Data Mining: Concepts and Techniques

(2nd Edition)

Solution Manual

Jiawei Han and **Micheline Kamber**

The University of Illinois at Urbana-Champaign

Note: For Instructors' reference only. Do not copy! Do not distribute!

Contents

Preface

For a rapidly evolving field like data mining, it is difficult to compose "typical" exercises and even more difficult to work out "standard" answers. Some of the exercises in *Data Mining: Concepts and Techniques* are themselves good research topics that may lead to future Master or Ph.D. theses. Therefore, our solution manual was prepared as a teaching aid to be used with a grain of salt. You are welcome to enrich this manual by suggesting additional interesting exercises and/or providing more thorough, or better alternative solutions. It is also possible that the solutions may contain typos or errors. If you should notice any, please feel free to point them out by sending your suggestions to *hanj@cs.uiuc.edu*. We appreciate your suggestions.

Acknowledgements

First, we would like to express our sincere thanks to Jian Pei and the following students in the CMPT-459 class "Data Mining and Data Warehousing" at Simon Fraser University in the semester of Fall 2000: Denis M. C. Chai, Meloney H.-Y. Chang, James W. Herdy, Jason W. Ma, Jiuhong Xu, Chunyan Yu, and Ying Zhou. They have all contributed substantially to the work on the solution manual of first edition of this book. For those questions that also appear in the first edition, the answers in this current solution manual are largely based on those worked out in the preparation of the first edition. Second, we would like to thank two Ph.D. candidates, Deng Cai and Hector Gonzalez, who have served as assistants in the teaching of our data mining course: *CS412: Introduction to Data Mining*, in the Department of Computer Science, University of Illinois at Urbana-Champaign, in Fall 2005. They have helped preparing and compiling the answers for some of the exercise questions. Moreover, our thanks go to several students, , whose answers to the class assignments have contributed to the improvements of this solution manual.

Chapter 1

Introduction

1.11 Exercises

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

   (a) Is it another hype?

   (b) Is it a simple transformation of technology developed from databases, statistics, and machine learning?

   (c) Explain how the evolution of database technology led to data mining.

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

   Answer:

   Data mining refers the process or method that extracts or “mines” interesting knowledge or patterns from large amounts of data.

   (a) Is it another hype?

   Data mining is not another hype. Instead, the need for data mining has arisen due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. Thus, data mining can be viewed as the result of the natural evolution of information technology.

   (b) Is it a simple transformation of technology developed from databases, statistics, and machine learning?

   No. Data mining is more than a simple transformation of technology developed from databases, statistics, and machine learning. Instead, data mining involves an integration, rather than a simple transformation, of techniques from multiple disciplines such as database technology, statistics, machine learning, high-performance computing, pattern recognition, neural networks, data visualization, information retrieval, image and signal processing, and spatial data analysis.

   (c) Explain how the evolution of database technology led to data mining.

   Database technology began with the development of data collection and database creation mechanisms that led to the development of effective mechanisms for data management including data storage and retrieval, and query and transaction processing. The large number of database systems offering query and transaction processing eventually and naturally led to the need for data analysis and understanding. Hence, data mining began its development out of this necessity.

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

   The steps involved in data mining when viewed as a process of knowledge discovery are as follows:

   - **Data cleaning**, a process that removes or transforms noise and inconsistent data
   - **Data integration**, where multiple data sources may be combined

- **Data selection**, where data relevant to the analysis task are retrieved from the database
- **Data transformation**, where data are transformed or consolidated into forms appropriate for mining
- **Data mining**, an essential process where intelligent and efficient methods are applied in order to extract patterns
- **Pattern evaluation**, a process that identifies the truly interesting patterns representing knowledge based on some interestingness measures
- **Knowledge presentation**, where visualization and knowledge representation techniques are used to present the mined knowledge to the user

■

2. Present an example where data mining is crucial to the success of a business. What *data mining functions* does this business need? Can they be performed alternatively by data query processing or simple statistical analysis?

Answer:

A department store, for example, can use data mining to assist with its target marketing mail campaign. Using data mining functions such as association, the store can use the mined strong association rules to determine which products bought by one group of customers are likely to lead to the buying of certain other products. With this information, the store can then mail marketing materials only to those kinds of customers who exhibit a high likelihood of purchasing additional products. Data query processing is used for data or information retrieval and does not have the means for finding association rules. Similarly, simple statistical analysis cannot handle large amounts of data such as those of customer records in a department store.

■

3. Suppose your task as a software engineer at *Big-University* is to design a data mining system to examine their university course database, which contains the following information: the name, address, and status (e.g., undergraduate or graduate) of each student, the courses taken, and their cumulative grade point average (GPA). Describe the *architecture* you would choose. What is the purpose of each component of this architecture?

Answer:

A data mining architecture that can be used for this application would consist of the following major components:

- A **database, data warehouse, or other information repository**, which consists of the set of databases, data warehouses, spreadsheets, or other kinds of information repositories containing the student and course information.
- A **database or data warehouse server** which fetches the relevant data based on users' data mining requests.
- A **knowledge base** that contains the domain knowledge used to guide the search or to evaluate the interestingness of resulting patterns. For example, the knowledge base may contain metadata which describes data from multiple heterogeneous sources.
- A **data mining engine**, which consists of a set of functional modules for tasks such as classification, association, classification, cluster analysis, and evolution and deviation analysis.
- A **pattern evaluation module** that works in tandem with the data mining modules by employing interestingness measures to help focus the search towards interestingness patterns.
- **A graphical user interface** that allows the user an interactive approach to the data mining system.

■

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

Answer:

- Differences between a data warehouse and a database: A **data warehouse** is a repository of information collected from multiple sources, over a history of time, stored under a unified schema, and used for data analysis and decision support; whereas a **database**, is a collection of interrelated data that represents the current status of the stored data. There could be multiple heterogeneous databases where the schema of one database may not agree with the schema of another. A database system supports ad-hoc query and on-line transaction processing. For more details, please refer to the section "Differences between operational database systems and data warehouses."
- Similarities between a data warehouse and a database: Both are repositories of information, storing huge amounts of persistent data.

■

5. Briefly describe the following *advanced database systems* and applications: object-relational databases, spatial databases, text databases, multimedia databases, the World Wide Web.

Answer:

- **An objected-oriented database** is designed based on the object-oriented programming paradigm where data are a large number of objects organized into classes and class hierarchies. Each entity in the database is considered as an object. The object contains a set of variables that describe the object, a set of messages that the object can use to communicate with other objects or with the rest of the database system, and a set of methods where each method holds the code to implement a message.
- **A spatial database** contains spatial-related data, which may be represented in the form of raster or vector data. Raster data consists of n-dimensional bit maps or pixel maps, and vector data are represented by lines, points, polygons or other kinds of processed primitives, Some examples of spatial databases include geographical (map) databases, VLSI chip designs, and medical and satellite images databases.
- **A text database** is a database that contains text documents or other word descriptions in the form of long sentences or paragraphs, such as product specifications, error or bug reports, warning messages, summary reports, notes, or other documents.
- **A multimedia database** stores images, audio, and video data, and is used in applications such as picture content-based retrieval, voice-mail systems, video-on-demand systems, the World Wide Web, and speech-based user interfaces.
- The **World-Wide Web** provides rich, world-wide, on-line information services, where data objects are linked together to facilitate interactive access. Some examples of distributed information services associated with the World-Wide Web include America Online, Yahoo!, AltaVista, and Prodigy.

■

6. Define each of the following *data mining functionalities*: characterization, discrimination, association and correlation analysis, classification, prediction, clustering, and evolution analysis. Give examples of each data mining functionality, using a real-life database that you are familiar with.

Answer:

- **Characterization** is a summarization of the general characteristics or features of a target class of data. For example, the characteristics of students can be produced, generating a profile of all the University first year computing science students, which may include such information as a high GPA and large number of courses taken.
- **Discrimination** is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes. For example, the general features of students with high GPA's may be compared with the general features of students with low GPA's. The resulting description could be a general comparative profile of the students such as 75% of the students with high GPA's are fourth-year computing science students while 65% of the students with low GPA's are not.

- **Association** is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data. For example, a data mining system may find association rules like

$$major(X, \text{“}computing\ science\text{””}) \Rightarrow owns(X, \text{“}personal\ computer\text{”}) \quad [support = 12\%, confidence = 98\%]$$

  where X is a variable representing a student. The rule indicates that of the students under study, 12% (**support**) major in computing science and own a personal computer. There is a 98% probability (**confidence**, or certainty) that a student in this group owns a personal computer.

- **Classification** differs from **prediction** in that the former is to construct a set of models (or functions) that describe and distinguish data class or concepts, whereas the latter is to predict some missing or unavailable, and often numerical, data values. Their similarity is that they are both tools for prediction: Classification is used for predicting the class label of data objects and prediction is typically used for predicting missing numerical data values.

- **Clustering** analyzes data objects without consulting a known class label. The objects are clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Each cluster that is formed can be viewed as a class of objects. Clustering can also facilitate *taxonomy formation*, that is, the organization of observations into a hierarchy of classes that group similar events together.

- **Data evolution analysis** describes and models regularities or trends for objects whose behavior changes over time. Although this may include characterization, discrimination, association, classification, or clustering of time-related data, distinct features of such an analysis include time-series data analysis, sequence or periodicity pattern matching, and similarity-based data analysis.

■

7. What is the difference between discrimination and classification? Between characterization and clustering? Between classification and prediction? For each of these pairs of tasks, how are they similar?

   Answer:

   - **Discrimination** differs from **classification** in that the former refers to a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes, while the latter is the process of finding a set of models (or functions) that describe and distinguish data classes or concepts for the purpose of being able to use the model to predict the class of objects whose class label is unknown. Discrimination and classification are similar in that they both deal with the analysis of class data objects.
   - **Characterization** differs from **clustering** in that the former refers to a summarization of the general characteristics or features of a target class of data while the latter deals with the analysis of data objects without consulting a known class label. This pair of tasks is similar in that they both deal with grouping together objects or data that are related or have high similarity in comparison to one another.
   - **Classification** differs from **prediction** in that the former is the process of finding a set of models (or functions) that describe and distinguish data class or concepts while the latter predicts missing or unavailable, and often numerical, data values. This pair of tasks is similar in that they both are tools for prediction: Classification is used for predicting the class label of data objects and prediction is typically used for predicting missing numerical data values.

   ■

8. Based on your observation, 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?

   Answer:

There is no standard answer for this question and one can judge the quality of an answer based on the freshness and quality of the proposal. For example, one may propose *partial periodicity* as a new kind of knowledge, where a pattern is partial periodic if only some offsets of a certain time period in a time series demonstrate some repeating behavior.

■

9. List and describe the five *primitives* for specifying a data mining task.

   Answer:

   The five primitives for specifying a data-mining task are:

   - **Task-relevant data:** This primitive specifies the data upon which mining is to be performed. It involves specifying the database and tables or data warehouse containing the relevant data, conditions for selecting the relevant data, the relevant attributes or dimensions for exploration, and instructions regarding the ordering or grouping of the data retrieved.
   - **Knowledge type to be mined:** This primitive specifies the specific data mining function to be performed, such as characterization, discrimination, association, classification, clustering, or evolution analysis. As well, the user can be more specific and provide pattern templates that all discovered patterns must match. These templates, or metapatterns (also called metarules or metaqueries), can be used to guide the discovery process.
   - **Background knowledge:** This primitive allows users to specify knowledge they have about the domain to be mined. Such knowledge can be used to guide the knowledge discovery process and evaluate the patterns that are found. Of the several kinds of background knowledge, this chapter focuses on concept hierarchies.
   - **Pattern interestingness measure:** This primitive allows users to specify functions that are used to separate uninteresting patterns from knowledge and may be used to guide the mining process, as well as to evaluate the discovered patterns. This allows the user to confine the number of uninteresting patterns returned by the process, as a data mining process may generate a large number of patterns. Interestingness measures can be specified for such pattern characteristics as simplicity, certainty, utility and novelty.
   - **Visualization of discovered patterns:** This primitive refers to the form in which discovered patterns are to be displayed. In order for data mining to be effective in conveying knowledge to users, data mining systems should be able to display the discovered patterns in multiple forms such as rules, tables, cross tabs (cross-tabulations), pie or bar charts, decision trees, cubes or other visual representations.

   ■

10. Describe why *concept hierarchies* are useful in data mining.

    Answer:

    Concept hierarchies define a sequence of mappings from a set of lower-level concepts to higher-level, more general concepts and can be represented as a set of nodes organized in a tree, in the form of a lattice, or as a partial order. They are useful in data mining because they allow the discovery of knowledge at multiple levels of abstraction and provide the structure on which data can be generalized (rolled-up) or specialized (drilled-down). Together, these operations allow users to view the data from different perspectives, gaining further insight into relationships hidden in the data. Generalizing has the advantage of compressing the data set, and mining on a compressed data set will require fewer I/O operations. This will be more efficient than mining on a large, uncompressed data set.

    ■

11. *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. Taking fraudulence detection as an example, propose two methods that can be used to detect outliers and discuss which one is more reliable.

    Answer:

- Using clustering techniques: After clustering, the different clusters represent the different kinds of data (transactions). The outliers are those data points that do not fall in any cluster. In such scenario, density based clustering methods might be a good choice.
- Using prediction (or regression) techniques: Constructed a probability (regression) model based on all the data. Those data which the real values is far from the predict values can be judged as outliers.

Outlier detection based on clustering techniques might be more reliable. Clustering is unsupervised, we can make no assumption of the data distribution (Density based methods). The regression (prediction) methods need us to make some assumptions of the data distribution.

■

12. Recent applications pay special attention to spatiotemporal data streams. A *spatiotemporal data stream* contains spatial information that changes over time, and is in the form of stream data, i.e., the data flow in-and-out like possibly infinite streams.

(a) Present three application examples of spatiotemporal data streams.

(b) Discuss what kind of interesting knowledge can be mined from such data streams, with limited time and resources.

(c) Identify and discuss the major challenges in spatiotemporal data mining.

(d) Using one application example, sketch a method to mine one kind of knowledge from such stream data efficiently.

Answer:

(a) Present three application examples of spatiotemporal data streams.

i. Sequences of sensor images of a geographical region along time.

ii. The climate images from satellites.

iii. Data that describes the evolution of natural phenomena, such as forest coverage, forest fire, and so on.

(b) Discuss what kind of interesting knowledge can be mined from such data streams, with limited time and resources.

The knowledge that can be mined from spatiotemporal data streams really depends on the applications. However, one unique type of knowledge about this kind of data is the patterns of spatial change with respect to the time. For example, the changing of the traffic status of several highway junctions in a city, from the early morning to rush hours and back to off-peak hours, can show clearly where the traffics come from and go to and hence, would help the traffic officer plan effective alternative lanes in order to reduce the traffic load. A sudden appearance of a point in the spectrum space image might inform there is a new planet creating. The changing of humidity, temperature, and pressure in the climate data might reveal some patterns of how a new typhoon is created.

(c) Identify and discuss the major challenges in spatiotemporal data mining.

One major challenge is how to deal with the continuing large-scale data. Since the data keep flowing in and each snapshot of data is usually huge (e.g., the spectrum image of space), it is almost impossible to store all the data. Some aggregation or compression techniques might have to be applied, and old raw data might have to be dropped. Mining under such aggregated (or lossy) data is challenging. In addition, some patterns may occur with respect to a long time period, while the data cannot be kept for such a long duration to reveal these patterns. The spatial data sensed may not be so accurate, so the algorithms must have high tolerance with noise.

(d) Using one application example, sketch a method to mine one kind of knowledge from such stream data efficiently.

Take the space image as the application. We seek to observe whether there is any new planet creating or any old planet disappearing. This is a change detection problem. Since the image frames keep coming, $f_1, \ldots, f_t, f_{t+1}, \ldots$, we can simplify the overall problem as detecting whether any planet appears or disappears between two consecutive image frame f_t and f_{t+1}. The algorithm can be sketched as:

i. For each incoming frame f_{t+1}, compare it with the previous frame f_t.
   A. Match the planets in f_{t+1} with f_t.
   B. Detect whether any unmatched planet in the two frames.
   C. If yes– report planet appearance if it is in new frame or disappearance if it is in old frame.

In fact, matching between two frames might not be easy because the earth is rotating and thus, the sensed data might have slight variation. Some advanced techniques from image processing might be borrowed.

The overall skeleton of the algorithm is simple. Each new coming image frame is only compared with the previous one, satisfying the time and resource constraint. The reported change would be useful since it is almost impossible for astronomers to dig into every frame to find out whether there is any planet appearing or disappearing.

■

13. Describe the differences between the following approaches for the integration of a data mining system with a database or data warehouse system: *no coupling, loose coupling, semitight coupling,* and *tight coupling.* State which approach you think is the most popular, and why.

Answer:

The differences between the following architectures for the integration of a data mining system with a database or data warehouse system are as follows.

- **No coupling:** The data mining system uses sources such as flat files to obtain the initial data set to be mined since no database system or data warehouse system functions are implemented as part of the process. Thus, this architecture represents a poor design choice.
- **Loose coupling:** The data mining system is not integrated with the database or data warehouse system beyond their use as the source of the initial data set to be mined, and possible use in storage of the results. Thus, this architecture can take advantage of the flexibility, efficiency and features such as indexing that the database and data warehousing systems may provide. However, it is difficult for loose coupling to achieve high scalability and good performance with large data sets as many such systems are memory-based.
- **Semitight coupling:** Some of the data mining primitives such as aggregation, sorting or precomputation of statistical functions are efficiently implemented in the database or data warehouse system, for use by the data mining system during mining-query processing. Also, some frequently used intermediate mining results can be precomputed and stored in the database or data warehouse system, thereby enhancing the performance of the data mining system.
- **Tight coupling:** The database or data warehouse system is fully integrated as part of the data mining system and thereby provides optimized data mining query processing. Thus, the data mining subsystem is treated as one functional component of an information system. This is a highly desirable architecture as it facilitates efficient implementations of data mining functions, high system performance, and an integrated information processing environment.

From the descriptions of the architectures provided above, it can be seen that tight coupling is the best alternative without respect to technical or implementation issues. However, as much of the technical infrastructure needed in a tightly coupled system is still evolving, implementation of such a system is non-trivial. Therefore, the most popular architecture is currently semitight coupling as it provides a compromise between loose and tight coupling.

■

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

Answer:

Challenges to data mining regarding data mining methodology and user interaction issues include the following: mining different kinds of knowledge in databases, interactive mining of knowledge at multiple levels of abstraction, incorporation of background knowledge, data mining query languages and ad hoc data mining, presentation and visualization of data mining results, handling noisy or incomplete data, and pattern evaluation. Below are the descriptions of the first three challenges mentioned:

- **Mining different kinds of knowledge in databases:** Different users are interested in different kinds of knowledge and will require a wide range of data analysis and knowledge discovery tasks such as data characterization, discrimination, association, classification, clustering, trend and deviation analysis, and similarity analysis. Each of these tasks will use the same database in different ways and will require different data mining techniques.
- **Interactive mining of knowledge at multiple levels of abstraction:** Interactive mining, with the use of OLAP operations on a data cube, allows users to focus the search for patterns, providing and refining data mining requests based on returned results. The user can then interactively view the data and discover patterns at multiple granularities and from different angles.
- **Incorporation of background knowledge:** Background knowledge, or information regarding the domain under study such as integrity constraints and deduction rules, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction. This helps to focus and speed up a data mining process or judge the interestingness of discovered patterns.

■

15. What are the major challenges of mining a huge amount of data (such as billions of tuples) in comparison with mining a small amount of data (such as a few hundred tuple data set)?

Answer:

One challenge to data mining regarding performance issues is the *efficiency and scalability* of data mining algorithms. Data mining algorithms must be efficient and scalable in order to effectively extract information from large amounts of data in databases within predictable and acceptable running times. Another challenge is the *parallel, distributed, and incremental* processing of data mining algorithms. The need for parallel and distributed data mining algorithms has been brought about by the huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods. Due to the high cost of some data mining processes, incremental data mining algorithms incorporate database updates without the need to mine the entire data again from scratch.

■

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

Answer:

The field of bio-informatics in-turn encompasses many other sub-fields like genomics, proteomics, molecular biology, and chemi-informatics. Each of these individual sub-fields in-turn has many research challenges associated with it. Here, I have summarized some major research challenges of data mining in the field of Bio-informatics have been outlined as given below:

- **Data explosion**: The biological data is growing at an exponential rate. In fact, it has been estimated that the genomic and proteomic data is doubling every 12 months. Also, most of this data is scattered around in unstructured and nonstandard form in various different databases throughout the research community. Many of the biological experiments do not yield exact results and are prone to errors because it is very difficult to model exact biological conditions and processes. For example, the structure of a protein is not rigid and is dependent on its environment. Hence, the structures determined by NMR or crystallography experiments may not be the exact structure of the protein. Also, since these experiments are performed in parallel by many institutions and scientists, they may all come up with slightly different structures. The consolidation and validation of these conflicting data is a difficult challenge by itself. Some research labs have come up with public domain repositories of data (Example: PDB, etc.) These have become very popular in the past few years. However, due to concerns of Intellectual Property, lot of useful biological information is buried in proprietary databases within large pharmaceutical companies.

- **Text-mining from research publications/repositories**: Most of the data generated in the biological research community is through experiments. Most of the results are published. But they are seldom recorded into databases with experiment details (who, when, how etc.). Hence, a lot of useful information is buried in published and un-published literature. This, in-turn, has given rise to the need for development of text-mining systems. For example, a lot of experimental results regarding protein interactions have been published in literature. Mining this information can give crucial insights into biological pathways and help predict potential interactions. The extraction and development of domain-specific Ontologies is also another related research challenge.
- **Mining large databases of compounds/molecules**: The major steps in a drug discovery phase include target identification, target validation, lead discovery, and lead-optimization. The most time-consuming stage is the lead discovery phase; in which, large database of compounds are needed to be mined for identify potential lead candidates that will suitably interact with the potential target. Currently, due to the lack of effective data-mining systems, this stage involves many trial-and-error iterations of wet-lab or protein-assay experiments. These experiments are highly time-consuming and costly. Hence, one of the current challenges in bio-informatics, include the development of intelligent and computational data mining systems that can eliminate false positives and generate more true positives before the wet-lab experimentation stage. This task is particularly challenging, because this would involve the development of a mining/screening system that can identify compounds that can dock better with the target compound. The docking problem is especially a tricky problem, as it is governed by many physical interactions at the molecular level. There have been some progress made in pair-wise docking area, where large time-consuming Molecular Dynamics Simulation(MD) based optimization methods can predict docking to a good degree of success. The main problem is the large solution space generated by the complex interactions at the molecular level. Still, molecular docking problem remains a fairly unsolved problem. The major research challenges in mining of these interactions include the development of fast and fairly accurate methods/algorithms for screening and ranking these compounds/molecules based on their ability to interact with a given compound/molecule. Some other related research areas also include, protein classification system based on structure and function.
- **Pattern Analysis and classification of micro-array data**: Owing to the progress made in the past decade or so. There has been a lot of progress made in the area of development of algorithms for analysis of genomic data. There are fairly well-developed statistical and other methods that are available for analysis of genomic data. A large research community in data-mining is focusing on adopting these pattern analysis and classification methods for mining micro-array and gene-expression data.

Data Stream

Data stream analysis presents multiple challenges. First, data streams are continuously flowing in and out as well as changing dynamically. The data analysis system that will successfully take care of this type of data needs to be highly efficient, very fast, and able to adapt to changing patterns that might emerge. Also, another major challenge is the size of this data as it is huge or infinite. In addition, a further challenge may be with the single or small number of scans that would be allowed.

■

Chapter 2

Data Preprocessing

2.8 Exercises

1. *Data quality* can be assessed in terms of accuracy, completeness, and consistency. Propose two other dimensions of data quality.

   Answer:

   Two other dimensions that can be used to assess the quality of data can be taken from the following: *timeliness, believability, value added*, *interpretability* and *accessability.* These can be used to assess quality with regard to the following factors:

   - **Timeliness:** Data must be available within a time frame that allows it to be useful for decision making.
   - **Believability:** Data values must be within the range of possible results in order to be useful for decision making.
   - **Value added:** Data must provide additional value in terms of information that offsets the cost of collecting and accessing it.
   - **Interpretability:** Data must not be so complex that the effort to understand the information it provides exceeds the benefit of its analysis.
   - **Accessability:** Data must be accessable so that the effort to collect it does not exceed the benefit from its use.

   ■

2. Suppose that the values for a given set of data are grouped into intervals. The intervals and corresponding frequencies are as follows.

age	*frequency*
1-5	200
5-15	450
15-20	300
20-50	1500
50-80	700
80-110	44

   Compute an *approximate median* value for the data.

   Answer:

   $L_1 = 20$, n= 3194, $(\sum_f)_l = 950$, $freq\_median = 1500$, width = 30, median = 32.94 years.

   ■

3. Give three additional commonly used statistical measures (i.e., not illustrated in this chapter) for the characterization of *data dispersion,* and discuss how they can be computed efficiently in large databases.

Answer:

Data dispersion, also known as variance analysis, is the degree to which numeric data tend to spread and can be characterized by such statistical measures as *mean deviation, measures of skewness* and the *coefficient of variation.*

The **mean deviation** is defined as the arithmetic mean of the absolute deviations from the means and is calculated as:

$$mean\ deviation = \frac{\sum_{i=1}^{n} |x - \bar{x}|}{n} \tag{2.1}$$

where, $\bar{x}$ is the arithmetic mean of the values and n is the total number of values. This value will be greater for distributions with a larger spread.

A common **measure of skewness** is:

$$\frac{\bar{x} - mode}{s} \tag{2.2}$$

which indicates how far (in standard deviations, s) the mean ($\bar{x}$) is from the mode and whether it is greater or less than the mode.

The **coefficient of variation** is the standard deviation expressed as a percentage of the arithmetic mean and is calculated as:

$$coefficient\ of\ variation = \frac{s}{\bar{x}} \times 100 \tag{2.3}$$

The variability in groups of observations with widely differing means can be compared using this measure.

Note that all of the input values used to calculate these three statistical measures are algebraic measures. Thus, the value for the entire database can be efficiently calculated by partitioning the database, computing the values for each of the separate partitions, and then merging theses values into an algebraic equation that can be used to calculate the value for the entire database.

The measures of dispersion described here were obtained from: Statistical Methods in Research and Production, fourth ed., Edited by Owen L. Davies and Peter L. Goldsmith, Hafner Publishing Company, NY:NY, 1972.

■

4. Suppose that the data for analysis includes the attribute *age*. The *age* values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

   (a) What is the *mean* of the data? What is the *median*?

   (b) What is the *mode* of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).

   (c) What is the *midrange* of the data?

   (d) Can you find (roughly) the first quartile ($Q1$) and the third quartile ($Q3$) of the data?

   (e) Give the *five-number summary* of the data.

   (f) Show a *boxplot* of the data.

   (g) How is a *quantile-quantile plot* different from a *quantile plot*?

Answer:

(a) What is the *mean* of the data? What is the *median*?

The (arithmetic) mean of the data is: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = 809/27 = 30$. The median (middle value of the ordered set, as the number of values in the set is odd) of the data is: 25.

(b) What is the *mode* of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).

This data set has two values that occur with the same highest frequency and is, therefore, bimodal. The modes (values occurring with the greatest frequency) of the data are 25 and 35.

(c) What is the *midrange* of the data?

The midrange (average of the largest and smallest values in the data set) of the data is: $(70+13)/2 = 41.5$

(d) Can you find (roughly) the first quartile ($Q1$) and the third quartile ($Q3$) of the data?

The first quartile (corresponding to the 25th percentile) of the data is: 20. The third quartile (corresponding to the 75th percentile) of the data is: 35.

(e) Give the *five-number summary* of the data.

The five number summary of a distribution consists of the minimum value, first quartile, median value, third quartile, and maximum value. It provides a good summary of the shape of the distribution and for this data is: 13, 20, 25, 35, 70.

(f) Show a *boxplot* of the data.

See Figure 2.1.

Figure 2.1: A boxplot of the data in Exercise 5.4.

(g) How is a *quantile-quantile plot* different from a *quantile plot*?

A quantile plot is a graphical method used to show the approximate percentage of values below or equal to the independent variable in a univariate distribution. Thus, it displays quantile information for all the data, where the values measured for the independent variable are plotted against their corresponding quantile.

A quantile-quantile plot however, graphs the quantiles of one univariate distribution against the corresponding quantiles of another univariate distribution. Both axes display the range of values measured for their corresponding distribution, and points are plotted that correspond to the quantile values of the two distributions. A line ($y = x$) can be added to the graph along with points representing where the first, second and third quantiles lie, in order to increase the graph's informational value. Points that lie above such a line indicate a correspondingly higher value for the distribution plotted on the y-axis, than for the distribution plotted on the x-axis at the same quantile. The opposite effect is true for points lying below this line.

■

5. In many applications, new data sets are incrementally added to the existing large data sets. Thus an important consideration for computing descriptive data summary is whether a measure can be computed efficiently in incremental manner. Use *count, standard deviation*, and *median* as examples to show that a distributive or algebraic measure facilitates efficient incremental computation, whereas a holistic measure does not.

Answer:

- Count: The current count can be stored as a value, and when x number of new values are added, you can easily update count by doing count+x. This is a distributive measure and is easily updated for incremental additions.
- Standard deviation: If you store the sum of the squared existing values and the count of the existing values, you can easily generate the new standard deviation using the formula provided in the book. You simply need to calculate the squared sum of the new numbers, and add that to the existing squared

sum, and update the count of the numbers, plug that into the calculation, and you get the new standard deviation. All of this is done without looking at the whole data set and is thus easy to compute.

- Median: To accurately calculate the median, you have to look at every number in your dataset. When you add a new number or numbers, you have to sort the new set and then find the median based on that new sorted set. This is much harder and thus makes incremental addition of new values difficult.

■

6. In real-world data, tuples with *missing values* for some attributes are a common occurrence. Describe various methods for handling this problem.

Answer:

The various methods for handling the problem of missing values in data tuples include:

(a) **Ignoring the tuple:** This is usually done when the class label is missing (assuming the mining task involves classification or description). This method is not very effective unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.

(b) **Manually filling in the missing value:** In general, this approach is time-consuming and may not be a reasonable task for large data sets with many missing values, especially when the value to be filled in is not easily determined.

(c) **Using a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as a label like *"Unknown,"* or $-\infty$. If missing values are replaced by, say, *"Unknown,"* then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common — that of *"Unknown."* Hence, although this method is simple, it is not recommended.

(d) **Using the attribute mean for quantitative (numeric) values or attribute mode for categorical (nominal) values:** For example, suppose that the average income of *AllElectronics* customers is $28,000. Use this value to replace any missing values for *income*.

(e) **Using the attribute mean for quantitative (numeric) values or attribute mode for categorical (nominal) values, for all samples belonging to the same class as the given tuple:** For example, if classifying customers according to *credit_risk*, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.

(f) **Using the most probable value to fill in the missing value:** This may be determined with regression, inference-based tools using Bayesian formalism, or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for *income*.

■

7. Suppose that the data for analysis include the attribute *age*. The *age* values for the data tuples are (in increasing order): 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) Use *smoothing by bin means* to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.

(b) How might you determine *outliers* in the data?

(c) What other methods are there for *data smoothing*?

Answer:

(a) Use *smoothing by bin means* to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.

The following steps are required to smooth the above data using smoothing by bin means with a bin depth of 3.

- **Step 1:** Sort the data. (This step is not required here as the data are already sorted.)
- **Step 2:** Partition the data into equidepth bins of depth 3.

  Bin 1: 13, 15, 16 Bin 2: 16, 19, 20 Bin 3: 20, 21, 22
  Bin 4: 22, 25, 25 Bin 5: 25, 25, 30 Bin 6: 33, 33, 35
  Bin 7: 35, 35, 35 Bin 8: 36, 40, 45 Bin 9: 46, 52, 70
- **Step 3:** Calculate the arithmetic mean of each bin.
- **Step 4:** Replace each of the values in each bin by the arithmetic mean calculated for the bin.

  Bin 1: 142/3, 142/3, 142/3 Bin 2: 181/3, 181/3, 181/3 Bin 3: 21, 21, 21
  Bin 4: 24, 24, 24 Bin 5: 262/3, 262/3, 262/3 Bin 6: 332/3, 332/3, 332/3
  Bin 7: 35, 35, 35 Bin 8: 401/3, 401/3, 401/3 Bin 9: 56, 56, 56

(b) How might you determine *outliers* in the data?

Outliers in the data may be detected by clustering, where similar values are organized into groups, or 'clusters'. Values that fall outside of the set of clusters may be considered outliers. Alternatively, a combination of computer and human inspection can be used where a predetermined data distribution is implemented to allow the computer to identify possible outliers. These possible outliers can then be verified by human inspection with much less effort than would be required to verify the entire initial data set.

(c) What other methods are there for *data smoothing*?

Other methods that can be used for data smoothing include alternate forms of binning such as smoothing by bin medians or smoothing by bin boundaries. Alternatively, equiwidth bins can be used to implement any of the forms of binning, where the interval range of values in each bin is constant. Methods other than binning include using regression techniques to smooth the data by fitting it to a function such as through linear or multiple regression. Also, classification techniques can be used to implement concept hierarchies that can smooth the data by rolling-up lower level concepts to higher-level concepts.

■

8. Discuss issues to consider during *data integration*.

Answer:

Data integration involves combining data from multiple sources into a coherent data store. Issues that must be considered during such integration include:

- **Schema integration:** The metadata from the different data sources must be integrated in order to match up equivalent real-world entities. This is referred to as the entity identification problem.
- **Handling redundant data:** Derived attributes may be redundant, and inconsistent attribute naming may also lead to redundancies in the resulting data set. Also, duplications at the tuple level may occur and thus need to be detected and resolved.
- **Detection and resolution of data value conflicts:** Differences in representation, scaling or encoding may cause the same real-world entity attribute values to differ in the data sources being integrated.

■

9. Suppose a hospital tested the age and body fat data for 18 randomly selected adults with the following result

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2

age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

(a) Calculate the mean, median and standard deviation of *age* and *%fat*.

(b) Draw the boxplots for *age* and *%fat*.

(c) Draw a *scatter plot* and a *q-q plot* based on these two variables.

(d) Normalize the two variables based on *z-score normalization*.

(e) Calculate the *Pearson correlation coefficient*. Are these two variables positively or negatively correlated?

Answer:

(a) Calculate the mean, median and standard deviation of *age* and *%fat*.

For the variable *age* the mean is 46.44, the median is 51, and the standard deviation is 12.85. For the variable *%fat* the mean is 28.78, the median is 30.7, and the standard deviation is 8.99.

(b) Draw the boxplots for *age* and *%fat*.

See Figure 2.2.

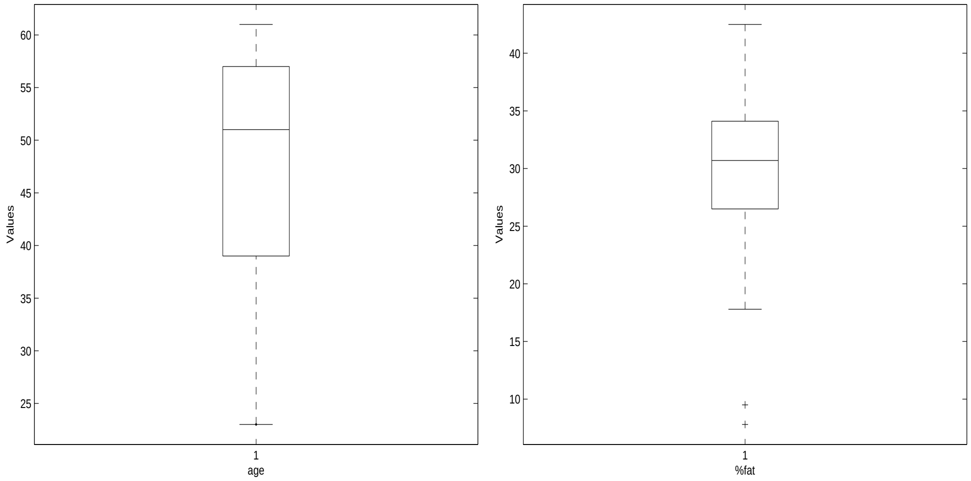

Figure 2.2: A boxplot of the variables *age* and *%fat* in Exercise 2.9.

(c) Draw a *scatter plot* and a *q-q plot* based on these two variables.

See Figure 2.3.

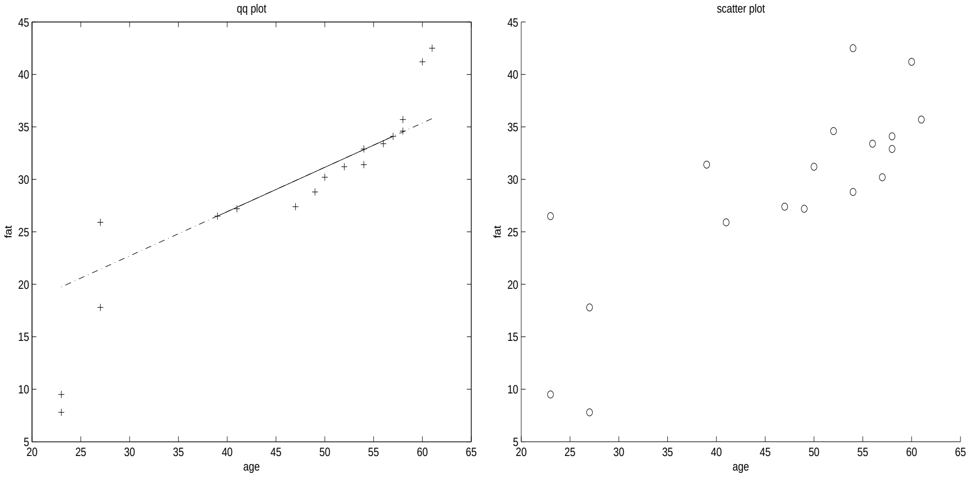

Figure 2.3: A *q-q plot* and a *scatter plot* of the variables *age* and *%fat* in Exercise 2.9.

(d) Normalize the two variables based on *z-score normalization*.

age	23	23	27	27	39	41	47	49	50	52	54	54	56	57	58	58	60	61
z-age	-1.83	-1.83	-1.51	-1.51	-0.58	-0.42	0.04	0.20	0.28	0.43	0.59	0.59	0.74	0.82	0.90	0.90	1.06	1.13
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7
z-%fat	-2.14	-0.25	-2.33	-1.22	0.29	-0.32	-0.15	-0.18	0.27	0.65	1.53	0.0	0.51	0.16	0.59	0.46	1.38	0.77

(e) Calculate the *Pearson correlation coefficient*. Are these two variables positively or negatively correlated?

The *Pearson correlation coefficient* is 0.82, the variables are positively correlated.

■

10. What are the value ranges of the following *normalization methods*?

(a) min-max normalization

(b) z-score normalization

(c) normalization by decimal scaling

Answer:

(a) min-max normalization
The rage is [new_min, new_max]

(b) z-score normalization
The range is $[(old\_min - mean)/stddev, (old\_max - mean)/stddev]$. In general the range for all possible data sets is $(-\infty, +\infty)$.

(c) normalization by decimal scaling
The range is $(-1.0, 1.0)$.

■

11. Use the two methods below to *normalize* the following group of data:

$200, 300, 400, 600, 1000$

(a) min-max normalization by setting $min = 0$ and $max = 1$

(b) z-score normalization

Answer:

(a) min-max normalization by setting $min = 0$ and $max = 1$

original data	200	300	400	600	1000
[0,1] normalized	0	0.125	0.25	0.5	1

(b) z-score normalization

original data	200	300	400	600	1000
z-score	-1.06	-0.7	-0.35	0.35	1.78

■

12. Using the data for *age* given in Exercise 2.4, answer the following:

(a) Use min-max normalization to transform the value 35 for *age* onto the range $[0.0, 1.0]$.

(b) Use z-score normalization to transform the value 35 for *age*, where the standard deviation of *age* is 12.94 years.

(c) Use normalization by decimal scaling to transform the value 35 for *age*.

(d) Comment on which method you would prefer to use for the given data, giving reasons as to why.

Answer:

(a) Use min-max normalization to transform the value 35 for *age* onto the range $[0.0, 1.0]$.

Using the corresponding equation with $min_A = 13$, $max_A = 70$, $new\_min_A = 0, new\_max_A = 1.0$, then $v = 35$ is transformed to $v' = 0.39$.

(b) Use z-score normalization to transform the value 35 for *age*, where the standard deviation of *age* is 12.94 years.

Using the corresponding equation where $A = 809/27 = 29.96$ and $\sigma_A = 12.94$, then $v = 35$ is transformed to $v' = 0.39$.

(c) Use normalization by decimal scaling to transform the value 35 for *age*.

Using the corresponding equation where $j = 2$, $v = 35$ is transformed to $v' = 0.35$.

(d) Comment on which method you would prefer to use for the given data, giving reasons as to why.

Given the data, one may prefer decimal scaling for normalization as such a transformation would maintain the data distribution and be intuitive to interpret, while still allowing mining on specific age groups. Min-max normalization has the undesired effect of not permitting any future values to fall outside the current minimum and maximum values without encountering an "out of bounds error". As it is probable that such values may be present in future data, this method is less appropriate. Also, z-score normalization transforms values into measures that represent their distance from the mean, in terms of standard deviations. It is probable that this type of transformation would not increase the information value of the attribute in terms of intuitiveness to users or in usefulness of mining results.

■

13. Use a flow chart to summarize the following procedures for *attribute subset selection*:

(a) stepwise forward selection

(b) stepwise backward elimination

(c) a combination of forward selection and backward elimination

Answer:

(a) Stepwise forward selection

See Figure 2.4.

(b) Stepwise backward elimination

See Figure 2.5.

(c) A combination of forward selection and backward elimination

See Figure 2.6.

■

14. Suppose a group of 12 *sales price* records has been sorted as follows:

$$5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215.$$

Partition them into three bins by each of the following methods.

(a) equal-frequency (equidepth) partitioning

(b) equal-width partitioning

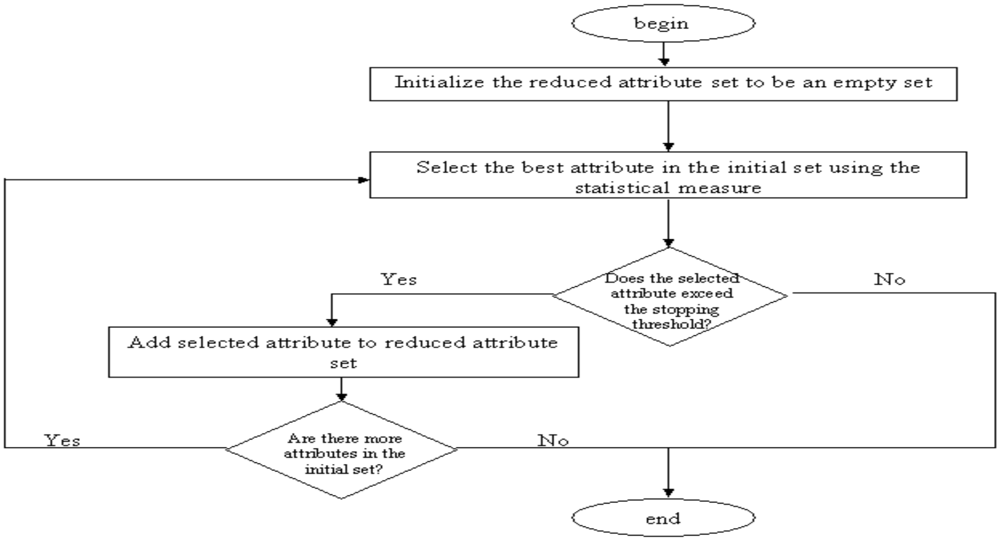
begin
Initialize the reduced attribute set to be an empty set
Select the best attribute in the initial set using the statistical measure
Does the selected attribute exceed the stopping threshold?
Yes
No
Add selected attribute to reduced attribute set
Are there more attributes in the initial set?
Yes
No
end

Figure 2.4: Stepwise forward selection.

(c) clustering

Answer:

(a) equal-frequency (equidepth) partitioning

bin 1	5,10,11,13
bin 2	15,35,50,55
bin 3	72,92,204,215

(b) equal-width partitioning
The width of each interval will be $(215 - 5)/3 = 70$.

bin 1	5,10,11,13,15,35,50,55,72
bin 2	92
bin 3	204,215

(c) clustering
We will use a simple clustering technique, divide the data along the 2 biggest gaps in the data.

bin 1	5,10,11,13,15
bin 2	35,50,55,72,92
bin 3	204,215

■

15. Using the data for *age* given in Exercise 2.4,

(a) Plot an equal-width histogram of width 10.

(b) Sketch examples of each of the following sampling techniques: SRSWOR, SRSWR, cluster sampling, stratified sampling. Use samples of size 5 and the strata “youth”, “middle-aged”, and “senior”.

Answer:

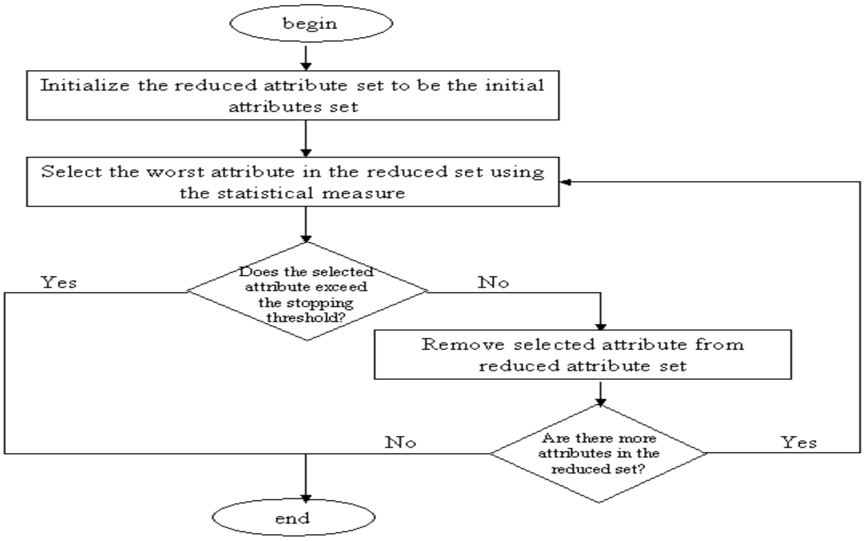
begin
Initialize the reduced attribute set to be the initial attributes set
Select the worst attribute in the reduced set using the statistical measure
Does the selected attribute exceed the stopping threshold?
Yes
No
Remove selected attribute from reduced attribute set
Are there more attributes in the reduced set?
No
Yes
end

Figure 2.5: Stepwise backward elimination.

(a) Plot an equiwidth histogram of width 10.

See Figure 2.7.

(b) Sketch examples of each of the following sampling techniques: SRSWOR, SRSWR, cluster sampling, stratified sampling. Use samples of size 5 and the strata *"young"*, *"middle-aged"*, and *"senior"*.

See Figure 2.8.

■

16. [Contributed by Chen Chen] The *median* is one of the most important holistic measures in data analysis. Propose several methods for median approximation. Analyze their respective complexity under different parameter settings and decide to what extent the real value can be approximated. Moreover, suggest a heuristic strategy to balance between accuracy and complexity and then apply it to all methods you have given.

Answer:

This question can be dealt with either theoretically or empirically, but doing some experiments to get the result is maybe more interesting.

We can give students some data sets sampled from different distributions, e.g., uniform, Gaussian (both two are symmetric) and exponential, gamma (both two are skewed). For example, if we use Equation (2.4) to do approximation as proposed in the chapter, the most straightforward way is to divide all data into k equal length intervals.

$$median = L_1 + (\frac{N/2 - (\sum freq)_l}{freq_{median}})width, \tag{2.4}$$

where L_1 is the lower boundary of the median interval, N is the number of values in the entire data set, $(\sum freq)_l$ is the sum of the frequencies of all of the intervals that are lower than the median interval, $freq_{median}$ is the frequency of the median interval, and $width$ is the width of the median interval.

Obviously, the error incurred will be decreased as k becomes larger and larger; however, the time used in the whole procedure will also increase. What I want is: analyzing this kind of relationship more formally. It seems the product of error made and time used is a good optimality measure. From this point, we can do

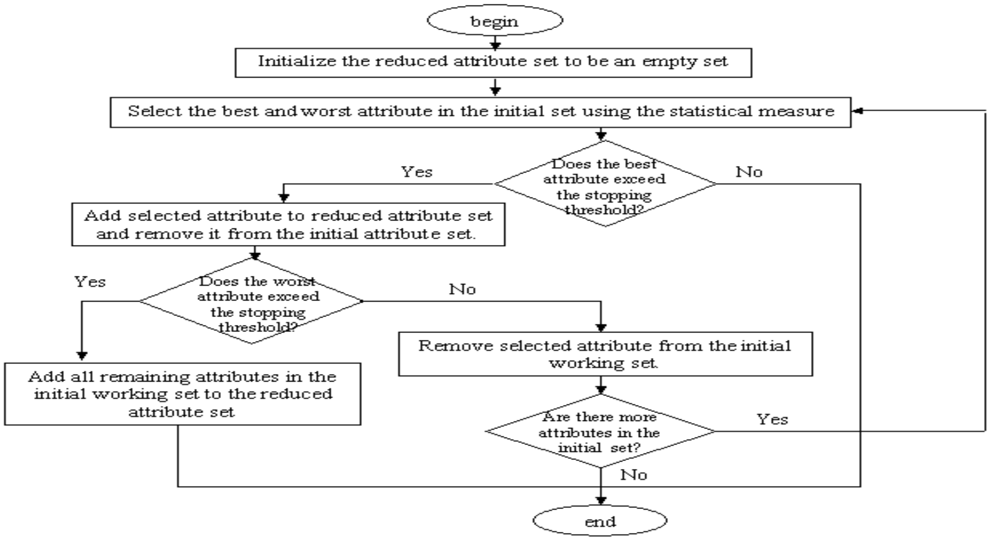
begin
Initialize the reduced attribute set to be an empty set
Select the best and worst attribute in the initial set using the statistical measure
Does the best attribute exceed the stopping threshold?
Yes
No
Add selected attribute to reduced attribute set and remove it from the initial attribute set.
Does the worst attribute exceed the stopping threshold?
Yes
No
Remove selected attribute from the initial working set.
Add all remaining attributes in the initial working set to the reduced attribute set
Are there more attributes in the initial set?
Yes
No
end

Figure 2.6: A combination of forward selection and backward elimination.

many tests for each type of distributions (so that the result won't be dominated by randomness) and find the k giving the best trade-off. In practice, this parameter value can be chosen to improve system performance. There are also some other approaches to approximate the median, students can propose them, analyze the best trade-off point, and compare the results among different approaches. A possible way is as following: Hierarchically divide the whole data set into intervals: at first, divide it into k regions, find the region in which the median resides; then secondly, divide this particular region into k sub-regions, find the sub-region in which the median resides; .... We will iteratively doing this, until the width of the sub-region reaches a predefined threshold, and then the median approximation formula as above stated is applied. By doing this, we can confine the median to a smaller area without globally partitioning all data into shorter intervals, which is expensive (the cost is proportional to the number of intervals).

■

17. [Contributed by Deng Cai] It is important to define or select similarity measures in data analysis. However, there is no commonly-accepted subjective similarity measure. Using different similarity measures may deduce different results. Nonetheless, some apparently different similarity measures may be equivalent after some transformation.

Suppose we have the following two-dimensional data set:

	A_1	A_2
$\boldsymbol{x_1}$	1.5	1.7
$\boldsymbol{x_2}$	2	1.9
$\boldsymbol{x_3}$	1.6	1.8
$\boldsymbol{x_4}$	1.2	1.5
$\boldsymbol{x_5}$	1.5	1.0

(a) Consider the data as two-dimensional data points. Given a new data point, $\boldsymbol{x} = (1.4, 1.6)$ as a query, rank the database points based on similarity with the query using (1) Euclidean distance, and (2) cosine similarity.

(b) Normalize the data set to make the norm of each data point equal to 1. Use Euclidean distance on the transformed data to rank the data points.

Answer:

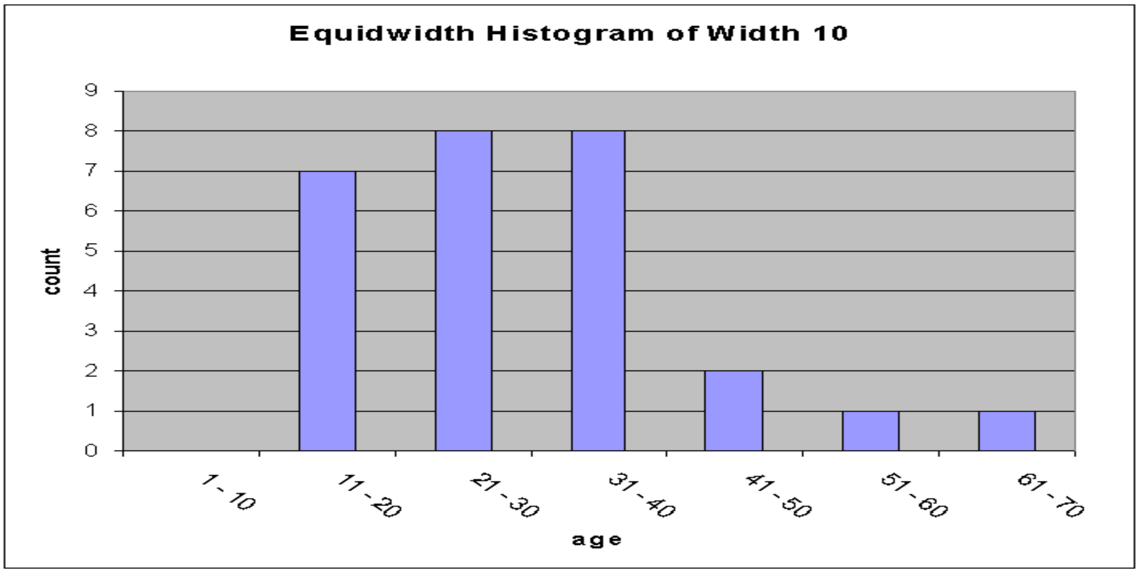

Figure 2.7: An equiwidth histogram of width 10 for *age*.

(a) Consider the data as two-dimensional data points. Given a new data point, $\boldsymbol{x} = (1.4, 1.6)$ as a query, rank the database points based on similarity with the query using (1) Euclidean distance, and (2) cosine similarity.

The Euclidean distance of two vectors is defined as: $\sqrt[2]{sum_i(x_i - y_i)^2}$. The cosine similarity of two vectors is defined as: $\frac{x^t y}{|x||y|}$. Using these definitions we get the distance of each point to the query point.

	x_1	x_2	x_3	x_4	x_5
Euclidean distance	0.14	0.67	0.28	0.22	0.61
Cosine similarity	0.9999	0.9957	0.9999	0.9990	0.9653

Based on the Euclidean distance the order is x_1, x_4, x_3, x_5, x_2, based on the cosine similarity the order is x_1, x_3, x_4, x_2, x_5.

(b) Normalize the data set to make the norm of each data point equal to 1. Use Euclidean distance on the transformed data to rank the data points.

After normalizing the data we get:

x	x_1	x_2	x_3	x_4	x_5
0.6585	0.6616	0.7250	0.6644	0.6247	0.8321
0.7526	0.7498	0.6887	0.7474	0.7809	0.5547

The new Euclidean distance is:

	x_1	x_2	x_3	x_4	x_5
Euclidean distance	0.0041	0.0922	0.0078	0.0441	0.2632

Based on the Euclidean distance of the normalized points, the order is x_1, x_3, x_4, x_2, x_5, which is the same as the cosine similarity order.

■

18. ChiMerge [Ker92] is a supervised, bottom-up (i.e., merge-based) *data discretization* method. It relies on χ^2 analysis: adjacent intervals with the least χ^2 values are merged together till the chosen stopping criterion satisfies.

Tuples

T1	13
T2	15
T3	16
T4	16
T5	19
T6	20
T7	20
T8	21
T9	22

T10	22
T11	25
T12	25
T13	25
T14	25
T15	30
T16	33
T17	33
T18	33

T19	33
T20	35
T21	35
T22	36
T23	40
T24	45
T25	46
T26	52
T27	70

SRSWOR vs. SRSWR

SRSWOR	(n = 5)
T4	16
T6	20
T10	22
T11	25
T26	32

SRSWR	(n = 5)
T7	20
T7	20
T20	35
T21	35
T25	46

Clustering sampling: Initial clusters

T1	13
T2	15
T3	16
T4	16
T5	19

T6	20
T7	20
T8	21
T9	22
T10	22

T11	25
T12	25
T13	25
T14	25
T15	30

T16	33
T17	33
T18	33
T19	33
T20	35

T21	35
T22	36
T23	40
T24	45
T25	46

T26	52
T27	70

Cluster sampling (m = 2)

T6	20
T7	20
T8	21
T9	22
T10	22

T21	35
T22	36
T23	40
T24	45
T25	46

Stratified Sampling

T1	13	young
T2	15	young
T3	16	young
T4	16	young
T5	19	young
T6	20	young
T7	20	young
T8	21	young
T9	22	young

T10	22	young
T11	25	young
T12	25	young
T13	25	young
T14	25	young
T15	30	middle age
T16	33	middle age
T17	33	middle age
T18	33	middle age

T19	33	middle age
T20	35	middle age
T21	35	middle age
T22	36	middle age
T23	40	middle age
T24	45	middle age
T25	46	middle age
T26	52	middle age
T27	70	senior

Stratified Sampling (according to age)

T4	16	young
T12	25	young
T17	33	middle age
T25	46	middle age
T27	70	senior

Figure 2.8: Examples of sampling: SRSWOR, SRSWR, cluster sampling, stratified sampling.

(a) Briefly describe how ChiMerge works.

(b) Take the IRIS data set, obtained from *http://www.ics.uci.edu/∼mlearn/MLRepository.html* (UC-Irvine Machine Learning Data Repository), as a data set to be discretized. Perform data discretization for each of the four numerical attributes using the ChiMerge method. (Let the stopping criteria be: *max-interval* = 6). You need to write a small program to do this to avoid clumsy numerical computation. Submit your simple analysis and your test results: split points, final intervals, and your documented source program.

Answer:

(a) Briefly describe how ChiMerge works.

The basic algorithm of chiMerge is:

```
begin
    sort values in ascending order
    assign a separate interval to each distinct value
    while stopping criteria not met
    begin
        compute $\chi^2$ of every pair of adjacent intervals
        merge the two intervals with smallest $\chi^2$ value
    end
end
```

(b) Take the IRIS data set, obtained from *http://www.ics.uci.edu/∼mlearn/MLRepository.html* (UC-Irvine Machine Learning Data Repository), as a data set to be discretized. Perform data discretization for each of the four numerical attributes using the ChiMerge method. (Let the stopping criteria be: *max-interval* = 6). You need to write a small program to do this to avoid clumsy numerical computation. Submit your simple analysis and your test results: split points, final intervals, and your documented source program.

The final intervals are:

Sepal length: [4.3 - 4.8],[4.9 - 4.9], [5.0 - 5.4], [5.5 - 5.7], [5.8 - 7.0], [7.1 - 7.9].
Sepal width: [2.0 - 2.2], [2.3 - 2.4], [2.5 - 2.8], [2.9 - 2.9], [3.0 - 3.3], [3.4 - 4.4].
Petal length: [1.0 - 1.9], [3.0 - 4.4], [4.5 - 4.7], [4.8 - 4.9], [5.0 - 5.1], [5.2 - 6.9].
Petal width: [0.1 - 0.6], [1.0 - 1.3], [1.4 - 1.6], [1.7 - 1.7], [1.8 - 1.8], [1.9 - 2.5].

The split points are:

Sepal length: 4.3, 4.9, 5.0, 5.5, 5.8, 7.1
Sepal width: 2.0, 2.3, 2.5, 2.9, 3.0, 3.4
Petal length: 1.0, 3.0, 4.5, 4.8, 5.0, 5.2
Petal width: 0.1, 1.0, 1.4, 1.7, 1.8, 1.9

■

19. Propose an algorithm, in pseudocode or in your favorite programming language, for the following:

(a) The automatic generation of a concept hierarchy for categorical data based on the number of distinct values of attributes in the given schema

(b) The automatic generation of a concept hierarchy for numerical data based on the *equal-width* partitioning rule

(c) The automatic generation of a concept hierarchy for numerical data based on the *equal-frequency* partitioning rule

Answer:

(a) The automatic generation of a concept hierarchy for categorical data based on the number of distinct values of attributes in the given schema

Pseudocode for the automatic generation of a concept hierarchy for categorical data based on the number of distinct values of attributes in the given schema:

```
begin
// array to hold name and distinct value count of attributes
// used to generate concept hierarchy
array count_ary[]; string count_ary[].name; // attribute name
int count_ary[].count; // distinct value count

// array to represent concept hierarchy (as an ordered list of values)
array concept_hierarchy[];

for each attribute 'A' in schema {
    distinct_count = count distinct 'A';
    insert ('A', 'distinct _count') into count_ary[];
}

sort count_ary[] ascending by count;

for (i = 0; i < count_ary[].length; i++) {
// generate concept hierarchy nodes
```

```
        concept_hierarchy[i] = count_ary[i].name;
} end
```

To indicate a minimal count threshold necessary for generating another level in the concept hierarchy, the user could specify an additional parameter.

(b) The automatic generation of a concept hierarchy for numeric data based on the *equiwidth* partitioning rule

```
begin
// numerical attribute to be used to generate concept hierarchy
string concept_attb;

// array to represent concept hierarchy (as an ordered list of values)
array concept_hierarchy[];

string concept_hierarchy[].name; // attribute name
int concept_hierarchy[].max; // max value of bin
int concept_hierarchy[].min; // min value of bin
int concept_hierarchy[].mean; // mean value of bin
int concept_hierarchy[].sum; // sum of bin
int concept_hierarchy[].count; // tuple count of bin

int range_min; // min data value − user specified
int range_max; // max data value − user specified
int step; // width of bins − user specified
int j=0;

// initialize concept hierarchy array
for (i=0; i < range_max; i+=step) {
        concept_hierarchy[j].name = ’level_’ + j;
        concept_hierarchy[j].min = i;
        concept_hierarchy[j].max = i + step − 1;
        j++;
}

// initialize final max value if necessary
if (i ¿=range_max) {
        concept_hierarchy[j].max = i + step − 1;
}

// assign each value to a bin by incrementing the appropriate sum and count values
for each tuple T in task relevant data set {
        int k=0;
        while (T.concept_attb > concept_hierarchy[k].max) { k++; }
        concept_hierarchy[k].sum += T.concept_attb;
        concept_hierarchy[k].count++;
}

// calculate the bin metric used to represent the value of each level
// in the concept hierarchy
for i=0; i < concept_hierarchy[].length; i++) {
        concept_hierarchy[i].mean = concept_hierarchy[i].sum / concept_hierarchy[i].count;
} end
```

The user can specify more meaningful names for the concept hierarchy levels generated by reviewing the maximum and minimum values of the bins, with respect to background knowledge about the data (i.e., assigning the labels *young*, *middle-aged* and *old* to a three level hierarchy generated for *age*.) Also, an alternative binning method could be implemented, such as smoothing by bin modes.

(c) The automatic generation of a concept hierarchy for numeric data based on the *equidepth* partitioning rule

Pseudocode for the automatic generation of a concept hierarchy for numeric data based on the equidepth partitioning rule:

```
begin
// numerical attribute to be used to generate concept hierarchy
string concept_attb;

// array to represent concept hierarchy (as an ordered list of values)
array concept_hierarchy[];
string concept_hierarchy[].name; // attribute name
int concept_hierarchy[].max; // max value of bin
int concept_hierarchy[].min; // min value of bin
int concept_hierarchy[].mean; // mean value of bin
int concept_hierarchy[].sum; // sum of bin
int concept_hierarchy[].count; // tuple count of bin

int bin_depth; // depth of bins to be used − user specified
int range_min; // min data value − user specified
int range_max; // max data value − user specified

// initialize concept hierarchy array
for (i=0; i < (range_max/bin_depth(; i++) {
    concept_hierarchy[i].name = 'level_' + i;
    concept_hierarchy[i].min = 0;
    concept_hierarchy[i].max = 0;
}

// sort the task-relevant data set sort data_set ascending by concept_attb;

int j=1; int k=0;

// assign each value to a bin by incrementing the appropriate sum,
// min and max values as necessary
for each tuple T in task relevant data set {
    concept_hierarchy[k].sum += T.concept_attb;
    concept_hierarchy[k].count++;
    if (T.concept_attb <= concept_hierarchy[k].min) {
        concept_hierarchy[k].min = T.concept_attb;
    }
    if (T.concept_attb >= concept_hierarchy[k].max) {
        concept_hierarchy[k].max = T.concept_attb;
    };
    j++;
    if (j > bin_depth) {
        k++; j=1;
    }
}

// calculate the bin metric used to represent the value of each level
// in the concept hierarchy
for i=0; i < concept_hierarchy[].length; i++) {
    concept_hierarchy[i].mean = concept_hierarchy[i].sum / concept_hierarchy[i].count;
}
end
```

This algorithm does not attempt to distribute data values across multiple bins in order to smooth out any difference between the actual depth of the final bin and the desired depth to be implemented. Also, the user can again specify more meaningful names for the concept hierarchy levels generated by reviewing the maximum and minimum values of the bins, with respect to background knowledge about the data.

■

20. Robust data loading poses a challenge in database systems because the input data are often dirty. In many cases, an input record may miss multiple values, some records could be *contaminated*, with some data values out of range or of a different data type than expected. Work out an automated *data cleaning and loading* algorithm so that the erroneous data will be marked, and contaminated data will not be mistakenly inserted into the database during data loading.

Answer:

```
begin
    for each record r
    begin
        check r for missing values
            If possible, fill in missing values according to domain knowledge
            (e.g. mean, mode, most likely value, etc).
        check r for out of range values
            If possible, correct out of range values according to domain knowledge
            (e.g. min or max value for the attribute).
        check r for erroneous data types
            If possible, correct data type using domain knowledge
        If r could not be corrected mark it as bad and output it to a log,
        otherwise load r into the database.
    end
end
```

The domain knowledge can be a combination of manual and automatic work. We can for example, use the data in the database to construct a decision tree to induce missing values for a given attribute, and at the same time have human entered rules on how to correct wrong data types.

■

Bibliography

[BR99] K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pages 359–370, Philadelphia, PA, June 1999.

[HPDW01] J. Han, J. Pei, G. Dong, and K. Wang. Efficient computation of iceberg cubes with complex measures. In *Proc. 2001 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'01)*, pages 1–12, Santa Barbara, CA, May 2001.

[Ker92] R. Kerber. Discretization of numeric attributes. In *Proc. 1992 Nat. Conf. Artificial Intelligence (AAAI'92)*, pages 123–128, AAAI/MIT Press, 1992.

[XHLW03] D. Xin, J. Han, X. Li, and B. W. Wah. Star-cubing: Computing iceberg cubes by top-down and bottom-up integration. In *Proc. 2003 Int. Conf. Very Large Data Bases (VLDB'03)*, Berlin, Germany, Sept. 2003.

[ZDN97] Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97)*, pages 159–170, Tucson, Arizona, May 1997.