Data Mining: Concepts and Techniques

3rd Edition

Solution Manual

Jiawei Han, Micheline Kamber, Jian Pei

The University of Illinois at Urbana-Champaign

Simon Fraser University

Version January 2, 2012

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 is intended to be used as a guide in answering the exercises of the textbook. You are welcome to enrich this manual by suggesting additional interesting exercises and/or providing more thorough, or better alternative solutions.

While we have done our best to ensure the correctness of the solutions, it is possible that some typos or errors may exist. 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.

To assist the teachers of this book to work out additional homework or exam questions, we have added one additional section "**Supplementary Exercises**" to each chapter of this manual. This section includes additional exercise questions and their suggested answers and thus may substantially enrich the value of this solution manual. Additional questions and answers will be incrementally added to this section, extracted from the assignments and exam questions of our own teaching. To this extent, our solution manual will be incrementally enriched and subsequently released in the future months and years.

Notes to the current release of the solution manual.

Due to the limited time, this release of the solution manual is a preliminary version. Many of the newly added exercises in the third edition have not provided the solutions yet. We apologize for the inconvenience. We will incrementally add answers to those questions in the next several months and release the new versions of updated solution manual in the subsequent months.

Acknowledgements

For each edition of this book, the solutions to the exercises were worked out by different groups of teach assistants and students. We sincerely express our thanks to all the teaching assistants and participating students who have worked with us to make and improve the solutions to the questions. In particular, for the first edition of the book, we would like to thanks Denis M. C. Chai, Meloney H.-Y. Chang, James W. Herdy, Jason W. Ma, Jiuhong Xu, Chunyan Yu, and Ying Zhou who took the class of *CMPT-459: Data Mining and Data Warehousing* at Simon Fraser University in the Fall semester of 2000 and contributed substantially to the solution manual of the 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.

For the solution manual of the second edition of the book, we would like to thank Ph.D. students and teaching assistants, Deng Cai and Hector Gonzalez, for the course *CS412: Introduction to Data Mining and Data Warehousing*, offered in the Fall semester of 2005 in the Department of Computer Science at the University of Illinois at Urbana-Champaign. They have helped prepare and compile the answers for the new exercises of the first seven chapters in our second edition. Moreover, our thanks go to several students from the *CS412* class in the Fall semester of 2005 and the *CS512: Data Mining: Principles and Algorithms* classes

in the Spring semester of 2006. Their answers to the class assignments have contributed to the advancement of this solution manual.

For the solution manual of the third edition of the book, we would like to thank Ph.D. students, Jialu Liu, Brandon Norick and Jingjing Wang, in the course *CS412: Introduction to Data Mining and Data Warehousing*, offered in the Fall semester of 2011 in the Department of Computer Science at the University of Illinois at Urbana-Champaign. They have helped checked the answers of the previous editions and did many modifications, and also prepared and compiled the answers for the new exercises in this edition. Moreover, our thanks go to teaching assistants, Xiao Yu, Lu An Tang, Xin Jin and Peixiang Zhao, from the *CS412* class and the *CS512: Data Mining: Principles and Algorithms* classes in the years of 2008–2011. Their answers to the class assignments have contributed to the advancement of this solution manual.

Contents

Chapter 1

Introduction

1.1 Exercises

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? Do 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.

   Answer:

   Data mining refers to 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.

iv. Incremental updating

Which implementation techniques do you prefer, and why?

Answer:

(a) Briefly describe each implementation technique.

A **ROLAP** technique for implementing a multiple dimensional view consists of intermediate servers that stand in between a relational back-end server and client front-end tools, thereby using a relational or extended-relational DBMS to store and manage warehouse data, and OLAP middleware to support missing pieces. A **MOLAP** implementation technique consists of servers, which support multidimensional views of data through array-based multidimensional storage engines that map multidimensional views directly to data cube array structures. A **HOLAP** implementation approach combines ROLAP and MOLAP technology, which means that large volumes of detailed data and some very low level aggregations can be stored in a relational database, while some high level aggregations are kept in a separate MOLAP store.

(b) For each technique, explain how each of the following functions may be implemented:

i. The generation of a data warehouse (including aggregation)
ROLAP: Using a ROLAP server, the generation of a data warehouse can be implemented by a relational or extended-relational DBMS using summary fact tables. The fact tables can store aggregated data and the data at the abstraction levels indicated by the join keys in the schema for the given data cube.
MOLAP: In generating a data warehouse, the MOLAP technique uses multidimensional array structures to store data and multiway array aggregation to compute the data cubes.
HOLAP: The HOLAP technique typically uses a relational database to store the data and some low level aggregations, and then uses a MOLAP to store higher-level aggregations.

ii. Roll-up
ROLAP: To roll-up on a dimension using the summary fact table, we look for the record in the table that contains a generalization on the desired dimension. For example, to roll-up the *date* dimension from *day* to *month*, select the record for which the *day* field contains the special value all. The value of the measure field, *dollars_sold*, for example, given in this record will contain the subtotal for the desired roll-up.
MOLAP: To perform a roll-up in a data cube, simply climb up the concept hierarchy for the desired dimension. For example, one could roll-up on the *location* dimension from *city* to *country*, which is more general.
HOLAP: The roll-up using the HOLAP technique will be similar to either ROLAP or MOLAP, depending on the techniques used in the implementation of the corresponding dimensions.

iii. Drill-down
ROLAP: To drill-down on a dimension using the summary fact table, we look for the record in the table that contains a generalization on the desired dimension. For example, to drill-down on the *location* dimension from *country* to *province_or_state*, select the record for which only the next lowest field in the concept hierarchy for *location* contains the special value all. In this case, the *city* field should contain the value all. The value of the measure field, *dollars_sold*, for example, given in this record will contain the subtotal for the desired drill-down.
MOLAP: To perform a drill-down in a data cube, simply step down the concept hierarchy for the desired dimension. For example, one could drill-down on the *date* dimension from *month* to *day* in order to group the data by *day* rather than by *month*.
HOLAP: The drill-down using the HOLAP technique is similar either to ROLAP or MOLAP depending on the techniques used in the implementation of the corresponding dimensions.

iv. Incremental updating

(a) Containing at least one Blueray DVD movie
The constraint is succinct and monotonic. This constraint can be mined efficiently using FP-growth as follows.
- All frequent Nintendo games are listed at the end of the list of frequent items L.
- Only those conditional pattern bases and FP-trees for frequent Blueray DVD movie need to be derived from the global FP-tree and mined recursively.

(b) Containing items whose sum of the prices is less than \$150
The constraint is antimonotonic. This constraint can be mined efficiently using Apriori as follows. Only candidates with sum of prices less than \$150 need to be checked.

(c) Containing one free item and other items whose sum of the prices is at least \$200 in total.
The constraint "*containing one free item*" is succinct, whereas the constraint "*sum of the prices is at least \$200 in total*" is monotonic and data antimonotonic. It can be mined efficiently using FP-growth as follows.
- Put all frequent free items at the end of the list of frequent items L (i.e., they will be pushed in first in mining)
- Only conditional pattern bases and FP-trees for frequent free items need to be derived from the global FP-tree and mined recursively. Other free items should be excluded from these conditional pattern bases and FP-trees.
- Once a pattern with sum of the price is at least \$200, no further constraint checking for total price is needed in recursive mining.
- A pattern as well as its conditional pattern base can be pruned, if the sum of the price of items in the pattern and the frequent ones in the pattern base is less than \$200 (based on the property of data anti-monotonicity).

(d) Where the average price of all the items is between \$100 and \$150.
The sub-constraints "*the average price is at least \$100*" and "*the average price is at most \$500*" are convertible. It can be mined efficiently using FP-growth as follows.
- All the frequent items are listed in price descending order; (if you use ascending order, you must rewrite the following two steps.)
- A pattern as well as its conditional pattern base can be pruned, if the average price of items in the pattern and those frequent ones in pattern base with prices greater than \$100 is less than \$100.
- A pattern as well as its conditional pattern base can also be pruned, if the average price of items in the pattern is greater than \$500.

■

8. Section 7.4.1 introduced a core-pattern-fusion method for **mining high-dimensional data**. Explain why a long pattern, if exists in the data set, is likely to be discovered by this method.

Answer:

The core Pattern-Fusion method utilizes the concept of core patterns to effectively discover colossal patterns. Not only do colossal patterns have far more core descendants than small patterns, but they have far more core patterns that are close to one another than small patterns. Therefore, the Pattern-Fusion method of discovering close core patterns and fusing them together has a high probability to detect any colossal pattern which exists. ■

9. Section 7.5.1 defined a **pattern distance measure** between closed patterns P_1 and P_2 as

$$Pat\_Dist(P_1, P_2) = 1 - \frac{|T(P_1) \cap T(P_2)|}{|T(P_1) \cup T(P_2)|},$$

Answer:

On a single scan of the database, for each attribute value we collect the following table of counts:

attribute A	$class_1$	$class_2$	...	$class_c$
value 1	$count_{1,1}$	$count_{1,2}$	...	$count_{1,c}$
...				
value k	$count_{k,1}$	$count_{k,2}$	...	$count_{k,c}$

Note that $count_{i,j}$ is the number of times that a tuple has value i of attribute A and belongs to $class_j$.

With these tables we can compute any probability of the form $P(class_i|tuple_j)$.

The size of these tables is, in general, much smaller than the database size because it only depends on the number of attributes, their distinct values, and the number of classes.

If we want to incorporate boosting we can train a few naïve Bayesian classifiers using a sample of the training data. For each new classifier we use the tuples we misclassified with increased weight; at test time we will collect the decisions of each classifier and weight them by the classifier's accuracy. In this case, we maintain separate count tables for each classifier.

■

9. Design an efficient method that performs effective naïve Bayesian classification over an *infinite* data stream (i.e., you can scan the data stream only once). If we wanted to discover the *evolution* of such classification schemes (e.g., comparing the classification scheme at this moment with earlier schemes, such as one from a week ago), what modified design would you suggest?

   Answer:

   The design is very similar to that presented in Exercise 8.6. We collect a set of attribute-value count tables, and update the counts as each new example streams in.

   To discover the evolution of the classification scheme, we can maintain counts for a few classifiers in parallel. For instance, we can keep one classifier based on the entire history of data, another based on the previous week of data, and another based on only the previous day of data. For the weekly classifier, we maintain separate counts for the previous seven days. At the end of each day, we discard the oldest day's counts and replace them with the counts of the previous day. For the daily classifier, we maintain separate counts for each hour, and similarly, each hour replace the oldest counts with the ones for the previous hour.

   ■

10. Show that accuracy is a function of *sensitivity* and*specificity*, that is, prove Equation 8.25.

    Answer:

$$\begin{aligned}\text{accuracy} &= \frac{TP+TN}{(P+N)} \\ &= \frac{TP}{(P+N)} + \frac{TN}{(P+N)} \\ &= \frac{TP}{(P+N)} \times \frac{P}{P} + \frac{TN}{(P+N)} \times \frac{N}{N} \\ &= sensitivity\frac{P}{(TP+TN)} + specificity\frac{N}{(TP+TN)}.\end{aligned}$$

    ■

11. The harmonic mean is one of several kinds of averages. Chapter 2 discussed how to compute the *arithmetic mean*, which is what most people typically think of when they compute an average. The