can be represented by its mode and its median (the middle value in an ordered sequence), but the mean cannot be defined. Note that nominal, binary, and ordinal attributes are *qualitative*. That is, they *describe* a feature of an object without giving an actual size or quantity. The values of such qualitative attributes are typically words representing categories. If integers are used, they represent computer codes for the categories, as opposed to measurable quantities (e.g., 0 for *small* drink size, 1 for *medium*, and 2 for *large*). In the following subsec tion we look at numeric attributes, which provide *quantitative* measurements of an object.

2.1.5 Numeric Attributes

A **numeric attribute** is *quantitative*; that is, it is a measurable quantity, represented in integer or real values. Numeric attributes can be *interval-scaled* or *ratio-scaled*.

Interval-Scaled Attributes

Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be positive, 0, or negative. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the *difference* between values.

Example 2.4 Interval-scaled attributes. A *temperature* attribute is interval-scaled. Suppose that we have the outdoor *temperature* value for a number of different days, where each day is an object. By ordering the values, we obtain a ranking of the objects with respect to *temperature*. In addition, we can quantify the difference between values. For example, a temperature of 20°C is five degrees higher than a temperature of 15°C. Calendar dates are another example. For instance, the years 2002 and 2010 are eight years apart.

Temperatures in Celsius and Fahrenheit do not have a true zero-point, that is, neither 0°C nor 0°F indicates “no temperature.” (On the Celsius scale, for example, the unit of measurement is 1/100 of the difference between the melting temperature and the boiling temperature of water in atmospheric pressure.) Although we can compute the *difference* between temperature values, we cannot talk of one temperature value as being a *multiple* of another. Without a true zero, we cannot say, for instance, that 10°C is twice as warm as 5°C. That is, we cannot speak of the values in terms of ratios. Similarly, there is no true zero-point for calendar dates. (The year 0 does not correspond to the beginning of time.) This brings us to ratio-scaled attributes, for which a true zero-point exits.

Because interval-scaled attributes are numeric, we can compute their mean value, in addition to the median and mode measures of central tendency.

Ratio-Scaled Attributes

A **ratio-scaled attribute** is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value. In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.

Example 2.5 **Ratio-scaled attributes.** Unlike temperatures in Celsius and Fahrenheit, the Kelvin (K) temperature scale has what is considered a true zero-point (0°K = −273.15°C): It is the point at which the particles that comprise matter have zero kinetic energy. Other examples of ratio-scaled attributes include *count* attributes such as *years of experience* (e.g., the objects are employees) and *number of words* (e.g., the objects are documents). Additional examples include attributes to measure weight, height, latitude and longitude