TURING 图灵计算机科学丛书

PEARSON

Introduction to Data Mining

数据挖掘导论（完整版）

Pang-Ning Tan
[美] Michael Steinbach 著
Vipin Kumar
范明 范宏建 等译

人民邮电出版社
POSTS & TELECOM PRESS

“这是一本全新的数据挖掘教材，值得大力推荐。” ——Jiawei Han，伊利诺伊大学教授

Introduction to Data Mining

数据挖掘导论（完整版）

本书全面介绍了数据挖掘，涵盖了五个主题：数据、分类、关联分析、聚类和异常检测。除异常检测外，每个主题都有两章。前一章涵盖基本概念、代表性算法和评估技术，而后一章讨论高级概念和算法。这样读者在透彻地理解数据挖掘的基础的同时，还能够了解更多重要的高级主题。

本书是明尼苏达大学和密歇根州立大学数据挖掘课程的教材，由于独具特色，正式出版之前就已经被斯坦福大学、得克萨斯大学奥斯汀分校等众多名校采用。

本书特色

- ◆与许多其他同类图书不同，本书将重点放在如何用数据挖掘知识解决各种实际问题。
- ◆只要求具备很少的预备知识——不需要数据库背景，只需要很少的统计学或数学背景知识。
- ◆书中包含大量的图表、综合示例和丰富的习题，并且使用示例、关键算法的简洁描述和习题，尽可能直接聚焦于数据挖掘的主要概念。
- ◆教辅内容极为丰富，包括课程幻灯片、学生课题建议、数据挖掘资源（如数据挖掘算法和数据集）、联机指南（使用实际的数据集和数据分析软件，为本书介绍的部分数据挖掘技术提供例子讲解）。
- ◆向采用本书作为教材的教师提供习题解答。

PEARSON

www.pearsonhighered.com

图灵网站：www.turingbook.com 热线：(010)51095186
反馈/投稿/推荐信箱：contact@turingbook.com
有奖勘误：debug@turingbook.com

分类建议 计算机/数据库

人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-115-24100-9

ISBN 978-7-115-24100-9

定价：69.00元

TURING 图灵计算机科学丛书

Introduction to Data Mining

数据挖掘导论

（完整版）

人民邮电出版社
北京

图书在版编目（CIP）数据

数据挖掘导论 : 完整版 / (美) 陈封能, (美) 斯坦巴赫 (Steinbach, M.), (美) 库玛尔 (Kumar, V.) 著 ; 范明等译. -- 2版. -- 北京 : 人民邮电出版社, 2011.1
(图灵计算机科学丛书)
ISBN 978-7-115-24100-9

Ⅰ. ①数… Ⅱ. ①陈… ②斯… ③库… ④范… Ⅲ. ①数据采集 Ⅳ. ①TP274

中国版本图书馆CIP数据核字(2010)第209213号

内 容 提 要

本书全面介绍了数据挖掘的理论和方法，旨在为读者提供将数据挖掘应用于实际问题所必需的知识。本书涵盖五个主题：数据、分类、关联分析、聚类和异常检测。除异常检测外，每个主题都包含两章：前面一章讲述基本概念、代表性算法和评估技术，后面一章较深入地讨论高级概念和算法。目的是使读者在透彻地理解数据挖掘基础的同时，还能了解更多重要的高级主题。此外，书中还提供了大量示例、图表和习题。

本书适合作为相关专业高年级本科生和研究生数据挖掘课程的教材，同时也可作为数据挖掘研究和应用开发人员的参考书。

图灵计算机科学丛书

数据挖掘导论（完整版）

◆ 著 [美] Pang-Ning Tan Michael Steinbach Vipin Kumar
译 范 明 范宏建 等
责任编辑 杨海玲
执行编辑 丁晓昀

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：787×1092 1/16
印张：30
字数：787千字 2011年1月第2版
印数：10 001－13 000册 2011年1月北京第1次印刷
著作权合同登记号 图字：01-2005-5236号
ISBN 978-7-115-24100-9

定价：69.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版 权 声 明

Preface of the Chinese Edition

It is with great pleasure that we welcome the Chinese translation of our book by Professors Fan, Dr. Fan, *et al.*, who have previously translated several well-known statistics and data mining texts. Data mining is an area in computer science that aims to analyze the rapidly increasing amounts of business, scientific, and engineering data for knowledge and other profitable uses. The field has seen tremendous growth and development, with the great influx of scholars and researchers, not only from the Western countries but also from the Far East. We thank Professors Fan and Dr. Fan for their effort in doing the translation, which allows the book to reach a much broader audience among those students and researchers who are well-versed in the Chinese language. We hope that the readers of our book will find it to be both useful and engaging, and wish them the greatest success.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar
Michigan State University and the University of Minnesota, December 2005

中 文 版 序

我们非常欢迎由范明教授和范宏建博士等人将我们的书翻译成中文，他们在此之前翻译了几本关于统计学和数据挖掘方面的著名教材。数据挖掘是计算机科学的一个领域，其目的是通过分析快速增长的商业、科学和工程数据来获取知识和其他利益。我们已经目睹了这个领域的迅猛增长和发展，学者和研究人员大量涌入其中，他们不仅来自西方国家，而且来自远东地区。我们感谢范明教授和范宏建博士，他们的翻译成果使本书得以传播到更广的读者群，包括那些精通中文的学生和研究人员。我们期望读者会发现这是一部有用的和引人入胜的书籍。祝你们成功!

Pang-Ning Tan
Michael Steinbach
Vipin Kumar
2005 年 12 月于密歇根州立大学和明尼苏达大学

完整版译者序

图灵教育已经走过了 5 年。在图灵公司成立之初，我受其之托，翻译 P. Tan、M. Steinbach 和 V. Kumar 的力作 *Introduction to Data Mining*。2006 年 5 月，该书的中文版《数据挖掘导论》正式与读者见面。

在过去的 4 年多时间里，这本书受到了许多读者的关注，众多高校和研究所把它作为研究生和高年级本科生的数据挖掘相关课程的首选教材和参考书之一。热心的读者对译文提出了许多有益的意见和建议。在此，我们表示衷心感谢！

在迎接图灵公司成立五周年之际，出版社决定对 2006 年的版本进行修订，并补充翻译原著的附录，出版该书的中文完整版。

如一些读者所愿，完整版包含了原著的 5 个附录，涉及线性代数、维度归约、概率统计、回归和优化。这些内容是数据挖掘的数学基础，许多读者都已经从相关的数学专著和教科书中获得了这些知识。原著包含它们是希望这本书是自包含的，使未系统学习过这些数学知识，或虽然学过但有点淡忘的读者在阅读本书时不必四处翻阅数学文献。用作教材时，可以根据学生的情况，选择这些附录作为预备知识提前讲述。完整版包含这些附录也有尊重原著作者的考虑。

完整版对原译文进行了勘误（包括原著作者的勘误），修订了一些翻译生硬的句子，希望能够增强译著的可读性。此外，对于个别术语的译法也做了适当的调整。例如，maximal frequent itemset 改译为“极大频繁项集”，closed frequent itemset 改译为“闭频繁项集”。

修订和附录 A~E 的翻译均由我本人完成。

Introduction to Data Mining 自出版以来受到了广泛欢迎，已经成为数据挖掘领域的经典文献，希望中文版也能受到更多读者的青睐。

范　明

2010 年 10 月于郑州大学

译 者 序

自从我和孟小峰等人翻译 J. Han 和 M. Kamber 的《数据挖掘：概念与技术》以来，我们高兴地看到数据挖掘的研究正在我国蓬勃开展。许多学者和研究人员都对这个新兴的学科领域表现出了极大的兴趣，他们之中不仅有来自数据库领域的专家，而且不乏统计学、人工智能和模式识别、机器学习等领域的研究者。国内的学者和研究者在数据挖掘方面的研究已经取得了一些令人鼓舞的成果，并且正在逐渐与国际学术界同步。

数据挖掘的产生和发展一直是分析和理解数据的实际需求推动的。数据挖掘研究的进展也正是在于一直重视与其他领域研究者的合作。数据挖掘从工业、农业、医疗卫生和商业的需求中获得动力，从统计学、机器学习等领域的长期研究与发展中汲取营养。我们相信，只要有理解数据的需求，就有推动数据挖掘研究与应用发展的动力；只要依靠多学科的团队，就能应对新的数据分析任务带来的挑战。

P. Tan、M. Steinbach 和 V. Kumar 编写的这本《数据挖掘导论》是继《数据挖掘：概念与技术》一书之后的另一本重要的数据挖掘著作。三位作者都从事数据挖掘研究多年，其中 Vipin Kumar 教授是数据挖掘和高性能计算领域的国际知名学者。本书原版在正式出版之前就已经被斯坦福大学、得克萨斯大学奥斯汀分校等众多名校采用。J. Han 教授也高度评价该书："这是一本全新数据挖掘的教材，值得大力推荐。它将成为我们的主要参考书。"

本书不需要读者具备数据库背景，只需要少量统计学或数学背景知识，而且取材涉及的学科和应用领域较多，实用性强，因此适合的读者面较广。本书强调如何用数据挖掘知识解决各种实际问题，强调所挖掘的知识模式的评估。例如，就像我们能够从天空中的白云想象出各种动物和物体一样，每个聚类算法能够从几乎所有的数据集中发现聚类。如果数据集合中根本不存在自然的簇，所产生的聚类很难说具有实际意义。

全书共分 10 章。范明负责第 1~8 章的翻译，范宏建负责第 9 章和第 10 章的翻译。蒋宏杰、贾玉祥、许红涛和温箐笛也参加本书的最初翻译工作。全书的译文由范明负责统一定稿。在翻译的过程中，对发现的错误进行了更正，并得到原书作者的确认。

感谢 P. Tan、M. Steinbach 和 V. Kumar 为中文版撰写序言。感谢人民邮电出版社图灵公司的编辑们，他们在第一时间内引进本书，并组织翻译，使得中文版能够如此之快地与读者见面。

译文中的错误和不当之处，敬请读者朋友指正。意见和建议请发往 mfan@zzu.edu.cn。希望读者喜欢这本译著，希望这本译著有助于推动我国的数据挖掘研究与应用的深入开展。

范 明

2006 年 2 月于郑州大学

译 者 简 介

范明 郑州大学信息工程学院教授，中国计算机学会数据库专业委员会委员、人工智能与模式识别专业委员会委员，长期从事计算机软件与理论教学和研究。主要讲授的课程包括程序设计、计算机操作系统、数据库系统原理、知识库系统原理、数据挖掘与数据仓库等。当前感兴趣的研究方向包括数据挖掘、数据仓库和机器学习。1989～1990年曾访问加拿大 Simon Fraser 大学计算机科学系，从事演绎数据库研究。1999年曾访问美国 Wright State 大学计算机科学与工程系，从事数据挖掘研究。先后发表论文40余篇。除本书外，近年来主持翻译的数据挖掘方面的著作还有 Jiawei Han 和 Micheline Kamber 的《数据挖掘：概念与技术》，Trevor Hastie、Robert Tibshirani 和 Jerome Friedman 的《统计学习基础：数据挖掘、推理与预测》。

范宏建 1999 年毕业于郑州大学计算机科学系，同年进入中国科学院软件研究所攻读硕士学位，次年赴澳大利亚墨尔本大学攻读博士学位，师从澳大利亚科学院院士 Kotagiri Ramamohanarao 教授，2004 年获计算机科学博士学位。先后在 WWW、PAKDD、RSFDGrC、IEEE GrC 和 Australian AI 等国际学术会议和 *IEEE Transactions on Knowledge and Data Engineering* 发表论文 10 余篇。目前是澳大利亚 AUSTRAC 的高级分析师，主要从事利用数据挖掘和机器学习技术进行金融数据分析的工作。

前　言

数据生成和收集技术的进步促使商业和科研领域产生了海量数据集。数据仓库能够存储多种数据，如：企业销售和运作的详细情况，地球轨道卫星发送回地球的高分辨率图像和遥感数据，对越来越多的有机体进行的基因组实验产生的序列、结构和机能数据。收集和存储数据变得轻松简便，已经完全改变了人们对数据分析的态度，人们开始尽可能地收集各个时期和各种来源的数据。人们相信收集的数据肯定会有价值，或者当初收集它就有明确的目的，或者只是先收集起来再说。

传统数据分析技术在应对这些新型数据集提出的挑战时存在种种局限性，而数据挖掘技术突破了这些局限。数据挖掘并不是要取代其他分析领域，而是以它们为基础。尽管数据挖掘的某些主题（如关联分析）是其独有的，但是，还有许多主题（如聚类、分类和异常检测）则建立在其他领域长期工作的基础之上。事实上，数据挖掘研究者们主动利用已有技术对增强和拓展这个领域以及推动它的快速发展起到了促进作用。

该领域一直强调与其他领域的研究者合作，因而充满了活力。要迎接新类型数据分析的挑战，抛开理解数据的人和数据所处的领域而简单地使用数据分析技术是不可行的。通常，能否组建好多学科研究团队，已经成为数据挖掘项目（如创建新的独创性算法）成败的决定因素。正如历史上统计学的许多进展都是由农业、工业、医疗卫生和商业需求推动的一样，今天，数据挖掘的许多进展也正在被这些领域的需求所推动。

自 1998 年春季开始，我们在明尼苏达大学为高年级本科生和研究生开设了数据挖掘课程。为这些课程准备的演示幻灯片和习题随着时间不断积累，成为本书的基础。数据挖掘的聚类技术综述最初是为该领域的某项研究而写的，它也成为本书第 8 章的雏形。随着时间的推移，又增加了关于数据、分类、关联分析和异常检测的几章。本书定稿后已在作者所在的学校（明尼苏达大学和密歇根州立大学）以及其他一些大学作为教材试用。

在此期间，出现了许多数据挖掘方面的书籍，但是都不能完全满足我们学生的需要——他们主要是计算机科学专业的研究生和本科生，也包括来自工科和其他专业的学生。他们的数学和计算机背景差异很大，但是都有一个共同目标：尽可能直接地学习数据挖掘，尽快地将其应用到各自的领域。因此，要求较多数学和统计学预备知识的书对他们中的许多人都没有吸引力，需要坚实的数据库背景的书也有同样的问题。为了满足这些学生需求而逐渐写成的本书，现在的完稿使用了大量例子、习题并用简洁的语言描述了关键算法，尽可能直接把重点放在数据挖掘的主要概念上。

概述

具体而言，本书全面介绍了数据挖掘，方便学生、教师、研究人员和专业人士理解有关概念和技术。本书所涵盖的领域包括数据预处理、可视化、预测建模、关联分析、聚类和异常检测。

目标是讲述每个主题的基本概念和算法，从而为读者提供将数据挖掘应用于实际问题所需的必要背景。此外，本书也为有志于从事数据挖掘和相关领域研究的读者提供一个起点。

本书涵盖五个主题：数据、分类、关联分析、聚类和异常检测。除异常检测外，每个主题都分两章讲述。对于分类、关联分析和聚类，前面一章讲述基本概念、代表性算法和评估技术，后面的一章深入讨论高级概念和算法。这样做的目的是使读者透彻地理解数据挖掘的基础，同时论述更多重要的高级主题。由于这种安排，本书既可用作为教材又可用作参考书。

为了帮助读者理解书中概念，我们提供大量示例、图表和习题。每一章的结尾给出了文献注释，是为那些对更高级的主题、重要的历史文献和当前趋势感兴趣的读者提供的。

致教师

作为一本教材，本书广泛适合于高年级本科生和研究生。由于学习这门课程的学生背景不同，他们可能不具备广博的统计学和数据库知识，因此本书只要求最低限度的预备知识——不需要数据库知识，并假定读者只有一般的统计学或数学背景。本书尽可能自成一体。统计学、线性代数和机器学习的必要基础知识或者已经融入正文，或者包含在附录中。

由于讨论主要数据挖掘主题的各章也是自成一体的，因此主题的讲授次序相当灵活。核心题材在第2、4、6、8和10章介绍。数据导论（第2章）应当最先讨论，基本的分类、关联分析和聚类（分别是第4、6、8章）可以以任意次序讲述。由于异常处理（第10章）与分类（第4章）和聚类（第8章）有一定的关系，这两章应当在第10章之前讲述。还可以根据课程安排和师生的兴趣从高级的分类、关联分析和聚类（分别为第5、7、9章）中选讲一些主题。我们也建议教师用数据挖掘的实际项目和练习强化课程的教学。尽管这样做很耗费时间，但是实践性的作业可以大大提高这门课程的价值。

支持材料

本书的教辅材料可以在 Addison-Wesley 的网站（www.aw-bc.com/cssupport）上找到[①]。提供给所有读者的支持材料如下。

- 课程幻灯片。
- 学生项目建议。
- 数据挖掘资源，如数据挖掘算法和数据集。
- 联机指南，使用实际的数据集和数据分析软件，为本书介绍的部分数据挖掘技术提供例子讲解。

其他支持材料（包括习题答案）只向采纳本书做教材的教师提供。意见和建议以及勘误请通过 dmbook@cs.unm.edu 发给作者。

致谢

许多人都为本书做出了贡献。我们首先向家人表示感谢，这本书是献给他们的。没有他们的耐心和支持，不可能写出本书。

我们要感谢明尼苏达大学和密歇根州立大学数据挖掘小组的学生所做的贡献。Eui-Hong

① 相关材料也可以从图灵网站（www.turingbook.com）本书网页免费注册下载。——编者注

（Sam）Han 和 Mahesh Joshi 帮助我们准备了最初的数据挖掘课程。他们编制的某些习题和演示幻灯片已经收录在本书及其辅助幻灯片中。小组中的其他学生也为本书的初稿提出建议或以各种方式做出贡献，他们是 Shyam Boriah、Haibin Cheng、Varun Chandola、Eric Eilertson、Levent Ertöz、Jing Gao、Rohit Gupta、Sridhar Iyer、Jung-Eun Lee、Benjamin Mayer、Aysel Ozgur、Uygar Oztekin、Gaurav Pandey、Kashif Riaz、Jerry Scripps、Gyorgy Simon、Hui Xiong、Jieping Ye 和 Pusheng Zhang。我们还要感谢明尼苏达大学和密歇根州立大学选修数据挖掘课程的学生，他们使用了本书的初稿，并提供了极富价值的反馈。我们特别感谢 Bernardo Craemer、Arifin Ruslim、Jamshid Vayghan 和 Yu Wei 的有益的建议。

Joydeep Ghosh（得克萨斯大学）和 Sanjay Ranka（佛罗里达大学）试用了本书的初稿。我们也直接从得克萨斯大学下列学生那里获得了许多有用的建议：Pankaj Adhikari、Rajiv Bhatia、Frederic Bosche、Arindam Chakraborty、Meghana Deodhar、Chris Everson、David Gardner、Saad Godil、Todd Hay、Clint Jones、Ajay Joshi、Joonsoo Lee、Yue Luo、Anuj Nanavati、Tyler Olsen、Sunyoung Park、Aashish Phansalkar、Geoff Prewett、Michael Ryoo、Daryl Shannon 和 Mei Yang。

Ronald Kostoff（ONR）阅读了聚类部分的初稿，并提出了许多建议。Musetta Steinbach 发现了图中的一些错误。

我们要感谢明尼苏达大学和密歇根州立大学的同事，他们帮助创建了良好的数据挖掘研究环境。他们是 Dan Boley、Joyce Chai、Anil Jain、Ravi Janardan、Rong Jin、George Karypis、Haesun Park、William F. Punch、Shashi Shekhar 和 Jaideep Srivastava。我们还要向我们的数据挖掘项目的合作者表示谢意，他们是 Ramesh Agrawal、Steve Cannon、Piet C. de Groen、Fran Hill、Yongdae Kim、Steve Klooster、Kerry Long、Nihar Mahapatra、Chris Potter、Jonathan Shapiro、Kevin Silverstein、Nevin Young 和 Zhi-Li Zhang。

明尼苏达大学和密歇根州立大学的计算机科学与工程系为本书写作及研究提供了计算资源和支持环境。ARDA、ARL、ARO、DOE、NASA 和 NSF 等机构为本书作者提供了研究资助。特别应该提到的是，Kamal Abdali、Dick Brackney、Jagdish Chandra、Joe Coughlan、Michael Coyle、Stephen Davis、Frederica Darema、Richard Hirsch、Chandrika Kamath、Raju Namburu、N. Radhakrishnan、James Sidoran、Bhavani Thuraisingham、Walt Tiernin、Maria Zemankova 和 Xiaodong Zhang 有力地支持了我们的数据挖掘和高性能计算研究。

与培生出版集团的工作人员的合作令人愉快。具体地，我们要感谢 Michelle Brown、Matt Goldstein、Katherine Harutunian、Marilyn Lloyd、Kathy Smith 和 Joyce Wells。我们还要感谢 George Nichols 帮助绘图，Paul Anagnostopoulos 提供 LaTeX 支持。我们感谢出版社邀请的审稿人：Chien-Chung Chan（阿克伦大学）、Zhengxin Chen（内布拉斯加大学奥马哈分校）、Chris Clifton（普度大学）、Joydeep Ghosh（得克萨斯大学奥斯汀分校）、Nazli Goharian（伊利诺伊理工学院）、J. Michael Hardin（阿拉巴马大学）、James Hearne（西华盛顿大学），Hillol Kargupta（马里兰大学巴尔的摩县分校和 Agnik 公司）、Eamonn Keogh（加利福尼亚大学里弗赛德分校）、Bing Liu（伊利诺伊大学芝加哥分校）、Mariofanna Milanova（阿肯色大学小石城分校）、Srinivasan Parthasarathy（俄亥俄州立大学）、Zbigniew W. Ras（北卡罗莱纳大学夏洛特分校）、Xintao Wu（北卡罗莱纳大学夏洛特分校）和 Mohammed J. Zaki（伦斯勒理工学院）。

目　录

第 1 章　绪论 …… 1
1.1　什么是数据挖掘 …… 2
1.2　数据挖掘要解决的问题 …… 2
1.3　数据挖掘的起源 …… 3
1.4　数据挖掘任务 …… 4
1.5　本书的内容与组织 …… 7
文献注释 …… 7
参考文献 …… 8
习题 …… 10

第 2 章　数据 …… 13
2.1　数据类型 …… 14
2.1.1　属性与度量 …… 15
2.1.2　数据集的类型 …… 18
2.2　数据质量 …… 22
2.2.1　测量和数据收集问题 …… 22
2.2.2　关于应用的问题 …… 26
2.3　数据预处理 …… 27
2.3.1　聚集 …… 27
2.3.2　抽样 …… 28
2.3.3　维归约 …… 30
2.3.4　特征子集选择 …… 31
2.3.5　特征创建 …… 33
2.3.6　离散化和二元化 …… 34
2.3.7　变量变换 …… 38
2.4　相似性和相异性的度量 …… 38
2.4.1　基础 …… 39
2.4.2　简单属性之间的相似度和相异度 …… 40
2.4.3　数据对象之间的相异度 …… 41
2.4.4　数据对象之间的相似度 …… 43
2.4.5　邻近性度量的例子 …… 43
2.4.6　邻近度计算问题 …… 48
2.4.7　选取正确的邻近性度量 …… 50
文献注释 …… 50
参考文献 …… 52
习题 …… 53

第 3 章　探索数据 …… 59
3.1　鸢尾花数据集 …… 59
3.2　汇总统计 …… 60
3.2.1　频率和众数 …… 60
3.2.2　百分位数 …… 61
3.2.3　位置度量：均值和中位数 …… 61
3.2.4　散布度量：极差和方差 …… 62
3.2.5　多元汇总统计 …… 63
3.2.6　汇总数据的其他方法 …… 64
3.3　可视化 …… 64
3.3.1　可视化的动机 …… 64
3.3.2　一般概念 …… 65
3.3.3　技术 …… 67
3.3.4　可视化高维数据 …… 75
3.3.5　注意事项 …… 79
3.4　OLAP 和多维数据分析 …… 79
3.4.1　用多维数组表示鸢尾花数据 …… 80
3.4.2　多维数据：一般情况 …… 81
3.4.3　分析多维数据 …… 82
3.4.4　关于多维数据分析的最后评述 …… 84
文献注释 …… 84
参考文献 …… 85
习题 …… 86

第 4 章　分类：基本概念、决策树与模型评估 …… 89
4.1　预备知识 …… 89
4.2　解决分类问题的一般方法 …… 90
4.3　决策树归纳 …… 92
4.3.1　决策树的工作原理 …… 92
4.3.2　如何建立决策树 …… 93
4.3.3　表示属性测试条件的方法 …… 95
4.3.4　选择最佳划分的度量 …… 96
4.3.5　决策树归纳算法 …… 101
4.3.6　例子：Web 机器人检测 …… 102
4.3.7　决策树归纳的特点 …… 103
4.4　模型的过分拟合 …… 106

4.4.1 噪声导致的过分拟合……107
4.4.2 缺乏代表性样本导致的过分拟合……109
4.4.3 过分拟合与多重比较过程……109
4.4.4 泛化误差估计……110
4.4.5 处理决策树归纳中的过分拟合……113
4.5 评估分类器的性能……114
4.5.1 保持方法……114
4.5.2 随机二次抽样……115
4.5.3 交叉验证……115
4.5.4 自助法……115
4.6 比较分类器的方法……116
4.6.1 估计准确度的置信区间……116
4.6.2 比较两个模型的性能……117
4.6.3 比较两种分类法的性能……118
文献注释……118
参考文献……120
习题……122

第 5 章 分类：其他技术……127

5.1 基于规则的分类器……127
5.1.1 基于规则的分类器的工作原理……128
5.1.2 规则的排序方案……129
5.1.3 如何建立基于规则的分类器……130
5.1.4 规则提取的直接方法……130
5.1.5 规则提取的间接方法……135
5.1.6 基于规则的分类器的特征……136
5.2 最近邻分类器……137
5.2.1 算法……138
5.2.2 最近邻分类器的特征……138
5.3 贝叶斯分类器……139
5.3.1 贝叶斯定理……139
5.3.2 贝叶斯定理在分类中的应用……140
5.3.3 朴素贝叶斯分类器……141
5.3.4 贝叶斯误差率……145
5.3.5 贝叶斯信念网络……147
5.4 人工神经网络……150
5.4.1 感知器……151
5.4.2 多层人工神经网络……153
5.4.3 人工神经网络的特点……155
5.5 支持向量机……156
5.5.1 最大边缘超平面……156
5.5.2 线性支持向量机：可分情况……157
5.5.3 线性支持向量机：不可分情况……162
5.5.4 非线性支持向量机……164
5.5.5 支持向量机的特征……168
5.6 组合方法……168
5.6.1 组合方法的基本原理……168
5.6.2 构建组合分类器的方法……169
5.6.3 偏倚-方差分解……171
5.6.4 装袋……173
5.6.5 提升……175
5.6.6 随机森林……178
5.6.7 组合方法的实验比较……179
5.7 不平衡类问题……180
5.7.1 可选度量……180
5.7.2 接受者操作特征曲线……182
5.7.3 代价敏感学习……184
5.7.4 基于抽样的方法……186
5.8 多类问题……187
文献注释……189
参考文献……190
习题……193

第 6 章 关联分析：基本概念和算法……201

6.1 问题定义……202
6.2 频繁项集的产生……204
6.2.1 先验原理……205
6.2.2 *Apriori* 算法的频繁项集产生……206
6.2.3 候选的产生与剪枝……208
6.2.4 支持度计数……210
6.2.5 计算复杂度……213
6.3 规则产生……215
6.3.1 基于置信度的剪枝……215
6.3.2 *Apriori* 算法中规则的产生……215
6.3.3 例：美国国会投票记录……217
6.4 频繁项集的紧凑表示……217
6.4.1 极大频繁项集……217
6.4.2 闭频繁项集……219
6.5 产生频繁项集的其他方法……221
6.6 FP 增长算法……223
6.6.1 FP 树表示法……224
6.6.2 FP 增长算法的频繁项集产生……225
6.7 关联模式的评估……228
6.7.1 兴趣度的客观度量……228
6.7.2 多个二元变量的度量……235
6.7.3 辛普森悖论……236
6.8 倾斜支持度分布的影响……237

文献注释 …… 240
参考文献 …… 244
习题 …… 250

第 7 章 关联分析：高级概念 …… 259

7.1 处理分类属性 …… 259
7.2 处理连续属性 …… 261
7.2.1 基于离散化的方法 …… 261
7.2.2 基于统计学的方法 …… 263
7.2.3 非离散化方法 …… 265
7.3 处理概念分层 …… 266
7.4 序列模式 …… 267
7.4.1 问题描述 …… 267
7.4.2 序列模式发现 …… 269
7.4.3 时限约束 …… 271
7.4.4 可选计数方案 …… 274
7.5 子图模式 …… 275
7.5.1 图与子图 …… 276
7.5.2 频繁子图挖掘 …… 277
7.5.3 类 *Apriori* 方法 …… 278
7.5.4 候选产生 …… 279
7.5.5 候选剪枝 …… 282
7.5.6 支持度计数 …… 285
7.6 非频繁模式 …… 285
7.6.1 负模式 …… 285
7.6.2 负相关模式 …… 286
7.6.3 非频繁模式、负模式和负相关模式比较 …… 287
7.6.4 挖掘有趣的非频繁模式的技术 … 288
7.6.5 基于挖掘负模式的技术 …… 288
7.6.6 基于支持度期望的技术 …… 290
文献注释 …… 292
参考文献 …… 293
习题 …… 295

第 8 章 聚类分析：基本概念和算法 …… 305

8.1 概述 …… 306
8.1.1 什么是聚类分析 …… 306
8.1.2 不同的聚类类型 …… 307
8.1.3 不同的簇类型 …… 308
8.2 K 均值 …… 310
8.2.1 基本 K 均值算法 …… 310
8.2.2 K 均值：附加的问题 …… 315
8.2.3 二分 K 均值 …… 316
8.2.4 K 均值和不同的簇类型 …… 317
8.2.5 优点与缺点 …… 318
8.2.6 K 均值作为优化问题 …… 319
8.3 凝聚层次聚类 …… 320
8.3.1 基本凝聚层次聚类算法 …… 321
8.3.2 特殊技术 …… 322
8.3.3 簇邻近度的 Lance-Williams 公式 …… 325
8.3.4 层次聚类的主要问题 …… 326
8.3.5 优点与缺点 …… 327
8.4 DBSCAN …… 327
8.4.1 传统的密度：基于中心的方法 … 327
8.4.2 DBSCAN 算法 …… 328
8.4.3 优点与缺点 …… 329
8.5 簇评估 …… 330
8.5.1 概述 …… 332
8.5.2 非监督簇评估：使用凝聚度和分离度 …… 332
8.5.3 非监督簇评估：使用邻近度矩阵 …… 336
8.5.4 层次聚类的非监督评估 …… 338
8.5.5 确定正确的簇个数 …… 339
8.5.6 聚类趋势 …… 339
8.5.7 簇有效性的监督度量 …… 340
8.5.8 评估簇有效性度量的显著性 …… 343
文献注释 …… 344
参考文献 …… 345
习题 …… 347

第 9 章 聚类分析：其他问题与算法 …… 355

9.1 数据、簇和聚类算法的特性 …… 355
9.1.1 例子：比较 K 均值和 DBSCAN …… 355
9.1.2 数据特性 …… 356
9.1.3 簇特性 …… 357
9.1.4 聚类算法的一般特性 …… 358
9.2 基于原型的聚类 …… 359
9.2.1 模糊聚类 …… 359
9.2.2 使用混合模型的聚类 …… 362
9.2.3 自组织映射 …… 369
9.3 基于密度的聚类 …… 372
9.3.1 基于网格的聚类 …… 372

9.3.2 子空间聚类……374
9.3.3 DENCLUE：基于密度聚类的一种基于核的方案……377
9.4 基于图的聚类……379
9.4.1 稀疏化……379
9.4.2 最小生成树聚类……380
9.4.3 OPOSSUM：使用 METIS 的稀疏相似度最优划分……381
9.4.4 Chameleon：使用动态建模的层次聚类……381
9.4.5 共享最近邻相似度……385
9.4.6 Jarvis-Patrick 聚类算法……387
9.4.7 SNN 密度……388
9.4.8 基于 SNN 密度的聚类……389
9.5 可伸缩的聚类算法……390
9.5.1 可伸缩：一般问题和方法……391
9.5.2 BIRCH……392
9.5.3 CURE……393
9.6 使用哪种聚类算法……395
文献注释……397
参考文献……398
习题……400

第 10 章 异常检测……403
10.1 预备知识……404
10.1.1 异常的成因……404
10.1.2 异常检测方法……404
10.1.3 类标号的使用……405
10.1.4 问题……405
10.2 统计方法……406
10.2.1 检测一元正态分布中的离群点……407
10.2.2 多元正态分布的离群点……408
10.2.3 异常检测的混合模型方法……410
10.2.4 优点与缺点……411
10.3 基于邻近度的离群点检测……411
10.4 基于密度的离群点检测……412
10.4.1 使用相对密度的离群点检测……413
10.4.2 优点与缺点……414
10.5 基于聚类的技术……414
10.5.1 评估对象属于簇的程度……415
10.5.2 离群点对初始聚类的影响……416
10.5.3 使用簇的个数……416
10.5.4 优点与缺点……416
文献注释……417
参考文献……418
习题……420

附录 A 线性代数……423
附录 B 维归约……433
附录 C 概率统计……445
附录 D 回归……451
附录 E 优化……457

第1章 绪 论

数据收集和数据存储技术的快速进步使得各组织机构可以积累海量数据。然而，提取有用的信息已经成为巨大的挑战。通常，由于数据量太大，无法使用传统的数据分析工具和技术处理它们。有时，即使数据集相对较小，但由于数据本身具有一些非传统特点，也不能使用传统的方法处理。在另外一些情况下，面临的问题不能使用已有的数据分析技术来解决。这样，就需要开发新的方法。

数据挖掘是一种技术，它将传统的数据分析方法与处理大量数据的复杂算法相结合。数据挖掘为探查和分析新的数据类型以及用新方法分析旧有数据类型提供了令人振奋的机会。本章，我们概述数据挖掘，并列举本书所涵盖的关键主题。我们首先介绍需要新的数据分析技术的一些大家熟知的应用。

商务 借助POS（销售点）数据收集技术[条码扫描器、射频识别（RFID）和智能卡技术]，零售商可以在其商店的收银台收集顾客购物的最新数据。零售商可以利用这些信息，加上电子商务网站的日志、电购中心的顾客服务记录等其他的重要商务数据，更好地理解顾客的需求，做出明智的商务决策。

数据挖掘技术可以用来支持广泛的商务智能应用，如顾客分析、定向营销、工作流管理、商店分布和欺诈检测等。数据挖掘还能帮助零售商回答一些重要的商务问题，如“谁是最有价值的顾客？”“什么产品可以交叉销售[①]或提升销售[②]？”“公司明年的收入前景如何？”这些问题催生了一种新的数据分析技术——关联分析（见第6章和第7章）。

医学、科学与工程 医学、科学与工程技术界的研究者正在快速积累大量数据，这些数据对获得有价值的新发现至关重要。例如，为了更深入地理解地球的气候系统，NASA已经部署了一系列的地球轨道卫星，不停地收集地表、海洋和大气的全球观测数据。然而，由于这些数据的规模和时空特性，传统的方法常常不适合分析这些数据集。数据挖掘开发的技术可以帮助地球科学家回答如下问题：“干旱和飓风等生态系统扰动的频度和强度与全球变暖之间有何联系？”“海洋表面温度对地表降水量和温度有何影响？”“如何准确地预测一个地区的生长季节的开始和结束？”

再举一个例子，分子生物学研究者希望利用当前收集的大量基因组数据，更好地理解基因的结构和功能。过去，传统方法只允许科学家在一个实验中每次研究少量基因。微阵列技术的最新突破已经能让科学家在多种情况下，比较数以千计的基因特性。这种比较有助于确定每个基因的作用，或许可以查出导致特定疾病的基因。然而，由于数据的噪声和高维性，需要新的数据分析

① cross-sell，指根据顾客的兴趣推荐或显示相关商品以增加销售机会。——译者注

② up-sell，指尝试向曾经购买的顾客销售价格更高的商品。——译者注

方法。除分析基因序列数据外，数据挖掘还能用来处理生物学的其他难题，如蛋白质结构预测、多序列校准、生物化学路径建模和种系发生学。

1.1　什么是数据挖掘

数据挖掘是在大型数据存储库中，自动地发现有用信息的过程。数据挖掘技术用来探查大型数据库，发现先前未知的有用模式。数据挖掘还可以预测未来观测结果，例如，预测一位新的顾客是否会在一家百货公司消费 100 美元以上。

并非所有的信息发现任务都被视为数据挖掘。例如，使用数据库管理系统查找个别的记录，或通过因特网的搜索引擎查找特定的 Web 页面，则是**信息检索**（information retrieval）领域的任务。虽然这些任务非常重要，可能涉及使用复杂的算法和数据结构，但是它们主要依赖传统的计算机科学技术和数据的明显特征来创建索引结构，从而有效地组织和检索信息。尽管如此，人们也在利用数据挖掘技术增强信息检索系统的能力。

数据挖掘与知识发现

数据挖掘是**数据库中知识发现**（knowledge discovery in database，KDD）不可缺少的一部分，而 KDD 是将未加工的数据转换为有用信息的整个过程，如图 1-1 所示。该过程包括一系列转换步骤，从数据的预处理到数据挖掘结果的后处理。

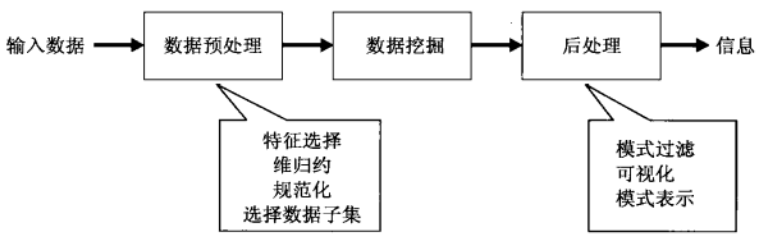


图 1-1　数据库中知识发现（KDD）过程

输入数据可以以各种形式存储（平展文件、电子数据表或关系表），并且可以驻留在集中的数据存储库中，或分布在多个站点上。数据**预处理**（preprocessing）的目的是将未加工的输入数据转换成适合分析的形式。数据预处理涉及的步骤包括融合来自多个数据源的数据，清洗数据以消除噪声和重复的观测值，选择与当前数据挖掘任务相关的记录和特征。由于收集和存储数据的方式多种多样，数据预处理可能是整个知识发现过程中最费力、最耗时的步骤。

“结束循环”（closing the loop）通常指将数据挖掘结果集成到决策支持系统的过程。例如，在商业应用中，数据挖掘的结果所揭示的规律可以结合商业活动管理工具，从而开展或测试有效的商品促销活动。这样的结合需要**后处理**（postprocessing）步骤，确保只将那些有效的和有用的结果集成到决策支持系统中。后处理的一个例子是可视化（见第 3 章），它使得数据分析者可以从各种不同的视角探查数据和数据挖掘结果。在后处理阶段，还能使用统计度量或假设检验，删除虚假的数据挖掘结果。

1.2　数据挖掘要解决的问题

前面提到，面临新的数据集带来的问题时，传统的数据分析技术常常遇到实际困难。下面是

一些具体的问题，它们引发了人们对数据挖掘开展研究。

可伸缩 由于数据产生和收集技术的进步，数吉字节、数太字节甚至数拍字节[①]的数据集越来越普遍。如果数据挖掘算法要处理这些海量数据集，则算法必须是可伸缩的（scalable）。许多数据挖掘算法使用特殊的搜索策略处理指数级搜索问题。为实现可伸缩可能还需要实现新的数据结构，才能以有效的方式访问每个记录。例如，当要处理的数据不能放进内存时，可能需要非内存算法。使用抽样技术或开发并行和分布算法也可以提高可伸缩程度。

高维性 现在，常常遇到具有成百上千属性的数据集，而不是几十年前常见的只具有少量属性的数据集。在生物信息学领域，微阵列技术的进步已经产生了涉及数千特征的基因表达数据。具有时间或空间分量的数据集也经常具有很高的维度。例如，考虑包含不同地区的温度测量结果的数据集，如果在一个相当长的时间周期内反复地测量，则维度（特征数）的增长正比于测量的次数。为低维数据开发的传统的数据分析技术通常不能很好地处理这样的高维数据。此外，对于某些数据分析算法，随着维度（特征数）的增加，计算复杂性迅速增加。

异种数据和复杂数据 通常，传统的数据分析方法只处理包含相同类型属性的数据集，或者是连续的，或者是分类的。随着数据挖掘在商务、科学、医学和其他领域的作用越来越大，越来越需要能够处理异种属性的技术。近年来，已经出现了更复杂的数据对象。这些非传统的数据类型的例子有：含有半结构化文本和超链接的 Web 页面集、具有序列和三维结构的 DNA 数据、包含地球表面不同位置上的时间序列测量值（温度、气压等）的气象数据。为挖掘这种复杂对象而开发的技术应当考虑数据中的联系，如时间和空间的自相关性、图的连通性、半结构化文本和 XML 文档中元素之间的父子联系。

数据的所有权与分布 有时，需要分析的数据并非存放在一个站点，或归属一个机构，而是地理上分布在属于多个机构的资源中。这就需要开发分布式数据挖掘技术。分布式数据挖掘算法面临的主要挑战包括：(1) 如何降低执行分布式计算所需的通信量？(2) 如何有效地统一从多个资源得到的数据挖掘结果？(3) 如何处理数据安全性问题？

非传统的分析 传统的统计方法基于一种假设-检验模式，即提出一种假设，设计实验来收集数据，然后针对假设分析数据。但是，这一过程劳力费神。当前的数据分析任务常常需要产生和评估数千种假设，因此需要自动地产生和评估假设，这促使人们开发了一些数据挖掘技术。此外，数据挖掘所分析的数据集通常不是精心设计的实验的结果，并且它们通常代表数据的时机性样本（opportunistic sample），而不是随机样本（random sample）。而且，这些数据集常常涉及非传统的数据类型和数据分布。

1.3 数据挖掘的起源

为迎接上述这些挑战，来自不同学科的研究者汇集到一起，开始着手开发可以处理不同数据类型的更有效的、可伸缩的工具。这些工作都是建立在研究者先前使用的方法学和算法之上，而在数据挖掘领域达到高潮。特别地，数据挖掘利用了来自如下一些领域的思想：(1) 来自统计学的抽样、估计和假设检验，(2) 人工智能、模式识别和机器学习的搜索算法、建模技术和学习理

① gigabytes, terabytes, petabytes 分别是 10^9, 10^{12}, 10^{15} 字节。——编者注

论。数据挖掘也迅速地接纳了来自其他领域的思想，这些领域包括最优化、进化计算、信息论、信号处理、可视化和信息检索。

一些其他领域也起到重要的支撑作用。特别地，需要数据库系统提供有效的存储、索引和查询处理支持。源于高性能（并行）计算的技术在处理海量数据集方面常常是重要的。分布式技术也能帮助处理海量数据，并且当数据不能集中到一起处理时更是至关重要。

图 1-2 展示数据挖掘与其他领域之间的联系。

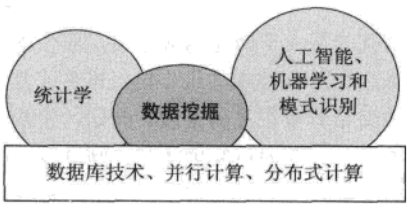


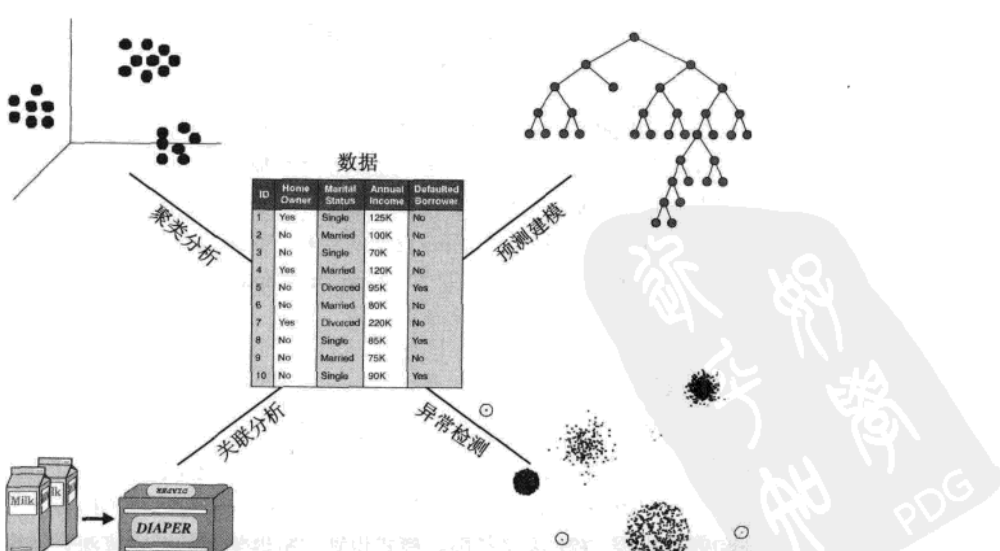
图 1-2　数据挖掘汇集了许多学科的知识

1.4　数据挖掘任务

通常，数据挖掘任务分为下面两大类。

- **预测任务**。这些任务的目标是根据其他属性的值，预测特定属性的值。被预测的属性一般称**目标变量**（target variable）或**因变量**（dependent variable），而用来做预测的属性称**说明变量**（explanatory variable）或**自变量**（independent variable）。
- **描述任务**。其目标是导出概括数据中潜在联系的模式（相关、趋势、聚类、轨迹和异常）。本质上，描述性数据挖掘任务通常是探查性的，并且常常需要后处理技术验证和解释结果。

图 1-3 展示本书其余部分讲述的四种主要数据挖掘任务。



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	80K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

图 1-3　四种主要数据挖掘任务

预测建模（predictive modeling） 涉及以说明变量函数的方式为目标变量建立模型。有两类预测建模任务：**分类**（classification），用于预测离散的目标变量；**回归**（regression），用于预测连续的目标变量。例如，预测一个 Web 用户是否会在网上书店买书是分类任务，因为该目标变量是二值的，而预测某股票的未来价格则是回归任务，因为价格具有连续值属性。两项任务目标都是训练一个模型，使目标变量预测值与实际值之间的误差达到最小。预测建模可以用来确定顾客对产品促销活动的反应，预测地球生态系统的扰动，或根据检查结果判断病人是否患有某种疾病。

例 1.1 预测花的类型 考虑如下任务：根据花的特征预测花的种类。本例考虑根据是否属于 Setosa、Versicolour、Virginica 这三类之一对鸢尾花（Iris）进行分类。为进行这一任务，我们需要一个数据集，包含这三类花的特性。一个具有这类信息的数据集是著名的鸢尾花数据集，可从加州大学欧文分校的机器学习数据库中得到（`http://www.ics.uci.edu/~mlearn`）。除花的种类之外，该数据集还包含萼片宽度、萼片长度、花瓣长度和花瓣宽度四个其他属性。（鸢尾花数据集和它的属性将在 3.1 节进一步介绍。）图 1-4 给出鸢尾花数据集中 150 种花的花瓣宽度与花瓣长度的对比图。花瓣宽度分成 *low*、*medium*、*high* 三类，分别对应于区间[0, 0.75)、[0.75, 1.75)、[1.75, ∞)。花瓣长度也分成 *low*、*medium*、*high* 三类，分别对应于区间[0, 2.5)、[2.5, 5)、[5, ∞)。根据花瓣宽度和长度的这些类别，可以推出如下规则。

花瓣宽度和花瓣长度为 low 蕴涵 Setosa。
花瓣宽度和花瓣长度为 medium 蕴涵 Versicolour。
花瓣宽度和花瓣长度为 high 蕴涵 Virginica。

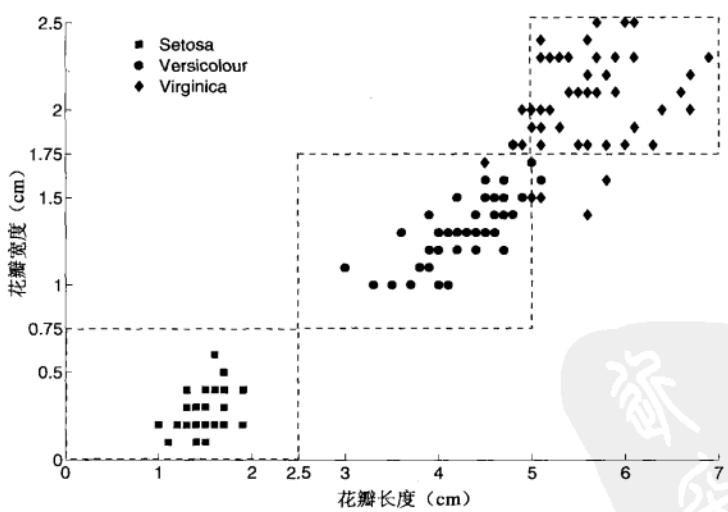


图 1-4 150 种鸢尾花的宽度与长度对比

尽管这些规则不能对所有的花进行分类，但是已经可以对大多数花很好地进行分类（尽管不完善）。注意：根据花瓣宽度和花瓣长度，Setosa 种类的花完全可以与 Versicolour 和 Virginica 种类的花分开，但是后两类花在这些属性上有一些重叠。 □

关联分析（association analysis）　用来发现描述数据中强关联特征的模式。所发现的模式通常用蕴涵规则或特征子集的形式表示。由于搜索空间是指数规模的，关联分析的目标是以有效的方式提取最有趣的模式。关联分析的应用包括找出具有相关功能的基因组、识别用户一起访问的Web页面、理解地球气候系统不同元素之间的联系等。

例1.2　购物篮分析　表1-1给出的事务是在一家杂货店收银台收集的销售数据。关联分析可以用来发现顾客经常同时购买的商品。例如，我们可能发现规则{尿布}→{牛奶}。该规则暗示购买尿布的顾客多半会购买牛奶。这种类型的规则可以用来发现各类商品中可能存在的交叉销售的商机。　□

表1-1　购物篮数据

事务ID	商　品
1	{面包，黄油，尿布，牛奶}
2	{咖啡，糖，小甜饼，鲑鱼}
3	{面包，黄油，咖啡，尿布，牛奶，鸡蛋}
4	{面包，黄油，鲑鱼，鸡}
5	{鸡蛋，面包，黄油}
6	{鲑鱼，尿布，牛奶}
7	{面包，茶，糖，鸡蛋}
8	{咖啡，糖，鸡，鸡蛋}
9	{面包，尿布，牛奶，盐}
10	{茶，鸡蛋，小甜饼，尿布，牛奶}

聚类分析（cluster analysis）　旨在发现紧密相关的观测值组群，使得与属于不同簇的观测值相比，属于同一簇的观测值相互之间尽可能类似。聚类可用来对相关的顾客分组、找出显著影响地球气候的海洋区域以及压缩数据等。

例1.3　文档聚类　表1-2给出的新闻文章可以根据它们各自的主题分组。每篇文章表示为词-频率对(w, c)的集合，其中w是词，而c是该词在文章中出现的次数。在该数据集中，有两个自然簇。第一个簇由前四篇文章组成，对应于经济新闻，而第二个簇包含后四篇文章，对应于卫生保健新闻。一个好的聚类算法应当能够根据文章中出现的词的相似性，识别这两个簇。

表1-2　新闻文章集合

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

□

异常检测（anomaly detection）的任务是识别其特征显著不同于其他数据的观测值。这样的观测值称为**异常点**（anomaly）或**离群点**（outlier）。异常检测算法的目标是发现真正的异常点，而避免错误地将正常的对象标注为异常点。换言之，一个好的异常检测器必须具有高检测率和低

误报率。异常检测的应用包括检测欺诈、网络攻击、疾病的不寻常模式、生态系统扰动等。

例 1.4 信用卡欺诈检测 信用卡公司记录每个持卡人所做的交易，同时也记录信用限度、年龄、年薪和地址等个人信息。由于与合法交易相比，欺诈行为的数目相对较少，因此异常检测技术可以用来构造用户的合法交易的轮廓。当一个新的交易到达时就与之比较。如果该交易的特性与先前所构造的轮廓很不相同，就把交易标记为可能是欺诈。 □

1.5 本书的内容与组织

本书从算法的角度介绍数据挖掘所使用的主要原理与技术。为了更好地理解数据挖掘技术如何用于各种类型的数据，研究这些原理与技术是至关重要的。对于有志于从事这个领域研究的读者，本书也可作为一个起点。

我们从数据（第 2 章）开始本书的技术讨论。该章讨论数据的基本类型、数据质量、预处理技术以及相似性和相异性度量。这些材料尽管可以快速阅读，但它却是数据分析的重要基础。第 3 章论及数据探查，讨论汇总统计、可视化技术和联机分析处理（On-Line Analytical Processing, OLAP），这些技术可用来快速透彻理解数据集。

第 4 章和第 5 章涵盖分类。第 4 章是基础，讨论决策树分类和一些重要的分类问题：过分拟合、性能评估和不同分类模型的比较。在此基础上，第 5 章介绍其他重要的分类技术：基于规则的系统、最近邻分类器、贝叶斯分类器、人工神经网络、支持向量机以及组合分类器。组合分类器是一组分类器。这一章还讨论多类问题和不平衡类问题。这些主题可以彼此独立地学习。

关联分析在第 6 章和第 7 章考察。第 6 章介绍关联分析的基础——频繁项集、关联规则以及产生它们的一些算法。特殊类型频繁项集（极大项集、闭项集和超团集）对于数据挖掘都是重要的，也在这一章讨论。该章最后讨论关联分析的评估度量。第 7 章考虑各种更高级的专题，包括如何将关联分析用于分类数据和连续数据，或用于具有概念分层的数据。（概念分层是对象的层次分类，例如库存商品→服装→鞋→运动鞋。）该章还介绍如何扩展关联分析，以发现序列模式（涉及次序的模式）、图中的模式、负联系（如果一个项出现，则其他项不出现）。

聚类分析在第 8 章和第 9 章讨论。第 8 章先介绍不同类型的簇，然后给出三种特定的聚类技术：K 均值、凝聚层次聚类和 DBSCAN。接下去讨论验证聚类算法结果的技术。更多的聚类概念和技术在第 9 章考察，包括模糊和概率聚类、自组织映射（SOM）、基于图的聚类和基于密度的聚类。这一章还讨论可伸缩问题和选择聚类算法需要考虑的因素。

最后一章（第 10 章）是关于异常检测的。在给出一些基本定义之后，介绍了若干类型的异常检测，包括统计的、基于距离的、基于密度的和基于聚类的。

尽管与统计学和机器学习相比，数据挖掘还很年轻，但是数据挖掘学科领域已经太大，很难用一本书涵盖。对于本书仅简略涉及的主题（如数据质量），我们在相应章的文献注释部分选列了一些参考文献。对于本书未涵盖的主题（如流数据挖掘和隐私保护数据挖掘），参考文献在本章下面的文献注释提供。

文献注释

数据挖掘已有许多教科书。引论性教科书包括 Dunham[10]、Han 和 Kamber[21]、Hand 等[23]以及 Roiger 和 Geatz[36]。更侧重于商务应用的数据挖掘书籍包括 Berry 和 Linoff[2]、Pyle[34]和

Parr Rud[33]。侧重统计学习的书籍包括 Cherkassky 和 Mulier[6]和 Hastie 等[24]。侧重机器学习或模式识别的一些书包括 Duda 等[9]、Kantardzic[25]，Mitchell[31]、Webb[41]以及 Witten 和 Frank[42]。还有一些更专业的书：Chakrabarti[4]（Web 挖掘）、Fayyad 等[13]（数据挖掘早期文献汇编）、Fayyad 等[11]（可视化）、Grossman 等[18]（科学与工程技术）、Kargupta 和 Chan[26]（分布式数据挖掘）、Wang 等[40]（生物信息学）以及 Zaki 和 Ho[44]（并行数据挖掘）。

有许多与数据挖掘相关的会议。致力于该领域研究的一些主要会议包括 ACM SIGKDD 知识发现与数据挖掘国际会议（KDD）、IEEE 数据挖掘国际会议（ICDM）、SIAM 数据挖掘国际会议（SDM）、欧洲数据库中知识发现的原理与实践会议（PKDD）和亚太知识发现与数据挖掘会议（PAKDD）。数据挖掘的文章也可以在其他主要会议上找到，如 ACM SIGMOD/PODS 会议、超大型数据库国际会议（VLDB）、信息与知识管理会议（CIKM）和数据工程国际会议（ICDE）、机器学习国际会议（ICML）以及人工智能全国学术会议（AAAI）。

数据挖掘方面的期刊包括《IEEE 知识与数据工程汇刊》（*IEEE Transactions on Knowledge and Data Engineering*）、《数据挖掘与知识发现》（*Data Mining and Knowledge Discovery*）、《知识与信息系统》（*Knowledge and Information Systems*）、《智能数据分析》（*Intelligent Data Analysis*）、《信息系统》（*Information Systems*）和《智能信息系统杂志》（*Journal of Intelligent Information Systems*）。

有大量数据挖掘的一般性文章界定该领域及其与其他领域（特别是与统计学）之间的联系。Fayyad等[12]介绍数据挖掘，以及如何将它与整个知识发现过程协调。Chen等[5]从数据库角度阐释数据挖掘。Ramakrishnan和Grama[35]给出数据挖掘的一般讨论，并提出若干观点。与Friedman[14]一样，Hand[22]讨论数据挖掘与统计学的区别。Lambert[29]考察统计学在大型数据集上的应用，并对数据挖掘与统计学各自的角色提出一些评论。Glymour等[16]考虑统计学可能为数据挖掘提供的教训。Smyth等[38]讨论诸如数据流、图形和文本等新的数据类型和应用如何推动数据挖掘演变。新出现的数据挖掘应用也被Han等[20]考虑，而Smyth[37]介绍数据挖掘研究所面临的一些挑战。Wu等[43]讨论如何将数据挖掘研究成果转化成实际工具。数据挖掘标准是Grossman等的文章[17]的主题。Bradley[3]讨论如何将数据挖掘算法扩展到大型数据集。

随着数据挖掘新的应用的出现，数据挖掘面临新的挑战。例如，近年来人们对数据挖掘破坏隐私问题的关注逐步上升，在电子商务和卫生保健领域的应用尤其如此。这样，人们对开发保护用户隐私的数据挖掘算法的兴趣逐步上升。为挖掘加密数据或随机数据而开发的技术称作**保护隐私的数据挖掘**。该领域的一些一般文献包括 Agrawal 和 Srikant 的文章[1]，Clifton 等[7]和 Kargupta 等[27]。Vassilios 等[39]提供一个综述。

近年来，我们看到快速产生连续的数据流的应用逐渐增加。数据流应用的例子包括网络通信流、多媒体流和股票价格。挖掘数据流时，必须考虑一些因素，如可用内存有限、需要联机分析、数据随时间而变等。流数据挖掘已经成为数据挖掘的一个重要领域。有关参考文献有 Domingos 和 Hulten[8]（分类）、Giannella 等[15]（关联分析）、Guha 等[19]（聚类）、Kifer 等[28]（变化检测）、Papadimitriou 等[32]（时间序列）以及 Law 等[30]（维归约）。

参考文献

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

[2] M. J. A. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer*

Relationship Management. Wiley Computer Publishing, 2nd edition, 2004.

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

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

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

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

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

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

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

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

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

[12] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*, pages1 - 34. AAAI Press, 1996.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Melbourne, Florida, December 2003. IEEE Computer Society.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

习　题

1. 讨论下列每项活动是否是数据挖掘任务。
   (a) 根据性别划分公司的顾客。
   (b) 根据可赢利性划分公司的顾客。
   (c) 计算公司的总销售额。
   (d) 按学生的标识号对学生数据库排序。
   (e) 预测掷一对骰子的结果。
   (f) 使用历史记录预测某公司未来的股票价格。
   (g) 监视病人心率的异常变化。
   (h) 监视地震活动的地震波。
   (i) 提取声波的频率。
2. 假定你是一个数据挖掘顾问，受雇于一家因特网搜索引擎公司。举例说明如何使用诸如聚类、分类、关联规则挖掘和异常检测等技术，让数据挖掘为公司提供帮助。

3. 对于如下每个数据集，解释数据的私有性是否是重要问题。
   (a) 从 1900 年到 1950 年收集的人口普查数据。
   (b) 访问你的 Web 站点的用户的 IP 地址和访问时间。
   (c) 从地球轨道卫星得到的图像。
   (d) 电话号码簿上的姓名和地址。
   (e) 从网上收集的姓名和电子邮件地址。

第2章

数　　据

本章讨论一些与数据相关的问题，它们对于数据挖掘的成败至关重要。

数据类型　数据集的不同表现在多方面。例如，用来描述数据对象的属性可以具有不同的类型——定量的或定性的，并且数据集可能具有特定的性质，例如，某些数据集包含时间序列或彼此之间具有明显联系的对象。毫不奇怪，数据的类型决定我们应使用何种工具和技术来分析数据。此外，数据挖掘研究常常是为了适应新的应用领域和新的数据类型的需要而展开的。

数据的质量　数据通常远非完美。尽管大部分数据挖掘技术可以忍受某种程度的数据不完美，但是注重理解和提高数据质量将改进分析结果的质量。通常必须解决的数据质量问题包括存在噪声和离群点，数据遗漏、不一致或重复，数据有偏差或者不能代表它应该描述的现象或总体情况。

使数据适合挖掘的预处理步骤　通常，原始数据必须加以处理才能适合于分析。处理一方面是要提高数据的质量，另一方面要让数据更好地适应特定的数据挖掘技术或工具。例如，可能需要将连续值属性（如长度）转换成具有离散的分类值的属性（如短、中、长），以便应用特定的技术。又如，数据集属性的数目常常需要减少，因为属性较少时许多技术用起来更加有效。

根据数据联系分析数据　数据分析的一种方法是找出数据对象之间的联系，之后使用这些联系而不是数据对象本身来进行其余的分析。例如，我们可以计算对象之间的相似度或距离，然后根据这种相似度或距离进行分析——聚类、分类或异常检测。诸如此类的相似性或距离度量很多，要根据数据的类型和特定的应用做出正确的选择。

例 2.1　与数据相关的问题　为了进一步解释这些问题的重要性，考虑下面的假想情况。你收到某个医学研究者发来的电子邮件，是关于你想要研究的一个项目的。邮件的内容如下：

> 你好，
>
> 我已附上先前邮件提及的数据文件。每行包含一个病人的信息，由 5 个字段组成。我们想使用前面 4 个字段预测最后一个字段。因为我要出去几天，所以没有时间为你提供关于这些数据的更多信息，但希望不会耽误你太多时间。如果你不介意的话，我回来之后是否可以开会讨论你的初步结果？我可能会邀请我们小组的其他成员参加。
>
> 谢谢！几天之后见！

尽管有些疑虑，你还是开始着手分析这些数据。文件的前几行如下：

012　232　33.5　0　10.7

020　121　16.9　2　210.1
027　165　24.0　0　427.6
…

粗略观察这些数据并未发现什么不对。你抛开疑虑，并开始分析。数据文件只有1000行，比你希望的小，仅仅两天之后你认为你已经取得一些进展。你去参加会议，在等待其他人时，你开始与一位参与该项目工作的统计人员交谈。当听说你正在分析该项目的数据时，她请你向她简略介绍你的结果。

统计人员：哦，你得到了所有病人的数据？

数据挖掘者：是的。我还没有足够的时间分析，但是我的确有了一些有趣的结果。

统计人员：真棒。病人数据集的数据问题太多，我没什么进展。

数据挖掘者：啊？我没有听到任何问题。

统计人员：喔，首先是字段5，这是我们要预测的变量。分析这类数据的人都知道，如果使用这些值的日志，结果会更好，但是我们后来才发现这一点。他们告诉你了吗？

数据挖掘者：没有。

统计人员：你一定听说过字段4的问题了吧？它的测量范围应当是1到10，而0表示有遗漏的值。但是，由于数据输入错误，所有的10都变成了0。可是，由于有些病人这个字段的值有遗漏，所以不能确定该字段上的0实际是0还是10。不少记录都存在此问题。

数据挖掘者：有意思。还有其他问题吗？

统计人员：是的。字段2和3也有不少问题。我猜想你可能已经注意到了。

数据挖掘者：是的。但是，这些字段只是字段5的弱预测子。

统计人员：无论如何，尽管有这些问题，你还能够完成一些分析，这真让人吃惊。

数据挖掘者：实际上，我的结果相当好。字段1是字段5的很强的预测子。我很奇怪以前怎么没人注意到。

统计人员：什么？字段1只是一个标识号。

数据挖掘者：无论如何，我的结果在那儿。

统计人员：啊，不！我才想起来。在按字段5排序记录之后，我们加上了一个ID号。它们之间是存在很强的联系，但毫无意义。很抱歉！ □

尽管这一场景代表一种极端情况，但它强调“了解数据”的重要性。为此，本章将处理上面提到的四个问题，列举一些基本难点和标准解决方法。

2.1　数据类型

通常，**数据集**可以看作**数据对象**的集合。数据对象有时也叫做记录、点、向量、模式、事件、案例、样本、观测或实体。数据对象用一组刻画对象基本特性（如物体质量或事件发生时间）的**属性**描述。属性有时也叫做变量、特性、字段、特征或维。

例2.2　学生信息　通常，数据集是一个文件，其中对象是文件的记录（或行），而每个字段（或列）对应于一个属性。例如，表2-1显示包含学生信息的数据集。每行对应于一个学生，

而每列是一个属性，描述学生的某一方面，如平均成绩（GPA）或标识号（ID）。

表 2-1 包含学生信息的样本数据集

学生 ID	年　级	平均成绩（GPA）	…
	⋮		
1034262	四年级	3.24	…
1052663	二年级	3.51	…
1082246	一年级	3.62	…
	⋮		

□

基于记录的数据集在平展文件或关系数据库系统中是最常见的，但是还有其他类型的数据集和存储数据的系统。在 2.1.2 节，我们将讨论数据挖掘经常遇到的其他类型的数据集。然而，我们先考虑属性。

2.1.1 属性与度量

本节我们考虑使用何种类型的属性描述数据对象，来处理描述数据的问题。我们首先定义属性，然后考虑属性类型的含义，最后介绍经常遇到的属性类型。

1. 什么是属性

我们先更详细地定义属性。

定义 2.1 **属性**（attribute）是对象的性质或特性，它因对象而异，或随时间而变化。

例如，眼球颜色因人而异，而物体的温度随时间而变。注意：眼球颜色是一种符号属性，具有少量可能的值{棕色，黑色，蓝色，绿色，淡褐色，……}，而温度是数值属性，可以取无穷多个值。

追根溯源，属性并非数字或符号。然而，为了讨论和精细地分析对象的特性，我们为它们赋予了数字或符号。为了用一种明确定义的方式做到这一点，我们需要测量标度。

定义 2.2 **测量标度**（measurement scale）是将数值或符号值与对象的属性相关联的规则（函数）。

形式上，测量过程是使用测量标度将一个值与一个特定对象的特定属性相关联。这看上去有点抽象，但是任何时候，我们总在进行这样的测量过程。例如，踏上浴室的磅秤称体重；将人分为男女；清点会议室的椅子数目，确定是否能够为所有与会者提供足够的座位。在所有这些情况下，对象属性的“物理值”都被映射到数值或符号值。

有了这些背景，现在我们可以讨论属性类型，这对于确定特定的数据分析技术是否适用于某种具体的属性是一个重要的概念。

2. 属性类型

从前面的讨论显而易见，属性的性质不必与用来度量它的值的性质相同。换句话说，用来代表属性的值可能具有不同于属性本身的性质，并且反之亦然。我们用两个例子解释。

例 2.3 **雇员年龄和 ID 号** 与雇员有关的两个属性是 ID 和年龄，这两个属性都可以用整数表示。然而，谈论雇员的平均年龄是有意义的，但是谈论雇员的平均 ID 却毫无意义。的确，我们希望 ID 属性所表达的唯一方面是它们互不相同。因而，对雇员 ID 的唯一合法操作就是判定它

们是否相等。但在使用整数表示雇员 ID 时，并没暗示有此限制。对于年龄属性而言，用来表示年龄的整数的性质与该属性的性质大同小异。尽管如此，这种对应仍不完备，例如，年龄有最大值，而整数没有。 □

例 2.4　线段长度　考虑图 2-1，它展示一些线段对象和如何用两种不同的方法将这些对象的长度属性映射到整数。从上到下，每条后继线段都是通过最上面的线段自我添加而形成的。这样，第二条线段是最上面的线段两次相连而形成的，第三条线段是最上面的线段三次相连而形成的，依次类推。从物理意义上讲，所有的线段都是第一条线段的倍数。这个事实由图右边的测量捕获，但未被左边的测量捕获。更准确地说，左边的测量标度仅仅捕获长度属性的序，而右边的标度同时捕获序和可加性的性质。因此，属性可以用一种不描述属性全部性质的方式测量。 □

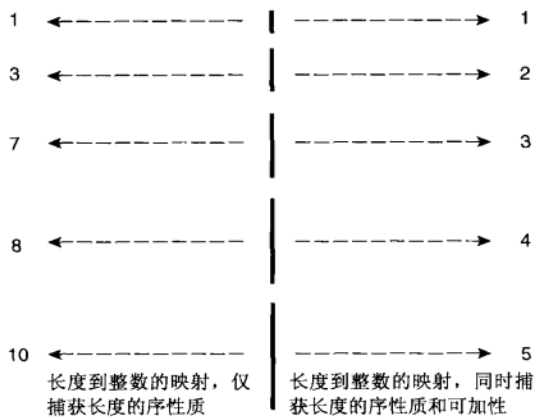


图 2-1　两种不同的测量标度下的线段长度测量

属性的类型告诉我们，属性的哪些性质反映在用于测量它的值中。知道属性的类型是重要的，因为它告诉我们测量值的哪些性质与属性的基本性质一致，从而使得我们可以避免诸如计算雇员的平均 ID 这样的愚蠢行为。注意，通常将属性的类型称作**测量标度的类型**。

3. 属性的不同类型

一种指定属性类型的有用（和简单）的办法是，确定对应于属性基本性质的数值的性质。例如，长度的属性可以有数值的许多性质。按照长度比较对象，确定对象的排序，以及谈论长度的差和比例都是有意义的。数值的如下性质（操作）常常用来描述属性。

(1) **相异性**　= 和 ≠。

(2) **序**　<、≤、>和≥。

(3) **加法**　+和 −。

(4) **乘法**　*和/。

给定这些性质，我们可以定义四种属性类型：**标称**（nominal）、**序数**（ordinal）、**区间**（interval）和**比率**（ratio）。表 2-2 给出这些类型的定义，以及每种类型上有哪些合法的统计操作等信息。每种属性类型拥有其上方属性类型上的所有性质和操作。因此，对于标称、序数和区间属性合法的任何性质或操作，对于比率属性也合法。换句话说，属性类型的定义是累积的。当然，对于某种属性类型合适的操作，对其上方的属性类型就不一定合适。

表 2-2 不同的属性类型

属性类型		描述	例子	操作
分类的（定性的）	标称	标称属性的值仅仅只是不同的名字，即标称值只提供足够的信息以区分对象（=，≠）	邮政编码、雇员 ID 号、眼球颜色、性别	众数、熵、列联相关、χ^2 检验
	序数	序数属性的值提供足够的信息确定对象的序（<，>）	矿石硬度、{好，较好，最好}、成绩、街道号码	中值、百分位、秩相关、游程检验、符号检验
数值的（定量的）	区间	对于区间属性，值之间的差是有意义的，即存在测量单位（+，−）	日历日期、摄氏或华氏温度	均值、标准差、皮尔逊相关、t 和 F 检验
	比率	对于比率变量，差和比率都是有意义的（*，/）	绝对温度、货币量、计数、年龄、质量、长度、电流	几何平均、调和平均、百分比变差

标称和序数属性统称**分类的**（categorical）或**定性的**（qualitative）属性。顾名思义，定性属性（如雇员 ID）不具有数的大部分性质。即便使用数（即整数）表示，也应当像对待符号一样对待它们。其余两种类型的属性，即区间和比率属性，统称**定量的**（quantitative）或**数值的**（numeric）属性。定量属性用数表示，并且具有数的大部分性质。注意：定量属性可以是整数值或连续值。

属性的类型也可以用不改变属性意义的变换来描述。实际上，心理学家 S. Smith Stevens 最先用**允许的变换**（permissible transformation）定义了表 2-2 所示的属性类型。例如，如果长度分别用米和英尺度量，其属性的意义并未改变。

对特定的属性类型有意义的统计操作是这样一些操作，当使用保持属性意义的变换对属性进行变换时，它们产生的结果相同。例如，用米和英尺为单位进行度量时，同一组对象的平均长度数值是不同的，但是两个平均值都代表相同的长度。表 2-3 给出表 2-2 中四种属性类型的允许的（保持意义的）变换。

表 2-3 定义属性层次的变换

属性类型		变换	注释
分类的（定性的）	标称	任何一对一变换，例如值的一个排列	如果所有雇员的 ID 号都重新赋值，不会出现任何不同
	序数	值的保序变换，即 新值=f(旧值)， 其中 f 是单调函数	包括好、较好、最好的属性可以完全等价地用值{1, 2, 3}或用{0.5, 1, 10}表示
数值的（定量的）	区间	新值 $=a*$旧值$+b$， 其中 a、b 是常数	华氏和摄氏温度的零度的位置不同，1 度的大小（即单位长度）也不同
	比率	新值$=a*$旧值	长度可以用米或英尺度量

例 2.5 温度标度 温度可以很好地解释前面介绍的一些概念。首先，温度可以是区间属性或比率属性，这取决于其测量标度。当温度用绝对标度测量时，从物理意义上讲，2° 的温度是 1° 的两倍；当温度用华氏或摄氏标度测量时则并非如此，因为这时 1° 温度与 2° 温度相差并不太多。问题是从物理意义上讲，华氏和摄氏标度的零点是硬性规定的，因此，华氏或摄氏温度的比率并无物理意义。 □

4. 用值的个数描述属性

区分属性的一种独立方法是根据属性可能取值的个数来判断。

- **离散的**（discrete） 离散属性具有有限个值或无限可数个值。这样的属性可以是分类的，如邮政编码或ID号，也可以是数值的，如计数。通常，离散属性用整数变量表示。**二元属性**（binary attribute）是离散属性的一种特殊情况，并只接受两个值，如真/假、是/否、男/女或0/1。通常，二元属性用布尔变量表示，或者用只取两个值0或1的整型变量表示。
- **连续的**（continuous） 连续属性是取实数值的属性。如温度、高度或重量等属性。通常，连续属性用浮点变量表示。实践中，实数值只能用有限的精度测量和表示。

从理论上讲，任何测量标度类型（标称的、序数的、区间的和比率的）都可以与基于属性值个数的任意类型（二元的、离散的和连续的）组合。然而，有些组合并不常出现，或者没有什么意义。例如，很难想象一个实际数据集包含连续的二元属性。通常，标称和序数属性是二元的或离散的，而区间和比率属性是连续的。然而，**计数属性**（count attribute）是离散的，也是比率属性。

5. 非对称的属性

对于非对称的属性（asymmetric attribute），出现非零属性值才是重要的。考虑这样一个数据集，其中每个对象是一个学生，而每个属性记录学生是否选修大学的某个课程。对于某个学生，如果他选修了对应于某属性的课程，该属性取值1，否则取值0。由于学生只选修所有可选课程中的很小一部分，这种数据集的大部分值为0。因此，关注非零值将更有意义、更有效。否则，如果在学生们不选修的课程上作比较，则大部分学生都非常相似。只有非零值才重要的二元属性是**非对称的二元属性**。这类属性对于关联分析特别重要。关联分析在第6章讨论。也可能有离散的或连续的非对称特征。例如，如果记录每门课程的学分，则结果数据集将包含**非对称的离散属性**或**连续属性**。

2.1.2　数据集的类型

数据集的类型有多种，并且随着数据挖掘的发展与成熟，还会有更多类型的数据集将用于分析。本节我们介绍一些很常见的类型。为方便起见，我们将数据集类型分成三组：记录数据、基于图形的数据和有序的数据。这些分类不能涵盖所有的可能性，肯定还存在其他的分组。

1. 数据集的一般特性

在提供特定类型数据集的细节之前，我们先讨论适用于许多数据集的三个特性，它们对数据挖掘技术具有重要影响，它们是维度、稀疏性和分辨率。

维度（dimensionality） 数据集的维度是数据集中的对象具有的属性数目。低维度数据往往与中、高维度数据有质的不同。确实，分析高维数据有时会陷入所谓**维灾难**（curse of dimensionality）。正因为如此，数据预处理的一个重要动机就是减少维度，称为**维归约**（dimensionality reduction）。这些问题在本章的后面会更深入地讨论。

稀疏性（sparsity） 有些数据集，如具有非对称特征的数据集，一个对象的大部分属性上的值都为0；在许多情况下，非零项还不到1%。实际上，稀疏性是一个优点，因为只有非零值才需要存储和处理。这将节省大量的计算时间和存储空间。此外，有些数据挖掘算法仅适合处理稀疏数据。

分辨率（resolution） 常常可以在不同的分辨率下得到数据，并且在不同的分辨率下数据的性质也不同。例如，在几米的分辨率下，地球表面看上去很不平坦，但在数十公里的分辨率下却

相对平坦。数据的模式也依赖于分辨率。如果分辨率太高，模式可能看不出，或者掩埋在噪声中；如果分辨率太低，模式可能不出现。例如，几小时记录一下气压变化可以反映出风暴等天气系统的移动；而在月的标度下，这些现象就检测不到。

2. 记录数据

许多数据挖掘任务都假定数据集是记录（数据对象）的汇集，每个记录包含固定的数据字段（属性）集。见图 2-2a。对于记录数据的大部分基本形式，记录之间或数据字段之间没有明显的联系，并且每个记录（对象）具有相同的属性集。记录数据通常存放在平展文件或关系数据库中。关系数据库当然不仅仅是记录的汇集，它还包含更多的信息，但是数据挖掘一般并不使用关系数据库的这些信息。更确切地说，数据库是查找记录的方便场所。下面介绍不同类型的记录数据，并用图 2-2 加以说明。

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

(a) 记录数据

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

(b) 事务数据

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

(c) 数据矩阵

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

(d) 文档-词矩阵

图 2-2 记录数据的不同变体

事务数据或购物篮数据 事务数据（transaction data）是一种特殊类型的记录数据，其中每个记录（事务）涉及一系列的项。考虑一个杂货店。顾客一次购物所购买的商品的集合就构成一个事务，而购买的商品是项。这种类型的数据称作**购物篮数据**（market basket data），因为记录中的项是顾客“购物篮”中的商品。事务数据是项的集合的集族，但是也能将它视为记录的集合，其中记录的字段是非对称的属性。这些属性常常是二元的，指出商品是否已买。更一般地，这些属性还可以是离散的或连续的，例如表示购买的商品数量或购买商品的花费。图 2-2b 展示了一个事务数据集，每一行代表一位顾客在特定时间购买的商品。

数据矩阵 如果一个数据集族中的所有数据对象都具有相同的数值属性集，则数据对象可以看作多维空间中的点（向量），其中每个维代表对象的一个不同属性。这样的数据对象集可以用一个 $m \times n$ 的矩阵表示，其中 m 行，一个对象一行；n 列，一个属性一列。（也可以将数据对象

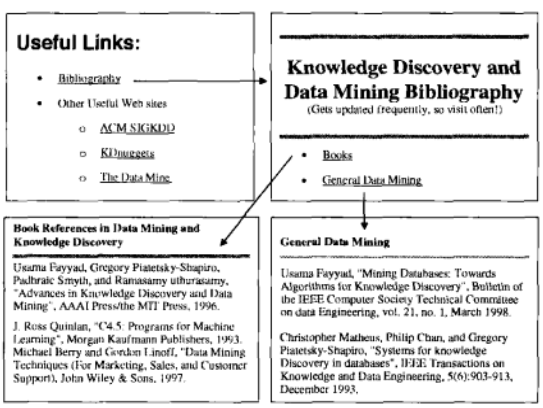
用列表示，属性用行表示。）这种矩阵称作**数据矩阵**（data matrix）或**模式矩阵**（pattern matrix）。数据矩阵是记录数据的变体，但是，由于它由数值属性组成，可以使用标准的矩阵操作对数据进行变换和处理，因此，对于大部分统计数据，数据矩阵是一种标准的数据格式。图 2-2c 示出一个样本数据矩阵。

稀疏数据矩阵　稀疏数据矩阵是数据矩阵的一种特殊情况，其中属性的类型相同并且是非对称的，即只有非零值才是重要的。事务数据是仅含 0-1 元素的稀疏数据矩阵的例子。另一个常见的例子是文档数据。特别地，如果忽略文档中词（术语）的次序，则文档可以用词向量表示，其中每个词是向量的一个分量（属性），而每个分量的值是对应词在文档中出现的次数。文档集合的这种表示通常称作**文档-词矩阵**（document-term matrix）。图 2-2d 显示了一个文档-词矩阵。文档是该矩阵的行，而词是矩阵的列。实践应用时，仅存放稀疏数据矩阵的非零项。

3. 基于图形的数据

有时，图形可以方便而有效地表示数据。我们考虑两种特殊情况：(1) 图形捕获数据对象之间的联系，(2) 数据对象本身用图形表示。

带有对象之间联系的数据　对象之间的联系常常携带重要信息。在这种情况下，数据常常用图形表示。一般把数据对象映射到图的结点，而对象之间的联系用对象之间的链和诸如方向、权值等链性质表示。考虑万维网上的网页，页面上包含文本和指向其他页面的链接。为了处理搜索查询，Web 搜索引擎收集并处理网页，提取它们的内容。然而，众所周知，指向或出自每个页面的链接包含了大量该页面与查询相关程度的信息，因而必须考虑。图 2-3a 显示了相互链接的网页集。



(a) 链接的网页　　(b) 苯分子

图 2-3　不同的图形数据

具有图形对象的数据　如果对象具有结构，即对象包含具有联系的子对象，则这样的对象常常用图形表示。例如，化合物的结构可以用图形表示，其中结点是原子，结点之间的链是化学键。图 2-3b 给出化合物苯的分子结构示意图，包含碳原子（黑色）和氢原子（灰色）。图形表示可以确定何种子结构频繁地出现在化合物的集合中，并且查明这些子结构中是否有某种子结构与诸如熔点或生成热等特定的化学性质有关。子结构挖掘是数据挖掘中分析这类数据的一个分支，将在 7.5 节讨论。

4. 有序数据

对于某些数据类型，属性具有涉及时间或空间序的联系。下面介绍各种类型的有序数据，并显示在图 2-4 中。

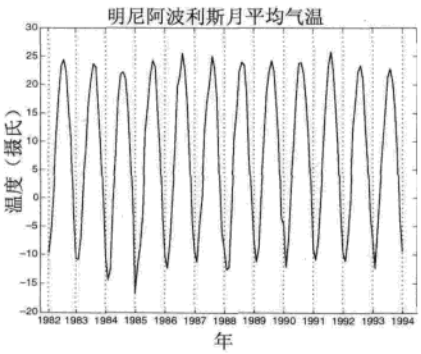
时间	顾客	购买的商品
*t*1	C1	A, B
*t*2	C3	A, C
*t*2	C1	C, D
*t*3	C2	A, D
*t*4	C2	E
*t*5	C1	A, E

顾客	购买时间与购买商品
C1	(*t*1: A,B) (*t*2:C,D) (*t*5:A,E)
C2	(*t*3: A, D) (*t*4: E)
C3	(*t*2: A, C)

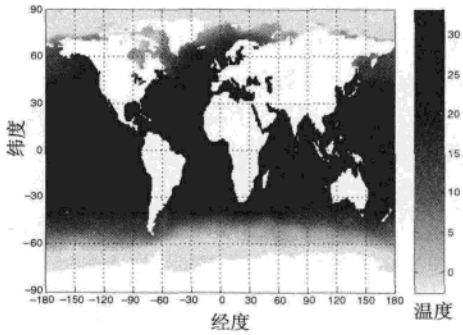
(a) 时序事务数据

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

(b) 基因组序列数据



(c) 温度时间序列



(d) 空间温度数据

图 2-4 不同的有序数据

时序数据 时序数据（sequential data）也称**时间数据**（temporal data），可以看作记录数据的扩充，其中每个记录包含一个与之相关联的时间。考虑存储事务发生时间的零售事务数据。时间信息可以帮助我们发现 “万圣节前夕糖果销售达到高峰” 之类的模式。时间也可以与每个属性相关联，例如，每个记录可以是一位顾客的购物历史，包含不同时间购买的商品列表。使用这些信息，就有可能发现 “购买 DVD 播放机的人趋向于在其后不久购买 DVD” 之类的模式。

图 2-4a 展示了一些时序事务数据。有 5 个不同的时间——*t*1、*t*2、*t*3、*t*4 和 *t*5；3 位不同的顾客——C1、C2 和 C3；5 种不同的商品——A、B、C、D 和 E。在图上面的表中，每行对应于一位顾客在特定的时间购买的商品。例如，在时间 *t*3，顾客 C2 购买了商品 A 和 D。下面的表显示相同的信息，但每行对应于一位顾客。每行包含涉及该顾客的所有事务信息，其中每个事务包含一些商品和购买这些商品的时间，例如，顾客 C3 在时间 *t*2 购买了商品 A 和 C。

序列数据 序列数据（sequence data）是一个数据集合，它是各个实体的序列，如词或字母的序列。除没有时间戳之外，它与时序数据非常相似，只是有序序列考虑项的位置。例如，动植物的遗传信息可以用称作基因的核苷酸的序列表示。与遗传序列数据有关的许多问题都涉及由核苷酸序列的相似性预测基因结构和功能的相似性。图 2-4b 显示用 4 种核苷酸表示的一段人类基

因码。所有 DNA 都可以用 A、T、G 和 C 四种核苷酸构造。

时间序列数据　时间序列数据（time series data）是一种特殊的时序数据，其中每个记录都是一个**时间序列**（time series），即一段时间以来的测量序列。例如，金融数据集可能包含各种股票每日价格的时间序列对象。再例如，考虑图 2-4c，该图显示明尼阿波利斯从 1982 年到 1994 年的月平均气温的时间序列。在分析时间数据时，重要的是要考虑**时间自相关**（temporal autocorrelation），即如果两个测量的时间很接近，则这些测量的值通常非常相似。

空间数据　有些对象除了其他类型的属性之外，还具有空间属性，如位置或区域。空间数据的一个例子是从不同的地理位置收集的气象数据（降水量、气温、气压）。空间数据的一个重要特点是**空间自相关性**（spatial autocorrelation），即物理上靠近的对象趋向于在其他方面也相似。这样，地球上相互靠近的两个点通常具有相近的气温和降水量。

空间数据的重要例子是科学和工程数据集，其数据取自二维或三维网格上规则或不规则分布的点上的测量或模型输出。例如，地球科学数据集记录在各种分辨率（如每度）下经纬度球面网格点（网格单元）上测量的温度和气压（见图 2-4d）。另一个例子，在瓦斯气流模拟中，可以针对模拟中的每个网格点记录流速和方向。

5. 处理非记录数据

大部分数据挖掘算法都是为记录数据或其变体（如事务数据和数据矩阵）设计的。通过从数据对象中提取特征，并使用这些特征创建对应于每个对象的记录，针对记录数据的技术也可以用于非记录数据。考虑前面介绍的化学结构数据。给定一个常见的子结构集合，每个化合物都可以用一个具有二元属性的记录表示，这些二元属性指出化合物是否包含特定的子结构。这样的表示实际上是事务数据集，其中事务是化合物，而项是子结构。

在某些情况下，容易用记录形式表示数据，但是这类表示并不能捕获数据中的所有信息。考虑这样的时间空间数据，它由空间网格每一点上的时间序列组成。通常，这种数据存放在数据矩阵中，其中每行代表一个位置，而每列代表一个特定的时间点。然而，这种表示并不能明确地表示属性之间存在的时间联系以及对象之间存在的空间联系。但并不是说这种表示不合适，而是说分析时必须考虑这些联系。例如，在使用数据挖掘技术时，假定属性之间在统计上是相互独立的并不是一个好主意。

2.2 数据质量

数据挖掘使用的数据常常是为其他用途收集的，或者在收集时未明确其目的。因此，数据挖掘常常不能“在数据源头控制质量”。相比之下，统计学的实验设计或调查往往其数据质量都达到了一定的要求。由于无法避免数据质量问题，因此数据挖掘着眼于两个方面：（1）数据质量问题的检测和纠正，（2）使用可以容忍低质量数据的算法。第一步的检测和纠正，通常称作**数据清理**（data cleaning）。

下面几节讨论数据质量。尽管也讨论某些与应用有关的问题，但是关注的焦点是测量和数据收集问题。

2.2.1 测量和数据收集问题

期望数据完美是不现实的。由于人的错误、测量设备的限制或数据收集过程的漏洞都可能导

致问题。数据的值乃至整个数据对象都可能会丢失。在有些情况下，可能有不真实的或重复的对象，即对应于单个“实际”对象出现了多个数据对象。例如，对于一个最近住过两个不同地方的人，可能有两个不同的记录。即使所有的数据都不缺，并且“看上去很好”，也可能存在不一致，如一个人身高 2m，但体重只有 2kg。

在下面几段，我们关注数据测量和收集方面的数据质量问题。我们先定义测量误差和数据收集错误，然后考虑涉及测量误差的各种问题：噪声、伪像、偏倚、精度和准确率。最后讨论可能同时涉及测量和数据收集的数据质量问题：离群点、遗漏和不一致的值、重复数据。

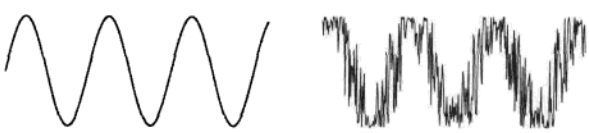
1. 测量误差和数据收集错误

术语**测量误差**（measurement error）是指测量过程中导致的问题。一个常见的问题是：在某种程度上，记录的值与实际值不同。对于连续属性，测量值与实际值的差称为**误差**（error）。术语**数据收集错误**（data collection error）是指诸如遗漏数据对象或属性值，或不当地包含了其他数据对象等错误。例如，一种特定种类动物研究可能包含了相关种类的其他动物，它们只是表面上与要研究的种类相似。测量误差和数据收集错误可能是系统的也可能是随机的。

我们只考虑一般的错误类型。在特定的领域，总有些类型的错误是常见的，并且常常有很好的技术来检测并纠正这些错误。例如，人工输入数据时键盘录入错误是常见的，因此许多数据输入程序具有检测技术，并且通过人工干预纠正这类错误。

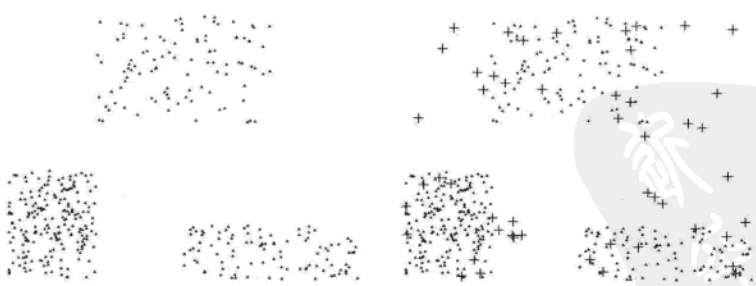
2. 噪声和伪像

噪声是测量误差的随机部分。这可能涉及值被扭曲或加入了谬误对象。图 2-5 显示被随机噪声干扰前后的时间序列。如果在时间序列上添加更多的噪声，形状将会消失。图 2-6 显示了三组添加一些噪声点（用“+”表示）前后的数据点集。注意，有些噪声点与非噪声点混在一起。



(a) 时间序列　　(b) 含噪声的时间序列

图 2-5　时间序列中的噪声



(a) 三组点　　(b) 含噪声点（+）

图 2-6　空间中的噪声

术语“噪声”通常用于包含时间或空间分量的数据。在这些情况下，常常可以使用信号或图像处理技术降低噪声，从而帮助发现可能“淹没在噪声中”的模式（信号）。尽管如此，完全消

除噪声通常是困难的，而许多数据挖掘工作都关注设计**鲁棒算法**（robust algorithm），即在噪声干扰下也能产生可以接受的结果。

数据错误可能是更确定性现象的结果，如一组照片在同一地方出现条纹。数据的这种确定性失真常称作**伪像**（artifact）。

3. 精度、偏倚和准确率

在统计学和实验科学中，测量过程和结果数据的质量用精度和偏倚度量。我们给出标准的定义，随后简略加以讨论。对于下面的定义，我们假定对相同的基本量进行重复测量，并使用测量值集合计算均值（平均值），作为实际值的估计。

定义 2.3　精度（precision）（同一个量的）重复测量值之间的接近程度。

定义 2.4　偏倚（bias）　测量值与被测量之间的系统的变差。

精度通常用值集合的标准差度量，而偏倚用值集合的均值与测出的已知值之间的差度量。只有那些通过外部手段能够得到测量值的对象，偏倚才是可确定的。假定我们有 1 克质量的标准实验室重量，并且想评估实验室的新天平的精度和偏倚。我们称重 5 次，得到下列值：{1.015, 0.990, 1.013, 1.001, 0.986}。这些值的均值是 1.001，因此偏倚是 0.001。用标准差度量，精度是 0.013。

通常使用更一般的术语**准确率**表示数据测量误差的程度。

定义 2.5　准确率（accuracy）　被测量的测量值与实际值之间的接近度。

准确率依赖于精度和偏倚，但是由于它是一个一般化的概念，因此没有用这两个量表达准确率的公式。

准确率的一个重要方面是**有效数字**（significant digit）的使用。其目标是仅使用数据精度所能确定的数字位数表示测量或计算结果。例如，对象的长度用最小刻度为毫米的米尺测量，则我们只能记录最接近毫米的长度数据，这种测量的精度为±0.5mm。我们不再详细地讨论有效数字，因为大部分读者应当在先前的课程中接触过，并且在理工科和统计学教材中讨论得相当深入。

诸如有效数字、精度、偏倚和准确率问题常常被忽视，但是对于数据挖掘、统计学和自然科学，它们都非常重要。通常，数据集并不包含数据精度信息，用于分析的程序返回的结果也没有这方面的信息。但是，缺乏对数据和结果准确率的理解，分析者将可能出现严重的数据分析错误。

4. 离群点

离群点（outlier）是在某种意义上具有不同于数据集中其他大部分数据对象的特征的数据对象，或是相对于该属性的典型值来说不寻常的属性值。我们也称其为**异常**（anomalous）对象或异常值。有许多定义离群点的方法，并且统计学和数据挖掘界已经提出了很多不同的定义。此外，区别噪声和离群点这两个概念是非常重要的。离群点可以是合法的数据对象或值。因此，不像噪声，离群点本身有时是人们感兴趣的对象。例如，欺诈和网络攻击检测中，目标就是从大量正常对象或事件中发现不正常的对象和事件。第 10 章更详细地讨论异常检测。

5. 遗漏值

一个对象遗漏一个或多个属性值的情况并不少见。有时可能会出现信息收集不全的情况，例如有的人拒绝透露年龄或体重。还有些情况下，某些属性并不能用于所有对象，例如表格常常有条件选择部分，仅当填表人以特定的方式回答前面的问题时，条件选择部分才需要填写，但为简

单起见存储了表格的所有字段。无论何种情况，在数据分析时都应当考虑遗漏值。

有许多处理遗漏值的策略（和这些策略的变种），每种策略可能适用于特定的情况。这些策略在下面列出，同时我们指出它们的优缺点。

删除数据对象或属性 一种简单而有效的策略是删除具有遗漏值的数据对象。然而，即使不完整的数据对象也包含一些有用的信息，并且，如果许多对象都有遗漏值，则很难甚至不可能进行可靠的分析。尽管如此，如果某个数据集只有少量的对象具有遗漏值，则忽略它们可能是合算的。一种与之相关的策略是删除具有遗漏值的属性。然而，做这件事要小心，因为被删除的属性可能对分析是至关重要的。

估计遗漏值 有时，遗漏值可以可靠地估计。例如，在考虑以大致平滑的方式变化的、具有少量但分散的遗漏值的时间序列时，遗漏值可以使用其他值来估计（插值）。另举一例，考虑一个具有许多相似数据点的数据集，与具有遗漏值的点邻近的点的属性值常常可以用来估计遗漏的值。如果属性是连续的，则可以使用最近邻的平均属性值；如果属性是分类的，则可以取最近邻中最常出现的属性值。为了更具体地解释，考虑地面站记录的降水量，对于未设地面站的区域，降水量可以使用邻近地面站的观测值估计。

在分析时忽略遗漏值 许多数据挖掘方法都可以修改，忽略遗漏值。例如，假定正在对数据对象聚类，需要计算各对数据对象间的相似性。如果某对的一个对象或两个对象都有某些属性有遗漏值，则可以仅使用没有遗漏值的属性来计算相似性。当然，这种相似性只是近似的，但是除非整个属性数目很少，或者遗漏值的数量很大，否则这种误差影响不大。同样地，许多分类方法都可以修改，便于处理遗漏值。

6. 不一致的值

数据可能包含不一致的值。比如地址字段列出了邮政编码和城市名，但是有的邮政编码区域并不包含在对应的城市中。可能是人工输入该信息时录颠倒了两个数字，或许是在手写体扫描时错读了一个数字。无论导致不一致值的原因是什么，重要的是能检测出来，并且如果可能的话，纠正这种错误。

有些不一致类型容易检测，例如人的身高不应当是负的。有些情况下，可能需要查阅外部信息源，例如当保险公司处理赔偿要求时，它将对照顾客数据库核对赔偿单上的姓名与地址。

检测到不一致后，有时可以对数据进行更正。产品代码可能有“校验”数字，或者可以通过一个备案的已知产品代码列表，复核产品代码；如果发现它不正确但接近一个已知代码，则纠正它。纠正不一致需要额外的或冗余的信息。

例 2.6 不一致的海洋表面温度 该例解释实际的时间序列数据中的不一致性。这些数据是在海洋的不同点测量的海洋表面温度（SST）。最早，人们利用船或浮标使用海洋测量方法收集SST 数据；而最近，开始使用卫星来收集这些数据。为了创建长期的数据集，需要使用这两种数据源。然而，由于数据来自不同的数据源，两部分数据存在微妙的不同。这种差异显示在图 2-7 中，该图显示了各年度之间 SST 值的相关性。如果某两个年度的 SST 值是正相关的，则对应于这两年的位置为白色，否则为黑色。（季节性的变化从数据中删除，否则所有的年都是高度相关的。）在数据汇集一起的地方（1983 年）有一个明显的变化。1958～1982 和 1983～1999 两组，每组内的年相互之间趋向于正相关，但与另一组的年负相关。这并不意味该数据不能用，但是分

析者应当考虑这种差异对数据挖掘分析的潜在影响。 □

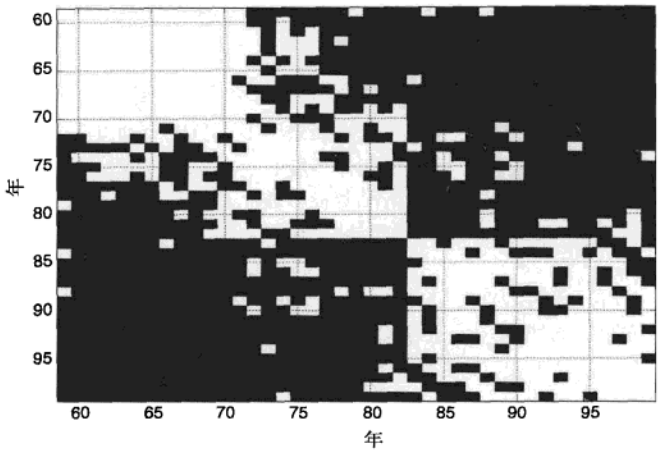


图 2-7　年对之间 SST 数据的相关性。白色区域表示正相关，黑色区域表示负相关

7. 重复数据

数据集可能包含重复或几乎重复的数据对象。许多人都收到过重复的邮件，因为他们以稍微不相同的名字多次出现在数据库中。为了检测并删除这种重复，必须处理两个主要问题。首先，如果两个对象实际代表同一个对象，则对应的属性值必然不同，必须解决这些不一致的值；其次，需要避免意外地将两个相似但并非重复的数据对象（如两个人具有相同姓名）合并在一起。术语**去重复**（deduplication）通常用来表示处理这些问题的过程。

在某些情况下，两个或多个对象在数据库的属性度量上是相同的，但是仍然代表不同的对象。这种重复是合法的。但是，如果某些算法设计中没有专门考虑这些属性可能相同的对象，就还是可能导致问题。本章习题 13 就是这么一个例子。

2.2.2　关于应用的问题

数据质量问题也可以从应用角度考虑，表达为“数据是高质量的，如果它适合预期的应用”。特别是对工商业界，数据质量的这种提议非常有用。类似的观点也出现在统计学和实验科学，那里强调精心设计实验来收集与特定假设相关的数据。与测量和数据收集一样，许多数据质量问题与特定的应用和领域有关。我们这里仍然只考虑一些一般性问题。

时效性　有些数据收集后就开始老化。比如说，如果数据提供正在发生的现象或过程的快照，如顾客的购买行为或 Web 浏览模式，则快照只代表有限时间内的真实情况。如果数据已经过时，则基于它的模型和模式也已经过时。

相关性　可用的数据必须包含应用所需要的信息。考虑构造一个模型，预测交通事故发生率。如果忽略了驾驶员的年龄和性别信息，那么除非这些信息可以间接地通过其他属性得到，否则模型的精度可能是有限的。

确保数据集中的对象相关不太容易。一个常见问题是**抽样偏倚**（sampling bias），指样本包含

的不同类型的对象与它们在总体中的出现情况不成比例。例如调查数据只反映对调查做出响应的那些人的意见。（抽样的其他问题将在 2.3.2 节进一步讨论。）由于数据分析的结果只反映现有的数据，抽样偏倚通常导致不正确的分析。

关于数据的知识 理想情况下，数据集附有描述数据的文档。文档的质量好坏决定它是支持还是干扰其后的分析。例如，如果文档标明若干属性是强相关的，则说明这些属性可能提供了高度冗余的信息，我们可以决定只保留一个。（考虑销售税和销售价格。）然而，如果文档很糟糕，例如，没有告诉我们某特定字段上的遗漏值用-9999 指示，则我们的数据分析就可能出问题。其他应该说明的重要特性是数据精度、特征的类型（标称的、序数的、区间的、比率的）、测量的刻度（如长度用米还是英尺）和数据的来源。

2.3 数据预处理

本节，我们讨论应当采用哪些预处理步骤，让数据更加适合挖掘。数据预处理是一个广泛的领域，包含大量以复杂的方式相关联的不同策略和技术。我们将讨论一些最重要的思想和方法，并试图指出它们之间的相互联系。具体地说，我们将讨论如下主题。

- 聚集。
- 抽样。
- 维归约。
- 特征子集选择。
- 特征创建。
- 离散化和二元化。
- 变量变换。

粗略地说，这些项目分为两类，即选择分析所需要的数据对象和属性以及创建/改变属性。这两种情况的目标都是改善数据挖掘分析工作，减少时间，降低成本和提高质量。细节参见以下几节。

术语注记：下面，我们有时将根据习惯用法，使用特征（feature）或变量（variable）指代属性（attribute）。

2.3.1 聚集

有时，“少就是多”，而聚集就是如此。**聚集**（aggregation）将两个或多个对象合并成单个对象。考虑一个由事务（数据对象）组成的数据集，它记录一年中不同日期在各地（明尼阿波利斯、芝加哥、巴黎……）商店的商品日销售情况，见表 2-4。对该数据集的事务进行聚集的一种方法，是用一个商店事务替换该商店的所有事务。这把每天出现在一个商店的成百上千个事务记录归约成单个日事务，而数据对象的个数减少为商店的个数。

表 2-4 包含顾客购买信息的数据集

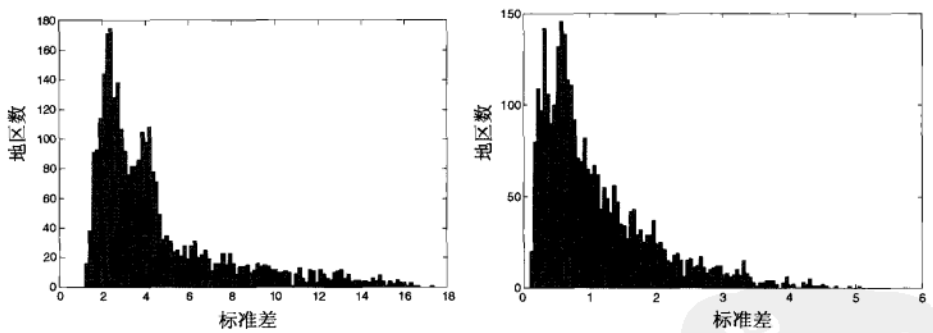
事务 ID	商品	商店位置	日期	价格	…
⋮	⋮	⋮	⋮	⋮	
101123	Watch	Chicago	09/06/04	$25.99	…
101123	Battery	Chicago	09/06/04	$5.99	…
101124	Shoes	Minneapolis	09/06/04	$75.00	…
⋮	⋮	⋮	⋮	⋮	

这里显而易见的问题是如何创建聚集事务，即在创建代表单个商店或日期的聚集事务时，如何合并所有记录的每个属性的值。定量属性（如价格）通常通过求和或求平均值进行聚集。定性属性（如商品）可以忽略或汇总成在一个商店销售的所有商品的集合。

表 2-4 中的数据也可以看作多维数组，其中每个属性是一个维。从这个角度，聚集是删除属性（如商品类型）的过程，或者是压缩特定属性不同值个数的过程，如将日期的可能值从 365 天压缩到 12 个月。这种类型的聚集通常用于 OLAP（Online Analytical Processing，联机分析处理）。OLAP 在第 3 章进一步讨论。

聚集的动机有多种。首先，数据归约导致的较小数据集需要较少的内存和处理时间，因此可以使用开销更大的数据挖掘算法。其次，通过高层而不是低层数据视图，聚集起到了范围或标度转换的作用。在前面的例子中，在商店位置和月份上的聚集给出数据按月、按商店，而不是按天、按商品的视图。最后，对象或属性群的行为通常比单个对象或属性的行为更加稳定。这反映了统计学事实：相对于被聚集的单个对象，诸如平均值、总数等聚集量具有较小的变异性。对于总数，实际变差大于单个对象的（平均）变差，但是变差的百分比较小；而对于均值，实际变差小于单个对象的（平均）变差。聚集的缺点是可能丢失有趣的细节。在商店的例子中，按月的聚集就丢失了星期几具有最高销售额的信息。

例 2.7　澳大利亚降水量　该例基于澳大利亚从 1982 年到 1993 年的降水量。我们把澳大利亚国土按经纬度 0.5° 乘 0.5° 大小分成 3030 个网格。图 2-8a 的直方图显示这些网格单元上的平均月降水量的标准差，而图 2-8b 的直方图显示相同位置的平均年降水量的标准差。可见，平均年降水量比平均月降水量的变异性小。所有降水量测量（以及它们的标准差）都以厘米（cm）为单位。　□



(a) 平均月降水量标准差的直方图　　(b) 平均年降水量标准差的直方图

图 2-8　澳大利亚从 1982 年到 1993 年月和年降水量标准差的直方图

2.3.2　抽样

抽样是一种选择数据对象子集进行分析的常用方法。在统计学中，抽样长期用于数据的事先调查和最终的数据分析。在数据挖掘中，抽样也非常有用。然而，在统计学和数据挖掘中，抽样的动机并不相同。统计学使用抽样是因为得到感兴趣的整个数据集的费用太高、太费时间，而数据挖掘使用抽样是因为处理所有的数据的费用太高、太费时间。在某些情况下，使用抽样的算法

可以压缩数据量，以便可以使用更好但开销较大的数据挖掘算法。

有效抽样的主要原理如下：如果样本是有代表性的，则使用样本与使用整个数据集的效果几乎一样。**而样本是有代表性的**，前提是它近似地具有与原数据集相同的（感兴趣的）性质。如果数据对象的均值（平均值）是感兴趣的性质，而样本具有近似于原数据集的均值，则样本就是有代表性的。由于抽样是一个统计过程，特定样本的代表性是变化的，因此我们所能做的最好的抽样方案就是选择一个确保以很高的概率得到有代表性的样本。如下所述，这涉及选择适当的样本容量和抽样技术。

1. 抽样方法

有许多抽样技术，但是这里只介绍少数最基本的抽样技术和它们的变形。最简单的抽样是**简单随机抽样**（simple random sampling）。对于这种抽样，选取任何特定项的概率相等。随机抽样有两种变形（其他抽样技术也一样）：(1) **无放回抽样**——每个选中项立即从构成**总体**的所有对象集中删除；(2) **有放回抽样**——对象被选中时不从总体中删除。在有放回抽样中，相同的对象可能被多次抽出。当样本与数据集相比相对较小时，两种方法产生的样本差别不大。但是，对于分析，有放回抽样较为简单，因为在抽样过程中，每个对象被选中的概率保持不变。

当总体由不同类型的对象组成，每种类型的对象数量差别很大时，简单随机抽样不能充分地代表不太频繁出现的对象类型。当分析需要所有类型的代表时，这可能出现问题。例如，当为稀有类构建分类模型时，样本中适当地提供稀有类是至关重要的，因此需要提供具有不同频率的感兴趣的项的抽样方案。**分层抽样**（stratified sampling）就是这样的方法，它从预先指定的组开始抽样。在最简单的情况下，尽管每组的大小不同，但是从每组抽取的对象个数相同。另一种变形是从每一组抽取的对象数量正比于该组的大小。

例 2.8 抽样与信息损失 一旦选定抽样技术，就需要选择样本容量。较大的样本容量增大了样本具有代表性的概率，但也抵消了抽样带来的许多好处。反过来，使用较小容量的样本，可能丢失模式，或检测出错误的模式。图 2-9a 显示包含 8 000 个二维点的数据集，而图 2-9b 和图 2-9c 显示从该数据集抽取的容量分别为 2 000 和 500 的样本。该数据集的大部分结构都出现在 2 000 个点的样本中，但是许多结构在 500 个点的样本中丢失了。 □

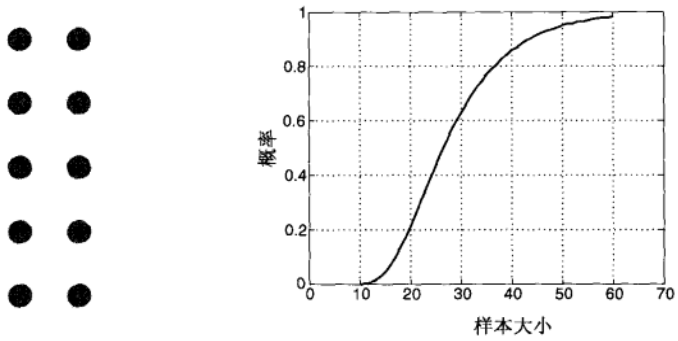
(a) 8 000 个点　(b) 2 000 个点　(c) 500 个点

图 2-9 抽样丢失结构的例子

例 2.9 确定适当的样本容量 为了说明确定合适的样本容量需要系统的方法，考虑下面的任务。

给定一个数据集，它包含少量容量大致相等的组。从每组至少找出一个代表点。假定每个组内的对象高度相似，但是不同组中的对象不太相似。还假定组的个数不多（例如，10个组）。图2-10a显示了一个理想簇（组）的集合，这些点可能从中抽取。

使用抽样可以有效地解决该问题。一种方法是取数据点的一个小样本，逐对计算点之间的相似性，然后形成高度相似的点组。从这些组每组取一个点，则可以得到具有代表性的点的集合。然而，按照该方法，我们需要确定样本的容量，它以很高的概率确保得到期望的结果，即从每个簇至少找出一个代表点。图2-10b显示随着样本容量从10变化到60时，从10个组的每一个得到一个对象的概率。有趣的是，使用容量为20的样本，只有很小的机会（20%）得到包含所有10个簇的样本。即便使用容量为30的样本，得到不包含所有10个簇中对象的样本的几率也很高（几乎40%）。该问题将在第8章习题4讨论聚类中进一步考察。 □



(a) 点的10个组　　(b) 样本包含所有10个组中点的概率

图2-10　从10个组找出具有代表性的点

2. 渐进抽样

合适的样本容量可能很难确定，因此有时需要使用**自适应**（adaptive）或**渐进抽样**（progressive sampling）方法。这些方法从一个小样本开始，然后增加样本容量直至得到足够容量的样本。尽管这种技术不需要在开始就确定正确的样本容量，但是需要评估样本的方法，确定它是否足够大。

例如，假定使用渐进抽样来学习一个预测模型。尽管预测模型的准确率随样本容量增加，但是在某一点准确率的增加趋于稳定。我们希望在稳定点停止增加样本容量。通过掌握模型准确率随样本逐渐增大的变化情况，并通过选取接近于当前容量的其他样本，我们可以估计出与稳定点的接近程度，从而停止抽样。

2.3.3　维归约

数据集可能包含大量特征。考虑一个文档的集合，其中每个文档是一个向量，其分量是文档中出现的每个词的频率。在这种情况下，通常有成千上万的属性（分量），每个代表词汇表中的一个词。再看一个例子，考虑包含过去30年各种股票日收盘价的时间序列数据集。在这种情况下，属性是特定天的价格，也数以千计。

维归约有多方面的好处。关键的好处是，如果维度（数据属性的个数）较低，许多数据挖掘算法的效果就会更好。这一部分是因为维归约可以删除不相关的特征并降低噪声，一部分是因为

维灾难。（维灾难在下面解释。）另一个好处是维归约可以使模型更容易理解，因为模型可能只涉及较少的属性。此外，维归约也可以更容易让数据可视化。即使维归约没有将数据归约到二维或三维，数据也可以通过观察属性对或三元组属性达到可视化，并且这种组合的数目也会大大减少。最后，使用维归约降低了数据挖掘算法的时间和内存需求。

术语“维归约”通常用于这样的技术：通过创建新属性，将一些旧属性合并在一起来降低数据集的维度。通过选择旧属性的子集得到新属性，这种维归约称为特征子集选择或特征选择。特征选择将在 2.3.4 节讨论。

下面我们简单介绍两个重要的主题：维灾难和基于线性代数方法（如主成分分析）的维归约技术。

1. 维灾难

维灾难是指这样的现象：随着数据维度的增加，许多数据分析变得非常困难。特别是随着维度增加，数据在它所占据的空间中越来越稀疏。对于分类，这可能意味没有足够的数据对象来创建模型，将所有可能的对象可靠地指派到一个类。对于聚类，点之间的密度和距离的定义（对聚类是至关重要的）失去了意义。（在 9.1.2 节、9.4.5 节和 9.4.7 节进一步讨论。）结果是，对于高维数据，许多分类和聚类算法（以及其他数据分析算法）都麻烦缠身——分类准确率降低，聚类质量下降。

2. 维归约的线性代数技术

维归约的一些最常用的方法是使用线性代数技术，将数据由高维空间投影到低维空间，特别是对于连续数据。**主成分分析**（Principal Components Analysis，PCA）是一种用于连续属性的线性代数技术，它找出新的属性（主成分），这些属性是原属性的线性组合，是相互**正交的**（orthogonal），并且捕获了数据的最大变差。例如，前两个主成分是两个正交属性，是原属性的线性组合，尽可能多地捕获了数据的变差。**奇异值分解**（Singular Value Decomposition，SVD）是一种线性代数技术，它与 PCA 有关，并且也用于维归约。

2.3.4 特征子集选择

降低维度的另一种方法是仅使用特征的一个子集。尽管看起来这种方法可能丢失信息，但是在存在冗余或不相关的特征的时候，情况并非如此。**冗余特征**重复了包含在一个或多个其他属性中的许多或所有信息。例如，一种产品的购买价格和所支付的销售税额包含许多相同的信息。**不相关特征**包含对于手头的数据挖掘任务几乎完全没用的信息，例如学生的 ID 号码对于预测学生的总平均成绩是不相关的。冗余和不相关的特征可能降低分类的准确率，影响所发现的聚类的质量。

尽管使用常识或领域知识可以立即消除一些不相关的和冗余的属性，但是选择最佳的特征子集通常需要系统的方法。特征选择的理想方法是：将所有可能的特征子集作为感兴趣的数据挖掘算法的输入，然后选取产生最好结果的子集。这种方法的优点是反映了最终使用的数据挖掘算法的目的和偏爱。然而，由于涉及 n 个属性的子集多达 2^n 个，这种方法在大部分情况下行不通，因此需要其他策略。有三种标准的特征选择方法：嵌入、过滤和包装。

嵌入方法（embedded approach） 特征选择作为数据挖掘算法的一部分是理所当然的。特别是在数据挖掘算法运行期间，算法本身决定使用哪些属性和忽略哪些属性。构造决策树分类器的算法（在第 4 章讨论）通常以这种方式运行。

过滤方法（filter approach）　使用某种独立于数据挖掘任务的方法，在数据挖掘算法运行前进行特征选择，例如我们可以选择属性的集合，它的属性对之间的相关度尽可能低。

包装方法（wrapper approach）　这些方法将目标数据挖掘算法作为黑盒，使用类似于前面介绍的理想算法，但通常并不枚举所有可能的子集来找出最佳属性子集。

由于嵌入方法与具体的算法有关，这里我们只进一步讨论过滤和包装方法。

1. 特征子集选择体系结构

可以将过滤和包装方法放到一个共同的体系结构中。特征选择过程可以看作由四部分组成：子集评估度量、控制新的特征子集产生的搜索策略、停止搜索判断和验证过程。过滤方法和包装方法的唯一不同是它们使用了不同的特征子集评估方法。对于包装方法，子集评估使用目标数据挖掘算法；对于过滤方法，子集评估技术不同于目标数据挖掘算法。下面的讨论提供了该方法的一些细节，汇总在图 2-11 中。

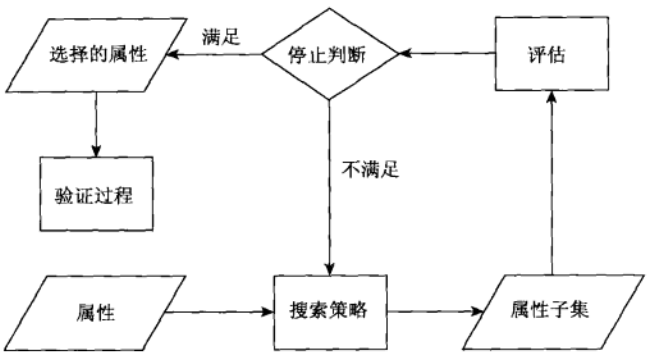


图 2-11　特征子集选择过程流程图

从概念上讲，特征子集选择是搜索所有可能的特征子集的过程。可以使用许多不同类型的搜索策略，但是搜索策略的计算花费应当较低，并且应当找到最优或近似最优的特征子集。通常不可能同时满足这两个要求，因此需要折中权衡。

搜索的一个不可缺少的组成部分是评估步骤，根据已经考虑的子集评价当前的特征子集。这需要一种评估度量，针对诸如分类或聚类等数据挖掘任务，确定属性特征子集的质量。对于过滤方法，这种度量试图预测实际的数据挖掘算法在给定的属性集上执行的效果如何；对于包装方法，评估包括实际运行目标数据挖掘应用，子集评估函数就是通常用于度量数据挖掘结果的评判标准。

因为子集的数量可能很大，考察所有的子集可能不现实，因此需要某种停止搜索判断。其策略通常基于如下一个或多个条件：迭代次数，子集评估的度量值是否最优或超过给定的阈值，一个特定大小的子集是否已经得到，大小和评估标准是否同时达到，使用搜索策略得到的选择是否可以实现改进。

最后，一旦选定特征子集，就要验证目标数据挖掘算法在选定子集上的结果。一种直截了当的评估方法是用全部特征的集合运行算法，并将全部结果与使用该特征子集得到的结果进行比较。如果顺利的话，特征子集产生的结果将比使用所有特征产生的结果更好，或者至少几乎一样

好。另一个验证方法是使用一些不同的特征选择算法得到特征子集，然后比较数据挖掘算法在每个子集上的运行结果。

2. 特征加权

特征加权是另一种保留或删除特征的办法。特征越重要，所赋予的权值越大，而不太重要的特征赋予较小的权值。有时，这些权值可以根据特征的相对重要性的领域知识确定，也可以自动确定。例如，有些分类方法，如支持向量机（第 5 章），产生分类模型，其中每个特征都赋予一个权值。具有较大权值的特征在模型中所起的作用更加重要。在计算余弦相似度时进行的对象规范化（2.4.5 节）也可以看作一类特征加权。

2.3.5 特征创建

常常可以由原来的属性创建新的属性集，更有效地捕获数据集中的重要信息。此外，新属性的数目可能比原属性少，使得我们可以获得前面介绍的维归约带来的所有好处。下面介绍三种创建新属性的相关方法：特征提取、映射数据到新的空间和特征构造。

1. 特征提取

由原始数据创建新的特征集称作**特征提取**（feature extraction）。考虑照片的集合，按照照片是否包含人脸分类。原始数据是像素的集合，因此对于许多分类算法都不适合。然而，如果对数据进行处理，提供一些较高层次的特征，诸如与人脸高度相关的某些类型的边和区域等，则会有更多的分类技术可以用于该问题。

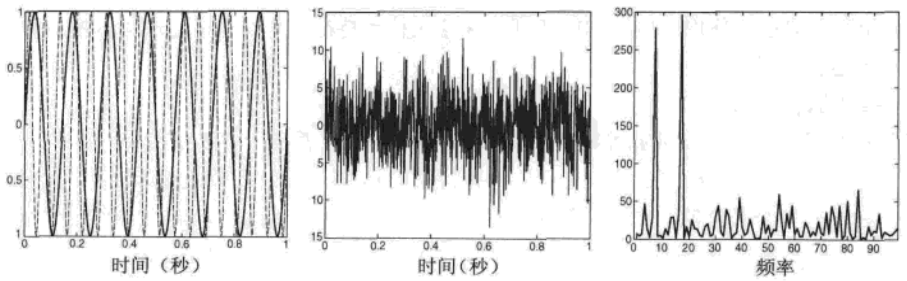
可是，最常使用的特征提取技术都是高度针对具体领域的。对于特定的领域，如图像处理，在过去一段时间已经开发了各种特征和提取特征的技术，但是这些技术在其他领域的应用却是有限的。因而，一旦数据挖掘用于一个相对较新的领域，一个关键任务就是开发新的特征和特征提取方法。

2. 映射数据到新的空间

使用一种完全不同的视角挖掘数据可能揭示出重要和有趣的特征。例如，考虑时间序列数据，它们常常包含周期模式。如果只有单个周期模式，并且噪声不多，则容易检测到该模式；另一方面，如果有大量周期模式，并且存在大量噪声，则很难检测这些模式。尽管如此，通过对该时间序列实施**傅里叶变换**（Fourier transform），将它转换成频率信息明显的表示，就能检测到这些模式。在下面的例子中，不必知道傅里叶变换的细节，只需要知道对于时间序列，傅里叶变换产生其属性与频率有关的新数据对象就足够了。

例 2.10 傅里叶分析 图 2-12b 中的时间序列是其他三个时间序列的和，其中两个显示在图 2-12a 中，其频率分别是每秒 7 个和 17 个周期，第三个时间序列是随机噪声。图 2-12c 显示功率频谱。在对原时间序列施加傅里叶变换后，可以计算功率频谱。（非正式地看，功率频谱正比于每个频率属性的平方。）尽管有噪声，图中有两个尖峰，对应于两个原来的、无噪声的时间序列的周期。再说一遍，本例的要点是：更好的特征可以揭示数据的重要性质。 □

也可以采用许多其他类型的变换。除傅里叶变换外，对于时间序列和其他类型的数据，经证实**小波变换**（wavelet transform）也是非常有用的。



(a) 两个时间序列　(b) 噪声时间序列　(c) 功率频谱

图 2-12　傅里叶变换应用：识别时间序列数据中的基本频率

3. 特征构造

有时，原始数据集的特征具有必要的信息，但其形式不适合数据挖掘算法。在这种情况下，一个或多个由原特征构造的新特征可能比原特征更有用。

例 2.11　密度　为了解释这一点，考虑一个包含人工制品信息的历史数据集。该数据集包含每个人工制品的体积和质量，以及其他信息。为简单起见，假定这些人工制品使用少量材料（木材、陶土、铜、黄金）制造，并且我们希望根据制造材料对它们分类。在此情况下，由质量和体积特征构造的密度特征（即密度 = 质量/体积）可以很直接地产生准确的分类。尽管有一些人试图通过考察已有特征的简单的数学组合来自动地进行特征构造，但是最常见的方法还是使用专家的意见构造特征。　□

2.3.6　离散化和二元化

有些数据挖掘算法，特别是某些分类算法，要求数据是分类属性形式。发现关联模式的算法要求数据是二元属性形式。这样，常常需要将连续属性变换成分类属性（**离散化**，discretization），并且连续和离散属性可能都需要变换成一个或多个二元属性（**二元化**，binarization）。此外，如果一个分类属性具有大量不同值（类别），或者某些值出现不频繁，则对于某些数据挖掘任务，通过合并某些值减少类别的数目可能是有益的。

与特征选择一样，最佳的离散化和二元化方法是“对于用来分析数据的数据挖掘算法，产生最好结果”的方法。直接使用这种判别标准通常是不实际的。因此，离散化和二元化一般要满足这样一种判别标准，它与所考虑的数据挖掘任务的性能好坏直接相关。

1. 二元化

一种分类属性二元化的简单技术如下：如果有 m 个分类值，则将每个原始值唯一地赋予区间$[0, m-1]$中的一个整数。如果属性是有序的，则赋值必须保持序关系。（注意，即使属性原来就用整数表示，但如果这些整数不在区间$[0, m-1]$中，则该过程也是必需的。）然后，将这 m 个整数的每一个都变换成一个二进制数。由于需要 $n = \lceil \log_2 m \rceil$个二进位表示这些整数，因此要使用 n 个二元属性表示这些二进制数。例如，一个具有 5 个值{*awful*, *poor*, *OK*, *good*, *great*}的分类变量需要三个二元变量 x_1、x_2、x_3。转换见表 2-5。

表 2-5 一个分类属性到三个二元属性的变换

分类值	整数值	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

这样的变换可能导致复杂化，如无意之中建立了转换后的属性之间的联系。例如，在表 2-5 中，属性 x_2 和 x_3 是相关的，因为 *good* 值使用这两个属性表示。此外，关联分析需要非对称的二元属性，其中只有属性的出现（值为 1）才是重要的。因此，对于关联问题，需要为每一个分类值引入一个二元属性，如表 2-6 所示。如果结果属性的个数太多，则可以在二元化之前使用下面介绍的技术减少分类值的个数。

表 2-6 一个分类属性到五个非对称二元属性的转换

分类值	整数值	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

同样，对于关联问题，可能需要用两个非对称的二元属性替换单个二元属性。考虑记录人的性别（男、女）的二元属性，对于传统的关联规则算法，该信息需要转换成两个非对称的二元属性，其中一个仅当是男性时为 1，而另一个仅当是女性时为 1。（对于非对称的二元属性，由于其提供一个二进制位信息需要占用存储器的两个二进制位，因而在信息的表示上不太有效。）

2. 连续属性离散化

通常，离散化应用于在分类或关联分析中使用到的属性上。一般来说，离散化的效果取决于所使用的算法，以及用到的其他属性。然而，属性离散化通常单独考虑。

连续属性变换成分类属性涉及两个子任务：决定需要多少个分类值，以及确定如何将连续属性值映射到这些分类值。在第一步中，将连续属性值排序后，通过指定 $n-1$ 个**分割点**（split point）把它们分成 n 个区间。在颇为平凡的第二步中，将一个区间中的所有值映射到相同的分类值。因此，离散化问题就是决定选择多少个分割点和确定分割点位置的问题。结果可以用区间集合$\{(x_0, x_1], (x_1, x_2], \cdots, (x_{n-1}, x_n)\}$表示，其中 x_0 和 x_n 可以分别为$-\infty$或$+\infty$，或者用一系列不等式 $x_0 < x \leqslant x_1, \cdots, x_{n-1} < x < x_n$ 表示。

非监督离散化 用于分类的离散化方法之间的根本区别在于使用类信息（监督，supervised）还是不使用类信息（非监督，unsupervised）。如果不使用类信息，则常使用一些相对简单的方法。例如，**等宽**（equal width）方法将属性的值域划分成具有相同宽度的区间，而区间的个数由用户指定。这种方法可能受离群点的影响而性能不佳，因此**等频率**（equal frequncy）或**等深**（equal depth）方法通常更为可取。等频率方法试图将相同数量的对象放进每个区间。作为非监督离散化的另一个例子，可以使用诸如 K 均值（见第 8 章）等聚类方法。最后，目测检查数据有时也可能是一种有效的方法。

例 2.12　离散化技术　本例解释如何对实际数据集使用这些技术。图 2-13a 显示了属于四个不同组的数据点，以及两个离群点——位于两边的大点。可以使用上述技术将这些数据点的 x 值离散化成四个分类值。（数据集中的点具有随机的 y 分量，可以更容易地看出每组有多少个点。）尽管目测检查该数据的方法效果很好，但不是自动的，因此我们主要讨论其他三种方法。使用等宽、等频率和 K 均值技术产生的分割点分别如图 2-13b、图 2-13c 和图 2-13d 所示，图中分割点用虚线表示。如果我们用不同组的不同对象被指派到相同分类值的程度来度量离散化技术的性能，则 K 均值性能最好，其次是等频率，最后是等宽。　□

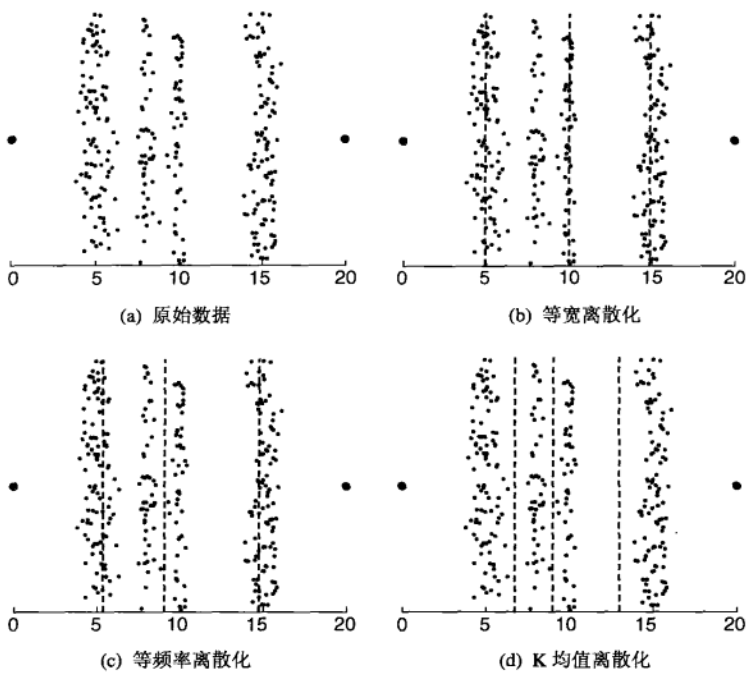


图 2-13　不同的离散化技术

监督离散化　上面介绍的离散化方法通常比不离散化好，但是记住最终目的并使用附加的信息（类标号）常常能够产生更好的结果。这并不奇怪，因为未使用类标号知识所构造的区间常常包含混合的类标号。一种概念上的简单方法是以极大化区间纯度的方式确定分割点。然而，实践中这种方法可能需要人为确定区间的纯度和最小的区间大小。为了解决这一问题，一些基于统计学的方法用每个属性值来分隔区间，并通过合并类似于根据统计检验得出的相邻区间来创建较大的区间。基于熵的方法是最有前途的离散化方法之一，我们将给出一种简单的基于熵的方法。

首先，需要定义**熵**（entropy）。设 k 是不同的类标号数，m_i 是某划分的第 i 个区间中值的个数，而 m_{ij} 是区间 i 中类 j 的值的个数。第 i 个区间的熵 e_i 由如下等式给出

$$e_i = -\sum_{j=1}^{k} p_{ij} \log_2 p_{ij}$$

其中，$p_{ij} = m_{ij}/m_i$ 是第 i 个区间中类 j 的概率（值的比例）。该划分的总熵 e 是每个区间的熵的加

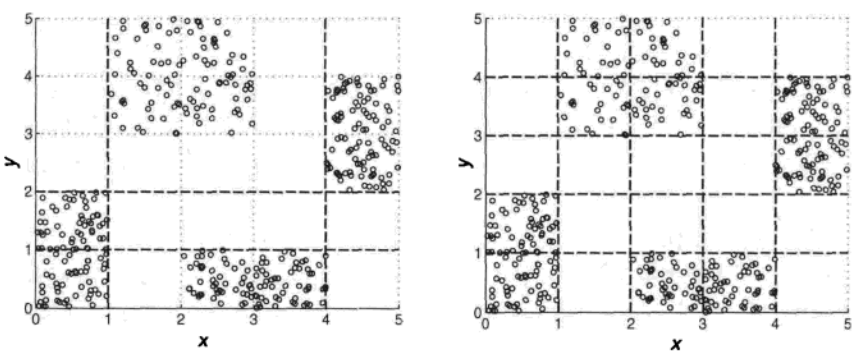
权平均，即

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

其中，m 是值的个数，$w_i = m_i/m$ 是第 i 个区间的值的比例，而 n 是区间个数。直观上，区间的熵是区间纯度的度量。如果一个区间只包含一个类的值（该区间非常纯），则其熵为 0 并且不影响总熵。如果一个区间中的值类出现的频率相等（该区间尽可能不纯），则其熵最大。

一种划分连续属性的简单方法是：开始，将初始值切分成两部分，让两个结果区间产生最小熵。该技术只需要把每个值看作可能的分割点即可，因为假定区间包含有序值的集合。然后，取一个区间，通常选取具有最大熵的区间，重复此分割过程，直到区间的个数达到用户指定的个数，或者满足终止条件。

例 2.13 两个属性离散化 该方法用来独立地离散化图 2-14 所示的二维数据的属性 x 和 y。在图 2-14a 所示的第一个离散化中，属性 x 和 y 被划分成三个区间。（虚线指示分割点。）在图 2-14b 所示的第二个离散化中，属性 x 和 y 被划分成五个区间。 □



(a) 三个区间　　(b) 五个区间

图 2-14 离散化四个点组（类）的属性 x 和 y

这个简单的例子解释了离散化的两个特点。首先，在二维中，点类是很好分开的，但在一维中，情况并非如此。一般而言，分别离散化每个属性通常只保证次最优的结果。其次，五个区间比三个好，但是，至少从熵的角度看，六个区间对离散化的改善不大。（没有给出六个区间的熵值和结果。）因而需要有一个终止标准，自动地发现划分的正确个数。

3. 具有过多值的分类属性

分类属性有时可能具有过多的值。如果分类属性是序数属性，则可以使用类似于处理连续属性的技术，以减少分类值的个数。然而，如果分类属性是标称的，就需要使用其他方法。考虑一所大学，它有许多系，因而系名属性可能具有数十个不同的值。在这种情况下，我们可以使用系之间联系的知识，将系合并成较大的组，如工程学、社会科学或生物科学。如果领域知识不能提供有用的指导，或者这样的方法会导致很差的分类性能，则需要使用更为经验性的方法，如仅当分组结果能提高分类准确率或达到某种其他数据挖掘目标时，才将值聚集到一起。

2.3.7　变量变换

变量变换（variable transformation）是指用于变量的所有值的变换。（尽管我们也偶尔用属性变换这个术语，但是遵循习惯用法，我们使用变量指代属性。）换言之，对于每个对象，变换都作用于该对象的变量值。例如，如果只考虑变量的量级，则可以通过取绝对值对变量进行变换。接下来的部分，我们讨论两种重要的变量变换类型：简单函数变换和规范化。

1. 简单函数

对于这种类型的变量变换，一个简单数学函数分别作用于每一个值。如果 x 是变量，这种变换的例子包括 x^k, $\log x$, e^x, $\sqrt{x}$, $1/x$, $\sin x$ 和 $|x|$。在统计学中，变量变换（特别是平方根、对数和倒数变换）常用来将不具有高斯（正态）分布的数据变换成具有高斯（正态）分布的数据。尽管这可能很重要，但是在数据挖掘中，其他理由可能更重要。假定感兴趣的变量是一次会话中的数据字节数，并且字节数的值域范围为 1 到 10 亿。这是一个很大的值域，使用常用对数变换将其进行压缩可能是有益的。这样的话，传输 10^8 和 10^9 字节的会话比传输 10 字节和 1000 字节的会话更为相似（9 − 8 = 1 对 3 − 1 = 2）。对于某些应用，如网络入侵检测，可能需要如此，因为前两个会话多半表示传输两个大文件，而后两个会话可能是两个完全不同的类型。

使用变量变换时需要小心，因为它们改变了数据的特性。尽管有时需要这样做，但是如果变换的特性没有深入理解，则可能出现问题。例如，变换 $1/x$ 虽然压缩了大于 1 的值，但是却放大了 0 和 1 之间的值，举例来说，{1, 2, 3}变换成{1, 1/2, 1/3}，但是{1, 1/2, 1/3}变换成{1, 2, 3}，这样，对于所有的值集，变换 $1/x$ 逆转了序。为了帮助弄清楚一个变换的效果，重要的是要问如下问题：需要保序吗？变换作用于所有的值，特别是负值和 0 吗？变换对 0 和 1 之间的值有何特别影响？本章习题 17 考察了变量变换的其他方面。

2. 规范化或标准化

另一种常见的变量变换类型是变量的**标准化**（standardization）或**规范化**（normalization）。在数据挖掘界，这两个术语常常可互换，然而，在统计学中，术语规范化可能与使变量**正态**（**高斯**）的变换相混淆。标准化或规范化的目标是使整个值的集合具有特定的性质。一个传统的例子是统计学中的“对变量标准化”。如果 $\bar{x}$ 是属性值的均值（平均值），而 s_x 是它们的标准差，则变换 $x' = (x - \bar{x})/s_x$ 创建一个新的变量，它具有均值 0 和标准差 1。如果要以某种方法组合不同的变量，则为了避免具有较大值域的变量左右计算结果，这种变换常常是必要的。例如，考虑使用年龄和收入两个变量对人进行比较。对于任意两个人，收入之差的绝对值（数百或数千元）多半比年龄之差的绝对值（小于 150）大很多。如果没有考虑到年龄和收入值域的差别，则对人的比较将被收入之差所左右。例如，如果两个人之间的相似性或相异性使用本章后面的相似性或相异性度量来计算，则在很多情况下（如欧几里得距离）收入值将左右计算结果。

均值和标准差受离群点的影响很大，因此通常需要修改上述变换。首先，用**中位数**（median）（即中间值）取代均值。其次，用**绝对标准差**（absolute standard deviation）取代标准差。例如，如果 x 是变量，则 x 的绝对标准差为 $\sigma_A = \sum_{i=1}^{m} |x_i - \mu|$，其中 x_i 是变量 x 的第 i 个值，m 是对象的个数，而 μ 是均值或中位数。存在离群点时，计算值集的位置（中心）和发散估计的其他方法分别在 3.2.3 节和 3.2.4 节介绍。这些度量也可以用来定义标准化变换。

2.4　相似性和相异性的度量

相似性和相异性是重要的概念，因为它们被许多数据挖掘技术所使用，如聚类、最近邻分类

和异常检测等。在许多情况下，一旦计算出相似性或相异性，就不再需要原始数据了。这种方法可以看作将数据变换到相似性（相异性）空间，然后进行分析。

首先，我们讨论基本要素——相似性和相异性的高层定义，并讨论它们之间的联系。为方便起见，我们使用术语**邻近度**（proximity）表示相似性或相异性。由于两个对象之间的邻近度是两个对象对应属性之间的邻近度的函数，因此我们首先介绍如何度量仅包含一个简单属性的对象之间的邻近度，然后考虑具有多个属性的对象的邻近度度量。这包括相关和欧几里得距离度量，以及 Jaccard 和余弦相似性度量。前二者适用于时间序列这样的稠密数据或二维点，后二者适用于像文档这样的稀疏数据。接下来，我们考虑与邻近度度量相关的若干重要问题。本节最后简略讨论如何选择正确的邻近度度量。

2.4.1 基础

1. 定义

两个对象之间的**相似度**（similarity）的非正式定义是这两个对象相似程度的数值度量。因而，两个对象越相似，它们的相似度就越高。通常，相似度是非负的，并常常在 0（不相似）和 1（完全相似）之间取值。

两个对象之间的**相异度**（dissimilarity）是这两个对象差异程度的数值度量。对象越类似，它们的相异度就越低。通常，术语**距离**（distance）用作相异度的同义词，正如我们将介绍的，距离常常用来表示特定类型的相异度。有时，相异度在区间[0, 1]中取值，但是相异度在 0 和∞之间取值也很常见。

2. 变换

通常使用变换把相似度转换成相异度或相反，或者把邻近度变换到一个特定区间，如[0, 1]。例如，我们可能有相似度，其值域从 1 到 10，但是我们打算使用的特定算法或软件包只能处理相异度，或只能处理[0, 1]区间的相似度。之所以在这里讨论这些问题，是因为在稍后讨论邻近度时，我们将使用这种变换。此外，这些问题相对独立于特定的邻近度度量。

通常，邻近度度量（特别是相似度）被定义为或变换到区间[0, 1]中的值。这样做的动机是使用一种适当的尺度，由邻近度的值表明两个对象之间的相似（或相异）程度。这种变换通常是比较直截了当的。例如，如果对象之间的相似度在 1（一点也不相似）和 10（完全相似）之间变化，则我们可以使用如下变换将它变换到[0, 1]区间：$s' = (s - 1)/9$，其中 s 和 s'分别是相似度的原值和新值。一般来说，相似度到[0, 1]区间的变换由如下表达式给出：$s' = (s - min\_s) / (max\_s - min\_s)$，其中 max_s 和 min_s 分别是相似度的最大值和最小值。类似地，具有有限值域的相异度也能用 $d' = (d - min\_d) / (max\_d - min\_d)$ 映射到[0, 1]区间。

然而，将邻近度映射到[0, 1]区间可能非常复杂。例如，如果邻近度度量原来在区间[0, ∞]上取值，则需要使用非线性变换，并且在新的尺度上，值之间不再具有相同的联系。对于从 0 变化到∞的相异度度量，考虑变换 $d' = d / (1 + d)$，相异度 0、0.5、2、10、100 和 1 000 分别被变换到 0、0.33、0.67、0.90、0.99 和 0.999。在原来相异性尺度上较大的值被压缩到 1 附近，但是否希望如此则取决于应用。另一个问题是邻近度度量的含义可能会被改变。例如，相关性（稍后讨论）是一种相似性度量，在区间[–1, 1]上取值，通过取绝对值将这些值映射到[0, 1]区间丢失了符号信息，而对于某些应用，符号信息可能是重要的（见本章习题 22）。

将相似度变换成相异度或相反也是比较直截了当的，尽管我们可能再次面临保持度量的含义问题和将线性尺度改变成非线性尺度的问题。如果相似度（相异度）落在[0, 1]区间，则相异度（相似度）可以定义为 $d = 1 - s$（或 $s = 1 - d$）。另一种简单的方法是定义相似度为负的相异度（或相反）。例如，相异度 0，1，10 和 100 可以分别变换成相似度 0, −1, −10 和−100。

负变换产生的相似度结果不必局限于[0, 1]区间，但是，如果希望的话，则可以使用变换 $s = 1/(d+1)$，$s = e^{-d}$ 或 $s = 1 - \dfrac{d - min\_d}{max\_d - min\_d}$。对于变换 $s = 1/(d+1)$，相异度 0, 1, 10, 100 分别被变换到 1, 0.5, 0.09, 0.01；对于 $s = e^{-d}$，它们分别被变换到 1.00, 0.37, 0.00, 0.00；对于 $s = 1 - \dfrac{d - min\_d}{max\_d - min\_d}$，它们分别被变换到 1.00, 0.99, 0.00, 0.00。在这里的讨论中，我们关注将相异度变换到相似度。相反方向的转换参见本章习题 23。

一般来说，任何单调减函数都可以用来将相异度转换到相似度（或相反）。当然，在将相似度变换到相异度（或相反），或者在将邻近度的值变换到新的尺度时，也必须考虑一些其他因素。我们提到过一些问题，涉及保持意义、扰乱标度和数据分析工具的需要，但是肯定还有其他问题。

2.4.2　简单属性之间的相似度和相异度

通常，具有若干属性的对象之间的邻近度用单个属性的邻近度的组合来定义，因此我们首先讨论具有单个属性的对象之间的邻近度。考虑由一个标称属性描述的对象，对于两个这样的对象，相似意味什么呢？由于标称属性只携带了对象的相异性信息，因此我们只能说两个对象有相同的值，或者没有。因而在这种情况下，如果属性值匹配，则相似度定义为 1，否则为 0；相异度用相反的方法定义：如果属性值匹配，相异度为 0，否则为 1。

对于具有单个序数属性的对象，情况更为复杂，因为必须考虑序信息。考虑一个在标度{*poor*, *fair*, *OK*, *good*, *wonderful*}上测量产品（例如，糖块）质量的属性。一个评定为 *wonderful* 的产品 P1 与一个评定为 *good* 的产品 P2 应当比它与一个评定为 *OK* 的产品 P3 更接近。为了量化这种观察，序数属性的值常常映射到从 0 或 1 开始的相继整数，例如，{*poor* = 0, *fair* =1, *OK* = 2, *good* = 3, *wonderful* = 4}。于是，P1 与 P2 之间的相异度 d(P1, P2) = 3 − 2 = 1，或者，如果我们希望相异度在 0 和 1 之间取值，d(P1, P2) = (3 − 2)/4 = 0.25；序数属性的相似度可以定义为 $s = 1 - d$。

序数属性相似度（相异度）的这种定义可能使读者感到有点担心，因为这里我们定义了相等的区间，而事实并非如此。如果根据实际情况，我们应该计算出区间或比率属性。值 *fair* 与 *good* 的差真和 *OK* 与 *wonderful* 的差相同吗？可能不相同，但是在实践中，我们的选择是有限的，并且在缺乏更多信息的情况下，这是定义序数属性之间邻近度的标准方法。

对于区间或比率属性，两个对象之间的相异性的自然度量是它们的值之差的绝对值。例如，我们可能将现在的体重与一年前的体重相比较，说“我重了 10 磅。”在这类情况下，相异度通常在 0 和∞之间，而不是在 0 和 1 之间取值。如前所述，区间或比率属性的相似度通常转换成相异度。

表 2-7 总结了这些讨论。在该表中，x 和 y 是两个对象，它们具有一个指明类型的属性，$d(x, y)$ 和 $s(x, y)$分别是 x 和 y 之间的相异度和相似度（分别用 d 和 s 表示）。其他方法也是可能的，但是表中的这些是最常用的。

表 2-7 简单属性的相似度和相异度

属性类型	相异度	相似度
标称的	$d=\begin{cases}0 & \text{如果} x=y\\ 1 & \text{如果} x\neq y\end{cases}$	$s=\begin{cases}1 & \text{如果} x=y\\ 0 & \text{如果} x\neq y\end{cases}$
序数的	$d=\lvert x-y\rvert/(n-1)$ （值映射到整数 0 到 $n-1$，其中 n 是值的个数）	$s=1-d$
区间或比率的	$d=\lvert x-y\rvert$	$s=-d$，$s=\frac{1}{1+d}$，$s=e^{-d}$，$s=1-\frac{d-min\_d}{max\_d-min\_d}$

下面两节介绍更复杂的涉及多个属性的对象之间的邻近性度量：(1)数据对象之间的相异度；(2)数据对象之间的相似度。这样分节可以更自然地展示使用各种邻近度度量的基本动机。然而，我们要强调的是使用上述技术，相似度可以变换成相异度，反之亦然。

2.4.3 数据对象之间的相异度

本节，我们讨论各种不同类型的相异度。我们从讨论距离（距离是具有特定性质的相异度）开始，然后给出一些更一般的相异度类型的例子。

距离

我们首先给出一些例子，然后使用距离的常见性质更正式地介绍距离。一维、二维、三维或高维空间中两个点 **x** 和 **y** 之间的**欧几里得距离**（Euclidean distance）d 由如下熟悉的公式定义：

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

其中，n 是维数，而 x_k 和 y_k 分别是 **x** 和 **y** 的第 k 个属性值（分量）。我们用图 2-15、表 2-8 和表 2-9 解释该公式，它们展示了这个点集、这些点的 x 和 y 坐标以及包含这些点之间距离的**距离矩阵**（distance matrix）。

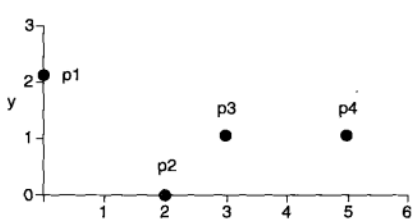


图 2-15 四个二维点

表 2-8 四个点的 x 和 y 坐标

点	x 坐标	y 坐标
p1	0	2
p2	2	0
p3	3	1
p4	5	1

表 2-9 表 2-8 的欧几里得距离矩阵

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

公式（2-1）给出的欧几里得距离可以用公式（2-2）的**闵可夫斯基距离**（Minkowski distance）来推广：

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} | x_k - y_k |^r \right)^{1/r} \tag{2-2}$$

其中 r 是参数。下面是闵可夫斯基距离的三个最常见的例子。

- $r = 1$，城市街区（也称曼哈顿、出租车、L_1 范数）距离。一个常见的例子是**汉明距离**（Hamming distance），它是两个具有二元属性的对象（即两个二元向量）之间不同的二进制位个数。
- $r = 2$，欧几里得距离（L_2 范数）。
- $r = \infty$，上确界（L_{max} 或 L_∞ 范数）距离。这是对象属性之间的最大距离。更正式地，L_∞ 距离由公式（2-3）定义：

$$d(\mathbf{x}, \mathbf{y}) = \lim_{r \to \infty} \left(\sum_{k=1}^{n} | x_k - y_k |^r \right)^{1/r} \tag{2-3}$$

注意不要将参数 r 与维数（属性数）n 混淆。欧几里得距离、曼哈顿距离和上确界距离是对 n 的所有值（1, 2, 3,…）定义的，并且指定了将每个维（属性）上的差的组合成总距离的不同方法。

表 2-10 和表 2-11 分别给出表 2-8 数据的 L_1 距离和 L_∞ 距离的邻近度矩阵。注意，所有的距离矩阵都是对称的，即第 ij 个表目与第 ji 个表目相同，例如，在表 2-9 中，第 4 行第 1 列和第 1 行第 4 列都包含值 5.1。

表 2-10　表 2-8 的 L_1 距离矩阵

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

表 2-11　表 2-8 的 L_∞ 距离矩阵

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

距离（如欧几里得距离）具有一些众所周知的性质。如果 $d(\mathbf{x}, \mathbf{y})$ 是两个点 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的距离，则如下性质成立。

(1) **非负性**。(a) 对于所有 $\mathbf{x}$ 和 $\mathbf{y}$，$d(\mathbf{x}, \mathbf{y}) \geqslant 0$，(b) 仅当 $\mathbf{x} = \mathbf{y}$ 时 $d(\mathbf{x}, \mathbf{y}) = 0$。

(2) **对称性**。对于所有 $\mathbf{x}$ 和 $\mathbf{y}$，$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$。

(3) **三角不等式**。对于所有 $\mathbf{x}$，$\mathbf{y}$ 和 $\mathbf{z}$，$d(\mathbf{x}, \mathbf{z}) \leqslant d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$。

满足以上三个性质的测度称为**度量**（metric）。有些人只对满足这三个性质的相异性度量使用术语距离，但在实践中常常违反这一约定。这里介绍的三个性质是有用的，数学上也是令人满意的。此外，如果三角不等式成立，则该性质可以用来提高依赖于距离的技术（包括聚类）的效率（见本章习题 25）。尽管如此，许多相异度都不满足一个或多个度量性质。下面我们给出两个这种测度的例子。

例 2.14　非度量的相异度：集合差　基于集合论中定义的两个集合差的概念举例。设有两个

集合 A 和 B，$A-B$ 是不在 B 中的 A 中元素的集合。例如，如果 $A=\{1,2,3,4\}$，而 $B=\{2,3,4\}$，则 $A-B=\{1\}$，而 $B-A=\varnothing$，即空集。我们可以将两个集合 A 和 B 之间的距离定义为 $d(A,B)=size(A-B)$，其中 $size$ 是一个函数，它返回集合元素的个数。该距离测度是大于或等于零的整数值，但不满足非负性的第二部分，也不满足对称性，同时还不满足三角不等式。然而，如果将相异度修改为 $d(A,B)=size(A-B)+size(B-A)$，则这些性质都可以成立（见本章习题 21）。 □

例 2.15 非度量的相异度：时间 这里给出一个更常见的例子，其中相异性测度并非度量，但依然是有用的。定义时间之间的距离测度如下：

$$d(t_1,t_2)=\begin{cases}t_2-t_1 & \text{如果 } t_1\leqslant t_2\\ 24+(t_2-t_1) & \text{如果 } t_1\geqslant t_2\end{cases} \tag{2-4}$$

例如，d(1PM, 2PM) = 1 小时，而 d(2PM, 1PM) = 23 小时。这种定义是有意义的，例如，在回答如下问题时就体现了这种定义的意义：“如果一个事件在每天下午 1 点发生，现在是下午 2 点，那么我们还需要等待多长时间才能等到该事件再度发生？” □

2.4.4 数据对象之间的相似度

对于相似度，三角不等式（或类似的性质）通常不成立，但是对称性和非负性通常成立。更明确地说，如果 $s(\mathbf{x},\mathbf{y})$是数据点 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的相似度，则相似度具有如下典型性质。

(1) 仅当 $\mathbf{x}=\mathbf{y}$ 时 $s(\mathbf{x},\mathbf{y})=1$。（$0\leqslant s\leqslant 1$）

(2) 对于所有 $\mathbf{x}$ 和 $\mathbf{y}$，$s(\mathbf{x},\mathbf{y})=s(\mathbf{y},\mathbf{x})$。（对称性）

对于相似度，没有与三角不等式对应的一般性质。然而，有时可以将相似度简单地变换成一种度量距离。稍后讨论的余弦相似性度量和 Jaccard 相似性度量就是两个例子。此外，对于特定的相似性度量，还可能在两个对象相似性上导出本质上与三角不等式类似的数学约束。

例 2.16 非对称相似性度量 考虑一个实验，实验中要求人们对屏幕上快速闪过的一小组字符进行分类。该实验的**混淆矩阵**（confusion matrix）记录每个字符被分类为自己的次数和被分类为另一个字符的次数。例如，假定“0”出现了 200 次，它被分类为“0”160 次，而被分类为“o”40 次。类似地，“o”出现 200 次并且分类为“o”170 次，但是分类为“0”只有 30 次。如果取这些计数作为两个字符之间相似性的度量，则得到一种相似性度量，但这种相似性度量不是对称的。在这种情况下，通过选取 $s'(\mathbf{x},\mathbf{y})=s'(\mathbf{y},\mathbf{x})=(s(\mathbf{x},\mathbf{y})+s(\mathbf{y},\mathbf{x}))/2$，相似性度量可以转换成对称的，其中 s'是新的相似性度量。 □

2.4.5 邻近性度量的例子

本节给出一些相似性和相异性度量的具体例子。

1. 二元数据的相似性度量

两个仅包含二元属性的对象之间的相似性度量也称为**相似系数**（similarity coefficient），并且通常在 0 和 1 之间取值，值为 1 表明两个对象完全相似，而值为 0 表明对象一点也不相似。有许多理由表明在特定情形下，一种系数为何比另一种好。

设 $\mathbf{x}$ 和 $\mathbf{y}$ 是两个对象，都由 n 个二元属性组成。这样的两个对象（即两个二元向量）的比较可生成如下四个量（频率）：

f_{00} = **x** 取 0 并且 **y** 取 0 的属性个数
f_{01} = **x** 取 0 并且 **y** 取 1 的属性个数
f_{10} = **x** 取 1 并且 **y** 取 0 的属性个数
f_{11} = **x** 取 1 并且 **y** 取 1 的属性个数

简单匹配系数（Simple Matching Coefficient, SMC）　一种常用的相似性系数是**简单匹配系数**，定义如下：

$$SMC = \frac{\text{值匹配的属性个数}}{\text{属性个数}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}} \tag{2-5}$$

该度量对出现和不出现都进行计数。因此，*SMC* 可以在一个仅包含是非题的测验中用来发现回答问题相似的学生。

Jaccard 系数（Jaccard Coefficient）　假定 **x** 和 **y** 是两个数据对象，代表一个事务矩阵（见 2.1.2 节）的两行（两个事务）。如果每个非对称的二元属性对应于商店的一种商品，则 1 表示该商品被购买，而 0 表示该商品未被购买。由于未被顾客购买的商品数远大于被其购买的商品数，因而像 *SMC* 这样的相似性度量将会判定所有的事务都是类似的。这样，常常使用 **Jaccard 系数**来处理仅包含非对称的二元属性的对象。Jaccard 系数通常用符号 *J* 表示，由如下等式定义：

$$J = \frac{\text{匹配的个数}}{\text{不涉及0-0匹配的属性个数}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \tag{2-6}$$

例 2.17　SMC 和 Jaccard 相似性系数　为了解释这两种相似性度量之间的差别，我们对如下二元向量计算 *SMC* 和 *J*：

x = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
y = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)

$f_{01} = 2$　**x** 取 0 并且 **y** 取 1 的属性个数
$f_{10} = 1$　**x** 取 1 并且 **y** 取 0 的属性个数
$f_{00} = 7$　**x** 取 0 并且 **y** 取 0 的属性个数
$f_{11} = 0$　**x** 取 1 并且 **y** 取 1 的属性个数

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

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

□

2. 余弦相似度

通常，文档用向量表示，向量的每个属性代表一个特定的词（术语）在文档中出现的频率。当然，实际情况要复杂得多，因为需要忽略常用词，并使用各种技术处理同一个词的不同形式、不同的文档长度以及不同的词频。

尽管文档具有数以百千计或数以万计的属性（词），但是每个文档向量都是稀疏的，因为它具有相对较少的非零属性值。（文档规范化并不对零词目创建非零词目，即文档规范化保持稀疏性。）这样，与事务数据一样，相似性不能依赖共享 0 的个数，因为任意两个文档多半都不会包

含许多相同的词，从而如果统计 0-0 匹配，则大多数文档都与其他大部分文档非常类似。因此，文档的相似性度量不仅应当像 Jaccard 度量一样需要忽略 0-0 匹配，而且还必须能够处理非二元向量。下面定义的**余弦相似度**（cosine similarity）就是文档相似性最常用的度量之一。如果 $\mathbf{x}$ 和 $\mathbf{y}$ 是两个文档向量，则

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \tag{2-7}$$

其中，“·”表示向量点积，$\mathbf{x} \cdot \mathbf{y} = \sum_{k=1}^{n} x_k y_k$，$\|\mathbf{x}\|$是向量 $\mathbf{x}$ 的长度，$\|\mathbf{x}\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$。

例 2.18 两个文档向量的余弦相似度 该例计算下面两个数据对象的余弦相似度，这些数据对象可能代表文档向量：

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

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

$\mathbf{x} \cdot \mathbf{y} = 3 \times 1 + 2 \times 0 + 0 \times 0 + 5 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 2 = 5$

$\|\mathbf{x}\| = \sqrt{3 \times 3 + 2 \times 2 + 0 \times 0 + 5 \times 5 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 2 \times 2 + 0 \times 0 + 0 \times 0} = 6.48$

$\|\mathbf{y}\| = \sqrt{1 \times 1 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 1 \times 1 + 0 \times 0 + 2 \times 2} = 2.45$

$\cos(\mathbf{x}, \mathbf{y}) = \mathbf{0.31}$ □

如图 2-16 所示，余弦相似度实际上是 $\mathbf{x}$ 和 $\mathbf{y}$ 之间夹角（余弦）的度量。这样，如果余弦相似度为 1，则 $\mathbf{x}$ 和 $\mathbf{y}$ 之间夹角为 0°，并且除大小（长度）之外，$\mathbf{x}$ 和 $\mathbf{y}$ 是相同的；如果余弦相似度为 0，则 $\mathbf{x}$ 和 $\mathbf{y}$ 之间夹角为 90°，并且它们不包含任何相同的词（术语）。

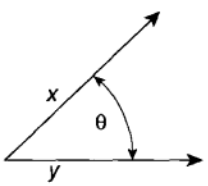


图 2-16 余弦度量的几何解释

公式（2-7）可以写成公式（2-8）的形式：

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}}{\|\mathbf{x}\|} \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|} = \mathbf{x}' \cdot \mathbf{y}' \tag{2-8}$$

其中，$\mathbf{x}' = \mathbf{x} / \|\mathbf{x}\|$，而 $\mathbf{y}' = \mathbf{y} / \|\mathbf{y}\|$。$\mathbf{x}$ 和 $\mathbf{y}$ 被它们的长度除，将它们规范化成具有长度 1。这意味在计算相似度时，余弦相似度不考虑两个数据对象的量值。（当量值是重要的时，欧几里得距离可能是一种更好的选择。）对于长度为 1 的向量，余弦度量可以通过简单地取点积计算。从而，在需要计算大量对象之间的余弦相似度时，将对象规范化，使之具有单位长度可以减少计算时间。

3. 广义 Jaccard 系数

广义 Jaccard 系数可以用于文档数据，并在二元属性情况下归约为 Jaccard 系数。广义 Jaccard 系数又称 Tanimoto 系数。（然而，还有一种系数也称 Tanimoto 系数。）该系数用 *EJ* 表示，由下式定义：

$$EJ(\mathbf{x},\mathbf{y})=\frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|^2+\|\mathbf{y}\|^2-\mathbf{x}\cdot\mathbf{y}} \tag{2-9}$$

4. 相关性

两个具有二元变量或连续变量的数据对象之间的相关性是对象属性之间线性联系的度量。（更一般属性之间的相关性计算可以类似地定义。）更准确地，两个数据对象 **x** 和 **y** 之间的**皮尔森相关**（Pearson's correlation）系数由下式定义：

$$\text{corr}(\mathbf{x},\mathbf{y})=\frac{\text{covariance}(\mathbf{x},\mathbf{y})}{\text{standard\_deviation}(\mathbf{x})\times\text{standard\_deviation}(\mathbf{y})}=\frac{s_{xy}}{s_x s_y} \tag{2-10}$$

这里我们使用标准的统计学记号和定义：

$$\text{covariance}(\mathbf{x},\mathbf{y})=s_{xy}=\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\bar{x})(y_k-\bar{y}) \tag{2-11}$$

$$\text{standard\_deviation}(\mathbf{x})=s_x=\sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\bar{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y})=s_y=\sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k-\bar{y})^2}$$

$$\bar{x}=\frac{1}{n}\sum_{k=1}^{n}x_k \text{ 是 } \mathbf{x} \text{ 的均值}$$

$$\bar{y}=\frac{1}{n}\sum_{k=1}^{n}y_k \text{ 是 } \mathbf{y} \text{ 的均值}$$

例 2.19　完全相关　相关度总是在−1 到 1 之间取值。相关度为 1（−1）意味 **x** 和 **y** 具有完全正（负）线性关系，即 $x_k=ay_k+b$，其中 a 和 b 是常数。下面两个 **x** 和 **y** 的值集分别给出相关度为−1 和+1 的情况。为简单起见，第一组中取 **x** 和 **y** 的均值为 0。

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

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

例 2.20　非线性关系　如果相关度为 0，则两个数据对象的属性之间不存在线性关系。然而，仍然可能存在非线性关系。在下面的例子中，数据对象的属性之间存在非线性关系 $y_k=x_k^2$，但是它们的相关度为 0。

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

例 2.21　相关性可视化　通过绘制对应属性值对可以很容易地判定两个数据对象 **x** 和 **y** 之间的相关性。图 2-17 给出了一些这种图，**x** 和 **y** 具有 30 个属性，这些属性的值随机地产生（服从正态分布），使得 **x** 和 **y** 的相关度从−1 到 1。图中每个小圆圈代表 30 个属性中的一个，其 x 坐标是 **x** 的一个属性的值，而其 y 坐标是 **y** 的相同属性的值。 □

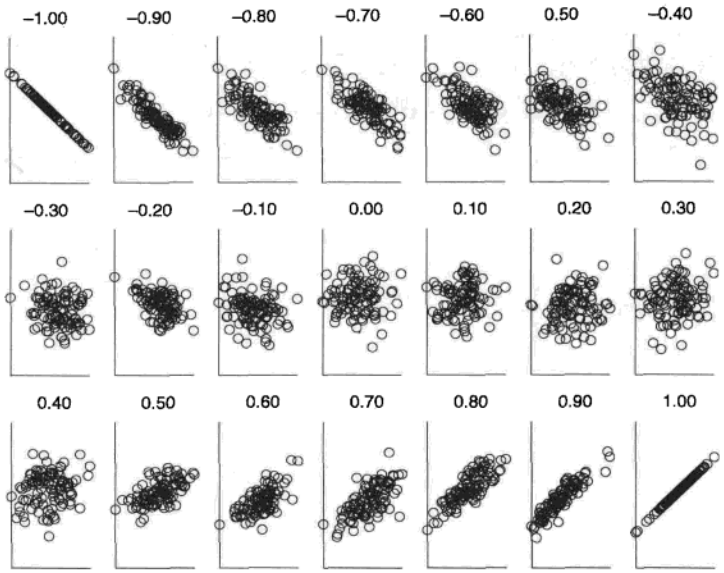


图 2-17 解释相关度从-1 到 1 的散布图

如果通过减去均值，然后规范化使其长度为 1 来变换 **x** 和 **y**，则它们的相关度可以通过求点积来计算。注意，这与其他情况下使用的标准化不同，在其他情况下，我们使用变换 $x'_k = (x_k - \bar{x})/s_x$ 和 $y'_k = (y_k - \bar{y})/s_y$。

Bregman 散度* 本节，我们简略介绍 Bregman 散度（Bregman divergence），它是一族具有共同性质的邻近函数。这样，可以构造使用 Bregman 发散函数的一般数据挖掘算法，如聚类算法，具体的例子是 K 均值聚类算法（8.2 节）。注意，本节需要向量计算方面的知识。

Bregman 散度是损失或失真函数。为了理解损失函数，考虑如下情况：设 **x** 和 **y** 是两个点，其中 **y** 是原来的点，而 **x** 是它的某个失真或近似，例如，**x** 可能是由于添加了一些随机噪声到 **y** 上而产生的。损失函数的目的是度量用 **x** 近似 **y** 导致的失真或损失。当然，**x** 和 **y** 越类似，失真或损失就越小，因而 Bregman 散度可以用作相异性函数。

有如下正式定义。

定义 2.6 Bregman 散度 给定一个严格凸函数 ϕ（连同一些通常满足的适度限制），由该函数生成的 Bregman 散度（损失函数）$D(\mathbf{x}, \mathbf{y})$通过下面的公式给出：

$$D(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \nabla\phi(\mathbf{y}), (\mathbf{x} - \mathbf{y}) \rangle \tag{2-12}$$

其中，$\nabla\phi(\mathbf{y})$是在 **y** 上计算的 ϕ 的梯度，$\mathbf{x} - \mathbf{y}$ 是 **x** 与 **y** 的向量差，而$\langle \nabla\phi(\mathbf{y}), (\mathbf{x} - \mathbf{y}) \rangle$是$\nabla\phi(\mathbf{y})$和$(\mathbf{x} - \mathbf{y})$的内积。对于欧几里得空间中的点，内积就是点积。

$D(\mathbf{x}, \mathbf{y})$可以写成 $D(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - L(\mathbf{x})$，其中 $L(\mathbf{x}) = \phi(\mathbf{y}) + \langle \nabla\phi(\mathbf{y}), (\mathbf{x} - \mathbf{y}) \rangle$代表在 **y** 上正切于函数 ϕ 的平面方程。使用微积分学的术语，$L(\mathbf{x})$是函数 ϕ 在 **y** 点附近的线性部分，而 Bregman 散度是一个函数与该函数的线性近似之间的差。选取不同的 ϕ，可以得到不同的 Bregman 散度。

例 2.22 我们使用平方欧几里得距离给出 Bregman 散度的一个具体例子。为了简化数学计

算，我们仅限于一维。设 x 和 y 是实数，而 $\phi(t)$ 是实数值函数，$\phi(t) = t^2$。在此情况下，梯度归结为导数，而点积归结为乘积。例如，公式（2-12）变成公式（2-13）。

$$D(x, y) = x^2 - y^2 - 2y(x - y) = (x - y)^2 \tag{2-13}$$

该例的图形在图 2-18 中给出，其中 $y = 1$。在 $x = 2$ 和 $x = 3$ 上给出了 Bregman 散度。 □

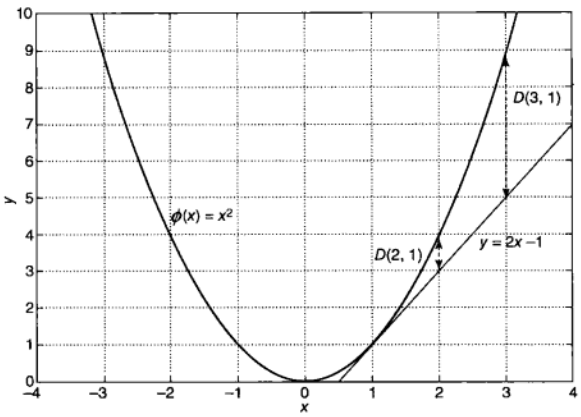


图 2-18　图示 Bregman 散度

2.4.6　邻近度计算问题

本节讨论与邻近性度量有关的一些重要问题：(1)当属性具有不同的尺度（scale）或相关时如何处理；(2)当对象包含不同类型的属性（例如，定量属性和定性属性）时如何计算对象之间的邻近度；(3)当属性具有不同的权重（即并非所有的属性都对对象的邻近度具有相等的贡献）时，如何处理邻近度计算。

1. 距离度量的标准化和相关性

距离度量的一个重要问题是当属性具有不同的值域时如何处理。（这种情况通常称作“变量具有不同的尺度。”）前面，使用欧几里得距离，基于年龄和收入两个属性来度量人之间的距离。除非这两个属性是标准化的，否则两个人之间的距离将被收入所左右。

一个相关的问题是，除值域不同外，当某些属性之间还相关时，如何计算距离。当属性相关、具有不同的值域（不同的方差）、并且数据分布近似于高斯（正态）分布时，欧几里得距离的拓广，Mahalanobis 距离是有用的。具体地说，两个对象（向量）$\mathbf{x}$ 和 $\mathbf{y}$ 之间的 **Mahalanobis 距离**定义为：

$$\text{mahalanobis}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{y})^T \tag{2-14}$$

其中 $\boldsymbol{\Sigma}^{-1}$ 是数据协方差矩阵的逆。注意，协方差矩阵 $\boldsymbol{\Sigma}$ 是这样的矩阵，它的第 ij 个元素是第 i 个和第 j 个属性的协方差，由公式（2-11）定义。

例 2.23　在图 2-19 中有 1000 个点，其 x 属性和 y 属性的相关度为 0.6。在椭圆长轴两端的两个大点之间的欧几里得距离为 14.7，但 Mahalanobis 距离仅为 6。实践中，计算 Mahalanobis 距离的费用昂贵，但是对于其属性相关的对象来说是值得的。如果属性相对来说不相关，只是具有不同的值域，则只需要对变量进行标准化就足够了。 □

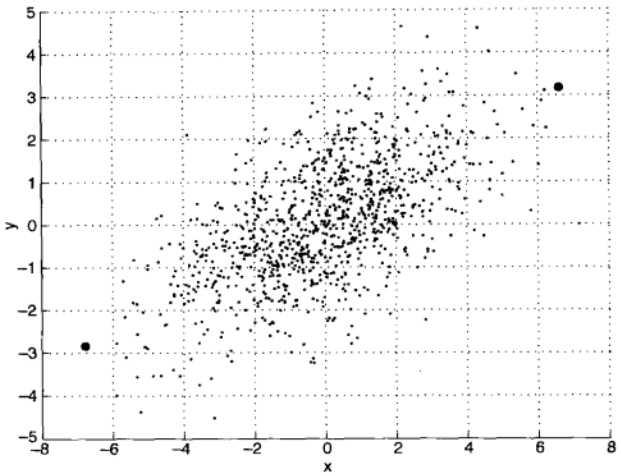


图 2-19 二维点的集合。两个大点代表的点之间的 Mahalanobis 距离为 6，它们的欧几里得距离为 14.7

2. 组合异种属性的相似度

前面的相似度定义所基于的方法都假定所有属性具有相同类型。当属性具有不同类型时，就需要更一般的方法。直截了当的方法是使用表 2-7 分别计算出每个属性之间的相似度，然后使用一种导致0和1之间相似度的方法组合这些相似度。总相似度一般定义为所有属性相似度的平均值。

不幸的是，如果某些属性是非对称属性，这种方法效果不好。例如，如果所有的属性都是非对称的二元属性，则相似性度量先归结为简单匹配系数——一种对于二元非对称属性并不合适的度量。处理该问题的最简单方法是：如果两个对象在非对称属性上的值都是 0，则在计算对象相似度时忽略它们。类似的方法也能很好地处理遗漏值。

概括地说，算法 2.1 可以有效地计算具有不同类型属性的两个对象 **x** 和 **y** 之间的相似度。修改该过程可以很轻松地处理相异度。

算法 2.1 异种对象的相似度

1：对于第 k 个属性，计算相似度 $s_k(\mathbf{x}, \mathbf{y})$，在区间[0, 1]中。

2：对于第 k 个属性，定义一个指示变量 δ_k，如下：

$\delta_k=0$，如果第 k 个属性是非对称属性，并且两个对象在该属性上的值都是 0，或者如果一个对象的第 k 个属性具有遗漏值

$\delta_k=1$，否则

3：使用如下公式计算两个对象之间的总相似度：

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^{n} \delta_k} \tag{2-15}$$

3. 使用权值

在前面的大部分讨论中，所有的属性在计算邻近度时都会被同等对待。但是，当某些属性对邻近度的定义比其他属性更重要时，我们并不希望这种同等对待的方式。为了处理这种情况，可以通过对每个属性的贡献加权来修改邻近度公式。

如果权 w_k 的和为 1，则公式（2-15）变成

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^{n} \delta_k} \tag{2-16}$$

闵可夫斯基距离的定义也可以修改为：

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} w_k \mid x_k - y_k \mid^r \right)^{1/r} \tag{2-17}$$

2.4.7　选取正确的邻近性度量

下面是一些一般观察，可能会对你有所帮助。首先，邻近性度量的类型应当与数据类型相适应。对于许多稠密的、连续的数据，通常使用距离度量，如欧几里得距离等。连续属性之间的邻近度通常用属性值的差来表示，并且距离度量提供了一种将这些差组合到总邻近性度量的良好方法。尽管属性可能有不同的取值范围和不同的重要性，但这些问题通常都可以用前面介绍的方法处理。

对于稀疏数据，常常包含非对称的属性，通常使用忽略 0-0 匹配的相似性度量。从概念上讲，这反映了如下事实：对于一对复杂对象，相似度依赖于它们共同具有的性质数目，而不是依赖于它们都缺失的性质数目。在特殊的情况下，对于稀疏的、非对称的数据，大部分对象都只具有少量被属性描述的性质，因此如果考虑它们都不具有的性质的话，它们都高度相似。余弦、Jaccard 和广义 Jaccard 度量对于这类数据是合适的。

数据向量还有一些其他特征需要考虑。例如，假定对于比较时间序列感兴趣。如果时间序列的量值是重要的（例如，每个时间序列表示同一单位不同年份的总销售），则可以使用欧几里得距离。如果时间序列代表不同的量（例如，血压和氧消耗量），通常需要确定时间序列是否具有相同的形状，而不是相同的量值，那么相关度可能更可取（使用考虑量和级的差异的内置规范化）。

在某些情况下，为了得到合适的相似性度量，数据的变换或规范化是重要的，因为这种变换并非总能在邻近性度量中提供，例如，时间序列数据可能具有显著影响相似性的趋势或周期模式。此外，正确地计算相似度还需要考虑时间延迟。最后，两个时间序列可能只在特定的时间周期上相似，例如，气温与天然气的用量之间存在很强的联系，但是这种联系仅出现在取暖季节。

实践考虑也是重要的。有时，一种或多种邻近性度量已经在某个特定领域使用，因此，其他人已经回答了应当使用何种邻近性度量的问题；另外，所使用的软件包或聚类算法可能完全限制了选择；如果关心效率，则我们可能希望选择具有某些性质的邻近性度量，这些性质（如三角不等式）可以用来降低邻近度计算量（见本章习题 25）。

然而，如果通常的实践或实践限制并未规定某种选择，则正确地选择邻近性度量可能是一项耗时的任务，需要仔细地考虑领域知识和度量使用的目的。可能需要评估许多不同的相似性度量，以确定哪些结果最有意义。

文献注释

理解待分析的数据至关重要，并且在基本层面，这是测量理论的主题。比如说，定义属性类型的初始动机是精确地指出哪些统计操作对何种数据是合法的。我们给出了测量理论的概述，这些源于 S. S. Stevens 的经典文章[79]。（表 2-2 和表 2-3 取自 Stevens[80]。）尽管这是最普遍的观点并且相当容易理解和使用，但是测量理论远不止这些。权威的讨论可以在测量理论基础的三卷系

列[63, 69, 81]中找到。同样值得关注的是 Hand[55]的文章，文中广泛地讨论了测量理论和统计学，并且附有该领域其他研究者的评论。最后，有许多书籍和文章都介绍了科学与工程学的特定领域中的测量问题。

数据质量是一个范围广泛的主题，涉及使用数据的每个学科。精度、偏倚、准确率的讨论和一些重要的图可以在许多科学、工程学和统计学的导论性教材中找到。数据质量“适合使用”的观点在 Redman 的书[76]中有更详细的解释。对数据质量感兴趣的人一定也会对 MIT 的总体数据质量管理计划[70, 84]感兴趣。然而，处理特定领域的数据质量问题所需要的知识最好是通过考察该领域的研究者的数据质量实践而得到。

与其他预处理任务相比，聚集是一个不够成形的主题。然而，聚集是数据库联机分析处理（OLAP）领域使用的主要技术之一，这将在第 3 章讨论。聚集在符号数据分析领域也起到了一些作用（Bock 和 Diday[47]）。该领域的一个目标是用符号数据对象汇总传统的记录数据，而符号数据对象的属性比传统属性更复杂。例如，这些属性的值可能是值的集合（类别）、区间、具有权重的值的集合（直方图）。符号数据分析的另一个目标是能够在由符号数据对象组成的数据上进行聚类、分类和其他类型的数据分析。

抽样是一个已经在统计学及其相关领域中透彻研究的主题。许多统计学导论性书籍（如 Lindgren 的书[65]）都有关于抽样的讨论，并且还有通篇讨论该主题的书，如 Cochran 的经典教科书[49]。Gu 和 Liu[54]提供了关于数据挖掘抽样综述，而 Olken 和 Rotem[72]提供了关于数据库抽样的综述。还有许多涉及数据挖掘和数据库抽样的文献也值得关注，包括 Palmer 和 Faloutsos[74]、Provost 等[75]、Toivonen[82]、Zaki 等[85]的文章。

在统计学，已经用于维归约的传统技术是多维定标（MDS）（Borg 和 Groenen[48]，Kruskal 和 Uslaner[64]）和主成分分析（PCA）（Jolliffe[58]），主成分分析类似于奇异值分解（SVD）（Demmel[50]）。

离散化是一个已经在数据挖掘领域广泛讨论的主题。有些分类算法只能使用分类属性，并且关联分析需要二元数据，这样就有了重要的动机，去考察如何最好地对连续属性进行二元化或离散化。对于关联分析，建议读者阅读 Srikant 和 Agrawal 的文章[78]，而分类领域离散化的一些有用的参考文献包括 Dougherty 等[51]、Elomaa 和 Rousu[52]、Fayyad 和 Irani[53]以及 Hussain 等[56]。

特征选择是另一个在数据挖掘领域被彻底研究的主题。Molina 等的综述[71]和 Liu 和 Motada 的两本书[66, 67]提供了涵盖该主题的广泛材料。其他有用的文章包括 Blum 和 Langley[46]、Kohavi 和 John[62]和 Liu 等[68]。

很难提供特征变换主题的参考文献，因为不同学科的实践差异很大。许多统计学书籍都讨论了变换，但是讨论通常都限于特定的目的，如确保变量的规范性，或者确保变量具有相等的方差。我们提供两种参考文献：Osborne[73]和 Tukey[83]。

尽管我们已经讨论了一些最常用的距离和相似性度量，但是还有数以百计的这样的度量，并且更多的度量还正在提出。与本章的其他许多主题一样，许多度量都局限于特定的领域，例如，在时间序列领域，见 Kalpakis 等[59]、Keogh 和 Pazzani[61]的文章。聚类方面的书提供了最好的一般讨论，特别是如下书籍：Anderberg[45]、Jain 和 Dubes[57]、Kaufman 和 Rousseeuw[60]以及 Sneath 和 Sokal[77]。

参考文献

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[68] H. Liu, H. Motoda, and L. Yu. Feature Extraction, Selection, and Construction. In N. Ye, editor, *The*

Handbook of Data Mining, pages 22 - 41. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2003.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

习 题

1. 在第 2 章的第一个例子中，统计人员说：“是的，字段 2 和 3 也有不少问题。”从所显示的三行样本数据，你能解释她为什么这样说吗？

2. 将下列属性分类成二元的、离散的或连续的，并将它们分类成定性的（标称的或序数的）或定量的（区间的或比率的）。某些情况下可能有多种解释，因此如果你认为存在二义性，简略给出你的理由。

   例子：年龄。**回答**：离散的、定量的、比率的。

   (a) 用 AM 和 PM 表示的时间。

   (b) 根据曝光表测出的亮度。

   (c) 根据人的判断测出的亮度。

   (d) 按度测出的 0 和 360 之间的角度。

   (e) 奥运会上授予的铜牌、银牌和金牌。

   (f) 海拔高度。

(g) 医院中的病人数。
(h) 书的 ISBN 号（查找网上的格式）。
(i) 用如下值表示的透光能力：不透明、半透明、透明。
(j) 军衔。
(k) 到校园中心的距离。
(l) 用每立方厘米克表示的物质密度。
(m) 外套寄存号码。（出席一个活动时，你通常会将外套交给服务生，然后他给你一个号码，你可以在离开时用它来领取你的外套。）

3. 某个地方公司的销售主管与你联系，他相信他已经设计出了一种评估顾客满意度的完美方法。他这样解释他的方案："这太简单了，我简直不敢相信，以前竟然没有人想到，我只是记录顾客对每种产品的抱怨次数，我在数据挖掘书中读到计数具有比率属性，因此，我的产品满意度度量必定具有比率属性。但是，当我根据顾客满意度度量评估产品并拿给老板看时，他说我忽略了显而易见的东西，说我的度量毫无价值。我想，他简直是疯了，没发现我们的畅销产品满意度最差，因为对它的抱怨最多。你能帮助我摆平他吗？"
   (a) 谁是对的，销售主管还是他的老板？如果你的回答是他的老板，你需要做些什么来修正满意度度量？
   (b) 对于原来的产品满意度度量的属性类型，你的想法是什么？
4. 几个月之后，习题 3 中提到的那个销售主管又同你联系。这次，他设计了一个更好的方法，用以评估顾客喜爱一种产品超过喜爱其他类似产品的程度。他解释说："在开发一种新产品时，我们通常创建一些变种并评估顾客更喜欢哪一种。我们的标准做法是同时散发所有的产品变种并要求他们根据喜爱程度对产品变种划分等级。然而，我们的评测题目很不明确，当有两个以上产品时尤其如此，这让测试占用了很长的时间。我建议对产品逐对比较，然后使用这些比较来划分等级，这样，如果我们有 3 个产品变种，我们就让顾客比较变种 1 和 2，然后 2 和 3，最后 3 和 1。使用我的方法，评测时间是原来的三分之一，但是进行评测的雇员抱怨说，他们不能从评测结果得到一致的等级评定。昨天，我的老板想要知道最新的产品评估。另外我还得告诉你，老的产品评估方法就是他提出的。你能帮助我吗？"
   (a) 销售主管是否陷入困境？他的方法能够根据顾客的喜好产生产品变种的有序等级吗？解释你的观点。
   (b) 是否有办法修正销售主管的方法？对于基于逐对比较创建序数度量，你作何评价？
   (c) 对于原来的产品评估方案，每个产品变种的总等级通过计算所有评测题目上的平均值得到，你是否认为这是一种合理的方法？你会采取哪种方法？
5. 你能想象一种情况，标识号对于预测是有用的吗？
6. 一位教育心理学家想使用关联分析来分析测试结果。测试包含 100 个问题，每个问题有 4 个可能答案。
   (a) 如何将该数据转换成适合关联分析的形式？
   (b) 能得到何种属性类型以及有多少个属性？
7. 下面哪种量更可能具有时间自相关性：日降水量，日气温？为什么？

8. 讨论：为什么文档-词矩阵是具有非对称的离散特征或非对称的连续特征的数据集的例子？
9. 许多科学领域依赖于观测而不是（或不仅是）设计的实验，比较涉及观测科学与实验科学和数据挖掘的数据质量问题。
10. 讨论测量精度与术语单精度和双精度之间的差别。单精度和双精度用在计算机科学，通常分别表示 32 位和 64 位浮点数。
11. 对于处理存放在文本文件而不是二进制格式中的数据，给出至少两个优点。
12. 区别噪声和离群点。确保考虑以下问题。
    (a) 噪声曾令人感兴趣或使人期望吗？离群点呢？
    (b) 噪声对象可能是离群点吗？
    (c) 噪声对象总是离群点吗？
    (d) 离群点总是噪声对象吗？
    (e) 噪声能将典型值变成例外值吗？反之呢？
13. 考虑发现数据对象的 K 个最近邻问题。某个程序员为该任务设计了算法 2.2。

算法 2.2　发现 K 个最近邻的算法

```
1: for i = 1 到数据对象个数 do
2:     找出第 i 个对象到其他所有对象的距离。
3:     按递减序对这些距离排序。
       （维持对象与距离的关联。）
4:     return 与排序表中前 K 个距离相关联的对象。
5: end for
```

    (a) 如果数据集中存在重复对象，讨论该算法可能存在的问题。假定对于相同的对象，距离函数只返回距离 0。
    (b) 如何解决该问题？
14. 对亚洲象群的成员测量如下属性：重量、高度、象牙长度、象鼻长度和耳朵面积。基于这些测量，可以使用 2.4 节的哪种相似性度量来对这些大象进行比较或分组？论证你的答案并说明特殊情况。
15. 给定 m 个对象的集合，这些对象划分成 K 组，其中第 i 组的大小为 m_i。如果目标是得到容量为 $n < m$ 的样本，下面两种抽样方案有什么区别？（假定使用有放回抽样。）
    (a) 从每组随机地选择 $n \times m_i/m$ 个元素。
    (b) 从数据集中随机地选择 n 个元素，而不管对象属于哪个组。
16. 考虑一个文档-词矩阵，其中 tf_{ij} 是第 i 个词（术语）出现在第 j 个文档中的频率，而 m 是文档数。考虑由下式定义的变量变换：

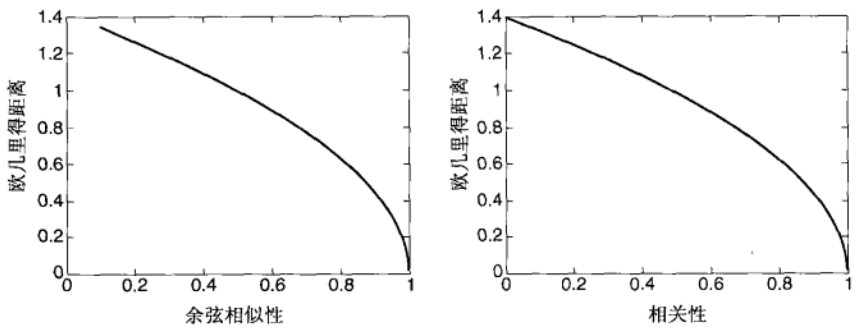
$$tf'_{ij} = tf_{ij} \cdot \log \frac{m}{df_i} \tag{2-18}$$

    其中，df_i 是出现第 i 个词的文档数，称作词的**文档频率**（document frequency）。该变换称作**逆文档频率**（inverse document frequency）变换。
    (a) 如果词出现在一个文档中，该变换的结果是什么？如果术语出现在每个文档中呢？
    (b) 该变换的目的可能是什么？

17. 假定我们对比率属性 x 使用平方根变换，得到一个新属性 x^*。作为分析的一部分，你识别出区间(a, b)，在该区间内，x^*与另一个属性 y 具有线性关系。
    (a) 换算成 x，(a, b)的对应区间是什么？
    (b) 给出 y 关联 x 的方程。
18. 本习题比较和对比某些相似性和距离度量。
    (a) 对于二元数据，L_1距离对应于汉明距离，即两个二元向量不同的二进位数。Jaccard 相似度是两个二元向量之间相似性的度量。计算如下两个二元向量之间的汉明距离和 Jaccard 相似度。
    $\mathbf{x}$ = 0101010001
    $\mathbf{y}$ = 0100011000
    (b) Jaccard 相似度与汉明距离哪种方法更类似于简单匹配系数，哪种方法更类似于余弦度量？解释你的结论。（注意：汉明度量是距离，而其他三种度量是相似性，但是不要被这一点所迷惑。）
    (c) 假定你正在根据包含共同基因的个数比较两个不同物种的有机体的相似性。你认为哪种度量更适合用来比较构成两个有机体的遗传基因，是汉明还是 Jaccard？解释你的结论。（假定每种动物用一个二元向量表示，其中如果一个基因出现在有机体中，则对应的属性取值 1，否则取值 0。）
    (d) 如果你想比较构成相同物种的两个有机体的遗传基因（例如，两个人），你会使用汉明距离，Jaccard 系数，还是一种不同的相似性或距离度量？解释原因。（注意，两个人的相同基因超过 99.9%。）
19. 对于下面的向量 $\mathbf{x}$ 和 $\mathbf{y}$，计算指定的相似性或距离度量。
    (a) $\mathbf{x} = (1, 1, 1, 1)$, $\mathbf{y} = (2, 2, 2, 2)$ 余弦、相关、欧几里得。
    (b) $\mathbf{x} = (0, 1, 0, 1)$, $\mathbf{y} = (1, 0, 1, 0)$ 余弦、相关、欧几里得、Jaccard。
    (c) $\mathbf{x} = (0, -1, 0, 1)$, $\mathbf{y} = (1, 0, -1, 0)$ 余弦、相关、欧几里得。
    (d) $\mathbf{x} = (1, 1, 0, 1, 0, 1)$, $\mathbf{y} = (1, 1, 1, 0, 0, 1)$ 余弦、相关、Jaccard。
    (e) $\mathbf{x} = (2, -1, 0, 2, 0, -3)$, $\mathbf{y} = (-1, 1, -1, 0, 0, -1)$ 余弦、相关。
20. 这里，进一步考察余弦度量和相关性度量。
    (a) 对于余弦度量，可能的值域是什么？
    (b) 如果两个对象的余弦度量为 1，它们相等吗？解释原因。
    (c) 如果余弦度量与相关性度量有关系的话，有何关系？（提示：在余弦和相关性相同或不同情况下，考虑诸如均值、标准差等统计量。）
    (d) 图 2-20a 显示 100 000 个随机生成的点的余弦度量与欧几里得距离之间的关系，这些点已经规范化，L_2 长度为 1。当向量的 L_2 长度为 1 时，关于欧几里得距离与余弦相似性之间的关系，你能得出什么样的一般观测结论？
    (e) 图 2-20b 显示 100 000 个随机生成的点的相关性度量与欧几里得距离之间的关系，这些点已经标准化，具有均值 0 和标准差 1。当向量已经标准化，具有均值 0 和标准差 1 时，关于欧几里得距离与相关性之间的关系，你能得出什么样的一般观测结论？
    (f) 当每个数据对象的 L_2 长度为 1 时，推导余弦相似度与欧几里得距离之间的数学关

系。

(g) 当每个数据点通过减去均值并除以其标准差标准化时，推导相似度与欧几里得距离之间的数学关系。



(a) 欧几里得距离与余弦度量之间的关系　　(b) 欧几里得距离与相关性之间的关系

图 2-20　习题 20 的图形

21. 证明下式给出的集合差度量满足 2.4.3 节的度量公理：

$$d(A, B) = size(A - B) + size(B - A) \tag{2-19}$$

其中，A 和 B 是集合，$A - B$ 是集合差。

22. 讨论如何将相关值从区间[−1, 1]映射到区间[0, 1]。注意，你所使用的变换类型可能取决于你的应用。因此，考虑两种应用：对时间序列聚类，给定一个时间序列预测另一个的性质。

23. 给定一个在区间[0, 1]取值的相似性度量，描述两种将该相似度变换成区间[0, ∞]中的相异度的方法。

24. 通常，邻近度定义在一对对象之间。
    (a) 阐述两种定义一组对象之间邻近度的方法。
    (b) 如何定义欧几里得空间中两个点集之间的距离？
    (c) 如何定义两个数据对象集之间的邻近度？（除邻近度定义在任意一对对象之间外，对数据对象不做任何假定。）

25. 给定欧几里得空间中一个点集 S，以及 S 中每个点到点 $\mathbf{x}$ 的距离。（$\mathbf{x}$ 是否属于 S 并不重要。）
    (a) 如果目标是发现点 $\mathbf{y}$（$\mathbf{y} \neq \mathbf{x}$）指定距离 ε 内的所有点，解释如何利用三角不等式和已经计算的到 $\mathbf{x}$ 的距离，来减少必需的距离计算数量。提示：三角不等式 $d(\mathbf{x}, \mathbf{z}) \leqslant d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{x})$ 可以写成 $d(\mathbf{x}, \mathbf{y}) \geqslant d(\mathbf{x}, \mathbf{z}) - d(\mathbf{y}, \mathbf{z})$。
    (b) $\mathbf{x}$ 和 $\mathbf{y}$ 之间的距离对距离计算的数量有何影响？
    (c) 假定你可以从原来的数据点的集合中发现一个较小的子集 S'，使得数据集中的每个点都至少到 S' 中一个点的距离不超过指定的 ε，并且你还得到了 S' 中每对点之间的距离矩阵。描述一种技术，使用这些信息，以最少的距离计算量，从数据集中计算到一个指定点距离不超过 β 的所有点的集合。

26. 证明 1 减 Jaccard 相似度是两个数据对象 **x** 和 **y** 之间的一种距离度量，该度量满足 2.4.3 节的度量公理。具体地，$d(\mathbf{x}, \mathbf{y}) = 1 - J(\mathbf{x}, \mathbf{y})$。
27. 证明定义为两个数据向量 **x** 和 **y** 之间夹角的距离度量满足 2.4.3 节的度量公理。具体地，$d(\mathbf{x}, \mathbf{y}) = \arccos(\cos(\mathbf{x}, \mathbf{y}))$。
28. 解释为什么计算两个属性之间的邻近度通常比计算两个对象之间的相似度简单。

第3章

探索数据

第2章讨论了知识发现过程中重要的高层数据问题。本章是**数据探索**导论，对数据进行初步研究，以便更好地理解它的特殊性质。数据探索有助于选择合适的数据预处理和数据分析技术。它甚至可以处理一些通常由数据挖掘解决的问题，例如，有时可以通过对数据进行直观检查来发现模式。此外，数据探索中使用的某些技术（如可视化）可以用于理解和解释数据挖掘结果。

本章包括三个主题：汇总统计、可视化和联机分析处理（OLAP）。汇总统计（如值集合的均值和标准差）和可视化技术（如直方图和散布图）是广泛用于数据探索的标准方法。OLAP是一种新近开发的包含一系列考察多维数组数据的技术。OLAP的分析功能集中在从多维数据数组中创建汇总表的各种方法。OLAP技术包括在不同的维上或不同的属性值上聚集数据，例如，如果给定基于产品、位置和日期记录的销售信息，则可以使用OLAP技术创建按月和按产品类别描述特定地点的销售活动汇总。

本章涵盖的主题与**探测性数据分析**（Exploratory Data Analysis，EDA）有许多重叠，EDA是卓越的统计学家John Tukey于20世纪70年代创建的。像EDA一样，本章特别强调可视化，而与EDA不同的是，本章并不包含诸如聚类分析和异常检测等主题，其原因有二：首先，数据挖掘将描述性数据分析技术本身看作目的，而统计学（EDA由此发源）趋向于将基于假设的检验作为最终目标；其次，聚类分析和异常检测都是很大的领域，需要用整章进行深入讨论。因此，聚类分析将在第8章和第9章给出，而异常检测则在第10章讨论。

3.1 鸢尾花数据集

在下面的讨论中，我们经常提到鸢尾花（Iris）数据集，该数据集可以从加州大学欧文分校（UCI）的机器学习库中得到。鸢尾花数据集包含150种鸢尾花的信息，每50种取自三个鸢尾花种之一：Setosa、Versicolour和Virginica。每个花的特征用下面5种属性描述。

(1) 萼片长度（厘米）。
(2) 萼片宽度（厘米）。
(3) 花瓣长度（厘米）。
(4) 花瓣宽度（厘米）。
(5) 类（Setosa, Versicolour, Virginica）。

花的萼片是花的外部结构，保护花的更脆弱的部分（如花瓣）。在许多花中，萼片是绿的，只有花瓣是鲜艳多彩的，然而，对于鸢尾花，萼片也是鲜艳多彩的。图3-1给出了一种Virginica鸢尾花的图片，鸢尾花的萼片比花瓣大并且下垂，而花瓣向上。

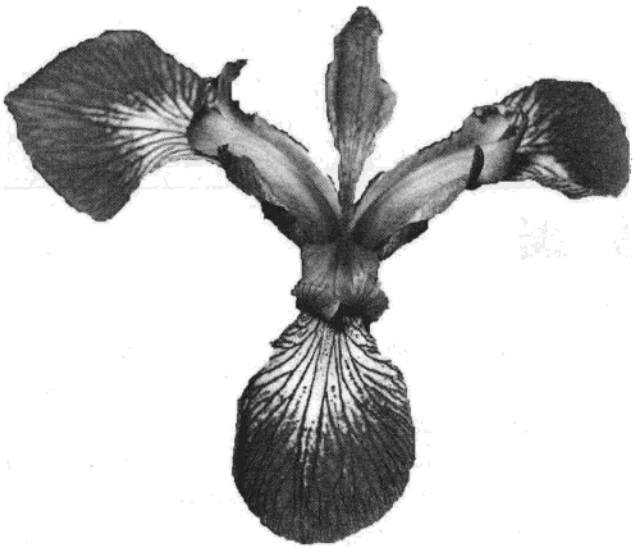


图 3-1　鸢尾花 Virginica 的图片。Robert H. Mohlenbrock @ USDA-NRCS PLANTS Database/USDA NRCS. 1995. 东北湿地植物志：野外办公室植物物种指南。东北国家技术中心，切斯特，宾夕法尼亚州（删除了背景）

3.2　汇总统计

汇总统计（summary statistics）是量化的（如均值和标准差），用单个数或数的小集合捕获可能很大的值集的各种特征。汇总统计的日常例子有家庭平均收入、四年内完成本科学位的学生比例。的确，对于许多人，汇总统计是最常见的统计形式。我们将集中讨论对单个属性值的汇总统计，但是也将简略介绍某些多变元汇总统计。

本节只考虑汇总统计的描述性质。然而，统计学将数据视为源于被各种参数刻画的基本统计过程，而这里讨论的某些汇总统计可以看作是产生数据的基本分布的统计参数的估计。

3.2.1　频率和众数

给定一个无序的、分类的值的集合，为了进一步刻画值的性质，除计算特定数据集中每个值出现的频率外没有多少的事情可做。给定一个在$\{v_1, \cdots, v_i, \cdots, v_k\}$上取值的分类属性 x 和 m 个对象的集合，值 v_i 的频率定义为：

$$\text{frequency}(v_i) = \frac{\text{具有属性值}v_i\text{的对象数}}{m} \tag{3-1}$$

分类属性的**众数**（mode）是具有最高频率的值。

例 3.1　考虑学生的集合。学生具有一个属性年级，可以从集合{一年级，二年级，三年级，四年级}中取值。表 3-1 显示年级属性每个值的学生人数，年级属性的众数是大学一年级，其频率为 0.33，这或许暗示因退学导致的减员或扩招。

表 3-1 一所假想大学中各年级学生人数

年级	人数	频率
一年级	200	0.33
二年级	160	0.27
三年级	130	0.22
四年级	110	0.18

□

分类属性常常（但并非总是）具有少量值，因此这些值的众数和频率可能是令人感兴趣的和有用的。注意，尽管如此，对于鸢尾花数据集和类属性，由于三种类型的花具有相同的频率，因而众数的概念并无意义。

对于连续数据，按照目前的定义，众数通常没有用，因为单个值的出现可能不超过一次，然而，在某些情况下，众数可能提供关于值的性质或关于出现遗漏值的重要信息。例如，以毫米为单位测量，20个人的身高通常不会重复，但是如果以分米为单位测量，则某些人可能具有相同的身高。此外，如果使用唯一的值表示遗漏值，则该值常常表现为众数。

3.2.2 百分位数

对于有序数据，考虑值集的**百分位数**（percentile）更有意义。具体地说，给定一个有序的或连续的属性 x 和 0 与 100 之间的数 p，第 p 个百分位数 x_p 是一个 x 值，使得 x 的 p%的观测值小于 x_p。例如，第 50 个百分位数是值 $x_{50\%}$，使得 x 的所有值的 50%小于 $x_{50\%}$。表 3-2 显示鸢尾花数据集的四个定量属性的百分位数。

表 3-2 萼片长度、萼片宽度、花瓣长度和花瓣宽度的百分位数（所有的值都以厘米为单位）

百分位数	萼片长度	萼片宽度	花瓣长度	花瓣宽度
0	4.3	2.0	1.0	0.1
10	4.8	2.5	1.4	0.2
20	5.0	2.7	1.5	0.2
30	5.2	2.8	1.7	0.4
40	5.6	3.0	3.9	1.2
50	5.8	3.0	4.4	1.3
60	6.1	3.1	4.6	1.5
70	6.3	3.2	5.0	1.8
80	6.6	3.4	5.4	1.9
90	6.9	3.6	5.8	2.2
100	7.9	4.4	6.9	2.5

例 3.2 从 1 到 10 的整数的百分位数 $x_{0\%}, x_{10\%}, \cdots, x_{90\%}, x_{100\%}$依次为：1.0, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.0。按照惯例，$\min(x) = x_{0\%}$，而 $\max(x) = x_{100\%}$。 □

3.2.3 位置度量：均值和中位数

对于连续数据，两个使用最广泛的汇总统计是**均值**（mean）和**中位数**（median），它们是值集位置的度量。考虑 m 个对象的集合和属性 x，设$\{x_1, \cdots, x_m\}$是这 m 个对象的 x 属性值（在这个具体的例子中，这些值是 m 个儿童的身高），设$\{x_{(1)}, \cdots, x_{(m)}\}$代表以非递减序排序后的 x 值，这样，$x_{(1)} = \min(x)$，而 $x_{(m)} = \max(x)$，于是均值和中位数定义如下：

$$\text{mean}(x) = \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3-2}$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{如果}m\text{是奇数，即}m = 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{如果}m\text{是偶数，即}m = 2r \end{cases} \tag{3-3}$$

概括地说，如果有奇数个值，则中位数是中间值；如果有偶数个值，则中位数是中间两个值的平均值。这样，对于 7 个值，中位数是 $x_{(4)}$，而对于 10 个值，中位数是 $\frac{1}{2}(x_{(5)} + x_{(6)})$。

尽管有时将均值解释为值集的中间，但是仅当值以对称方式分布时，才是对的。如果值的分布是倾斜的，则中位数是中间的一个更好的指示符。此外，均值对于离群值很敏感；对于包含离群值的数据，中位数可以再次更稳健地提供值集中间的估计。

为了克服传统均值定义的问题，有时使用**截断均值**（trimmed mean）概念。指定 0 和 100 之间的百分位数 p，丢弃高端和低端$(p/2)$%的数据，然后用常规的方法计算均值，所得的结果即是截断均值。中位数是 $p = 100$%时的截断均值，而标准均值是对应于 $p = 0$%的截断均值。

例 3.3　考虑值集{1, 2, 3, 4, 5, 90}。这些值的均值是 17.5，而中位数是 3.5，$p = 40$%时的截断均值也是 3.5。 □

例 3.4　鸢尾花数据集的四个定量属性的均值、中位数和截断均值（$p = 20$%）在表 3-3 给出。除属性花瓣长度外，其余三个属性的三个位置度量具有相似的值。 □

表 3-3　萼片长度、萼片宽度、花瓣长度和花瓣宽度的均值、中位数和截断均值（所有值都以厘米为单位）

度量	萼片长度	萼片宽度	花瓣长度	花瓣宽度
均值	5.84	3.05	3.76	1.20
中位数	5.80	3.00	4.35	1.30
截断均值（20%）	5.79	3.02	3.72	1.12

3.2.4　散布度量：极差和方差

连续数据的另一组常用的汇总统计是值集的弥散或散布度量。这种度量表明属性值是否散布很宽，或者是否相对集中在单个点（如均值）附近。

最简单的散布度量是**极差**（range）。给定属性 x，它具有 m 个值$\{x_1, \cdots, x_m\}$，x 的极差定义为：

$$\text{range}(x) = \max(x) - \min(x) = x_{(m)} - x_{(1)} \tag{3-4}$$

尽管极差标识最大散布，但是如果大部分值都集中在一个较窄的范围内，并且更极端的值的个数相对较少，则可能会引起误解。因此，作为散布的度量，**方差**（variance）更可取。通常，属性 x 的（观测）值的方差记作 s_x^2，并在下面定义。**标准差**（standard deviation）是方差的平方根，记作 s_x，它与 x 具有相同的单位。

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{x})^2 \tag{3-5}$$

均值可能被离群值扭曲，并且由于方差用均值计算，因此它也对离群值敏感。确实，方差对离群值特别敏感，因为它使用均值与其他值的差的平方。这样常常需要使用比值集散布更稳健的

估计。下面是三种这样的度量的定义：**绝对平均偏差**（absolute average deviation，AAD）、**中位数绝对偏差**（median absolute deviation，MAD）和**四分位数极差**（interquartile range，IQR）。表 3-4 显示鸢尾花数据集的这些度量。

$$\mathrm{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \bar{x}| \tag{3-6}$$

$$\mathrm{MAD}(x) = median(\{|x_1 - \bar{x}|, \cdots, |x_m - \bar{x}|\}) \tag{3-7}$$

$$\text{interquartile range}\ (x) = x_{75\%} - x_{25\%} \tag{3-8}$$

表 3-4 萼片长度、萼片宽度、花瓣长度和花瓣宽度的极差、标准差（std）、绝对平均偏差（AAD）、中位绝对偏差（MAD）和中间四分位数极差（IQR）（所有值都以厘米为单位）

度量	萼片长度	萼片宽度	花瓣长度	花瓣宽度
极差	3.6	2.4	5.9	2.4
std	0.8	0.4	1.8	0.8
AAD	0.7	0.3	1.6	0.6
MAD	0.7	0.3	1.2	0.7
IQR	1.3	0.5	3.5	1.5

3.2.5 多元汇总统计

包含多个属性的数据（多元数据）的位置度量可以通过分别计算每个属性的均值或中位数得到。这样，给定一个数据集，数据对象的均值 $\bar{\mathbf{x}}$ 由

$$\bar{\mathbf{x}} = (\overline{x_1}, \cdots, \overline{x_n}) \tag{3-9}$$

给出，其中 $\overline{x_i}$ 是第 i 个属性 x_i 的均值。

对于多元数据，每个属性的散布可以独立于其他属性，使用 3.2.4 节介绍的方法计算。然而，对于具有连续变量的数据，数据的散布更多地用**协方差矩阵**（covariance matrix）$\mathbf{S}$ 表示，其中，$\mathbf{S}$ 的第 ij 个元素 s_{ij} 是数据的第 i 个和第 j 个属性的协方差。这样，如果 x_i 和 x_j 分别是第 i 个和第 j 个属性，则

$$s_{ij} = \mathrm{covariance}(x_i, x_j) \tag{3-10}$$

而 $\mathrm{covariance}(x_i, x_j)$ 由

$$\mathrm{covariance}(x_i, x_j) = \frac{1}{m-1}\sum_{k=1}^{m}(x_{ki} - \overline{x_i})(x_{kj} - \overline{x_j}) \tag{3-11}$$

给出，其中 x_{ki} 和 x_{kj} 分别是第 k 个对象的第 i 和第 j 个属性的值。注意，$\mathrm{covariance}(x_i, x_i) = \mathrm{variance}(x_i)$。这样，协方差矩阵的对角线上是属性的方差。

两个属性的协方差是两个属性一起变化并依赖于变量大小的度量。协方差的值接近于 0 表明两个变量不具有（线性）关系，但是不能仅靠观察协方差的值来确定两个变量之间的关联程度。因为两个属性的相关性直接指出两个属性（线性）相关的程度，对于数据探索，相关性比协方差更可取。（另见 2.4.5 节关于相关的讨论。）**相关矩阵**（correlation matrix）$\mathbf{R}$ 的第 ij 个元素是数据的第 i 个和第 j 个属性之间的相关性。如果 x_i 和 x_j 分别是第 i 个和第 j 个属性，则

$$r_{ij} = \text{correlation}(x_i, x_j) = \frac{\text{covariance}(x_i, x_j)}{s_i s_j} \tag{3-12}$$

其中，s_i和s_j分别是x_i和x_j的方差。**R**的对角线上的元素是correlation $(x_i, x_i) = 1$，而其他元素在–1和1之间。考虑包含每对对象而不是每对属性之间相关性的相关矩阵也是有用的。

3.2.6　汇总数据的其他方法

当然，还有其他类型的汇总统计，例如，值集的**倾斜度**（skewness）度量值对称地分布在均值附近的程度。另外还有一些其他数据特征，很难定量地度量，例如，值的分布是否是多模态的（multimodal），即数据具有多个“肿块”，大部分值集中在那里。然而，在许多情况下，理解关于属性值如何分布的更复杂、更微妙的方面，最有效的方法是通过直方图观察这些值。（直方图在下一节讨论。）

3.3　可视化

数据可视化是指以图形或表格的形式显示信息。成功的可视化需要将数据（信息）转换成可视的形式，以便能够借此分析或报告数据的特征和数据项或属性之间的关系。可视化的目标是形成可视化信息的人工解释和信息的意境模型。

在日常生活中，可视化技术（如图和表等）常常是优先选择的方法，用来解释气象、经济和政治选举的结果。同样，尽管在大多数技术学科（包括数据挖掘）中通常强调算法或数学方法，但是可视化技术也能在数据分析方面起关键性作用。事实上，有时将可视化技术在数据挖掘方面的应用称作**可视化数据挖掘**（visual data mining）。

3.3.1　可视化的动机

使用可视化技术的首要动机是人们能够快速吸取大量可视化信息，并发现其中的模式。考虑图3-2，它以摄氏度为单位显示1982年7月的海洋表面温度（SST）。这张图汇总大约250 000个数据，并且一目了然。例如，在这张图上可以很容易地看出，海洋温度在赤道最高，而在两极最低。

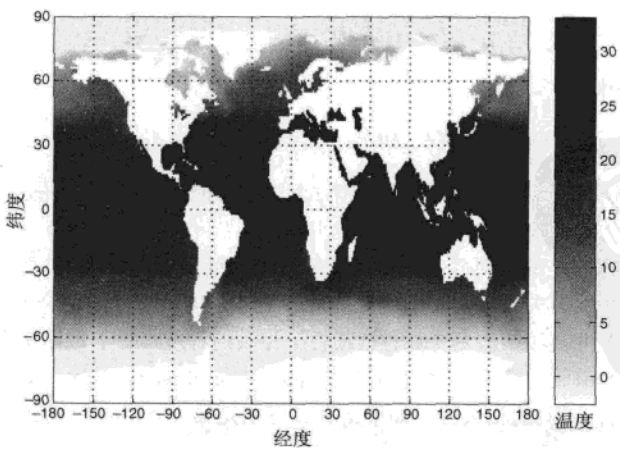


图3-2　1982年7月的海洋表面温度（SST）

可视化的另一个动机是利用“锁在人脑袋中”的领域知识。尽管使用领域知识是数据挖掘的一项重要任务，但是在统计学或算法工具中通常无法充分地利用这种知识，或者是不可能利用的。在某些情况下，可以使用非可视化工具进行分析，然后用可视化的方式提供结果，由领域专家进行评估。在其他情况下，让领域专家检查可视化数据可能是发现有意义的模式的最佳方法，因为利用领域知识，通常可以快速排除许多无意义的模式，并且直接聚焦到重要的模式上。

3.3.2 一般概念

本节考察一些与可视化有关的一般概念，特别是考察将数据和它的属性可视化的一般方法。本节将简略提及一些可视化技术，并在其后讨论具体的方法时更详细地介绍。我们假定读者熟悉线图、条形图和散布图。

1. 表示：将数据映射到图形元素

可视化的第一步是将信息映射成可视形式，即将信息中的对象、属性和联系映射成可视的对象、属性和联系。也就是说，数据对象、它们的属性，以及数据对象之间的联系要转换成诸如点、线、形状和颜色等图形元素。

对象通常用三种方法表示。首先，如果只考虑对象的单个分类属性，则通常根据该属性的值将对象聚成类，并且把这些类作为表的项或屏幕的区域显示。（本章后面给出的例子是交叉表和条形统计图表。）其次，如果对象具有多个属性，则可以将对象显示为表的一行（或列），或显示为图的一条线。最后，对象常常解释为二维或三维空间中的点，其中点可能用几何图形表示，如圆圈、十字叉或方框。

对于属性，其表示取决于属性的类型，即取决于属性是标称的、序数的还是连续的（区间的或比率的）。序数的和连续的属性可以映射成连续的、有序的图形特征，如在 x，y 或 z 轴上的位置，亮度，颜色，或尺寸（直径、宽度或高度等）。对于分类属性，每个类别可以映射到不同的位置、颜色、形状、方位、修饰物或表的列。然而，对于标称属性，由于它的值是无序的，因此在使用具有与其值相关的固有序的图形特征（如颜色、位置等）时，就需要特别小心。换言之，用来表示序数值的图形元素通常有序，但是标称值没有序。

通过图形元素表示的关系或者是显式的，或者是隐式的。对于图形数据，通常使用标准的图形表示——点和点间的连线。如果点（数据对象）或连线（关系）具有自己的属性或特性，则这些属性也可以图示。例如，如果点是城市，连线是公路，则点的直径可以表示人口，而连线的宽度可以表示交通流量。

通常将对象和属性映射到图形元素，隐含地将数据中的联系映射到图形对象之间的联系。例如，如果数据对象代表具有位置的物理对象（如城市），则对应于数据对象的图形对象的相对位置趋向于自然地保持对象的实际相对位置。同样，如果两个或三个连续属性取作点的坐标值，则结果图通常呈现属性和数据点的联系，因为看上去靠近的数据点具有相似的属性值。

一般地，很难确保将对象和属性的映射表示成图形元素之间易于观察的联系。的确，这是可视化的最主要难点之一。在任意给定的数据集中，有许多蕴涵的联系，因此可视化的主要难点是选择一种技术，让关注的联系易于观察。

2. 安排

如前所述，对于好的可视化来说，正确选择对象和属性的可视化表示是基本的要求。在可视化显示中，项的安排也是至关重要的。我们用两个例子解释这一点。

例 3.5　本例解释重新安排表中数据的重要性。在表 3-5 中，显示具有 6 个二元属性的 9 个数据对象，对象和属性之间没有明显的联系，至少乍一看如此。然而，重新排列该表的行和列后，如表 3-6 所示，则可以清楚地看出表中只有两类对象——一类的前三个属性取 1，而另一类的后三个属性取 1。 □

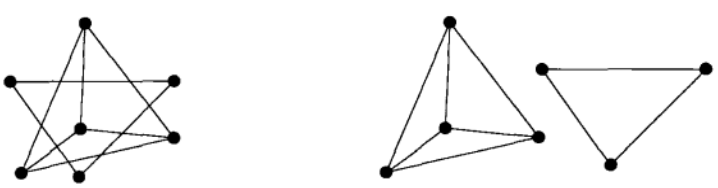
表 3-5　具有 6 个二元属性（列）的 9 个对象（行）的表

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

表 3-6　具有 6 个二元属性（列）的 9 个对象（行）的表，排列后使得行和列的联系明朗

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

例 3.6　考虑图 3-3a，该图显示一个图的可视化。如果将连通子图分开，如图 3-3b 所示，则结点和图之间的联系就变得更加简单易懂。 □



(a) 图的原视图　　(b) 图的连通子图分开后的视图

图 3-3　图的两种可视化

3. 选择

可视化的另一个关键概念是**选择**（selection），即删除或不突出某些对象和属性。具体说来，尽管只具有少数维的数据对象通常可以使用直截了当的方法映射成二维或三维图形表示，但是还没有令人完全满意和一般的方法表示具有许多属性的数据。同样，如果有很多数据对象，则可视化所有对象可能导致显示过于拥挤。如果有许多属性和许多对象，则情况会更加复杂。

处理很多属性的最常用方法是使用属性子集（通常是两个属性）。如果维度不太高，则可以构造双变量（双属性）图矩阵用于联合观察。（图 3-16 显示鸢尾花数据集属性对的散布图矩阵。）或者说，可视化程序可以自动地显示一系列二维图，其中次序由用户或根据某种预定义的策略控制，让可视化二维图的集族提供数据的更完全的视图。

选择一对（或少数）属性的技术是一类维归约，并且有许多更复杂的维归约可以使用，如主成分分析（PCA）。

当数据点的个数很多（例如超过数百个）或者数据的极差很大时，充分显示每个对象的信息是困难的，有些数据点可能遮掩其他数据点，或者数据对象可能占据不了足够多的像素来清楚地

显示其特征。例如，如果只有一个像素可用于显示，则对象的形状不能用于对象特性编码。在这些情况下，或者通过放大数据的特定区域，或者通过选取数据点样本，能够删除某些对象是有用的。

3.3.3 技术

可视化技术对于分析的数据类型通常是专用性的。的确，新的可视化技术和方法，以及已有方法的变形，正在不断地创建，以应对新的数据类型和可视化任务。

尽管可视化具有专门性和特殊性，但是仍然有一般性方法可对可视化技术进行分类。一种分类是基于所涉及的属性个数（1、2、3 或多），或者基于数据是否具有某种特殊的性质（如层次结构或图结构）。可视化方法也可以根据所涉及的属性类型分类。另一种分类是根据应用类型：科学的、统计学的或信息学的可视化。下面的讨论将使用三种类型：少量属性的可视化，具有时间和/或空间属性的数据可视化，以及具有大量属性的数据可视化。

这里讨论的大部分可视化技术都可以在一些数学和统计学软件包中找到，其中一些是免费的。万维网上还有大量数据集免费提供。建议读者在阅读下面各段时，试试这些可视化技术。

1. 少量属性的可视化

本段考察用于具有少量属性的可视化数据的技术。有些技术（如直方图）可以显示单个属性观测值分布，其他技术（如散布图）旨在显示两个属性值之间的关系。

茎叶图 茎叶图（stem and leaf plot）可以用来观测一维整型或连续数据的分布。（开始，我们假定数据是整型的，然后解释如何将茎叶图用于连续数据。）对于最简单的一类茎叶图，将值分组，其中每组包含的值除最后一位数字外相同。每个组成为茎，而组中的最后一位数字成为叶。因此，如果值是两位整数（例如，35、36、42 和 51），则茎是高位数字（例如，3、4 和 5），而叶是低位数字（如 1、2、5 和 6）。通过垂直绘制茎，水平绘制叶，可以提供数据分布的可视表示。

例 3.7 图 3-4 给出的数据取自鸢尾花数据集，是以厘米为单位的萼片长度（乘以 10，取整数值）。为方便起见，值已经排序。

这个数据的茎叶图显示在图 3-5 中。图 3-4 中的每个数先根据它的十位数字放到一个垂直的组 4，5，6 或 7 中，然后将它的最后一位数字放到冒号右边。通常，特别是当数据量很大时，需要将茎分裂。例如，不是将十位数字为 4 的所有值都放在一个“桶”中，而是将茎 4 重复两次：40-44 的所有值放在对应于第一个茎的桶中，而 45~49 的所有值放在对应于第二个茎的桶中。这种方法显示在图 3-6 的茎叶图中。其他变形也是可能的。 □

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

图 3-4 鸢尾花数据集中的萼片长度数据

```
4 :  34444566667788888999999
5 :  000000000011111111122223444444555555566666677777777888888999
6 :  000000111111222233333333444444455555667777777889999
7 :  0122234677779
```

图 3-5　鸢尾花数据集中的萼片长度的茎叶图

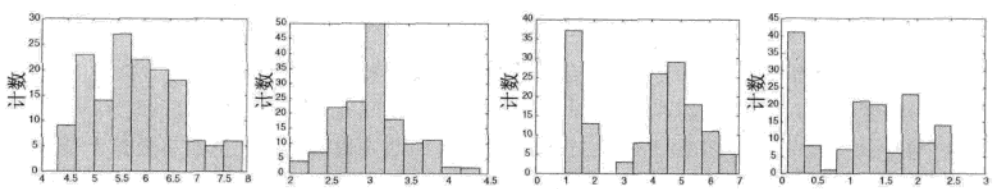
```
4 :  3444
4 :  566667788888999999
5 :  000000000011111111122223444444
5 :  55555556666667777777888888999
6 :  000000111111222233333333444444
6 :  5555566777777778889999
7 :  0122234
7 :  677779
```

图 3-6　鸢尾花数据集中对应于数字的桶分裂后的萼片长度的茎叶图

直方图　茎叶图是一种类型的**直方图**（histogram）。该图通过将可能的值分散到箱中，并显示落入每个箱中的对象数，显示属性值的分布。对于分类属性，每个值在一个箱中。如果值过多，则使用某种方法将值合并。对于连续属性，将值域划分成箱（通常是等宽的，但不必是等宽的），并对每个箱中的值计数。

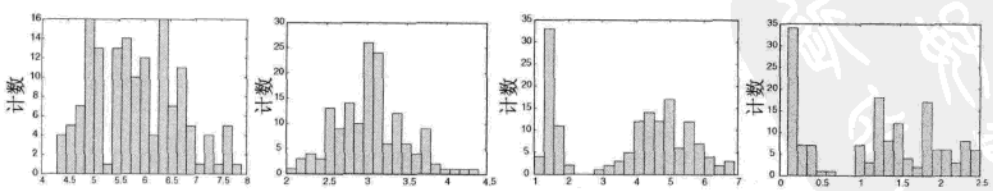
一旦有了每个箱的计数，就可以构造**条形图**（bar plot），即每个箱用一个条形表示，并且每个条形的面积正比于落在对应区间的值（对象）的个数。如果所有的区间都是等宽的，则所有的条形的宽度相同，并且条形的高度正比于落在对应箱中值的个数。

例 3.8　图 3-7 显示萼片长度、萼片宽度、花瓣长度和花瓣宽度的直方图（10 个箱）。由于直方图的形状依赖于箱的个数，同一数据但具有 20 个箱的直方图显示在图 3-8 中。　□



(a) 萼片长度　(b) 萼片宽度　(c) 花瓣长度　(d) 花瓣宽度

图 3-7　四个鸢尾花属性的直方图（10 个箱）



(a) 萼片长度　(b) 萼片宽度　(c) 花瓣长度　(d) 花瓣宽度

图 3-8　四个鸢尾花属性的直方图（20 个箱）

直方图有一些变形。**相对频率直方图**（relative frequency histogram）用相对频率取代计数，

然而，这只是一种 y 轴尺度的变化，直方图的形状并不改变。另一种常见的变形是 **Pareto 直方图**（Pareto histogram），它专门针对无序的分类数据，Pareto 直方图与普通直方图一样，只是分类按计数排序，让计数从左到右递减。

二维直方图 二维直方图（two-dimensional histogram）也是一种类型的直方图。它将每个属性划分成区间，而两个区间集定义值的二维长方体。

例 3.9 图 3-9 显示花瓣长度和花瓣宽度的二维直方图。由于每个属性划分成 3 个箱，因此有 9 个矩形二维箱。每个长方体条的高度指示落入箱中的对象（在此情况下是花）的个数。大部分花落入 3 个沿着对角线的箱中。查看一维分布不可能看到这种情况。 □

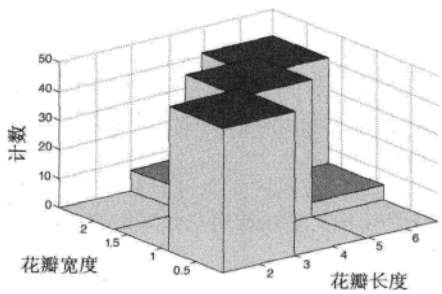


图 3-9 鸢尾花数据集花瓣长度和花瓣宽度的二维直方图

尽管二维直方图可以用来观察关于两个属性的值如何同时出现的有趣问题，但是观察它们比较困难。例如，不难想象这样一种情况，某些柱体被其他柱体遮掩。

盒状图 盒状图（box plot）是另一种显示一维数值属性值分布的方法。图 3-10 显示萼片长度的加标记的盒状图。盒的下端和上端分别指示第 25 和第 75 个百分位数，而盒中的线指示第 50 个百分位数的值，底部和顶部的尾线分别指示第 10 和第 90 个百分位数，离群值用“+”显示。盒状图相对紧凑，因此可以将许多盒状图放在一个图中。还可以使用占据较少空间的盒状图的简化版本。

例 3.10 鸢尾花数据集前 4 个属性的盒状图显示在图 3-11 中。也可以使用盒状图来比较不同对象类之间的属性如何变化，如图 3-12 所示。 □

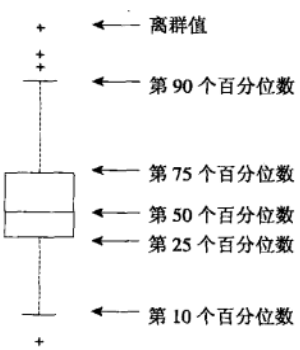


图 3-10 萼片长度盒状图描述

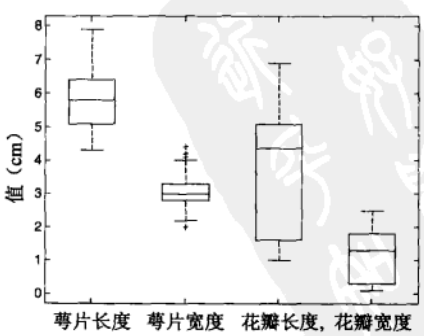
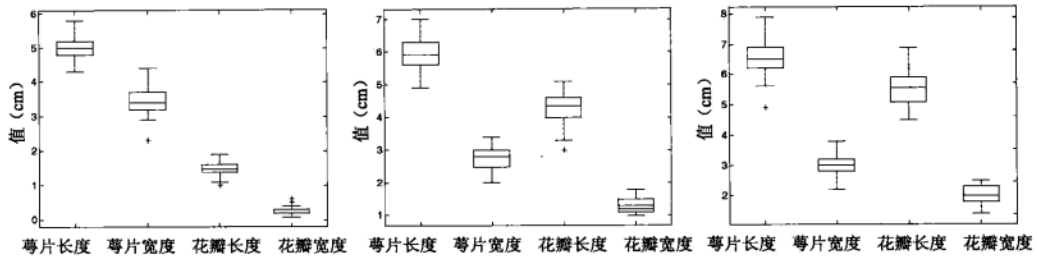


图 3-11 鸢尾花属性的盒状图



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

图 3-12　鸢尾花种类的盒状图

饼图　饼图（pie chart）类似于直方图，但通常用于具有相对较少的值的分类属性。饼图使用圆的相对面积显示不同值的相对频率，而不是像直方图那样使用条形的面积或高度。尽管饼图在通俗文章中很常见，但是它们在技术性出版物中并不常用，因为相对面积的大小很难确定。在技术方面，直方图更可取。

例 3.11　图 3-13 给出了一个饼图，显示鸢尾花数据集的鸢尾花种类的分布。在该例中，三种类型的花都具有相同的频率。　□

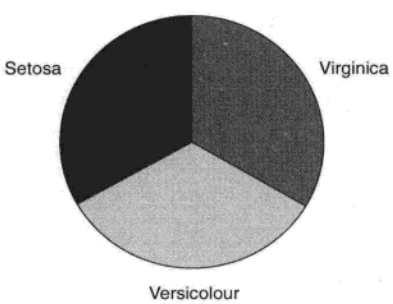
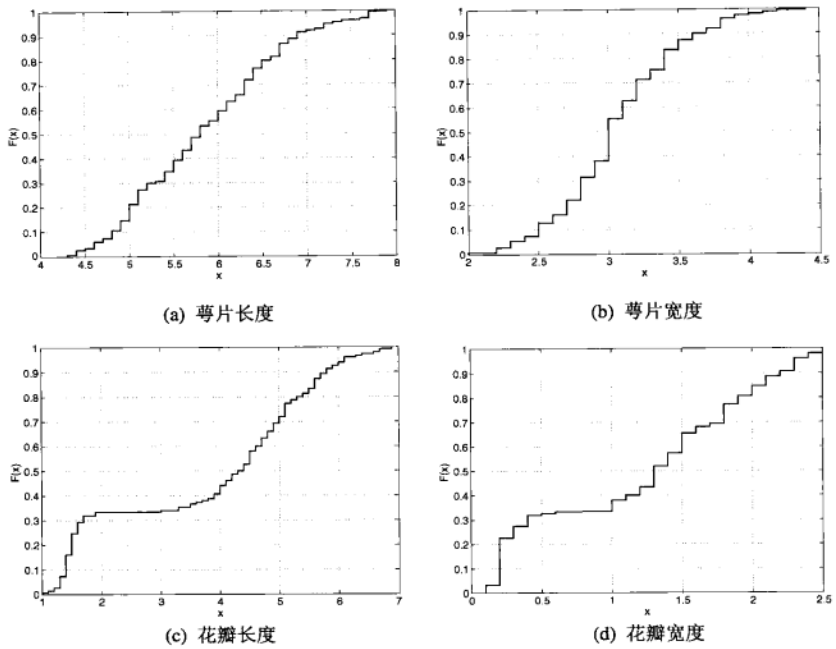


图 3-13　鸢尾花的类型分布

百分位数图和经验累计分布函数　一种更定量地显示数据分布的图是经验累计分布函数图。尽管这种类型的图听上去可能很复杂，但是概念相当简单。对于统计分布的每个值，一个**累计分布函数**（cumulative distribution function，CDF）显示点小于该值的概率。对于每个观测值，一个**经验累计分布函数**（empirical cumulative distribution function，ECDF）显示小于该值的点的百分比。由于点的个数是有限的，经验累计分布函数是一个阶梯函数。

例 3.12　图 3-14 显示了鸢尾花属性的 ECDF。属性的百分位数提供了类似的信息，图 3-15 显示了表 3-2 中鸢尾花数据集的 4 个连续属性**百分位数图**（percentile plot）。读者应当将这些图与图 3-7 和图 3-8 的直方图进行比较。　□

散布图　大部分人都在某种程度上熟悉散布图，本书也在 2.4.5 节使用过散布图来解释线性相关。散布图使用数据对象两个属性的值作为 x 和 y 坐标值，每个数据对象都作为平面上的一个点绘制（假定属性值是整数或实数）。



(a) 萼片长度 (b) 萼片宽度

(c) 花瓣长度 (d) 花瓣宽度

图 3-14 4 个鸢尾花属性的经验 CDF

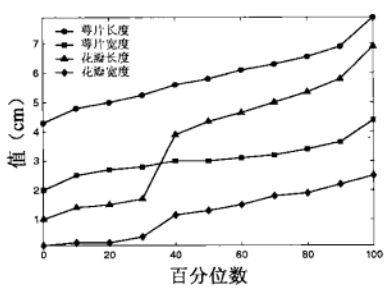


图 3-15 萼片长度、萼片宽度、花瓣长度和花瓣宽度的百分位数图

例 3.13 图 3-16 显示了鸢尾花数据集的每对属性的散布图。不同的鸢尾花种类使用不同的标记表示。属性对的散布图安排在一种称作**散布图矩阵**（scatter plot matrix）的表格形式中，提供了一种有组织的方式，以同时考察许多散布图。 □

散布图有两个主要用途。其一，它们图形化地显示两个属性之间的关系。在 2.4.5 节，我们看到如何使用散布图判定线性相关程度（见图 2-17）。直接使用散布图，或者使用变换后属性的散布图，也可以判定非线性关系。

其二，当类标号给出时，可以使用散布图考察两个属性将类分开的程度。如果可以画一条直线（或一条更复杂的曲线）将两个属性定义的平面分成区域，每个区域包含一个类的大部分对象，则可能基于这对指定的属性构造精确的分类器；否则的话，就需要更多的属性或更复杂的方法建立分类器。在图 3-16 中，许多属性对（例如，花瓣宽度和花瓣长度）都提供了适度的鸢尾花种类分隔。

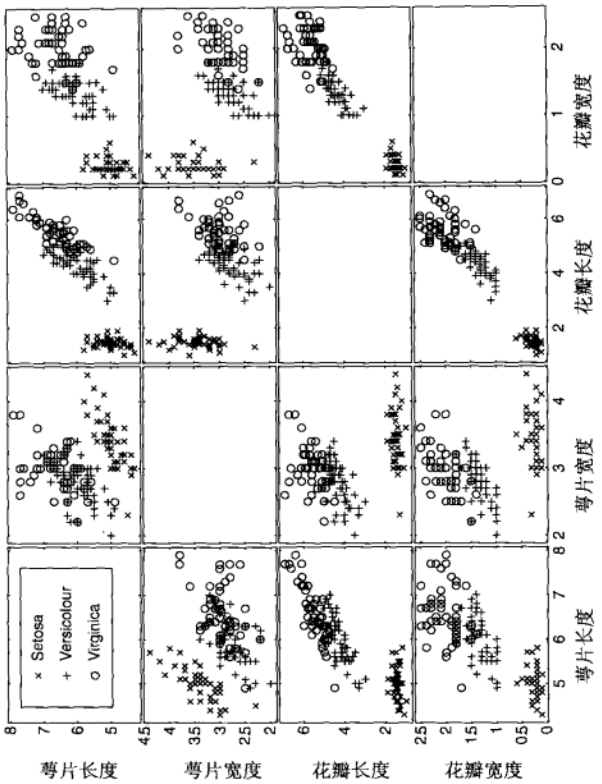


图 3-16　鸢尾花数据集的散布图矩阵

例 3.14　使用散布图显示数据集的三个属性有两种不同的方法。第一种，根据三个，而不是两个属性的值来显示每个对象。图 3-17 显示了鸢尾花数据集的三个属性的三维散布图。第二种，将其中一个属性与标记的某种特性（如大小、颜色或形状）相关联。图 3-18 显示了鸢尾花数据集的三个属性的散布图，其中属性萼片宽度映射到标记的大小。　□

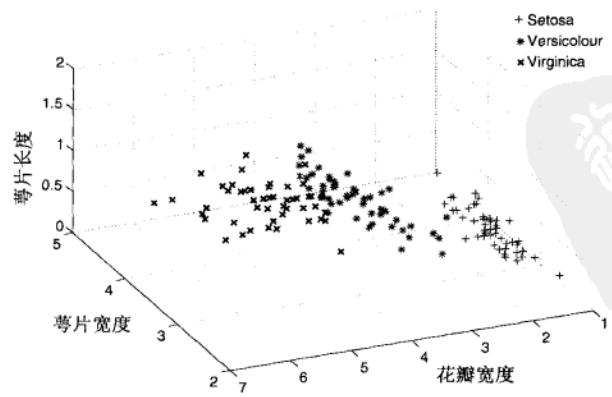


图 3-17　萼片宽度、萼片长度和花瓣宽度的三维散布图

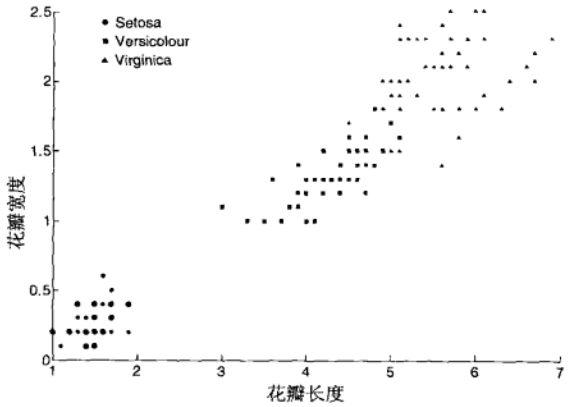


图 3-18 花瓣长度对花瓣宽度的散布图，标记的大小指示萼片宽度

扩展的二维和三维图 如图 3-18 所示，可以扩展成二维或三维图，以便提供一些附加的属性。例如，使用颜色或阴影、大小、形状，散布图可以显示三个附加信息，可以表达五个或六个维。然而，需要小心，随着数据可视表达的复杂性增加，对于解释信息的人就变得更加困难。将六个维的信息放进二维或三维图中没有多少好处，如果做的话也不可能理解。

2. 可视化时间空间数据

数据常常有空间或时间属性，例如，数据可能是在某空间栅格上的观测值的集合，如地球表面上的压力，或物体模拟在各个栅格点上的模拟温度，这些观测值也可以在不同的时间点得到。此外，数据也可能只有一个时间分量，如反映每日股票价格的时间序列数据。

等高线图 对于某些三维数据，两个属性指定平面上的位置，而第三个属性具有连续值，如温度或海拔高度。对于这样的数据，一种有用的可视化工具是**等高线图**（contour plot）。等高线图将平面划分成一些区域，区域中的第三个属性（温度或海拔高度）的值粗略地相等。等高线图的常见例子是显示地面位置海拔高度的等高线图。

例 3.15 图 3-19 显示 1998 年 12 月份平均海洋表面温度（SST）的等高线图，地面温度被随意地设定为 0℃。在许多等高线地图（如图 3-19 中的等高线图）中，将两个区域分开的**等高线**（contour line）用分开区域的值标记。为简明起见，删除了其中一些标记。 □

曲面图 与等高线图一样，**曲面图**（surface plot）使用两个属性表示 x 和 y 坐标，曲面图的第三个属性用来指示高出前两个属性定义的平面的高度。尽管这种图可能是有用的，但是这要求至少某个范围内，对于前两个属性值的所有组合，第三个属性的值都有定义。此外，如果曲面不太规则，除非交互地观察，否则很难看到所有信息。因而，曲面图通常用来描述数学函数，或变化相对光滑的物理曲面。

例 3.16 图 3-20 显示 12 个点的集合周围密度的曲面图。这个例子将在 9.3.3 节进一步讨论。 □

矢量场图 在某些数据中，一个特性可能同时具有值和方向。例如，考虑物质流或随位置改变的密度。在这些情况下，同时显示方向和量的图可能是有用的。这种类型的图称作**矢量图**（vector plot）。

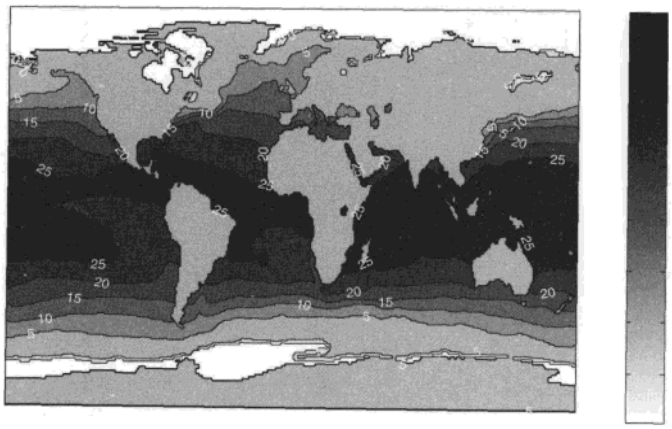
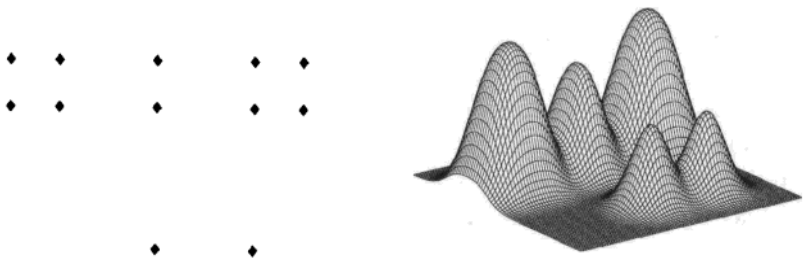


图 3-19　1998 年 12 月份 SST 的等高线图



(a) 12 个点的集合　　(b) 总密度函数——曲面图

图 3-20　12 个点的集合的密度

例 3.17　图 3-21 显示图 3-20b 中两个较小密度尖峰的密度等高线图，并附以密度梯度向量。

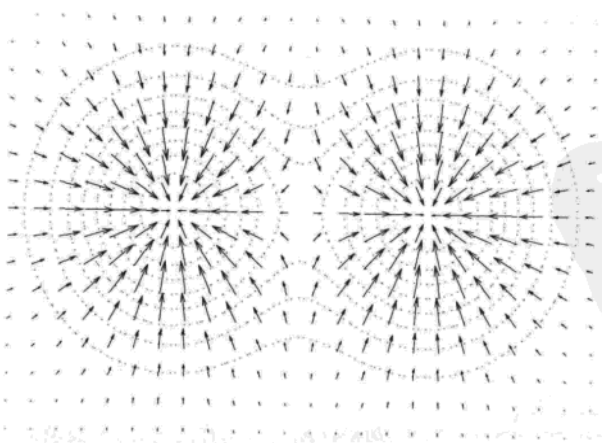


图 3-21　图 3-20 中下端两个密度尖峰的密度梯度（变化）矢量图　□

低维切片 考虑时间空间数据集，它记录不同地点和时间上的某种量，如温度或气压。这样的数据有四个维，不容易用迄今所介绍的图来显示。然而，通过显示一组图，每月一个，可以显示数据的各个“切片”。通过考察特定区域的逐月改变，就可能注意到所出现的变化，包括可能因为季节原因而导致的变化。

例 3.18 该例的基本数据集是从 1982 年到 1999 年、在 2.5°乘 2.5°的经纬度网格上的月平均海平面气压（SLP）。一年 12 个月的气压图显示在图 3-22 中，在这个例子中，我们对 1982 年特定月份的切片感兴趣。更一般地，我们可以考虑沿任意维的数据切片。 □

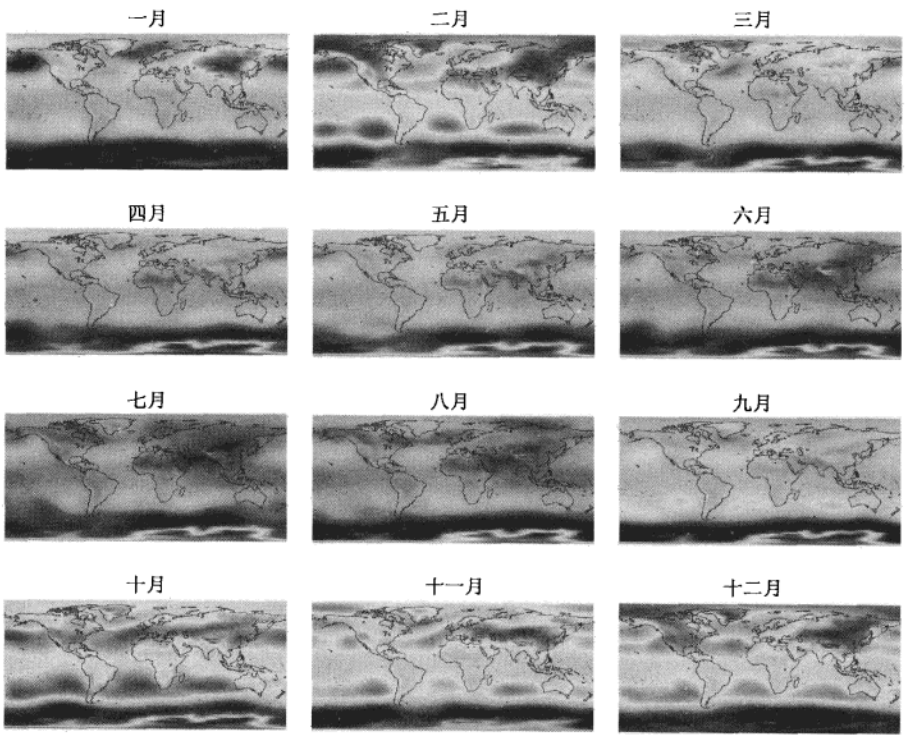


图 3-22 1982 年海平面气压月报图

动画 无论是否涉及时间，处理数据切片的另一种方法是使用动画，其基本思想是显示数据的相继二维切片。人的视觉系统很适合检测视觉变化，并且常常能够注意到可能很难用其他方式检测到的变化。尽管动画具有视觉吸引力，但是一组静止的图（如图 3-22 中的那些）可能更有用，因为这种类型的可视化使得我们可以按任意次序、使用任意多时间来研究这些信息。

3.3.4 可视化高维数据

本节介绍可以显示更多维的可视化技术，使用这些技术所能观察的维数比使用刚刚讨论过的技术观察的更多。然而，即便这些技术也有一些局限性：它们只能显示数据的某些侧面。

矩阵 图像可以看作像素的矩形阵列，其中每个像素用它的颜色和亮度刻画，数据矩阵是值

的矩形阵列，那么，将数据矩阵的每个元素与图像中的一个像素相关联，就可以把数据矩阵看作图像，像素的亮度和颜色由矩阵对应元素的值决定。

在对数据矩阵可视化时，有一些重要的实用性考虑：如果类标号已知，则重新排列数据矩阵的次序，使得某个类的所有对象聚在一起，这是很有用的方法，例如，这可以很容易地检查某个类的所有对象是否在某些属性上具有相似的属性值；如果不同的属性具有不同的值域，则可以对属性标准化，使其均值为 0，标准差为 1，这防止具有最大量值的属性在视觉上左右图形。

例 3.19　图 3-23 显示鸢尾花数据集的标准化数据矩阵，前 50 行代表 Setosa 种类的鸢尾花，接下来的 50 行代表 Versicolour，最后 50 行代表 Virginica。Setosa 花的花瓣宽度和长度远低于平均值，而 Versicolour 花的花瓣宽度和长度在平均值附近，Virginica 花的花瓣宽度和长度高于平均值。□

寻找数据对象集的邻近矩阵图中的结构也是很有用的。当类标号已知时，最好对相似矩阵的行列排序，以便将某个类的所有对象聚在一起。这样可以目视评估每个类的内聚性，与其他类的分离性。

例 3.20　图 3-24 显示鸢尾花数据集的相关矩阵，该矩阵的行和列也已经重新组织，使得特定种类的花在一起。每组内的花相互之间最为相似，但是 Versicolour 和 Virginica 与它们和 Setosa 相比更为类似。□

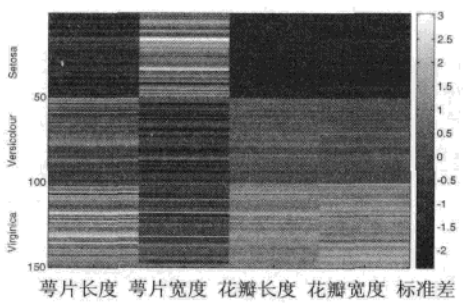


图 3-23　鸢尾花数据矩阵图，其中列已经标准化，均值为 0，标准差为 1

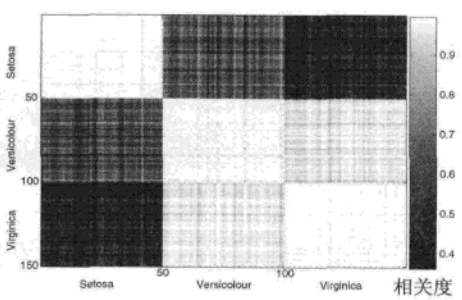


图 3-24　鸢尾花数据集的相关矩阵

如果类标号未知，则多种技术（矩阵重新定序和顺序排列）都可以用来重新安排相似矩阵的行和列，以便一组组高度相似的对象和属性放在一起并可以通过视觉识别。实际上，这是一种简单聚类。关于如何使用邻近矩阵考察数据的聚类结构见 8.5.3 节。

平行坐标系　平行坐标系（parallel coordinates）每个属性一个坐标轴，但是与传统的坐标系不同，平行坐标系不同的坐标轴是平行的，而不是正交的。此外，对象用线而不是用点表示，具体地说，对象每个属性的值映射到与该属性相关联的坐标轴上的点，然后将这些点连接起来形成代表该对象的线。

你可能担心这将产生混乱，然而，在许多情况下，对象趋向于分成少数几组，其中每个组内的点具有类似的属性值，如果这样的话，并且数据对象的数量不太多，则结果平行坐标图可以揭示有意义的模式。

例 3.21 图 3-25 显示鸢尾花数据集 4 个数值属性的平行坐标图。代表不同类的对象的线由其浓淡和类型来区分，这里使用三种不同类型的线——实线、点线和虚线。该平行坐标图表明，三个类关于花瓣宽度和花瓣长度分开得相当好，但关于萼片长度和萼片宽度分开得不太好。图 3-26 是相同数据的另一个平行坐标图，与前一图相比只是坐标轴的次序不同。 □

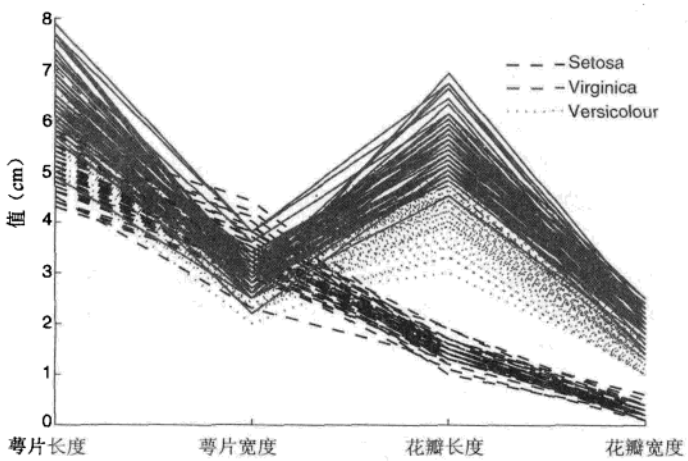


图 3-25 4 个鸢尾花属性的平行坐标图

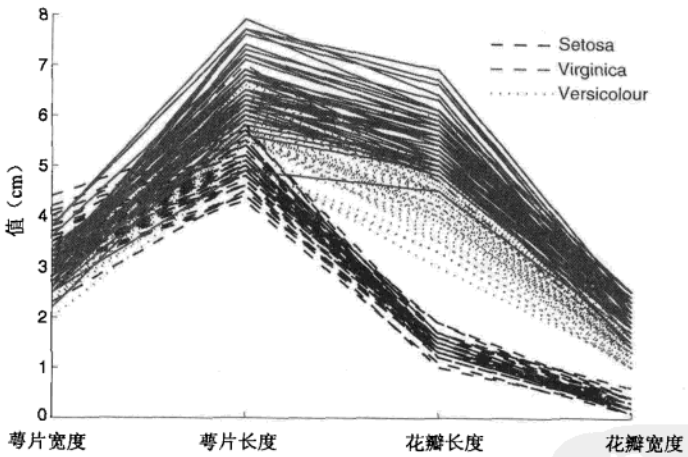


图 3-26 4 个鸢尾花属性的平行坐标图，属性重新定序，以突出组的相似性和相异性

平行坐标图的缺点之一是，在这种图中模式的检测可能取决于坐标轴的序。例如，如果线交叉太多，则图形就变得模糊不清，因此，需要安排坐标轴，以得到具有较少交叉的坐标轴序列。比较图 3-26 和图 3-25，在图 3-26 中，萼片宽度（最混杂的属性）在图的最左边；而在图 3-25 中，该属性在中间。

星形坐标和 Chernoff 脸

显示多维数据的另一种方法是用非文字传达信息的符号——**图示符**（glyph）或**图标**（icon）对对象编码。更明确地说，对象的每个属性映射到图示符的一个特征，使得属性的值决定特征的

准确性质。这样，只需要扫一眼我们就可以辨别两个对象的差异。

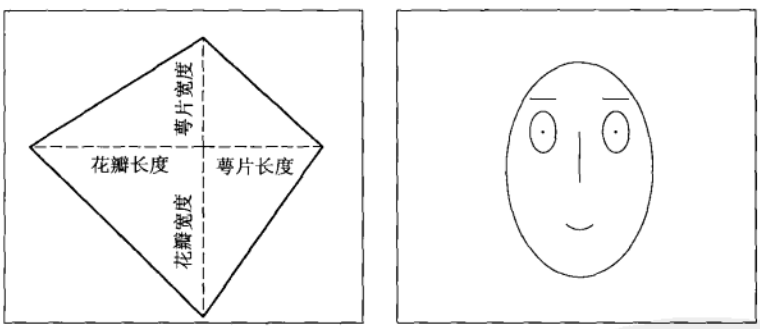
星形坐标（star coordinates）是该方法的一个例子。该技术对每个属性使用一个坐标轴，这些坐标轴从一个中心点向四周辐射，就像车轮的辐条，均匀地散开。通常，所有的属性值都映射到[0,1]区间。

使用如下过程将对象映射到星形坐标系：将对象的每个属性值转换成一个分数，代表它在该属性的最大和最小值之间的距离，把这个分数映射到对应于该属性的坐标上的点，再将每个点用线段连接到相邻坐标轴上的点，形成一个多边形，多边形的大小和形状提供了对象属性值的视觉描述。为了便于解释，每个对象都使用单独的坐标系，换句话说，每个对象映射成一个多边形。星形坐标图的一个例子是鸢尾花 150 号花的星形坐标图，如图 3-27a 所示。

还可以将特征值映射到更为熟悉的对象，如脸。该技术以其创建者 Herman Chernoff 的名字命名为 **Chernoff 脸**（Chernoff face）。在这种技术中，每个属性与脸部的一个特征相关联，而属性的值确定脸部特征的表达方式。这样，随着对应的数据特征值的增加，脸的形状可能拉长。Chernoff 脸的一个例子是鸢尾花 150 号花的 Chernoff 脸，如图 3-27b 所示。

用于将脸映射到 4 个特征的方案在下面列出。脸部的其他特征，如眼间宽度和口的长度，是给定的省缺值。

数据特征	面部特征
萼片长度	脸部大小
萼片宽度	前额/颚相对弧长
花瓣长度	前额形状
花瓣宽度	颚的形状



(a) 鸢尾花 150 号花的星形图　　(b) 鸢尾花 150 号花的 Chernoff 脸

图 3-27　鸢尾花数据集鸢尾花 150 号花的星形坐标图和 Chernoff 脸

例3.22　图3-28和图3-29提供了用这两种方法观察多维数据的更多图示，这两个图分别显示取自鸢尾花数据集的15种花的星形图和脸状图，前5种花属于Setosa种类，中间5种属于Versicolour种类，而最后5种属于Virginica种类。□

尽管这些图有很好的视觉效果，但是它们不能很好地伸缩，因此对于许多数据挖掘问题，其应用受到限制。尽管如此，它们仍然可以作为快速比较用其他技术选择的少量对象集的一种手段。

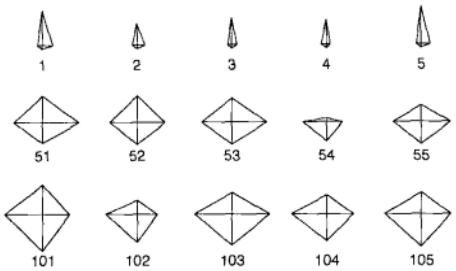


图 3-28 使用星形坐标的 15 种鸢尾花的图形

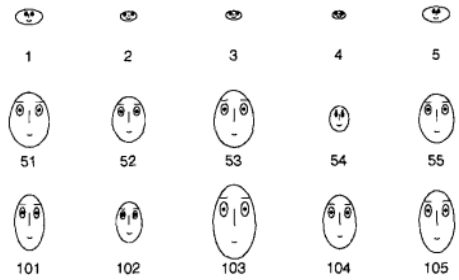


图 3-29 使用 Chernoff 脸的 15 种鸢尾花的图形

3.3.5 注意事项

下面给出可视化注意事项的简短列表，以结束本节关于可视化的讨论。尽管这些指南颇具智慧，但也不能盲目遵循，指南永远不能取代对手头问题的深思熟虑。

ACCENT 原则。下面是 D. A. Burn 提出（经 Michael Friendly 改编）的有效图形显示的 ACCENT 原则。

- **理解**（Apprehension）。正确察觉变量之间关系的能力。图形能够最大化对变量之间关系的理解吗？
- **清晰性**（Clarity）。以目视识别图形中所有元素的能力。最重要的元素或关系在视觉上最突出吗？
- **一致性**（Consistency）。根据与以前的图形的相似性解释图形的能力。元素、符号形状和颜色与以前图形使用的一致吗？
- **有效性**（Efficiency）。用尽可能简单的方法描绘复杂关系的能力。图形元素的使用经济吗？图形容易解释吗？
- **必要性**（Necessity）。对图形和图形元素的需要。与其他替代方法（表、文本）相比，图形是提供数据的更有用的形式吗？为表示关系，所有的图形元素都是必要的吗？
- **真实性**（Truthfulness）。通过图形元素相对于隐式或显式尺度的大小，确定图形元素所代表的真实值的能力。图形元素可以准确地定位和定标吗？

Tufte 指南。Edward R. Tufte 列举了如下图形的优点（graphical excellence）原则。

- 图形的优点是感兴趣的（物质的、统计的和设计的）数据的良好设计的表示。
- 图形的优点包括与清晰性、精确性和有效性相关的复杂思想。
- 图形的优点是它在最小的空间内、以最少的笔墨、在最短的时间内为观察者提供最多的信息。
- 图形的优点几乎总是多元的。
- 图形的优点需要表述数据的真实性。

3.4 OLAP 和多维数据分析

本节考察来自将数据集看作多维数组的技术和见解。大量数据库系统支持这种观点，特别是 OLAP（联机分析处理）系统。事实上，OLAP 系统的一些术语和能力已经使它进入被数百万人

使用的电子数据表程序。OLAP 系统还非常关注交互式数据分析，并提供可视化数据和产生汇总统计的广泛能力。由于这些原因，我们的多维数据分析方法将基于 OLAP 系统常见的术语和概念。

3.4.1　用多维数组表示鸢尾花数据

大部分数据集都可以用表来表示，其中每一行是一个对象，每一列是一个属性。在许多情况下，也可以将数据看作多维数组。我们通过将鸢尾花数据集表示成多维数组来解释这种方法。

表 3-7 是通过如下方法创建的：离散化花瓣长度和花瓣宽度属性，使它们取值低、中和高，然后统计鸢尾花数据集中具有特定的花瓣宽度、花瓣长度和种类的花的数量。（对于花瓣宽度，类别低、中和高分别对应于区间[0, 0.75), [0.75, 1.75)和[1.75, ∞)；对于花瓣长度，类别低、中和高分别对应于区间[0, 2.5), [2.5, 5)和[5, ∞)。）表中没有显示空组合——不包含任何一种花的组合。

表 3-7　具有花瓣宽度、花瓣长度和种类特定组合的花的数量

花瓣长度	花瓣宽度	种类	计数
低	低	Setosa	46
低	中	Setosa	2
中	低	Setosa	2
中	中	Versicolour	43
中	高	Versicolour	3
中	高	Virginica	3
高	中	Versicolour	2
高	中	Virginica	3
高	高	Versicolour	2
高	高	Virginica	44

该数据可以组织成多维数组，如图 3-30 所示，其中，三个维分别对应于花瓣宽度、花瓣长度和种类。为清晰起见，显示了该数组的三个二维表切片，每个对应于一个种类——见表 3-8、表 3-9 和表 3-10。表 3-7 和图 3-30 包含的信息是相同的，只是，在图 3-30（以及表 3-8、表 3-9 和表 3-10）显示的多维表示中，属性花瓣宽度、花瓣长度和种类的值是数组下标。

重要的是从多维的观点观察数据可以获得深入透彻的了解。表 3-8、表 3-9 和表 3-10 显示，每个鸢尾花种类由花瓣宽度和花瓣长度值的不同组合来刻画，Setosa 花具有较低的宽度和长度，Versicolour 花具有中等的宽度和长度，而 Virginica 花具有较高的宽度和长度。

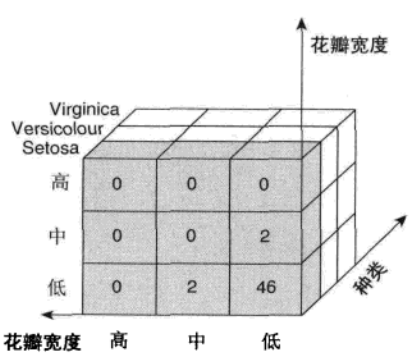


图 3-30　鸢尾花数据集的多维数组表示

表3-8 依照Setosa种类的花的花瓣长度和宽度的交叉表

		宽度		
		低	中	高
长度	低	46	2	0
	中	2	0	0
	高	0	0	0

表 3-9 依照 Versicolour 种类的花的花瓣长度和宽度的交叉表

		宽度		
		低	中	高
长度	低	0	0	0
	中	0	43	3
	高	0	2	2

表 3-10 依照 Virginica 种类的花的花瓣长度和宽度的交叉表

		宽度		
		低	中	高
长度	低	0	0	0
	中	0	0	3
	高	0	3	44

3.4.2 多维数据：一般情况

前一节给出了一个具体的例子，使用多维方法表示和分析一个熟悉的数据集。这里，详细介绍一般的方法。

开始通常使用表的形式表示数据（如表 3-7），这种表称作**事实表**（fact table）。用多维数组表示数据需要两个步骤：维的识别和分析所关注的属性的识别。维是分类属性，或者如前面的例子所示，是转换成分类属性的连续属性。属性值充当对应于该属性的维的数组下标，而属性值的个数是维的大小。在前面的例子中，每个属性有三个可能的值，因此每个维的大小都是 3，并且可以通过 3 个值索引。这产生了 $3 \times 3 \times 3$ 的多维数组。

属性值的每个组合（每个不同的属性一个值）定义了多维数组的一个单元。使用前面的例子解释，如果花瓣长度 = 低，花瓣宽度 = 中，而种类 = Setosa，则标识了一个值为 2 的特定单元。即，数据集中只有两种花具有指定的属性值。注意，表 3-7 中数据集的每一行（对象）对应于多维数组的一个单元。

每个单元的内容代表一个我们在分析时感兴趣的**目标量**（target quantity）（目标变量或属性）的值。在鸢尾花例子中，目标量是其花瓣宽度和长度落入特定范围内的花的个数。目标属性是定量的，因为多维数据分析的关键目标是观察聚集量，如总和或平均值。

下面总结用表形式表示的数据集创建多维数据表示的过程：首先确定用作维的分类属性以及用作分析目标的定量属性，然后将表的每一行（对象）映射到多维数组的一个单元，单元的下标由被选作维的属性的值指定，而单元的值是目标属性的值，假定没有被数据定义的单元的值为 0。

例 3.23 为了进一步解释刚刚讨论的概念，我们给出一个更传统的涉及销售的例子。这个例子的事实表由表 3-11 给出，多维表示的维是产品 ID、地点和日期属性，而目标属性是收入。图 3-31 显示了该数据集的多维表示，这个较大、更复杂的数据集将用来解释多维数据分析的其他概念。 □

表 3-11　不同地点和时间的产品销售收入（单位：美元）

产品 ID	地点	日期	收入
⋮	⋮	⋮	⋮
1	Minneapolis	Oct. 18 2004	$250
1	Chicago	Oct. 18 2004	$79
⋮	⋮	⋮	
1	Paris	Oct. 18 2004	301
⋮	⋮	⋮	⋮
27	Minneapolis	Oct. 18 2004	$2 321
27	Chicago	Oct. 18 2004	$3 278
⋮	⋮	⋮	
27	Paris	Oct. 18 2004	$1 325
⋮	⋮	⋮	⋮

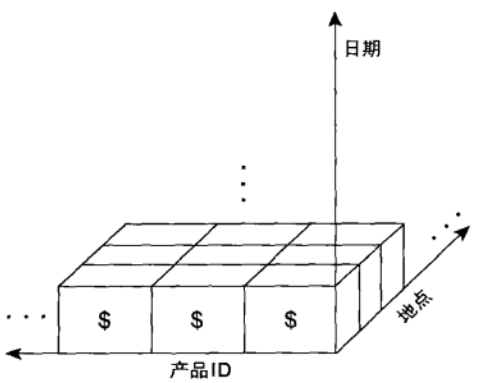


图 3-31　销售数据的多维表示

3.4.3　分析多维数据

本节介绍不同的多维分析技术。重点讨论数据立方体的创建和相关操作，如切片、切块、维归约、上卷和下钻。

1. 数据立方体：计算聚集量

从多维角度看待数据的主要动机就是需要以多种方式聚集数据。在产品销售的例子中，我们可能希望找出特定年份、特定产品的总销售收入，或者希望得到每一地点所有产品的年销售收入。计算聚集总和涉及固定某些属性（维）的值，在其余属性（维）的所有可能的值上求和。还有其他感兴趣的聚集量，但是为了简单起见，我们只讨论求和。

表 3-12 显示对于日期和产品的各种组合，在所有地点上求和的结果。为简单起见，假定所有的日期在一年之内。如果一年有 365 天，并且有 1 000 种产品，则表 3-12 有 365 000 个表项（总和），每个产品-日期对一个。也可以指定商店位置和日期，在产品上求和，或者指定地点和产品，在所有的日期上求和。

表 3-13 显示表 3-12 的**边缘总和**（marginal total）。这些总和是进一步在日期或产品上求和的结果。在表 3-13 中，产品 1 的总销售收入是$370 000，通过在第一行上（在所有日期上）求和得到。2004 年 1 月 1 日的总销售收入是$527 362，通过在第一列上（在所有产品上）求和得到。总销售收入是$227 352 127，通过在所有行和所有的列（所有的时间和产品）上求和得到。所有这

些总和都针对所有地点来的，因为表 3-13 的表项包括所有的地点。

表 3-12 对于固定的时间和产品，在所有地点上求和产生的总和

		日期			
		2004.1.1	2004.1.2	…	2004.12.31
产品 ID	1	$1 001	$987	…	$891
	⋮	⋮			⋮
	27	$10 265	$10 225	…	$9 325
	⋮	⋮			⋮

表 3-13 包括边缘总和的表 3-12

		日期				总和
		2004.1.1	2004.1.2	…	2004.12.31	
产品 ID	1	$1 001	$987	…	$891	$370 000
	⋮	⋮			⋮	⋮
	27	$10 265	$10 225	…	$9 325	$3 800 020
	⋮	⋮			⋮	⋮
总和		$527 362	$532 953	…	$631 221	$227 352 127

该例的要点是，从多维数组可以计算大量不同的总和（聚集），取决于我们在多少个属性上求和。假定有 n 个维，第 i 个维（属性）有 s_i 个可能的值，只在单个属性上求和有 n 种不同的方式，如果我们在维 j 上求和，则可以得到 $s_1*\cdots*s_{j-1}*s_{j+1}*\cdots*s_n$ 个总和，$n-1$ 个其他属性（维）的属性值的每种可能组合一个，在一个属性上求和得到的总和形成一个 $n-1$ 维数组，并且有 n 个这样的总和数组。在产品销售例子中，在一个属性上求和导致三组总和，每组总和可以用一个二维表显示。

如果我们在两个维上求和（或许从在一个维上求和得到的总和数组之一开始），则我们将得到具有 $n-2$ 个维的总和多维数组，总共有 C_n^2 个这样的总和数组。对于产品销售例子，有 $C_3^2 = 3$ 个总和数组，分别是在地点和产品、地点和时间、产品和时间上求和的结果。一般地，在 k 维上求和产生 C_n^k 个总和数组，每个具有 $n-k$ 维。

数据的多维表示，连同所有可能的总和（聚集）称作**数据立方体**（data cube）。尽管叫立方体，每个维的大小（属性值的个数）却不必相等。此外，数据立方体可能多于或少于三个维。更重要的是，数据立方体是称为**交叉表**（cross-tabulation）的统计学技术的推广，如果加上边缘总和，则表 3-8、表 3-9 和表 3-10 就成了交叉表的典型例子。

2. 维归约和转轴

前面介绍的聚集可以看作一种形式的**维归约**，具体说来，通过在第 j 维上求和，删除第 j 维，概念上讲，这将第 j 维上的每个单元“列”单缩成一个单元。对于销售和鸢尾花例子，在一个维上聚集将数据的维度从 3 归约 2。如果 s_j 是第 j 维上的可能值个数，则单元数约减了一个因子 s_j。本章习题 17 要求读者考察这种类型的维归约和 PCA 之间的区别。

转轴（pivoting）是指在除两个维之外的所有维上聚集。结果是一个二维交叉表，只有两个指定的维作为留下的维。表 3-13 是一个在日期和产品上转轴的例子。

3. 切片和切块

这两个生动的名字涉及相当直截了当的操作。**切片**（slicing）是通过对一个或多个维指定特定的值，从整个多维数组中选择一组单元。表 3-8、表 3-9 和表 3-10 是通过为种类维指定三个不同的值得到的鸢尾花数据集的三个切片。**切块**（dicing）涉及通过指定属性值区间选择单元子集，这等价于由整个数组定义子数组。在实践中，两个操作都可以通过在某些维上聚集来实现。

4. 上卷和下钻

在第 2 章，属性值在某种意义上被看作是“原子”的，然而，实际情况并非总是如此。例如，每个日期有一些与之相关联的性质，如年、月和星期；数据也可以被认为属于一个特定的商业季度，或者，如果应用与教育有关，看作学校的季或学期；地点也有各种性质：洲、国、州（省）和城市；产品也可以划分成各种类别，如服装、电子产品和家具。

通常，这些类别可以组织成树或格。例如，年由月或星期组成，而它们都由日组成；地点也可以划分成国家，国家包含州（或其他地方政府单位），而州又包含城市；类似地，产品类可以进一步划分，例如，产品类别家具可以划分成子类别：椅子、桌子、沙发，等等。

层次结构促使上卷和下钻操作的出现。为了解释这一点，考虑最初的销售数据，它是多维数组，记录每天的销售。我们可以按月聚集（**上卷**，roll up）销售数据。反过来，给定时间为划分成月份的数据表示，我们可能希望将月销售总和分解（**下钻**，drill down）成日销售总和，当然，这要求基本销售数据的时间粒度是按天的。

这样，上卷和下钻操作与聚集相关。然而，它们不同于迄今为止所讨论的聚集操作，它们在一个维内聚集单元，而不是在整个维上聚集。

3.4.4　关于多维数据分析的最后评述

就 OLAP 和相关系统所蕴涵的意义而言，多维数据分析将数据看作多维数组，并聚集数据，以便更好地分析数据的结构。对于鸢尾花数据，这种分析清楚地展现了花瓣长度和宽度的不同。对商务数据（如销售数据）的分析也能揭示许多有意义的模式，如有利润（或无利润）的商店或产品。

如前所述，有多种类型的数据库系统支持多维数据分析。其中某些系统基于关系数据库，称作 ROLAP 系统。业已设计了专门使用多维数据表示作为其基本数据模型的、更专业的数据库系统，这种系统称为 MOLAP 系统。除了这些类型的系统之外，还开发了统计数据库（SDB），来存储和分析各种统计数据，如政府和其他大型机构收集的人口普查和公共卫生数据。关于 OLAP 和 SDB 的参考文献在文献注释中提供。

文献注释

汇总统计在大部分统计学导论中都被详细讨论，如[92]。探测式数据分析的参考文献有 Tukey 的经典文献[104]、Velleman 和 Hoaglin 的书[105]。

作为大部分电子制表软件（Microsoft EXCEL[95]）、统计程序（SAS[99]、SPSS[102]、R[96]和 S-PLUS[98]）以及数学软件（MATLAB[94]和 Mathematica[93]）的必不可少的一部分，基本的可视化技术可以很容易地使用，本章的大部分图都是使用 MATLAB 制做的，作为 R 项目的开放源码软件包，统计软件包 R 是免费的。

关于可视化的文献极多，涵盖了许多领域，跨越了数十年。该领域的经典著作之一是 Tufte

的书[103]。Spence 的书[101]对本章的可视化部分影响很大，在原理和技术两方面都是信息可视化的有用文献，该书还提供了许多动态可视化技术的详尽讨论，而本章并未包含这些内容。另外两本关于可视化的可能也值得关注的书是 Card 等[87]和 Fayyad 等[89]的书。

最后，万维网上有关于可视化的大量信息。由于 Web 站点来去频繁。最好的办法是使用“信息可视化”、“数据可视化”或“统计图形”进行搜索。然而，我们想特意提到 Friendly 的《数据可视化图库》（*The Gallery of Data Visualization*）[90]。本章介绍的有效图形显示的 ACCENT 原则可以在这里找到，或在 Burn 最初提出该原则的文章[86]中找到。

有多种图形技术，可以用来考察数据是否是高斯分布的，或者是某种其他分布。此外，还有一些图，用以显示观测值在某种意义上是否是统计显著的。我们没有涵盖这些技术，建议读者查阅前面提到的统计和数学软件包。

多维数据分析已经以各种形式存在多年。最早的文章之一是关系数据库之父 Codd 的白皮书[88]。数据立方体是 Gray 等[91]提出的，他们介绍了在关系数据库框架下，创建和操纵数据立方体的各种操作。统计数据库与 OLAP 的比较由 Shoshani[100]给出。关于 OLAP 的特殊信息可以在数据库销售商的文档资料和许多畅销书中找到。许多数据库教材也有 OLAP 的一般性讨论，通常是在数据仓库上下文中，例如，Ramakrishnan 和 Gehrke 的书[97]。

参考文献

[86] D. A. Burn. Designing Effective Statistical Graphs. In C. R. Rao, editor, *Handbook of Statistics 9*. Elsevier/North-Holland, Amsterdam, The Netherlands, September 1993.

[87] S. K. Card, J. D. MacKinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA, January 1999.

[88] E. F. Codd, S. B. Codd, and C. T. Smalley. Providing OLAP (On-line Analytical Processing) to User-Analysts: An ITMandate. White Paper, E.F. Codd and Associates, 1993.

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

[90] M. Friendly. Gallery of Data Visualization. http://www.math.yorku.ca/SCS/Gallery/, 2005.

[91] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Journal Data Mining and Knowledge Discovery*, 1(1):29 - 53, 1997.

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

[93] Mathematica 5.1. Wolfram Research, Inc. http://www.wolfram.com/, 2005.

[94] MATLAB 7.0. The MathWorks, Inc. http://www.mathworks.com, 2005.

[95] Microsoft Excel 2003. Microsoft, Inc. http://www.microsoft.com/, 2003.

[96] R: A language and environment for statistical computing and graphics. The R Project for Statistical Computing. http://www.r-project.org/, 2005.

[97] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill, 3rd edition, August 2002.

[98] S-PLUS. Insightful Corporation. http://www.insightful.com, 2005.

[99] SAS: Statistical Analysis System. SAS Institute Inc. http://www.sas.com/, 2005.

[100] A. Shoshani. OLAP and statistical databases: similarities and differences. In *Proc. of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, pages 185 - 196. ACM Press, 1997.

[101] R. Spence. *Information Visualization*. ACM Press, New York, December 2000.

[102] SPSS: Statistical Package for the Social Sciences. SPSS, Inc. http://www.spss.com/, 2005.

[103] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, March 1986.
[104] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977.
[105] P. Velleman and D. Hoaglin. *The ABC's of EDA: Applications, Basics, and Computing of Exploratory Data Analysis*. Duxbury, 1981.

习 题

1. 从 UCI 机器学习库取得一个数据集，并尽可能多地使用本章介绍的不同的可视化技术。文献注释和本书网站提供了可视化软件的线索。
2. 至少指出使用颜色可视地提供信息的两个优点和两个缺点。
3. 关于三维图形，安排问题是什么？
4. 讨论使用抽样减少需要显示的数据对象个数的优缺点。简单随机抽样（无放回）是一种好的抽样方法吗？为什么是，为什么不是？
5. 如何创建可视表示来显示描述如下系统的信息。
   (a) 计算机网络。确保包括网络的静态性质（如连接性）和动态性质（如通信量）。
   (b) 在特定时间，特定的植物和动物种类在全世界的分布。
   (c) 对于一组基准数据库程序，计算机资源（如处理机时间、内存和磁盘）的使用情况。
   (d) 过去 30 年内，一个特定国家的工人职业的变化。假定提供了每个人每年的信息，还提供了性别和文化程度。

   确保处理了以下问题。
   - **表示**。如何将对象、属性和联系映射到可视化元素？
   - **安排**。关于如何显示可视化元素，是否有需要考虑的特殊问题？具体的例子可能是视点的选择、透明度的使用、或将特定的对象组分开。
   - **选择**。如何处理大量属性和数据对象？
6. 相对于标准直方图，指出茎叶图的一个优点和一个缺点。
7. 如何处理直方图依赖于箱的个数和位置的问题？
8. 描述盒状图如何提供属性值是否对称分布的信息。关于图3-11显示的属性的分布对称性，你有何种结论？
9. 使用图 3-12，比较萼片长度、萼片宽度、花瓣长度和花瓣宽度。
10. 评论使用盒状图考察具有如下 4 个属性的数据集：年龄、体重、身高和收入。
11. 对于为什么花瓣长度和宽度的大部分值都落在图 3-9 沿对角线的桶中，给出一个可能的解释。
12. 使用图 3-14 和图 3-15，识别花瓣宽度和花瓣长度属性共同的特性。
13. 简单线图（如图 2-12 显示两个时间序列的图）可以用来有效地显示高维数据。例如，在图 2-12 中，容易看出两个时间序列的频率是不同的。时间序列的什么特性使得高维数据可以有效地可视化？
14. 描述产生稀疏和稠密数据立方体情况类型。使用本书之外的例子加以解释。
15. 如何扩展多维数据分析概念，使得目标变量可以是定性变量？换言之，何种汇总统计或数据可视化是令人感兴趣的？

16. 由表 3-14 构造数据立方体。这是稠密还是稀疏数据立方体？如果是稀疏的，识别出空单元。

表 3-14　习题 16 的事实表

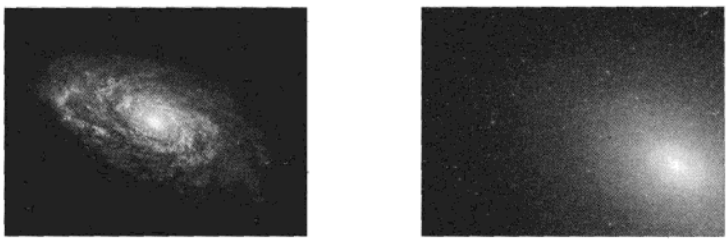
产品 ID	地点 ID	销售量
1	1	10
1	3	6
2	1	5
2	2	22

17. 讨论基于聚集的维归约与基于 PCA 和 SVD 等技术的维归约的区别。

第4章

分类：基本概念、决策树与模型评估

分类任务就是确定对象属于哪个预定义的目标类。分类问题是一个普遍存在的问题，有许多不同的应用。例如：根据电子邮件的标题和内容检查出垃圾邮件，根据核磁共振扫描的结果区分肿瘤是恶性的还是良性的，根据星系的形状对它们进行分类（见图 4-1）。



(a) 螺旋状的星系　　(b) 椭圆状的星系

图 4-1　星系的分类。图片来源于 NASA 网站

本章介绍分类的基本概念，讨论诸如模型的过分拟合等关键问题，并提供评估和比较分类技术性能的方法。尽管本章主要关注一种称作决策树归纳的技术，但是本章讨论的大部分内容也适用于其他的分类技术，其中很多技术将在第 5 章介绍。

4.1　预备知识

分类任务的输入数据是记录的集合。每条记录也称实例或样例，用元组$(\mathbf{x}, y)$表示，其中 $\mathbf{x}$ 是属性的集合，而 y 是一个特殊的属性，指出样例的类标号（也称为分类属性或目标属性）。表 4-1 列出一个样本数据集，用来将脊椎动物分为以下几类：哺乳类、鸟类、鱼类、爬行类和两栖类。属性集指明脊椎动物的性质，如体温、表皮覆盖、繁殖后代的方式、飞行的能力和在水中生存的能力等。尽管表 4-1 中的属性主要是离散的，但是属性集也可以包含连续特征。另一方面，类标号却必须是离散属性，这正是区别分类与回归（regression）的关键特征。回归是一种预测建模任务，其中目标属性 y 是连续的。

定义 4.1　分类（classification）　分类任务就是通过学习得到一个**目标函数**（target function）f，把每个属性集 $\mathbf{x}$ 映射到一个预先定义的类标号 y。

目标函数也称**分类模型**（classification model）。分类模型可以用于以下目的。

描述性建模　分类模型可以作为解释性的工具，用于区分不同类中的对象。例如，对于生物

表 4-1　脊椎动物的数据集

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	否	是	是	否	鸟类
猫	恒温	软毛	是	否	否	是	否	哺乳类
豹纹鲨	冷血	鳞片	是	是	否	否	否	鱼类
海龟	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
鳗	冷血	鳞片	否	是	否	否	否	鱼类
蝾螈	冷血	无	否	半	否	是	是	两栖类

学家或者其他人，一个描述性模型有助于概括表 4-1 中的数据，并说明哪些特征决定一种脊椎动物是哺乳类、爬行类、鸟类、鱼类或者两栖类。

预测性建模　分类模型还可以用于预测未知记录的类标号。如图 4-2 所示，分类模型可以看作是一个黑箱，当给定未知记录的属性集上的值时，它自动地赋予未知样本类标号。例如，假设有一种叫作毒蜥（gila monster）的生物，其特征如下：

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
毒蜥	冷血	鳞片	否	否	否	是	是	?

可以使用根据表 4-1 中的数据集建立的分类模型来确定该生物所属的类。

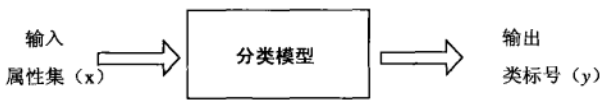


图 4-2　分类器的任务是根据输入属性集 x 确定类标号 y

分类技术非常适合预测或描述二元或标称类型的数据集，对于序数分类（例如，把人分类为高收入、中等收入或低收入组），分类技术不太有效，因为分类技术不考虑隐含在目标类中的序关系。其他形式的联系，如子类与超类的关系（例如，人类和猿都是灵长类动物，而灵长类是哺乳类的子类）也被忽略。本章余下的部分只考虑二元的或标称类型的类标号。

4.2　解决分类问题的一般方法

分类技术（或分类法）是一种根据输入数据集建立分类模型的系统方法。分类法的例子包括决策树分类法、基于规则的分类法、神经网络、支持向量机和朴素贝叶斯分类法。这些技术都使用一种**学习算法**（learning algorithm）确定分类模型，该模型能够很好地拟合输入数据中类标号和属性集之间的联系。学习算法得到的模型不仅要很好地拟合输入数据，还要能够正确地预测未知样本的类标号。因此，训练算法的主要目标就是建立具有很好的泛化能力模型，即建立能够准确地预测未知样本类标号的模型。

图 4-3 展示解决分类问题的一般方法。首先，需要一个**训练集**（training set），它由类标号已知的记录组成。使用训练集建立分类模型，该模型随后将运用于**检验集**（test set），检验集由类标号未知的记录组成。

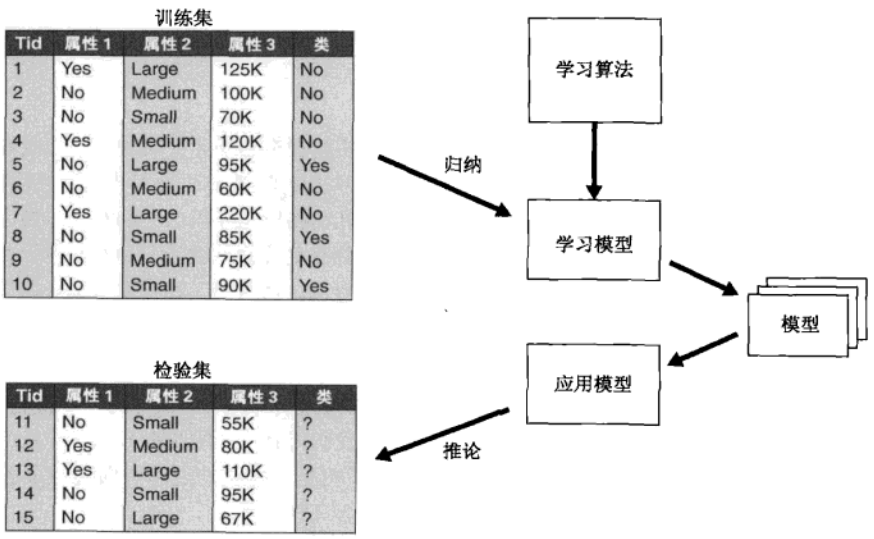


图 4-3 建立分类模型的一般方法

分类模型的性能根据模型正确和错误预测的检验记录计数进行评估，这些计数存放在称作**混淆矩阵**（confusion matrix）的表格中。表 4-2 描述二元分类问题的混淆矩阵。表中每个表项 f_{ij} 表示实际类标号为 i 但被预测为类 j 的记录数，例如，f_{01} 代表原本属于类 0 但被误分为类 1 的记录数。按照混淆矩阵中的表项，被分类模型正确预测的样本总数是（$f_{11}+f_{00}$），而被错误预测的样本总数是（$f_{10}+f_{01}$）。

表 4-2 二类问题的混淆矩阵

		预测的类	
		类 =1	类 =0
实际的类	类 =1	f_{11}	f_{10}
	类 =0	f_{01}	f_{00}

虽然混淆矩阵提供衡量分类模型性能的信息，但是用一个数汇总这些信息更便于比较不同模型的性能。为实现这一目的，可以使用**性能度量**（performance metric），如**准确率**（accuracy），其定义如下：

$$\text{准确率} = \frac{\text{正确预测数}}{\text{预测总数}} = \frac{f_{11}+f_{00}}{f_{11}+f_{10}+f_{01}+f_{00}} \tag{4-1}$$

同样，分类模型的性能可以用**错误率**（error rate）来表示，其定义如下：

$$\text{错误率} = \frac{\text{错误预测数}}{\text{预测总数}} = \frac{f_{10}+f_{01}}{f_{11}+f_{10}+f_{01}+f_{00}} \tag{4-2}$$

大多数分类算法都在寻求这样一些模型，当把它们应用于检验集时具有最高的准确率，或者等价地，具有最低的错误率。我们将在 4.5 节再次讨论模型的评估问题。

4.3　决策树归纳

本节介绍**决策树**（decision tree）分类法，这是一种简单但却广泛使用的分类技术。

4.3.1　决策树的工作原理

为了解释决策树分类的工作原理，考虑上一节中介绍的脊椎动物分类问题的简化版本。这里，我们不把脊椎动物分为五个不同的物种，而只考虑两个类别：哺乳类动物和非哺乳类动物。

假设科学家发现了一个新的物种，怎么判断它是哺乳动物还是非哺乳动物呢？一种方法是针对物种的特征提出一系列问题。第一个问题可能是，该物种是冷血动物还是恒温动物。如果它是冷血的，则该物种肯定不是哺乳动物；否则它或者是某种鸟，或者是某种哺乳动物。如果它是恒温的，需要接着问：该物种是由雌性产崽进行繁殖的吗？如果是，则它肯定为哺乳动物，否则它有可能是非哺乳动物（鸭嘴兽和针鼹这些产蛋的哺乳动物除外）。

上面的例子表明，通过提出一系列精心构思的关于检验记录属性的问题，可以解决分类问题。每当一个问题得到答案，后续的问题将随之而来，直到我们得到记录的类标号。这一系列的问题和这些问题的可能回答可以组织成决策树的形式，决策树是一种由结点和有向边组成的层次结构。图 4-4 显示哺乳类动物分类问题的决策树，树中包含三种结点。

- **根结点**（root node），它没有入边，但有零条或多条出边。
- **内部结点**（internal node），恰有一条入边和两条或多条出边。
- **叶结点**（leaf node）或**终结点**（terminal node），恰有一条入边，但没有出边。

在决策树中，每个叶结点都赋予一个类标号。**非终结点**（non-terminal node）（包括根结点和内部结点）包含属性测试条件，用以分开具有不同特性的记录。例如，在图 4-4 中，在根结点处，使用体温这个属性把冷血脊椎动物和恒温脊椎动物区别开来。因为所有的冷血脊椎动物都是非哺乳动物，所以用一个类称号为非哺乳动物的叶结点作为根结点的右子女。如果脊椎动物的体温是恒温的，则接下来用胎生这个属性来区分哺乳动物与其他恒温动物（主要是鸟类）。

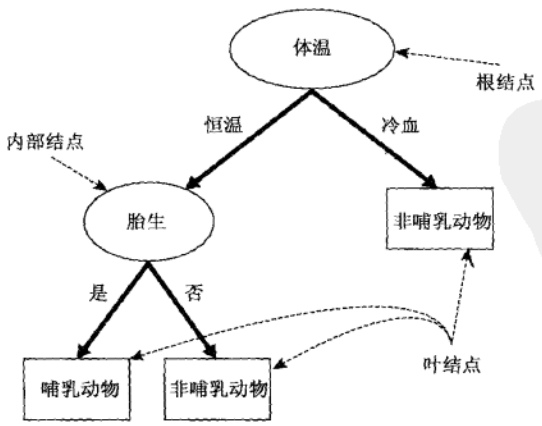


图 4-4　哺乳动物分类问题的决策树

一旦构造了决策树，对检验记录进行分类就相当容易了。从树的根结点开始，将测试条件用于检验记录，根据测试结果选择适当的分支。沿着该分支或者到达另一个内部结点，使用新的测试条件，或者到达一个叶结点。到达叶结点之后，叶结点的类称号就被赋值给该检验记录。例如，图 4-5 显示应用决策树预测火烈鸟的类标号所经过的路径，路径终止于类称号为非哺乳动物的叶结点。

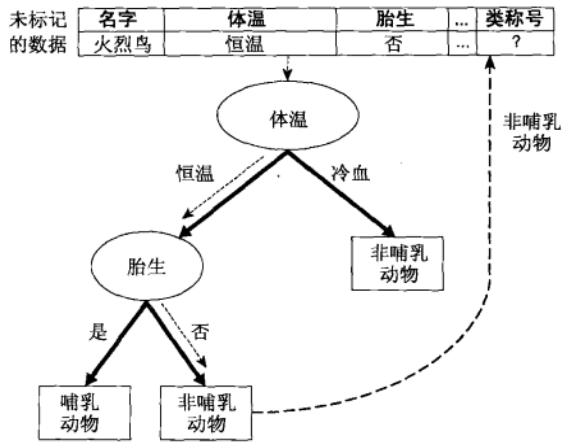


图 4-5 对一种未标记的脊椎动物分类。虚线表示在未标记的脊椎动物上使用各种属性测试条件的结果。该脊椎动物最终被指派到非哺乳动物类

4.3.2 如何建立决策树

原则上讲，对于给定的属性集，可以构造的决策树的数目达指数级。尽管某些决策树比其他决策树更准确，但是由于搜索空间是指数规模的，找出最佳决策树在计算上是不可行的。尽管如此，人们还是开发了一些有效的算法，能够在合理的时间内构造出具有一定准确率的次最优决策树。这些算法通常都采用贪心策略，在选择划分数据的属性时，采取一系列局部最优决策来构造决策树，Hunt 算法就是一种这样的算法。Hunt 算法是许多决策树算法的基础，包括 ID3、C4.5 和 CART。本节讨论 Hunt 算法并解释它的一些设计问题。

1. Hunt 算法

在 Hunt 算法中，通过将训练记录相继划分成较纯的子集，以递归方式建立决策树。设 D_t 是与结点 t 相关联的训练记录集，而 $y = \{y_1, y_2, \cdots, y_c\}$ 是类标号，Hunt 算法的递归定义如下。

(1) 如果 D_t 中所有记录都属于同一个类 y_t，则 t 是叶结点，用 y_t 标记。

(2) 如果 D_t 中包含属于多个类的记录，则选择一个**属性测试条件**（attribute test condition），将记录划分成较小的子集。对于测试条件的每个输出，创建一个子女结点，并根据测试结果将 D_t 中的记录分布到子女结点中。然后，对于每个子女结点，递归地调用该算法。

为了解释该算法如何执行，考虑如下问题：预测贷款申请者是会按时归还贷款，还是会拖欠贷款。对于这个问题，训练数据集可以通过考察以前贷款者的贷款记录来构造。在图 4-6 所示的例子中，每条记录都包含贷款者的个人信息，以及贷款者是否拖欠贷款的类标号。

	二元的	分类的	连续的	类
Tid	有房者	婚姻状况	年收入	拖欠贷款者
1	是	单身	125K	否
2	否	已婚	100K	否
3	否	单身	70K	否
4	是	已婚	120K	否
5	否	离异	95K	是
6	否	已婚	60K	否
7	是	离异	220K	否
8	否	单身	85K	是
9	否	已婚	75K	否
10	否	单身	90K	是

图 4-6　训练数据集：预测拖欠银行贷款的贷款者

该分类问题的初始决策树只有一个结点，类标号为“*拖欠贷款者 = 否*”（见图 4-7a），意味大多数贷款者都按时归还贷款。然而，该树需要进一步的细化，因为根结点包含两个类的记录。根据“*有房者*”测试条件，这些记录被划分为较小的子集，如图 4-7b所示。选取属性测试条件的理由稍后讨论，目前，我们假定此处这样选是划分数据的最优标准。接下来，对根结点的每个子女递归地调用 Hunt 算法。从图 4-6 给出的训练数据集可以看出，有房的贷款者都按时偿还了贷款，因此，根结点的左子女为叶结点，标记为“*拖欠贷款者 = 否*”（见图 4-7b）。对于右子女，我们需要继续递归调用 Hunt 算法，直到所有的记录都属于同一个类为止。每次递归调用所形成的决策树显示在图 4-7c 和图 4-7d 中。

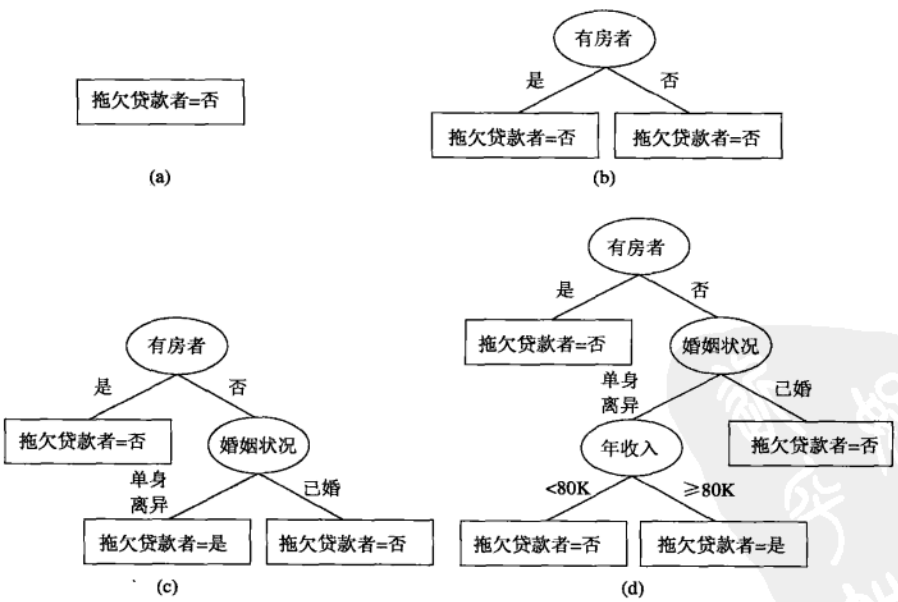


图 4-7　Hunt 算法构造决策树

如果属性值的每种组合都在训练数据中出现，并且每种组合都具有唯一的类标号，则 Hunt 算法是有效的。但是对于大多数实际情况，这些假设太苛刻了，因此，需要附加的条件来处理以下的情况。

(1) 算法的第二步所创建的子女结点可能为空，即不存在与这些结点相关联的记录。如果没有一个训练记录包含与这样的结点相关联的属性值组合，这种情形就可能发生。这时，该结点成为叶结点，类标号为其父结点上训练记录中的多数类。

(2) 在第二步，如果与 D_t 相关联的所有记录都具有相同的属性值（目标属性除外），则不可能进一步划分这些记录。在这种情况下，该结点为叶结点，其标号为与该结点相关联的训练记录中的多数类。

2. 决策树归纳的设计问题

决策树归纳的学习算法必须解决下面两个问题。

(1) **如何分裂训练记录？** 树增长过程的每个递归步都必须选择一个属性测试条件，将记录划分成较小的子集。为了实现这个步骤，算法必须提供为不同类型的属性指定测试条件的方法，并且提供评估每种测试条件的客观度量。

(2) **如何停止分裂过程？** 需要有结束条件，以终止决策树的生长过程。一个可能的策略是分裂结点，直到所有的记录都属于同一个类，或者所有的记录都具有相同的属性值。尽管两个结束条件对于结束决策树归纳算法都是充分的，但是还可以使用其他的标准提前终止树的生长过程。提前终止的优点将在 4.4.5 节讨论。

4.3.3 表示属性测试条件的方法

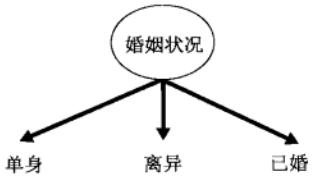
决策树归纳算法必须为不同类型的属性提供表示属性测试条件和其对应输出的方法。

二元属性 二元属性的测试条件产生两个可能的输出，如图 4-8 所示。

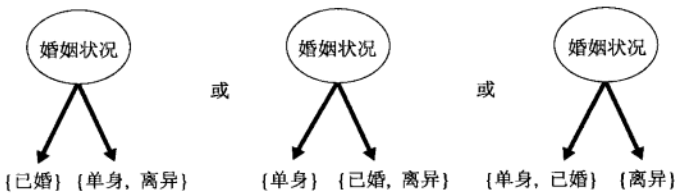
图 4-8 二元属性的测试条件

标称属性 由于标称属性有多个属性值，它的测试条件可以用两种方法表示，如图 4-9 所示。对于多路划分（图 4-9a），其输出数取决于该属性不同属性值的个数。例如，如果属性婚姻状况有三个不同的属性值——单身、已婚、离异，则它的测试条件就会产生一个三路划分。另一方面，某些决策树算法（如 CART）只产生二元划分，它们考虑创建 k 个属性值的二元划分的所有 $2^{k-1}-1$ 种方法。图 4-9b 显示了把婚姻状况的属性值划分为两个子集的三种不同的分组方法。

序数属性 序数属性也可以产生二元或多路划分，只要不违背序数属性值的有序性，就可以对属性值进行分组。图 4-10 显示了按照属性衬衣尺码划分训练记录的不同的方法。图 4-10a 和图 4-10b 中的分组保持了属性值间的序关系，而图 4-10c 所示的分组则违反了这一性质，因为它把小号和大号分为一组，把中号和加大号放在另一组。

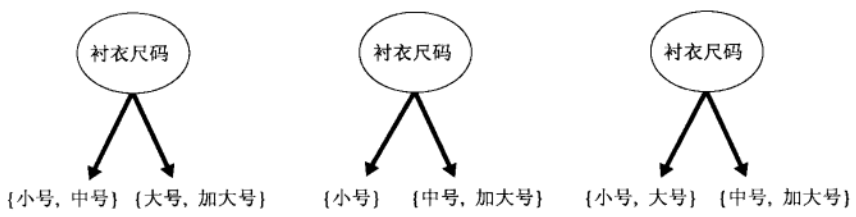


(a) 多路划分



(b) 二元划分（通过属性值分组）

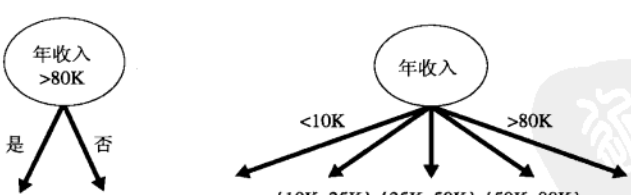
图 4-9　标称属性的测试条件



(a)　(b)　(c)

图 4-10　序数属性值分组的不同方式

连续属性　对于连续属性来说，测试条件可以是具有二元输出的比较测试（$A < v$）或（$A \geqslant v$），也可以是具有形如 $v_i \leqslant A < v_{i+1}$（$i = 1, \cdots, k$）输出的范围查询，图 4-11 显示了这些方法的差别。对于二元划分，决策树算法必须考虑所有可能的划分点 v，并从中选择产生最佳划分的点。对于多路划分，算法必须考虑所有可能的连续值区间。可以采用 2.3.6 节介绍的离散化的策略，离散化之后，每个离散化区间赋予一个新的序数值，只要保持有序性，相邻的值还可以聚集成较宽的区间。



(a)　(b)

图 4-11　连续属性的测试条件

4.3.4　选择最佳划分的度量

有很多度量可以用来确定划分记录的最佳方法，这些度量用划分前和划分后记录的类分布定义。

设 $p(i|t)$表示给定结点 t 中属于类 i 的记录所占的比例，有时，我们省略结点 t，直接用 p_i 表示该比例。在两类问题中，任意结点的类分布都可以记作(p_0, p_1)，其中 $p_1 = 1 - p_0$。例如，考虑图 4-12 中的测试条件，划分前的类分布是(0.5, 0.5)，因为来自每个类的记录数相等。如果使用性别属性来划分数据，则子女结点的类分布分别为(0.6, 0.4)和(0.4, 0.6)，虽然划分后两个类的分布不再平衡，但是子女结点仍然包含两个类的记录；按照第二个属性车型进行划分，将得到纯度更高的划分。

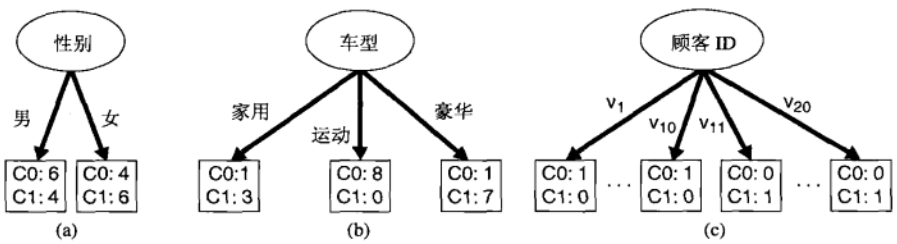


图 4-12 多路划分与二元划分

选择最佳划分的度量通常是根据划分后子女结点不纯性的程度。不纯的程度越低，类分布就越倾斜。例如，类分布为(0, 1)的结点具有零不纯性，而均衡分布 (0.5, 0.5) 的结点具有最高的不纯性。不纯性度量的例子包括：

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t) \tag{4-3}$$

$$\text{Gini}(t) = 1-\sum_{i=0}^{c-1}[p(i|t)]^2 \tag{4-4}$$

$$\text{Classification error}(t) = 1-\max_i[p(i|t)] \tag{4-5}$$

其中 c 是类的个数，并且在计算熵时，$0\log_2 0 = 0$。

图 4-13 显示了二元分类问题不纯性度量值的比较，p 表示属于其中一个类的记录所占的比例。从图中可以看出，三种方法都在类分布均衡时（即当 $p = 0.5$ 时）达到最大值，而当所有记录都属于同一个类时（p 等于 1 或 0）达到最小值。下面我们给出三种不纯性度量方法的计算实例。

结点 N_1	计数	Gini = 1 – (0/6)² – (6/6)² = 0
类 =0	0	Entropy = – (0/6)log₂ (0/6) – (6/6)log₂(6/6) = 0
类 =1	6	Error = 1 – max[0/6, 6/6] = 0

结点 N_2	计数	Gini = 1 – (1/6)² – (5/6)² = 0.278
类 =0	1	Entropy = – (1/6)log₂ (1/6) – (5/6)log₂(5/6) = 0.650
类 =1	5	Error = 1 – max[1/6, 5/6] = 0.167

结点 N_3	计数	Gini = 1 – (3/6)² – (3/6)² = 0.5
类 =0	3	Entropy = – (3/6)log₂ (3/6) – (3/6)log₂(3/6) = 1
类 =1	3	Error = 1 – max[3/6, 3/6] = 0.5

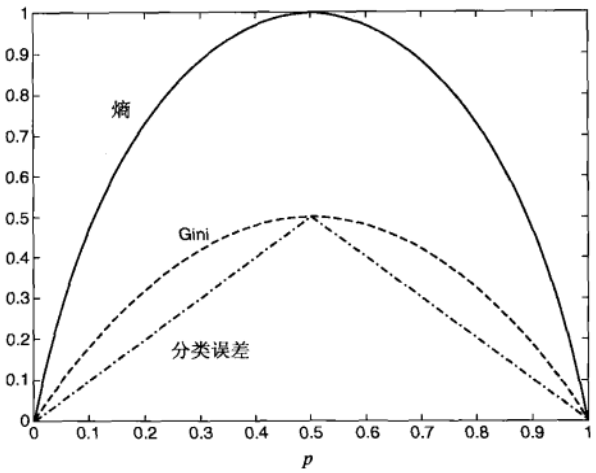


图 4-13　二元分类问题不纯性度量之间的比较

从上面的例子及图 4-13 可以看出，不同的不纯性度量是一致的。根据计算，结点 N_1 具有最低的不纯性度量值，接下来依次是 N_2，N_3。虽然结果是一致的，但是作为测试条件的属性选择仍然因不纯性度量的选择而异，如本章习题 3 所示。

为了确定测试条件的效果，我们需要比较父结点（划分前）的不纯程度和子女结点（划分后）的不纯程度，它们的差越大，测试条件的效果就越好。增益Δ是一种可以用来确定划分效果的标准：

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j) \tag{4-6}$$

其中，$I(.)$是给定结点的不纯性度量，N 是父结点上的记录总数，k 是属性值的个数，$N(v_j)$是与子女结点 v_j 相关联的记录个数。决策树归纳算法通常选择最大化增益Δ的测试条件，因为对所有的测试条件来说，I(parent)是一个不变的值，所以最大化增益等价于最小化子女结点的不纯性度量的加权平均值。最后，当选择熵（entropy）作为公式（4-6）的不纯性度量时，熵的差就是所谓**信息增益**（information gain）Δ_{info}。

1. 二元属性的划分

考虑图 4-14 中的图表，假设有两种方法将数据划分成较小的子集。划分前，Gini 指标等于 0.5，因为属于两个类的记录个数相等。如果选择属性 A 划分数据，结点 N_1 的 Gini 指标等于 0.4898，而 N_2 的 Gini 指标等于 0.480，派生结点的 Gini 指标的加权平均为 $(7/12) \times 0.4898 + (5/12) \times 0.480 = 0.486$。类似的，我们可以计算属性 B 的 Gini 指标加权平均是 0.371。因为属性 B 具有更小的 Gini 指标，它比属性 A 更可取。

2. 标称属性的划分

正如前面提到的，标称属性可以产生二元划分或多路划分，如图 4-15 所示。二元划分的 Gini 指标的计算与二元属性类似。对于车型属性第一种二元分组，{运动，豪华}的 Gini 指标是 0.4922，而{家用}的 Gini 指标是 0.3750。该分组的 Gini 指标加权平均是：

$$16/20 \times 0.4922 + 4/20 \times 0.3750 = 0.468$$

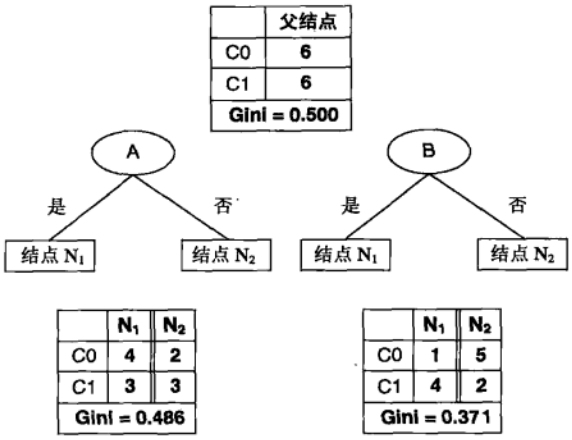


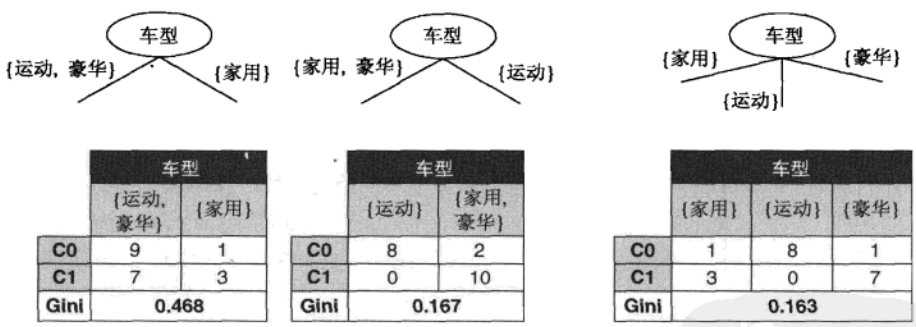
图 4-14 划分二元属性

类似的，对第二种二元分组{运动}和{家用，豪华}，Gini 指标加权平均是 0.167。第二种分组的 Gini 指标值相对较低，因为其对应的子集的纯度高得多。

对于多路划分，需要计算每个属性值 Gini 指标。Gini({家用}) = 0.375，Gini({运动}) = 0，Gini({豪华}) = 0.219，多路划分的总 Gini 指标等于：

$$4/20 \times 0.375 + 8/20 \times 0 + 8/20 \times 0.219 = 0.163$$

多路划分的 Gini 指标比两个二元划分都小。这一结果并不奇怪，因为二元划分实际上合并了多路划分的某些输出，自然降低了子集的纯度。



(a) 二元划分 (b) 多路划分

图 4-15 划分标称属性

3. 连续属性的划分

考虑图 4-16 所示的例子，其中测试条件“年收入 $\leqslant v$”用来划分拖欠贷款分类问题的训练记录。用穷举方法确定 v 的值，将 N 个记录中所有的属性值都作为候选划分点。对每个候选 v，都要扫描一次数据集，统计年收入大于和小于 v 的记录数，然后计算每个候选的 Gini 指标，并从中选择具有最小值的候选划分点。这种方法的计算代价是昂贵的，因为对每个候选划分点计算 Gini 指标需要 $O(N)$次操作，由于有 N 个候选，总的计算复杂度为 $O(N^2)$。为了降低计算复杂度，按照年收入将训练记录排序，所需要的时间为 $O(N\log N)$，从两个相邻的排过序的属性值中选择

中间值作为候选划分点，得到候选划分点 55, 65, 72 等。无论如何，与穷举方法不同，在计算候选划分点的 Gini 指标时，不需考察所有 N 个记录。

类	No	No	No	Yes	Yes	Yes	No	No	No	No
年收入										
排序后的值 →	60	70	75	85	90	95	100	120	125	220
划分点 →	55	65	72	80	87	92	97	110	122	172	230

	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

图 4-16　连续属性的划分

对第一个候选 $v = 55$，没有年收入小于\$55K 的记录，所以年收入 < \$55K 的派生结点的 Gini 指标是 0；另一方面，年收入大于或等于\$55K 的样本记录数目分别为 3（类 Yes）和 7（类 No），这样，该结点的 Gini 指标是 0.420。该候选划分的总 Gini 指标等于 $0 \times 0 + 1 \times 0.420 = 0.420$。

对第二个候选 $v = 65$，通过更新上一个候选的类分布，就可以得到该候选的类分布。更具体地说，新的分布通过考察具有最低年收入（即\$60K）的记录的类标号得到。因为该记录的类标号是 No，所以类 No 的计数从 0 增加到 1（对于年收入 < \$65K），和从 7 降到 6（对于年收入 > \$65K），类 Yes 的分布保持不变。新的候选划分点的加权平均 Gini 指标为 0.400。

重复这样的计算，直到算出所有候选的 Gini 指标值，如图 4-16 所示。最佳的划分点对应于产生最小 Gini 指标值的点，即 $v = 97$。该过程代价相对较低，因为更新每个候选划分点的类分布所需的时间是一个常数。该过程还可以进一步优化：仅考虑位于具有不同类标号的两个相邻记录之间的候选划分点。例如，因为前三个排序后的记录（分别具有年收入\$60K、\$70K 和\$75K）具有相同的类标号，所以最佳划分点肯定不会在\$60K 和\$75K 之间，因此，候选划分点 $v =$ \$55K、\$65K、\$72K、\$87K、\$92K、\$110K、\$122K、\$172K和\$230K 都将被忽略，因为它们都位于具有相同类标号的相邻记录之间。该方法使得候选划分点的个数从 11 个降到 2 个。

4. 增益率

熵和 Gini 指标等不纯性度量趋向有利于具有大量不同值的属性。图 4-12 显示了三种可供选择的测试条件，划分本章习题 2 中的数据集。第一个测试条件*性别*与第二个测试条件*车型*相比，容易看出*车型*似乎提供了更好的划分数据的方法，因为它产生更纯的派生结点。然而，如果将这两个条件与*顾客 ID* 相比，后者看来产生更纯的划分，但*顾客 ID* 却不是一个有预测性的属性，因为每个样本在该属性上的值都是唯一的。即使在不太极端情形下，也不会希望产生大量输出的测试条件，因为与每个划分相关联的记录太少，以致不能作出可靠的预测。

解决该问题的策略有两种。第一种策略是限制测试条件只能是二元划分，CART 这样的决策树算法采用的就是这种策略；另一种策略是修改评估划分的标准，把属性测试条件产生的输出数也考虑进去，例如，决策树算法 C4.5 采用称作**增益率**（gain ratio）的划分标准来评估划分。增益率定义如下：

$$Gain\ ratio = \frac{\Delta_{\text{info}}}{Split\ Info} \tag{4-7}$$

其中，划分信息 $Split\ Info = -\sum_{i=1}^{k} P(v_i)\log_2 P(v_i)$，而 k 是划分的总数。例如，如果每个属性值具有相同的记录数，则$\forall i$：$P(v_i) = 1/k$，而划分信息等于 $\log_2 k$。这说明如果某个属性产生了大量的划分，它的划分信息将会很大，从而降低了增益率。

4.3.5 决策树归纳算法

算法 4.1 给出了称作 `TreeGrowth` 的决策树归纳算法的框架。该算法的输入是训练记录集 E 和属性集 F。算法递归地选择最优的属性来划分数据（步骤 7），并扩展树的叶结点（步骤 11 和步骤 12），直到满足结束条件（步骤 1）。算法的细节如下。

(1) 函数 `createdNode()`为决策树建立新结点。决策树的结点或者是一个测试条件，记作 *node.test_cond*，或者是一个类标号，记作 *node.label*。

(2) 函数 `find_best_split()`确定应当选择哪个属性作为划分训练记录的测试条件。如前所述，测试条件的选择取决于使用哪种不纯性度量来评估划分，一些广泛使用的度量包括熵、Gini 指标和χ^2统计量。

(3) 函数 `Classify()`为叶结点确定类标号。对于每个叶结点 t，令 $p(i|t)$表示该结点上属于类 i 的训练记录所占的比例，在大多数情况下，都将叶结点指派到具有多数记录的类：

$$leaf.label = \underset{i}{\operatorname{argmax}}\ p(i|t) \tag{4-8}$$

其中，操作 argmax 返回最大化 $p(i|t)$的参数值 i。$p(i|t)$除了提供确定叶结点类标号所需要的信息之外，还可以用来估计分配到叶结点 t 的记录属于类 i 的概率。5.7.2 节和 5.7.3 节讨论如何使用这种概率估计，在不同的代价函数下，确定决策树的性能。

(4) 函数 `stopping_cond()`通过检查是否所有的记录都属于同一个类，或者都具有相同的属性值，决定是否终止决策树的增长。终止递归函数的另一种方法是，检查记录数是否小于某个最小阈值。

算法 4.1 决策树归纳算法的框架

```
TreeGrowth(E, F)
 1: if stopping_cond(E, F) = true then
 2:    leaf = createNode()
 3:    leaf.label = Classify(E)
 4:    return leaf
 5: else
 6:    root = createNode()
 7:    root.test_cond = find_best_split(E, F)
 8:    令 V = {v | v 是 root.test_cond 的一个可能的输出}
 9:    for 每个 v∈V do
10:       E_v = {e | root.test_cond (e) = v 并且 e ∈ E}
11:       child = TreeGrowth(E_v, F)
12:       将 child 作为 root 的派生结点添加到树中，并将边(root→child)标记为 v
13:    end for
14: end if
15: return root
```

建立决策树之后，可以进行**树剪枝**（tree-pruning），以减小决策树的规模。决策树过大容易受所谓**过分拟合**（overfitting）现象的影响。通过修剪初始决策树的分支，剪枝有助于提高决策树的泛化能力。过分拟合和树剪枝问题将在 4.4 节更详细地讨论。

4.3.6　例子：Web 机器人检测

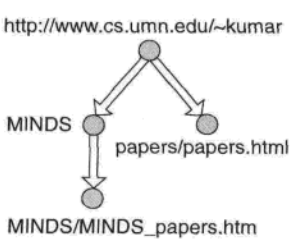
Web使用挖掘就是利用数据挖掘的技术，从Web访问日志中提取有用的模式。这些模式能够揭示站点访问者的一些有趣特性：例如，一个人频繁地访问某个Web站点，并打开介绍同一产品的网页，如果商家提供一些打折或免费运输的优惠，这个人很可能会购买这种商品。

在Web使用挖掘中，重要的是要区分用户访问和Web机器人（Web robot）访问。Web机器人（又称Web爬虫）是一个软件程序，它可以自动跟踪嵌入网页中的超链接，定位和获取Internet上的信息。这些程序安装在搜索引擎的入口，收集索引网页必须的文档。在应用Web挖掘技术分析人类的浏览习惯之前，必须过滤掉Web机器人的访问。

本节介绍如何使用决策树分类法来区分正常的用户访问和由Web机器人产生的访问。输入数据取自Web服务器日志，它的一个样本显示在图4-17a中，每行对应于Web客户（用户或Web机器人）的一个页面访问请求。Web日志记录的字段包括客户端的IP地址、请求的时间戳、请求访问的文档的网址、文档的大小、客户的身份（通过用户代理字段获得）。Web会话是客户在一次网站访问期间发出的请求序列，每个Web会话都可以用有向图来建模，其中结点对应于网页，而有向边对应于连接网页的超链。图4-17b显示Web服务器日志中第一次Web会话的图形表示。

会话	IP 地址	时间戳	请求方法	请求的 Web 页面	协议	状态	字节数	提交者	用户代理
1	160.11.11.11	08/Aug/2004 10:15:21	GET	http://www.cs.umn.edu/~kumar	HTTP/1.1	200	6424		Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
1	160.11.11.11	08/Aug/2004 10:15:34	GET	http://www.cs.umn.edu/~kumar/MINDS	HTTP/1.1	200	41378	http://www.cs.umn.edu/~kumar	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
1	160.11.11.11	08/Aug/2004 10:15:41	GET	http://www.cs.umn.edu/~kumar/MINDS/MINDS_papers.htm	HTTP/1.1	200	1018516	http://www.cs.umn.edu/~kumar/MINDS	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
1	160.11.11.11	08/Aug/2004 10:16:11	GET	http://www.cs.umn.edu/~kumar/papers/papers.html	HTTP/1.1	200	7463	http://www.cs.umn.edu/~kumar	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
2	35.9.2.2	08/Aug/2004 10:16:15	GET	http://www.cs.umn.edu/~steinbac	HTTP/1.0	200	3149		Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7) Gecko/20040616

(a) Web 服务器日志样本



(b) Web 会话图

属性名	描述
totalPages	一次 Web 会话提取的页面总数
ImagePages	一次 Web 会话提取的图像页总数
TotalTime	网站访问者所用的时间
RepeatedAccess	一次 Web 会话多次请求同一页面
ErrorRequest	请求网页的错误
GET	使用 GET 方式提出的请求百分比
POST	使用 POST 方式提出的请求百分比
HEAD	使用 HEAD 方式提出的请求百分比
Breadth	Web 遍历的宽度
Depth	Web 遍历的深度
MultiIP	使用多个 IP 地址的会话
MultiAgent	使用多个代理的会话

(c) Web 机器人检测的导出属性

图 4-17　Web 机器人检测的输入数据

为了对 Web 会话进行分类，需要构造描述每次会话特性的特征。图 4-17c 显示了 Web 机器人检测任务使用的一些特征。显著的特征有遍历的深度和宽度。深度确定请求页面的最大距离，其中距离用自网站入口点的超链数量度量。例如，假设主页 http://www.cs.umn.edu/~kumar 的深度为 0，则 http://www.cs.umn.edu/~kumar/MINDS/MINDS_papers.htm 的深度为 2。根据图 4-17b 中的 Web 图，第一次会话的深度等于 2。宽度属性度量 Web 图的宽度。例如，图 4-17b 中显示的 Web 会话的宽度等于 2。

用于分类的数据集包含 2916 个记录，Web 机器人（类 1）和正常用户（类 2）会话的个数相等，10%的数据用于训练，而 90%的数据用于检验。生成的决策树模型显示在图 4-18 中，该决策树在训练集上的错误率为 3.8%，在检验集上的错误率为 5.3%。

该模型表明可以从以下 4 个方面区分出 Web 机器人和正常用户。

(1) Web 机器人的访问倾向于宽而浅，而正常用户访问比较集中（窄而深）。

(2) 与正常用户不同，Web 机器人很少访问与 Web 文档相关的图片页。

(3) Web 机器人的会话的长度趋于较长，包含了大量请求页面。

(4) Web 机器人更可能对相同的文档发出重复的请求，因为正常用户访问的网页常常会被浏览器保存。

```
决策树:
depth = 1:
| breadth> 7 : class 1
| breadth<= 7:
| | breadth <= 3:
| | | ImagePages> 0.375: class 0
| | | ImagePages<= 0.375:
| | | | totalPages<= 6: class 1
| | | | totalPages> 6:
| | | | | breadth <= 1: class 1
| | | | | breadth > 1: class 0
| | width > 3:
| | | MultiIP = 0:
| | | | ImagePages<= 0.1333: class 1
| | | | ImagePages> 0.1333:
| | | | breadth <= 6: class 0
| | | | breadth > 6: class 1
| | | MultiIP = 1:
| | | | TotalTime <= 361: class 0
| | | | TotalTime > 361: class 1
depth> 1:
| MultiAgent = 0:
| | depth > 2: class 0
| | depth < 2:
| | | MultiIP = 1: class 0
| | | MultiIP = 0:
| | | | breadth <= 6: class 0
| | | | breadth > 6:
| | | | | RepeatedAccess <= 0.322: class 0
| | | | | RepeatedAccess > 0.322: class 1
| MultiAgent = 1:
| | totalPages <= 81: class 0
| | totalPages > 81: class 1
```

图 4-18 Web 机器人检测的决策树模型

4.3.7 决策树归纳的特点

下面是对决策树归纳算法重要特点的总结。

(1) 决策树归纳是一种构建分类模型的非参数方法。换句话说，它不要求任何先验假设，不假定类和其他属性服从一定的概率分布（不像第 5 章介绍的一些技术）。

(2) 找到最佳的决策树是 NP 完全问题。许多决策树算法都采取启发式的方法指导对假设空间的搜索。例如，4.3.5 节中介绍的算法就采用了一种贪心的、自顶向下的递归划分策略建立决策树。

(3) 已开发的构建决策树技术不需要昂贵的计算代价，即使训练集非常大，也可以快速建立模型。此外，决策树一旦建立，未知样本分类非常快，最坏情况下的时间复杂度是 $O(w)$，其中 w 是树的最大深度。

(4) 决策树相对容易解释，特别是小型的决策树。在很多简单的数据集上，决策树的准确率也可以与其他分类算法相媲美。

(5) 决策树是学习离散值函数的典型代表。然而，它不能很好地推广到某些特定的布尔问题。一个著名的例子是奇偶函数，当奇数（偶数）个布尔属性为真时其值为 0（1）。对这样的函数准确建模需要一棵具有 2^d 个结点的满决策树，其中 d 是布尔属性的个数（见本章习题 1）。

(6) 决策树算法对于噪声的干扰具有相当好的鲁棒性，采用避免过分拟合的方法之后尤其如此。避免过分拟合的方法将在 4.4 节介绍。

(7) 冗余属性不会对决策树的准确率造成不利的影响。一个属性如果在数据中它与另一个属性是强相关的，那么它是冗余的。在两个冗余的属性中，如果已经选择其中一个作为用于划分的属性，则另一个将被忽略。然而，如果数据集中含有很多不相关的属性（即对分类任务没有用的属性），则某些不相关属性可能在树的构造过程中偶然被选中，导致决策树过于庞大。通过在预处理阶段删除不相关属性，特征选择技术能够帮助提高决策树的准确率。我们将在 4.4.3 节考察不相关属性过多的问题。

(8) 由于大多数的决策树算法都采用自顶向下的递归划分方法，因此沿着树向下，记录会越来越少。在叶结点，记录可能太少，对于叶结点代表的类，不能做出具有统计意义的判决，这就是所谓的**数据碎片**（data fragmentation）问题。解决该问题的一种可行的方法是，当样本数小于某个特定阈值时停止分裂。

(9) 子树可能在决策树中重复多次，如图 4-19 所示，这使得决策树过于复杂，并且可能更难解释。当决策树的每个内部结点都依赖单个属性测试条件时，就会出现这种情形。由于大多数的决策树算法都采用分治划分策略，因此在属性空间的不同部分可以使用相同的测试条件，从而导致子树重复问题。

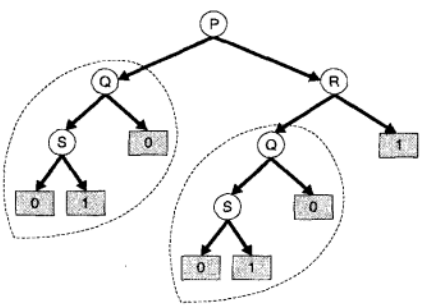


图 4-19　子树重复问题。相同的子树可能出现在不同的分支

(10) 迄今为止，本章介绍的测试条件每次都只涉及一个属性。这样，可以将决策树的生长过程看成划分属性空间为不相交的区域的过程，直到每个区域都只包含同一类的记录（见图

4-20)。两个不同类的相邻区域之间的边界称作**决策边界**(decision boundary)。由于测试条件只涉及单个属性，因此决策边界是直线，即平行于“坐标轴”，这就限制了决策树对连续属性之间复杂关系建模的表达能力。图 4-21 显示了一个数据集，使用一次只涉及一个属性的测试条件的决策树算法很难有效地对它进行分类。

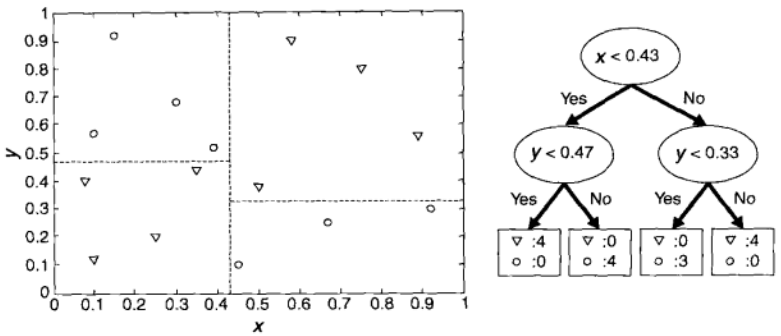


图 4-20 二维数据集的决策树及其决策边界示例

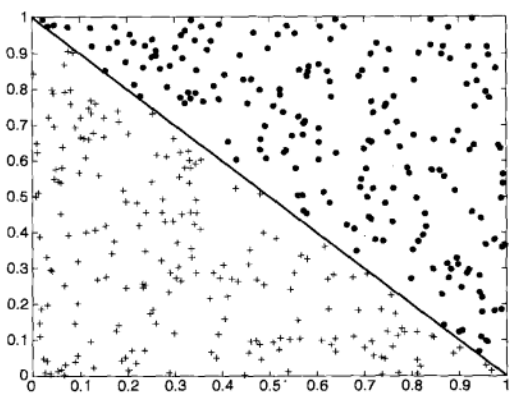


图 4-21 使用仅涉及单个属性的测试条件不能有效划分的数据集的例子

斜决策树(oblique decision tree)可以克服以上的局限，因为它允许测试条件涉及多个属性。图 4-21 中的数据集可以很容易地用斜决策树表示，该斜决策树只有一个结点，其测试条件为：

$$x + y < 1$$

尽管这种技术具有更强的表达能力，并且能够产生更紧凑的决策树，但是为给定的结点找出最佳测试条件的计算可能是相当复杂的。

构造归纳(constructive induction)提供另一种将数据划分成齐次非矩形区域的方法(见 2.3.5 节)，该方法创建复合属性，代表已有属性的算术或逻辑组合。新属性提供了更好的类区分能力，并在决策树归纳之前就增广到数据集中。与斜决策树不同，构造归纳不需要昂贵的花费，因为在构造决策树之前，它只需要一次性地确定属性的所有相关组合。相比之下，在扩展每个内部结点时，斜决策树都需要动态地确定正确的属性组合。然而，构造归纳会产生冗余的属性，因为新创建的属性是已有属性的组合。

(11) 研究表明不纯性度量方法的选择对决策树算法的性能影响很小，这是因为许多度量方法相互之间都是一致的，如图4-13所示。实际上，树剪枝对最终决策树的影响比不纯性度量的选择的影响更大。

4.4　模型的过分拟合

分类模型的误差大致分为两种：**训练误差**（training error）和**泛化误差**（generalization error）。训练误差也称**再代入误差**（resubstitution error）或**表现误差**（apparent error），是在训练记录上误分类样本比例，而泛化误差是模型在未知记录上的期望误差。

回顾 4.2 节，一个好的分类模型不仅要能够很好地拟合训练数据，而且对未知样本也要能准确地分类。换句话说，一个好的分类模型必须具有低训练误差和低泛化误差。这一点非常重要，因为对训练数据拟合度过高的模型，其泛化误差可能比具有较高训练误差的模型高。这种情况就是所谓的模型过分拟合。

二维数据过分拟合的例子　关于过分拟合问题的具体例子，考虑图 4-22 所示的二维数据集。数据集中的数据点属于两个类，分别标记为类“o”和类“+”，类“o”的数据点由三个高斯分布混合产生，而类“+”的数据点用一个均匀分布产生。数据集中，总共有 1 200 个数据点是属于类“o”，1 800 个属于类“+”，其中 30%的点用于训练，剩下的 70%用于检验。对训练集使用以 Gini 指标作为不纯性度量的决策树分类法。为了研究过分拟合的影响，对初始的、完全生长的决策树进行了不同程度的剪枝。图 4-23 显示了决策树的训练误差和检验误差。

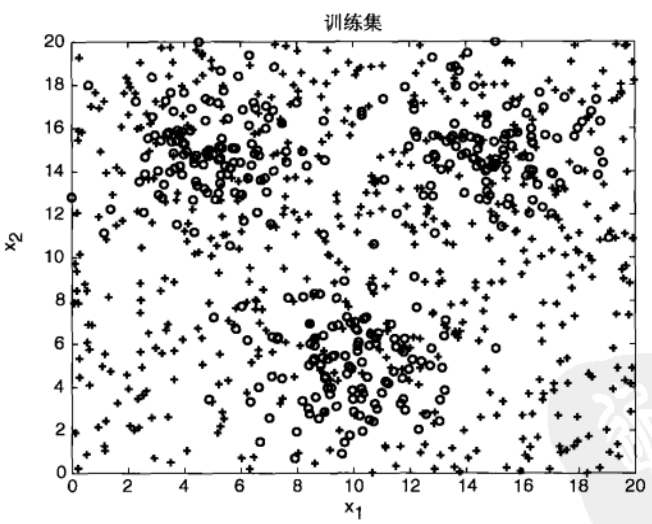


图 4-22　具有两个类的数据集的例子

注意，当决策树很小时，训练和检验误差都很大，这种情况称作**模型拟合不足**（model underfitting）。出现拟合不足的原因是模型尚未学习到数据的真实结构，因此，模型在训练集和检验集上的性能都很差。随着决策树中结点数的增加，模型的训练误差和检验误差都会随之降低。然而，一旦树的规模变得太大，即使训练误差还在继续降低，但是检验误差开始增大，这种现象称为**模型过分拟合**（model overfitting）。

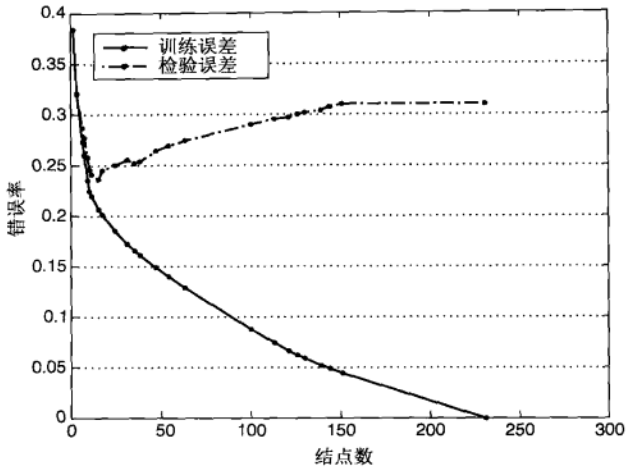
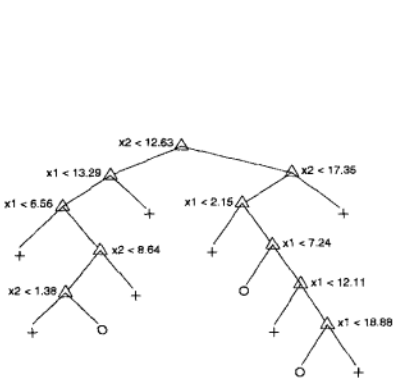
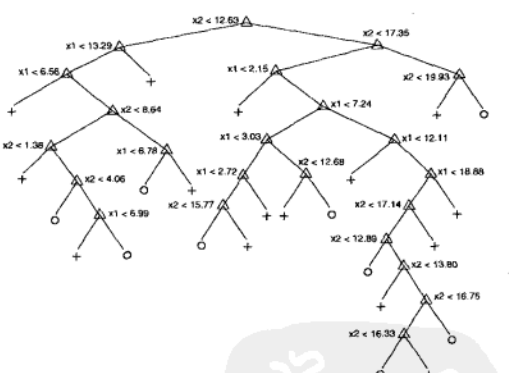


图 4-23 训练误差和检验误差

为了理解过分拟合现象，注意模型的训练误差随模型的复杂度增加而降低，例如，可以扩展树的叶结点，直到它完全拟合训练数据。虽然这样一棵复杂的决策树的训练误差为 0，但是检验误差可能很大，因为该树可能包含这样的结点，它们偶然地拟合训练数据中某些噪声。这些结点降低了决策树的性能，因为它们不能很好的泛化到检验样本。图 4-24 是两棵具有不同结点数的决策树，结点数少的决策树具有较高的训练误差，但是与更复杂的树相比，它具有较低的检验误差。



(a) 包含 11 个叶结点的决策树



(b) 包含 24 个叶结点的决策树

图 4-24 具有不同模型复杂度的决策树

过分拟合与拟合不足是两种与模型复杂度有关的异常现象。本节余下的部分将继续讨论造成模型过分拟合的一些潜在因素。

4.4.1 噪声导致的过分拟合

考虑表 4-3 和表 4-4 中哺乳动物的分类问题的训练数据集合和检验数据集合。十个训练记录中有两个被错误地标记：蝙蝠和鲸被错误地标记为非哺乳类动物，而不是哺乳类动物。

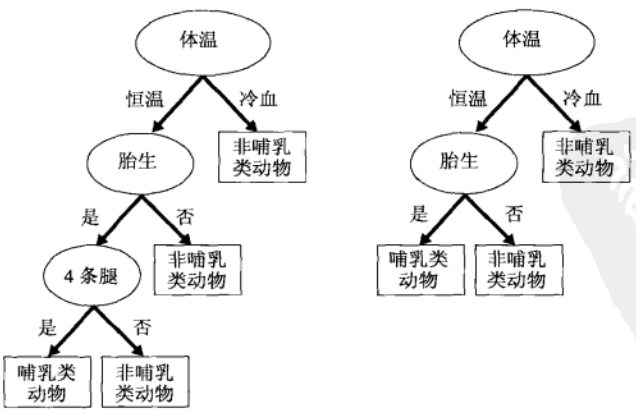
表 4-3　哺乳类动物分类的训练数据集样本。打星号的类标号代表错误标记的记录

名称	体温	胎生	4 条腿	冬眠	类标号
豪猪	恒温	是	是	是	是
猫	恒温	是	是	否	是
蝙蝠	恒温	是	否	是	否*
鲸	恒温	是	否	否	否*
蝾螈	冷血	否	是	是	否
科莫多巨蜥	冷血	否	是	否	否
蟒蛇	冷血	否	否	是	否
鲑鱼	冷血	否	否	否	否
鹰	恒温	否	否	否	否
虹鳉	冷血	是	否	否	否

表 4-4　哺乳类动物分类的检验数据集样本

名称	体温	胎生	4 条腿	冬眠	类标号
人	恒温	是	否	否	是
鸽子	恒温	否	否	否	否
象	恒温	是	是	否	是
豹纹鲨	冷血	是	否	否	否
海龟	冷血	否	是	否	否
企鹅	冷血	否	否	否	否
鳗	冷血	否	否	否	否
海豚	恒温	是	否	否	是
针鼹	恒温	否	是	是	是
希拉毒蜥	冷血	否	是	是	否

完全拟合训练数据的决策树显示在图 4-25a 中。虽然该树的训练误差为 0，但它在检验数据上的误差高达 30%。人和海豚都被误分类为非哺乳类动物，因为它们在属性体温、胎生、4 条腿上的属性值与训练数据中被错误标记的样本属性值相同。另一方面，针鼹是个例外，其检验记录中的类标号与训练集中相似的记录的类标号相反。例外导致的错误是不可避免的，它设定了分类器可以达到的最小错误率。



(a) 模型 M_1　　(b) 模型 M_2

图 4-25　根据表 4-3 中的数据集建立的决策树

相反，图 4-25b 中决策树 M_2 具有较低的检验误差（10%），尽管它的训练误差较高（20%）。很明显，决策树 M_1 过分拟合了训练数据，因为存在一个更简单、但在检验数据集上具有更低检验误差的模型。模型 M_1 中的属性测试条件 4 条脚具有欺骗性，因为它拟合了误标记的训练记录，导致了对检验集中记录的误分类。

4.4.2 缺乏代表性样本导致的过分拟合

根据少量训练记录做出分类决策的模型也容易受过分拟合的影响。由于训练数据缺乏具有代表性的样本，在没有多少训练记录的情况下，学习算法仍然继续细化模型就会产生这样的模型。下面举例说明。

考虑表4-5中的五个训练记录，表中所有的记录都是正确标记的，对应的决策树在图4-26中。尽管它的训练误差为0，但是它的检验误差却高达30%。

表 4-5 哺乳动物分类的训练集样本

名称	体温	胎生	4 条腿	冬眠	类标号
蝾螈	冷血	否	是	是	否
虹鳉	冷血	是	否	否	否
鹰	恒温	否	否	否	否
弱夜鹰	恒温	否	否	是	否
鸭嘴兽	恒温	否	是	是	是

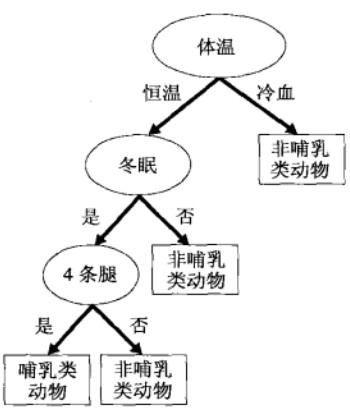


图 4-26 根据表 4-5 中的数据集建立的决策树

人、大象和海豚都被误分类，因为决策树把恒温但不冬眠的脊柱动物划分为非哺乳动物。决策树做出这样的分类决策是因为只有一个训练记录（鹰）具有这些特性。这个例子清楚地表明，当决策树的叶结点没有足够的代表性样本时，很可能做出错误的预测。

4.4.3 过分拟合与多重比较过程

模型的过分拟合可能出现在使用所谓的多重比较过程（multiple comparison procedure）的学习算法中。为了理解多重比较过程，考虑预测未来 10 个交易日股市是升还是降的任务。如果股票分析家简单地随机猜测，则对任意交易日预测正确的概率是 0.5，然而，10 次猜测至少正确预测 8 次的概率是：

$$\frac{C_{10}^{8}+C_{10}^{9}+C_{10}^{10}}{2^{10}}=0.0547$$

这看起来不大可能。

假设我们想从50个股票分析家中选择一个投资顾问，策略是选择在未来的10个交易日做出最多正确预测的分析家。该策略的缺点是，即使所有的分析家都用随机猜测做出预测，至少有一个分析家做出8次正确预测的概率是：

$$1-(1-0.0547)^{50}=0.9399$$

这相当高。尽管每个分析家做出8次正确预测的概率很低，但是把他们放在一起，找到一个能够做出8次正确预测的分析家的概率却很高。此外，不能保证这样的分析家以后还能通过随机猜测继续做出准确的预测。

多重比较过程与模型过分拟合有什么关系呢？许多学习算法都利用一个独立的候选集$\{\gamma_i\}$，然后从中选取最大化给定标准的$\gamma_{\max}$。算法将把$\gamma_{\max}$添加到当前模型中，以提高模型的整体性能。重复这一过程，直到没有进一步的提高。例如，在决策树增长过程中，可以进行多种测试，以确定哪个属性能够最好地划分训练数据，只要观察到的改进是统计显著的，就选取导致最佳划分的属性来扩展决策树。

设T_0是初始决策树，T_x是插入属性x的内部结点后的决策树。原则上，如果观察到的增益$\Delta(T_0, T_x)$大于某个预先定义的阈值α，就可以将x添加到树中。如果只有一个属性测试条件，则可以通过选择足够大的阈值α来避免插入错误的结点。然而，在实践中，可用的属性测试条件不止一个，并且决策树算法必须从候选集$\{x_1, x_2, \cdots, x_k\}$中选择最佳属性$x_{\max}$来划分数据。在这种情况下，算法实际上是使用多重比较过程来决定是否需要扩展决策树。更具体地说，这是测试$\Delta(T_0, T_{x_{\max}}) > \alpha$，而不是测试$\Delta(T_0, T_x) > \alpha$。随着候选个数$k$的增加，找到$\Delta(T_0, T_{x_{\max}}) > \alpha$的几率也在增大。除非根据$k$修改增益函数$\Delta$或阈值$\alpha$，否则算法会不经意间在模型上增加一些欺骗性的结点，导致模型过分拟合。

当选择属性$x_{\max}$的训练记录集很小时，这种影响就变得更加明显，因为当训练记录较少时，函数$\Delta(T_0, T_{x_{\max}})$的方差会很大。因此，当训练记录很少时，找到$\Delta(T_0, T_{x_{\max}}) > \alpha$的概率就增大了。决策树增长到一定深度就会经常发生这种情形，这样会降低结点所覆盖的记录数，提高了添加不必要结点的可能性。大量的候选属性和少量的训练记录最后导致了模型的过分拟合。

4.4.4　泛化误差估计

虽然过分拟合的主要原因一直是个争辩的话题，大家还是普遍同意模型的复杂度对模型的过分拟合有影响，如图4-23所示。问题是，如何确定正确的模型复杂度？理想的复杂度是能产生最低泛化误差的模型的复杂度。然而，在建立模型的过程中，学习算法只能访问训练数据集（见图4-3），对检验数据集，它一无所知，因此也不知道所建立的决策树在未知记录上的性能。我们所能做的就是估计决策树的泛化误差。本节提供一些估计泛化误差的方法。

1. 使用再代入估计

再代入估计方法假设训练数据集可以很好地代表整体数据，因而，可以使用训练误差（又称再代入误差）提供对泛化误差的乐观估计。在这样的前提下，决策树归纳算法简单地选择产生最低训练误差的模型作为最终的模型。然而，训练误差通常是泛化误差的一种很差的估计。

例 4.1 考虑图 4-27 中的二叉决策树。假设两棵决策树都由相同的训练数据产生，并且都根据每个叶结点多数类做出分类决策。注意，左边的树 T_L 复杂一些，它扩展了右边决策树 T_R 的某些叶结点。左决策树的训练误差是 $e(T_L) = 4/24 = 0.167$，而右决策树的训练误差是 $e(T_R) = 6/24 = 0.25$。根据再代入估计，左决策树要优于右决策树。 □

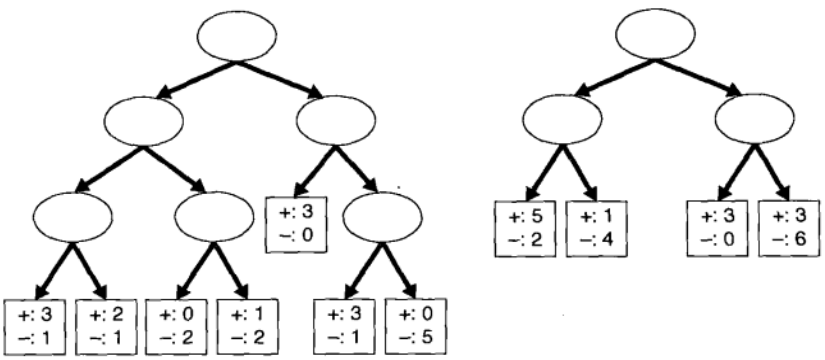


图 4-27 由相同的训练数据产生的两棵决策树

2. 结合模型复杂度

如前所述，模型越是复杂，出现过分拟合的几率就越高，因此，我们更喜欢采用较为简单的模型。这种策略与应用众所周知的**奥卡姆剃刀**（Occam's razor）或**节俭原则**（principle of parsimony）一致。

定义 4.2 **奥卡姆剃刀**：给定两个具有相同泛化误差的模型，较简单的模型比较复杂的模型更可取。

奥卡姆剃刀是很直观的原则，因为复杂模型中的附加成分很大程度上是完全对偶然的拟合。用爱因斯坦的话来说，“所有事情都应该尽可能简单，但不是简化。”下面我们介绍两种把模型复杂度与分类模型评估结合在一起的方法。

悲观误差评估 第一种方法明确使用训练误差与模型复杂度罚项（penalty term）的和计算泛化误差。结果泛化误差可以看作模型的悲观误差估计（pessimistic error estimate）。例如，设 $n(t)$ 是结点 t 分类的训练记录数，$e(t)$是被误分类的记录数。决策树 T 的悲观误差估计 $e_g(T)$可以用下式计算：

$$e_g(T) = \frac{\sum_{i=1}^{k}[e(t_i) + \Omega(t_i)]}{\sum_{i=1}^{k} n(t_i)} = \frac{e(T) + \Omega(T)}{N_t}$$

其中，k 是决策树的叶结点数，$e(T)$决策树的总训练误差，N_t 是训练记录数，$\Omega(t_i)$是每个结点 t_i 对应的罚项。

例 4.2 考虑图 4-27 中的二叉决策树。如果罚项等于 0.5，左边的决策树的悲观误差估计为：

$$e_g(T_L)=\frac{4+7\times0.5}{24}=\frac{7.5}{24}=0.3125$$

右边的决策树的悲观误差估计为：

$$e_g(T_R)=\frac{6+4\times0.5}{24}=\frac{8}{24}=0.3333$$

这样，左边的决策树比右边的决策树具有更好的悲观误差估计。对二叉树来说，0.5 的罚项意味着只要至少能够改善一个训练记录的分类，结点就应当扩展，因为扩展一个结点等价于总误差增加 0.5，代价比犯一个训练错误小。

如果对于所有的结点 t，$\Omega(t)=1$，左边的决策树的悲观误差估计为 $e_g(T_L)=11/24=0.458$，右边的决策树的悲观误差估计为 $e_g(T_R)=10/24=0.417$。因此，右边的决策树比左边的决策树具有更好的悲观错误率。这样，除非能够减少一个以上训练记录的误分类，否则结点不应当扩展。□

最小描述长度原则　另一种结合模型复杂度的方法是基于称作最小描述长度（minimum description length, MDL）原则的信息论方法。为了解释说明该原则，考虑图 4-28 中的例子。在该例中，A 和 B 都是已知属性 **x** 值的给定记录集。另外，A 知道每个记录的确切类标号，而 B 却不知道这些信息。B 可以通过要求 A 顺序传送类标号而获得每个记录的分类。一条消息需要 $\Theta(n)$比特的信息，其中 n 是记录总数。

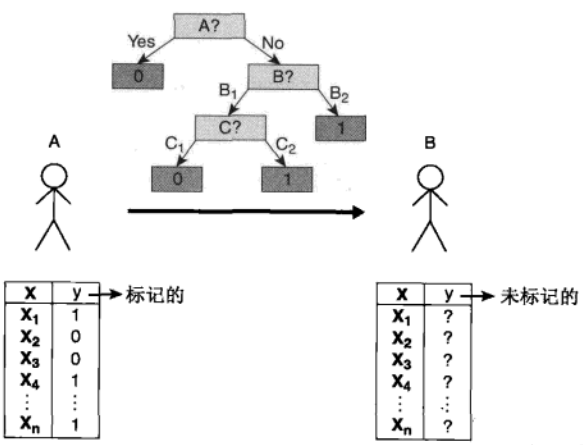


图 4-28　最小描述长度（MDL）原则

另一种可能是，A 决定建立一个分类模型，概括 **x** 和 y 之间的关系。在传送给 B 前，模型用压缩形式编码。如果模型的准确率是 100%，那么传输的代价就等于模型编码的代价。否则，A 还必须传输哪些记录被模型错误分类信息。传输的总代价是：

$$Cost(model, data)=Cost(model)+Cost(data\mid model) \tag{4-9}$$

其中，等式右边的第一项是模型编码的开销，而第二项是误分类记录编码的开销。根据 MDL 原则，我们寻找最小化开销函数的模型。本章习题 9 给出了一个如何计算决策树总描述长度的例子。

3. 估计统计上界

泛化误差也可以用训练误差的统计修正来估计。因为泛化误差倾向于比训练误差大，所以统

计修正通常是计算训练误差的上界，考虑到达决策树一个特定叶结点的训练记录数。例如，决策树算法 C4.5 中，假定每个叶结点上的错误服从二项分布。为了计算泛化误差，我们需要确定训练误差的上限，在下面的例子中解释说明。

例 4.3 考虑图 4-27 所示的二叉决策树的最左分支，注意，T_R 中的最左叶结点被扩展为 T_L 中的两个子女结点。在划分前，该结点的错误率是 2/7 = 0.286。用正态分布近似二项分布，可以推导出错误率 e 的上界是：

$$e_{\text{upper}}(N, e, \alpha) = \frac{e + \frac{z_{\alpha/2}^2}{2N} + z_{\alpha/2}\sqrt{\frac{e(1-e)}{N} + \frac{z_{\alpha/2}^2}{4N^2}}}{1 + \frac{z_{\alpha/2}^2}{N}} \tag{4-10}$$

其中，α是置信水平，$z_{\alpha/2}$是标准正态分布的标准化值，而 N 是计算 e 的训练记录总数。将$\alpha = 25\%$，$N = 7$，$e = 2/7$ 代入，错误率的上限是 $e_{\text{upper}}(7, 2/7, 0.25) = 0.503$，对应于 $7 \times 0.503 = 3.521$ 个错误。如果我们扩展结点为 T_L 中的子女结点，子女结点的训练误差分别为 1/4=0.250，1/3 = 0.333。使用公式（4-10），错误率上限分别是 $e_{\text{upper}}(4, 1/4, 0.25) = 0.537$，$e_{\text{upper}}(3, 1/3, 0.25) = 0.650$。子女结点的总训练误差是 $4 \times 0.537 + 3 \times 0.650 = 4.098$，大于 T_R 中相应结点的估计误差。 □

4. 使用确认集

在该方法中，不是用训练集估计泛化误差，而是把原始的训练数据集分为两个较小的子集，一个子集用于训练，而另一个称作确认集，用于估计泛化误差。典型的做法是，保留 2/3 的训练集来建立模型，剩余的 1/3 用作误差估计。

该方法常常用于通过参数控制获得具有不同复杂度模型的分类技术。通过调整学习算法中的参数（如决策树中剪枝的程度），直到学习算法产生的模型在确认集上达到最低的错误率，可以估计最佳模型的复杂度。虽然该方法为评估模型在未知样本上的性能提供了较好办法，但用于训练的记录减少了。

4.4.5 处理决策树归纳中的过分拟合

在前面章节中，我们介绍了一些估计分类模型泛化误差的方法。对于泛化误差可靠的估计能让学习算法搜索到准确的模型，而且不会对训练数据过分拟合。本节介绍两种在决策树归纳上避免过分拟合的策略。

先剪枝（提前终止规则） 在这种方法中，树增长算法在产生完全拟合整个训练数据集的完全增长的决策树之前就停止决策树的生长。为了做到这一点，需要采用更具限制性的结束条件，例如，当观察到的不纯性度量的增益（或估计的泛化误差的改进）低于某个确定的阈值时就停止扩展叶结点。这种方法的优点在于避免产生过分拟合训练数据的过于复杂的子树，然而，很难为提前终止选取正确的阈值。阈值太高将导致拟合不足的模型，而阈值太低就不能充分地解决过分拟合的问题。此外，即便使用已有的属性测试条件得不到显著的增益，接下来的划分也可能产生较好的子树。

后剪枝 在该方法中，初始决策树按照最大规模生长，然后进行剪枝的步骤，按照自底向上的方式修剪完全增长的决策树。修剪有两种做法：(1)用新的叶结点替换子树，该叶结点的类标号

由子树下记录中的多数类确定；或者(2)用子树中最常使用的分支代替子树。当模型不能再改进时终止剪枝步骤。与先剪枝相比，后剪枝技术倾向于产生更好的结果，因为不像先剪枝，后剪枝是根据完全增长的决策树做出的剪枝决策，先剪枝则可能过早终止决策树的生长。然而，对于后剪枝，当子树被剪掉后，生长完全决策树的额外的计算就被浪费了。

图4-29展示了4.3.6节Web机器人检测的简化后的决策树模型。注意根在depth = 1的子树已经用涉及属性ImagePages的一个分支替换，这种方法又称**子树提升**（subtree raising）；depth > 1且MultiAgent = 0的子树被类标号为0的叶结点替换，这种方法称作**子树替换**（subtree replacement）；depth > 1且MultiAgent = 1的子树完整保留。

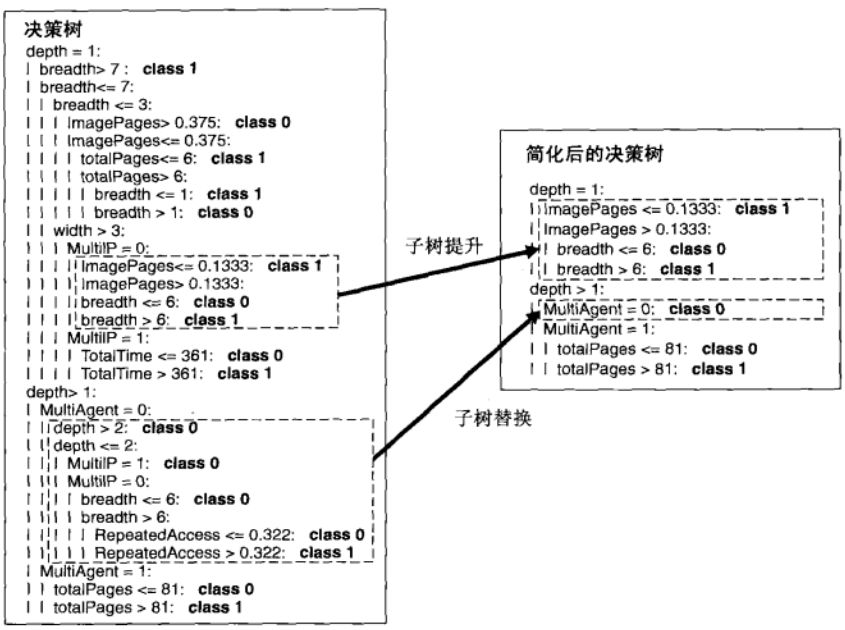


图4-29　Web机器人检测决策树的后剪枝

4.5　评估分类器的性能

4.4.4节中介绍了几种在训练过程中估计模型泛化误差的方法。估计误差有助于学习算法进行**模型选择**（model selection），即找到一个具有合适复杂度、不易发生过分拟合的模型。模型一旦建立，就可以应用到检验数据集上，预测未知记录的类标号。

测试模型在检验集上的性能是有用的，因为这样的测量给出模型泛化误差的无偏估计。在检验集上计算出的准确率或错误率可以用来比较不同分类器在相同领域上的性能。然而，为了做到这一点，检验记录的类标号必须是已知的。本节回顾一些常用的评估分类器性能的方法。

4.5.1　保持方法

在保持（Holdout）方法中，将被标记的原始数据划分成两个不相交的集合，分别称为训练集和检验集。在训练数据集上归纳分类模型，在检验集上评估模型的性能。训练集和检验集的划

分比例通常根据分析家的判断（例如，50-50，或者 2/3 作为训练集、1/3 作为检验集）。分类器的准确率根据模型在检验集上的准确率估计。

保持方法有一些众所周知的局限性。第一，用于训练的被标记样本较少，因为要保留一部分记录用于检验，因此，建立的模型不如使用所有被标记样本建立的模型好。第二，模型可能高度依赖于训练集和检验集的构成。一方面，训练集越小，模型的方差越大，另一方面，如果训练集太大，根据用较小的检验集估计的准确率又不太可靠。这样的估计具有很宽的置信区间。最后，训练集和检验集不再是相互独立的。因为训练集和检验集来源于同一个数据集，在一个子集中超出比例的类在另一个子集就低于比例，反之亦然。

4.5.2 随机二次抽样

可以多次重复保持方法来改进对分类器性能的估计，这种方法称作随机二次抽样（random subsampling）。设 acc_i 是第 i 次迭代的模型准确率，总准确率是 $acc_{sub} = \sum_{i=1}^{k} acc_i / k$。随机二次抽样也会遇到一些与保持方法同样的问题，因为在训练阶段也没有利用尽可能多的数据。并且，由于它没有控制每个记录用于训练和检验的次数，因此，有些用于训练的记录使用的频率可能比其他记录高很多。

4.5.3 交叉验证

替代随机二次抽样的一种方法是交叉验证（cross-validation）。在该方法中，每个记录用于训练的次数相同，并且恰好检验一次。为了解释该方法，假设把数据分为相同大小的两个子集，首先，我们选择一个子集作训练集，而另一个作检验集，然后交换两个集合的角色，原先作训练集的现在做检验集，反之亦然，这种方法叫二折交叉验证。总误差通过对两次运行的误差求和得到。在这个例子中，每个样本各作一次训练样本和检验样本。k 折交叉验证是对该方法的推广，把数据分为大小相同的 k 份，在每次运行，选择其中一份作检验集，而其余的全作为训练集，该过程重复 k 次，使得每份数据都用于检验恰好一次。同样，总误差是所有 k 次运行的误差之和。k 折交叉验证方法的一种特殊情况是令 $k = N$，其中 N 是数据集的大小，在这种所谓**留一**（leave-one-out）方法中，每个检验集只有一个记录。该方法的优点是使用尽可能多的训练记录，此外，检验集之间是互斥的，并且有效地覆盖了整个数据集；该方法的缺点是整个过程重复 N 次，计算上开销很大，此外，因为每个检验集只有一个记录，性能估计度量的方差偏高。

4.5.4 自助法

迄今为止，我们介绍的方法都是假定训练记录采用不放回抽样，因此，训练集和检验集都不包含重复记录。在自助（bootstrap）方法中，训练记录采用有放回抽样，即已经选作训练的记录将放回原来的记录集中，使得它等机率地被重新抽取。如果原始数据有 N 个记录，可以证明，平均来说，大小为 N 的自助样本大约包含原始数据中 63.2%的纪录。这是因为一个记录被自助抽样抽取的概率是 $1 - (1 - 1/N)^N$，当 N 充分大时，该概率逐渐逼近 $1 - e^{-1} = 0.632$。没有抽中的记录就成为检验集的一部分，将训练集建立的模型应用到检验集上，得到自助样本准确率的一个估计 ε_i。抽样过程重复 b 次，产生 b 个自助样本。

按照如何计算分类器的总准确率，有几种不同的自助抽样法。常用的方法之一是**.632 自助**（.632 bootstrap），它通过组合每个自助样本的准确率（ε_i）和由包含所有标记样本的训练集计算

的准确率（acc_s）计算总准确率（acc_{boot}）：

$$acc_{\text{boot}}=\frac{1}{b}\sum_{i=1}^{b}(0.632\times\varepsilon_i+0.368\times acc_s) \tag{4-11}$$

4.6　比较分类器的方法

比较不同分类器的性能，以确定在给定的数据集上哪个分类器效果更好是很有用的。但是，依据数据集的大小，两个分类器准确率上的差异可能不是统计显著的。本节介绍一些统计检验方法，可以用来比较不同模型和分类器的性能。

为了更好地解释，考虑一对分类模型 M_A 和 M_B。假设 M_A 在包含 30 个记录的检验集上的准确率达到 85%，而 M_B 在包含 5 000 个记录的不同检验集上达到 75%的准确率。根据这些信息，M_A 比 M_B 好吗？

上面的例子提出了涉及性能度量的统计显著性的两个关键问题。

(1) 尽管 M_A 的准确率比 M_B 高，但是它是在较小的检验集上检验的。M_A 的准确率的置信程度有多高？

(2) 可以把准确率的差解释为检验集的复合的变差吗？

第一个问题与估计给定模型准确率的置信区间有关，第二个问题涉及检验观测离差的统计显著性。本节余下部分将详细考察这些问题。

4.6.1　估计准确度的置信区间

为确定置信区间，需要建立支配准确率度量的概率分布。本节介绍一种方法，通过将分类任务用二项式实验建模来推导置信区间。二项式实验的特性如下。

(1) 实验由 N 个独立的试验组成，其中每个试验有两种可能的结果：成功或失败。

(2) 每个试验成功的概率 p 是常数。

二项式实验的一个例子是统计 N 次抛硬币正面朝上的次数。如果 X 是 N 次试验观察到的成功次数，则 X 取一个特定值 v 的概率由均值为 Np、方差为 $Np(1-p)$的二项分布给出：

$$P(X=v)=C_N^v p^v(1-p)^{N-v}$$

例如，如果硬币是均匀的（$p=0.5$），抛 50 次硬币，正面朝上 20 次的概率是：

$$P(X=20)=C_{50}^{20}0.5^{20}(1-0.5)^{30}=0.0419$$

如果该实验重复多次，正面朝上的期望平均次数为 $50\times0.5=25$，方差为 $50\times0.5\times0.5=12.5$。

预测检验记录类标号的任务也可以看作是二项式实验。给定一个包含 N 个记录的检验集，令 X 是被模型正确预测的记录数，p 是模型真正准确率。通过把预测任务用二项式实验建模，X 服从均值为 Np、方差为 $Np(1-p)$的二项分布。可以证明经验准确率 $acc=X/N$ 也是均值为 p，方差为 $p(1-p)/N$ 的二项分布（见习题 12）。尽管可以用二项分布来估计 acc 的置信区间，但是当 N 充分大时，通常用正态分布来近似。根据正态分布，可以推导出 acc 的置信区间为：

$$P\left(-Z_{\alpha/2}\leqslant\frac{acc-p}{\sqrt{p(1-p)/N}}\leqslant Z_{1-\alpha/2}\right)=1-\alpha \tag{4-12}$$

其中 $Z_{\alpha/2}$ 和 $Z_{1-\alpha/2}$ 分别是在置信水平 $(1-\alpha)$下由标准正态分布得到的上界和下界。因为标准正

态分布关于 $Z=0$ 对称，于是我们有 $Z_{\alpha/2}=Z_{1-\alpha/2}$。重新整理不等式，得到 p 的置信区间如下：

$$\frac{2\times N\times acc+Z_{\alpha/2}^2\pm Z_{\alpha/2}\sqrt{Z_{\alpha/2}^2+4Nacc-4Nacc^2}}{2(N+Z_{\alpha/2}^2)} \tag{4-13}$$

下表给出了在不同置信水平下 $Z_{\alpha/2}$ 的值：

$1-\alpha$	0.99	0.98	0.95	0.9	0.8	0.7	0.5
$Z_{\alpha/2}$	2.58	2.33	1.96	1.65	1.28	1.04	0.67

例 4.4 考虑一个模型，它在 100 个检验记录上具有 80%的准确率。在 95%的置信水平下，模型的真实准确率的置信区间是什么？根据上面的表，95%的置信水平对应于 $Z_{\alpha/2}=1.96$。将它代入公式（4-13）得到置信区间在 71.1%和 86.7%之间。下表给出了随着记录数 N 的增大所产生的置信区间：

N	20	50	100	500	1000	5000
置信	0.584	0.670	0.711	0.763	0.774	0.789
区间	−0.919	−0.888	−0.867	−0.833	−0.824	−0.811

注意，随着 N 的增大，置信区间变得更加紧凑。 □

4.6.2 比较两个模型的性能

考虑一对模型 M_1 和 M_2，它们在两个独立的检验集 D_1 和 D_2 上进行评估，令 n_1 是 D_1 中的记录数，n_2 是 D_2 中的记录数。另外，假设 M_1 在 D_1 上的错误率为 e_1，M_2 在 D_2 上的错误率为 e_2。目标是检验 e_1 与 e_2 的观察差是否是统计显著的。

假设 n_1 和 n_2 都充分大，e_1 和 e_2 可以使用正态分布来近似。如果用 $d=e_1-e_2$ 表示错误率的观测差，则 d 服从均值为 d_t（其实际差）、方差为 σ_d^2 的正态分布。d 的方差为：

$$\sigma_d^2=\hat{\sigma}_d^2=\frac{e_1(1-e_1)}{n_1}+\frac{e_2(1-e_2)}{n_2} \tag{4-14}$$

其中 $e_1(1-e_1)/n_1$ 和 $e_2(1-e_2)/n_2$ 是错误率的方差。最后，在置信水平$(1-\alpha)$ %下，可以证明实际差 d_t 的置信区间由下式给出：

$$d_t=d\pm z_{\alpha/2}\hat{\sigma}_d \tag{4-15}$$

例 4.5 考虑本节开始所描述的问题。模型 M_A 在 $N_1=30$ 个检验记录上的错误率 $e_1=0.15$，而 M_B 在 $N_2=5000$ 个检验记录上的错误率 $e_2=0.25$。错误率的观察差 $d=|0.15-0.25|=0.1$。在这个例子中，我们使用双侧检验来检查 $d_t=0$ 还是 $d_t\neq 0$。错误率观察差的估计方差计算如下：

$$\hat{\sigma}_d^2=\frac{0.15(1-0.15)}{30}+\frac{0.25(1-0.25)}{5000}=0.0043$$

或 $\hat{\sigma}_d=0.0655$。把该值代入公式（4-15），我们得到在 95%的置信水平下，d_t 置信区间如下：

$$d_t=0.1\pm 1.96\times 0.0655=0.1\pm 0.128$$

由于该区间跨越了值 0，我们可以断言在 95%的置信水平下，该观察差不是统计显著的。 □

在什么置信水平下，我们可以拒绝假设 $d_t = 0$？为此，需要确定 $Z_{\alpha/2}$ 的值，使得 d_t 的置信区间不会跨越值 0，可以颠倒前面的计算，找出使不等式 $d > Z_{\alpha/2}\hat{\sigma}_d$ 成立的 $Z_{\alpha/2}$ 的值。代入 d 和 $\hat{\sigma}_d$ 的值，得到 $Z_{\alpha/2} < 1.527$，当 $(1-\alpha) < 0.936$ 时这个值第一次出现（对于双侧检验）。该结果表明在 93.6%或者更低的置信水平下，我们可以拒绝原假设。

4.6.3　比较两种分类法的性能

假设我们想用 k 折交叉验证的方法比较两种分类法的性能。首先，把数据集 D 划分为 k 个大小相等部分，然后，使用每种分类法，在 $k-1$ 份数据上构建模型，并在剩余的划分上进行检验，这个步骤重复 k 次，每次使用不同的划分进行检验。

令 M_{ij} 表示分类技术 L_i 在第 j 次迭代产生的模型，注意，每对模型 M_{1j} 和 M_{2j} 在相同的划分 j 上进行检验。用 e_{1j} 和 e_{2j} 分别表示它们的错误率，它们在第 j 折上的错误率之差可以记作 $d_j = e_{1j} - e_{2j}$。如果 k 充分大，则 d_j 服从于均值为 d_t^{cv}（错误率的真实差）、方差为 σ^{cv} 的正态分布。与前面的方法不同，观察的差的总方差用下式进行估计：

$$\hat{\sigma}^2_{d^{cv}} = \frac{\sum_{j=1}^{k}(d_j - \bar{d})^2}{k(k-1)} \tag{4-16}$$

其中，$\bar{d}$ 是平均差。对于这个方法，我们需要用 t 分布计算 d_t^{cv} 的置信区间：

$$d_t^{cv} = \bar{d} \pm t_{(1-\alpha),k-1}\hat{\sigma}_{d^{cv}}$$

系数 $t_{(1-\alpha),k-1}$ 可以通过两个参数（置信水平$(1-\alpha)$和自由度 $k-1$）查概率表得到。该 t 分布的概率表在表 4-6 中给出。

例 4.6　假设两个分类技术产生的模型的准确率估计差的均值等于 0.05，标准差等于 0.002。如果使用 30 折交叉验证方法估计准确率，则在 95%置信水平下，真实准确率差为：

$$d_t^{cv} = 0.05 \pm 1.70 \times 0.002 \tag{4-17}$$

因为置信区间不跨越 0 值，两个分类法的观察差是统计显著的。　□

表 4-6　t 分布的概率表

$k-1$	$(1-\alpha)$				
	0.90	0.95	0.975	0.99	0.995
1	3.08	6.31	12.7	31.8	63.7
2	1.89	2.92	4.30	6.96	9.92
4	1.53	2.13	2.78	3.75	4.60
9	1.38	1.83	2.26	2.82	3.25
14	1.34	1.76	2.14	2.62	2.98
19	1.33	1.73	2.09	2.54	2.86
24	1.32	1.71	2.06	2.49	2.80
29	1.31	1.70	2.04	2.46	2.76

文献注释

早期的分类系统是针对组织大量对象的集合。例如，杜威数字和国会图书馆的分类系统就是为大量的图书进行分类和索引设计的。分类是在领域专家的帮助下，用人工方式进行。

很多年来，自动分类一直是热门的研究课题。对于经典统计学分类的研究有时称作**判别分析**（discriminant analysis），其目标是根据预测子变量集来预测对象的组成员关系。著名的典型方法就是 Fisher 线性判别分析 [117]，它寻求产生不同类对象之间最大区分能力的数据的线性投影。

许多模式识别问题也需要区分不同类的对象。例子包括语音识别、手写字符识别和图像分类。对用于模式识别的分类技术的应用感兴趣的读者可以参阅 Jain 等[122]和 Kulkarni 等[128]的综述文章，或者参阅 Bishop[107]、Duda 等[114]和 Fukunaga[118] 的经典模式识别的书籍。分类也是神经网络、统计学习和机器学习领域的一个主要研究课题。有关各种分类技术的深入讨论请参阅 Cherkassky 和 Mulier[112]、Hastie 等[120]、Michie 等 [133]和 Mitchell[136]的书籍。

关于决策树归纳算法的全面评述可以在 Buntine[110]、Moret[137]、Murthy[138]和 Safavian 等[147]的综述文章中找到。一些著名的决策树算法包括 CART[108]、ID3[143]、C4.5[145]和 CHAID[125]。ID3 和 C4.5 都采用熵度量作为划分函数。对 C4.5 决策树算法的深入讨论请参阅 Quinlan[145]。除了解释决策树的生长和剪枝方法外，Quinlan[145]还介绍了怎样修改算法处理具有遗漏值的数据集。CART 算法是 Breiman 等[108]开发的，它使用 Gini 指标作为划分函数。CHAID[125]在决策树生长过程中使用χ^2统计检验确定最佳的划分点。

本章所介绍的决策树算法都假定划分条件一次只选择一个属性。斜决策树可以使用多个属性，在内部结点形成属性测试条件[121，152]。Breiman 等[108]提供了一个选项，可以在他们 CART 实现中使用属性的线性组合。归纳斜决策树的其他方法由 Heath 等[121]、Murthy 等[139]、Cantú-Paz 和 Kamath[111]、Utgoff 和 Brodley[152]提出。尽管斜决策树提高了决策树的表达能力，但在每个结点确定合适的测试条件在计算代价上仍然是一个挑战。另一种提高决策树表达能力，而不使用斜决策树的方法是**构造归纳**（constructive induction）[132]。该方法通过由原始属性创建复合特征，简化了学习复杂的划分函数的任务。

除自顶向下方法外，其他生长决策树的策略包括 Landeweerd 等[130]、Pattipati 和 Alexandridis[142]提出自底向上的方法，Kim 和 Landgrebe[126]提出的双向的方法。Schuermann 和 Doster[150]、Wang 和 Suen[154]提出了使用一种**软划分标准**（soft splitting criterion）来解决数据碎片问题。在这个方法中，每个记录以不同的概率指派到决策树的不同分支。

模型的过分拟合是一个必须解决的重要问题，确保决策树分类器在未知记录上也有同样好的性能。很多作者都讨论过模型的过分拟合问题，包括 Breiman 等[108]、Schaffer[148]、Mingers[135]、Jensen 和 Cohen[123]。尽管噪声的存在通常被认为是产生过分拟合的主要原因之一[135，140]，但是 Jensen 和 Cohen[123]却认为过分拟合是因为在多比较过程中使用了不正确的假设检验。

Schapire[149]定义泛化误差是“错误分类新样本的概率”，而检验误差是“在新抽取的检验集上出错的比例。”这样，泛化误差被认为是分类器的期望检验误差。泛化误差有时也可以认为是模型的真实误差[136]，即关于从训练集抽样的相同总体分布随机抽取数据点，它的期望误差。这些定义事实上是等价的，如果训练集和检验集来自相同的总体分布，这种情况在很多数据挖掘和机器学习应用中都经常遇到。

奥卡姆剃刀原理通常被认为是哲学家“奥卡姆的威廉”提出的。Domingos[113]告诫不能把奥卡姆剃刀误解为比较具有相似训练误差，而不是泛化误差的模型。关于决策树避免过分拟合的剪枝方法的综述由 Breslow 和 Aha[109]、Esposito 等[116]给出。其他典型的剪枝方法包括降低误差的剪枝[144]、悲观误差剪枝[144]、最小误差剪枝[141]、临界值剪枝[134]、代价复杂度剪枝[108]和基于误差的剪枝[145]。Quinlan 和 Rivest 提出使用最小描述长度原则对决策树剪枝[146]。

Kohavi[127]使用不同的评估方法，做了广泛的实验研究来比较性能度量，评估方法包括随机二次抽样、自助抽样和 k 折交叉验证，他们的结果表明最佳的评估方法是基于 10-折分层的交叉验证。Efron 和 Tibshirani[115]在理论和实验上比较了交叉验证和称作 632+规则的自助方法。

当前的技术（如 C4.5）要求整个训练数据集都能装入内存。为开发决策树归纳算法的并行和可伸缩的版本，已经做了大量工作。已提出的算法包括 Mehta 等[131]的 SLIQ、Shafer 等[151]的 SPRINT、Wang 和 Zaniolo[153]的 CMP、Alsabti 等[106]的 CLOUDS、Gehrke 等[119]的 RainForest 和 Joshi 等[124]的 ScalParC。关于数据挖掘的并行算法综述请参阅[129]。

参考文献

[106] K. Alsabti, S. Ranka, and V. Singh. CLOUDS: A Decision Tree Classifier for Large Datasets. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 2 - 8, New York, NY, August 1998.

[107] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, U.K., 1995.

[108] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

[109] L. A. Breslow and D. W. Aha. Simplifying Decision Trees: A Survey. *Knowledge Engineering Review*, 12(1):1 - 40, 1997.

[110] W. Buntine. Learning classification trees. In *Artificial Intelligence Frontiers in Statistics*, pages 182 - 201. Chapman & Hall, London, 1993.

[111] E. Cantú-Paz and C. Kamath. Using evolutionary algorithms to induce oblique decision trees. In *Proc. of the Genetic and Evolutionary Computation Conf.*, pages 1053 - 1060, San Francisco, CA, 2000.

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

[113] P. Domingos. The Role of Occam's Razor in Knowledge Discovery. *Data Mining and Knowledge Discovery*, 3(4):409 - 425, 1999.

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

[115] B. Efron and R. Tibshirani. Cross-validation and the Bootstrap: Estimating the Error Rate of a Prediction Rule. Technical report, Stanford University, 1995.

[116] F. Esposito, D. Malerba, and G. Semeraro. A Comparative Analysis of Methods for Pruning Decision Trees. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(5):476 - 491, May 1997.

[117] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179 - 188, 1936.

[118] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.

[119] J. Gehrke, R. Ramakrishnan, and V. Ganti. RainForest—A Framework for Fast Decision Tree Construction of Large Datasets. *Data Mining and Knowledge Discovery*, 4(2/3):127 - 162, 2000.

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

[121] D. Heath, S. Kasif, and S. Salzberg. Induction of Oblique Decision Trees. In *Proc. of the 13th Intl. Joint Conf. on Artificial Intelligence*, pages 1002 - 1007, Chambery, France, August 1993.

[122] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review.*IEEE Tran. Patt. Anal. and Mach. Intellig.*, 22(1):4 - 37, 2000.

[123] D. Jensen and P. R. Cohen. Multiple Comparisons in Induction Algorithms. *Machine Learning*, 38(3):309 - 338, March 2000.

[124] M. V. Joshi, G. Karypis, and V. Kumar. ScalParC: A New Scalable and Efficient Parallel Classification Algorithm for Mining Large Datasets. In *Proc. of 12th Intl.Parallel Processing Symp. (IPPS/SPDP)*, pages 573 - 579, Orlando, FL, April 1998.

[125] G. V. Kass. An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, 29:119 - 127, 1980.

[126] B. Kim and D. Landgrebe. Hierarchical decision classifiers in high-dimensional and large class data. *IEEE Trans. on Geoscience and Remote Sensing*, 29(4):518 - 528, 1991.

[127] R. Kohavi. A Study on Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proc. of the 15th Intl. Joint Conf. on Artificial Intelligence*, pages 1137 - 1145, Montreal, Canada, August 1995.

[128] S. R. Kulkarni, G. Lugosi, and S. S. Venkatesh. Learning Pattern Classification—A Survey. *IEEE Tran. Inf. Theory*, 44(6):2178 - 2206, 1998.

[129] V. Kumar, M. V. Joshi, E.-H. Han, P. N. Tan, and M. Steinbach. High Performance Data Mining. In *High Performance Computing for Computational Science (VECPAR 2002)*, pages 111 - 125. Springer, 2002.

[130] G. Landeweerd, T. Timmers, E. Gersema, M. Bins, and M. Halic. Binary tree versus single level tree classification of white blood cells. *Pattern Recognition*, 16:571 - 577, 1983.

[131] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A Fast Scalable Classifier for Data Mining. In *Proc. of the 5th Intl. Conf. on Extending Database Technology*, pages 18 - 32, Avignon, France, March 1996.

[132] R. S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111 - 116, 1983.

[133] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Upper Saddle River, NJ, 1994.

[134] J. Mingers. Expert Systems—Rule Induction with Statistical Data. *J Operational Research Society*, 38:39 - 47, 1987.

[135] J. Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4:227 - 243, 1989.

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

[137] B. M. E. Moret. Decision Trees and Diagrams. *Computing Surveys*, 14(4):593 - 623, 1982.

[138] S. K. Murthy. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, 2(4):345 - 389, 1998.

[139] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *J of Artificial Intelligence Research*, 2:1 - 33, 1994.

[140] T. Niblett. Constructing decision trees in noisy domains. In *Proc. of the 2nd European Working Session on Learning*, pages 67 - 78, Bled, Yugoslavia, May 1987.

[141] T. Niblett and I. Bratko. Learning Decision Rules in Noisy Domains. In *Research and Development in Expert Systems III*, Cambridge, 1986. Cambridge University Press.

[142] K. R. Pattipati andM. G. Alexandridis. Application of heuristic search and information theory to sequential fault diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(4):872 - 887, 1990.

[143] J. R. Quinlan. Discovering rules by induction from large collection of examples. In D. Michie, editor, *Expert Systems in the Micro Electronic Age*. Edinburgh University Press, Edinburgh, UK, 1979.

[144] J. R. Quinlan. Simplifying Decision Trees. *Intl. J. Man-Machine Studies*, 27:221 - 234, 1987.

[145] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan-Kaufmann Publishers, San Mateo, CA, 1993.

[146] J. R. Quinlan and R. L. Rivest. Inferring Decision Trees Using the Minimum Description Length Principle. *Information and Computation*, 80(3):227 - 248, 1989.

[147] S. R. Safavian and D. Landgrebe. A Survey of Decision Tree Classifier Methodology. *IEEE Trans. Systems, Man and Cybernetics*, 22:660 - 674, May/June 1998.

[148] C. Schaffer. Overfitting avoidence as bias. *Machine Learning*, 10:153 - 178, 1993.

[149] R. E. Schapire. The Boosting Approach to Machine Learning: An Overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[150] J. Schuermann and W. Doster. A decision-theoretic approach in hierarchical classifier design. *Pattern Recognition*, 17:359 - 369, 1984.

[151] J. C. Shafer, R. Agrawal, and M. Mehta. SPRINT: A Scalable Parallel Classifier for Data Mining. In

Proc. of the 22nd VLDB Conf., pages 544 - 555, Bombay, India, September 1996.

[152] P. E. Utgoff and C. E. Brodley. An incremental method for finding multivariate splits for decision trees. In *Proc. of the 7th Intl. Conf. on Machine Learning*, pages 58 - 65, Austin, TX, June 1990.

[153] H. Wang and C. Zaniolo. CMP: A Fast Decision Tree Classifier Using Multivariate Predictions. In *Proc. of the 16th Intl. Conf. on Data Engineering*, pages 449 - 460, San Diego, CA, March 2000.

[154] Q. R. Wang and C. Y. Suen. Large tree classifier with heuristic search and global training. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):91 - 102, 1987.

习　题

1. 为四个布尔属性 A，B，C 和 D 的奇偶函数画一棵完全决策树。可以简化该决策树吗？
2. 考虑表 4-7 中二元分类问题的训练样本。
   (a) 计算整个训练样本集的 Gini 指标值。
   (b) 计算属性顾客 ID 的 Gini 指标值。
   (c) 计算属性性别的 Gini 指标值。
   (d) 计算使用多路划分属性车型的 Gini 指标值。
   (e) 计算使用多路划分属性衬衣尺码的 Gini 指标值。
   (f) 下面哪个属性更好，性别、车型还是衬衣尺码？
   (g) 解释为什么属性顾客 ID 的 Gini 值最低，但却不能作为属性测试条件。

表 4-7　习题 2 的数据集

顾客 ID	性别	车型	衬衣尺码	类
1	男	家用	小	C0
2	男	运动	中	C0
3	男	运动	中	C0
4	男	运动	大	C0
5	男	运动	加大	C0
6	男	运动	加大	C0
7	女	运动	小	C0
8	女	运动	小	C0
9	女	运动	中	C0
10	女	豪华	大	C0
11	男	家用	大	C1
12	男	家用	加大	C1
13	男	家用	中	C1
14	男	豪华	加大	C1
15	女	豪华	小	C1
16	女	豪华	小	C1
17	女	豪华	中	C1
18	女	豪华	中	C1
19	女	豪华	中	C1
20	女	豪华	大	C1

3. 考虑表 4-8 中的二元分类问题的训练样本集。
   (a) 整个训练样本集关于类属性的熵是多少？
   (b) 关于这些训练样本，a_1 和 a_2 的信息增益是多少？
   (c) 对于连续属性 a_3，计算所有可能的划分的信息增益。
   (d) 根据信息增益，哪个是最佳划分（在 a_1, a_2 和 a_3 中）？

(e) 根据分类错误率，哪个是最佳划分（在 a_1 和 a_2 中）？

(f) 根据 Gini 指标，哪个是最佳划分（在 a_1 和 a_2 中）？

表 4-8　练习 3 的数据集

实例	a_1	a_2	a_3	目标类
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−

4. 证明：将结点划分为更小的后继结点之后，结点熵不会增加。

5. 考虑如下二元分类问题的数据集。

A	B	类标号
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

(a) 计算按照属性 *A* 和 *B* 划分时的信息增益。决策树归纳算法将会选择哪个属性？

(b) 计算按照属性 *A* 和 *B* 划分时 Gini 指标。决策树归纳算法将会选择哪个属性？

(c) 从图 4-13 可以看出熵和 Gini 指标在区间[0, 0.5]都是单调递增的，而在区间[0.5, 1]都是单调递减的。有没有可能信息增益和 Gini 指标增益支持不同的属性？解释你的理由。

6. 考虑如下训练样本集。

X	*Y*	*Z*	C1 类样本数	C2 类样本数
0	0	0	5	40
0	0	1	0	15
0	1	0	10	5
0	1	1	45	0
1	0	0	10	5
1	0	1	25	0
1	1	0	5	20
1	1	1	0	15

(a) 用本章所介绍的贪心法计算两层的决策树。使用分类错误率作为划分标准。决策树的总错误率是多少？

(b) 使用 X 作为第一个划分属性，两个后继结点分别在剩余的属性中选择最佳的划分属性，重复步骤(a)。所构造决策树的错误率是多少？

(c) 比较(a)和(b)的结果。评述在划分属性选择上启发式贪心法的作用。

7. 下表汇总了具有三个属性 A, B, C，以及两个类标号＋，－的数据集。建立一棵两层决策树。

A	B	C	实例数	
			+	−
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

(a) 根据分类错误率，哪个属性应当选作第一个划分属性？对每个属性，给出相依表和分类错误率的增益。

(b) 对根结点的两个子女重复以上问题。

(c) 最终的决策树错误分类的实例数是多少？

(d) 使用 C 作为划分属性，重复(a)、(b)和(c)。

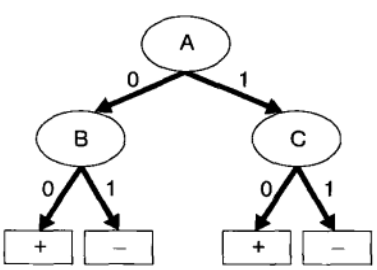
(e) 使用(c)和(d)中的结果分析决策树归纳算法贪心的本质。

8. 考虑图 4-30 中的决策树。

(a) 使用乐观方法计算决策树的泛化错误率。

(b) 使用悲观方法计算决策树的泛化错误率。（为了简单起见，使用在每个叶结点增加因子 0.5 的方法。）

(c) 使用提供的确认集计算决策树的泛化误差。这种方法叫作**降低误差剪枝**（reduced error pruning）。



训练集

实例	A	B	C	类
1	0	0	0	+
2	0	0	1	+
3	0	1	0	+
4	0	1	1	−
5	1	0	0	+
6	1	0	0	+
7	1	1	0	−
8	1	0	1	+
9	1	1	0	−
10	1	1	0	−

确认集

实例	A	B	C	类
11	0	0	0	+
12	0	1	1	+
13	1	1	0	+
14	1	0	1	−
15	1	0	0	+

图 4-30　习题 8 的决策树和数据集

9. 考虑图 4-31 中的决策树。假设产生决策树的数据集包含 16 个二元属性三个类 C_1、C_2和 C_3。

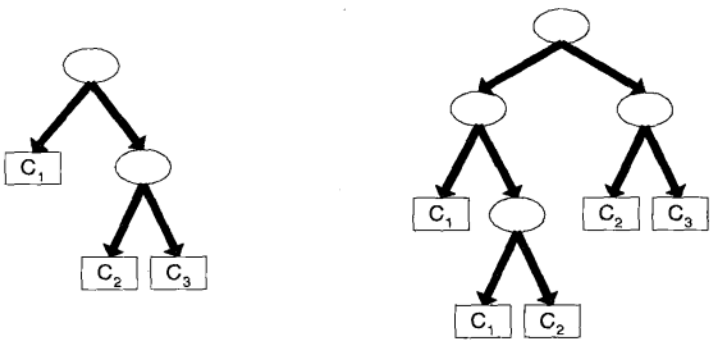
根据最小描述长度原则计算每棵决策树的总描述长度。

- 树的整体描述长度由下式给出：

$$Cost(tree, data) = Cost(tree) + Cost(data \mid tree)$$

- 树的每个内部结点用划分属性的 ID 进行编码。如果有 m 个属性，为每个属性编码的代价是 $\log_2 m$ 个二进位。
- 每个叶结点使用与之相关联的类的 ID 编码。如果有 k 个类，为每个类编码的代价是 $\log_2 k$ 个二进位。
- $Cost(tree)$是对决策树的所有结点编码的开销。为了简化计算，可以假设决策树的总开销是对每个内部结点和叶结点编码开销的总和。
- $Cost(data \mid tree)$是对决策树在训练集上分类错误编码的开销。每个错误用 $\log_2 n$ 个二进位编码，其中 n 是训练实例的总数。

根据 MDL 原则，哪棵决策树更好？



(a) 具有 7 个错误的决策树　　(b) 具有 4 个错误的决策树

图 4-31　习题 9 的决策树

10. 尽管.632 自助方法可以对模型的准确率做出可靠的估计，但是该方法也有明显的局限性[127]。考虑一个二类的问题，其中数据包含的正样本和负样本的数目相等，假设每个样本的类标号都是随机产生的，所使用的分类器是一棵未进行剪枝的决策树（即完全忠实的写照）。使用下面的方法确定分类器的准确率。

(a) 保持方法，使用三分之二的数据作为训练数据，剩余的三分之一作检验数据。

(b) 十折交叉验证。

(c) .632 自助方法。

(d) 从(a)、(b)、(c)的结果看，哪种方法对分类器的准确率提供了可靠的估计？

11. 考虑如下测试分类法 A 是否优于另一个分类法 B 的方法。设 N 是数据集的大小，p_A 是分类法 A 的准确率，p_B 是分类法 B 的准确率，而 $p = (p_A + p_B)/2$ 是两种分类法的平均准确率。为了测试分类法 A 是否显著优于 B，使用如下 Z 统计量：

$$Z = \frac{p_A - p_B}{\sqrt{\frac{2p(1-p)}{N}}}$$

如果 $Z > 1.96$，则认为分类法 A 优于分类法 B。表 4-9 在不同的数据集上比较了三个不同分类法的准确率：决策树分类法，朴素贝叶斯分类法和支持向量机。（后两种分类法在第 5 章中介绍。）

表 4-9　各种分类法准确率的比较

数据集	大小（N）	决策树（%）	朴素贝叶斯（%）	支持向量机（%）
Anneal	898	92.09	79.62	87.19
Australia	690	85.51	76.81	84.78
Auto	205	81.95	58.05	70.73
Breast	699	95.14	95.99	96.42
Cleve	303	76.24	83.50	84.49
Credit	690	85.80	77.54	85.07
Diabetes	768	72.40	75.91	76.82
German	1000	70.90	74.70	74.40
Glass	214	67.29	48.59	59.81
Heart	270	80.00	84.07	83.70
Hepatitis	155	81.94	83.23	87.10
Horse	368	85.33	78.80	82.61
Ionosphere	351	89.17	82.34	88.89
Iris	150	94.67	95.33	96.00
Labor	57	78.95	94.74	92.98
Led7	3200	73.34	73.16	73.56
Lymphography	148	77.03	83.11	86.49
Pima	768	74.35	76.04	76.95
Sonar	208	78.85	69.71	76.92
Tic-tac-toe	958	83.72	70.04	98.33
Vehicle	846	71.04	45.04	74.94
Wine	178	94.38	96.63	98.88
Zoo	101	93.07	93.07	96.04

用下面的 3×3 的表格汇总表 4-9 中给定的分类法在数据上的分类性能：

赢-输-平局	决策树	朴素贝叶斯	支持向量机
决策树	0 – 0 – 23		
朴素贝叶斯		0 – 0 – 23	
支持向量机			0 – 0 – 23

表格中每个单元的内容包含比较行与列的两个分类器时的赢、输和平局的数目。

12. 设 X 是一个均值为 Np、方差为 $Np(1-p)$的二元随机变量。证明比率 X/N 也服从均值为 p、方差为 $p(1-p)/N$ 的二项分布。

第5章

分类：其他技术

上一章介绍了一种简单但很有效的分类技术，称为决策树归纳。该章还详细地讨论了模型的过分拟合和分类器的评估问题。本章讲述构建分类模型的其他技术——从最简单的基于规则的分类器和最近邻分类器到更高级的支持向量机和组合方法。其他重要问题，如类失衡和多类问题等，也在本章后面部分进行讨论。

5.1 基于规则的分类器

基于规则的分类器是使用一组“if ... then ...”规则来对记录进行分类的技术。表 5-1 的例子中给出脊椎动物分类问题的基于规则的分类器产生的一个模型。该模型的规则用析取范式 $R = (r_1 \vee r_2 \vee \cdots \vee r_k)$表示，其中 R 称作**规则集**，而 r_i 是分类规则或析取项。

表 5-1 脊椎动物分类问题的规则集举例

r_1: (胎生 = 否) ∧ (飞行动物 = 是) → 鸟类
r_2: (胎生 = 否) ∧ (水生动物 = 是) → 鱼类
r_3: (胎生 = 是) ∧ (体温 = 恒温) → 哺乳类
r_4: (胎生 = 否) ∧ (飞行动物 = 否) → 爬行类
r_5: (水生动物 = 半) → 两栖类

每一个分类规则可以表示为如下形式：

$$r_i: (\text{条件}_i) \rightarrow y_i \tag{5-1}$$

规则左边称为**规则前件**（rule antecedent）或**前提**（precondition）。它是属性测试的合取：

$$\text{条件}_i = (A_1\ op\ v_1) \wedge (A_2\ op\ v_2) \wedge \cdots \wedge (A_k\ op\ v_k) \tag{5-2}$$

其中(A_j, v_j)是属性-值对，op 是比较运算符，取自集合$\{=, \neq, <, >, \leqslant, \geqslant\}$。每一个属性测试$(A_j\ op\ v_j)$称为一个合取项。规则右边称为**规则后件**（rule consequent），包含预测类 y_i。

如果规则 r 的前件和记录 x 的属性匹配，则称 r 覆盖 x。当 r 覆盖给定的记录时，称 r 被激发或被触发。作为例子，我们考虑表 5-1 中的规则 r_1 和两种脊椎动物——鹰和灰熊的以下属性：

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠
鹰	恒温	羽毛	否	否	是	是	否
灰熊	恒温	软毛	是	否	否	是	是

r_1 覆盖第一种脊椎动物，因为鹰的属性满足它的前件。r_1 不覆盖第二种脊椎动物，因为灰熊是胎生的且不能飞，故而违背了 r_1 的前件。

分类规则的质量可以用覆盖率（coverage）和准确率（accuracy）来度量。给定数据集 D 和

分类规则 $r: A \rightarrow y$，规则的覆盖率定义为 D 中触发规则 r 的记录所占的比例。另一方面，准确率或置信因子定义为触发 r 的记录中类标号等于 y 的记录所占的比例。这两个度量的形式化定义如下：

$$\text{Coverage}(r) = \frac{|A|}{|D|}$$

$$\text{Accuracy}(r) = \frac{|A \cap y|}{|A|} \tag{5-3}$$

其中 $|A|$ 是满足规则前件的记录数，$|A \cap y|$ 是同时满足规则前件和后件的记录数，D 是记录总数。

例 5.1　考虑表 5-2 中的数据集。规则

$$(\text{胎生} = \text{是}) \wedge (\text{体温} = \text{恒温}) \rightarrow \text{哺乳类}$$

的覆盖率是 33%，因为 15 个记录中有 5 个满足规则前件。该规则的准确率是 100%，因为规则覆盖的五个脊椎动物都是哺乳类。 □

表 5-2　脊椎动物数据集

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	否	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	否	是	是	否	鸟类
猫	恒温	软毛	是	否	否	是	否	哺乳类
虹鳉	冷血	鳞片	是	是	否	否	否	鱼类
美洲鳄	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
鳗鲡	冷血	鳞片	否	是	否	否	否	鱼类
蠎螈	冷血	无	否	半	否	是	是	两栖类

5.1.1　基于规则的分类器的工作原理

基于规则的分类器根据测试记录所触发的规则来对记录进行分类。为了说明一个基于规则的分类器是怎样工作的，考虑表 5-1 所示的规则集和下面的脊椎动物：

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠
狐猴	恒温	软毛	是	否	否	是	是
海龟	冷血	鳞片	否	半	否	是	否
角鲛鲨	冷血	鳞片	是	是	否	否	否

- 第一个脊椎动物——狐猴，是恒温动物，能生育幼仔。这触发规则 r_3，因此归为哺乳类。
- 第二个脊椎动物——海龟，同时触发规则 r_4 和 r_5。由于两个规则预测的类别相互冲突（爬行类和两栖类），它们的冲突类别必须得到解决。
- 没有规则可以用来分类角鲛鲨。在这种情况下，即使测试记录不被规则覆盖，我们需要

确保分类器仍能对记录做出可靠的预测。

上面的例子表明基于规则的分类器所产生的规则集的两个重要性质。

互斥规则（Mutually Exclusive Rule） 如果规则集 R 中不存在两条规则被同一条记录触发，则称规则集 R 中的规则是互斥的。这个性质确保每条记录至多被 R 中的一条规则覆盖。表 5-3 是一个互斥规则集的例子。

穷举规则（Exhaustive Rule） 如果对属性值的任一组合，R 中都存在一条规则加以覆盖，则称规则集 R 具有穷举覆盖。这个性质确保每一条记录都至少被 R 中的一条规则覆盖。假设体温和胎生是二元变量，则表 5-3 中的规则集具有穷举覆盖。

表 5-3 一个互斥和穷举的规则集的例子

r_1: (体温 = 冷血) → 非哺乳类
r_2: (体温 = 恒温) ∧ (胎生 = 是) → 哺乳类
r_3: (体温 = 恒温) ∧ (胎生 = 否) → 非哺乳类

这两个性质共同作用，保证每一条记录被且仅被一条规则覆盖。然而，很多基于规则的分类器，包括表 5-1 中所示的分类器，都不满足这两个性质。如果规则集不是穷举的，那么必须添加一个默认规则 r_d: () → y_d 来覆盖那些未被覆盖的记录。默认规则的前件为空，当所有其他规则失效时触发。y_d 是默认类，通常被指定为没有被现存规则覆盖的训练记录的多数类。

如果规则集不是互斥的，那么一条记录可能被多条规则覆盖，这些规则的预测可能会相互冲突。解决这个问题有如下两种方法。

有序规则（ordered rule） 在这种方法中，规则集中的规则按照优先级降序排列，优先级的定义有多种方法（如基于准确率、覆盖率、总描述长度或规则产生的顺序等）。有序的规则集也称为**决策表**（decision list）。当测试记录出现时，由覆盖记录的最高秩的规则对其进行分类，这就避免由多条分类规则来预测而产生的类冲突的问题。

无序规则（unordered rule） 这种方法允许一条测试记录触发多条分类规则，把每条被触发规则的后件看作是对相应类的一次投票，然后计票确定测试记录的类标号。通常把记录指派到得票最多的类。在某些情况下，投票可以用规则的准确率加权。使用无序规则来建立基于规则的分类器有利也有弊。首先，无序规则方法在分类一个测试记录时，不易受由于选择不当的规则而产生的错误的影响（而基于有序规则的分类器则对规则排序方法的选择非常敏感）。其次，建立模型的开销也相对较小，因为不必维护规则的顺序。然而，对测试记录进行分类却是一件很繁重的任务，因为测试记录的属性要与规则集中的每一条规则的前件作比较。

在本节的剩余部分，我们将重点讨论使用有序规则的基于规则的分类器。

5.1.2 规则的排序方案

对规则的排序可以逐条规则进行或者逐个类进行，图 5-1 给出两种方案的区别。

基于规则的排序方案 这个方案依据规则质量的某种度量对规则排序。这种排序方案确保每一个测试记录都是由覆盖它的“最好的”规则来分类。该方案的潜在缺点是规则的秩越低越难解释，因为每个规则都假设所有排在它前面的规则不成立。例如，图 5-1 左图基于规则的排序中第

四条规则

$$(\text{水生动物} = \text{半}) \rightarrow \text{两栖类}$$

有如下解释：如果一种脊椎动物没有羽毛或不能飞，并且是冷血的和半水生的，那么它属于两栖类。附加条件（脊椎动物没有羽毛或不能飞，并且是冷血的）是因为该脊椎动物不满足前三条规则。如果规则的数量很大，则解释处在列表尾部的规则将是一件非常麻烦的任务。

基于类的排序方案　在这种方案中，属于同一个类的规则在规则集 R 中一起出现。然后，这些规则根据它们所属的类信息一起排序。同一个类的规则之间的相对顺序并不重要，只要其中一个规则被激发，类标号就会赋给测试记录。这使得规则的解释稍微容易一些。然而，质量较差的规则可能碰巧预测较高秩的类，从而导致高质量的规则被忽略。

基于规则的排序
(表皮覆盖=羽毛，飞行动物=是) ==>鸟类
(体温=恒温，胎生=是) ==>哺乳类
(体温=恒温，胎生=否) ==>鸟类
(水生动物=半) ==>两栖类
(表皮覆盖=鳞片，水生动物=否) ==>爬行类
(表皮覆盖=鳞片，水生动物=是) ==>鱼类
(表皮覆盖=无) ==>两栖类

基于类的排序
(表皮覆盖=羽毛，飞行动物=是) ==>鸟类
(体温=恒温，胎生=否) ==>鸟类
(体温=恒温，胎生=是) ==>哺乳类
(水生动物=半) ==>两栖类
(表皮覆盖=无) ==>两栖类
(表皮覆盖=鳞片，水生动物=否) ==>爬行类
(表皮覆盖=鳞片，水生动物=是) ==>鱼类

图 5-1　基于规则的排序方案和基于类的排序方案的比较

由于大多数著名的基于规则的分类器（如 C4.5 规则和 RIPPER）都采用基于类的排序方案，本节剩余部分的讨论将主要集中在这种排序方案上。

5.1.3　如何建立基于规则的分类器

为了建立基于规则的分类器，需要提取一组规则来识别数据集的属性和类标号之间的关键联系。提取分类规则的方法有两大类：(1)直接方法，直接从数据中提取分类规则；(2)间接方法，从其他分类模型（如决策树和神经网络）中提取分类规则。

直接方法把属性空间分为较小的子空间，以便于属于一个子空间的所有记录可以使用一个分类规则进行分类。间接方法使用分类规则为较复杂的分类模型提供简洁的描述。5.1.4 节和 5.1.5 节分别对这两种方法进行详细讨论。

5.1.4　规则提取的直接方法

顺序覆盖（sequential covering）算法经常被用来直接从数据中提取规则，规则基于某种评估度量以贪心的方式增长。该算法从包含多个类的数据集中一次提取一个类的规则。对于脊椎动物分类问题，顺序覆盖算法可能先产生对鸟类进行分类的规则，然后依次是哺乳类、两栖类、爬行类，最后是鱼类的分类规则（见图 5-1）。决定哪一个类的规则最先产生的标准取决于多种因素，如类的普遍性（即训练记录中属于特定类的记录的比例），或者给定类中误分类记录的代价。

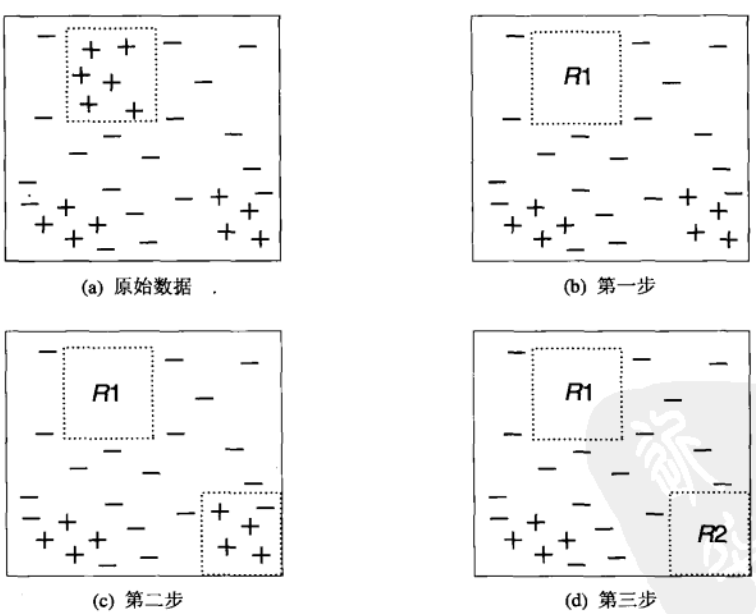
算法 5.1 给出顺序覆盖算法的描述。算法开始时决策表 R 为空。接下来用函数 Learn-One-Rule

提取类 y 的覆盖当前训练记录集的最佳规则。在提取规则时，类 y 的所有训练记录被看作是正例，而其他类的训练记录则被当成反例。如果一个规则覆盖大多数正例，没有或仅覆盖极少数反例，那么该规则是可取的。一旦找到这样的规则，就删掉它所覆盖的训练记录，并把新规则追加到决策表 R 的尾部。重复这个过程，直到满足终止条件。然后，算法继续产生下一个类的规则。

算法 5.1 顺序覆盖算法

```
1: 令 E 是训练记录，A 是属性-值对的集合{(A_j, v_j)}
2: 令 Y_o 是类的有序集{y_1, y_2, ···, y_k}
3: 令 R = { } 是初始规则列表
4: for 每个类 y∈Y_o − {y_k} do
5:     while 终止条件不满足 do
6:         r ← Learn-One-Rule (E, A, y)
7:         从 E 中删除被 r 覆盖的训练记录
8:         追加 r 到规则列表尾部：R←R ∨ r
9:     end while
10: end for
11: 把默认规则{}→y_k 插入到规则列表 R 尾部
```

图 5-2 演示在包含一组正例和反例的数据集上顺序覆盖算法是怎样工作的。规则 $R1$ 首先被提取出来，（覆盖如图 5-2b 所示）因为它覆盖的正例最多。接下来去掉 $R1$ 覆盖的所有训练记录，算法继续寻找下一个最好的规则，即 $R2$。



(a) 原始数据　(b) 第一步　(c) 第二步　(d) 第三步

图 5-2 顺序覆盖算法示例

1. Learn-One-Rule 函数

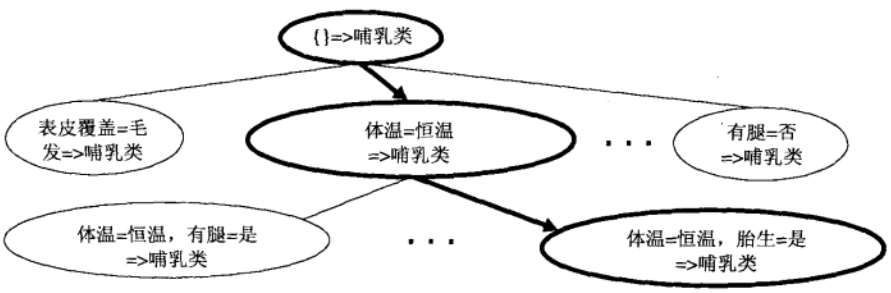
Learn-One-Rule 函数的目标是提取一个分类规则，该规则覆盖训练集中的大量正例，没有或

仅覆盖少量反例。然而，由于搜索空间呈指数大小，要找到一个最佳的规则的计算开销很大。Learn-One-Rule 函数通过以一种贪心的方式的增长规则来解决指数搜索问题。它先产生一个初始规则 r，并不断对该规则求精，直到满足某种终止条件为止。然后，修剪该规则，以改进它的泛化误差。

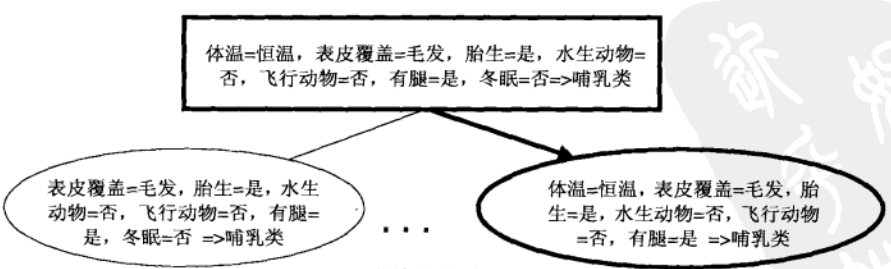
规则增长策略　常见的分类规则增长策略有两种：从一般到特殊和从特殊到一般。在从一般到特殊的策略中，先建立一个初始规则 r：{}→y，其中左边是一个空集，右边包含目标类。该规则的质量很差，因为它覆盖训练集中的所有样例。接着加入新的合取项来提高规则的质量。图 5-3a 显示脊椎动物分类问题的从一般到特殊的规则增长策略。合取项体温=恒温首先被选择作为规则的前件。算法接下来探查所有可能的候选，并贪心地选择下一个合取项胎生=是，将其添加到规则的前件中。继续该过程，直到满足终止条件为止（例如，加入的合取项已不能提高规则的质量）。

对于从特殊到一般的策略，可以随机地选择一个正例作为规则增长的初始种子。在求精步，通过删除规则的一个合取项，使其覆盖更多的正例来泛化规则。图 5-3b 给出了脊椎动物分类问题的从特殊到一般的方法。假设选择哺乳类的一个正例作为初始种子。初始规则与种子的属性值包含相同的合取项。为了提高覆盖率，删除合取项冬眠=否，以泛化规则。重复求精步，直到满足终止条件为止，例如，当规则开始覆盖反例时停止。

由于规则的贪心的方式增长，以上方法可能会产生次优规则。为了避免这种问题，可以采用束状搜索（beam search）。算法维护 k 个最佳候选规则，各候选规则各自在其前件中添加或删除合取项而独立地增长。评估候选规则的质量，选出 k 个最佳候选进入下一轮迭代。



(a) 从一般到特殊



(b) 从特殊到一般

图 5-3　从一般到特殊和从特殊到一般的规则增长策略

规则评估 在规则增长过程中，需要一种评估度量来确定应该添加（或删除）哪个合取项。准确率就是一个很明显的选择，因为它明确地给出了被规则正确分类的训练样例的比例。然而，把准确率作为标准的一个潜在的局限性是它没有考虑规则的覆盖率。例如，考虑一个训练集，它包含 60 个正例和 100 个反例。假设有如下两个候选规则。

规则 r_1：覆盖 50 个正例和 5 个反例，
规则 r_2：覆盖 2 个正例和 0 个反例。

r_1 和 r_2 的准确率分别为 90.9%和 100%。然而，r_1 是较好的规则，尽管其准确率较低。r_2 的高准确率具有潜在的欺骗性，因为它的覆盖率太低了。

下面的方法可以用来处理该问题。

(1) 可以使用统计检验剪除覆盖率较低的规则。例如，我们可以计算下面的似然比（likelihood ratio）统计量：

$$R = 2\sum_{i=1}^{k} f_i \log(f_i/e_i)$$

其中，k 是类的个数，f_i 是被规则覆盖的类 i 的样本的观测频率，e_i 是规则作随机猜测的期望频率。注意 R 是满足自由度为 $k-1$ 的 χ^2 分布。较大的 R 值说明该规则做出的正确预测数显著地大于随机猜测的结果。例如，由于 r_1 覆盖 55 个样例，则正类的期望频率为 $e_+ = 55\times60/160 = 20.625$，而负类的期望频率为 $e_- = 55\times100 / 160 = 34.375$。因此 r_1 的似然比为：

$$R(r_1) = 2\times [50\times\log_2(50/20.625)+5\times\log_2(5/34.375)] = 99.9$$

同理，r_2 的期望频率分别为 $e_+ = 2\times60/160 = 0.75$ 和 $e_- = 2\times100/160 = 1.25$。$r_2$ 的似然比统计量为：

$$R(r_2) = 2\times [2\times\log_2(2/0.75)+0\times\log_2(0/1.25)] = 5.66$$

因此，该统计量显示规则 r_1 比 r_2 好。

(2) 可以使用一种考虑规则覆盖率的评估度量。考虑如下评估度量：

$$\text{Laplace} = \frac{f_+ + 1}{n + k} \tag{5-4}$$

$$m \text{ 估计} = \frac{f_+ + kp_+}{n + k} \tag{5-5}$$

其中 n 是规则覆盖的样例数，f_+是规则覆盖的正例数，k 是类的总数，p_+是正类的先验概率。注意，当 $p_+ = 1/k$ 时，m 估计等价于 Laplace 度量。由于规则的覆盖率，这两个度量达到了准确率和正类先验概率之间的平衡。如果规则不覆盖任何训练样例，那么 Laplace 度量减小到 $1/k$，该值等于类符合均匀分布时正类的先验概率。当 $n = 0$ 时，m 估计也降到先验概率（p_+）。然而，当规则的覆盖率很高时，两个度量都渐近地趋向于规则的准确率 f_+/n。回到前面的例子，r_1 的 Laplace 度量为 51/57 = 89.47%，很接近它的准确率。相反，r_2 的 Laplace 度量（75%）比它的准确率小很多，这是因为 r_2 的覆盖率太小了。

(3) 另一种可以使用的评估度量是考虑规则的支持度计数的评估度量。**FOIL 信息增益**（FOIL's information gain）就是一种这样的度量。规则的支持度计数对应于它所覆盖的正例数。假设规则 $r: A \rightarrow +$覆盖 p_0 个正例和 n_0 个反例。增加新的合取项 B，扩展后的规则 $r': A \wedge B \rightarrow +$覆

盖 p_1 个正例和 n_1 个反例。根据以上信息，扩展后规则的 FOIL 信息增益定义为：

$$\text{FOIL 信息增益} = p_1 \times \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0} \right) \tag{5-6}$$

由于该度量与 p_1 和 $p_1 / (p_1 + n_1)$成正比，所以它更倾向于选择那些高支持度计数和高准确率的规则。上例中 r_1 和 r_2 的 FOIL 信息增益分别为 63.87 和 2.83，因此规则 r_1 比 r_2 好。

规则剪枝　可以对 Learn-One-Rule 函数产生的规则进行剪枝，以改善它们的泛化误差。为了确定是否需要进行剪枝，我们可以使用 4.4 节所介绍的方法来估计规则的泛化误差。例如，如果剪枝后，确认集上的误差减少了，那么就保持简化后的规则。另一种方法是比较剪枝前后规则的悲观误差（见 4.4.4 节）。如果剪枝后改进了悲观误差，就用简化后的规则替换原规则。

2. 顺序覆盖基本原理

规则提取出来后，顺序覆盖算法必须删除该规则所覆盖的所有正例和反例。下面的例子给出了这样做的理由。

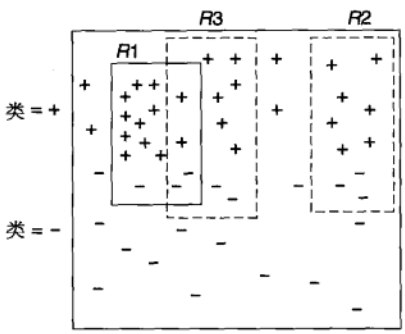


图 5-4　在顺序覆盖算法中删除训练记录。*R*1，*R*2 和 *R*3 分别代表三个不同规则所覆盖的区域

图 5-4 显示了从包含 29 个正例和 21 个反例的数据集中提取的三个可能的规则 *R*1、*R*2 和 *R*3。*R*1、*R*2 和 *R*3 的准确率分别是 12/15（80%）、7/10（70%）和 8/12（66.7%）。*R*1 先产生，因为它的准确率最高。*R*1 产生后，很明显需要删除它所覆盖的所有正例，以便算法产生的下一条规则不同于 *R*1。下一步，假设算法可以产生 *R*2 和 *R*3。尽管 *R*2 的准确率比 *R*3 高，但是 *R*1 和 *R*3 一起覆盖了 18 个正例和 5 个反例（总体准确率达到 78.3%），而 *R*1 和 *R*2 一起覆盖了 19 个正例和 6 个反例（总体准确率只有 76%）。如果在计算准确率之前就删除 *R*1 所覆盖的正例和反例，那么 *R*2 或 *R*3 对准确率的这种增量影响就会更明显。具体地说，如果不删除 *R*1 所覆盖的正例，那么我们就会高估 *R*3 的准确率；如果不删除 *R*1 所覆盖的反例，则会低估 *R*3 的准确率。在后一种情况下，我们最终可能会选择规则 *R*2，尽管 *R*3 所造成的虚假正例误差有一半已经被先前的规则 *R*1 所解决。

3. RIPPER 算法

为了阐明规则提取的直接方法，考虑一种广泛使用的规则归纳算法，叫作 RIPPER 算法。该算法的复杂度几乎线性地随训练样例的数目增长，并且特别适合为类分布不平衡的数据集建立模型。RIPPER 也能很好地处理噪声数据集，因为它使用一个确认数据集来防止模型过分拟合。

对两类问题，RIPPER 算法选择以多数类作为默认类，并为预测少数类学习规则。对于多类

问题，先按类的频率对类进行排序，设 $(y_1, y_2, \cdots, y_c)$ 是排序后的类，其中 y_1 是最不频繁的类，而 y_c 是最频繁的类。在第一次迭代中，把属于 y_1 的样例标记为正例，而把其他类的样例标记为反例，使用顺序覆盖算法产生区分正例和反例的规则。接下来，RIPPER 提取区分 y_2 和其他类的规则。重复该过程，直到剩下类 y_c，此时 y_c 作为默认类。

规则增长 RIPPER 算法使用从一般到特殊的策略进行规则增长，使用 FOIL 信息增益来选择最佳合取项添加到规则前件中。当规则开始覆盖反例时，停止添加合取项。新规则根据其在确认集上的性能进行剪枝。计算下面的度量来确定规则是否需要剪枝：$(p - n)/(p + n)$，其中 p 和 n 分别是被规则覆盖的确认集中的正例和反例数目，关于规则在确认集上的准确率，该度量是单调的。如果剪枝后该度量值增加，那么就去掉该合取项。剪枝是从最后添加的合取项开始的。例如，给定规则 $ABCD \rightarrow y$，RIPPER 算法先检查 D 是否应该剪枝，然后是 CD、BCD 等。尽管原来的规则仅覆盖正例，但是剪枝后的规则可能会覆盖训练集中的一些反例。

建立规则集 规则生成后，它所覆盖的所有正例和反例都要被删除。只要该规则不违反基于最小描述长度原则的终止条件，就将它添加到规则集中。如果新规则把规则集的总描述长度增加了至少 d 个比特位，那么 RIPPER 就停止把该规则加入到规则集（默认的 d 是 64 位）。RIPPER 使用的另一个终止条件是规则在确认集上的错误率不超过 50%。

RIPPER 算法也采用其他的优化步骤来决定规则集中现存的某些规则能否被更好的规则替代。对优化方法的细节感兴趣的读者可以查阅本章后面提到的参考文献。

5.1.5 规则提取的间接方法

本节介绍一种由决策树生成规则集的方法。原则上，决策树从根结点到叶结点的每一条路径都可以表示为一个分类规则。路径中的测试条件构成规则前件的合取项，叶结点的类标号赋给规则后件。图 5-5 显示了一个由决策树生成规则集的例子。注意，规则集是完全的，包含的规则是互斥的。但是，如下面的例子所示，其中某些规则可以加以简化。

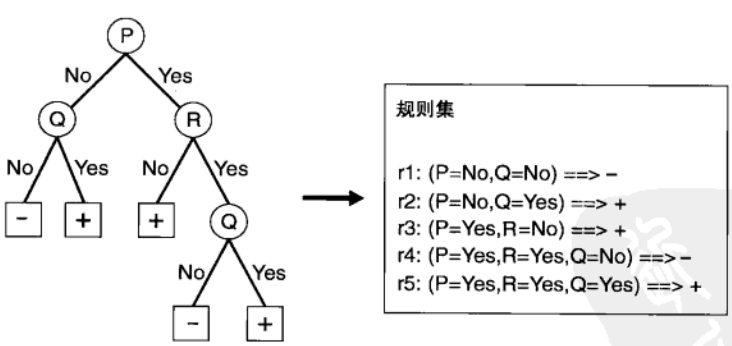


图 5-5 把决策树转化为分类规则

例 5.2 考虑图 5-5 中的以下三个规则：

$r2$: $(P = No) \wedge (Q = Yes) \rightarrow +$

$r3$: $(P = Yes) \wedge (R = No) \rightarrow +$

$r5$: $(P = Yes) \wedge (R = Yes) \wedge (Q = Yes) \rightarrow +$

观察到，当 Q 的值是 Yes 时，规则集总是预测正类。因此，可以把这些规则简化为：

$r2'$: (Q = Yes) → +

$r3$: (P = Yes) ∧ (R = No) → +

保留 $r3$ 来覆盖正类的剩余样例。尽管简化后的规则不再是互斥的，但它们比较简单并易于解释。□

下面，介绍 C4.5 规则算法所采用的从决策树生成规则集的方法。图 5-6 给出了表 5-2 中的数据集对应的决策树及生成的分类规则。

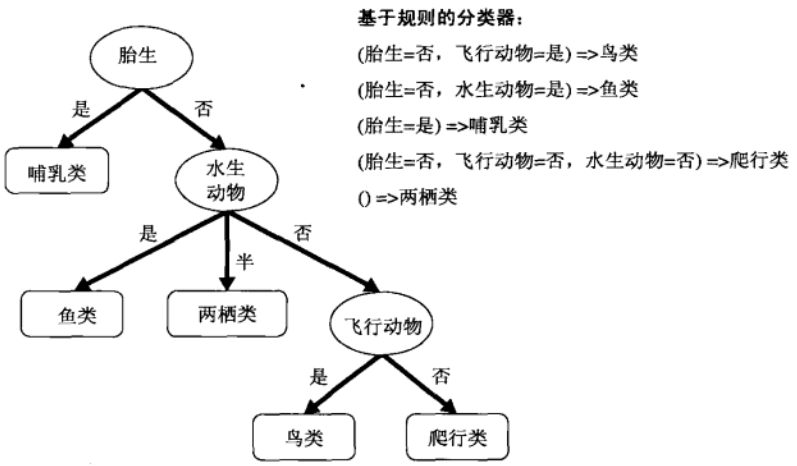


图 5-6　脊椎动物分类问题的决策树生成的分类规则

规则产生　决策树中从根结点到叶结点的每一条路径都产生一条分类规则。给定一个分类规则 r: $A \rightarrow y$，考虑简化后的规则 r': $A' \rightarrow y$，其中 A' 是从 A 去掉一个合取项后得到的。只要简化后的规则的误差率低于原规则的误差率，就保留其中悲观误差率最低的规则。重复规则剪枝步骤，直到规则的悲观误差不能再改进为止。由于某些规则在剪枝后会变得相同，因此必须丢弃重复规则。

规则排序　产生规则集后，C4.5 规则算法使用基于类的排序方案对提取的规则定序。预测同一个类的规则分到同一个子集中。计算每个子集的总描述长度，然后各类按照总描述长度由小到大排序。具有最小描述长度的类优先级最高，因为期望它包含最好的规则集。类的总描述长度等于 $L_{\text{exception}} + g \times L_{\text{model}}$，其中 $L_{\text{exception}}$ 是对误分类样例编码所需的比特位数，L_{model} 是对模型编码所需要的比特位数，而 g 是调节参数，默认值为 0.5。调节参数的值取决于模型中冗余属性的数量，如果模型含有很多冗余属性，那么调节参数的值会很小。

5.1.6　基于规则的分类器的特征

基于规则的分类器有如下特点。

- 规则集的表达能力几乎等价于决策树，因为决策树可以用互斥和穷举的规则集表示。基于规则的分类器和决策树分类器都对属性空间进行直线划分，并将类指派到每个划分。然而，如果基于规则的分类器允许一条记录触发多条规则的话，就可以构造一个更加复

杂的决策边界。

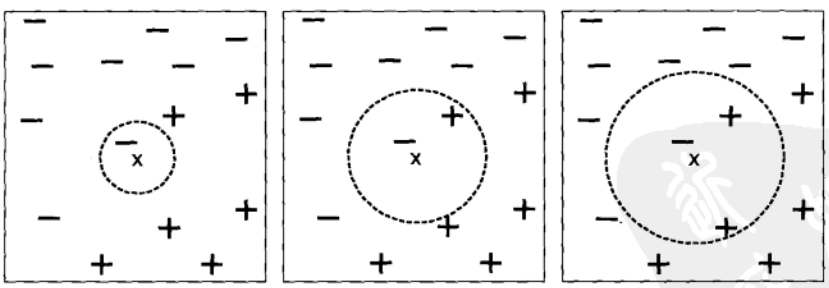
- 基于规则的分类器通常被用来产生更易于解释的描述性模型，而模型的性能却可与决策树分类器相媲美。
- 被很多基于规则的分类器（如 RIPPER）所采用的基于类的规则定序方法非常适于处理类分布不平衡的数据集。

5.2 最近邻分类器

图 4-3 中显示的分类框架包括两个步骤：(1)归纳步，由训练数据建立分类模型；(2)演绎步，把模型应用于测试样例。决策树和基于规则的分类器是**积极学习方法**（eager learner）的例子，因为如果训练数据可用，它们就开始学习从输入属性到类标号的映射模型。与之相反的策略是推迟对训练数据的建模，直到需要分类测试样例时再进行。采用这种策略的技术被称为**消极学习方法**（lazy learner）。消极学习的一个例子是 **Rote 分类器**（Rote classifier），它记住整个训练数据，仅当测试实例的属性和某个训练样例完全匹配时才进行分类。该方法一个明显的缺点是有些测试记录不能被分类，因为没有任何训练样例与它们相匹配。

使该方法更灵活的一个途径是找出和测试样例的属性相对接近的所有训练样例。这些训练样例称为**最近邻**（nearest neighbor），可以用来确定测试样例的类标号。使用最近邻确定类标号的合理性用下面的谚语最能说明："如果走像鸭子，叫像鸭子，看起来还像鸭子，那么它很可能就是一只鸭子。"最近邻分类器把每个样例看作 d 维空间上的一个数据点，其中 d 是属性个数。给定一个测试样例，我们使用 2.4 节中介绍的任意一种邻近性度量，计算该测试样例与训练集中其他数据点的邻近度。给定样例 z 的 k-最近邻是指和 z 距离最近的 k 个数据点。

图5-7给出了位于圆圈中心的数据点的1-最近邻、2-最近邻和3-最近邻。该数据点根据其近邻的类标号进行分类。如果数据点的近邻中含有多个类标号，则将该数据点指派到其最近邻的多数类。在图5-7a中，数据点的1-最近邻是一个负例，因此该点被指派到负类。如果最近邻是三个，如图5-7c所示，其中包括两个正例和一个负例，根据多数表决方案，该点被指派到正类。在最近邻中正例和负例个数相同的情况下（见图5-7b），可随机选择一个类标号来分类该点。



(a) 1-最近邻　(b) 2-最近邻　(c) 3-最近邻

图 5-7　一个实例的 1-最近邻、2-最近邻和 3-最近邻

前面的讨论中强调了选择合适的 k 值的重要性。如果 k 太小，则最近邻分类器容易受到由于训练数据中的噪声而产生的过分拟合的影响；相反，如果 k 太大，最近邻分类器可能会误分类测试样例，因为最近邻列表中可能包含远离其近邻的数据点（见图 5-8）。

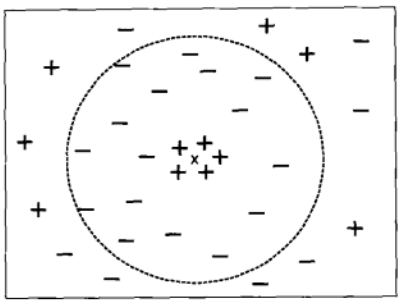


图 5-8　k 较大时的 k-最近邻分类

5.2.1　算法

算法 5.2 是对最近邻分类方法的一个高层描述。对每一个测试样例 $z=(\mathbf{x}', y')$，算法计算它和所有训练样例$(\mathbf{x}, y)\in D$ 之间的距离（或相似度），以确定其最近邻列表 D_z。如果训练样例的数目很大，那么这种计算的开销就会很大。然而，高效的索引技术可以降低为测试样例找最近邻时的计算量。

算法 5.2　k-最近邻分类算法

1：令 k 是最近邻数目，D 是训练样例的集合
2：**for** 每个测试样例 $z=(\mathbf{x}', y')$ **do**
3：　计算 z 和每个样例$(\mathbf{x}, y)\in D$ 之间的距离 $d(\mathbf{x}', \mathbf{x})$
4：　选择离 z 最近的 k 个训练样例的集合 $D_z\subseteq D$
5：　$y'=\underset{v}{\operatorname{argmax}}\sum_{(\mathbf{x}_i, y_i)\in D_z} I(v=y_i)$
6：**end for**

一旦得到最近邻列表，测试样例就会根据最近邻中的多数类进行分类：

$$\text{多数表决：}\quad y'=\underset{v}{\operatorname{argmax}}\sum_{(\mathbf{x}_i, y_i)\in D_z} I(v=y_i) \tag{5-7}$$

其中，v 是类标号，y_i 是一个最近邻的类标号，$I(.)$是指示函数，如果其参数为真，则返回 1，否则，返回 0。

在多数表决方法中，每个近邻对分类的影响都一样，这使得算法对 k 的选择很敏感，如图 5-7 所示。降低 k 的影响的一种途径就是根据每个最近邻 $\mathbf{x}_i$ 距离的不同对其作用加权：$w_i=1/d(\mathbf{x}', \mathbf{x}_i)^2$。结果使得远离 z 的训练样例对分类的影响要比那些靠近 z 的训练样例弱一些。使用距离加权表决方案，类标号可以由下面的公式确定：

$$\text{距离加权表决：}\quad y'=\underset{v}{\operatorname{argmax}}\sum_{(\mathbf{x}_i, y_i)\in D_z} w_i\times I(v=y_i) \tag{5-8}$$

5.2.2　最近邻分类器的特征

最近邻分类器的特点总结如下。

- 最近邻分类属于一类更广泛的技术，这种技术称为基于实例的学习，它使用具体的训练实例进行预测，而不必维护源自数据的抽象（或模型）。基于实例的学习算法需要邻近性

度量来确定实例间的相似性或距离，还需要分类函数根据测试实例与其他实例的邻近性返回测试实例的预测类标号。

- 像最近邻分类器这样的消极学习方法不需要建立模型，然而，分类测试样例的开销很大，因为需要逐个计算测试样例和训练样例之间的相似度。相反，积极学习方法通常花费大量计算资源来建立模型，模型一旦建立，分类测试样例就会非常快。
- 最近邻分类器基于局部信息进行预测，而决策树和基于规则的分类器则试图找到一个拟合整个输入空间的全局模型。正是因为这样的局部分类决策，最近邻分类器（k 很小时）对噪声非常敏感。
- 最近邻分类器可以生成任意形状的决策边界，这样的决策边界与决策树和基于规则的分类器通常所局限的直线决策边界相比，能提供更加灵活的模型表示。最近邻分类器的决策边界还有很高的可变性，因为它们依赖于训练样例的组合。增加最近邻的数目可以降低这种可变性。
- 除非采用适当的邻近性度量和数据预处理，否则最近邻分类器可能做出错误的预测。例如，我们想根据身高（以米为单位）和体重（以磅为单位）等属性来对一群人分类。属性高度的可变性很小，从 1.5 米到 1.85 米，而体重范围则可能是从 90 磅到 250 磅。如果不考虑属性值的单位，那么邻近性度量可能就会被人的体重差异所左右。

5.3 贝叶斯分类器

在很多应用中，属性集和类变量之间的关系是不确定的。换句话说，尽管测试记录的属性集和某些训练样例相同，但是也不能正确地预测它的类标号。这种情况产生的原因可能是噪声，或者出现了某些影响分类的因素却没有包含在分析中。例如，考虑根据一个人的饮食和锻炼的频率来预测他是否有患心脏病的危险。尽管大多数饮食健康、经常锻炼身体的人患心脏病的机率较小，但仍有人由于遗传、过量抽烟、酗酒等其他原因而患病。确定一个人的饮食是否健康、体育锻炼是否充分也是需要论证的课题，这反过来也会给学习问题带来不确定性。

本节将介绍一种对属性集和类变量的概率关系建模的方法。首先介绍**贝叶斯定理**（Bayes theorem），它是一种把类的先验知识和从数据中收集的新证据相结合的统计原理；然后解释贝叶斯定理在分类问题中的应用，接下来描述贝叶斯分类器的两种实现：朴素贝叶斯和贝叶斯信念网络。

5.3.1 贝叶斯定理

考虑两队之间的足球比赛：队 0 和队 1。假设 65%的比赛队 0 胜出，剩余的比赛队 1 获胜。队 0 获胜的比赛中只有 30%是在队 1 的主场，而队 1 取胜的比赛中 75%是主场获胜。如果下一场比赛在队 1 的主场进行，哪一支球队最有可能胜出呢？

这个问题可以由著名的贝叶斯定理来解答。为了完整起见，我们先讲一下概率论中的一些基本定义。

假设 X，Y 是一对随机变量，它们的联合概率 $P(X=x,Y=y)$是指 X 取值 x 且 Y 取值 y 的概率，条件概率是指一随机变量在另一随机变量取值已知的情况下取某一特定值的概率。例如，条件概率 $P(Y=y|X=x)$是指在变量 X 取值 x 的情况下，变量 Y 取值 y 的概率。X 和 Y 的联合概率和条件概率满足如下关系：

$$P(X,Y) = P(Y|X) \times P(X) = P(X|Y) \times P(Y) \tag{5-9}$$

调整公式（5-9）最后两个表达式得到下面公式，称为贝叶斯定理：

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{5-10}$$

贝叶斯定理可以用来解决本节开头的预测问题。为表述方便，用随机变量 X 代表东道主，随机变量 Y 代表比赛的胜利者。X 和 Y 可在集合{0,1}中取值。那么问题中给出的信息可总结如下：

队 0 取胜的概率是 $P(Y=0)=0.65$，

队 1 取胜的概率是 $P(Y=1)=1-P(Y=0)=0.35$，

队 1 取胜时作为东道主的概率是 $P(X=1|Y=1)=0.75$，

队 0 取胜时队 1 作为东道主的概率是 $P(X=1|Y=0)=0.3$。

我们的目的是计算 $P(Y=1|X=1)$，即队 1 在主场获胜的概率，并与 $P(Y=0|X=1)$比较。应用贝叶斯定理得到：

$$\begin{aligned}
P(Y=1|X=1) &= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1)} \\
&= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1,Y=1)+P(X=1,Y=0)} \\
&= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1|Y=1)P(Y=1)+P(X=1|Y=0)P(Y=0)} \\
&= \frac{0.75\times 0.35}{0.75\times 0.35+0.3\times 0.65} \\
&= 0.5738
\end{aligned}$$

其中，在第二行中应用了全概率公式。进一步，$P(Y=0|X=1)=1-P(Y=1|X=1)=0.4262$。因为 $P(Y=1|X=1)>P(Y=0|X=1)$，所以，队 1 更有机会赢得下一场比赛。

5.3.2　贝叶斯定理在分类中的应用

在描述贝叶斯定理怎样应用于分类之前，我们先从统计学的角度对分类问题加以形式化。设 $\mathbf{X}$ 表示属性集，Y 表示类变量。如果类变量和属性之间的关系不确定，那么我们可以把 $\mathbf{X}$ 和 Y 看作随机变量，用 $P(Y|\mathbf{X})$以概率的方式捕捉二者之间的关系。这个条件概率又称为 Y 的**后验概率**（posterior probability），与之相对地，$P(Y)$称为 Y 的**先验概率**（prior probability）。

在训练阶段，我们要根据从训练数据中收集的信息，对 $\mathbf{X}$ 和 Y 的每一种组合学习后验概率 $P(Y|\mathbf{X})$。知道这些概率后，通过找出使后验概率 $P(Y'|\mathbf{X}')$最大的类 Y'可以对测试记录 $\mathbf{X}'$进行分类。为了解释这种方法，考虑任务：预测一个贷款者是否会拖欠还款。图 5-9 中的训练集有如下属性：有房、婚姻状况和年收入。拖欠还款的贷款者属于类 Yes，还清贷款的贷款者属于类 No。

假设给定一测试记录有如下属性集：$\mathbf{X}$ =（有房 = 否，婚姻状况 = 已婚，年收入= $120K）。要分类该记录，我们需要利用训练数据中的可用信息计算后验概率 $P(\text{Yes}|\mathbf{X})$和 $P(\text{No}|\mathbf{X})$。如果 $P(\text{Yes}|\mathbf{X}) > P(\text{No}|\mathbf{X})$，那么记录分类为 Yes，反之，分类为 No。

	二元变量	分类变量	连续变量	类变量
Tid	有房	婚姻状况	年收入	拖欠贷款
1	是	单身	125K	否
2	否	已婚	100K	否
3	否	单身	70K	否
4	是	已婚	120K	否
5	否	离异	95K	是
6	否	已婚	60K	否
7	是	离异	220K	否
8	否	单身	85K	是
9	否	已婚	75K	否
10	否	单身	90K	是

图 5-9 预测贷款拖欠问题的训练集

准确估计类标号和属性值的每一种可能组合的后验概率非常困难，因为即便属性数目不是很大，仍然需要很大的训练集。此时，贝叶斯定理很有用，因为它允许我们用先验概率 $P(Y)$、**类条件**（class-conditional）概率 $P(\mathbf{X}|Y)$和证据 $P(\mathbf{X})$来表示后验概率：

$$P(Y|\mathbf{X}) = \frac{P(\mathbf{X}|Y)P(Y)}{P(\mathbf{X})} \tag{5-11}$$

在比较不同 Y 值的后验概率时，分母 $P(\mathbf{X})$总是常数，因此可以忽略。先验概率 $P(Y)$可以通过计算训练集中属于每个类的训练记录所占的比例很容易地估计。对类条件概率 $P(\mathbf{X}|Y)$的估计，我们介绍两种贝叶斯分类方法的实现：朴素贝叶斯分类器和贝叶斯信念网络。5.3.3 节和 5.3.5 节分别描述了这两种实现方法。

5.3.3 朴素贝叶斯分类器

给定类标号 y，朴素贝叶斯分类器在估计类条件概率时假设属性之间条件独立。条件独立假设可形式化地表述如下：

$$P(\mathbf{X}|Y=y) = \prod_{i=1}^{d} P(X_i|Y=y) \tag{5-12}$$

其中每个属性集 $\mathbf{X} = \{X_1, X_2, \cdots, X_d\}$包含 d 个属性。

1. 条件独立性

在深入研究朴素贝叶斯分类法如何工作的细节之前，让我们先介绍条件独立概念。设 $\mathbf{X}$，$\mathbf{Y}$ 和 $\mathbf{Z}$ 表示三个随机变量的集合。给定 $\mathbf{Z}$，$\mathbf{X}$ 条件独立于 $\mathbf{Y}$，如果下面的条件成立：

$$P(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) = P(\mathbf{X}|\mathbf{Z}) \tag{5-13}$$

条件独立的一个例子是一个人的手臂长短和他（她）的阅读能力之间的关系。你可能会发现手臂较长的人阅读能力也较强。这种关系可以用另一个因素解释，那就是年龄。小孩子的手臂往往比较短，也不具备成人的阅读能力。如果年龄一定，则观察到的手臂长度和阅读能力之间的关系就消失了。因此，我们可以得出结论，在年龄一定时，手臂长度和阅读能力二者条件独立。

$\mathbf{X}$ 和 $\mathbf{Y}$ 之间的条件独立也可以写成类似于公式（5-12）的形式：

$$
\begin{aligned}
P(\mathbf{X},\mathbf{Y}|\mathbf{Z}) &= \frac{P(\mathbf{X},\mathbf{Y},\mathbf{Z})}{P(\mathbf{Z})} \\
&= \frac{P(\mathbf{X},\mathbf{Y},\mathbf{Z})}{P(\mathbf{Y},\mathbf{Z})} \times \frac{P(\mathbf{Y},\mathbf{Z})}{P(\mathbf{Z})} \\
&= P(\mathbf{X}|\mathbf{Y},\mathbf{Z}) \times P(\mathbf{Y}|\mathbf{Z}) \\
&= P(\mathbf{X}|\mathbf{Z}) \times P(\mathbf{Y}|\mathbf{Z}) \qquad (5\text{-}14)
\end{aligned}
$$

其中，公式（5-13）用于得到公式（5-14）的最后一行。

2. 朴素贝叶斯分类器如何工作

有了条件独立假设，就不必计算 $\mathbf{X}$ 的每一个组合的类条件概率，只需对给定的 Y，计算每一个 X_i 的条件概率。后一种方法更实用，因为它不需要很大的训练集就能获得较好的概率估计。

分类测试记录时，朴素贝叶斯分类器对每个类 Y 计算后验概率：

$$
P(Y|\mathbf{X}) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(\mathbf{X})} \qquad (5\text{-}15)
$$

由于对所有的 Y，$P(\mathbf{X})$是固定的，因此只要找出使分子 $P(Y)\prod_{i=1}^{d} P(X_i|Y)$ 最大的类就足够了。在接下来的两部分，我们描述几种估计分类属性和连续属性的条件概率 $P(X_i|Y)$的方法。

3. 估计分类属性的条件概率

对分类属性 X_i，根据类 y 中属性值等于 x_i 的训练实例的比例来估计条件概率 $P(X_i = x_i|Y = y)$。例如，在图 5-9 给出的训练集中，还清贷款的 7 个人中 3 个人有房，因此，条件概率 $P(\text{有房}=\text{是}|\text{No})$等于 3/7。同理，拖欠还款的人中单身的条件概率 $P(\text{婚姻状况}=\text{单身}|\text{Yes}) = 2/3$。

4. 估计连续属性的条件概率

朴素贝叶斯分类法使用两种方法估计连续属性的类条件概率。

(1) 可以把每一个连续的属性离散化，然后用相应的离散区间替换连续属性值。这种方法把连续属性转换成序数属性。通过计算类 y 的训练记录中落入 X_i 对应区间的比例来估计条件概率 $P(X_i|Y = y)$。估计误差由离散策略（见 2.3.6 节）和离散区间的数目决定。如果离散区间的数目太大，则就会因为每一个区间中训练记录太少而不能对 $P(X_i|Y)$做出可靠的估计。相反，如果区间数目太小，有些区间就会含有来自不同类的记录，因此失去了正确的决策边界。

(2) 可以假设连续变量服从某种概率分布，然后使用训练数据估计分布的参数。高斯分布通常被用来表示连续属性的类条件概率分布。该分布有两个参数，均值μ和方差σ^2。对每个类 y_j，属性 X_i 的类条件概率等于：

$$
P(X_i = x_i|Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}} \qquad (5\text{-}16)
$$

参数μ_{ij}可以用类 y_j 的所有训练记录关于 X_i 的样本均值（$\bar{x}$）来估计。同理，参数σ_{ij}^2可以用这些训练记录的样本方差（s^2）来估计。例如，考虑图 5-9 中年收入这一属性。该属性关于类 No 的样本均值和方差如下：

$$\bar{x} = \frac{125+100+70+\cdots+75}{7} = 100$$

$$s^2 = \frac{(125-110)^2+(100-110)^2+\cdots+(75-110)^2}{7(6)} = 2975$$

$$s = \sqrt{2975} = 54.54$$

给定一测试记录，应征税的收入等于 120K 美元，其类条件概率计算如下：

$$P(\text{收入}=\$120\text{K}|\text{No}) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2\times 2975}} = 0.0072$$

注意，前面对类条件概率的解释有一定的误导性。公式（5-16）的右边对应于一个**概率密度函数**（probability density function）$f(X_i; \mu_{ij}, \sigma_{ij})$。因为该函数是连续的，所以随机变量 X_i 取某一特定值的概率为 0。取而代之，我们应该计算 X_i 落在区间 x_i 到 $x_i+\varepsilon$ 的条件概率，其中 ε 是一个很小的常数：

$$P(x_i \leqslant X_i \leqslant x_i+\varepsilon \,|\, Y=y_j) = \int_{x_i}^{x_i+\varepsilon} f(X_i;\mu_{ij},\sigma_{ij})dX_i \approx f(x_i;\mu_{ij},\sigma_{ij}) \times \varepsilon \tag{5-17}$$

由于 ε 是每个类的一个常量乘法因子，在对后验概率 $P(Y|\mathbf{X})$ 进行规范化的时候就抵消掉了。因此，我们仍可以使用公式（5-16）来估计类条件概率 $P(X_i|Y)$。

5. 朴素贝叶斯分类器举例

考虑图5-10a中的数据集。我们可以计算每个分类属性的类条件概率，同时利用前面介绍的方法计算连续属性的样本均值和方差。这些概率汇总在图5-10b中。

Tid	有房	婚姻状况	年收入	拖欠贷款
1	是	单身	125K	否
2	否	已婚	100K	否
3	否	单身	70K	否
4	是	已婚	120K	否
5	否	离婚	95K	是
6	否	已婚	60K	否
7	是	离婚	220K	否
8	否	单身	85K	是
9	否	已婚	75K	否
10	否	单身	90K	是

(a)

P(有房=是|No) =3/7
P(有房=否|No) =4/7
P(有房=是|Yes) =0
P(有房=否|Yes) =1
P(婚姻状况=单身|No) =2/7
P(婚姻状况=离婚|No) =1/7
P(婚姻状况=已婚|No) =4/7
P(婚姻状况=单身|Yes) =2/3
P(婚姻状况=离婚|Yes) =1/3
P(婚姻状况=已婚|Yes) =0

年收入：
如果类=No： 样本均值=110
样本方差=2975
如果类=Yes： 样本均值=90
样本方差=25

(b)

图 5-10 贷款分类问题的朴素贝叶斯分类器

为了预测测试记录 **X**=（有房=否，婚姻状况=已婚，年收入=$120K）的类标号，需要计算后验概率 $P(\text{No}|\mathbf{X})$ 和 $P(\text{Yes}|\mathbf{X})$。回想一下我们前面的讨论，这些后验概率可以通过计算先验概率 $P(Y)$ 和类条件概率 $\prod_i P(X_i|Y)$ 的乘积来估计，对应于公式（5-15）右端的分子。

每个类的先验概率可以通过计算属于该类的训练记录所占的比例来估计。因为有 3 个记录属于类 Yes，7 个记录属于类 No，所以 $P(\text{Yes}) = 0.3$，$P(\text{No}) = 0.7$。使用图 5-10b 中提供的信息，类条件概率计算如下：

$$\begin{aligned} P(\mathbf{X}|\text{No}) &= P(\text{有房}=\text{否}|\text{No}) \times P(\text{婚姻状况}=\text{已婚}|\text{No}) \times P(\text{年收入}=\$120\text{K}|\text{No}) \\ &= 4/7 \times 4/7 \times 0.0072 = 0.0024 \\ P(\mathbf{X}|\text{Yes}) &= P(\text{有房}=\text{否}|\text{Yes}) \times P(\text{婚姻状况}=\text{已婚}|\text{Yes}) \times P(\text{年收入}=\$120\text{K}|\text{Yes}) \\ &= 1 \times 0 \times 1.2 \times 10^{-9} = 0 \end{aligned}$$

放到一起可得到类 No 的后验概率 $P(\text{No}|\mathbf{X}) = \alpha \times 7/10 \times 0.0024 = 0.0016\alpha$，其中 $\alpha = 1/P(\mathbf{X})$是个常量。同理，可以得到类 Yes 的后验概率等于 0，因为它的类条件概率等于 0。因为 $P(\text{No}|\mathbf{X}) > P(\text{Yes}|\mathbf{X})$，所以记录分类为 No。

6. 条件概率的 m 估计

前面的例子体现了从训练数据估计后验概率时的一个潜在问题：如果有一个属性的类条件概率等于 0，则整个类的后验概率就等于 0。仅使用记录比例来估计类条件概率的方法显得太脆弱了，尤其是当训练样例很少而属性数目又很大时。

一种更极端的情况是，当训练样例不能覆盖那么多的属性值时，我们可能就无法分类某些测试记录。例如，如果 $P(\text{婚姻状况}=\text{离婚}|\text{No})$为 0 而不是 1/7，那么具有属性集 $\mathbf{X}=$（有房=是，婚姻状况=离婚，年收入=\$120K）的记录的类条件概率如下：

$$\begin{aligned} P(\mathbf{X}|\text{No}) &= 3/7 \times 0 \times 0.0072 = 0 \\ P(\mathbf{X}|\text{Yes}) &= 0 \times 1/3 \times 1.2 \times 10^{-9} = 0 \end{aligned}$$

朴素贝叶斯分类器无法分类该记录。解决该问题的途径是使用 m 估计（m-estimate）方法来估计条件概率：

$$P(x_i|y_j) = \frac{n_c + mp}{n + m} \tag{5-18}$$

其中，n 是类 y_j 中的实例总数，n_c 是类 y_j 的训练样例中取值 x_i 的样例数，m 是称为等价样本大小的参数，而 p 是用户指定的参数。如果没有训练集（即 $n = 0$），则 $P(x_i|y_j) = p$。因此 p 可以看作是在类 y_j 的记录中观察属性值 x_i 的先验概率。等价样本大小决定先验概率 p 和观测概率 n_c/n 之间的平衡。

在前面的例子中，条件概率 $P(\text{婚姻状况}=\text{已婚}|\text{Yes}) = 0$，因为类中没有训练样例含有该属性值。使用 m 估计方法，$m = 3$，$p = 1/3$，则条件概率不再是 0：

$$P(\text{婚姻状况}=\text{已婚}|\text{Yes}) = (0 + 3 \times 1/3)/(3 + 3) = 1/6$$

如果假设对类 Yes 的所有属性 $p = 1/3$，对类 No 的所有属性 $p = 2/3$，则

$$\begin{aligned} P(\mathbf{X}|\text{No}) &= P(\text{有房}=\text{否}|\text{No}) \times P(\text{婚姻状况}=\text{已婚}|\text{No}) \times P(\text{年收入}=\$120\text{K}|\text{No}) \\ &= 6/10 \times 6/10 \times 0.0072 = 0.0026 \\ P(\mathbf{X}|\text{Yes}) &= P(\text{有房}=\text{否}|\text{Yes}) \times P(\text{婚姻状况}=\text{已婚}|\text{Yes}) \times P(\text{年收入}=\$120\text{K}|\text{Yes}) \\ &= 4/6 \times 1/6 \times 1.2 \times 10^{-9} = 1.3 \times 10^{-10} \end{aligned}$$

类 No 的后验概率 $P(\mathbf{X}|\text{No}) = \alpha \times 7/10 \times 0.0026 = 0.0018\alpha$，而类 Yes 的后验概率 $P(\mathbf{X}|\text{Yes}) = \alpha \times 3/10 \times 1.3 \times 10^{-10} = 4.0 \times 10^{-11}\alpha$。尽管分类结果不变，但是当训练样例较少时，$m$ 估计通常是一种更加健壮的概率估计方法。

7. 朴素贝叶斯分类器的特征

朴素贝叶斯分类器一般具有以下特点。

- 面对孤立的噪声点，朴素贝叶斯分类器是健壮的。因为在从数据中估计条件概率时，这些点被平均。通过在建模和分类时忽略样例，朴素贝叶斯分类器也可以处理属性值遗漏问题。
- 面对无关属性，该分类器是健壮的。如果 X_i 是无关属性，那么 $P(X_i|Y)$ 几乎变成了均匀分布。X_i 的类条件概率不会对总的后验概率的计算产生影响。
- 相关属性可能会降低朴素贝叶斯分类器的性能，因为对这些属性，条件独立的假设已不成立。例如，考虑下面的概率：

$$P(A=0|Y=0)=0.4,\quad P(A=1|Y=0)=0.6$$
$$P(A=0|Y=1)=0.6,\quad P(A=1|Y=1)=0.4$$

其中，A 是二元属性，Y 是二元类变量。假设存在另一个二值属性 B，当 $Y=0$ 时，B 与 A 完全相关；当 $Y=1$ 时，B 与 A 相互独立。简单地说，假设 B 的类条件概率与 A 相同。给定一个记录，含有属性 $A=0$，$B=0$，其后验概率计算如下：

$$P(Y=0|A=0,B=0)=\frac{P(A=0|Y=0)P(B=0|Y=0)P(Y=0)}{P(A=0,B=0)}$$
$$=\frac{0.16\times P(Y=0)}{P(A=0,B=0)}$$

$$P(Y=1|A=0,B=0)=\frac{P(A=0|Y=1)P(B=0|Y=1)P(Y=1)}{P(A=0,B=0)}$$
$$=\frac{0.36\times P(Y=1)}{P(A=0,B=0)}$$

如果 $P(Y=0)=P(Y=1)$，则朴素贝叶斯分类器将把该记录指派到类 1。然而，事实上

$$P(A=0,B=0|Y=0)=P(A=0|Y=0)=0.4$$

因为当 $Y=0$ 时，A 和 B 完全相关。结果，$Y=0$ 的后验概率是：

$$P(Y=0|A=0,B=0)=\frac{P(A=0,B=0|Y=0)P(Y=0)}{P(A=0,B=0)}$$
$$=\frac{0.4\times P(Y=0)}{P(A=0,B=0)}$$

比 $Y=1$ 的后验概率大，因此，该记录实际应该分类为类 0。

5.3.4 贝叶斯误差率

假设我们知道支配 $P(\mathbf{X}|Y)$ 的真实概率分布。使用贝叶斯分类方法，我们就能确定分类任务的

理想决策边界，如下例所示。

例 5.3　考虑任务：根据体长区分美洲鳄和鳄鱼。一条成年鳄鱼的平均体长大约 15 英尺，而一条成年美洲鳄的体长大约 12 英尺。假设它们的体长 x 服从标准差为 2 英尺的高斯分布，那么二者的类条件概率表示如下：

$$P(X|\text{鳄鱼}) = \frac{1}{\sqrt{2\pi}\cdot 2}\exp\left[-\frac{1}{2}\left(\frac{X-15}{2}\right)^2\right] \tag{5-19}$$

$$P(X|\text{美洲鳄}) = \frac{1}{\sqrt{2\pi}\cdot 2}\exp\left[-\frac{1}{2}\left(\frac{X-12}{2}\right)^2\right] \tag{5-20}$$

图 5-11 给出了鳄鱼和美洲鳄类条件概率的比较。假设它们的先验概率相同，理想决策边界 $\hat{x}$ 满足：

$$P(X=\hat{x}|\text{鳄鱼}) = P(X=\hat{x}|\text{美洲鳄})$$

利用公式（5-19）和公式（5-20），得到：

$$\left(\frac{\hat{x}-15}{2}\right)^2 = \left(\frac{\hat{x}-12}{2}\right)^2$$

解得 $\hat{x}$=13.5。该例的决策边界处在两个均值的中点。 □

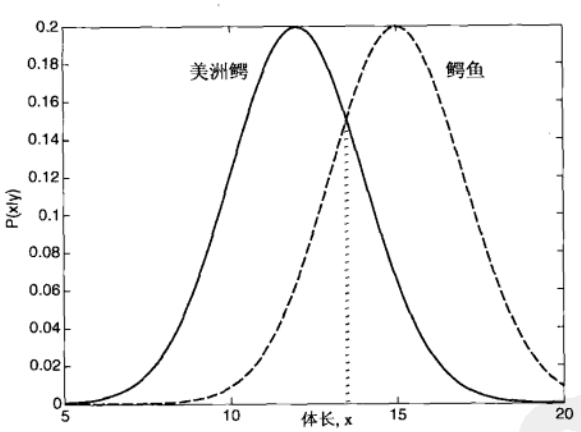


图 5-11　鳄鱼和美洲鳄似然函数比较

当先验概率不同时，决策边界朝着先验概率较小的类移动（见习题 10）。此外，给定数据上的任何分类器所达到的最小误差率都是可计算的。上例中的理想决策边界把体长小于 $\hat{x}$ 的分类为美洲鳄，把体长大于 $\hat{x}$ 的分类为鳄鱼。该分类器的误差率等于鳄鱼的后验概率曲线下面的区域（从 0 到 $\hat{x}$）加上美洲鳄后验概率曲线下面的区域（从 $\hat{x}$ 到 ∞）：

$$\text{Error} = \int_0^{\hat{x}} P(\text{鳄鱼}\,|\,X)\mathrm{d}X + \int_{\hat{x}}^{\infty} P(\text{美洲鳄}\,|\,X)\mathrm{d}X$$

总误差率称为**贝叶斯误差率**（Bayes error rate）。

5.3.5 贝叶斯信念网络

朴素贝叶斯分类器的条件独立假设似乎太严格了，特别是对那些属性之间有一定相关性的分类问题。本节介绍一种更灵活的类条件概率 $P(\mathbf{X}|Y)$的建模方法。该方法不要求给定类的所有属性都条件独立，而是允许指定哪些属性条件独立。我们先讨论怎样表示和建立该概率模型，接着举例说明怎样使用模型进行推理。

1. 模型表示

贝叶斯信念网络（Bayesian belief networks，BBN），简称贝叶斯网络，用图形表示一组随机变量之间的概率关系。贝叶斯网络有两个主要成分。

(1) 一个有向无环图（dag），表示变量之间的依赖关系。

(2) 一个概率表，把各结点和它的直接父结点关联起来。

考虑三个随机变量 A、B 和 C，其中 A 和 B 相互独立，并且都直接影响第三个变量 C。三个变量之间的关系可以用图 5-12a 中的有向无环图概括。图中每个结点表示一个变量，每条弧表示两个变量之间的依赖关系。如果从 X 到 Y 有一条有向弧，则 X 是 Y 的**父母**，Y 是 X 的**子女**。另外，如果网络中存在一条从 X 到 Z 的有向路经，则 X 是 Z 的**祖先**，而 Z 是 X 的**后代**。例如，在图 5-12b 中，A 是 D 的后代，D 是 B 的祖先，而且 B 和 D 都不是 A 的后代结点。贝叶斯网络的一个重要性质表述如下：

性质 1　条件独立　贝叶斯网络中的一个结点，如果它的父母结点已知，则它条件独立于它的所有非后代结点。

图 5-12b 中，给定 C，A 条件独立于 B 和 D，因为 B 和 D 都是 A 的非后代结点。朴素贝叶斯分类器中的条件独立假设也可以用贝叶斯网络来表示，如图 5-12c 所示，其中 y 是目标类，$\{X_1, X_2, \cdots, X_d\}$是属性集。

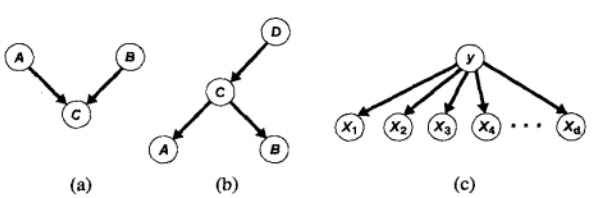


图 5-12　使用有向无环图表示概率关系

除了网络拓扑结构要求的条件独立性外，每个结点还关联一个概率表。

(1) 如果结点 X 没有父母结点，则表中只包含先验概率 $P(X)$。

(2) 如果结点 X 只有一个父母结点 Y，则表中包含条件概率 $P(X|Y)$。

(3) 如果结点 X 有多个父母结点$\{Y_1, Y_2, \cdots, Y_k\}$，则表中包含条件概率 $P(X|Y_1, Y_2, \cdots, Y_k)$。

图 5-13 是贝叶斯网络的一个例子，对心脏病或心口痛患者建模。假设图中每个变量都是二值的。心脏病结点（HD）的父母结点对应于影响该疾病的危险因素，例如锻炼(E) 和饮食(D)等。心脏病结点的子结点对应于该病的症状，如胸痛（CP）和高血压（BP）等。如图所示，心口痛（Hb）可能源于不健康的饮食，同时又可能导致胸痛。

影响疾病的危险因素对应的结点只包含先验概率，而心脏病、心口痛以及它们的相应症状所对应的结点都包含条件概率。为了节省空间，图中省略了一些概率。注意 $P(X=\bar{x}) = 1 - P(X=x)$ ，

$P(X=\bar{x}\,|Y)=1-P(X=x|Y)$，其中 $\bar{x}$ 表示和 x 相反的结果。因此，省略的概率可以很容易求得。例如，条件概率：

$$\begin{aligned}&P(\text{心脏病}=\text{No}|\text{锻炼}=\text{No},\ \text{饮食}=\text{健康})\\&=1-P(\text{心脏病}=\text{Yes}|\text{锻炼}=\text{No},\ \text{饮食}=\text{健康})\\&=1-0.55=0.45\end{aligned}$$

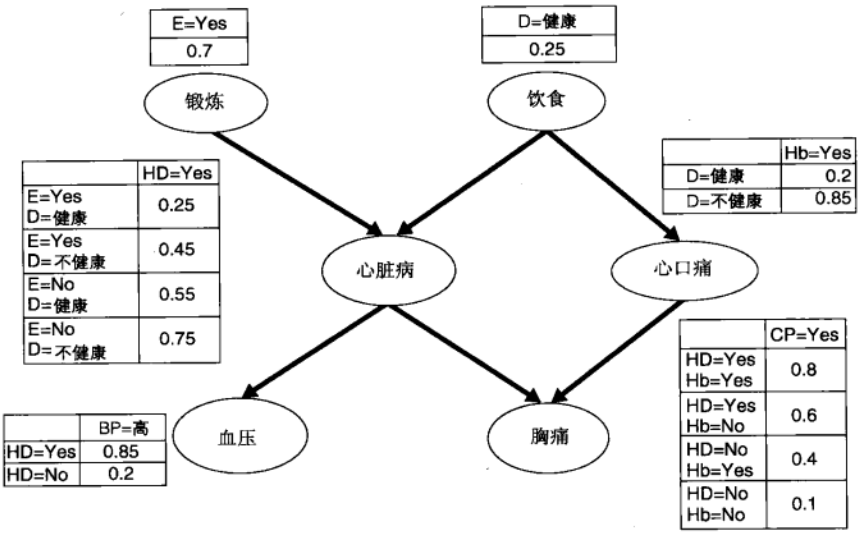


图 5-13　发现心脏病和心口痛病人的贝叶斯网络

2. 建立模型

贝叶斯网络的建模包括两个步骤：(1)创建网络结构；(2)估计每一个结点的概率表中的概率值。网络拓扑结构可以通过对主观的领域专家知识编码获得。算法 5.3 给出了归纳贝叶斯网络拓扑结构的一个系统的过程。

算法 5.3　贝叶斯网络拓扑结构的生成算法

1：设 $T=(X_1, X_2, \cdots, X_d)$表示变量的全序
2：**for** j=1 to d **do**
3：　令 $X_{T(j)}$表示 T 中第 j 个次序最高的变量
4：　令$\pi(X_{T(j)})=\{X_{T(1)}, X_{T(2)}, \cdots, X_{T(j-1)}\}$表示排在 $X_{T(j)}$前面的变量的集合
5：　从$\pi(X_{T(j)})$中去掉对 X_j 没有影响的变量（使用先验知识）
6：　在 $X_{T(j)}$和$\pi(X_{T(j)})$中剩余的变量之间画弧
7：**end for**

例 5.4　考虑图 5-13 中的变量。执行步骤 1 后，设变量次序为(E, D, HD, Hb, CP, BP)。从变量 D 开始，经过步骤 2 到步骤 7，我们得到如下条件概率。

- $P(D|E)$化简为 $P(D)$。
- $P(HD|E, D)$不能化简。
- $P(Hb|HD, E, D)$化简为 $P(Hb|D)$。

- $P(CP|Hb, HD, E, D)$化简为 $P(CP|Hb, HD)$。
- $P(BP|CP, Hb, HD, E, D)$化简为 $P(BP|HD)$。

基于以上条件概率，创建结点之间的弧(E, HD), (D, HD), (D, Hb), (HD, CP), (Hb, CP)和(HD, BP)。这些弧构成了图 5-13 所示的网络结构。 □

算法 5.3 保证生成的拓扑结构不包含环，这一点也很容易证明。如果存在环，那么至少有一条弧从低序结点指向高序结点，并且至少存在另一条弧从高序结点指向低序结点。由于算法 5.3 不允许从低序结点到高序结点的弧存在，因此拓扑结构中不存在环。

然而，如果我们对变量采用不同的排序方案，得到的网络拓扑结构可能会有变化。某些拓扑结构可能质量很差，因为它在不同的结点对之间产生了很多条弧。从理论上讲，可能需要检查所有 $d!$种可能的排序才能确定最佳的拓扑结构，这是一项计算开销很大的任务。替代的方法是把变量分为原因变量和结果变量，然后从各原因变量向其对应的结果变量画弧。这种方法简化了贝叶斯网络结构的建立。

一旦找到了合适的拓扑结构，与各结点关联的概率表就确定了。对这些概率的估计比较容易，与朴素贝叶斯分类器中所用的方法类似。

3. 使用 BBN 进行推理举例

假设我们对使用图 5-13 中的 BBN 来诊断一个人是否患有心脏病感兴趣。下面阐释在不同的情况下如何做出诊断。

情况一：没有先验信息 在没有任何先验信息的情况下，可以通过计算先验概率 $P(\text{HD=Yes})$和 $P(\text{HD=No})$来确定一个人是否可能患心脏病。为了表述方便，设$\alpha \in \{\text{Yes}, \text{No}\}$表示锻炼的两个值，$\beta \in \{\text{健康}, \text{不健康}\}$表示饮食的两个值。

$$\begin{aligned} P(\text{HD=Yes}) &= \sum_{\alpha}\sum_{\beta} P(\text{HD} = \text{Yes} \mid E = \alpha, D = \beta)P(E = \alpha, D = \beta) \\ &= \sum_{\alpha}\sum_{\beta} P(\text{HD} = \text{Yes} \mid E = \alpha, D = \beta)P(E = \alpha)P(D = \beta) \\ &= 0.25\times0.7\times0.25 + 0.45\times0.7\times0.75 + 0.55\times0.3\times0.25 + 0.75\times0.3\times0.75 \\ &= 0.49 \end{aligned}$$

因为 $P(\text{HD=No}) = 1 - P(\text{HD=Yes}) = 0.51$，所以，此人不得心脏病的机率略微大一点。

情况二：高血压 如果一个人有高血压，可以通过比较后验概率 $P(\text{HD=Yes} \mid \text{BP=高})$和 $P(\text{HD=No} \mid \text{BP=高})$来诊断他是否患有心脏病。为此，我们必须先计算 $P(\text{BP=高})$：

$$\begin{aligned} P(\text{BP} = \text{高}) &= \sum_{\gamma} P(\text{BP} = \text{高} \mid \text{HD} = \gamma)P(\text{HD} = \gamma) \\ &= 0.85\times0.49 + 0.2\times0.51 = 0.5185 \end{aligned}$$

其中$\gamma \in \{\text{Yes}, \text{No}\}$。因此，此人患心脏病的后验概率是：

$$\begin{aligned} P(\text{HD=Yes} \mid \text{BP=高}) &= \frac{P(\text{BP} = \text{高} \mid \text{HD} = \text{Yes})P(\text{HD} = \text{Yes})}{P(\text{BP} = \text{高})} \\ &= \frac{0.85\times0.49}{0.5185} = 0.8033 \end{aligned}$$

同理，$P(\text{HD=No}|\text{BP=高}) = 1-0.8033 = 0.1967$。因此，当一个人有高血压时，他患心脏病的危险就增加了。

情况三：高血压、饮食健康、经常锻炼身体　假设得知此人经常锻炼身体并且饮食健康。这些新信息会对诊断造成怎样的影响呢？加上这些新信息，此人患心脏病的后验概率：

$$
\begin{aligned}
&P(\text{HD=Yes}|\text{BP=高}, D=\text{健康}, E=\text{Yes}) \\
&= \left[\frac{P(\text{BP}=\text{高}\,|\,\text{HD}=\text{Yes}, D=\text{健康}, E=\text{Yes})}{P(\text{BP}=\text{高}\,|\,D=\text{健康}, E=\text{Yes})}\right] \times P(\text{HD}=\text{Yes}\,|\,D=\text{健康}, E=\text{Yes}) \\
&= \frac{P(\text{BP}=\text{高}\,|\,\text{HD}=\text{Yes})P(\text{HD}=\text{Yes}\,|\,D=\text{健康}, E=\text{Yes})}{\sum_{\gamma} P(\text{BP}=\text{高}\,|\,\text{HD}=\gamma)P(\text{HD}=\gamma\,|\,D=\text{健康}, E=\text{Yes})} \\
&= \frac{0.85\times 0.25}{0.85\times 0.25 + 0.2\times 0.75} \\
&= 0.5862
\end{aligned}
$$

而此人不患心脏病的概率是：

$$P(\text{HD=No}|\ \text{BP}=\text{高}, D=\text{健康}, E=\text{Yes}) = 1-0.5862 = 0.4138$$

因此模型暗示健康的饮食和有规律的体育锻炼可以降低患心脏病的危险。

4. BBN 的特点

下面是 BBN 模型的一般特点。

(1) BBN 提供了一种用图形模型来捕获特定领域的先验知识的方法。网络还可以用来对变量间的因果依赖关系进行编码。

(2) 构造网络可能既费时又费力。然而，一旦网络结构确定下来，添加新变量就十分容易。

(3) 贝叶斯网络很适合处理不完整的数据。对有属性遗漏的实例可以通过对该属性的所有可能取值的概率求和或求积分来加以处理。

(4) 因为数据和先验知识以概率的方式结合起来了，所以该方法对模型的过分拟合问题是非常鲁棒的。

5.4　人工神经网络

人工神经网络（ANN）的研究是由试图模拟生物神经系统而激发的。人类的大脑主要由称为**神经元**（neuron）的神经细胞组成，神经元通过叫作**轴突**（axon）的纤维丝连在一起。当神经元受到刺激时，神经脉冲通过轴突从一个神经元传到另一个神经元。一个神经元通过**树突**（dendrite）连接到其他神经元的轴突，树突是神经元细胞体的延伸物。树突和轴突的连接点叫作**神经键**（synapse）。神经学家发现，人的大脑通过在同一个脉冲反复刺激下改变神经元之间的神经键连接强度来进行学习。

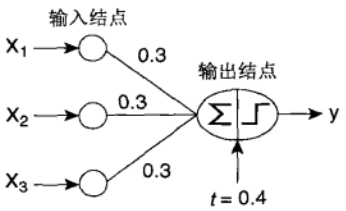
类似于人脑的结构，ANN 由一组相互连接的结点和有向链构成。本节将分析一系列 ANN 模型，从介绍最简单的模型——**感知器**（perceptron）开始，看看如何训练这种模型来解决分类问题。

5.4.1 感知器

考虑图 5-14 中的图表。左边的表显示一个数据集，包含三个布尔变量(x_1, x_2, x_3)和一个输出变量 y，当三个输入中至少有两个是 0 时，y 取−1，而至少有两个大于 0 时，y 取+1。

X_1	X_2	X_3	y
1	0	0	−1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	−1
0	1	0	−1
0	1	1	1
0	0	0	−1

(a) 数据集



(b) 感知器

图 5-14 使用感知器模拟一个布尔函数

图 5-14b 展示了一个简单的神经网络结构——感知器。感知器包含两种结点：几个输入结点，用来表示输入属性；一个输出结点，用来提供模型输出。神经网络结构中的结点通常叫作神经元或单元。在感知器中，每个输入结点都通过一个加权的链连接到输出结点。这个加权的链用来模拟神经元间神经键连接的强度。像生物神经系统一样，训练一个感知器模型就相当于不断调整链的权值，直到能拟合训练数据的输入输出关系为止。

感知器对输入加权求和，再减去偏置因子 t，然后考察结果的符号，得到输出值 $\hat{y}$。图 5-14b 中的模型有三个输入结点，各结点到输出结点的权值都等于 0.3，偏置因子 $t = 0.4$。模型的输出计算公式如下：

$$\hat{y} = \begin{cases} 1 & \text{如果} 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1 & \text{如果} 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0 \end{cases} \tag{5-21}$$

例如，如果 $x_1 = 1$，$x_2 = 1$，$x_3 = 0$，那么 $\hat{y} = +1$，因为 $0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4$ 是正的。另外，如果 $x_1 = 0$，$x_2 = 1$，$x_3 = 0$，那么 $\hat{y} = -1$，因为加权和减去偏置因子值为负。

注意感知器的输入结点和输出结点之间的区别。输入结点简单地把接收到的值传送给输出链，而不作任何转换。输出结点则是一个数学装置，计算输入的加权和，减去偏置项，然后根据结果的符号产生输出。更具体地，感知器模型的输出可以用如下数学方式表示：

$$\hat{y} = sign(w_d x_d + w_{d-1} x_{d-1} + \cdots + w_2 x_2 + w_1 x_1 - t) \tag{5-22}$$

其中，$w_1, w_2, \cdots, w_d$ 是输入链的权值，而 $x_1, x_2, \cdots, x_d$ 是输入属性值。符号函数，作为输出神经元的**激活函数**（activation function），当参数为正时输出+1，参数为负时输出−1。感知器模型可以写成下面更简洁的形式：

$$\hat{y} = sign[w_d x_d + w_{d-1} x_{d-1} + \cdots + w_1 x_1 + w_0 x_0] = sign(\mathbf{w} \cdot \mathbf{x}) \tag{5-23}$$

其中，$w_0 = -t$，$x_0 = 1$，$\mathbf{w} \cdot \mathbf{x}$ 是权值向量 $\mathbf{w}$ 和输入属性向量 $\mathbf{x}$ 的点积。

学习感知器模型

在感知器模型的训练阶段，权值参数 $\mathbf{w}$ 不断调整直到输出和训练样例的实际输出一致。算法 5.4 中给出了感知器学习算法的概述。

算法 5.4　感知器学习算法

1：令 $D=\{(\mathbf{x}_i,y_i)|i=1,2,\cdots,N\}$是训练样例集
2：用随机值初始化权值向量 $\mathbf{w}^{(0)}$
3：**repeat**
4：　**for** 每个训练样例$(\mathbf{x}_i, y_i)\in D$ **do**
5：　　计算预测输出 $\hat{y}_i^{(k)}$
6：　　**for** 每个权值 w_j **do**
7：　　更新权值 $w_j^{(k+1)}=w_j^{(k)}+\lambda(y_i-\hat{y}_i^{(k)})x_{ij}$
8：　　**end for**
9：　**end for**
10：**until** 满足终止条件

算法的主要计算是第 7 步中的权值更新公式：

$$w_j^{(k+1)}=w_j^{(k)}+\lambda(y_i-\hat{y}_i^{(k)})x_{ij} \tag{5-24}$$

其中 $w^{(k)}$是第 k 次循环后第 i 个输入链上的权值，参数λ称为**学习率**（learning rate），x_{ij}是训练样例 $\mathbf{x}_i$ 的第 j 个属性值。权值更新公式的理由是相当直观的。从公式（5-24）可以看出，新权值 $w^{(k+1)}$是等于旧权值 $w^{(k)}$加上一个正比于预测误差$(y-\hat{y})$的项。如果预测正确，那么权值保持不变。否则，按照如下方法更新。

- 如果 $y=+1$，$\hat{y}=-1$，那么预测误差$(y-\hat{y})=2$。为了补偿这个误差，需要通过提高所有正输入链的权值、降低所有负输入链的权值来提高预测输出值。
- 如果 $y=-1$，$\hat{y}=+1$，那么预测误差$(y-\hat{y})=-2$。为了补偿这个误差，我们需要通过降低所有正输入链的权值、提高所有负输入链的权值来减少预测输出值。

在权值更新公式中，对误差项影响最大的链需要的调整最大。然而，权值不能改变太大，因为仅仅对当前训练样例计算了误差项。否则的话，以前的循环中所作的调整就会失效。学习率λ，其值在 0 和 1 之间，可以用来控制每次循环时的调整量。如果λ接近 0，那么新权值主要受旧权值的影响；相反，如果λ接近 1，则新权值对当前循环中的调整量更加敏感。在某些情况下，可以使用一个自适应的λ值：λ在前几次循环时值相对较大，而在接下来的循环中逐渐减小。

公式（5-23）中所示的感知器模型关于参数 **w** 和属性 **x** 是线性的。因此，设 $\hat{y}=0$，得到的感知器的决策边界是一个把数据分为–1 和+1 两个类的线性超平面。图 5-15 显示了把感知器学习算法应用到图 5-14 中的数据集上所得到的决策边界。对于线性可分的分类问题，感知器学习算法保证收敛到一个最优解（只要学习率足够小）。如果问题不是线性可分的，那么算法就不会收敛。图 5-16 给出了一个由 XOR 函数得到的非线性可分数据的例子。感知器找不到该数据的正确解，因为没有线性超平面可以把训练实例完全分开。

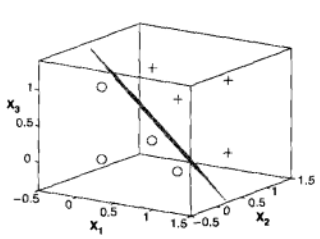


图 5-15　图 5-14 中的数据的感知器决策边界

x_1	x_2	y
0	0	−1
1	0	1
0	1	1
1	1	−1

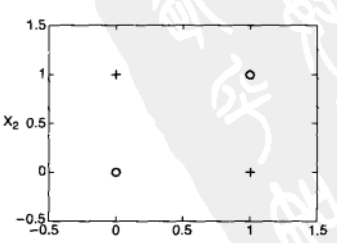


图 5-16　XOR 分类问题。没有线性超平面可以分开这两个类

5.4.2 多层人工神经网络

人工神经网络结构比感知器模型更复杂。这些额外的复杂性来源于多个方面。

(1) 网络的输入层和输出层之间可能包含多个中间层，这些中间层叫作**隐藏层**（hidden layer），隐藏层中的结点称为**隐藏结点**（hidden node）。这种结构称为多层神经网络（见图 5-17）。在**前馈**（feed-forward）神经网络中，每一层的结点仅和下一层的结点相连。感知器就是一个单层的前馈神经网络，因为它只有一个结点层——输出层——进行复杂的数学运算。在**递归**（recurrent）神经网络中，允许同一层结点相连或一层的结点连到前面各层中的结点。

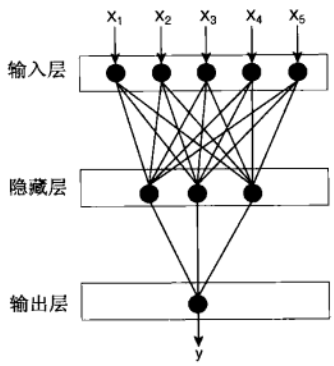


图 5-17 多层前馈人工神经网络（ANN）举例

(2) 除了符号函数外，网络还可以使用其他激活函数，如图 5-18 所示的线性函数、S 型（逻辑斯缔）函数、双曲正切函数等。这些激活函数允许隐藏结点和输出结点的输出值与输入参数呈非线性关系。

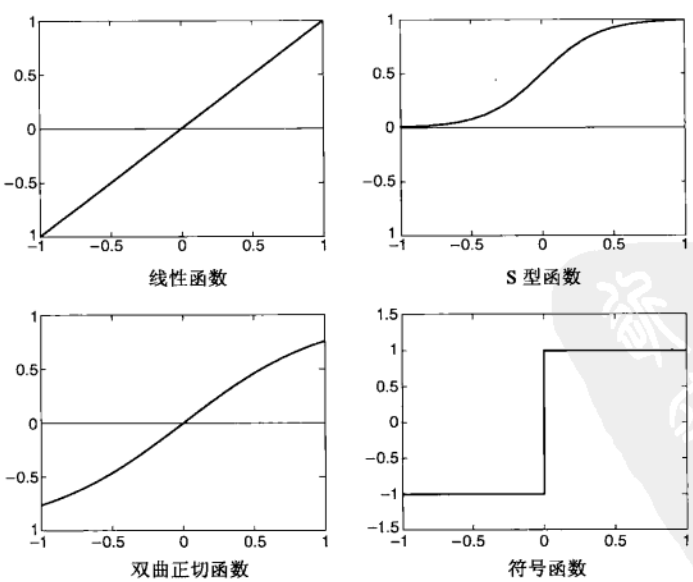
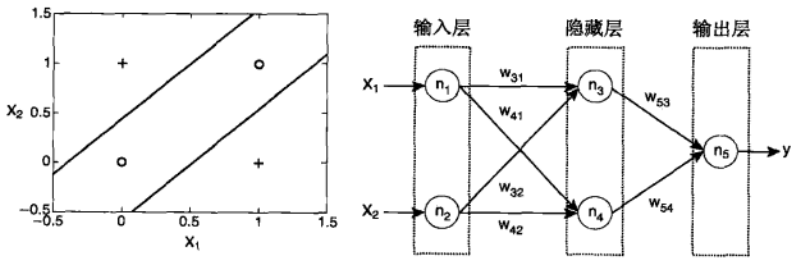


图 5-18 人工神经网络中激活函数的类型

这些附加的复杂性使得多层神经网络可以对输入和输出变量间更复杂的关系建模。例如，考虑上一节中描述的 XOR 问题。实例可以用两个超平面进行分类，这两个超平面把输入空间划分到各自的类，如图 5-19a 所示。因为感知器只能构造一个超平面，所以它无法找到最优解。该问题可以使用两层前馈神经网络加以解决，见图 5-19b。直观上，我们可以把每个隐藏结点看作一个感知器，每个感知器构造两个超平面中的一个，输出结点简单地综合各感知器的结果，得到的决策边界如图 5-19a 所示。



(a) 决策边界　　(b) 神经网络拓扑结构

图 5-19　XOR 问题的两层前馈神经网络

要学习 ANN 模型的权值，需要一个有效的算法，该算法在训练数据充足时可以收敛到正确的解。一种方法是把网络中的每个隐藏结点或输出结点看作一个独立的感知器单元，使用与公式（5-24）相同的权值更新公式。显然，这种方法行不通，因为缺少隐藏结点的真实输出的先验知识。这使得很难确定各隐藏结点的误差项$(y-\hat{y})$。下面介绍一种基于梯度下降的神经网络权值学习方法。

1. 学习 ANN 模型

ANN 学习算法的目的是确定一组权值 $\mathbf{w}$，最小化误差的平方和：

$$E(\mathbf{w})=\frac{1}{2}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2 \tag{5-25}$$

注意，误差平方和取决于 $\mathbf{w}$，因为预测类 $\hat{y}$ 是赋予隐藏结点和输出结点的权值的函数。图 5-20 显示了一个误差曲面的例子，该曲面是两个参数 w_1 和 w_2 的函数。当 $\hat{y}_i$ 是参数 $\mathbf{w}$ 的线性函数时，通常得到这种类型的误差曲面。如果将 $\hat{y}=\mathbf{w}\cdot\mathbf{x}$ 代入公式（5-25），则误差函数变成参数的二次函数，就可以很容易地找到全局最小解。

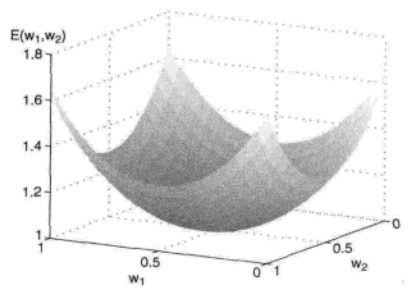


图 5-20　两个参数模型的误差曲面 $E(w_1, w_2)$

大多数情况下，由于激活函数的选择（如 S 型或双曲正切函数），ANN 的输出是参数的非线性函数。这样，就不能直接推导出 $\mathbf{w}$ 的全局最优解了。像基于梯度下降的方法等贪心算法可以很有效地求解优化问题。梯度下降方法使用的权值更新公式可以写成：

$$w_j \leftarrow w_j - \lambda \frac{\partial E(\mathbf{w})}{\partial w_j} \tag{5-26}$$

其中，λ是学习率。式中第二项说的是权值应该沿着使总体误差项减小的方向增加。然而，由于误差函数是非线性的，因此，梯度下降方法可能会陷入局部极小值。

梯度下降方法可以用来学习神经网络中输出结点和隐藏结点的权值。对于隐藏结点，学习的计算量并不小，因为在不知道输出值的情况下，很难估计结点的误差项$\partial E/\partial w_j$。一种称为**反向传播**（back-propagation）的技术可以用来解决该问题。该算法的每一次迭代包括两个阶段：前向阶段和后向阶段。在前向阶段，使用前一次迭代所得到的权值计算网络中每一个神经元的输出值。计算是向前进行的，即先计算第 k 层神经元的输出，再计算第 k+1 层的输出。在后向阶段，以相反的方向应用权值更新公式，即先更新 k+1 层的权值，再更新第 k 层的权值。使用反向传播方法，可以用第 k+1 层神经元的误差来估计第 k 层神经元的误差。

2. ANN 学习中的设计问题

在训练神经网络来学习分类任务之前，应该先考虑以下设计问题。

(1) 确定输入层的结点数目。每一个数值输入变量或二元输入变量对应一个输入结点。如果输入变量是分类变量，则可以为每一个分类值创建一个结点，也可以用$\lceil \log_2 k \rceil$个输入结点对 k 元变量进行编码。

(2) 确定输出层的结点数目。对于 2-类问题，一个输出结点足矣；而对于 k-类问题，则需要 k 个输出结点。

(3) 选择网络拓扑结构（例如，隐藏层数和隐藏结点数，前馈还是递归网络结构）。注意，目标函数表示取决于链上的权值、隐藏结点数和隐藏层数、结点的偏置以及激活函数的类型。找出合适的拓扑结构不是件容易的事。一种方法是，开始的时候使用一个有足够多的结点和隐藏层的全连接网络，然后使用较少的结点重复该建模过程。这种方法非常耗时。另一种方法是，不重复建模过程，而是删除一些结点，然后重复模型评价过程来选择合适的模型复杂度。

(4) 初始化权值和偏置。随机赋值常常是可取的。

(5) 去掉有遗漏值的训练样例，或者用最合理的值来代替。

5.4.3 人工神经网络的特点

人工神经网络的一般特点概括如下。

(1) 至少含有一个隐藏层的多层神经网络是一种**普适近似**（universal approximator），即可以用来近似任何目标函数。由于 ANN 具有丰富的假设空间，因此对于给定的问题，选择合适的拓扑结构来防止模型的过分拟合是很重要的。

(2) ANN 可以处理冗余特征，因为权值在训练过程中自动学习。冗余特征的权值非常小。

(3) 神经网络对训练数据中的噪声非常敏感。处理噪声问题的一种方法是使用确认集来确定模型的泛化误差，另一种方法是每次迭代把权值减少一个因子。

(4) ANN 权值学习使用的梯度下降方法经常会收敛到局部极小值。避免局部极小值的方法是

在权值更新公式中加上一个动量项（momentum term）。

(5) 训练 ANN 是一个很耗时的过程，特别是当隐藏结点数量很大时。然而，测试样例分类时非常快。

5.5　支持向量机

支持向量机（support vector machine, SVM）已经成为一种倍受关注的分类技术。这种技术具有坚实的统计学理论基础，并在许多实际应用（如手写数字的识别、文本分类等）中展示了大有可为的实践效用。此外，SVM 可以很好地应用于高维数据，避免了维灾难问题。这种方法具有一个独特的特点，它使用训练实例的一个子集来表示决策边界，该子集称作**支持向量**（support vector）。

为了解释 SVM 的基本思想，首先介绍**最大边缘超平面**（maximal margin hyperplane）的概念以及选择它的基本原理。然后，描述在线性可分的数据上怎样训练一个线性的 SVM，从而明确地找到这种最大边缘超平面。最后，介绍如何将 SVM 方法扩展到非线性可分的数据上。

5.5.1　最大边缘超平面

图5-21显示了一个数据集，包含属于两个不同类的样本，分别用方块和圆圈表示。这个数据集是线性可分的，即可以找到这样一个超平面，使得所有的方块位于这个超平面的一侧，而所有的圆圈位于它的另一侧。然而，正如图5-21所示，可能存在无穷多个那样的超平面。虽然它们的训练误差都等于零，但是不能保证这些超平面在未知实例上运行得同样好。根据在检验样本上的运行效果，分类器必须从这些超平面中选择一个来表示它的决策边界。

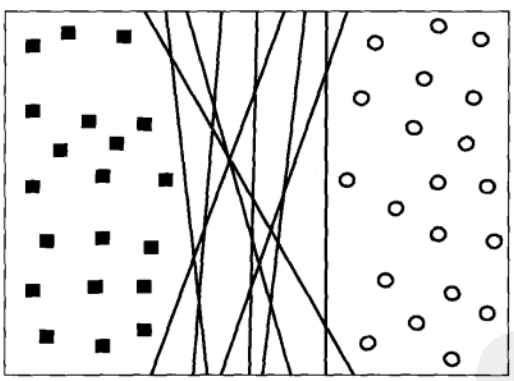


图 5-21　一个线性可分数据集上的可能决策边界

为了更好地理解不同的超平面对泛化误差的影响，考虑两个决策边界 B_1 和 B_2，如图 5-22 所示。这两个决策边界都能准确无误地将训练样本划分到各自的类中。每个决策边界 B_i 都对应着一对超平面，分别记为 b_{i1} 和 b_{i2}。其中，b_{i1} 是这样得到的：平行移动一个和决策边界平行的超平面，直到触到最近的方块为止；类似地，平行移动一个和决策边界平行的超平面，直到触到最近的圆圈，可以得到 b_{i2}。这两个超平面之间的间距称为分类器的边缘。通过图 5-22 中的图解，注意到 B_1 的边缘显著大于 B_2 的边缘。在这个例子中，B_1 就是训练样本的最大边缘超平面。

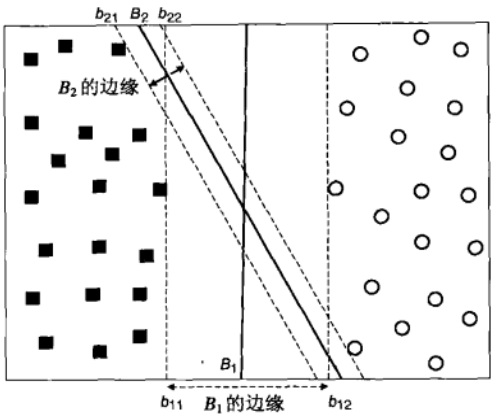


图 5-22 决策边界的边缘

最大边缘的基本原理

具有较大边缘的决策边界比那些具有较小边缘的决策边界具有更好的泛化误差。直觉上，如果边缘比较小，决策边界任何轻微的扰动都可能对分类产生显著的影响。因此，那些决策边界边缘较小的分类器对模型的过分拟合更加敏感，从而在未知的样本上的泛化能力很差。

统计学习理论给出了线性分类器边缘与其泛化误差之间关系的形式化解释，我们称这种理论为**结构风险最小化**（structural risk minimization, SRM）理论。该理论根据分类器的训练误差 R_e、训练样本数 N 和模型的复杂度 h（即它的**能力**（capacity）），给出了分类器的泛化误差的一个上界 R。具体地说，在概率 $1-\eta$ 下，分类器的泛化误差在最坏情况下满足

$$R \leqslant R_e + \varphi\left(\frac{h}{N}, \frac{\log(\eta)}{N}\right) \tag{5-27}$$

其中，φ是能力 h 的单调增函数。上面的不等式读者可能感觉很熟悉，这是因为它和 4.4.4 节最小描述长度（MDL）原理中的等式十分相似。在这一点上，SRM 是泛化误差的另外一种表达方式，它体现了训练误差和模型复杂度之间的折中。

线性模型的能力与它的边缘逆相关。具有较小边缘的模型具有较高的能力，因为与具有较大边缘的模型不同，具有较小边缘的模型更灵活、能拟合更多的训练集。然而，依据 SRM 原理，随着能力增加，泛化误差的上界也随之提高。因此，需要设计最大化决策边界的边缘的线性分类器，以确保最坏情况下的泛化误差最小。**线性 SVM**（linear SVM）就是这样的分类器，下一节将要详细介绍。

5.5.2 线性支持向量机：可分情况

线性 SVM 是这样一个分类器，它寻找具有最大边缘的超平面，因此它也经常被称为**最大边缘分类器**（maximal margin classifier）。为了深刻理解 SVM 是如何学习这样的边界的，我们首先对线性分类器的决策边界和边缘进行一些初步的讨论。

1. 线性决策边界

考虑一个包含 N 个训练样本的二元分类问题。每个样本表示为一个二元组$(\mathbf{x}_i, y_i)$（$i=1, 2, \cdots, N$），其中 $\mathbf{x}_i=(x_{i1}, x_{i2}, \cdots, x_{id})^T$，对应于第 i 个样本的属性集。为方便计，令 $y_i \in \{-1,1\}$表示它的类标

号。一个线性分类器的决策边界可以写成如下形式：

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{5-28}$$

其中，$\mathbf{w}$ 和 b 是模型的参数。

图 5-23 显示了包含圆圈和方块的二维训练集。图中的实线表示决策边界，它将训练样本一分为二，划入各自的类中。任何位于决策边界上的样本都必须满足公式（5-28）。例如，如果 $\mathbf{x}_a$ 和 $\mathbf{x}_b$ 是两个位于决策边界上的点，则

$$\mathbf{w} \cdot \mathbf{x}_a + b = 0$$
$$\mathbf{w} \cdot \mathbf{x}_b + b = 0$$

两个方程相减便得到：

$$\mathbf{w} \cdot (\mathbf{x}_b - \mathbf{x}_a) = 0$$

其中，$\mathbf{x}_b - \mathbf{x}_a$ 是一个平行于决策边界的向量，它的方向是从 $\mathbf{x}_a$ 到 $\mathbf{x}_b$。由于点积的结果为零，因此 $\mathbf{w}$ 的方向必然垂直于决策边界，如图 5-23 所示。

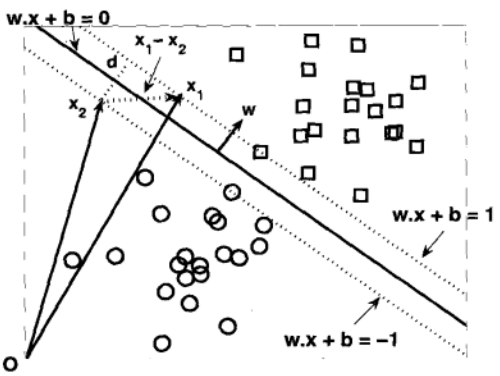


图 5-23　SVM 的决策边界和边缘

对于任何位于决策边界上方的方块 $\mathbf{x}_s$，我们可以证明：

$$\mathbf{w} \cdot \mathbf{x}_s + b = k \tag{5-29}$$

其中 $k > 0$。同理，对于任何位于决策边界下方的圆圈 $\mathbf{x}_c$，我们可以证明：

$$\mathbf{w} \cdot \mathbf{x}_c + b = k' \tag{5-30}$$

其中 $k' < 0$。如果标记所有的方块的类标号为+1，标记所有的圆圈的类标号为−1，则可以用以下的方式预测任何测试样本 $\mathbf{z}$ 的类标号 y：

$$y = \begin{cases} 1 & \text{如果} \mathbf{w} \cdot \mathbf{z} + b > 0 \\ -1 & \text{如果} \mathbf{w} \cdot \mathbf{z} + b < 0 \end{cases} \tag{5-31}$$

2. 线性分类器的边缘

考虑那些离决策边界最近的方块和圆圈。由于该方块位于决策边界的上方，因此对于某个正值 k，它必然满足公式（5-29）；而对于某个负值 k'，圆圈必须满足公式（5-30）。调整决策边界的参数 $\mathbf{w}$ 和 b，两个平行的超平面 b_{i1} 和 b_{i2} 可以表示如下：

$$b_{i1}: \mathbf{w}\cdot\mathbf{x}+b=1 \tag{5-32}$$

$$b_{i2}: \mathbf{w}\cdot\mathbf{x}+b=-1 \tag{5-33}$$

决策边界的边缘由这两个超平面之间的距离给定。为了计算边缘，令 $\mathbf{x}_1$ 是 b_{i1} 上的一个数据点，$\mathbf{x}_2$ 是 b_{i2} 上的一个数据点，如图 5-23 所示。将 $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 分别代入公式（5-32）和公式（5-33）中，则边缘 d 可以通过两式相减得到：

$$\begin{aligned}\mathbf{w}\cdot(\mathbf{x}_1-\mathbf{x}_2)&=2\\ \|\mathbf{w}\|\times d&=2\\ \therefore d&=\frac{2}{\|\mathbf{w}\|}\end{aligned} \tag{5-34}$$

3. 学习线性 SVM 模型

SVM 的训练阶段包括从训练数据中估计决策边界的参数 $\mathbf{w}$ 和 b。选择的参数必须满足下面两个条件：

$$\begin{aligned}\mathbf{w}\cdot\mathbf{x}_i+b&\geqslant 1 \quad \text{如果 } y_i=1\\ \mathbf{w}\cdot\mathbf{x}_i+b&\leqslant -1 \quad \text{如果 } y_i=-1\end{aligned} \tag{5-35}$$

这些条件要求所有类标号为 1 的训练实例（即方块）都必须位于超平面 $\mathbf{w}\cdot\mathbf{x}+b=1$ 上或位于它的上方，而那些类标号为–1 的训练实例（即圆圈）都必须位于超平面 $\mathbf{w}\cdot\mathbf{x}+b=-1$ 上或位于它的下方。这两个不等式可以概括为如下更紧凑的形式：

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geqslant 1, \quad i=1,2,\cdots,N \tag{5-36}$$

尽管前面的条件也可以用于其他线性分类器（包括感知器），但是 SVM 增加了一个要求：其决策边界的边缘必须是最大的。然而，最大化边缘等价于最小化下面的目标函数：

$$f(\mathbf{w})=\frac{\|\mathbf{w}\|^2}{2} \tag{5-37}$$

定义 5.1 线性 SVM：可分情况 SVM 的学习任务可以形式化地描述为以下被约束的优化问题：

$$\min_{\mathbf{w}}\frac{\|\mathbf{w}\|^2}{2}$$

$$\text{受限于} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geqslant 1, \quad i=1,2,\cdots,N$$

由于目标函数是二次的，而约束在参数 $\mathbf{w}$ 和 b 上是的线性的，因此这个问题是一个**凸**（convex）优化问题，可以通过标准的**拉格朗日乘子**（Lagrange multiplier）方法求解。下面简要介绍一下求解这个优化问题的主要思想。

首先，必须改写目标函数，考虑施加在解上的约束。新目标函数称为该优化问题的拉格朗日函数：

$$L_P=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{N}\lambda_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1\right) \tag{5-38}$$

其中，参数 λ_i 称为拉格朗日乘子。拉格朗日函数中的第一项与原目标函数相同，而第二项则捕获了不等式约束。为了理解改写原目标函数的必要性，考虑公式（5-37）给出的原目标函数。容易证明当 $\mathbf{w}=0$（即零向量，它的每一个分量均为 0）时函数取得最小值。然而，这样的解违背了定义 5.1 中给出的约束条件，因为 b 没有可行解。事实上，如果 $\mathbf{w}$ 和 b 的解违反不等式约束，即

如果 $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1<0$，则解是不可行的。公式（5-38）给出的拉格朗日函数通过从原目标函数减去约束条件的方式合并了约束条件。假定$\lambda_i\geqslant 0$，则任何不可行解仅仅是增加了拉格朗日函数的值。

为了最小化拉格朗日函数，必须对 L_P 关于 $\mathbf{w}$ 和 b 求偏导，并令它们等于零：

$$\frac{\partial L_P}{\partial \mathbf{w}}=0\Rightarrow \mathbf{w}=\sum_{i=1}^{N}\lambda_i y_i\mathbf{x}_i \tag{5-39}$$

$$\frac{\partial L_P}{\partial_b}=0\Rightarrow \sum_{i=1}^{N}\lambda_i y_i=0 \tag{5-40}$$

因为拉格朗日乘子是未知的，因此我们仍然不能得到 $\mathbf{w}$ 和 b 的解。如果定义 5.1 只包含等式约束，而不是不等式约束，则我们可以利用从该等式约束中得到的 N 个方程，加上公式（5-39）和公式（5-40），从而得到 $\mathbf{w}$，b 和λ_i的可行解。注意，等式约束的拉格朗日乘子是可以取任意值的自由参数。

处理不等式约束的一种方法就是把它变换成一组等式约束。只要限制拉格朗日乘子非负，这种变换便是可行的。这种变换导致如下拉格朗日乘子约束，称作 Karuch-Kuhn-Tucher（KKT）条件：

$$\lambda_i \geqslant 0 \tag{5-41}$$

$$\lambda_i[y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1]=0 \tag{5-42}$$

乍一看，拉格朗日乘子的数目好像和训练样本的数目一样多。事实上，应用公式（5-42）给定的约束后，许多拉格朗日乘子都变为零。该约束表明，除非训练实例满足方程 $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)=1$，否则拉格朗日乘子$\lambda_i$必须为零。那些$\lambda_i>0$ 的训练实例位于超平面 b_{i1} 或 b_{i2} 上，称为支持向量。不在这些超平面上的训练实例肯定满足$\lambda_i=0$。公式（5-39）和公式（5-42）还表明，定义决策边界的参数 $\mathbf{w}$ 和 b 仅依赖于这些支持向量。

对前面的优化问题求解仍是一项十分棘手的任务，因为它涉及大量参数：$\mathbf{w}$，b 和λ_i。通过将拉格朗日函数变换成仅包含拉格朗日乘子的函数（称作对偶问题），可以简化该问题。为了变换成对偶问题，首先将公式（5-39）和公式（5-40）代入到公式（5-38）中。这将导致该优化问题的如下对偶公式：

$$L_D=\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j\mathbf{x}_i\cdot\mathbf{x}_j \tag{5-43}$$

对偶拉格朗日函数和原拉格朗日函数的主要区别如下：

(1) 对偶拉格朗日函数仅涉及拉格朗日乘子和训练数据，而原拉格朗日函数除涉及拉格朗日乘子外还涉及决策边界的参数。尽管如此，这两个优化问题的解是等价的。

(2) 公式（5-43）中的二次项前有个负号，这说明原来涉及拉格朗日函数 L_p 的最小化问题已经变换成了涉及对偶拉格朗日函数 L_D 的最大化问题。

对于大型数据集，对偶优化问题可以使用数值计算技术来求解，如使用二次规划（已经超出本书的范围）。一旦找到一组λ_i，就可以通过公式（5-39）和公式（5-42）来求得 $\mathbf{w}$ 和 b 的可行解。决策边界可以表示成：

$$\left(\sum_{i=1}^{N}\lambda_i y_i\mathbf{x}_i\cdot\mathbf{x}\right)+b=0 \tag{5-44}$$

b 可以通过求解支持向量公式（5-42）得到。由于λ_i是通过数值计算得到的，因此可能存在数值误差，计算出的 b 值可能不唯一。它取决于公式（5-42）中使用的支持向量。实践中，使用 b 的平均值作为决策边界的参数。

例 5.5 考虑图 5-24 给出的二维数据集，它包含 8 个训练实例。使用二次规划方法，可以求解公式（5-43）给出的优化问题，得到每一个训练实例的拉格朗日乘子λ_i，如表的最后一列所示。注意，仅前面两个实例具有非零的拉格朗日乘子。这些实例对应于该数据集的支持向量。

令 $\mathbf{w}=(w_1, w_2)$，b 为决策边界的参数。使用公式（5-39），我们可以按如下方法求解 w_1 和 w_2：

$$w_1 = \sum_i \lambda_i y_i x_{i1} = 65.5261 \times 1 \times 0.3858 + 65.5261 \times (-1) \times 0.4871 = -6.64$$

$$w_2 = \sum_i \lambda_i y_i x_{i2} = 65.5261 \times 1 \times 0.4687 + 65.5261 \times (-1) \times 0.611 = -9.32$$

偏倚项 b 可以使用公式（5-42）对每个支持向量进行计算：

$$b^{(1)} = 1 - \mathbf{w} \cdot \mathbf{x}_1 = 1 - (-6.64)(0.3858) - (-9.32)(0.4687) = 7.9300$$

$$b^{(2)} = 1 - \mathbf{w} \cdot \mathbf{x}_2 = -1 - (-6.64)(0.4871) - (-9.32)(0.611) = 7.9289$$

对这些值取平均，得到 $b = 7.93$。对应于这些参数的决策边界显示在图 5-24 中。 □

x_1	x_2	y	拉格朗日乘子
0.3858	0.4687	1	65.5261
0.4871	0.611	−1	65.5261
0.9218	0.4103	−1	0
0.7382	0.8936	−1	0
0.1763	0.0579	1	0
0.4057	0.3529	1	0
0.9355	0.8132	−1	0
0.2146	0.0099	1	0

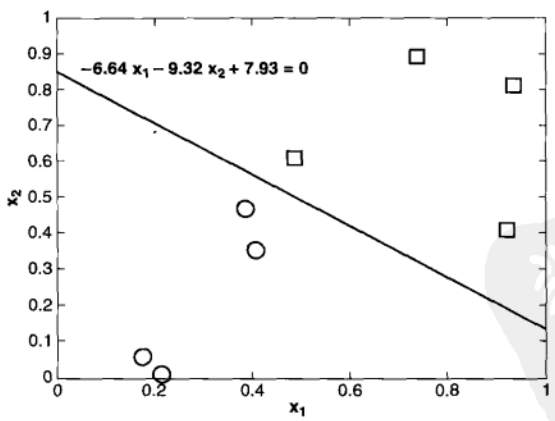


图 5-24 一个线性可分数据集的例子

确定了决策边界的参数之后，检验实例 $\mathbf{z}$ 就可以按以下的公式来分类：

$$f(\mathbf{z}) = sign(\mathbf{w} \cdot \mathbf{z} + b) = sign(\sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i \cdot \mathbf{z} + b)$$

如果 $f(\mathbf{z}) = 1$，则检验实例被分为到正类，否则分到负类。

5.5.3　线性支持向量机：不可分情况

图 5-25 给出了一个和图 5-22 相似的数据集，不同处在于它包含了两个新样本 P 和 Q。尽管决策边界 B_1 误分类了新样本，而 B_2 正确分类了它们，但是这并不表示 B_2 是一个比 B_1 好的决策边界，因为这些新样本可能只是训练数据集中的噪声。B_1 可能仍然比 B_2 更可取，因为它具有较宽的边缘，从而对过分拟合不太敏感。然而，上一节给出的 SVM 公式只能构造没有错误的决策边界。这一节考察如何修正公式，利用一种称为**软边缘**（soft margin）的方法，学习允许一定训练错误的决策边界。更为重要的是，本节给出的方法允许 SVM 在一些类线性不可分的情况下构造线性的决策边界。为了做到这一点，SVM 学习算法必须考虑边缘的宽度与线性决策边界允许的训练错误数目之间的折中。

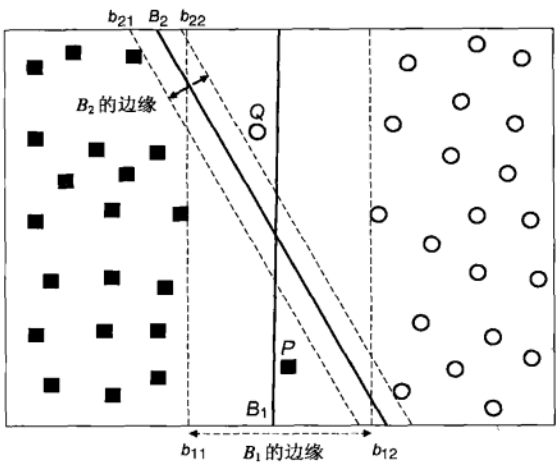


图 5-25　不可分情况下 SVM 的决策边界

尽管公式（5-37）给定的原目标函数仍然是可用的，但是决策边界 B_1 不再满足公式（5-36）给定的所有约束。因此，必须放松不等式约束，以适应非线性可分数据。可以通过在优化问题的约束中引入正值的**松弛变量**（slack variable）ξ来实现，如下式所示：

$$\begin{aligned} \mathbf{w}\cdot\mathbf{x}_i + b \geqslant 1 - \xi_i \quad & \text{如果 } y_i = 1 \\ \mathbf{w}\cdot\mathbf{x}_i + b \leqslant -1 + \xi_i \quad & \text{如果 } y_i = -1 \end{aligned} \tag{5-45}$$

其中，$\forall i: \xi_i > 0$。

为了理解松弛变量ξ_i的意义，考虑图 5-26。圆圈 **P** 是一个实例，它违反公式（5-35）给定的约束。设 $\mathbf{w}\cdot\mathbf{x} + b = -1 + \xi$是一条经过点 **P**，且平行于决策边界的直线。可以证明它与超平面 $\mathbf{w}\cdot\mathbf{x} + b = -1$ 之间的距离为$\xi/\|\mathbf{w}\|$。因此，ξ提供了决策边界在训练样本 **P** 上的误差估计。

理论上，可以使用和前面相同的目标函数，然后加上公式（5-45）给定的约束来确定决策边界。然而，由于在决策边界误分样本的数量上没有限制，学习算法可能会找到这样的决策边界，它的边缘很宽，但是误分了许多训练实例，如图5-27所示。为了避免这个问题，必须修改目标函数，以惩罚那些松弛变量值很大的决策边界。修改后的目标函数如下：

$$f(\mathbf{w})=\frac{\|\mathbf{w}\|^2}{2}+C\left(\sum_{i=1}^{N}\xi_i\right)^k$$

其中 C 和 k 是用户指定的参数，表示对误分训练实例的惩罚。为了简化该问题，在本节的剩余部分假定 $k=1$。参数 C 可以根据模型在确认集上的性能选择。

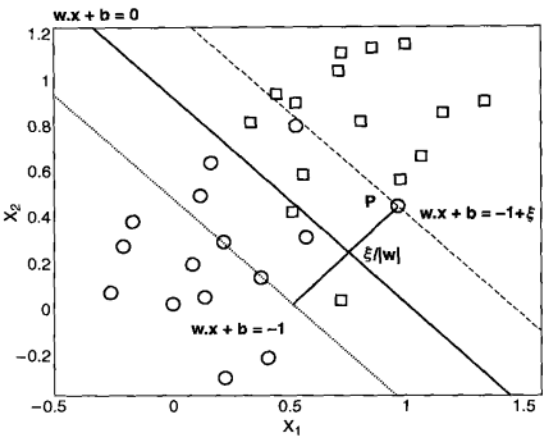


图 5-26 不可分数据的松弛变量

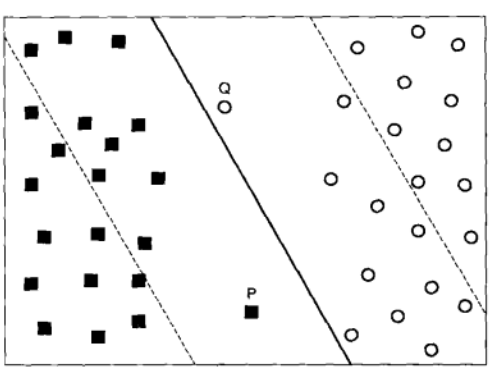


图 5-27 一个具有宽边缘但训练误差很高的决策边界

由此，被约束的优化问题的拉格朗日函数可以记作如下形式：

$$L_P=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\lambda_i\{y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1+\xi_i\}-\sum_{i=1}^{N}\mu_i\xi_i \tag{5-46}$$

其中，前面两项是需要最小化的目标函数，第三项表示与松弛变量相关的不等式约束，而最后一项是要求 ξ_i 的值非负的结果。此外，利用如下的 KKT 条件，可以将不等式约束变换成等式约束：

$$\xi_i\geqslant 0,\ \lambda_i\geqslant 0,\ \mu_i\geqslant 0 \tag{5-47}$$

$$\lambda_i\{y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1+\xi_i\}=0 \tag{5-48}$$

$$\mu_i\xi_i=0 \tag{5-49}$$

注意，公式（5-48）中的拉格朗日乘子 λ_i 是非零的当且仅当训练实例位于直线 $\mathbf{w}\cdot\mathbf{x}_i+b=\pm 1$ 上或

$\xi_i > 0$。另一方面，对于许多误分类的训练实例（即满足$\xi_i > 0$），公式（5-49）中的拉格朗日乘子μ_i都为零。

令L关于$\mathbf{w}, b$和ξ_i的一阶导数为零，就得到如下公式：

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{i=1}^{N} \lambda_i y_i x_{ij} = 0 \Rightarrow w_j = \sum_{i=1}^{N} \lambda_i y_i x_{ij} \tag{5-50}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \lambda_i y_i = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0 \tag{5-51}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0 \Rightarrow \lambda_i + \mu_i = C \tag{5-52}$$

将公式（5-50）、（5-51）和（5-52）代入拉格朗日函数中，得到如下的对偶拉格朗日函数：

$$\begin{aligned} L_D &= \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + C\sum_i \xi_i \\ &\quad -\sum_i \lambda_i \{y_i (\sum_j \lambda_j y_j \mathbf{x}_i \cdot \mathbf{x}_j + b) - 1 + \xi_i\} \\ &\quad -\sum_i (C - \lambda_i)\xi_i \\ &= \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \end{aligned} \tag{5-53}$$

它与线性可分数据上的对偶拉格朗日函数相同（参见公式（5-40））。尽管如此，施加于拉格朗日乘子λ_i上的约束与在线性可分情况下略微不同。在线性可分情况下，拉格朗日乘子必须是非负的，即$\lambda_i \geqslant 0$。另一方面，公式（5-52）表明λ_i不应该超过C（由于μ_i和λ_i都是非负的）。因此，非线性可分数据的拉格朗日乘子被限制在$0 \leqslant \lambda_i \leqslant C$。

然后，可以使用二次规划技术，求对偶问题的数值解，得到拉格朗日乘子λ_i。可以将这些乘子代入公式（5-50）和KKT条件中，从而得到决策边界的参数。

5.5.4　非线性支持向量机

上一节描述的SVM公式构建一个线性的决策边界，从而把训练实例划分到它们各自的类中。本节提出了一种把SVM应用到具有非线性决策边界数据集上的方法。这里的关键在于将数据从原先的坐标空间$\mathbf{x}$变换到一个新的坐标空间$\Phi(\mathbf{x})$中，从而可以在变换后的坐标空间中使用一个线性的决策边界来划分样本。进行变换后，就可以应用上一节介绍的方法在变换后的空间中找到一个线性的决策边界。

1. 属性变换

为了说明怎样进行属性变换可以生成一个线性的决策边界，我们考察图5-28a给出的二维数据集，它包含方块（类标号$y = 1$）和圆圈（类标号$y = -1$）。数据集是这样生成的，所有的圆圈都聚集在图的中心附近，而所有的方块都分布在离中心较远的地方。可以使用下面的公式对数据集中的实例分类：

$$y(x_1, x_2) = \begin{cases} 1 & \text{如果} \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{否则} \end{cases} \tag{5-54}$$

因此，数据集的决策边界可以表示如下：

$$\sqrt{(x_1-0.5)^2+(x_2-0.5)^2}=0.2$$

这可以进一步简化为下面的二次方程：

$$x_1^2-x_1+x_2^2-x_2=-0.46$$

需要一个非线性变换Φ，将数据从原先的特征空间映射到一个新的空间，决策边界在这个空间下成为线性的。假定选择下面的变换：

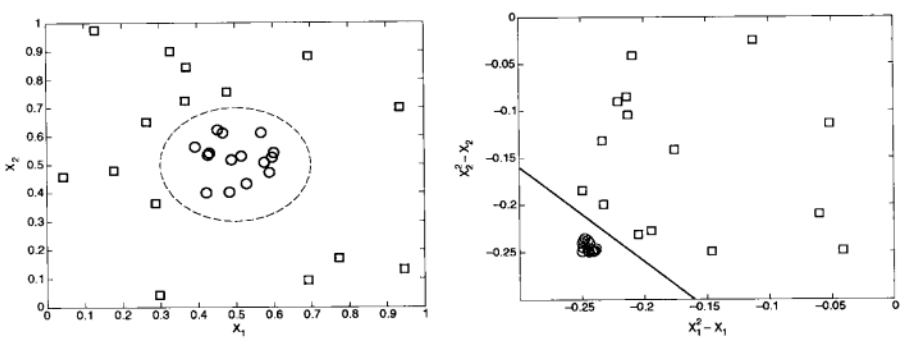
$$\Phi:(x_1,x_2)\rightarrow(x_1^2,x_2^2,\sqrt{2}x_1,\sqrt{2}x_2,\sqrt{2}x_1x_2,1) \tag{5-55}$$

在变换后的空间中，我们找到参数$\mathbf{w}=(w_0,w_1,\cdots,w_5)$，使得：

$$w_5x_1^2+w_4x_2^2+w_3\sqrt{2}x_1+w_2\sqrt{2}x_2+w_1\sqrt{2}x_1x_2+w_0=0$$

例如，对于前面给定的数据，以$x_1^2-x_1$和$x_2^2-x_2$为坐标绘图。图 5-28b 显示在变换后的空间中，所有的圆圈都位于图的左下方。因此，可以构建一个线性的决策边界从而把数据划分到各自所属的类中。

这种方法的一个潜在问题是，对于高维数据可能产生维灾难，在本节稍后，将介绍非线性 SVM 如何避免这个问题（使用一种称为核技术的方法）。



(a) 原二维空间中的决策边界　　(b) 变换后空间中的决策边界

图 5-28　分类具有非线性决策边界的数据

2. 学习非线性 SVM 模型

尽管属性变换方法看上去大有可为，但是存在一些实现问题。首先，它不清楚应当使用什么类型的映射函数，才可以确保在变换后的空间构建线性决策边界。一种选择是把数据变换到无限维空间中，但是这样的高维空间可能很难处理。其次，即使知道合适的映射函数，在高维特征空间中解约束优化问题仍然是计算代价很高的任务。

为了解释这些问题并考察处理它们的方法，假定存在一个合适的函数$\Phi(\mathbf{x})$来变换给定的数据集。变换后，我们需要构建一个线性的决策边界，把样本划分到它们各自所属的类中。在变换后的空间中，线性决策边界具有以下形式：$\mathbf{w}\cdot\Phi(\mathbf{x})+b=0$。

定义 5.2　非线性 SVM　非线性 SVM 的学习任务可以形式化地表达为以下的优化问题：

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{受限于}\quad y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b)\geqslant 1,\quad i=1,2,\cdots,N$$

注意，非线性 SVM 的学习任务和线性 SVM（参见定义 5.1）很相似。主要的区别在于，学习任务是在变换后的属性$\Phi(\mathbf{x})$，而不是在原属性 $\mathbf{x}$ 上执行的。采用 5.5.2 节和 5.5.3 节介绍的线性 SVM 所使用的方法，可以得到该受约束的优化问题的对偶拉格朗日函数：

$$L_D = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j) \tag{5-56}$$

使用二次规划技术得到λ_i后，就可以通过下面的方程导出参数 $\mathbf{w}$ 和 b：

$$\mathbf{w} = \sum_i \lambda_i y_i \Phi(\mathbf{x}_i) \tag{5-57}$$

$$\lambda_i\{y_i\sum_j \lambda_j y_j \Phi(\mathbf{x}_j)\cdot\Phi(\mathbf{x}_i)+b)-1\}=0 \tag{5-58}$$

这类似于公式（5-39）和公式（5-40）的线性 SVM。最后，可以通过下式对检验实例 $\mathbf{z}$ 进行分类：

$$f(\mathbf{z})=sign(\mathbf{w}\cdot\Phi(\mathbf{z})+b)=sign\left(\sum_{i=1}^{n}\lambda_j y_j \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{z})+b\right) \tag{5-59}$$

注意，除了公式（5-57）外，其余的计算公式（5-58）和公式（5-59）都涉及计算变换后的空间中向量对之间的点积$\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$（即相似度）。这种运算是相当麻烦的，可能导致维灾难问题。这个问题的一种突破性解决方案是一种称为**核技术**（kernel trick）的方法。

3. 核技术

点积经常用来度量两个输入向量间的相似度。例如，在 2.4.5 节介绍的余弦相似度可以定义为规范化后具有单位长度的两个向量间的点积。类似地，点积$\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$可以看作两个实例 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 在变换后的空间中的相似性度量。

核技术是一种使用原属性集计算变换后的空间中的相似度的方法。考虑公式（5-55）中的映射函数Φ。两个输入向量 $\mathbf{u}$ 和 $\mathbf{v}$ 在变换后的空间中的点积可以写成如下形式：

$$\begin{aligned}\Phi(\mathbf{u})\cdot\Phi(\mathbf{v}) &= (u_1^2,u_2^2,\sqrt{2}u_1,\sqrt{2}u_2,\sqrt{2}u_1u_2,1)\cdot(v_1^2,v_2^2,\sqrt{2}v_1,\sqrt{2}v_2,\sqrt{2}v_1v_2,1)\\ &= u_1^2v_1^2+u_2^2v_2^2+2u_1v_1+2u_2v_2+2u_1u_2v_1v_2+1\\ &= (\mathbf{u}\cdot\mathbf{v}+1)^2\end{aligned} \tag{5-60}$$

该分析表明，变换后的空间中的点积可以用原空间中的相似度函数表示：

$$K(\mathbf{u},\mathbf{v})=\Phi(\mathbf{u})\cdot\Phi(\mathbf{v})=(\mathbf{u}\cdot\mathbf{v}+1)^2 \tag{5-61}$$

这个在原属性空间中计算的相似度函数 K 称为**核函数**（kernel function）。核技术有助于处理如何实现非线性 SVM 的一些问题。首先，由于在非线性 SVM 中使用的核函数必须满足一个称为 **Mercer 定理**的数学原理，因此我们不需要知道映射函数Φ的确切形式。Mercer 原理确保核函数总可以用某高维空间中两个输入向量的点积表示。SVM 核的变换后空间也称为**再生核希尔伯特空间**（Reproducing Kernel Hilbert Space，RKHS）。其次，相对于使用变换后的属性集$\Phi(\mathbf{x})$，使用核函数计算点积的开销更小。第三，既然计算在原空间中进行，维灾难问题就可以避免。

图5-29显示了一个非线性决策边界，它是通过使用公式（5-61）给出的多项式核函数的SVM获得的。检验实例**z**可以通过下式分类：

$$\begin{aligned} f(\mathbf{z}) &= sign\left(\sum_{i=1}^{n}\lambda_i y_i \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{z})+b\right) \\ &= sign\left(\sum_{i=1}^{n}\lambda_i y_i K(\mathbf{x}_i,\mathbf{z})+b\right) \\ &= sign\left(\sum_{i=1}^{n}\lambda_i y_i (\mathbf{x}_i\cdot\mathbf{z}+1)^2+b\right) \end{aligned} \tag{5-62}$$

其中 b 是使用公式（5-58）得到的参数。非线性 SVM 得到的决策边界与图 5-28a 中显示的真实决策边界非常相似。

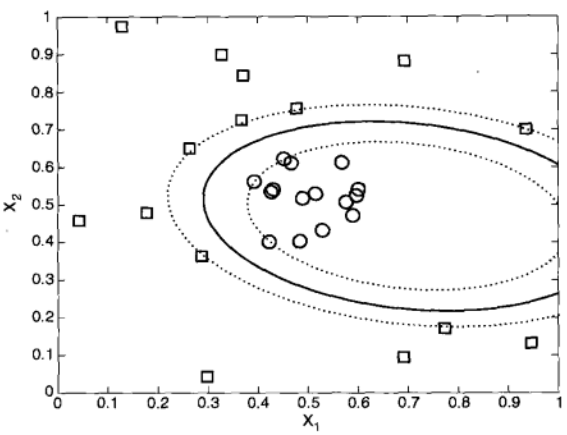


图 5-29 具有多项式核的非线性 SVM 产生的决策边界

4. Mercer 定理

对非线性 SVM 使用的核函数主要的要求是，必须存在一个相应的变换，使得计算一对向量的核函数等价于在变换后的空间中计算这对向量的点积。这个要求可以用 Mercer 定理形式化地陈述。

定理 5.1 Mercer 定理 *核函数 K 可以表示为：*

$$K(\boldsymbol{u},\boldsymbol{v}) = \Phi(\boldsymbol{u})\cdot\Phi(\boldsymbol{v})$$

当且仅当对于任意满足 $\int g(\boldsymbol{x})^2 d\boldsymbol{x}$ 为有限值的函数 $g(\boldsymbol{x})$，则

$$\int K(\boldsymbol{x},\boldsymbol{y})\, g(\boldsymbol{x})\, g(\boldsymbol{y})\, d\boldsymbol{x}\, d\boldsymbol{y} \geqslant 0$$

满足定理 5.1 的核函数称为正定（positive definite）核函数。下面给出一些这种函数的例子：

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p \tag{5-63}$$

$$K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)} \tag{5-64}$$

$$K(\mathbf{x},\mathbf{y}) = \tanh(k\mathbf{x}\cdot\mathbf{y}-\delta) \tag{5-65}$$

例5.6　考虑公式（5-63）给出的多项式核函数。令 $g(\boldsymbol{x})$是一个具有有限 L_2 范数的函数，即 $\int g(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} < \infty$。

$$
\begin{aligned}
&\int (\mathbf{x}\cdot\mathbf{y}+1)^p g(\mathbf{x})g(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\
&= \int \sum_{i=0}^{p} \binom{p}{i} (\mathbf{x}\cdot\mathbf{y})^i g(\mathbf{x})g(\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\
&= \sum_{i=0}^{p} \binom{p}{i} \int \sum_{\alpha_1,\alpha_2,\cdots} \binom{i}{\alpha_1\alpha_2\cdots} \left[(x_1y_1)^{\alpha_1}(x_2y_2)^{\alpha_2}(x_3y_3)^{\alpha_3}\cdots\right] g(x_1,x_2,\cdots)g(y_1,y_2,\cdots)\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}y_1\mathrm{d}y_2\cdots \\
&= \sum_{i=0}^{p} \sum_{\alpha_1,\alpha_2,\cdots} \binom{p}{i} \binom{i}{\alpha_1\alpha_2\cdots} \left[\int x_1^{\alpha_1}x_2^{\alpha_2}\cdots g(x_1,x_2,\cdots)\mathrm{d}x_1\mathrm{d}x_2\cdots\right]^2
\end{aligned}
$$

由于积分结果非负，因此多项式核函数满足 Mercer 定理。 □

5.5.5　支持向量机的特征

SVM 具有许多很好的性质，因此它已经成为广泛使用的分类算法之一。下面简要总结一下 SVM 的一般特征。

(1) SVM 学习问题可以表示为凸优化问题，因此可以利用已知的有效算法发现目标函数的全局最小值。而其他的分类方法（如基于规则的分类器和人工神经网络）都采用一种基于贪心学习的策略来搜索假设空间，这种方法一般只能获得局部最优解。

(2) SVM 通过最大化决策边界的边缘来控制模型的能力。尽管如此，用户必须提供其他参数，如使用的核函数类型、为了引入松弛变量所需的代价函数 C 等。

(3) 通过对数据中每个分类属性值引入一个哑变量，SVM 可以应用于分类数据。例如，如果婚姻状况有三个值{单身，已婚，离异}，可以对每一个属性值引入一个二元变量。

(4) 本节所给出的 SVM 公式表述是针对二类问题的。5.8 节将给出把 SVM 扩展到多类问题的一些方法。

5.6　组合方法

除最近邻方法外，本章迄今为止已经介绍的分类技术都是使用从训练数据得到的单个分类器来预测未知样本的类标号。本节将介绍一些技术，通过聚集多个分类器的预测来提高分类准确率。这些技术称为**组合**（ensemble）或**分类器组合**（classifier combination）方法。组合方法由训练数据构建一组**基分类器**（base classifier），然后通过对每个基分类器的预测进行投票来进行分类。本节将解释为什么组合方法比任意单分类器的效果好，并提供构建组合分类器的技术。

5.6.1　组合方法的基本原理

下面的例子说明了组合方法为什么能够改善分类器的性能。

例5.7　考虑25个二元分类器的组合，其中每一个分类器的误差 ε 均为0.35。组合分类器通过对这些基分类器的预测进行多数表决的方法来预测检验样本的类标号。如果所有基分类器都是等同的，则组合分类器也将对基分类器预测错误的样本误分类。因此，组合分类器的误差率仍然是0.35。另一方面，如果基分类器是相互独立的（即它们的误差是不相关的），则仅当超过一

半的基分类器都预测错误时，组合分类器才会作出错误的预测。在这种情况下，组合分类器的误差率为：

$$e_{\text{ensemble}} = \sum_{i=13}^{25} C_{25}^{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06 \tag{5-66}$$

远低于基分类器的误差率。 □

图 5-30 显示对于不同的基分类器误差率（ε），25 个二元分类器的组合分类器误差率（e_{ensemble}）。对角线表示所有基分类器都是等同的情况，而实线则表示所有基分类器独立时的情况。注意，当 $\varepsilon > 0.5$ 时，组合分类器的性能比不上基分类器。

前面的例子说明，组合分类器的性能优于单个分类器必须满足两个必要的条件：(1)基分类器之间应该是相互独立的；(2)基分类器应当好于随机猜测分类器。实践上，很难保证基分类器之间完全独立。尽管如此，我们看到在基分类器轻微相关的情况下，组合方法可以提高分类的准确率。

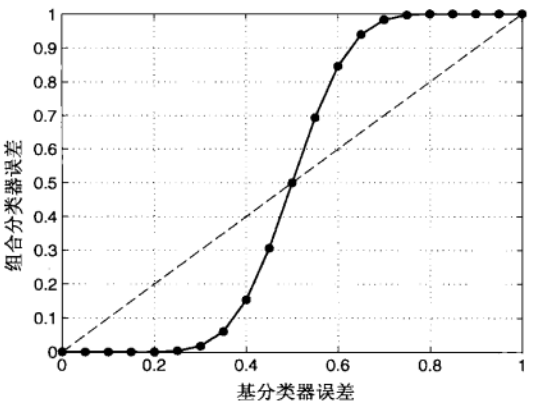


图 5-30 基分类器和组合分类器误差的比较

5.6.2 构建组合分类器的方法

图 5-31 给出了组合方法的逻辑视图。其基本的思想是，在原始数据上构建多个分类器，然后在分类未知样本时聚集它们的预测结果。下面构建组合分类器的几种方法。

(1) **通过处理训练数据集**。这种方法根据某种抽样分布，通过对原始数据进行再抽样来得到多个训练集。抽样分布决定一个样本选作训练的可能性大小，并且可能因试验而异。然后，使用特定的学习算法为每个训练集建立一个分类器。**装袋**（bagging）和**提升**（boosting）是两种处理训练数据集的组合方法。这些方法将在 5.6.4 节和 5.6.5 节更详细地介绍。

(2) **通过处理输入特征**。在这种方法中，通过选择输入特征的子集来形成每个训练集。子集可以随机选择，也可以根据领域专家的建议选择。一些研究表明，对那些含有大量冗余特征的数据集，这种方法的性能非常好。**随机森林**（Random forest）就是一种处理输入特征的组合方法，它使用决策树作为基分类器。随机森林将在 5.6.6 节介绍。

(3) **通过处理类标号**。这种方法适用于类数足够多的情况。通过将类标号随机划分成两个不相交的子集 A_0 和 A_1，把训练数据变换为二类问题。类标号属于子集 A_0 的训练样本指派到类 0，而那些类标号属于子集 A_1 的训练样本被指派到类 1。然后，使用重新标记过的数据来训练一个基

分类器。重复重新标记类和构建模型步骤多次，就得到一组基分类器。当遇到一个检验样本时，使用每个基分类器 C_i 预测它的类标号。如果检验样本被预测为类 0，则所有属于 A_0 的类都得到一票。相反，如果它被预测为类 1，则所有属于 A_1 的类都得到一票。最后统计选票，将检验样本指派到得票最高的类。后面介绍的**错误-纠正输出编码**（error-correcting output coding）方法就是这种方法的一个例子。

(4) **通过处理学习算法**。许多学习算法都可以这样来处理：在同一个训练数据集上多次执行算法可能得到不同的模型。例如，通过改变一个人工神经网络的拓扑结构或各个神经元之间联系的初始权值，就可以得到不同的模型。同样，通过在树生成过程中注入随机性，可以得到决策树的组合分类器。例如，在每一个结点上，可以随机地从最好的 k 个属性中选择一个属性，而不是选择该结点上最好的属性来进行划分。

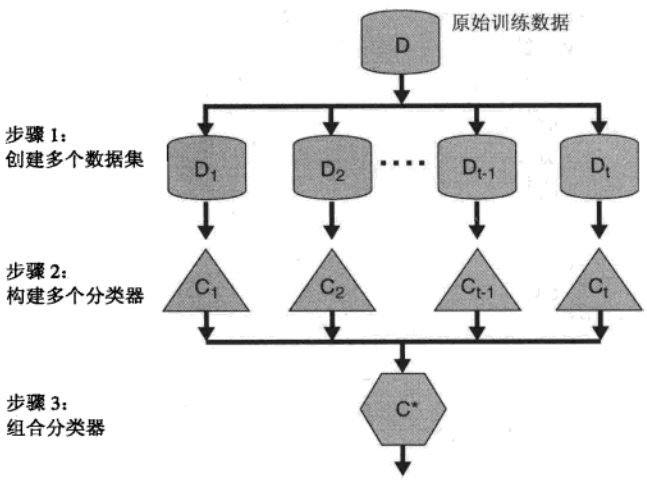


图 5-31　组合学习方法的逻辑视图

前三种属于一般性方法，适用于任何分类器，而第四种方法依赖于使用的分类器类型。对于大部分方法，基分类器可以顺序产生（一个接一个）或并行产生（一次性产生）。算法 5.5 显示了以顺序方式构建组合分类器的步骤。第一步是从原始数据集 D 中创建一个训练数据集。训练数据集可以与 D 相同或是 D 的轻微修改，这取决于所使用的组合方法的类型。训练集的大小一般和原始数据集保持一致，但样本的分布可能不同，即某些样本可能在训练集中多次出现，而有些样本可能一次也不出现。然后，为每个训练集 D_i 构建一个基分类器 C_i。组合方法对于**不稳定的分类器**（unstable classifier）效果较好。不稳定的分类器是对训练集微小的变化都很敏感的基分类器。不稳定的分类器的例子包括决策树、基于规则的分类器和人工神经网络。正如将在 5.6.3 节中讨论的那样，训练样本的可变性是分类器误差的主要来源之一。通过聚集在不同的训练集上构建的基分类器，有助于减少这种类型的误差。

最后，通过组合基分类器 $C_i(\mathbf{x})$的预测来对检验样本 $\mathbf{x}$ 进行分类：

$$C^*(\mathbf{x}) = Vote(C_1(\mathbf{x}), C_2(\mathbf{x}), \cdots, C_k(\mathbf{x}))$$

可以对单个预测值进行多数表决，或用基分类器的准确率对每个预测值进行加权来得到类标号。

算法 5.5 组合方法的一般过程

1：令 D 表示原始训练数据集，k 表示基分类器的个数，T 表示检验数据集
2：**for** $i = 1$ to k **do**
3：　由 D 创建训练集 D_i
4：　由 D_i 构建基分类器 C_i
5：**end for**
6：**for** 每一个检验记录 $x \in T$ **do**
7：　$C^*(x) = Vote(C_1(\mathbf{x}), C_2(\mathbf{x}), \cdots, C_k(\mathbf{x}))$
8：**end for**

5.6.3 偏倚-方差分解

偏倚-方差分解是分析预测模型的预测误差的形式化方法。下面的例子给出了这种方法的直观解释。

图 5-32 显示了以特定角度发射的射弹的弹道轨迹。假设射弹在某一位置 x 击中地面，距离目标位置 t 的距离为 d。依赖于发射力，每次试验观察到的距离都不一样。观察到的距离可以分解为几个部分。第一部分称为**偏倚**（bias），度量目标位置与射弹击中地面的位置之间的平均距离。偏倚量依赖于射弹的发射角度。第二部分称为**方差**（variance），度量 x 和射弹击中地面的平均位置 $\bar{x}$ 之间的偏差。方差可以解释为施加于射弹上的发射力改变的结果。最后，如果目标是不固定的，则观察到距离也受目标位置变化的影响。这要考虑与目标位置的可变性相关的**噪声**（noise）部分。将这些成分放到一块，平均距离可以表示为：

$$d_{f,\theta}(y, t) = \text{Bias}_\theta + \text{Variance}_f + \text{Noise}_t \tag{5-67}$$

其中，f 是发射力，θ是发射的角度。

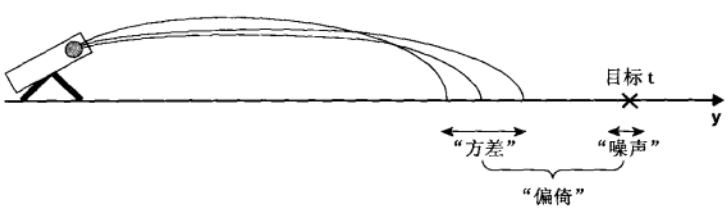


图 5-32 偏倚-方差分解

可以使用同样的方法来分析预测给定样本类标号的任务。对一个给定的分类器，一些预测可能是正确的，而另一些可能完全不沾边。我们可以将一个分类器的期望误差分解为公式（5-67）中的三项和，其中期望误差是分类器误分一个给定样本的概率。本节的剩余部分将介绍分类的偏倚、方差和噪声的含义。

通常，训练分类器，以最小化训练误差。然而，分类器的用途在于，必须能够对它从没遇到过的样本的类标号作出预测。这要求分类器将它的决策边界泛化到没有训练样本可用的区域——一种依赖于分类器的设计选择的决策。例如，决策树归纳的一个关键设计问题是得到具有最低期望误差的树所需的剪枝量。图 5-33 显示了两棵决策树 T_1 和 T_2，它们从同一训练集上得到，但具有不同的复杂度。决策树 T_2 是通过对决策树 T_1 进行剪枝，直到最大深度为 2 得到的；另一方面，

T_1 却只在它的决策树上做了很少的剪枝。这些设计选择将导致分类器的偏倚，类似于前面例子中射弹发射的偏倚。一般来说，分类器关于它的决策边界性质所做的假定越强，分类器的偏倚就越大。因此，T_2 具有更大的偏倚，因为与 T_1 相比，它对决策边界的假定更强（反映在树的大小上）。其他可能导致分类器偏倚的设计选择包括人工神经网络的拓扑结构和最近邻分类器中考虑的近邻的个数。

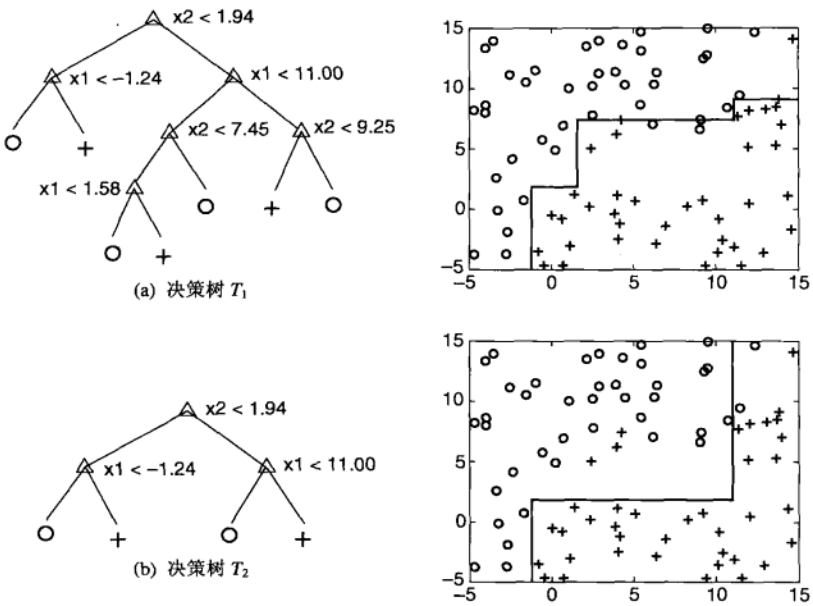
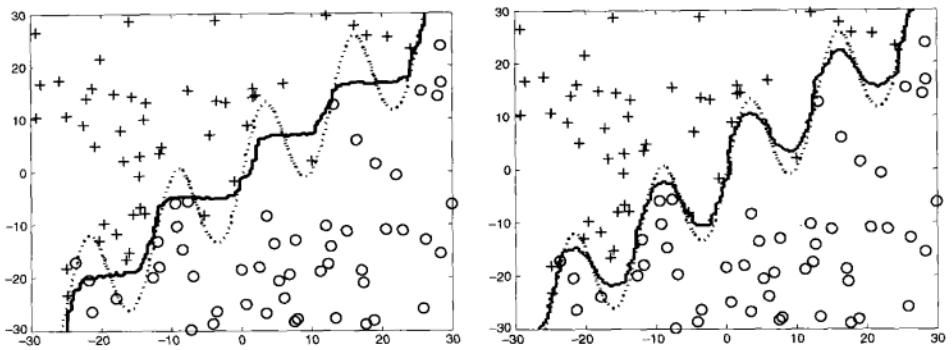


图 5-33　从相同训练数据上得到的复杂度不同的两棵决策树

分类器的期望误差也受训练数据可变性的影响，因为训练集合的不同的成分可能导致不同的决策边界。这类似于施加于射弹上的发射力不同时 x 的方差。期望误差的最后一个成分与目标类的固有噪声相关。对某些领域来说，目标类可能是不确定的，即具有相同属性值的实例可能有不同的类标号。即使知道实际的决策边界，这样的误差也是不可避免的。

期望误差中的偏倚和方差取决于使用的分类器的类型。图 5-34 比较了决策树产生的决策边界和 1-最近邻分类器产生的决策边界。对于每种分类器，绘制从 100 个训练集归纳的“平均”模型的决策边界，其中每个训练集包含 100 个样本。同时用虚线画出从中产生这些数据的实际决策边界。实际决策边界和“平均”决策边界之间的差反映了分类器的偏倚。模型平均后，观察到实际决策边界和 1-最近邻分类器的决策边界之间的差别要小于与决策树分类器的差别。这一结果表明 1-最近邻分类器的偏倚要低于决策树分类器的偏倚。

另一方面，1-最近邻分类器对训练样本的组成更加敏感。如果考察从不同训练集上归纳得到的模型，1-最近邻分类器的决策边界的可变性比决策树分类器大。因此，相对于 1-最近邻分类器，决策树分类器的决策边界具有较低的方差。



(a) 决策树的决策边界 (b) 1-最近邻的决策边界

图 5-34 决策树和 1-最近邻分类器的偏倚

5.6.4 装袋

装袋（bagging）又称自助聚集（boot strap aggregating），是一种根据均匀概率分布从数据集中重复抽样（有放回的）的技术。每个自助样本集都和原数据集一样大。由于抽样过程是有放回的，因此一些样本可能在同一个训练数据集中出现多次，而其他一些却可能被忽略。一般来说，自助样本 D_i 大约包含 63%的原训练数据，因为每一个样本抽到 D_i 的概率为 $1-(1-1/N)^N$，如果 N 足够大，这个概率将收敛于 $1-1/e \approx 0.632$。装袋的基本过程概括在算法 5.6 中。训练过 k 个分类器后，测试样本被指派到得票最高的类。

算法 5.6 装袋算法

1：设 k 为自助样本集的数目
2：**for** $i=1$ to k **do**
3：　生成一个大小为 N 的自助样本集 D_i
4：　在自助样本集 D_i 上训练一个基分类器 C_i
5：**end for**
6：$C^*(x)=\underset{y}{\operatorname{argmax}}\sum_i \delta(C_i(x)=y)$

{如果参数为真则$\delta(\cdot)=1$，否则$\delta(\cdot)=0$}

为了说明装袋如何进行，考虑表 5-4 给出的数据集。设 x 表示一维属性，y 表示类标号。假设使用这样一个分类器，它是仅包含一层的二叉决策树，具有一个测试条件 $x \leqslant k$，其中 k 是使得叶结点熵最小的分裂点。这样的树也称为**决策树桩**（decision stump）。

表 5-4 用于构建装袋组合分类器的数据集例子

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	−1	−1	−1	−1	1	1	1

不进行装袋，能产生的最好的决策树桩的分裂点为 $x \leqslant 0.35$ 或 $x \leqslant 0.75$。无论选择哪一个，树的准确率最多为 70%。假设我们在数据集上应用 10 个自助样本集的装袋过程，图 5-35 给出了每轮装袋选择的训练样本。在每个表的右边，给出了分类器产生的决策边界。

装袋第 1 轮

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9	$x \leqslant 0.35 \Rightarrow y = 1$
y	1	1	1	1	−1	−1	−1	−1	1	1	$x > 0.35 \Rightarrow y = -1$

装袋第 2 轮

x	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1	1	1	$x \leqslant 0.65 \Rightarrow y = 1$
y	1	1	1	−1	−1	1	1	1	1	1	$x > 0.65 \Rightarrow y = 1$

装袋第 3 轮

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9	$x \leqslant 0.35 \Rightarrow y = 1$
y	1	1	1	−1	−1	−1	−1	−1	1	1	$x > 0.35 \Rightarrow y = -1$

装袋第 4 轮

x	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9	$x \leqslant 0.3 \Rightarrow y = 1$
y	1	1	1	−1	−1	−1	−1	−1	1	1	$x > 0.3 \Rightarrow y = -1$

装袋第 5 轮

x	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1	$x \leqslant 0.35 \Rightarrow y = 1$
y	1	1	1	−1	−1	−1	−1	1	1	1	$x > 0.35 \Rightarrow y = -1$

装袋第 6 轮

x	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1	$x \leqslant 0.75 \Rightarrow y = -1$
y	1	−1	−1	−1	−1	−1	−1	1	1	1	$x > 0.75 \Rightarrow y = 1$

装袋第 7 轮

x	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1	$x \leqslant 0.75 \Rightarrow y = -1$
y	1	−1	−1	−1	−1	1	1	1	1	1	$x > 0.75 \Rightarrow y = 1$

装袋第 8 轮

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1	$x \leqslant 0.75 \Rightarrow y = -1$
y	1	1	−1	−1	−1	−1	−1	1	1	1	$x > 0.75 \Rightarrow y = 1$

装袋第 9 轮

x	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1	$x \leqslant 0.75 \Rightarrow y = -1$
y	1	1	−1	−1	−1	−1	−1	1	1	1	$x > 0.75 \Rightarrow y = 1$

装袋第 10 轮

x	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9	$x \leqslant 0.05 \Rightarrow y = -1$
y	1	1	1	1	1	1	1	1	1	1	$x > 0.05 \Rightarrow y = 1$

图 5-35　装袋的例子

通过对每个基分类器所作的预测使用多数表决来分类表 5-4 给出的整个数据集。图 5-36 给出了预测结果。由于类标号是−1 或+1，因此应用多数表决等价于对 y 的预测值求和，然后考察结果的符号（参看图 5-36 中的第二行到最后一行）。注意，组合分类器完全正确地分类了原始数据集中的 10 个样本。

轮	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	1	1	1	−1	−1	−1	−1	−1	−1	−1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	−1	−1	−1	−1	−1	−1	−1
4	1	1	1	−1	−1	−1	−1	−1	−1	−1
5	1	1	1	−1	−1	−1	−1	−1	−1	−1
6	−1	−1	−1	−1	−1	−1	−1	1	1	1
7	−1	−1	−1	−1	−1	−1	−1	1	1	1
8	−1	−1	−1	−1	−1	−1	−1	1	1	1
9	−1	−1	−1	−1	−1	−1	−1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
和	2	2	2	−6	−6	−6	−6	2	2	2
符号	1	1	1	−1	−1	−1	−1	1	1	1
实际类	1	1	1	−1	−1	−1	−1	1	1	1

图 5-36　使用装袋方法构建组合分类器的例子

前面的例子也说明了使用组合方法的又一个优点：增强了目标函数的表达功能。即使每个基分类器都是一个决策树桩，组合的分类器也能表示一棵深度为 2 的决策树。

装袋通过降低基分类器方差改善了泛化误差。装袋的性能依赖于基分类器的稳定性。如果基分类器是不稳定的，装袋有助于减低训练数据的随机波动导致的误差；如果基分类器是稳定的，即对训练数据集中的微小变化是鲁棒的，则组合分类器的误差主要是由基分类器的偏倚所引起的。在这种情况下，装袋可能不会对基分类器的性能有显著改善，装袋甚至可能降低分类器的性能，因为每个训练集的有效容量比原数据集大约小 37%。

最后，由于每一个样本被选中的概率都相同，因此装袋并不侧重于训练数据集中的任何特定实例。因此，用于噪声数据，装袋不太受过分拟合的影响。

5.6.5 提升

提升是一个迭代的过程，用来自适应地改变训练样本的分布，使得基分类器聚焦在那些很难分的样本上。不像装袋，提升给每一个训练样本赋一个权值，而且可以在每一轮提升过程结束时自动地调整权值。训练样本的权值可以用于以下方面。

(1) 可以用作抽样分布，从原始数据集中提取出自助样本集。

(2) 基分类器可以使用权值学习有利于高权值样本的模型。

本节描述一个算法，它利用样本的权值来确定其训练集的抽样分布。开始时，所有样本都赋予相同的权值 $1/N$，从而使得它们被选作训练的可能性都一样。根据训练样本的抽样分布来抽取样本，得到新的样本集。然后，由该训练集归纳一个分类器，并用它对原数据集中的所有样本进行分类。每一轮提升结束时更新训练样本的权值。增加被错误分类的样本的权值，而减小被正确分类的样本的权值。这迫使分类器在随后迭代中关注那些很难分类的样本。

下表给出了每轮提升选择的样本。

提升（第一轮）	7	3	2	8	7	9	4	10	6	3
提升（第二轮）	5	4	9	4	2	5	1	7	4	2
提升（第三轮）	4	4	8	10	4	5	4	6	3	4

开始，所有的样本都赋予相同的权值。然而，由于抽样是有放回的，因此某些样本可能被选中多次，如样本 3 和 7。然后，使用由这些数据建立的分类器对所有样本进行分类。假定样本 4 很难分类，随着它被重复地误分类，该样本的权值在后面的迭代中将会增加。同时，前一轮没有被选中的样本（如样本 1 和 5）也有更好的机会在下一轮被选中，因为前一轮对它们的预测多半是错误的。随着提升过程进行，最难分类的那些样本将有更大的机会被选中。通过聚集每个提升轮得到的基分类器，就得到最终的组合分类器。

在过去的几年里，已经开发了几个提升算法的实现。这些算法的差别在于：(1)每轮提升结束时如何更新训练样本的权值；(2)如何组合每个分类器的预测。下面，主要考察称为 AdaBoost 的实现。

AdaBoost

令$\{(\mathbf{x}_j, y_j) \mid j = 1, 2, \cdots, N\}$表示包含 N 个训练样本的集合。在 AdaBoost 算法中，基分类器 C_i 的重要性依赖于它的错误率。错误率ε_i定义为：

$$\varepsilon_i = \frac{1}{N}\left[\sum_{j=1}^{N} w_j I\left(C_i(\mathbf{x}_j) \neq y_j\right)\right] \tag{5-68}$$

其中，如果谓词 p 为真，则 $I(p)=1$，否则为 0。基分类器 C_i 的重要性由如下参数给出：

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$

注意，如果错误率接近 0，则 α_i 具有一个很大的正值，而当错误率接近 1 时，α_i 有一个很大的负值，如图 5-37 所示。

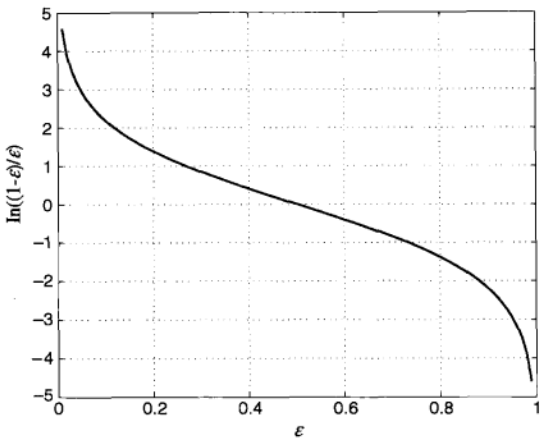


图 5-37　作为训练误差 ε 的函数绘制 α 的曲线

参数 α_i 也被用来更新训练样本的权值。为了说明这一点，假定 $w_i^{(j)}$ 表示在第 j 轮提升迭代中赋给样本 $(\mathbf{x}_i, y_i)$ 的权值。AdaBoost 的权值更新机制由下式给出：

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} e^{-\alpha_j} & \text{如果} C_j(\mathbf{x}_i) = y_i \\ e^{\alpha_j} & \text{如果} C_j(\mathbf{x}_i) \neq y_i \end{cases} \tag{5-69}$$

其中，Z_j 是一个正规因子，用来确保 $\sum_i w_i^{(j+1)} = 1$。公式（5-69）给出的权值更新公式增加那些被错误分类样本的权值，并减少那些已经被正确分类的样本的权值。

AdaBoost 算法将每一个分类器 C_j 的预测值根据 α_j 进行加权，而不是使用多数表决的方案。这种机制有助于 AdaBoost 惩罚那些准确率很差的模型，如那些在较早的提升轮产生的模型。另外，如果任何中间轮产生高于 50%的误差，则权值将被恢复为开始的一致值 $w_i = 1/N$，并重新进行抽样。算法 5.7 给出了 AdaBoost 算法的描述。

现在看看提升方法在表 5-4 给出的数据集上是怎么工作的。最初，所有的样本具有相等的权值。三轮提升后，选作训练的样本如图 5-38a 所示。在每轮提升结束时使用公式（5-69）来更新每一个样本的权值。

不使用提升，决策树桩的准确率至多达到 70%。使用 AdaBoost，预测结果在图 5-39b 给出。组合分类器的最终预测结果通过取每个基分类器预测的加权平均得到，显示在图 5-39b 的最后一行。注意，AdaBoost 完全正确地分类了训练数据集中的所有样本。

算法 5.7 AdaBoost 算法

1：$\mathbf{w}=\{w_j=1/N \mid j=1,2,\cdots,N\}$。{初始化 N 个样本的权值。}
2：令 k 表示提升的轮数。
3：**for** $i=1$ to k **do**
4：　根据 $\mathbf{w}$，通过对 D 进行抽样（有放回）产生训练集 D_i。
5：　在 D_i 上训练基分类器 C_i。
6：　用 C_i 对原训练集 D 中的所有样本分类。
7：　$\varepsilon_i=\frac{1}{N}\left[\sum_j w_j\delta(C_i(x_j)\neq y_j)\right]$，{计算加权误差。}
8：　**if** $\varepsilon_i>0.5$ **then**
9：　　$\mathbf{w}=\{w_j=1/N \mid j=1,2,\cdots,N\}$。{重新设置 N 个样本的权值。}
10：　　返回步骤 4。
11：　**end if**
12：　$\alpha_i=\frac{1}{2}\ln\frac{1-\varepsilon_i}{\varepsilon_i}$
13：　根据公式（5-69）更新每个样本的权值。
14：**end for**
15：$C^*(\mathbf{x})=\underset{y}{\operatorname{argmax}}\sum_{j=1}^{T}\alpha_j\delta(C_j(\mathbf{x})=y)$

第 1 轮提升

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	−1	−1	−1	−1	−1	−1	−1	1	1

第 2 轮提升

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

第 3 轮提升

x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	−1	−1	−1	−1	−1	−1	−1	−1

(a) 提升选择的训练记录

轮	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.311	0.311	0.311	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.029	0.029	0.029	0.228	0.228	0.228	0.228	0.009	0.009	0.009

(b) 训练记录的权值

图 5-38 提升的例子

对提升的一个重要分析结果显示，组合分类器的训练误差受下式的限制：

$$e_{\text{ensemble}}\leqslant\prod_i\left[\sqrt{\varepsilon_i(1-\varepsilon_i)}\right] \tag{5-70}$$

其中 ε_i 是基分类器 i 的错误率。如果基分类器的错误率低于 50%，则 $\varepsilon_i=0.5-\gamma_i$，其中 γ_i 度量了分类器比随机猜测强多少。则组合分类器的训练误差的边界变为：

$$e_{\text{ensemble}}\leqslant\prod_i\sqrt{1-4\gamma_i^2}\leqslant\exp\left(-2\sum_i\gamma_i^2\right) \tag{5-71}$$

如果对所有的 i 都有 $\gamma_i<\gamma^*$，则组合分类器的训练误差呈指数递减，从而导致算法快速收敛。尽管如此，由于它倾向于那些被错误分类的样本，提升技术很容易受过分拟合的影响。

轮	划分点	左类	右类	α
1	0.75	−1	1	1.738
2	0.05	1	1	2.7784
3	0.3	1	−1	4.1195

(a)

轮	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	−1	−1	−1	−1	−1	−1	−1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	−1	−1	−1	−1	−1	−1	−1
和	5.16	5.16	5.16	−3.08	−3.08	−3.08	−3.08	0.397	0.397	0.397
符号	1	1	1	−1	−1	−1	−1	1	1	1

(b)

图 5-39　使用 AdaBoost 方法构建的组合分类器的例子

5.6.6　随机森林

随机森林（random forest）是一类专门为决策树分类器设计的组合方法。它组合多棵决策树作出的预测，其中每棵树都是基于随机向量的一个独立集合的值产生的，如图 5-40 所示。与 AdaBoost 使用的自适应方法不同，AdaBoost 中概率分布是变化的，以关注难分类的样本，而随机森林则采用一个固定的概率分布来产生随机向量。使用决策树装袋是随机森林的特例，通过随机地从原训练集中有放回地选取 N 个样本，将随机性加入到构建模型的过程中。整个模型构建过程中，装袋也使用同样的均匀概率分布来产生它的自助样本。

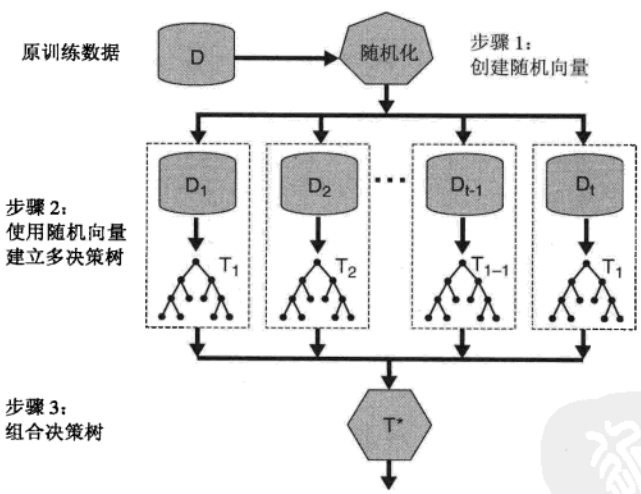


图 5-40　随机森林

已经从理论上证明，当树的数目足够大时，随机森林的泛化误差的上界收敛于下面的表达式：

$$\text{泛化误差} \leqslant \frac{\bar{\rho}(1-s^2)}{s^2} \tag{5-72}$$

其中 $\bar{\rho}$ 是树之间的平均相关系数，s 是度量树型分类器的“强度”的量。一组分类器的强度是指分类器的平均性能，而性能以分类器的余量（M）用概率算法度量：

$$M(\mathbf{X}, Y) = P(\hat{Y}_\theta = Y) - \max_{Z \neq Y} P(\hat{Y}_\theta = Z) \tag{5-73}$$

其中 $\hat{Y}_\theta$ 是根据某随机向量 θ 构建的分类器对 $\mathbf{X}$ 作出的预测类。余量越大，分类器正确预测给定的样本 $\mathbf{X}$ 的可能性就越大。公式（5-72）是相当直观的，随着树的相关性增加或组合分类器的强度降低，泛化误差的上界趋向于增加。随机化有助于减少决策树之间的相关性，从而改善组合分类器的泛化误差。

每棵决策树都使用一个从某固定概率分布产生的随机向量。可以使用多种方法将随机向量合并到树的增长过程中。第一种方法是随机选择 F 个输入特征来对决策树的结点进行分裂。这样，分裂结点的决策是根据这 F 个选定的特征，而不是考察所有可用的特征来决定。然后，让树完全增长而不进行任何修剪，这可能有助于减少结果树的偏倚。树构建完毕之后，就可以使用多数表决的方法来组合预测。这种方法称为 Forest-RI，其中 RI 指随机输入选择。为了增加随机性，可以使用装袋来为 Forest-RI 产生自助样本。随机森林的强度和相关性都取决于 F 的大小。如果 F 足够小，树的相关性趋向于减弱；另一方面，树分类器的强度趋向于随着输入特征数 F 的增加而提高。作为折中，通常选取特征的数目为 $F = \log_2 d + 1$，其中 d 是输入特征数。由于在每一个结点仅仅需要考察特征的一个子集，这种方法将显著减少算法的运行时间。

如果原始特征 d 的数目太小，则很难选择一个独立的随机特征集合来建立决策树。一种加大特征空间的办法是创建输入特征的线性组合。具体地说，在每一个结点，新特征通过随机选择 L 个输入特征来构建。这些输入特征用区间[−1, 1]上的均匀分布产生的系数进行线性组合。在每个结点，产生 F 个这种随机组合的新特征，并且从中选择最好的来分裂结点。这种方法称为 Forest-RC。

生成随机树的第三种方法是：在决策树的每一个结点，从 F 个最佳划分中随机选择一个。除非 F 足够大，否则这种方法可能产生比 Forest-RI 和 Forest-RC 相关性更强的树。这种方法也没有 Forest-RI 和 Forest-RC 节省运行时间，因为算法需要在决策树的每个结点考察所有的分裂特征。

实验表明，随机森林的分类准确率可以与 AdaBoost 相媲美。它对噪声更加鲁棒，运行速度比 AdaBoost 快得多。下一节将比较各种组合算法的分类准确率。

5.6.7 组合方法的实验比较

表 5-5 显示了将决策树分类器与装袋、提升和随机森林的性能相比较的实验结果。每一种组合方法的基分类器都由 50 棵决策树组成。表中报告的分类准确率通过十折交叉验证得到。注意，在许多数据集上，组合分类器都优于单个的决策树分类器。

表 5-5 决策树分类器与三种组合方法的准确率比较

数据集	（属性，类，记录）个数	决策树（%）	装袋（%）	提升（%）	RF（%）
Anneal	(39, 6, 898)	92.09	94.43	95.43	95.43
Australia	(15, 2, 690)	85.51	87.10	85.22	85.80
Auto	(26, 7, 205)	81.95	85.37	85.37	84.39
Breast	(11, 2, 699)	95.14	96.42	97.28	96.14
Cleve	(14, 2, 303)	76.24	81.52	82.18	82.18
Credit	(16, 2, 690)	85.8	86.23	86.09	85.8
Diabetes	(9, 2, 768)	72.40	76.30	73.18	75.13
German	(21, 2, 1000)	70.90	73.40	73.00	74.5

（续）

数据集	（属性，类，记录）个数	决策树（%）	装袋（%）	提升（%）	RF（%）
Glass	(10, 7, 214)	67.29	76.17	77.57	78.04
Heart	(14, 2, 270)	80.00	81.48	80.74	83.33
Hepatitis	(20, 2, 155)	81.94	81.29	83.87	83.23
Horse	(23, 2, 368)	85.33	85.87	81.25	85.33
Ionosphere	(35, 2, 351)	89.17	92.02	93.73	93.45
Iris	(5, 3, 150)	94.67	94.67	94.00	93.33
Labor	(17, 2, 57)	78.95	84.21	89.47	84.21
Led7	(8, 10, 3200)	73.34	73.66	73.34	73.06
Lymphography	(19, 4, 148)	77.03	79.05	85.14	82.43
Pima	(9, 2, 768)	74.35	76.69	73.44	77.60
Sonar	(61, 2, 208)	78.85	78.85	84.62	85.58
Tic-tac-toe	(10, 2, 958)	83.72	93.84	98.54	95.82
Vehicle	(19, 4, 846)	71.04	74.11	78.25	74.94
Waveform	(22, 3, 5000)	76.44	83.30	83.90	84.04
Wine	(14, 3, 178)	94.38	96.07	97.75	97.75
Zoo	(17, 7, 101)	93.07	93.07	95.05	97.03

5.7　不平衡类问题

具有不平衡类分布的数据集在许多实际应用中都会见到。例如，一个监管产品生产线的下线产品的自动检测系统会发现，不合格产品的数量远远低于合格产品的数量。同样，在信用卡欺诈检测中，合法交易远远多于欺诈交易。在这两个例子中，属于不同类的实例数量都不成比例。不平衡程度随应用不同而不同——一个在六西格玛原则下运行的制造厂可能会在一百万件出售给顾客的产品中发现四件不合格品，而信用卡欺诈的量级可能是百分之一。尽管它们不常出现，但是在这些应用中，稀有类的正确分类比多数类的正确分类更有价值。然而，由于类分布是不平衡的，这就给那些已有的分类算法带来了很多问题。

准确率经常用来比较分类器的性能，然而它可能不适合评价从不平衡数据集得到的模型。例如，如果1%的信用卡交易是欺骗行为，则预测每个交易都合法的模型具有99%的准确率，尽管它检测不到任何欺骗交易。另外，用来指导学习算法的度量（如决策树归纳中的信息增益）也需要进行修改，以关注那些稀有类。

检测稀有类的实例好比大海捞针。因为这些实例很少出现，因此描述稀有类的模型趋向于是高度特殊化的。例如，在基于规则的分类器中，为稀有类提取的规则通常涉及大量的属性，并很难简化为更一般的、具有很高覆盖率的规则（不像那些多数类的规则）。这样的模型也很容易受训练数据中噪声的影响。因此，许多已有的算法不能很好地检测稀有类的实例。

本节将给出一些为处理不平衡类问题而开发的方法。首先，介绍除准确率外的一些可选度量，以及一种称为ROC分析的图形化方法。然后，描述如何使用代价敏感学习和基于抽样的方法来改善稀有类的检测。

5.7.1　可选度量

由于准确率度量将每个类看得同等重要，因此它可能不适合用来分析不平衡数据集。在不平

衡数据集中，稀有类比多数类更有意义。对于二元分类，稀有类通常记为正类，而多数类被认为是负类。表 5-6 显示了汇总分类模型正确和不正确预测的实例数目的混淆矩阵。

表 5-6 类不是同等重要的二类分类问题的混淆矩阵

		预测的类	
		+	−
实际的类	+	f_{++}(TP)	f_{+-}(FN)
	−	f_{-+}(FP)	f_{--}(TN)

在谈到混淆矩阵列出的计数时，经常用到下面的术语。

- 真正（true positive, TP）或 f_{++}，对应于被分类模型正确预测的正样本数。
- 假负（false negative, FN）或 f_{+-}，对应于被分类模型错误预测为负类的正样本数。
- 假正（false positive, FP）或 f_{-+}，对应于被分类模型错误预测为正类的负样本数。
- 真负（ture negative, TN）或 f_{--}，对应于被分类模型正确预测的负样本数。

混淆矩阵中的计数可以表示为百分比的形式。**真正率**（true positive rate, *TPR*）或**灵敏度**（sensitivity）定义为被模型正确预测的正样本的比例，即：

$$TPR = TP/(TP + FN)$$

同理，**真负率**（Ture Negative Rate, TNR）或**特指度**（specificity）定义为被模型正确预测的负样本的比例，即：

$$TNR = TN/(TN + FP)$$

最后，**假正率**（false positive rate, *FPR*）定义为被预测为正类的负样本比例，即：

$$FPR = FP/(TN + FP)$$

而**假负率**（false negative rate, *FNR*）定义为被预测为负类的正样本比例，即：

$$FNR = FN/(TP + FN)$$

召回率（recall）和**精度**（precision）是两个广泛使用的度量，用于成功预测一个类比预测其他类更加重要的应用。下面给出精度（p）和召回率（r）的形式化定义：

$$p = \frac{TP}{TP + FP} \tag{5-74}$$

$$r = \frac{TP}{TP + FN} \tag{5-75}$$

精度确定在分类器断言为正类的那部分记录中实际为正类的记录所占的比例。精度越高，分类器的假正类错误率就越低。召回率度量被分类器正确预测的正样本的比例。具有高召回率的分类器很少将正样本误分为负样本。实际上，召回率的值等于真正率。

可以构造一个基线模型，它最大化其中一个度量而不管另一个。例如，将每一个记录都声明为正类的模型具有完美的召回率，但它的精度却很差。相反，将匹配训练集中任何一个正记录的检验记录都指派为正类的模型具有很高的精度，但召回率很低。构建一个最大化精度和召回率的模型是分类算法的主要任务之一。

精度和召回率可以合并成另一个度量，称为 F_1 度量。

$$F_1=\frac{2rp}{r+p}=\frac{2\times TP}{2\times TP+FP+FN} \tag{5-76}$$

原则上，F_1 表示召回率和精度的调和均值，即：

$$F_1=\frac{2}{\frac{1}{r}+\frac{1}{p}}$$

两个数 x 和 y 的调和均值趋向于接近较小的数。因此，一个高的 F_1 度量值确保精度和召回率都比较高。下面的例子比较了调和均值、几何均值和算术均值。

例 5.8　考虑两个正数 $a=1$ 和 $b=5$。它们的算术均值 $\mu_a=(a+b)/2=3$，几何均值 $\mu_g=\sqrt{ab}=2.236$。它们的调和均值 $\mu_h=(2\times 1\times 5)/6=1.667$，它比算术均值和几何均值更接近于 a 和 b 中的较小值。 □

更一般地，可以用 F_β 度量考察召回率和精度之间的折中：

$$F_\beta=\frac{(\beta^2+1)rp}{r+\beta^2 p}=\frac{(\beta^2+1)\times TP}{(\beta^2+1)TP+\beta^2 FP+FN} \tag{5-77}$$

精度和召回率分别是 $\beta=0$ 和 $\beta=\infty$ 时 F_β 的特例。低 β 值使得 F_β 值接近于精度，高 β 值使得 F_β 值接近于召回率。

更一般地，俘获 F_β 值和准确率的度量是加权准确率度量，由下式定义：

$$\text{加权准确率}=\frac{w_1TP+w_4TN}{w_1TP+w_2FP+w_3FN+w_4TN} \tag{5-78}$$

加权准确率和其他的性能度量值之间的关系汇总在下表中：

度量	w_1	w_2	w_3	w_4
召回率	1	1	0	0
精度	1	0	1	0
F_β 值	β^2+1	β^2	1	0
准确率	1	1	1	1

5.7.2　接受者操作特征曲线

接受者操作特征（receiver operating characteristic，ROC）曲线是显示分类器真正率和假正率之间折中的一种图形化方法。在一个 ROC 曲线中，真正率（TPR）沿 y 轴绘制，而假正率（FPR）显示在 x 轴上。沿着曲线的每个点对应于一个分类器归纳的模型。图 5-41 显示了一对分类器 M_1 和 M_2 的 ROC 曲线。

ROC 曲线上有几个关键点，它们都有公认的解释。

($TPR=0, FPR=0$)：把每个实例都预测为负类的模型。

($TPR=1, FPR=1$)：把每个实例都预测为正类的模型。

($TPR=1, FPR=0$)：理想模型。

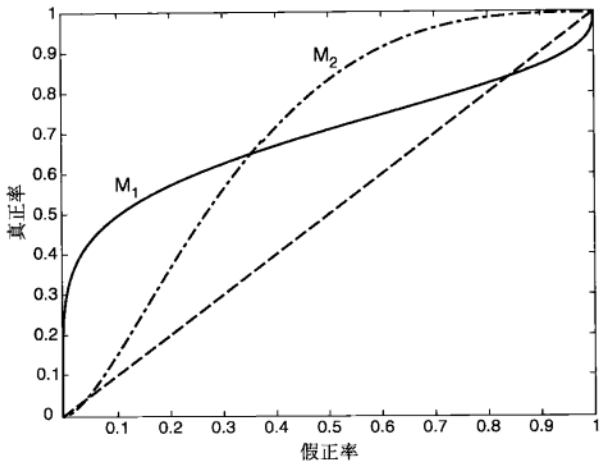


图 5-41 两个不同分类器的 ROC 曲线

一个好的分类模型应该尽可能靠近图的左上角，而一个随机猜测的模型应位于连接点($TPR = 0, FPR = 0$)和($TPR = 1, FPR = 1$)的主对角线上。随机猜测是指以固定的概率 p 把记录分为正类，而不考虑它的属性集。例如，考虑一个包含 n_+个正实例和 n_-个负实例的数据集。随机分类器期望正确地分类 pn_+个正实例，而误分 pn_-个负实例，因此，分类器的 TPR 是$(pn_+)/n_+ = p$，而它的 FPR 是$(pn_-)/n_- = p$。由于 TPR 和 FPR 相等，因此随机预测分类器的 ROC 曲线总是位于主对角线上。

ROC 曲线有助于比较不同分类器的相对性能。在图 5-41 中，当 FPR 小于 0.36 时，M_1要好于 M_2，而 FPR 大于 0.36 时 M_2较好。很明显，这两个分类器各有各的长处。

ROC 曲线下方的面积（AUC）提供了评价模型的平均性能的另一种办法。如果模型是完美的，则它在 ROC 曲线下方的面积等于 1。如果模型仅仅是简单地随机猜测，则 ROC 曲线下方的面积等于 0.5。如果一个模型好于另一个，则它的 ROC 曲线下方面积较大。

产生 ROC 曲线

为了绘制 ROC 曲线，分类器应当能够产生连续值输出，可以用来从最有可能到最不可能分为正类的记录，对它的预测排序。这些输出可能对应于贝叶斯分类器产生的后验概率或人工神经网络产生的数值输出。下面给出产生 ROC 曲线的过程。

(1) 假定为正类定义了连续值输出，对检验记录按它们的输出值递增排序。

(2) 选择秩最低的检验记录（即输出值最低的记录），把选择的记录以及那些秩高于它的记录指派为正类。这种方法等价于把所有的检验实例都分为正类。因为所有的正检验实例都被正确分类，而所有的负测试实例都被误分，因此 $TPR=FPR=1$。

(3) 从排序列表中选择下一个检验记录，把选择的记录以及那些秩高于它的记录指派为正类，而把那些秩低于它的记录指派为负类。通过考察前面选择的记录的实际类标号来更新 TP 和 FP 计数。如果前面选择的记录为正类，则 TP 计数减少而 FP 计数不变。如果前面选择的记录为负类，则 FP 计数减少而 TP 计数不变。

(4) 重复步骤 3 并相应地更新 TP 和 FP 计数，直到最高秩的记录被选择。

(5) 根据分类器的 FPR 画出 TPR 曲线。

图5-42显示了一个如何计算 ROC 曲线的例子。检验集中有5个正实例和5个负实例。检验记录的类标号显示在表的第一行。第二行对应于每个记录排序后的输出值，例如，它们可能对应于朴素贝叶斯分类器产生的后验概率 $P(+|\mathbf{x})$。接下来的六行包括 *TP* 计数、*FP* 计数、*TN* 计数和 *FN* 计数，以及它们对应的 *TPR* 和 *FPR*。于是从左到右填表。开始时，所有的记录都被预测为正类，因此 $TP = FP = 5$，$TPR = FPR = 1$。然后，指派有最低输出值的检验实例为负类。因为选择的记录实际上是正类，因此 *TP* 计数从5减到4，而 *FP* 计数不变，并相应地更新 *TPR* 和 *FPR* 值。重复这个过程直至到达列表的末尾，这时 $TPR=0$，$FPR=0$。这个例子的 ROC 曲线如图5-43所示。

类	+	−	+	−	−	−	+	−	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

图 5-42　构造 ROC 曲线

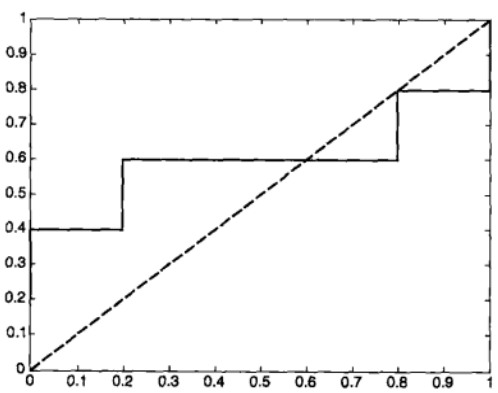


图 5-43　图 5-42 所示数据集的 ROC 曲线

5.7.3　代价敏感学习

代价矩阵对将一个类的记录分类到另一个类的惩罚进行编码。令 $C(i, j)$表示预测一个 i 类记录为 j 类的代价。使用这种记号，$C(+, -)$是犯一个假负错误的代价，而 $C(-, +)$是产生一个假警告的代价。代价矩阵中的一个负项表示对正确分类的奖励。给定一个 N 个记录的检验集，模型 M 的总代价是：

$$C_t(M) = TP\times C(+,+) + FP\times C(-,+) + FN\times C(+,-) + TN\times C(-,-) \tag{5-79}$$

在 0/1 代价矩阵中，即 $C(+, +) = C(-, -) = 0$ 而 $C(+, -) = C(-, +) = 1$，可以证明总代价等价于误分类的数目。

$$C_t(M) = 0\times(TP+TN) + 1\times(FP+FN) = N\times Err \tag{5-80}$$

其中，*Err* 是分类器的误差率。

例 5.9 考虑表 5-7 所示的代价矩阵。犯假负错误的代价是犯假警告的 100 倍。换句话说，漏检测出任何 1 个正样本与犯 100 个假警告一样糟糕。给定具有表 5-8 所示的混淆矩阵的分类模型，每一个模型的总代价是：

$$C_t(M_1) = 150 \times (-1) + 60 \times 1 + 40 \times 100 = 3910$$

$$C_t(M_2) = 250 \times (-1) + 5 \times 1 + 45 \times 100 = 4255$$

注意，尽管模型 M_2 同时改善了它的真正计数和假正计数，但是仍然较差，因为这些改善是建立在增加代价更高的假负错误之上。而标准的准确率度量更趋向于 M_2 优于 M_1。 □

表 5-7 例 5.9 的代价矩阵

		预测的类	
		类 =+	类 =−
实际的类	类 =+	−1	100
	类 =−	1	0

表 5-8 两个分类模型的混淆矩阵

模型 M_1		预测的类	
		类 +	类 −
实际的类	类 +	150	40
	类 −	60	250

模型 M_2		预测的类	
		类 +	类 −
实际的类	类 +	250	45
	类 −	5	200

代价敏感分类技术在构建模型的过程中考虑代价矩阵，并产生代价最低的模型。例如，如果假负错误代价最高，则学习算法将通过向负类扩展它的决策边界来减少这些错误，如图5-44所示。这种方法产生的模型覆盖更多的正类样本，尽管其代价是产生了一些额外的假警告。

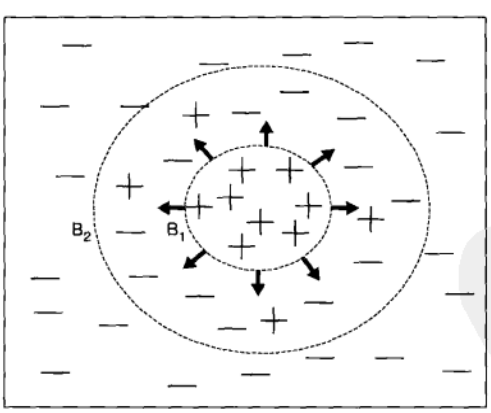


图 5-44 修改决策边界（从 B_1 到 B_2）以减少分类器的假负错误

有许多办法将代价信息加入分类算法中。例如，在决策树归纳过程中，代价信息可以用来：(1)选择用以分裂数据的最好的属性；(2)决定子树是否需要剪枝；(3)处理训练记录的权值，使得学习算法收敛到代价最低的决策树；(4)修改每个叶结点上的决策规则。为了解释最后一种方法，令 $p(i|t)$表示属于叶结点 t 的类 i 的训练记录所占的比例。如果下面的条件成立，一个典型的二元

分类问题的决策规则将正类指派到结点 t：

$$\begin{aligned} & p(+|t) > p(-|t) \\ \Rightarrow\ & p(+|t) > (1 - p(+|t)) \\ \Rightarrow\ & 2p(+|t) > 1 \\ \Rightarrow\ & p(+|t) > 0.5 \end{aligned} \tag{5-81}$$

前面的决策规则表明，叶结点的类标号取决于到达该结点的训练记录的多数类。注意，这个规则假定对于正样本和负样本，误分的代价都是相同的。这个决策规则等价于 4.3.5 节的公式（4-48）给出的表达式。

代价敏感的学习算法不是采用多数表决，而是赋予结点 t 类标号 i，如果它最小化如下表达式：

$$C(i|t) = \sum_j p(j|t)C(j,i) \tag{5-82}$$

在 $C(+, +) = C(-, -) = 0$ 的情况下，叶结点 t 被指派为正类，如果：

$$\begin{aligned} & p(+|t)C(+,-) > p(-|t)C(-,+) \\ \Rightarrow\ & p(+|t)C(+,-) > (1 - p(+|t))C(-,+) \\ \Rightarrow\ & p(+|t) > \frac{C(-,+)}{C(-,+) + C(+,-)} \end{aligned} \tag{5-83}$$

这个表达式说明，可以把决策规则的阈值从 0.5 修改为 $C(-, +)/(C(+, -) + C(-, +))$，得到一个代价敏感的分类器。如果 $C(-, +) < C(+, -)$，则阈值将小于 0.5。这个结果是有意义的，因为一个假负错误比一个假警告代价高。降低阈值将向负类扩展决策边界，如图 5-44 所示。

5.7.4　基于抽样的方法

抽样是处理不平衡类问题的另一种广泛使用的方法。抽样的主要思想是改变实例的分布，从而帮助稀有类在训练数据集中得到很好的表示。用于抽样的一些现有的技术包括不充分抽样（undersampling）、过分抽样（oversampling）和两种技术的混合。为了解释这些技术，考虑一个包含 100 个正样本和 1 000 个负样本的数据集。

在不充分抽样的情况下，取 100 个负样本的一个随机抽样，与所有的正样本一起形成训练集。这种方法的一个潜在的问题是，一些有用的负样本可能没有选出来用于训练，因此会生成一个不太优的模型。克服这个问题的一个可行的方法是多次执行不充分抽样，并归纳类似于组合学习方法的多分类器。也可以使用聚焦的不充分抽样（focused undersampling），这时抽样程序精明地确定应该被排除的负样本，如那些远离决策边界的样本。

过分抽样复制正样本，直到训练集中正样本和负样本一样多。图 5-45 说明了使用分类法（如决策树）构建决策边界时过分抽样对其产生的影响。不使用过分抽样，只有在图 5-45a 中左下角的那些正样本被正确分类，位于图中间的正样本没有被正确分类，因为没有足够的样本来确定分离正样本和负样本的新决策边界。过分抽样提供了需要的额外样本，确保围绕该正样本的决策边界不被剪除，如图 5-45b 所示。

(a) 不使用过分抽样　　(b) 使用过分抽样

图 5-45　解释稀有类过分抽样的影响

然而，对于噪声数据，过分抽样可能导致模型过分拟合，因为一些噪声样本也可能被复制多次。原则上，过分抽样没有向训练集中添加任何新的信息。对正样本的复制仅仅是阻止学习算法剪掉模型中描述包含很少训练样本的区域的那部分（即小的不相连的部分）。增加的正样本有可能增加建立模型的计算时间。

混合方法使用二者的组合，对多数类进行不充分抽样，而对稀有类进行过分抽样来获得均匀的类分布。不充分抽样可以采用随机或聚焦的子抽样。另一方面，过分抽样可以通过复制已有的正样本，或在已有的正样本的邻域中产生新的正样本来实现。在后一种方法中，必须首先确定每一个已有的正样本的 k-最近邻，然后，在连接正样本和一个 k-最近邻的线段上的某个随机点产生一个新的正样本。重复该过程，直到正样本的数目达到要求。不像数据复制方法，新的样本能够向外扩展正类的决策边界，与图 5-44 中的方法类似。然而，这种方法仍然可能受模型过分拟合的影响。

5.8　多类问题

本章描述的一些分类技术（如支持向量机和 AdaBoost）原先是为二元分类问题设计的。然而在解决许多现实世界的问题（如特征识别、人脸识别和文本分类等）时，输入数据都被划分为多于两个类。本节给出扩展二元分类器以处理多类问题的方法。为了说明这些方法，令 $Y = \{y_1, y_2, \cdots, y_K\}$ 是输入数据的类标号的集合。

第一种方法将多类问题分解成 K 个二类问题。为每一个类 $y_i \in Y$ 创建一个二类问题，其中所有属于 y_i 的样本都被看作正类，而其他样本作为负类。然后，构建一个二元分类器，将属于 y_i 的样本从其他类中分离出来。这种方法称为一对其他（1-r）方法。

第二种方法称为一对一（1-1）方法，它构建 $K(K-1)/2$ 个二类分类器，每一个分类器用来区分一对类(y_i, y_j)。当为类(y_i, y_j)构建二类分类器时，不属于 y_i 或 y_j 的样本被忽略掉。不论 1-1 还是 1-r 方法，都是通过组合所有二元分类器的预测对检验实例分类。组合预测的典型做法是使用投票表决，将检验样本指派到得票最多的类。在 1-r 方法中，如果一个样本被分为负类，则除正类之外的所有类都得到一票。然而，这种方法可能导致不同类的平局。另一种可能是将二元分类器的输出变换成概率估计，然后将检验实例指派到具有最高概率的类。

例 5.10　考虑一个多类问题，其中 $Y = \{y_1, y_2, y_3, y_4\}$。假设根据 1-r 方法将一个检验实例分类为(+, −, −, −)。换言之，当 y_1 作为正类时它被分为正类，而当 y_2，y_3，y_4 作为正类时它被分为

负类。使用简单的多数表决，既然 y_1 得到最高的投票数 4，而其他类仅仅得到 3 票，因此检验实例被分类为 y_1。

假定使用 1-1 方法将检验实例分类如下：

二类分类器类对	$+:y_1$ $-:y_2$	$+:y_1$ $-:y_3$	$+:y_1$ $-:y_4$	$+:y_2$ $-:y_3$	$+:y_2$ $-:y_4$	$+:y_3$ $-:y_4$
分类	+	+	−	+	−	+

表的上面两行对应选来构建分类器的类对(y_i, y_j)，而最后一行表示检验实例的预测类。在组合预测后，y_1 和 y_4 都得到 2 票 ，而 y_2 和 y_3 仅仅得到 1 票。依赖于平局处理策略，检验实例被分为 y_1 或 y_4。 □

纠错输出编码

前面介绍的两种方法的一个问题是，它们对二元分类的错误太敏感。对于例 5.10 中给出的 1-r 方法，如果有一个二元分类器作出了错误的预测，则组合分类器可能就以平局或一个错误的预测结束。例如，假设由于第三个分类器的误分，检验实例被分为(+, −, +, −)。这时，除非考虑与每个类预测相关联的概率，否则很难决定样本应分为 y_1 类还是 y_3 类。

纠错输出编码（error-correcting output coding，ECOC）方法提供了一种处理多类问题的更鲁棒的方法。这种方法受信息理论中通过噪声信道发送信息的启发。其基本思想是借助于代码字向传输信息中增加一些冗余，从而使得接收方能发现接收信息中的一些错误，而且如果错误量很少，还可能恢复原始信息。

对于多类学习，每个类 y_i 用一个长度为 n 的唯一的位串来表示，称为它的代码字。然后训练 n 个二元分类器，预测代码字串的每个二进位。检验实例的预测类由这样的代码字给出，该代码字到二元分类器产生的代码字海明距离最近。注意，两个位串之间的海明距离是它们的不同的二进位的数目。

例 5.11　考虑一个多类问题，其中 $Y = \{y_1, y_2, y_3, y_4\}$。假定使用下面的 7 位代码字对类进行编码：

类	代码字						
y_1	1	1	1	1	1	1	1
y_2	0	0	0	0	1	1	1
y_3	0	0	1	1	0	0	1
y_4	0	1	0	1	0	1	0

代码字的每个二进位用来训练一个二元分类器。如果一个检验实例被二类分类器分类为(0, 1, 1, 1, 1, 1, 1)，则该代码字与 y_1 之间的海明距离为 1，而与其他类之间的海明距离为 3。因此，该检验实例被分类为 y_1。 □

纠错码的一个有趣的性质是，如果任意代码字对之间的最小海明距离为 d，则输出代码任意 $\lfloor (d-1)/2 \rfloor$ 个错误可以使用离它最近的代码字纠正。在例 5.11 中，因为任意代码字对之间的最小海明距离为 4，因此组合分类器可以容忍 7 个二类分类器中的 1 个出错。如果出错的分类器超过一个，则组合分类器将不能校正这些错误。

一个很重要的问题是如何为不同的类设计合适的代码字集合。从编码理论来说，目前已经开

发出了大量的能够产生具有有限海明距离的 n 位代码字的算法。然而，这些算法的讨论已经超出本书范围。值得一提的是，为通信任务设计纠错码明显不同于多类学习的纠错码。对通信任务，代码字应该最大化各行之间的海明距离，使得纠错可以进行。然而，多类学习要求将代码字列向和行向的距离很好地分开。较大的列向距离可以确保二元分类器是相互独立的，而这正是组合学习算法的一个重要要求。

文献注释

Mitchell[208]从机器学习的角度极好地介绍了许多分类技术。对分类的更广泛的论述还可以从下面文献中找到：Duda 等[180]、Webb[219]、Fukunaga[187]、Bishop[159]、Hastie 等[192]、Cherkassky 和 Mulier[167]、Witten 和 Frank[221]、Hand 等[190]、Han 和 Kamber[189]以及 Dunham[181]。

基于规则分类器的直接方法采用顺序覆盖模式来归纳分类规则。Holt 的 1R[195]是最简单的基于规则的分类器，因为它的规则集只包含单个的规则。尽管简单，但是 Holt 发现，对于一些属性和类标号显示出很强的一对一联系的数据集，1R 的性能和其他分类器相当。基于规则的分类器的其他例子包括 IREP[184]、RIPPER[170]、CN2[168,169]、AQ[207]、RISE[176]和 ITRULE[214]。表 5-9 显示了其中 4 个分类器的特征对比。

表 5-9 各种基于规则分类器的对比

	RIPPER	CN2（无序的）	CN2（有序的）	AQR
规则增长策略	一般到特殊	一般到特殊	一般到特殊	一般到特殊（以一个正样本作种子）
评估度量	FOIL 信息增益	拉普拉斯	熵和似然率	真正类的个数
停止规则增长条件	所有样本都属于同一个类	无性能提高	无性能提高	规则只覆盖正类
规则剪枝	减少错误	无	无	无
实例删除	正的和负的	正的	正的	正的和负的
停止增加规则条件	Error > 50%或基于 MDL	无性能提高	无性能提高	所有正样本都被覆盖
规则集剪枝	替换或修改规则	统计检验	无	无
搜索策略	贪心	定向搜索	定向搜索	定向搜索

对基于规则的分类器，规则的前件可以推广到包含任意命题或一阶逻辑表达式（例如，Horn 子句）。对基于一阶逻辑规则分类器感兴趣的读者可以参阅相关文献，如[208]或关于归纳逻辑程序设计的大量文献[209]。Quinlan[211]给出了 C4.5 算法，从决策树中提取分类规则。Andrews 等[157]给出了从人工神经网络中提取规则的间接的方法。

Cover 和 Hart[172]从贝叶斯定理的角度给出了最近邻分类方法的综述。Aha 在[155]中提供了基于实例方法的理论和实验评价。PEBLS 是 Cost 和 Salzberg[171]提出的一种最近邻分类算法，它能处理包含标称属性的数据集。在 PEBLS 中，赋予每个训练实例一个权重因子，而因子取决于实例帮助作出正确预测的次数。Han 等[188]给出了一个调整权重的最近邻算法，使用一种贪心的、爬山式的优化算法来学习特征的权重。

朴素贝叶斯分类器有许多作者介绍过，包括Langley等[203]，Ramoni和Sebastiani[212]，Lewis[204]，以及Domingos和Pazzani[178]。尽管朴素贝叶斯分类器中使用的独立性假设看上去很

不现实，但是在诸如文本分类等应用领域，该方法的性能却出奇地好。通过允许某些属性相互依赖，贝叶斯信念网络提供了一种更灵活的方法。Heckerman在[194]中给出了关于贝叶斯信念网络的很好的指南。

Vapnik[217,218]已经写了两本关于支持向量机（SVM）的权威书籍。关于SVM和核方法的其他一些有用的资源包括Cristianini和Shawe-Taylor[173]以及Schölkopf和Smola[213]的书。另外还有一些关于SVM的评论文章，包括Burges[164]，Bennet等[158]，Hearst[193]和Mangasarian[205]。

Dietterich[174]给出了机器学习中组合方法的概述。Breiman[161]提出了装袋方法。Freund和Schapire[186]提出了AdaBoost算法。Arcing是自适应再抽样和组合（adaptive resampling and combining）的缩写，它是Breiman[162]提出的提升算法的一个变形。它对训练实例赋予不一致的权重来对数据进行再抽样，从而建立训练数据集的组合分类器。不像AdaBoost，在决定测试样本的类标号时，基分类器的投票是不加权的。随机森林方法在Breiman[163]中有所介绍。

关于挖掘稀有类和不平衡数据集的工作可以参看Chawla等[166]和Weiss[220]写的综述。许多作者都介绍过挖掘不平衡数据集的基于样本的方法，如Kubat和Matwin[202]、Japkowitz[196]以及Drummond和Holte[179]。Joshi等[199]讨论了提升算法对稀有类建模的局限性。挖掘稀有类的其他一些算法包括SMOTE[165]、PNrule[198]和CREDOS[200]。

存在一些适合分不平衡类问题的可选度量。精度、召回率和F_1度量是信息检索中广泛使用的度量[216]。ROC分析原先用于信号检测理论。Bradley[160]研究了以ROC曲线下方面积作为机器学习算法的性能度量的应用。Provost和Fawcett在[210]中提出了使用ROC曲线的凸起来比较分类器性能的方法。Ferri等[185]开发了一种在决策树分类器上进行ROC分析的方法。他们还提出了在树增长过程中使用ROC曲线下方面积（AUC）作为分裂标准。Joshi[197]从分析稀有类的角度考察了这些度量的性能。

关于代价敏感学习的大量文献可以在ICML2000关于代价敏感学习研讨会的联机论文集中找到。Elkan [182]研究了代价矩阵的特征。Margineantu和Dietterich[206]考察了将代价信息合并到C4.5学习算法中的多种方法，包括包装的方法，基于类分布的方法和基于损失（loss-based）的方法。其他一些独立于算法的代价敏感学习方法包括AdaCost[183]，MetaCost[177]和costing[222]。

关于多类学习，也存在大量文献。这包括Hastie和Tibshirani[191]、Allwein等[156]、Kong和Dietterich[201]以及Tax和Duin[215]的著作。Dietterich和Bakiri[175]提出了纠错输出编码（ECOC）方法。他们也介绍了适合解决多类问题的代码设计技术。

参考文献

[155] D. W. Aha. *A study of instance-based algorithms for supervised learning tasks: mathematical, empirical, and psychological evaluations*. PhD thesis, University of California, Irvine, 1990.

[156] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing Multiclass to Binary: A Unifying Approach to Margin Classifiers. *Journal of Machine Learning Research*, 1:113 - 141, 2000.

[157] R. Andrews, J. Diederich, and A. Tickle. A Survey and Critique of Techniques For Extracting Rules From Trained Artificial Neural Networks. *Knowledge Based Systems*, 8(6):373 - 389, 1995.

[158] K. Bennett and C. Campbell. Support Vector Machines: Hype or Hallelujah. *SIGKDD Explorations*, 2(2):1 - 13, 2000.

[159] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, U.K., 1995.

[160] A. P. Bradley. The use of the area under the ROC curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30(7):1145 - 1149, 1997.

[161] L. Breiman. Bagging Predictors. *Machine Learning*, 24(2):123 - 140, 1996.

[162] L. Breiman. Bias, Variance, and Arcing Classifiers. Technical Report 486, University of California, Berkeley, CA, 1996.

[163] L. Breiman. Random Forests. *Machine Learning*, 45(1):5 - 32, 2001.

[164] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121 - 167, 1998.

[165] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321 - 357, 2002.

[166] N. V. Chawla, N. Japkowicz, and A. Kolcz. Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explorations*, 6(1):1 - 6, 2004.

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

[168] P. Clark and R. Boswell. Rule Induction with CN2: Some Recent Improvements. In *Machine Learning: Proc. of the 5th European Conf. (EWSL-91)*, pages 151 - 163, 1991.

[169] P. Clark and T. Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3(4): 261 - 283, 1989.

[170] W. W. Cohen. Fast Effective Rule Induction. In *Proc. of the 12th Intl. Conf. on Machine Learning*, pages 115 - 123, Tahoe City, CA, July 1995.

[171] S. Cost and S. Salzberg. A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning*, 10:57 - 78, 1993.

[172] T. M. Cover and P. E. Hart. Nearest Neighbor Pattern Classification. *Knowledge Based Systems*, 8(6):373 - 389, 1995.

[173] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[174] T. G. Dietterich. Ensemble Methods in Machine Learning. In *First Intl. Workshop on Multiple Classifier Systems*, Cagliari, Italy, 2000.

[175] T. G. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, 2:263 - 286, 1995.

[176] P. Domingos. The RISE system: Conquering without separating. In *Proc. of the 6th IEEE Intl. Conf. on Tools with Artificial Intelligence*, pages 704 - 707, New Orleans, LA, 1994.

[177] P. Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 155 - 164, San Diego, CA, August 1999.

[178] P. Domingos and M. Pazzani. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, 29(2-3):103 - 130, 1997.

[179] C. Drummond and R. C. Holte. C4.5, Class imbalance, and Cost sensitivity: Why under-sampling beats over-sampling. In *ICML'2004 Workshop on Learning from Imbalanced Data Sets II*, Washington, DC, August 2003.

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

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

[182] C. Elkan. The Foundations of Cost-Sensitive Learning. In *Proc. of the 17th Intl. Joint Conf. on Artificial Intelligence*, pages 973 - 978, Seattle, WA, August 2001.

[183] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: misclassification costsensitive boosting. In *Proc. of the 16th Intl. Conf. on Machine Learning*, pages 97 - 105, Bled, Slovenia, June 1999.

[184] J. Fürnkranz and G. Widmer. Incremental reduced error pruning. In *Proc. of the 11th Intl. Conf. on Machine Learning*, pages 70 - 77, New Brunswick, NJ, July 1994.

[185] C. Ferri, P. Flach, and J. Hernandez-Orallo. Learning Decision Trees Using the Area Under the ROC Curve. In *Proc. of the 19th Intl. Conf. on Machine Learning*, pages 139 - 146, Sydney, Australia, July 2002.

[186] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 - 139, 1997.

[187] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.

[188] E.-H. Han, G. Karypis, and V. Kumar. Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification. In *Proc. of the 5th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Lyon, France, 2001.

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

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

[191] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, 26(2):451 - 471, 1998.

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

[193] M. Hearst. Trends & Controversies: Support Vector Machines. *IEEE Intelligent Systems*, 13(4):18 - 28, 1998.

[194] D. Heckerman. Bayesian Networks for Data Mining. *Data Mining and Knowledge Discovery*, 1(1): 79 - 119, 1997.

[195] R. C. Holte. Very Simple Classification Rules Perform Well on Most Commonly Used Data sets. *Machine Learning*, 11:63 - 91, 1993.

[196] N. Japkowicz. The Class Imbalance Problem: Significance and Strategies. In *Proc. of the 2000 Intl. Conf. on Artificial Intelligence: Special Track on Inductive Learning*, volume 1, pages 111 - 117, Las Vegas, NV, June 2000.

[197] M. V. Joshi. On Evaluating Performance of Classifiers for Rare Classes. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, Maebashi City, Japan, December 2002.

[198] M. V. Joshi, R. C. Agarwal, and V. Kumar. Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction. In *Proc. of 2001 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 91 - 102, Santa Barbara, CA, June 2001.

[199] M. V. Joshi, R. C. Agarwal, and V. Kumar. Predicting rare classes: can boosting make any weak learner strong? In *Proc. of the 8th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 297 - 306, Edmonton, Canada, July 2002.

[200] M. V. Joshi and V. Kumar. CREDOS: Classification Using Ripple Down Structure (A Case for Rare Classes). In *Proc. of the SIAM Intl. Conf. on Data Mining*, pages 321 - 332, Orlando, FL, April 2004.

[201] E. B. Kong and T. G. Dietterich. Error-Correcting Output Coding Corrects Bias and Variance. In *Proc. of the 12th Intl. Conf. on Machine Learning*, pages 313 - 321, Tahoe City, CA, July 1995.

[202] M. Kubat and S. Matwin. Addressing the Curse of Imbalanced Training Sets: One Sided Selection. In *Proc. of the 14th Intl. Conf. on Machine Learning*, pages 179 - 186, Nashville, TN, July 1997.

[203] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *Proc. of the 10th National Conf. on Artificial Intelligence*, pages 223 - 228, 1992.

[204] D. D. Lewis. Naive Bayes at Forty: The Independence Assumption in Information Retrieval. In *Proc. of the 10th European Conf. on Machine Learning (ECML 1998)*, pages 4 - 15, 1998.

[205] O. Mangasarian. Data Mining via Support Vector Machines. Technical Report Technical Report 01-05, Data Mining Institute, May 2001.

[206] D. D. Margineantu and T. G. Dietterich. Learning Decision Trees for Loss Minimization in Multi-Class Problems. Technical Report 99-30-03, Oregon State University, 1999.

[207] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains. In *Proc. of 5th National Conf. on Artificial Intelligence*, Orlando, August 1986.

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

[209] S. Muggleton. *Foundations of Inductive Logic Programming*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[210] F. J. Provost and T. Fawcett. Analysis and Visualization of Classifier Performance: Comparison under

Imprecise Class and Cost Distributions. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 43 - 48, Newport Beach, CA, August 1997.

[211] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan-Kaufmann Publishers, San Mateo, CA, 1993.

[212] M. Ramoni and P. Sebastiani. Robust Bayes classifiers. *Artificial Intelligence*, 125:209 - 226, 2001.

[213] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[214] P. Smyth and R. M. Goodman. An Information Theoretic Approach to Rule Induction from Databases. *IEEE Trans. on Knowledge and Data Engineering*, 4(4):301 - 316, 1992.

[215] D. M. J. Tax and R. P. W. Duin. Using Two-Class Classifiers for Multiclass Classification. In *Proc. of the 16th Intl. Conf. on Pattern Recognition (ICPR 2002)*, pages 124 - 127, Quebec, Canada, August 2002.

[216] C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, 1978.

[217] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[218] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

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

[220] G. M. Weiss. Mining with Rarity: A Unifying Framework. *SIGKDD Explorations*, 6 (1):7 - 19, 2004.

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

[222] B. Zadrozny, J. C. Langford, and N. Abe. Cost-Sensitive Learning by Cost-Proportionate Example Weighting. In *Proc. of the 2003 IEEE Intl. Conf. on Data Mining*, pages 435 - 442, Melbourne, FL, August 2003.

习　题

1. 考虑一个二值分类问题，属性集和属性值如下。
   - 空调 ={可用，不可用}。
   - 引擎 ={好，差}。
   - 行车里程 ={高，中，低}。
   - 生锈 ={是，否}。

   假设一个基于规则的分类器产生的规则集如下。

   行车里程=高 → 价值=低
   行车里程=低 → 价值=高
   空调=可用，引擎=好 → 价值=高
   空调=可用，引擎=差 → 价值=低
   空调=不可用 → 价值=低

   (a) 这些规则是互斥的吗？
   (b) 这些规则集是完全的吗？
   (c) 规则需要排序吗？
   (d) 规则集需要默认类吗？

2. RIPPER 算法（Cohen[170]）是早期算法 IREP（ Fürnkranz 和 Widmer[184]）的扩展。两个算法都使用**减少误差剪枝**（reduced-error pruning）方法来确定一个规则是否需要剪枝。减少误差剪枝方法使用一个确认集来估计分类器的泛化误差。考虑下面两个规则：

$$R_1: A \rightarrow C$$

$$R_2: A \wedge B \rightarrow C$$

R_2 是由 R_1 的左边添加合取项 B 得到的。现在的问题是，从规则增长和规则剪枝的角度来确定 R_2 是否比 R_1 好。为了确定规则是否应该剪枝，IREP 计算下面的度量：

$$v_{\text{IREP}} = \frac{p + (N - n)}{P + N}$$

其中，P 是确认集中正例的总数，N 是确认集中反例的总数，p 是确认集中被规则覆盖的正例数，而 n 是确认集中被规则覆盖的反例数。实际上，v_{IREP} 类似于确认集的分类准确率。IREP 偏向于 v_{IREP} 值较高的规则。另一方面，RIPPER 使用下面的度量来确定规则是否应该剪枝：

$$v_{\text{RIPPER}} = \frac{p - n}{p + n}$$

(a) 假设 R_1 覆盖 350 个正例和 150 个反例，而 R_2 覆盖 300 个正例和 50 个反例。计算 R_2 相对于 R_1 的 FOIL 信息增益。

(b) 考虑一个确认集，包含 500 个正例和 500 个反例。假设 R_1 覆盖 200 个正例和 50 个反例，R_2 覆盖 100 个正例和 5 个反例。计算 R_1 和 R_2 的 v_{IREP}，IREP 偏向于哪个规则？

(c) 计算(b)问题中的 v_{RIPPER}，RIPPER 偏向于哪个规则？

3. C4.5 规则是从决策树生成规则的间接方法的一个实现，而 RIPPER 是从数据中生成规则的直接方法的一个实现。

(a) 讨论两种方法的优缺点。

(b) 考虑一个数据集，其中类的大小差别很大（即有些类比其他类大得多）。在为较小的类寻找高准确率规则方面，哪一种方法（C4.5 规则和 RIPPER）更好？

4. 考虑一个训练集，包含 100 个正例和 400 个反例。对于下面的候选规则：

R_1: $A \rightarrow +$（覆盖 4 个正例和 1 个反例）

R_2: $B \rightarrow +$（覆盖个 30 个正例和 10 个反例）

R_3: $C \rightarrow +$（覆盖 100 个正例和 90 个反例）

根据下面的度量，确定最好规则和最差规则。

(a) 规则准确率。

(b) FOIL 信息增益。

(c) 似然比统计量。

(d) 拉普拉斯度量。

(e) m 度量（$k = 2$ 且 $p_+ = 0.2$）。

5. 图 5-4 给出了分类规则 R_1、R_2 和 R_3 的覆盖率。根据以下度量确定最好规则和最差规则。

(a) 似然比统计量。

(b) 拉普拉斯度量。

(c) m 度量（k=2 且 p_+=0.58）。

(d) 发现规则 R_1 后的准确率，这里不删除 R_1 覆盖的任何样例。

(e) 发现规则 R_1 后的准确率，这里仅删除 R_1 覆盖的正例。

(f) 发现规则 R_1 后的准确率，这里删除 R_1 覆盖的所有正例和反例。

6. (a) 假设本科生中抽烟的比例是 15%，研究生中抽烟的比例是 23%。如果大学生中研究生占 1/5，其余是本科生，那么抽烟的学生是研究生的概率是多少？

   (b) 根据(a)中的信息，随机选择一个大学里的学生，那么，该生是研究生或本科生的可能性哪个大？

   (c) 同(b)，假设学生是个抽烟者。

   (d) 假设 30%的研究生住学生宿舍，只有 10%的本科生住学生宿舍。如果一个学生抽烟又住宿舍，那么他（她）是研究生或本科生的可能性哪个大？可以假设住宿舍的学生和抽烟的学生相互独立。

7. 考虑表 5-10 中的数据集。

表 5-10　习题 7 的数据集

记录	*A*	*B*	*C*	类
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

   (a) 估计条件概率 $P(A|+)$，$P(B|+)$，$P(C|+)$，$P(A|-)$，$P(B|-)$和 $P(C|-)$。

   (b) 根据(a)中的条件概率，使用朴素贝叶斯方法预测测试样本($A=0, B=1, C=0$)的类标号。

   (c) 使用 m 估计方法（$p=1/2$ 且 $m=4$）估计条件概率。

   (d) 同(b)，使用(c)中的条件概率。

   (e) 比较估计概率的两种方法。哪一种更好？为什么？

8. 考虑表 5-11 中的数据集。

表 5-11　习题 8 的数据集

实例	*A*	*B*	*C*	类
1	0	0	1	−
2	1	0	1	+
3	0	1	0	−
4	1	0	0	−
5	1	0	1	+
6	0	0	1	+
7	1	1	0	−
8	0	0	0	−
9	0	1	0	+
10	1	1	1	+

(a) 估计条件概率 $P(A=1|+)$，$P(B=1|+)$，$P(C=1|+)$，$P(A=1|-)$，$P(B=1|-)$和 $P(C=1|-)$。

(b) 根据(a)中的条件概率，使用朴素贝叶斯方法预测测试样本($A=1, B=1, C=1$)的类标号。

(c) 比较 $P(A=1)$，$P(B=1)$和 $P(A=1, B=1)$。陈述 A、B 之间的关系。

(d) 对 $P(A=1)$，$P(B=0)$和 $P(A=1, B=0)$重复(c)的分析。

(e) 比较 $P(A=1, B=1|类=+)$与 $P(A=1|类=+)$和 $P(B=1|类=+)$。给定类+，变量 A、B 条件独立吗？

9. (a) 解释朴素贝叶斯分类器在图 5-46 数据集上的工作过程。

(b) 如果每个类进一步分割，得到四个类（$A1, A2, B1, B2$），朴素贝叶斯会工作得更好吗？

(c) 决策树在该数据集上怎样工作（两类问题）？四个类呢？

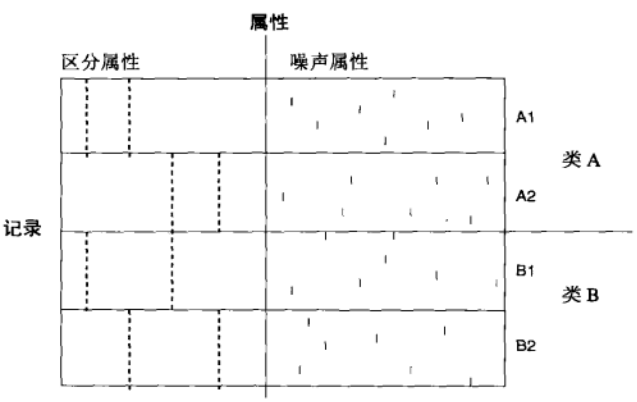


图 5-46　习题 9 的数据集

10. 使用下面的信息，重复例 5.3 的分析，寻找决策边界位置。

(a) 先验概率 $P(鳄鱼)=2\times P(美洲鳄)$。

(b) 先验概率 $P(美洲鳄)=2\times P(鳄鱼)$。

(c) 先验概率相同，但标准差不同，例如，$\sigma(鳄鱼)=4$，$\sigma(美洲鳄)=2$。

11. 图 5-47 给出了表 5-12 中的数据集对应的贝叶斯信念网络（假设所有属性都是二元的）。

(a) 画出网络中每个结点对应的概率表。

(b) 使用贝叶斯网络计算 P(引擎=差，空调=不可用)。

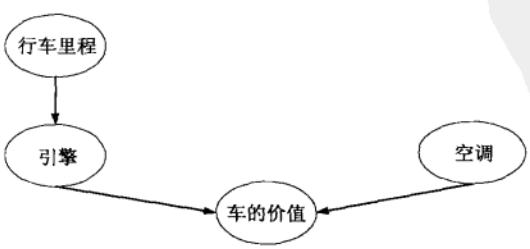


图 5-47　贝叶斯信念网络

表 5-12　习题 11 的数据集

行车里程	引擎	空调	车的价值=高的记录数	车的价值=低的记录数
高	好	可用	3	4
高	好	不可用	1	2
高	差	可用	1	5
高	差	不可用	0	4
低	好	可用	9	0
低	好	不可用	5	1
低	差	可用	1	2
低	差	不可用	0	2

12. 给定图 5-48 所示的贝叶斯网络，计算下面概率。

(a) $P(B=$好，$F=$空，$G=$空，$S=$是$)$[①]。

(b) $P(B=$差，$F=$空，$G=$非空，$S=$否$)$。

(c) 如果电池是差的，计算车发动起来的概率。

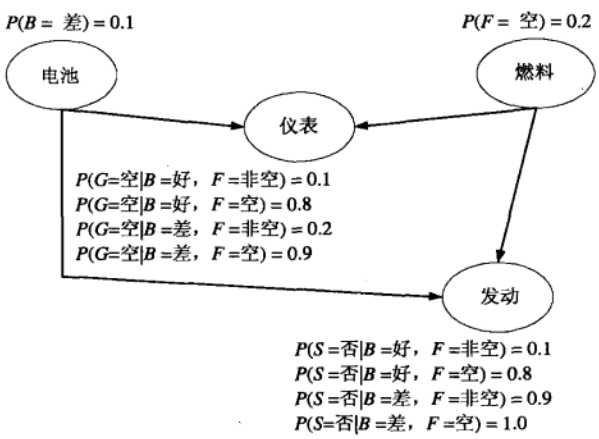


图 5-48　习题 12 的贝叶斯信念网络

13. 考虑表 5-13 中的一维数据集。

(a) 根据 1-最近邻、3-最近邻、5-最近邻及 9-最近邻，对数据点 $x = 5.0$ 分类（使用多数表决）。

(b) 使用 5.2.1 节中描述的距离加权表决方法重复前面的分析。

表 5-13　习题 13 的数据集

x	0.5	3.0	4.5	4.6	4.9	5.2	5.3	5.5	7.0	9.5
y	−	−	+	+	+	−	−	+	−	−

14. 5.2 节中描述的最近邻算法可以扩充以便处理标称属性。Cost 和 Salzberg[171]提出了一个最近邻算法的变形，称作 PEBLS（并行的基于实例的学习系统，Parallel Examplar-Based Learning System），它使用改进的值差度量（Modified Value Difference Metric，MVDM）

① B 表示电池，F 表示燃料，G 表示仪表，S 表示发动。——译者注

来度量一个标称属性的两个值之间的距离。给定一对标称属性值 V_1 和 V_2，二者之间的距离定义为：

$$d(V_1, V_2) = \sum_{i=1}^{k} \left| \frac{n_{i1}}{n_1} - \frac{n_{i2}}{n_2} \right| \tag{5-84}$$

其中，n_{ij} 是类 i 中具有属性值 V_j 的样例数，n_j 是具有属性值 V_j 的样例总数。

考虑图 5-9 中贷款分类问题的训练集。使用 MVDM 来计算属性*是否有房*和*婚姻状况*的每一对属性值之间的距离。

15. 对下面的每一个布尔函数，说出问题是否线性可分。
    (a) *A* AND *B* AND *C*
    (b) NOT *A* AND *B*
    (c) (*A* OR *B*) AND (*A* OR *C*)
    (d) (*A* XOR *B*) AND (*A* OR *B*)
16. (a) 说明感知器模型怎样表示两个布尔变量之间的 AND 和 OR 函数。
    (b) 评论使用线性函数作为多层神经网络的激活函数的缺点。
17. 请评价两个分类模型 M_1 和 M_2 的性能。所选择的测试集包含 26 个二值属性，记作 *A* 到 *Z*。

    表 5-14 是模型应用到测试集时得到的后验概率（图中只显示正类的后验概率）。因为这是二类问题，所以 $P(-) = 1 - P(+)$, $P(-|A,\cdots,Z) = 1 - P(+|A,\cdots,Z)$。假设需要从正类中检测实例。
    (a) 画出 M_1 和 M_2 的 ROC 曲线（画在同一幅图中）。哪个模型更好？给出理由。
    (b) 对模型 M_1，假设截止阈值 $t = 0.5$。换句话说，任何后验概率大于 t 的测试实例都被看作正例。计算模型在此阈值下的精度、召回率和 *F* 度量。
    (c) 对模型 M_2 使用相同的截止阈值重复(b)的分析。比较两个模型的 *F* 度量值，哪个模型更好？所得结果和从 ROC 曲线中得到的结论一致吗？
    (d) 使用阈值 $t = 0.1$ 对模型 M_1 重复(b)的分析。$t = 0.5$ 和 $t = 0.1$ 哪一个阈值更好？该结果和你从 ROC 曲线中得到的一致吗？

表 5-14　习题 17 的后验概率

实例	真实类	$P(+\|A,\cdots,Z,M_1)$	$P(+\|A,\cdots,Z,M_2)$
1	+	0.73	0.61
2	+	0.69	0.03
3	−	0.44	0.68
4	−	0.55	0.31
5	+	0.67	0.45
6	+	0.47	0.09
7	−	0.08	0.38
8	−	0.15	0.05
9	+	0.45	0.01
10	−	0.35	0.04

18. 下面的数据集包含两个属性 *X* 和 *Y*，两个类标号“+”和“−”。每个属性取三个不同的值：0，1 或 2。“+”类的概念是 $Y = 1$，“−”类的概念是 $X = 0 \vee X = 2$。

X	Y	实例数	
		+	−
0	0	0	100
1	0	0	0
2	0	0	100
0	1	10	100
1	1	10	0
2	1	10	100
0	2	0	100
1	2	0	0
2	2	0	100

(a) 建立该数据集的决策树。该决策树能捕捉到“+”和“−”的概念吗？

(b) 决策树的准确率、精度、召回率和F_1度量各是多少？（注意，精度、召回率和F_1度量均是对“+”类定义。）

(c) 使用下面的代价函数建立新的决策树：

$$C(i,j)=\begin{cases}0 & \text{如果}i=j\\ 1 & \text{如果}i=+,j=-\\ \dfrac{-\text{实例个数}}{+\text{实例个数}} & \text{如果}i=-,j=+\end{cases}$$

（提示：只需改变原决策树的叶结点。）新决策树能捕捉到“+”的概念吗？

(d) 新决策树的准确率、精度、召回率和F_1度量各是多少？

19. (a) 考虑两类问题的代价矩阵。设$C(+,+)=C(-,-)=p$，$C(+,-)=C(-,+)=q$，且$q>p$。证明：最小化代价函数等价于最大化分类器准确率。

(b) 证明：代价矩阵是比例不变量（scale-invariant）。例如，如果代价矩阵的大小进行伸缩$C(i,j)\rightarrow\beta C(i,j)$，$\beta$是伸缩因子，则决策阈值（公式（5-82））保持不变。

(c) 证明：代价矩阵是平移不变量（translation-invariant）。即代价矩阵的每一个元素都加上一个常量，不会影响决策阈值（公式（5-82））。

20. 考虑任务：为随机数据建立分类器，其中属性值随机产生，与类标号无关。假设数据集包含两个类“+”和“−”的记录。数据集的一半用于训练，而剩下的一半用于测试。

(a) 假设数据集中正例和反例的数目相等，决策树分类器把所有测试记录预测为正类。则分类器在测试数据上的期望误差率是多少？

(b) 假设分类器把每个测试记录预测为正类的概率是0.8，预测为负类的概率是0.2，重复前面的分析。

(c) 假设2/3的数据属于正类，1/3的数据属于负类。分类器把每个测试记录预测为正类的期望误差是多少？

(d) 假设分类器把每个测试记录预测为正类的概率是2/3，预测为负类的概率是1/3，重复前面的分析。

21. 导出不可分数据的线性SVM的对偶拉格朗日函数（dual Lagrangian），其中目标函数是：

$$f(\mathbf{w})=\frac{\|\mathbf{w}\|^2}{2}+C\left(\sum_{i=1}^{N}\xi_i\right)^2$$

22. 考虑XOR问题，其中有四个训练点：

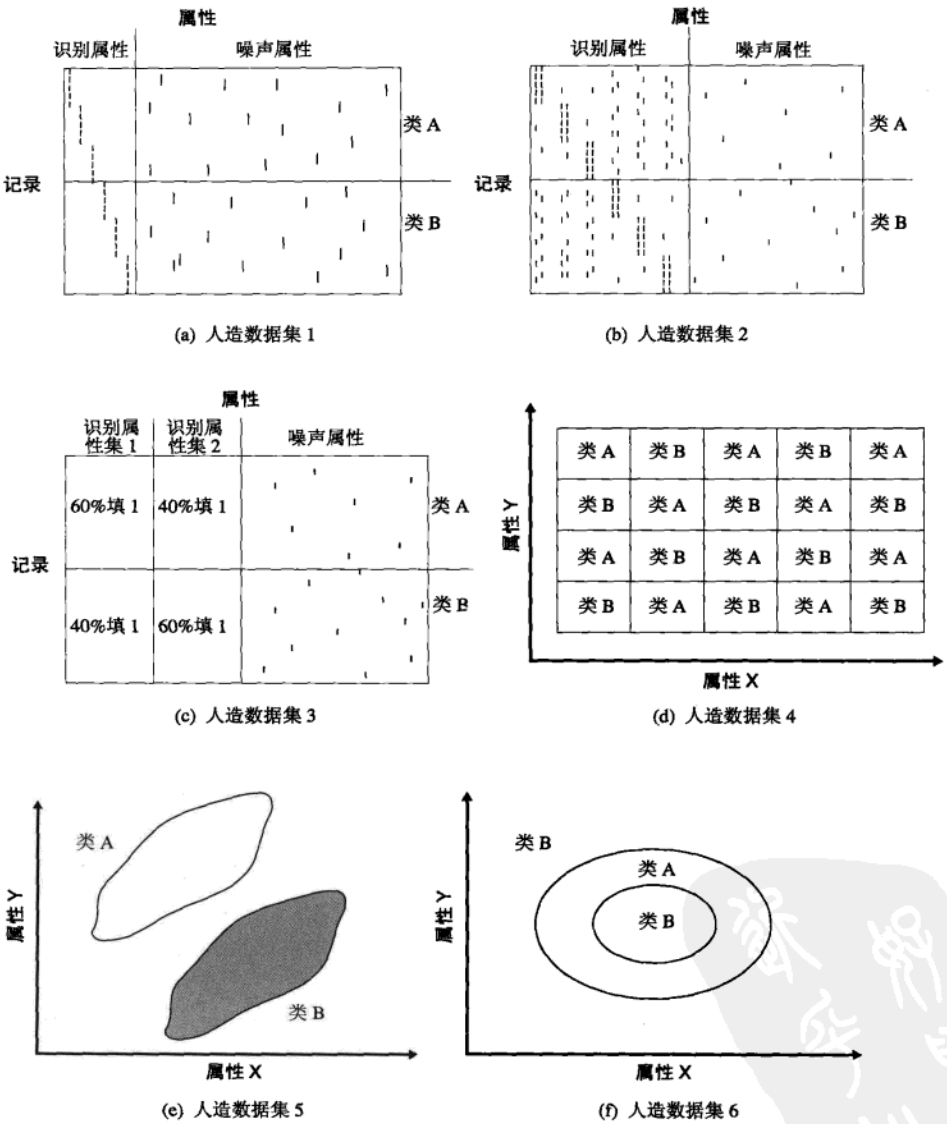
$$(1,1,-),\quad (1,0,+),(0,1,+),(0,0,-)$$

把数据转化为下面的特征空间：

$$\Phi=(1,\sqrt{2}x_1,\sqrt{2}x_2,\sqrt{2}x_1x_2,x_1^2,x_2^2)$$

找出转化后空间的最大边缘线性决策边界。

23. 给定图 5-49 所示的数据集，解释在此数据集上，决策树、朴素贝叶斯和 k-最近邻分类器是怎样工作的。



(a) 人造数据集 1　(b) 人造数据集 2

(c) 人造数据集 3　(d) 人造数据集 4

(e) 人造数据集 5　(f) 人造数据集 6

图 5-49　习题 23 的数据集

第6章

关联分析：基本概念和算法

许多商业企业在日复一日的运营中积聚了大量的数据。例如，食品商店的收银台每天都收集大量的顾客购物数据。表 6-1 给出一个这种数据的例子，通常称作**购物篮事务**（market basket transaction）。表中每一行对应一个事务，包含一个唯一标识 TID 和给定顾客购买的商品的集合。零售商对分析这些数据很感兴趣，以便了解他们的顾客的购买行为。可以使用这种有价值的信息来支持各种商务应用，如市场促销，库存管理和顾客关系管理等。

表 6-1　购物篮事务的例子

TID	项　集
1	{面包，牛奶}
2	{面包，尿布，啤酒，鸡蛋}
3	{牛奶，尿布，啤酒，可乐}
4	{面包，牛奶，尿布，啤酒}
5	{面包，牛奶，尿布，可乐}

本章主要是介绍一种称作**关联分析**（association analysis）的方法，用于发现隐藏在大型数据集中的有意义的联系。所发现的联系可以用关联规则（association rule）或频繁项集的形式表示。例如，从表 6-1 所示的数据中可以提取出如下规则：

$$\{尿布\} \rightarrow \{啤酒\}$$

该规则表明尿布和啤酒的销售之间存在着很强的联系，因为许多购买尿布的顾客也购买啤酒。零售商们可以使用这类规则，帮助他们发现新的交叉销售商机。

除了购物篮数据外，关联分析也可以应用于其他领域，如生物信息学、医疗诊断、网页挖掘和科学数据分析等。例如，在地球科学数据分析中，关联模式可以揭示海洋、陆地和大气过程之间的有趣联系。这样的信息能够帮助地球科学家更好地理解地球系统中不同的自然力之间的相互作用。尽管这里提供的技术一般可以都用于更广泛的数据集，但是为了便于解释，讨论将主要集中在购物篮数据上。

在对购物篮数据进行关联分析时，需要处理两个关键的问题：第一，从大型事务数据集中发现模式可能在计算上要付出很高的代价；第二，所发现的某些模式可能是虚假的，因为它们可能是偶然发生的。本章的其余部分主要是围绕这两个问题组织。本章的第一部分解释关联分析的基本概念和用来有效地挖掘这种模式的算法。第二部分处理发现模式的评估问题，以避免产生虚假结果。

6.1　问题定义

这一节讲述关联分析中使用的基本术语，并提供该任务的形式化描述。

二元表示　购物篮数据可以用表 6-2 所示的二元形式来表示，其中每行对应一个事务，而每列对应一个项。项可以用二元变量表示，如果项在事务中出现，则它的值为 1，否则为 0。因为通常认为项在事务中出现比不出现更重要，因此项是**非对称**（asymmetric）二元变量。或许这种表示是实际购物篮数据极其简单的展现，因为这种表示忽略数据的某些重要的方面，如所购商品的数量和价格等。处理这种非二元数据的方法将在第 7 章讨论。

表 6-2　购物篮数据的二元 0/1 表示

TID	面　包	牛　奶	尿　布	啤　酒	鸡　蛋	可　乐
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

项集和支持度计数　令 $I=\{i_1, i_2, \cdots, i_d\}$是购物篮数据中所有项的集合，而 $T=\{t_1, t_2, \cdots, t_N\}$是所有事务的集合。每个事务 t_i包含的项集都是 I 的子集。在关联分析中，包含 0 个或多个项的集合被称为项集（itemset）。如果一个项集包含 k 个项，则称它为 k-项集。例如，{啤酒，尿布，牛奶}是一个 3-项集。空集是指不包含任何项的项集。

事务的宽度定义为事务中出现项的个数。如果项集 X 是事务 t_j的子集，则称事务 t_j包括项集 X。例如，在表 6-2 中第二个事务包括项集{面包，尿布}，但不包括项集{面包，牛奶}。项集的一个重要性质是它的支持度计数，即包含特定项集的事务个数。数学上，项集 X 的支持度计数$\sigma(X)$可以表示为：

$$\sigma(X)=|\{t_i \mid X \subseteq t_i, t_i \in T\}|$$

其中，符号$|\cdot|$表示集合中元素的个数。在表 6-2 显示的数据集中，项集{啤酒，尿布，牛奶}的支持度计数为 2，因为只有 2 个事务同时包含这 3 个项。

关联规则（association rule）　关联规则是形如 $X \rightarrow Y$ 的蕴涵表达式，其中 X 和 Y 是不相交的项集，即 $X \cap Y=\varnothing$。关联规则的强度可以用它的**支持度**（support）和**置信度**（confidence）度量。支持度确定规则可以用于给定数据集的频繁程度，而置信度确定 Y 在包含 X 的事务中出现的频繁程度。支持度（s）和置信度（c）这两种度量的形式定义如下：

$$s(X \rightarrow Y)=\frac{\sigma(X \cup Y)}{N} \tag{6-1}$$

$$c(X \rightarrow Y)=\frac{\sigma(X \cup Y)}{\sigma(X)} \tag{6-2}$$

例 6.1　考虑规则{牛奶，尿布}→{啤酒}。由于项集{牛奶，尿布，啤酒}的支持度计数是 2，而事务的总数是 5，所以规则的支持度为 2/5 = 0.4。规则的置信度是项集{牛奶，尿布，啤酒}的支持度计数与项集{牛奶，尿布}支持度计数的商。由于存在 3 个事务同时包含牛奶和尿布，所以该规则的置信度为 2/3 = 0.67。　□

为什么使用支持度和置信度？ 支持度是一种重要度量，因为支持度很低的规则可能只是偶然出现。从商务角度来看，低支持度的规则多半也是无意义的，因为对顾客很少同时购买的商品进行促销可能并无益处（6.8 节讨论的情况则是例外）。因此，支持度通常用来删去那些无意义的规则。此外，正如 6.2.1 节所示，支持度还具有一种期望的性质，可以用于关联规则的有效发现。

另一方面，置信度度量通过规则进行推理具有可靠性。对于给定的规则 $X \rightarrow Y$，置信度越高，Y 在包含 X 的事务中出现的可能性就越大。置信度也可以估计 Y 在给定 X 下的条件概率。

应当小心解释关联分析的结果。由关联规则作出的推论并不必然蕴涵因果关系。它只表示规则前件和后件中的项明显地同时出现。另一方面，因果关系需要关于数据中原因和结果属性的知识，并且通常涉及长期出现的联系（例如，臭氧损耗导致全球变暖）。

关联规则挖掘问题的形式描述 关联规则的挖掘问题可以形式地描述如下：

定义 6.1 关联规则发现 给定事务的集合 T，关联规则发现是指找出支持度大于等于 *minsup* 并且置信度大于等于 *minconf* 的所有规则，其中 *minsup* 和 *minconf* 是对应的支持度和置信度阈值。

挖掘关联规则的一种原始方法是：计算每个可能规则的支持度和置信度。但是这种方法的代价很高，令人望而却步，因为可以从数据集提取的规则的数目达指数级。更具体地说，从包含 d 个项的数据集提取的可能规则的总数为：

$$R = 3^d - 2^{d+1} + 1 \tag{6-3}$$

此式的证明作为习题留给读者（见本章习题 5）。即使对于表 6-1 所示的小数据集，这种方法也需要计算 $3^6 - 2^7 + 1 = 602$ 条规则的支持度和置信度。使用 $minsup = 20\%$和 $minconf = 50\%$，80%以上的规则将被丢弃，使得大部分计算是无用的开销。为了避免进行不必要的计算，事先对规则剪枝，而无须计算它们的支持度和置信度的值将是有益的。

提高关联规则挖掘算法性能的第一步是拆分支持度和置信度要求。由公式（6-1）可以看出，规则 $X \rightarrow Y$ 的支持度仅依赖于其对应项集 $X \cup Y$ 的支持度。例如，下面的规则有相同的支持度，因为它们涉及的项都源自同一个项集{啤酒，尿布，牛奶}：

{啤酒，尿布}→{牛奶}，　{啤酒，牛奶}→{尿布}，
{尿布，牛奶}→{啤酒}，　{啤酒}→{尿布，牛奶}，
{牛奶}→{啤酒，尿布}，　{尿布}→{啤酒，牛奶}

如果项集{啤酒，尿布，牛奶}是非频繁的，则可以立即剪掉这 6 个候选规则，而不必计算它们的置信度值。

因此，大多数关联规则挖掘算法通常采用的一种策略是，将关联规则挖掘任务分解为如下两个主要的子任务。

(1) **频繁项集产生**：其目标是发现满足最小支持度阈值的所有项集，这些项集称作频繁项集（frequent itemset）。

(2) **规则的产生**：其目标是从上一步发现的频繁项集中提取所有高置信度的规则，这些规则称作强规则（strong rule）。

通常，频繁项集产生所需的计算开销远大于产生规则所需的计算开销。频繁项集和关联规产

生的有效技术将分别在 6.2 节和 6.3 节讨论。

6.2　频繁项集的产生

格结构（lattice structure）常常被用来枚举所有可能的项集。图6-1显示 $I=\{a, b, c, d, e\}$的项集格。一般来说，一个包含 k 个项的数据集可能产生2^k-1个频繁项集，不包括空集在内。由于在许多实际应用中 k 的值可能非常大，需要探查的项集搜索空间可能是指数规模的。

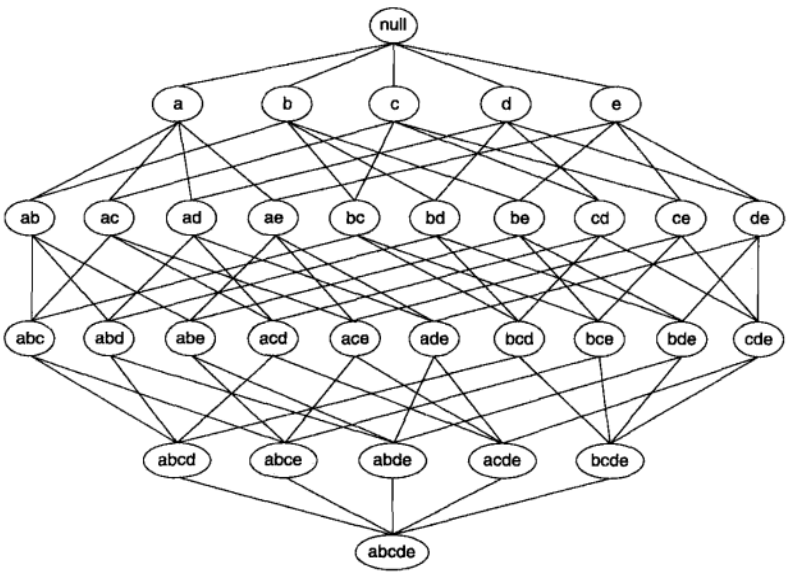


图 6-1　项集的格

发现频繁项集的一种原始方法是确定格结构中每个**候选项集**（candidate itemset）的支持度计数。为了完成这一任务，必须将每个候选项集与每个事务进行比较，如图 6-2 所示。如果候选项集包含在事务中，则候选项集的支持度计数增加。例如，由于项集{面包，牛奶}出现在事务 1，4 和 5 中，其支持度计数将增加 3 次。这种方法的开销可能非常大，因为它需要进行 $O(NMw)$次比较，其中 N 是事务数，$M=2^k-1$ 是候选项集数，而 w 是事务的最大宽度。

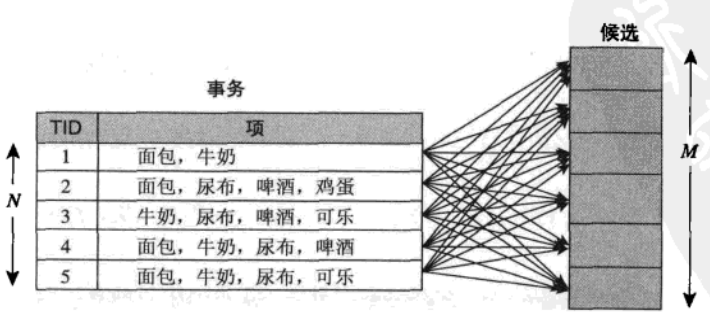


图 6-2　计算候选项集的支持度

有几种方法可以降低产生频繁项集的计算复杂度。

(1) **减少候选项集的数目**（M）。下一节介绍的先验（apriori）原理，是一种不用计算支持度值而删除某些候选项集的有效方法。

(2) **减少比较次数**。替代将每个候选项集与每个事务相匹配，可以使用更高级的数据结构，或者存储候选项集或者压缩数据集，来减少比较次数。这些策略将在 6.2.4 节和 6.6 节讨论。

6.2.1 先验原理

本节描述如何使用支持度度量，帮助减少频繁项集产生时需要探查的候选项集个数。使用支持度对候选项集剪枝基于如下原理。

定理 6.1 先验原理 如果一个项集是频繁的，则它的所有子集一定也是频繁的。

为了解释先验原理的基本思想，考虑图 6-3 所示的项集格。假定{c, d, e}是频繁项集。显而易见，任何包含项集{c, d, e}的事务一定包含它的子集{c, d}, {c, e}, {d, e}, {c}, {d}和{e}。这样，如果{c, d, e}是频繁的，则它的所有子集（图 6-3 中的阴影项集）一定也是频繁的。

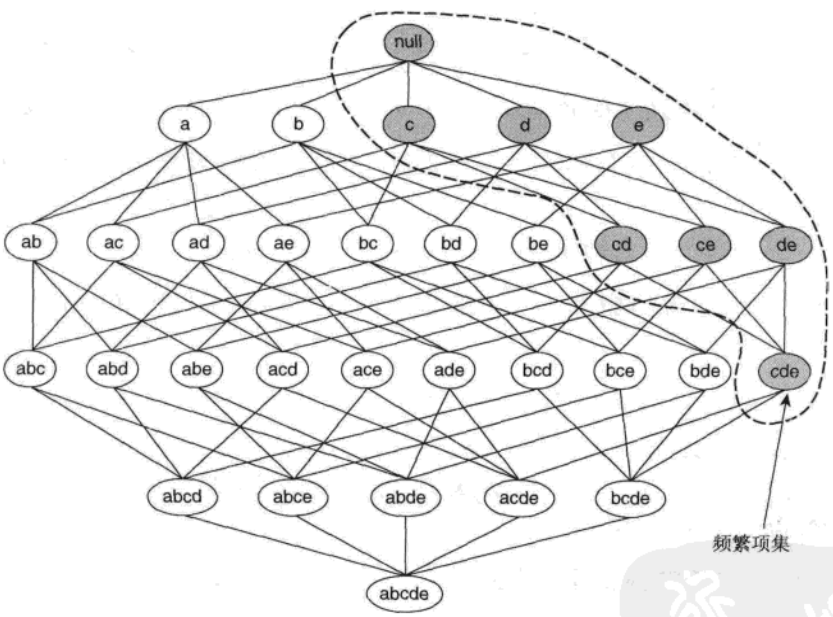


图 6-3 先验原理的图示。如果{c, d, e}是频繁的，则它的所有子集也是频繁的

相反，如果项集{a, b}是非频繁的，则它的所有超集也一定是非频繁的。如图 6-4 所示，一旦发现{a, b}是非频繁的，则整个包含{a, b}超集的子图可以被立即剪枝。这种基于支持度度量修剪指数搜索空间的策略称为**基于支持度的剪枝**（support-based pruning）。这种剪枝策略依赖于支持度度量的一个关键性质，即一个项集的支持度决不会超过它的子集的支持度。这个性质也称支持度度量的**反单调性**（anti-monotone）。

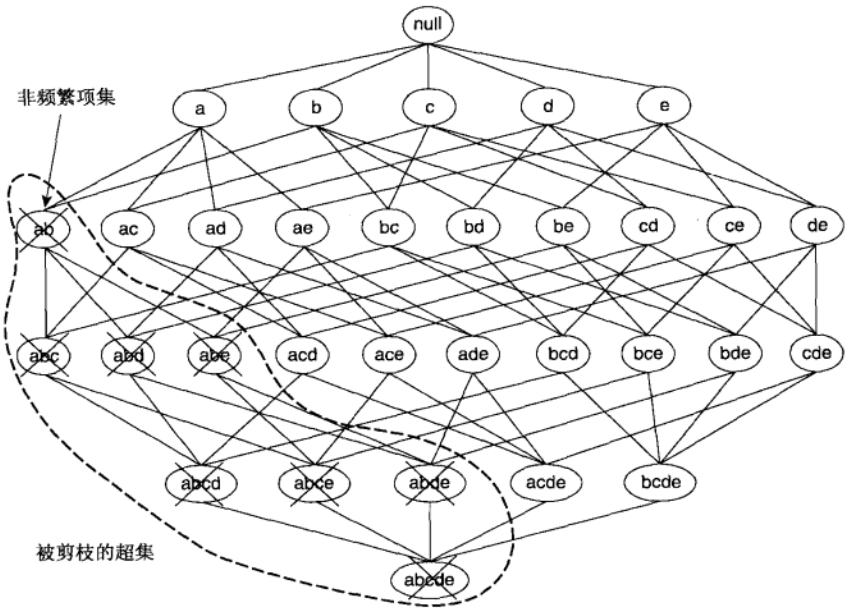


图 6-4　基于支持度的剪枝的图示。如果{a,b}是非频繁的，则它的所有超集也是非频繁的

定义 6.2　单调性　令 I 是项的集合，$J=2^I$ 是 I 的幂集。度量 f 是单调的（或向上封闭的），如果

$$\forall X,Y\in J:(X\subseteq Y)\rightarrow f(X)\leqslant f(Y)$$

这表明如果 X 是 Y 的子集，则 $f(X)$ 一定不超过 $f(Y)$。另一方面，f 是反单调的（或向下封闭的），如果

$$\forall X,Y\in J:(X\subseteq Y)\rightarrow f(Y)\leqslant f(X)$$

表示如果 X 是 Y 的子集，则 $f(Y)$ 一定不超过 $f(X)$。

正如下节所述，任何具有反单调性的度量都能够直接结合到挖掘算法中，可以对候选项集的指数搜索空间进行有效地剪枝。

6.2.2　*Apriori* 算法的频繁项集产生

*Apriori*算法是第一个关联规则挖掘算法，它开创性地使用基于支持度的剪枝技术，系统地控制候选项集指数增长。对于表 6-1 中所示的事务，图 6-5 给出*Apriori*算法频繁项集产生部分的一个高层实例。假定支持度阈值是 60%，相当于最小支持度计数为 3。

初始时每个项都被看作候选1-项集。对它们的支持度计数之后，候选项集{可乐}和{鸡蛋}被丢弃，因为它们出现的事务少于3个。在下一次迭代，仅使用频繁1-项集来产生候选2-项集，因为先验原理保证所有非频繁的1-项集的超集都是非频繁的。由于只有4个频繁1-项集，因此算法产生的候选2-项集的数目为 $C_4^2=6$。计算它们的支持度值之后，发现这6个候选项集中的2个，{啤酒，面包}和{啤酒，牛奶}是非频繁的。剩下的4个候选项集是频繁的，因此用来产生候选3-项集。不使

用基于支持度的剪枝，使用该例给定的6个项，将形成 $C_6^3 = 20$ 个候选3-项集。依据先验原理，只需要保留其子集都频繁的候选3-项集。具有这种性质的唯一候选是{面包，尿布，牛奶}。

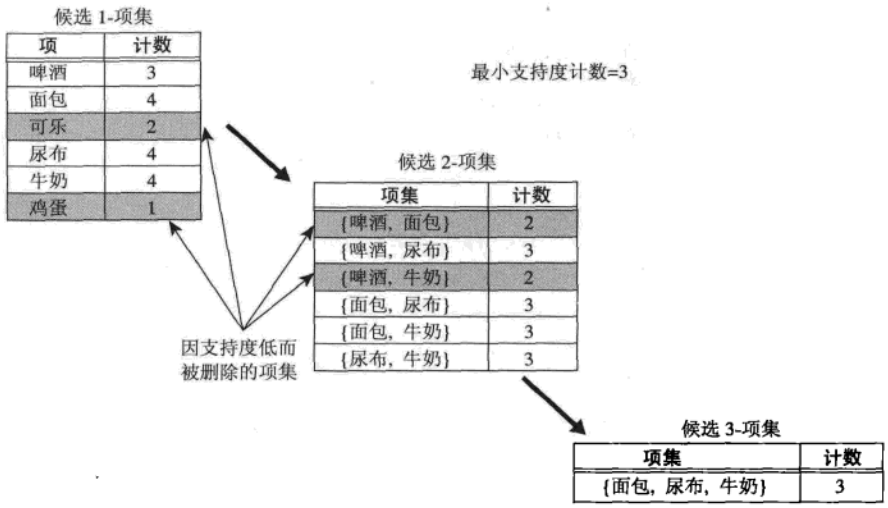


图 6-5 使用 *Apriori* 算法产生频繁项集的例子

通过计算产生的候选项集数目，可以看出先验剪枝策略的有效性。枚举所有项集（到 3-项集）的蛮力策略将产生 $C_6^1 + C_6^2 + C_6^3 = 6 + 15 + 20 = 41$ 个候选；而使用先验原理，将减少为 $C_6^1 + C_4^2 + 1 = 6 + 6 + 1 = 13$ 个候选。甚至在这个简单的例子中，候选项集的数目也降低了 68%。

算法 6.1 中给出了 *Apriori* 算法产生频繁项集部分的伪代码。令 C_k 为候选 k-项集的集合，而 F_k 为频繁 k-项集的集合：

- 该算法初始通过单遍扫描数据集，确定每个项的支持度。一旦完成这一步，就得到所有频繁 1-项集的集合 F_1（步骤 1 和步骤 2）。
- 接下来，该算法将使用上一次迭代发现的频繁(k–1)-项集，产生新的候选 k-项集（步骤 5）。候选的产生使用 apriori-gen 函数实现，将在 6.2.3 节介绍。
- 为了对候选项的支持度计数，算法需要再次扫描一遍数据集（步骤 6~10）。使用子集函数确定包含在每一个事务 t 中的 C_k 中的所有候选 k-项集。子集函数的实现在 6.2.4 节介绍。
- 计算候选项的支持度计数之后，算法将删去支持度计数小于 *minsup* 的所有候选项集（步骤 12）。
- 当没有新的频繁项集产生，即 $F_k = \varnothing$ 时，算法结束（步骤 13）。

*Apriori*算法的频繁项集产生的部分有两个重要的特点：第一，它是一个**逐层**（level-wise）算法，即从频繁1-项集到最长的频繁项集，它每次遍历项集格中的一层；第二，它使用**产生–测试**（generate-and-test）策略来发现频繁项集。在每次迭代之后，新的候选项集都由前一次迭代发现的频繁项集产生，然后对每个候选的支持度进行计数，并与最小支持度阈值进行比较。该算法需要的总迭代次数是 k_{max}+1，其中 k_{max} 是频繁项集的最大长度。

算法 6.1　*Apriori* 算法的频繁项集产生

```
1:  k = 1
2:  F_k = {i | i∈I ∧ σ({i}) ≥ N × minsup}      {发现所有的频繁 1-项集}
3:  repeat
4:    k = k+1
5:    C_k = apriori-gen(F_{k-1})      {产生候选项集}
6:    for 每个事务 t∈T do
7:      C_t = subset(C_k, t)      {识别属于 t 的所有候选}
8:      for 每个候选项集 c∈C_t do
9:        σ(c) = σ(c) + 1      {支持度计数增值}
10:     end for
11:   end for
12:   F_k = {c | c∈C_k ∧ σ(c) ≥ N × minsup}      {提取频繁 k-项集}
13: until F_k = ∅
14: Result = ∪F_k
```

6.2.3　候选的产生与剪枝

算法 6.1 步骤 5 的 apriori-gen 函数通过如下两个操作产生候选项集。

(1) **候选项集的产生**。该操作由前一次迭代发现的频繁$(k-1)$-项集产生新的候选 k-项集。

(2) **候选项集的剪枝**。该操作采用基于支持度的剪枝策略，删除一些候选 k-项集。

为了解释候选项集剪枝操作，考虑候选 k-项集 $X=\{i_1, i_2, \cdots, i_k\}$。算法必须确定它的所有真子集 $X-\{i_j\}(\forall j=1, 2, \cdots, k)$是否都是频繁的，如果其中一个是非频繁的，则 X 将会被立即剪枝。这种方法能够有效地减少支持度计数过程中所考虑的候选项集的数量。对于每一个候选 k-项集，该操作的复杂度是 $O(k)$。然而，随后我们将明白，并不需要检查给定候选项集的所有 k 个子集。如果 k 个子集中的 m 个用来产生候选项集，则在候选项集剪枝时只需要检查剩下的 $k-m$ 个子集。

理论上，存在许多产生候选项集的方法。下面列出了对有效的候选项集产生过程的要求。

(1) 它应当避免产生太多不必要的候选。一个候选项集是不必要的，如果它至少有一个子集是非频繁的。根据支持度的反单调属性，这样的候选项集肯定是非频繁的。

(2) 它必须确保候选项集的集合是完全的，即候选项集产生过程没有遗漏任何频繁项集。为了确保完全性，候选项集的集合必须包含所有频繁项集的集合，即$\forall k: F_k \subseteq C_k$。

(3) 它应该不会产生重复候选项集。例如：候选项集$\{a, b, c, d\}$可能会通过多种方法产生，如合并$\{a, b, c\}$和$\{d\}$，合并$\{b, d\}$和$\{a, c\}$，合并$\{c\}$和$\{a, b, d\}$等。候选项集的重复产生将会导致计算的浪费，因此为了效率应该避免。

接下来，将简要地介绍几种候选产生过程，其中包括 apriori-gen 函数使用的方法。

蛮力方法　蛮力方法把所有的 k-项集都看作可能的候选，然后使用候选剪枝除去不必要的候选（见图 6-6）。第 k 层产生的候选项集的数目为 C_d^k，其中 d 是项的总数。虽然候选产生是相当简单的，但是候选剪枝的开销极大，因为必须考察的项集数量太大。设每一个候选项集所需的计算量为 $O(k)$，这种方法的总复杂度为 $O\left(\sum_{k=1}^{d} kC_d^k\right)=O(d\cdot 2^{d-1})$。

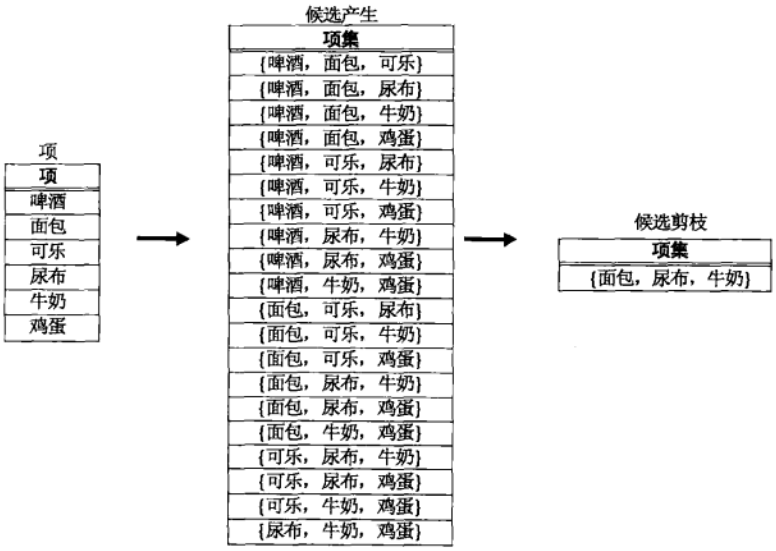


图 6-6 产生候选 3-项集的蛮力方法

$F_{k-1}\times F_1$ 方法 另一种的产生候选项集的方法是用其他频繁项来扩展每个频繁(k–1)-项集。图 6-7 显示了如何用频繁项（如面包）扩展频繁 2-项集{啤酒，尿布}，产生候选 3-项集{啤酒，尿布，面包}。这种方法将产生 $O(|F_{k-1}|\times|F_1|)$个候选 k-项集，其中$|F_j|$ 表示频繁 j-项集的个数。这种方法总复杂度是 $O(\Sigma_k k|F_{k-1}||F_1|)$。

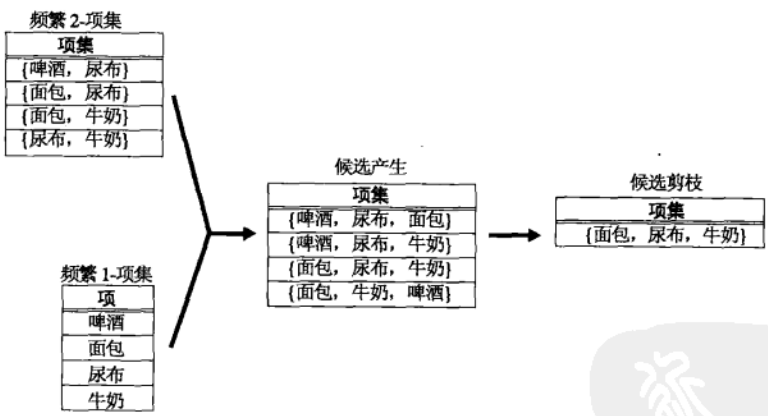


图 6-7 通过合并频繁(k–1)-项集和频繁 1-项集生成和剪枝候选 k-项集。
注意：某些候选是不必要，因为它们的子集是非频繁的

这种方法是完备的，因为每一个频繁 k-项集都是由一个频繁(k–1)-项集和一个频繁 1-项集组成的。因此，所有的频繁 k-项集是这种方法所产生的候选 k-项集的一部分。然而，这种方法很难避免重复地产生候选项集。例如，项集{面包，尿布，牛奶}不仅可以由合并项集{面包，尿布}和{牛奶}得到，而且还可以由合并{面包，牛奶}和{尿布}得到，或者由合并{尿布，牛奶}和{面包}得到。避免产生重复的候选项集的一种方法是确保每个频繁项集中的项以字典序存储，每个

频繁$(k-1)$-项集 X 只用字典序比 X 中所有的项都大的频繁项进行扩展。例如，项集{面包，尿布}可以用项集{牛奶}扩展，因为“牛奶”（Milk）在字典序下比“面包”（Bread）和“尿布”（Diapers）都大。然而，不应当用{面包}扩展{尿布，牛奶}或用{尿布}扩展{面包，牛奶}，因为它们违反了字典序条件。

尽管这种方法比蛮力方法有明显改进，但是仍会产生大量不必要的候选。例如，通过合并{啤酒，尿布}和{牛奶}而得到的候选是不必要的，因为它的一个子集{啤酒，牛奶}是非频繁的。有几种启发式方法能够减少不必要的候选数量。例如，对于每一个幸免于剪枝的候选 k-项集，它的每一个项必须至少在 $k-1$个$(k-1)$-项集中出现，否则，该候选就是非频繁的。例如，项集{啤酒，尿布，牛奶}是一个可行的候选3-项集，仅当它的每一个项（包括“啤酒”）都必须在两个频繁2-项集中出现。由于只有一个频繁2-项集包含“啤酒”，因此所有包含“啤酒”的候选都是非频繁的。

$F_{k-1}\times F_{k-1}$ 方法　函数 apriori-gen 的候选产生过程合并一对频繁$(k-1)$-项集，仅当它们的前 $k-2$ 个项都相同。令 $A=\{a_1, a_2, \cdots, a_{k-1}\}$和 $B=\{b_1, b_2, \cdots, b_{k-1}\}$是一对频繁$(k-1)$-项集，合并 A 和 B，如果它们满足如下条件：

$$a_i = b_i \quad (i=1, 2, \cdots, k-2) \text{ 并且 } a_{k-1} \neq b_{k-1}$$

在图 6-8 中，频繁项集{面包，尿布}和{面包，牛奶}合并，形成了候选 3-项集{面包，尿布，牛奶}。算法不会合并项集{啤酒，尿布}和{尿布，牛奶}，因为它们的第一个项不相同。实际上，如果{啤酒，尿布，牛奶}是可行的候选，则它应当由{啤酒，尿布}和{啤酒，牛奶}合并而得到。这个例子表明了候选项产生过程的完全性和使用字典序避免重复的候选的优点。然而，由于每个候选都由一对频繁$(k-1)$-项集合并而成，因此需要附加的候选剪枝步骤来确保该候选的其余$k-2$个子集是频繁的。

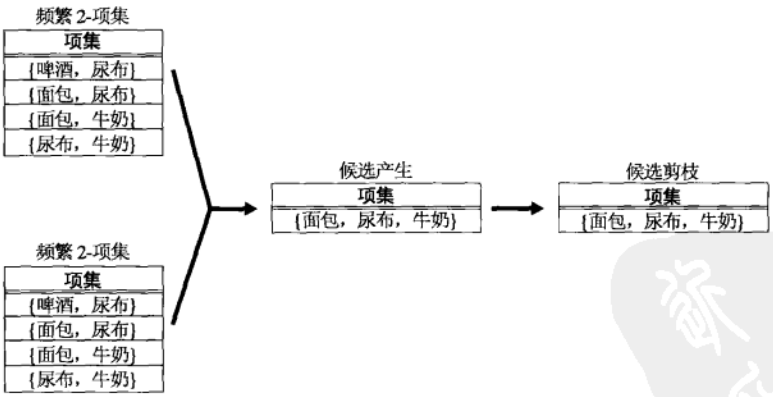


图 6-8　通过合并一对频繁$(k-1)$-项集生成和剪枝候选 k-项集

6.2.4　支持度计数

支持度计数过程确定在apriori-gen函数的候选项剪枝步骤保留下来的每个候选项集出现的频繁程度。支持度计数在算法 6.1 的第 6 步到第 11 步实现。支持度计数的一种方法是，将每个事

务与所有的候选项集进行比较（见图 6-2），并且更新包含在事务中的候选项集的支持度计数。这种方法是计算昂贵的，尤其当事务和候选项集的数目都很大时。

另一种方法是枚举每个事务所包含的项集，并且利用它们更新对应的候选项集的支持度。例如，考虑事务 t，它包含 5 个项{1, 2, 3, 5, 6}。该事务包含 $C_5^3=10$ 个大小为 3 的项集，其中的某些项集可能对应于所考察的候选 3-项集，在这种情况下，增加它们的支持度。那些不与任何候选项集对应的事务 t 的子集可以忽略。

图 6-9 显示了枚举事务 t 中所有 3-项集的系统的方法。假定每个项集中的项都以递增的字典序排列，则项集可以这样枚举：先指定最小项，其后跟随较大的项。例如，给定 $t=\{1,2,3,5,6\}$，它的所有 3-项集一定以项 1、2 或 3 开始。不必构造以 5 或 6 开始的 3-项集，因为事务 t 中只有两个项的标号大于等于 5。图 6-9 中第一层的前缀结构描述了指定包含在事务 t 中的 3-项集的第一项的方法。例如，1 2 3 5 6 表示这样的 3-项集，它以 1 开始，后随两个取自集合{2, 3, 5, 6}的项。

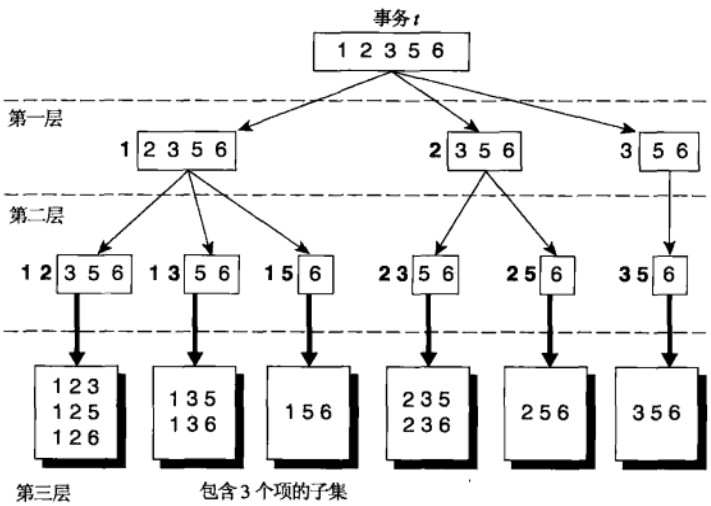


图 6-9 枚举事务 t 的所有包含 3 个项的子集

确定了第一项之后，第二层的前缀结构表示选择第二项的方法。例如：1 2 3 5 6 表示以{1,2}为前缀，后随项 3、5 或 6 的项集。最后，第三层的前缀结构显示了事务 t 包含的所有的 3-项集。例如，以{1, 2}为前缀的 3-项集是{1, 2, 3}，{1, 2, 5}和{1, 2, 6}；而以{2, 3}为前缀的 3-项集是{2, 3, 5}和{2, 3, 6}。

图 6-9 中所示的前缀结构演示了如何系统地枚举事务所包含的项集，即通过从最左项到最右项依次指定项集的项。然而还必须确定每一个枚举的 3-项集是否对应于一个候选项集，如果它与一个候选匹配，则相应候选项集的支持度计数增值。下面，将解释如何使用 Hash 树来有效地进行匹配操作。

使用 Hash 树进行支持度计数

在 *Apriori* 算法中，候选项集划分为不同的桶，并存放在 Hash 树中。在支持度计数期间，包含在事务中的项集也散列到相应的桶中。这种方法不是将事务中的每个项集与所有的候选项集进行比较，而是将它与同一桶内候选项集进行匹配，如图 6-10 所示。

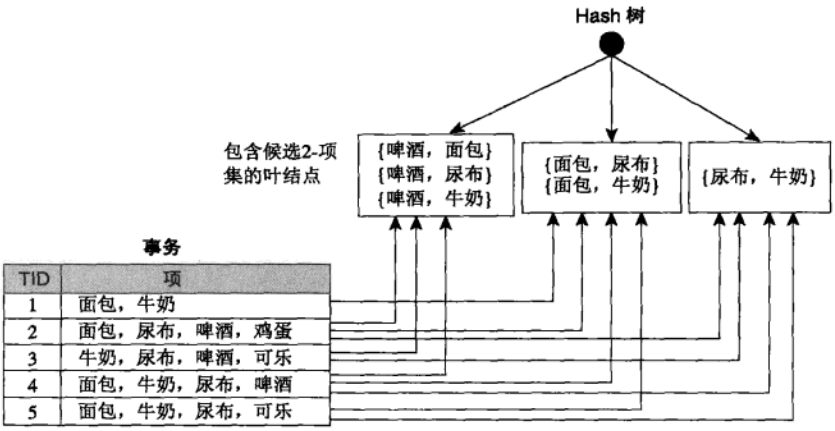


图 6-10　使用 Hash 树结构的项集支持度计数

图6-11是一棵 Hash 树结构的例子。树的每个内部结点都使用 Hash 函数 $h(p) = p \bmod 3$来确定应当沿着当前结点的哪个分支向下。例如，项1，4和7应当散列到相同的分支（即最左分支），因为除以3之后它们都具有相同的余数。所有的候选项集都存放在 Hash 树的叶结点中。图6-11中显示的 Hash 树包含15个候选3-项集，分布在9个叶结点中。

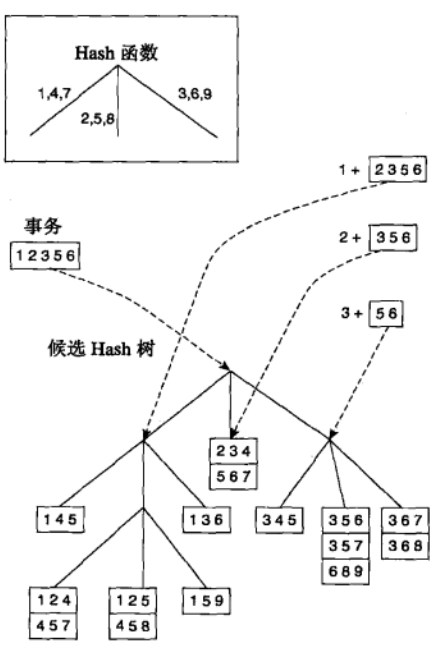


图 6-11　在 Hash 树的根结点散列一个事务

考虑一个事务 $t = \{1, 2, 3, 5, 6\}$。为了更新候选项集的支持度计数，必须这样遍历 Hash 树：所有包含属于事务 t 的候选 3-项集的叶结点至少访问一次。注意，包含在 t 中的候选 3-项集必须

以项 1，2 或 3 开始，如图 6-9 中第一层前缀结构所示。这样，在 Hash 树的根结点，事务中的项 1，2 和 3 将分别散列。项 1 被散列到根结点的左子女，项 2 被散列到中间子女，而项 3 被散列到右子女。在树的下一层，事务根据图 6-9 中的第二层结构列出的第二项进行散列。例如，在根结点散列项 1 之后，散列事务的项 2、3 和 5。项 2 和 5 散列到中间子女，而 3 散列到右子女，如图 6-12 所示。继续该过程，直至到达 Hash 树的叶结点。存放在被访问的叶结点中的候选项集与事务进行比较，如果候选项集是该事务的子集，则增加它的支持度计数。在这个例子中，访问了 9 个叶结点中的 5 个，15 个项集中的 9 个与事务进行比较。

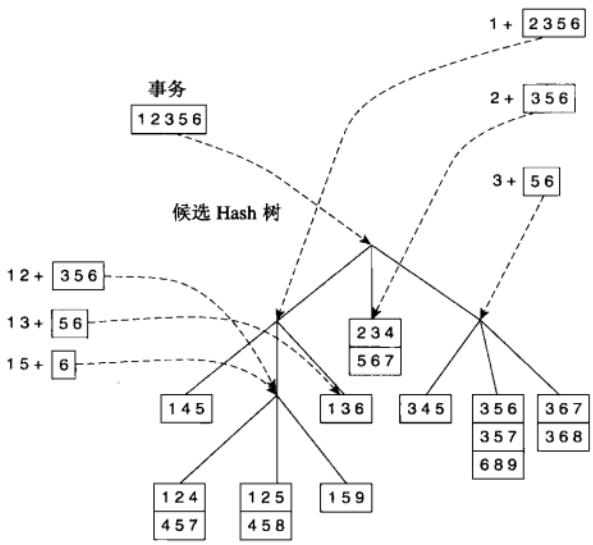


图 6-12 在候选项集 Hash 树根的最左子树上的子集操作

6.2.5 计算复杂度

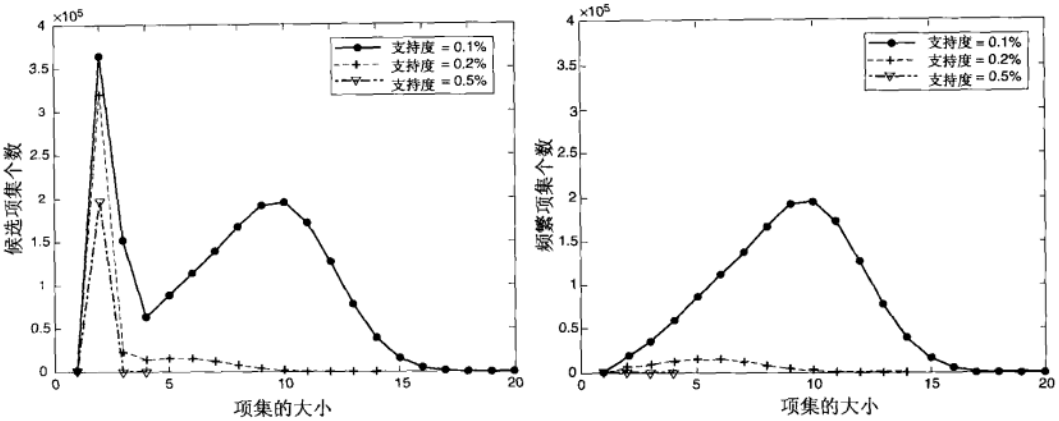
Apriori 算法的计算复杂度受如下因素影响。

支持度阈值 降低支持度阈值通常将导致更多的频繁项集。这给算法的计算复杂度带来不利影响，因为必须产生更多候选项集并对其计数，如图 6-13 所示。随着支持度阈值的降低，频繁项集的最大长度将增加。而随着频繁项集最大长度的增加，算法需要扫描数据集的次数也将增多。

项数（维度） 随着项数的增加，需要更多的空间来存储项的支持度计数。如果频繁项集的数目也随着数据维度增加而增长，则由于算法产生的候选项集更多，计算量和 I/O 开销将增加。

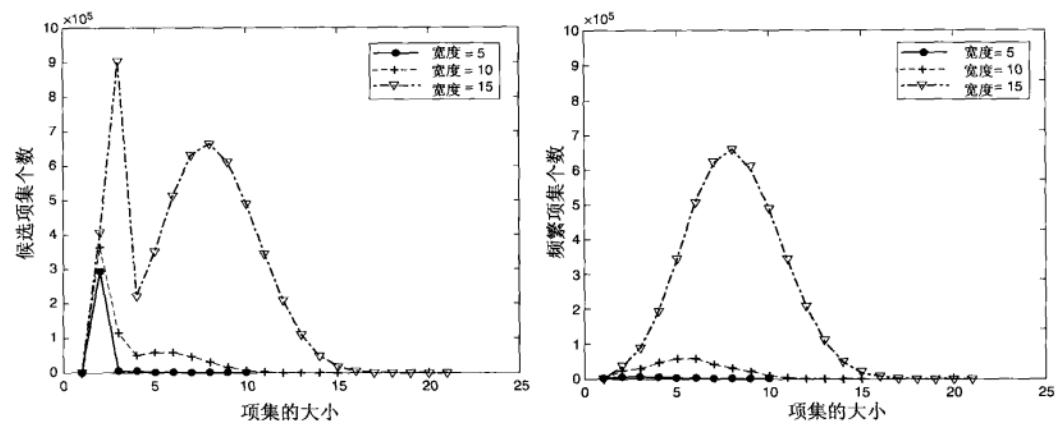
事务数 由于 *Apriori* 算法反复扫描数据集，因此它的运行时间随着事务数增加而增加。

事务的平均宽度 对于密集数据集，事务的平均宽度可能很大，这将在两个方面影响 *Apriori* 算法的复杂度。首先，频繁项集的最大长度随事务平均宽度增加而增加，因而，在候选项产生和支持度计数时必须考察更多候选项集，如图 6-14 所示；其次，随着事务宽度的增加，事务中将包含更多的项集，这将增加支持度计数时 Hash 树的遍历次数。



(a) 候选项集的个数　　(b) 频繁项集的个数

图 6-13　支持度阈值对候选项集和频繁项集的数量的影响



(a) 候选项集的个数　　(b) 频繁项集的个数

图 6-14　事务的平均宽度对候选项集和频繁项集的数量的影响

下面，详细分析 *Apriori* 算法的时间复杂度。

频繁1-项集的产生　对于每个事务，需要更新事务中出现的每个项的支持度计数。假定 w 为事务的平均宽度，则该操作需要的时间为 $O(Nw)$，其中 N 为事务的总数。

候选的产生　为了产生候选 k-项集，需要合并一对频繁$(k-1)$-项集，确定它们是否至少有 $k-2$ 个项相同。每次合并操作最多需要 $k-2$ 次相等比较。在最好情况下，每次合并都产生一个可行的候选 k-项集；在最坏的情况下，算法必须合并上次迭代发现的每对频繁$(k-1)$-项集。因此，合并频繁项集的总开销为：

$$\sum_{k=2}^{w}(k-2)|C_k| < \text{合并开销} < \sum_{k=2}^{w}(k-2)|F_{k-1}|^2$$

Hash 树在候选产生时构造，以存放候选项集。由于 Hash 树的最大深度为 k，将候选项集散列到 Hash 树的开销为 $O(\sum_{k=2}^{w} k\,|C_k|)$。在候选项剪枝的过程中，需要检验每个候选 k-项集的 k–2 个子集是否频繁。由于在 Hash 树上查找一个候选的花费是 $O(k)$，因此候选剪枝需要的时间是 $O(\sum_{k=2}^{w} k(k-2)\,|C_k|)$。

支持度计数 每个长度为$|t|$的事务将产生 $C_{|t|}^{k}$ 个 k-项集。这也是每个事务遍历 Hash 树的有效次数。支持度计数的开销为 $O(N\sum_{k} C_w^k \alpha_k)$，其中 w 是事务的最大宽度，α_k 是更新 Hash 树中一个候选 k-项集的支持度计数的开销。

6.3 规则产生

本节介绍如何有效地从给定的频繁项集中提取关联规则。忽略那些前件或后件为空的规则（$\varnothing \rightarrow Y$ 或 $Y \rightarrow \varnothing$），每个频繁 k-项集能够产生多达 2^k–2 个关联规则。关联规则可以这样提取：将项集 Y 划分成两个非空的子集 X 和 $Y-X$，使得 $X \rightarrow Y-X$ 满足置信度阈值。注意：这样的规则必然已经满足支持度阈值，因为它们是由频繁项集产生的。

例 6.2 设 $X=\{1, 2, 3\}$是频繁项集。可以由 X 产生 6 个候选关联规则：$\{1, 2\}\rightarrow\{3\}$, $\{1, 3\}\rightarrow\{2\}$, $\{2, 3\}\rightarrow\{1\}$, $\{1\}\rightarrow\{2, 3\}$, $\{2\}\rightarrow\{1, 3\}$和$\{3\}\rightarrow\{1, 2\}$。由于它们的支持度都等于 X 的支持度，这些规则一定满足支持度阈值。 □

计算关联规则的置信度并不需要再次扫描事务数据集。考虑规则$\{1, 2\}\rightarrow\{3\}$，它是由频繁项集 $X=\{1, 2, 3\}$产生的。该规则的置信度为$\sigma(\{1, 2, 3\})/\sigma(\{1, 2\})$。因为$\{1, 2, 3\}$是频繁的，支持度的反单调性确保项集$\{1, 2\}$一定也是频繁的。由于这两个项集的支持度计数已经在频繁项集产生时得到，因此不必再扫描整个数据集。

6.3.1 基于置信度的剪枝

不像支持度度量，置信度不具有任何单调性。例如：规则 $X\rightarrow Y$ 的置信度可能大于、小于或等于规则 $\tilde{X}\rightarrow\tilde{Y}$ 的置信度，其中 $\tilde{X}\subseteq X$ 且 $\tilde{Y}\subseteq Y$（见本章习题 3）。尽管如此，当比较由频繁项集 Y 产生的规则时，下面的定理对置信度度量成立。

定理 6.2 *如果规则 $X \rightarrow Y-X$ 不满足置信度阈值，则形如 $X' \rightarrow Y-X'$ 的规则一定也不满足置信度阈值，其中 X' 是 X 的子集。*

为了证明该定理，考虑如下两个规则：$X' \rightarrow Y-X'$ 和 $X\rightarrow Y-X$，其中 $X'\subset X$。这两个规则的置信度分别为$\sigma(Y)/\sigma(X')$和$\sigma(Y)/\sigma(X)$。由于 X'是 X 的子集，所以$\sigma(X')\geqslant\sigma(X)$。因此，前一个规则的置信度不可能大于后一个规则。

6.3.2 *Apriori* 算法中规则的产生

*Apriori*算法使用一种逐层方法来产生关联规则，其中每层对应于规则后件中的项数。初始，提取规则后件只含一个项的所有高置信度规则，然后，使用这些规则来产生新的候选规则。例如，

如果{*acd*}→{*b*}和{*abd*}→{*c*}是两个高置信度的规则，则通过合并这两个规则的后件产生候选规则{*ad*}→{*bc*}。图 6-15 显示了由频繁项集{*a*, *b*, *c*, *d*}产生关联规则的格结构。如果格中的任意结点具有低置信度，则根据定理 6.2，可以立即剪掉该结点生成的整个子图。假设规则{*bcd*}→{*a*}具有低置信度，则可以丢弃后件包含*a*的所有规则，包括{*cd*}→{*ab*}, {*bd*}→{*ac*}, {*bc*}→{*ad*}和{*d*}→{*abc*}。

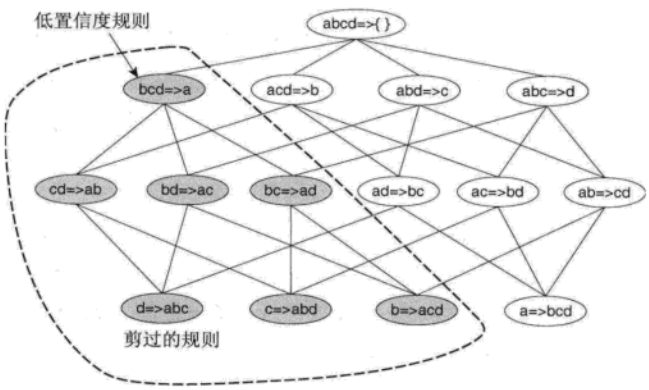


图 6-15　使用置信度度量对关联规则进行剪枝

算法 6.2 和算法 6.3 给出了关联规则产生的伪代码。注意，算法 6.3 中的 ap-genrules 过程与算法 6.1 中的频繁项集产生的过程类似。二者唯一的不同是，在规则产生时，不必再次扫描数据集来计算候选规则的置信度，而是使用在频繁项集产生时计算的支持度计数来确定每个规则的置信度。

算法 6.2　*Apriori* 算法中的规则产生

```
1: for 每一个频繁 k-项集 f_k ， k≥2 do
2:    H_1={i | i ∈ f_k }      {规则的 1-项后件}
3:    call ap-genrules(f_k, H_1)
4: end for
```

算法 6.3　过程 ap-genrules(f_k, H_m)

```
1:  k = |f_k|      {频繁项集的大小}
2:  m = |H_m|      {规则后件的大小}
3:  if k > m + 1 then
4:     H_{m+1} = apriori-gen(H_m)
5:     for 每个 h_{m+1} ∈ H_{m+1} do
6:        conf = σ(f_k)/ σ(f_k − h_{m+1})
7:        if conf≥minconf then
8:           output: 规则 (f_k − h_{m+1}) → h_{m+1}
9:        else
10:          从 H_{m+1} delete h_{m+1}
11:       end if
12:    end for
13:    call ap-genrules(f_k, H_{m+1})
14: end if
```

6.3.3 例：美国国会投票记录

本节演示对美国众议院议员投票记录应用关联分析的结果。这些数据来自1984年美国国会投票数据库，可以在UCI机器学习库中找到。每一个事务包含议员的党派信息，以及他/她对16个关键问题的投票记录。数据集中共有435个事务和34个项。表6-3列出了所有的项。

表 6-3 1984 年美国国会投票记录的二元属性列表。信息源：UCI 机器学习库

1. Republican	18. aid to Nicaragua = no
2. Democrat	19. MX-missile = yes
3. handicapped-infants = yes	20. MX-missile = no
4. handicapped-infants = no	21. immigration = yes
5. water project cost sharing = yes	22. immigration = no
6. water project cost sharing = no	23. synfuel corporation cutback = yes
7. budget-resolution = yes	24. synfuel corporation cutback = no
8. budget-resolution = no	25. education spending = yes
9. physician fee freeze = yes	26. education spending = no
10. physician fee freeze = no	27. right-to-sue = yes
11. aid to El Salvador = yes	28. right-to-sue = no
12. aid to El Salvador = no	29. crime = yes
13. religious groups in schools = yes	30. crime = no
14. religious groups in schools = no	31. duty-free-exports = yes
15. anti-satellite test ban = yes	32. duty-free-exports = no
16. anti-satellite test ban = no	33. export administration act = yes
17. aid to Nicaragua = yes	34. export administration act = no

设定 *minsup* = 30%和 *minconf* = 90%，对数据集使用 *Apriori* 算法。表6-4列举了算法产生的一些高置信度的规则。前两个规则暗示大部分同时投“援助萨尔瓦多”（aid to El Salvador）赞成票、投“预算决议案”（budget resolution）和“MX 导弹决议案”（MX missile）反对票的是共和党人；而同时投“援助萨尔瓦多”反对票、投“预算决议案”和“MX 导弹决议案”赞成票的是民主党人。这些高置信度的规则示出关键的问题可以将国会成员分为两个政党。如果降低最小置信度，将会发现很难找到区分政党的特定问题。例如，当最小置信度为40%时，这些规则暗示对于一个问题两个政党的投票差不多——投反对票的成员中，52.3%是共和党人， 47.7%的是民主党人。

表 6-4 从 1984 年美国国会投票记录中提取的关联规则

关联规则	置信度
{budget resolution = no, MX-missile = no, aid to El Salvador = yes}→{Republican}	91.0%
{budget resolution = yes, MX-missile = yes, aid to El Salvador = no}→{Democrat}	97.5%
{crime = yes, right-to-sue = yes, physician fee freeze = yes} →{Republican}	93.5%
{crime = no, right-to-sue = no, physician fee freeze = no}→{ Democrat }	100%

6.4 频繁项集的紧凑表示

实践中，由事务数据集产生的频繁项集的数量可能非常大。因此，从中识别出可以推导出其他所有的频繁项集的、较小的、具有代表性的项集是有用的。本节将介绍两种具有代表性的项集：极大频繁项集和闭频繁项集。

6.4.1 极大频繁项集

定义 6.3 极大频繁项集（maximal frequent itemset） 极大频繁项集是这样的频繁项集，

它的直接超集都不是频繁的。

为了解释这一概念，考虑图6-16所示的项集格。格中的项集分为两组：频繁项集和非频繁项集。图中虚线表示频繁项集的边界。位于边界上方的每个项集都是频繁的，而位于边界下方的项集（阴影结点）都是非频繁的。在边界附近的结点中，$\{a, d\}$, $\{a, c, e\}$和$\{b, c, d, e\}$都是极大频繁项集，因为它们的直接超集都是非频繁的。例如，项集$\{a, d\}$是极大频繁的，因为它的所有直接超集$\{a, b, d\}$, $\{a, c, d\}$和$\{a, d, e\}$都是非频繁的；相反，项集$\{a, c\}$不是极大的，因为它的一个直接超集$\{a, c, e\}$是频繁的。

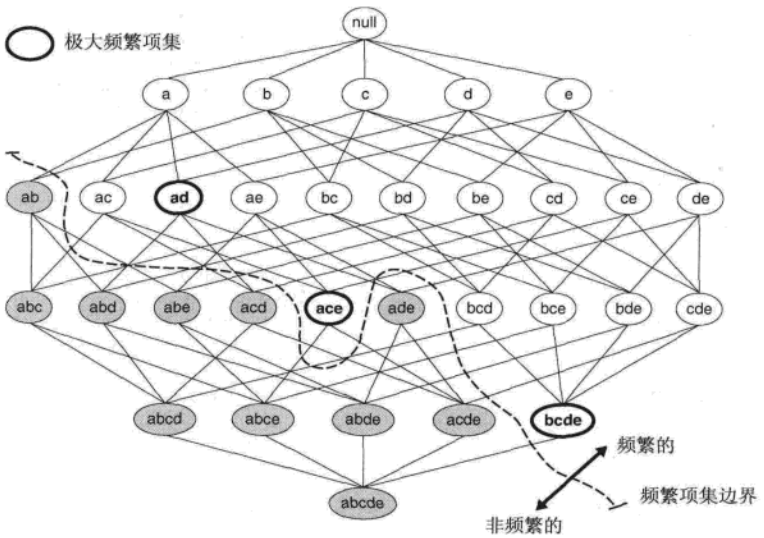


图 6-16　极大频繁项集

极大频繁项集有效地提供了频繁项集的紧凑表示。换句话说，极大频繁项集形成了可以导出所有频繁项集的最小的项集的集合。例如，图 6-16 中的频繁项集可以分为以下两组。

- 以项 a 开始、可能包含项 c, d 和 e 的频繁项集。这一组包含的项集有$\{a\}$, $\{a, c\}$, $\{a, d\}$, $\{a, e\}$和$\{a, c, e\}$。
- 以项 b, c, d 或 e 开始的频繁项集。这一组包含的项集有$\{b\}$, $\{b, c\}$, $\{c, d\}$, $\{b, c, d, e\}$等。

属于第一组的频繁项集是$\{a, c, e\}$或$\{a, d\}$的子集，而属于第二组的频繁项集都是$\{b, c, d, e\}$的子集。因此，极大频繁项集$\{a, c, e\}$, $\{a, d\}$和$\{b, c, d, e\}$提供了图 6-16 中显示的频繁项集的紧凑表示。

对于可能产生长频繁项集的数据集，极大频繁项集提供了颇有价值的表示，因为这种数据集中的频繁项集可能有指数多个。尽管如此，仅当存在一种有效的算法，可以直截了当地发现极大频繁项集而不需要枚举它们的所有子集时，这种方法才是实用的。将在 6.5 节中简略介绍一种这样的方法。

尽管提供了一种紧凑表示，但是极大频繁项集却不包含它们子集的支持度信息。例如，极大频繁项集$\{a, c, e\}$, $\{a, d\}$和$\{b, c, d, e\}$不能够提供它们子集的支持度的任何信息。因此，这就需要再一遍扫描数据集，来确定那些非极大的频繁项集的支持度计数。在某些情况下，可能需要得到保持支持度信息的频繁项集的最小表示。下一节将介绍一种这样的表示。

6.4.2 闭频繁项集

闭项集提供了频繁项集的一种最小表示，该表示不丢失支持度信息。下面给出闭项集的形式定义。

定义 6.4 闭项集（closed itemset） 项集 X 是闭的，如果它的直接超集都不具有和它相同的支持度计数。

换句话说，如果至少存在一个 X 的直接超集，其支持度计数与 X 相同，X 就不是闭的。闭项集的例子显示在图 6-17 中。为了更好地解释每个项集的支持度计数，格中每个结点（项集）都标出了与它相关联的事务的 ID。例如，由于结点$\{b, c\}$与事务 ID 1，2 和 3 相关联，因此它的支持度计数为 3。从图中给定的事务可以看出，包含 b 的每个事务也包含 c，因此，由于$\{b\}$和$\{b, c\}$的支持度是相同的，所以$\{b\}$不是闭项集。同样，由于 c 出现在所有包含 a 和 d 的事务中，所以项集$\{a, d\}$不是闭的。另一方面，$\{b, c\}$是闭项集，因为它的支持度计数与它的任何超集都不同。

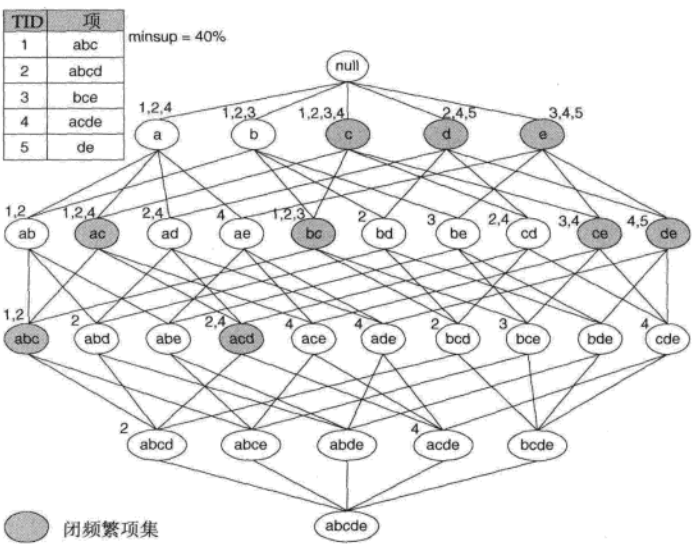


图 6-17 闭频繁项集的例子（最小支持度为 40%）

定义 6.5 闭频繁项集（closed frequent itemset） 一个项集是闭频繁项集，如果它是闭的，并且它的支持度大于或等于最小支持度阈值。

在前面的例子中，假定支持度阈值为 40%，则项集$\{b, c\}$是闭频繁项集，因为它的支持度是 60%。图 6-17 中其他闭频繁项集用带阴影结点指示出。

有一些算法可以直接从给定的数据集中提取闭频繁项集。对于这些算法的进一步讨论，感兴趣的读者可以查阅本章的文献注释。可以使用闭频繁项集的支持度来确定那些非闭的频繁项集的支持度。例如，考虑图 6-17 所示的频繁项集$\{a, d\}$。因为该项集不是闭的，所以它的支持度一定与它的某个直接超集相同，关键是确定哪个超集（$\{a, b, d\}$, $\{a, c, d\}$或$\{a, d, e\}$）恰好与$\{a, d\}$具有相同的支持度计数。*Apriori* 原理表明任何包含$\{a, d\}$超集的事务一定包含$\{a, d\}$，然而，包

含$\{a, d\}$的事务不一定要包含$\{a, d\}$的超集。由于这个原因，$\{a, d\}$的支持度一定等于它的超集的最大支持度。由于$\{a, c, d\}$的支持度大于$\{a, b, d\}$和$\{a, d, e\}$的支持度，因此$\{a, d\}$的支持度一定等于$\{a, c, d\}$的支持度。使用这种方法，可以开发一个算法，用以计算非闭频繁项集的支持度。算法 6.4 显示了这个算法的伪代码。该算法以从特殊到一般的方式进行处理，即从最大的频繁项集到最小的频繁项集。这是因为，为了找出非闭频繁项集的支持度，必须要知道它的所有超集的支持度。

算法 6.4　使用闭频繁项集进行支持度计数

```
1: 设 C 是闭频繁项集的集合
2: 设 k_max 是闭频繁项集的最大长度
3: F_kmax = {f | f ∈ C, |f| = k_max }      {找出长度为 k_max 的所有频繁项集}
4: for k = k_max − 1 downto 1 do
5:    F_k = {f | f ⊂ F_{k+1}, |f| = k}      {找出长度为 k 的所有频繁项集}
6:    for 每个 f ∈ F_k do
7:       if f ∉ C then
8:          f.support = max{f'.support | f' ∈ F_{k+1}, f ⊂ f'}
9:       end if
10:   end for
11: end for
```

为了举例说明使用闭频繁项集的优点，考虑表 6-5 中的数据集，它包含 10 个事务和 15 个项。数据集中的这些项大致可以分为 3 组：(1)组 A，包含项 a_1 到 a_5；(2)组 B，包含项 b_1 到 b_5；(3)组 C，包含项 c_1 到 c_5。注意，每一组的项与同组的项有极好的关联，并且不与其他组中的项同时出现。假定支持度阈值为 20%，频繁项集的总数为 $3 \times (2^5 - 1) = 93$。然而，该数据只有 3 个闭频繁项集：$\{a_1, a_2, a_3, a_4, a_5\}$，$\{b_1, b_2, b_3, b_4, b_5\}$和$\{c_1, c_2, c_3, c_4, c_5\}$。对于分析，仅提供这些闭频繁项集，而不是所有的频繁项集就足够了。

表 6-5　挖掘闭项集的事务数据集

序　号	a_1	a_2	a_3	a_4	a_5	b_1	b_2	b_3	b_4	b_5	c_1	c_2	c_3	c_4	c_5
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
5	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
6	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

对于删除冗余的关联规则，闭频繁项集是非常有用的。关联规则 $X \rightarrow Y$ 是冗余的，如果存在另一个关联规则 $X' \rightarrow Y'$使得两个规则的支持度和置信度都相同，其中，X'是 X 的子集，并且 Y'是 Y 的子集。在图 6-17 显示的例子中，$\{b\}$不是闭频繁项集，而$\{b, c\}$是闭的。由于关联规则$\{b\} \rightarrow \{d, e\}$与$\{b, c\} \rightarrow \{d, e\}$具有相同的支持度和置信度，所以规则$\{b\} \rightarrow \{d, e\}$是冗余的。如果使用闭频繁项集产生规则，则不会产生这样的冗余规则。

最后，极大频繁项集都是闭的，因为任何极大频繁项集都不可能与它的直接超集具有相同的支持度计数。频繁项集、极大频繁项集和闭频繁项集之间的关系显示在图 6-18 中。

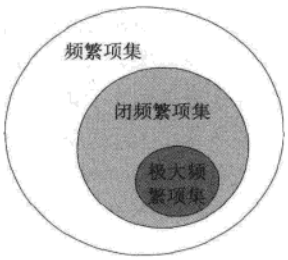


图 6-18 频繁项集、极大频繁项集和闭频繁项集之间的关系

6.5 产生频繁项集的其他方法

*Apriori*算法是在早期成功地处理频繁项集产生的组合爆炸问题的算法之一。它通过使用先验原理对指数搜索空间进行剪枝，成功地处理了组合爆炸问题。尽管显著地提高了性能，但是该算法还是会导致不可低估的I/O开销，因为它需要多次扫描事务数据集。此外，正如 6.2.5 节所提到的，对于稠密数据集，由于事务数据宽度的增加，*Apriori*算法的性能显著降低。为了克服这些局限性和提高*Apriori*算法的效率，已经开发了一些替代方法。下面是这些方法的简略描述。

项集格遍历 概念上，可以把频繁项集的搜索看作遍历图 6-1 中的项集格。算法使用的搜索策略指明了频繁项集产生过程中如何遍历格结构。根据频繁项集在格中的布局，某些搜索策略优于其他策略。下面概述这些策略。

- **一般到特殊与特殊到一般：***Apriori*算法使用了“一般到特殊”的搜索策略，合并两个频繁(k–1)-项集得到候选k-项集。只要频繁项集的最大长度不是太长，这种“一般到特殊”的搜索策略是有效的。使用这种策略效果最好的频繁项集的布局显示在图 6-19a中，其中较黑的结点代表非频繁项集。相反，“特殊到一般”的搜索策略在发现更一般的频繁项集之前，先寻找更特殊的频繁项集。这种策略对于发现稠密事务中的极大频繁项集是有用的。稠密事务中频繁项集的边界靠近格的底部，如图 6-19b所示。可以使用先验原理剪掉极大频繁项集的所有子集。具体说来，如果候选k-项集是极大频繁项集，则不必考察它的任意k–1 项子集。然而，如果候选k-项集是非频繁的，则必须在下一迭代考察它所有k–1项子集。另外一种策略是结合“一般到特殊”和“特殊到一般”的搜索策略，尽管这种双向搜索方法需要更多的空间存储候选项集，但是给定图 6-19c所示的布局，该方法有助于加快确定频繁项集边界。

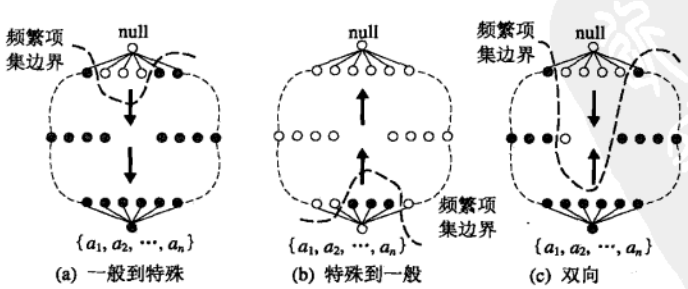
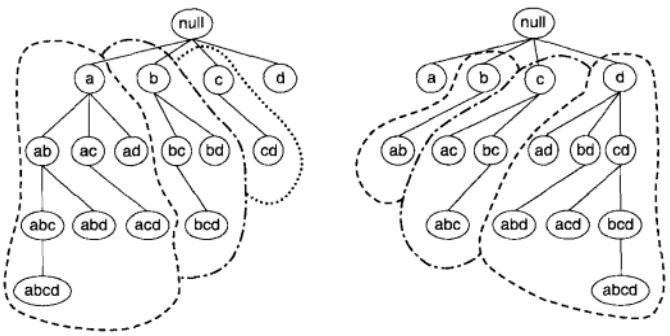


图 6-19 一般到特殊、特殊到一般和双向搜索

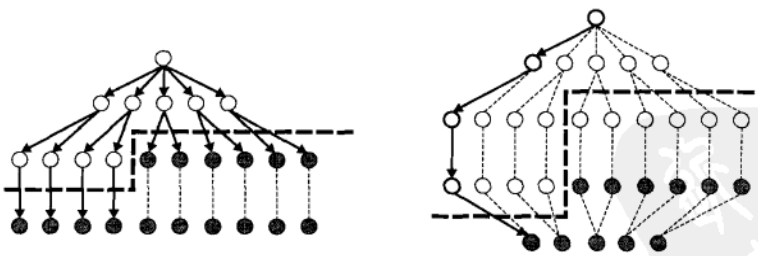
- **等价类**：另外一种遍历的方法是先将格划分为两个不相交的结点组（或等价类）。频繁项集产生算法依次在每个等价类内搜索频繁项集。例如，*Apriori*算法采用的逐层策略可以看作根据项集的大小划分格，即在处理较大项集之前，首先找出所有的频繁 1-项集。等价类也可以根据项集的前缀或后缀来定义。在这种情况下，两个项集属于同一个等价类，如果它们共享长度为k的相同前缀或后缀。在基于前缀的方法中，算法首先搜索以前缀a开始的频繁项集，然后是以前缀b等开始的频繁项集，然后是c，如此下去。基于前缀和基于后缀的等价类都可以使用图 6-20 所示的类似于树的结构来演示。



(a) 前缀树　　(b) 后缀树

图 6-20　基于项集前缀和后缀的等价类

- **宽度优先与深度优先**：*Apriori*算法采用宽度优先的方法遍历格，如图6-21a所示。它首先发现所有频繁1-项集，接下来是频繁2-项集，如此下去直到没有新的频繁项集产生为止。也可以以深度优先的方式遍历项集格，如图6-21b和图6-22所示。比如说，算法可以从图6-22中的结点a开始，计算其支持度计数并判断它是否频繁。如果是，算法渐增地扩展下层结点，即*ab*, *abc*，等等，直到到达一个非频繁结点，如*abcd*。然后，回溯到下一个分支，比如说*abce*，并且继续搜索。



(a) 宽度优先　　(b) 深度优先

图 6-21　宽度优先和深度优先遍历

通常，深度优先搜索方法用于发现极大频繁项集的算法。这种方法比宽度优先方法更快地检测到频繁项集边界。一旦发现一个极大频繁项集，就可以在它的子集上进行剪枝。例如，如果图 6-22 中的结点 *bcde* 是极大频繁项集，则算法就不必访问以 *bd*, *be*, *c*, *d* 和 *e* 为根的子树，因为它们不可能包含任何极大频繁项集。然而，如果 *abc* 是极大频繁项集，则只有 *ac* 和 *bc* 这样的结点

不是极大频繁项集，但以它们为根的子树还可能包含极大频繁项集。深度优先方法还允许使用不同的基于项集支持度的剪枝方法。例如，假定项集{*a*, *b*, *c*}和{*a*, *b*}具有相同的支持度，则可以跳过以 *abd* 和 *abe* 为根的子树，因为可以确保它们不包含任何极大频繁项集。该问题的证明作为习题留给读者。

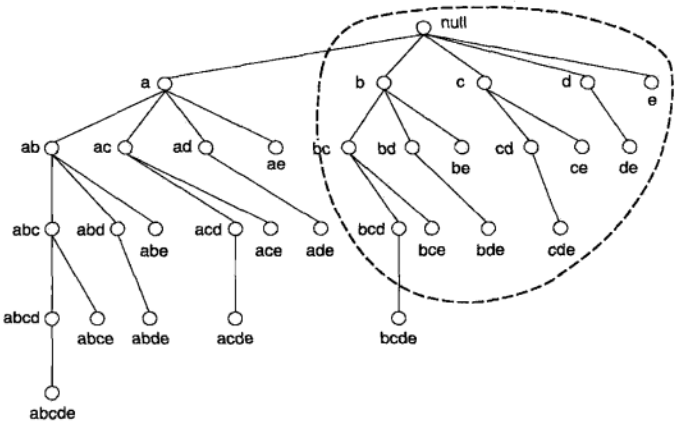


图 6-22 使用深度优先方法产生候选项集

事务数据集的表示 事务数据集的表示方法有多种。表示方法的选择可能影响计算候选项集支持度的 I/O 开销。图 6-23 显示了两种表示购物篮事务的不同方法。左边的表示法称作**水平**（horizontal）数据布局，许多关联规则挖掘算法（包括 *Apriori*）都采用这种表示法；另一种可能的方法是存储与每一个项相关联的事务标识符列表（TID 列表），这种表示法称作**垂直**（vertical）数据布局。候选项集的支持度通过取其子项集 TID 列表的交得到。随着不断进展，处理较大的项集，TID 列表的长度不断收缩。然而，这种方法存在一个问题：TID 列表的初始集合可能太大，以致无法放进内存，因此就需要特殊的巧妙技术来压缩 TID 列表。下一节，将介绍另外一种表示数据的有效方法。

水平数据布局

TID	项
1	a,b,e
2	b,c,d
3	c,e
4	a,c,d
5	a,b,c,d
6	a,e
7	a,b
8	a,b,c
9	a,c,d
10	b

垂直数据布局

a	b	c	d	e
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

图 6-23 水平和垂直数据形式

6.6 FP 增长算法

本节介绍另一种称作 FP 增长的算法。该算法采用完全不同的方法来发现频繁项集。该算法

不同于*Apriori*算法的“产生-测试”范型，而是使用一种称作FP树的紧凑数据结构组织数据，并直接从该结构中提取频繁项集。下面详细说明该方法。

6.6.1　FP树表示法

FP树是一种输入数据的压缩表示，它通过逐个读入事务，并把每个事务映射到FP树中的一条路径来构造。由于不同的事务可能会有若干个相同的项，因此它们的路径可能部分重叠。路径相互重叠越多，使用FP树结构获得的压缩的效果越好。如果FP树足够小，能够存放在内存中，就可以直接从这个内存中的结构提取频繁项集，而不必重复地扫描存放在硬盘上的数据。

图6-24显示了一个数据集，它包含10个事务和5个项。图中还绘制了读入前3个事务之后FP树的结构。树中每一个结点都包括一个项的标记和一个计数，计数显示映射到给定路径的事务个数。初始，FP树仅包含一个根结点，用符号null标记。随后，用如下方法扩充FP树：

(1) 扫描一次数据集，确定每个项的支持度计数。丢弃非频繁项，而将频繁项按照支持度的递减排序。对于图6-24中的数据集，a是最频繁的项，接下来依次是b, c, d和e。

(2) 算法第二次扫描数据集，构建FP树。读入第一个事务$\{a, b\}$之后，创建标记为a和b的结点。然后形成null→a→b路径，对该事务编码。该路径上的所有结点的频度计数为1。

(3) 读入第二个事务$\{b, c, d\}$之后，为项b, c和d创建新的结点集。然后，连接结点null→b→c→d，形成一条代表该事务的路径。该路径上的每个结点的频度计数也等于1。尽管前两个事务具有一个共同项b，但是它们的路径不相交，因为这两个事务没有共同的前缀。

(4) 第三个事务$\{a, c, d, e\}$与第一个事务共享一个共同前缀项a，所以第三个事务的路径null→a→c→d→e与第一个事务的路径null→a→b部分重叠。因为它们的路径重叠，所以结点a的频度计数增加为2，而新创建的结点c, d和e的频度计数等于1。

(5) 继续该过程，直到每个事务都映射到FP树的一条路径。读入所有的事务后形成的FP树显示在图6-24的底部。

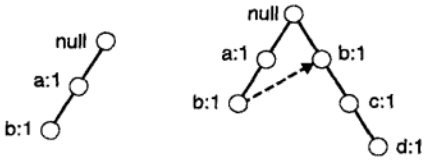
通常，FP树的大小比未压缩的数据小，因为购物篮数据的事务常常共享一些共同项。在最好情况下，所有的事务都具有相同的项集，FP树只包含一条结点路径。当每个事务都具有唯一项集时，导致最坏情况发生，由于事务不包含任何共同项，FP树的大小实际上与原数据的大小一样，然而，由于需要附加的空间为每个项存放结点间的指针和计数，FP树的存储需求增大。

FP树的大小也取决于项的排序方式。如果颠倒前面例子的序，即项按照支持度由小到大排列，则结果FP树显示在图6-25中。该树显得更加茂盛，因为根结点上的分支数由2增加到5，并且包含了高支持度项a和b的结点数由3增加到12。尽管如此，支持度计数递减序并非总是导致最小的树。例如，假设加大图6-24给定的数据集，增加100个事务包含$\{e\}$、80个事务包含$\{d\}$、60个事务包含$\{c\}$、40个事务包含$\{b\}$。现在，项e是最频繁的，接下来依次是d, c, b和a。使用加大的事务数据，支持度计数递减序将导致类似于图6-25中的FP树，而基于支持度计数递增序将产生一棵类似于图6-24(iv)的较小的FP树。

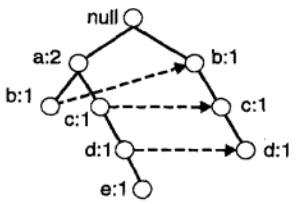
FP树还包含一个连接具有相同项的结点的指针列表。这些指针在图6-24和图6-25中用虚线表示，有助于方便快速地访问树中的项。下一节，解释如何使用FP树和它的相应指针产生频繁项集。

事务数据集

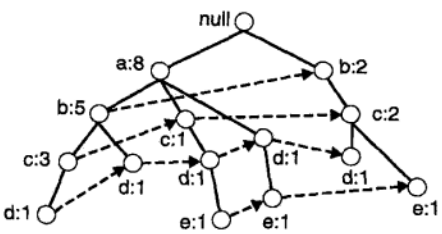
TID	项
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}



(i) 读入 TID = 1 之后　(ii) 读入 TID = 2 之后



(iii) 读入 TID = 3 之后



(iv) 读入 TID = 10 之后

图 6-24　构造 FP 树

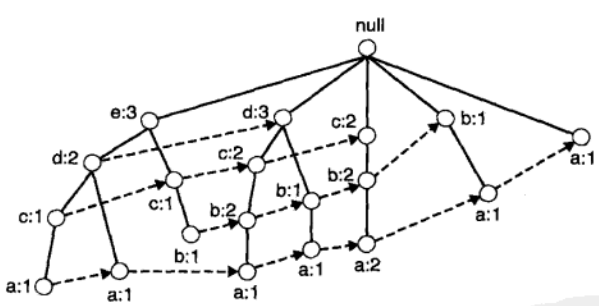


图 6-25　对图 6-24 所示数据集使用不同的项序方案的 FP 树表示

6.6.2　FP 增长算法的频繁项集产生

FP 增长（FP-growth）是一种以自底向上方式探索树，由 FP 树产生频繁项集的算法。给定图 6-24 所示的树，算法首先查找以 e 结尾的频繁项集，接下来依次是 d, c, b，最后是 a。这种用于发现以某一个特定项结尾的频繁项集的自底向上策略等价于 6.5 节介绍的基于后缀的方法。由于每一个事务都映射到 FP 树中的一条路径，因而通过仅考察包含特定结点（例如 e）的路径，就可以发现以 e 结尾的频繁项集。使用与结点 e 相关联的指针，可以快速访问这些路径。图 6-26a 显示了所提取的路径。稍后详细解释如何处理这些路径，以得到频繁项集。

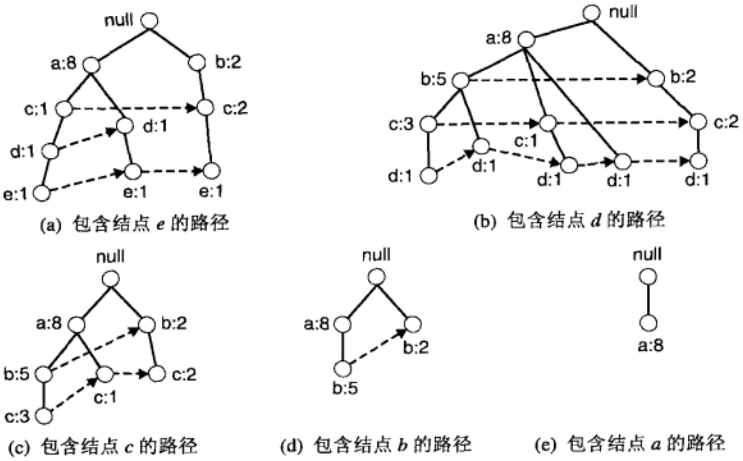


图 6-26　将频繁项集产生的问题分解成多个子问题，其中每个子问题分别涉及发现以 e, d, c, b 和 a 结尾的频繁项集

发现以 e 结尾的频繁项集之后，算法通过处理与结点 d 相关联的路径，进一步寻找以 d 结尾的频繁项集。图 6-26b 显示了对应的路径。继续该过程，直到处理了所有与结点 c, b 和 a 相关联的路径为止。图 6-26c、图 6-26d、图 6-26e 分别显示了这些项的路径，而它们对应的频繁项集汇总在表 6-6 中。

表 6-6　依据相应的后缀排序的频繁项集

后　缀	频 繁 项 集
e	$\{e\},\{d,e\},\{a,d,e\},\{c,e\},\{a,e\}$
d	$\{d\},\{c,d\},\{b,c,d\},\{a,c,d\},\{b,d\},\{a,b,d\},\{a,d\}$
c	$\{c\},\{b,c\},\{a,b,c\},\{a,c\}$
b	$\{b\},\{a,b\}$
a	$\{a\}$

FP增长采用分治策略将一个问题分解为较小的子问题，从而发现以某个特定后缀结尾的所有频繁项集。例如，假设对发现所有以e结尾的频繁项集感兴趣。为了实现这个目的，必须首先检查项集$\{e\}$本身是否频繁。如果它是频繁的，则考虑发现以de结尾的频繁项集子问题，接下来是ce和ae。依次，每一个子问题可以进一步划分为更小的子问题。通过合并这些子问题得到的结果，就可以找到所有以e结尾的频繁项集。这种分治策略是FP增长算法采用的关键策略。

为了更具体地说明如何解决这些子问题，考虑发现所有以 e 结尾的频繁项集的任务。

(1) 第一步收集包含 e 结点的所有路径。这些初始的路径称为**前缀路径**（prefix path），如图 6-27a 所示。

(2) 由图 6-27a 中所显示的前缀路径，通过把与结点 e 相关联的支持度计数相加得到 e 的支持度计数。假定最小支持度为 2，因为$\{e\}$的支持度是 3 所以它是频繁项集。

(3) 由于$\{e\}$是频繁的，因此算法必须解决发现以 de、ce、be 和 ae 结尾的频繁项集的子问题。在解决这些子问题之前，必须先将前缀路径转化为**条件 FP 树**（conditional FP-tree）。除了用于发现以某特定后缀结尾的频繁项集之外，条件 FP 树的结构与 FP 树类似。条件 FP 树通过以下步骤得到。

(a) 首先，必须更新前缀路径上的支持度计数，因为某些计数包括那些不含项 e 的事务。例

如，图 6-27a 中的最右边路径 null→*b*:2→*c*:2→*e*:1，包括并不含项 *e* 的事务{*b*, *c*}。因此，必须将该前缀路径上的计数调整为 1，以反映包含{*b*, *c*, *e*}的事务的实际个数。

(b) 删除 *e* 的结点，修剪前缀路径。删除这些结点是因为，沿这些前缀路径的支持度计数已经更新，以反映包含 *e* 的那些事务，并且发现以 *de*, *ce*, *be* 和 *ae* 结尾的频繁项集的子问题不再需要结点 *e* 的信息。

(c) 更新沿前缀路径上的支持度计数之后，某些项可能不再是频繁的。例如，结点 *b* 只出现了 1 次，它的支持度计数等于 1，这就意味着只有一个事务同时包含 *b* 和 *e*。因为所有以 *be* 结尾的项集一定都是非频繁的，所以在其后的分析中可以安全地忽略 *b*。

e 的条件 FP 树显示在图 6-27b 中。该树看上去与原来的前缀路径不同，因为频度计数已经更新，并且结点 *b* 和 *e* 已被删除。

(4) FP 增长使用 *e* 的条件 FP 树来解决发现以 *de*, *ce*, *be* 和 *ae* 结尾的频繁项集的子问题。为了发现以 *de* 结尾的频繁项集，从项 *e* 的条件 FP 树收集 *d* 的所有前缀路径（图 6-27c）。通过将与结点 *d* 相关联的频度计数求和，得到项集{*d*, *e*}的支持度计数。因为项集{*d*, *e*}的支持度计数等于 2，所以它是频繁项集。接下来，算法采用第 3 步介绍的方法构建 *de* 的条件 FP 树。更新了支持度计数并删除了非频繁项 *c* 之后，*de* 的条件 FP 树显示在图 6-27d 中。因为该条件 FP 树只包含一个支持度等于最小支持度的项 *a*，算法提取出频繁项集{*a*, *d*, *e*}并转到下一个子问题，产生以 *ce* 结尾的频繁项集。处理 *c* 的前缀路径后，只发现项集{*c*, *e*}是频繁的。接下来，算法继续解决下一个子问题并发现项集{*a*, *e*}是剩下唯一的频繁项集。

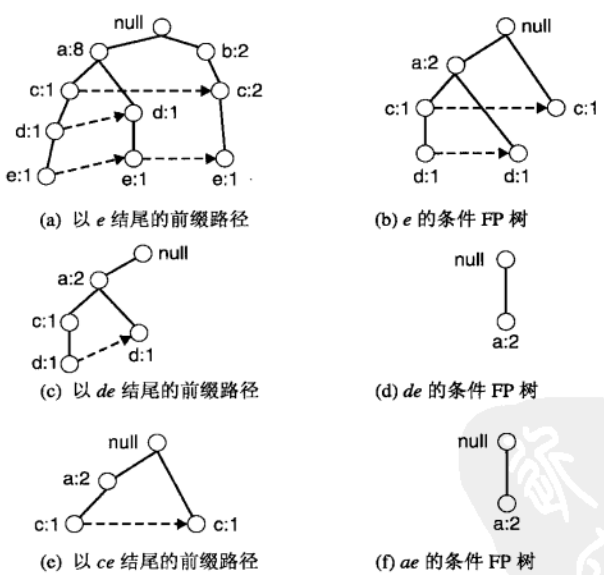


图 6-27 使用 FP 增长算法发现以 *e* 结尾的频繁项集的例子

这个例子解释了FP增长算法中使用的分治方法。每一次递归，都要通过更新前缀路径中的支持度计数和删除非频繁的项来构建条件FP树。由于子问题是不相交的，因此FP增长不会产生任何重复的项集。此外，与结点相关联的支持度计数允许算法在产生相同的后缀项时进行支持度

计数。

FP增长是一个有趣的算法，它展示了如何使用事务数据集的压缩表示来有效地产生频繁项集。此外，对于某些事务数据集，FP增长算法比标准的*Apriori*算法要快几个数量级。FP增长算法的运行性能依赖于数据集的**压缩因子**（compaction factor）。如果生成的条件FP树非常茂盛（在最坏情形下，是一棵满前缀树），则算法的性能显著下降，因为算法必须产生大量的子问题，并且需要合并每个子问题返回的结果。

6.7　关联模式的评估

关联分析算法具有产生大量模式的潜在能力。例如，虽然表6-1中所显示的数据集只有6个项，但是在特定的支持度和置信度阈值下，它能够产生数以百计的关联规则。由于真正的商业数据库的数据量和维数都非常大，很容易产生数以千计、甚至是数以百万计的模式，而其中很大一部分可能是不感兴趣的。筛选这些模式，以识别最有趣的模式并非一项平凡的任务，因为“一个人的垃圾可能是另一个人的财富”。因此，建立一组广泛接受的评价关联模式质量的标准是非常重要的。

第一组标准可以通过统计论据建立。涉及相互独立的项或覆盖少量事务的模式被认为是不令人感兴趣的，因为它们可能反映数据中的伪联系。这些模式可以使用**客观兴趣度度量**（objective interestingness measure）来排除，客观兴趣度度量使用从数据推导出的统计量来确定模式是否是有趣的。客观兴趣度度量的例子包括支持度、置信度和相关性。

第二组标准可以通过主观论据建立，即模式被主观地认为是无趣的，除非它能够揭示料想不到的信息或提供导致有益的行动的有用信息。例如，规则{黄油}→{面包}可能不是有趣的，尽管有很高的支持度和置信度，但是它表示的关系显而易见。另一方面，规则{尿布}→{啤酒}是有趣的，因为这种联系十分出乎意料，并且可能为零售商提供新的交叉销售机会。将主观知识加入到模式的评价中是一项困难的任务，因为需要来自领域专家的大量先验信息。

下面是一些将主观信息加入到模式发现任务中的方法。

- **可视化**（visualization） 这种方法需要友好的环境，保持用户参与，允许领域专家解释和检验被发现的模式，与数据挖掘系统交互。
- **基于模板的方法**（template-based approach） 这种方法允许用户限制挖掘算法提取的模式类型。只把满足用户指定的模板的规则提供给用户，而不是报告提取所有模式。
- **主观兴趣度度量**（subjective interestingness measure） 主观度量可以基于领域信息来定义，如概念分层（将在7.3节讨论）或商品利润等。然后，使用这些度量来过滤那些显而易见和没有实际价值的模式。

对主观兴趣度度量感兴趣的读者可以参阅本章文献注释中列举的文献。

6.7.1　兴趣度的客观度量

客观度量是一种评估关联模式质量的数据驱动的方法。它不依赖于领域，只需要最小限度用户的输入信息，它不需要通过设置阈值来过滤低质量的模式。客观度量常常基于**相依表**（contingency table）中列出的频度计数来计算。表6-7显示了一对二元变量A和B的相依表。使用记号$\overline{A}$（$\overline{B}$）表示A（B）不在事务中出现。在这个2×2的表中，每个f_{ij}都代表一个频度计数。例如，f_{11}表示A和B同时出现在一个事务中的次数，f_{01}表示包含B但不包含A的事务的个数。

行和f_{1+}表示A的支持度计数，而列和f_{+1}表示B的支持度计数。最后，尽管讨论主要关注非对称的二元变量，相依表也可以应用于其他属性类型，如对称的二元变量、标称变量和序数变量。

表 6-7 变量 A 和 B 的 2 路相依表

	B	$\bar{B}$	
A	f_{11}	f_{10}	f_{1+}
$\bar{A}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

支持度-置信度框架的局限性 现有的关联规则的挖掘算法需要使用支持度和置信度来除去没有意义的模式。支持度的缺点在于许多潜在的有意义的模式由于包含支持度小的项而被删去，这一点将在随后的 6.8 节介绍。置信度的缺点更加微妙，用下面的例子最适于说明。

例 6.3 假定希望分析爱喝咖啡和爱喝茶的人之间的关系。收集一组人关于饮料偏爱的信息，并汇总在表 6-8 中。

表 6-8 1000 个人的饮料偏爱

	咖啡	咖啡	
茶	150	50	200
茶	650	150	800
	800	200	1000

可以使用表中给出的信息来评估关联规则{茶}→{咖啡}。猛一看，似乎喜欢喝茶的人也喜欢喝咖啡，因为该规则的支持度（15%）和置信度（75%）都相当的高。这个推论也许是可以接受的，但是所有的人中，不管他是否喝茶，喝咖啡的人的比例为 80%，而喝咖啡的饮茶者却只占 75%。也就是说，一个人如果喝茶，则他喝咖啡的可能性由 80%减到了 75%。因此，尽管规则{茶}→{咖啡}有很高的置信度，但是它却是一个误导。 □

置信度的缺陷在于该度量忽略了规则后件中项集的支持度。的确，如果考虑喝咖啡者的支持度，则将毫不奇怪地发现许多喝茶的人也喝咖啡。更奇怪的是喝咖啡的饮茶者所占的比例实际少于所有喝咖啡的人所占的比例，这表明饮茶者和喝咖啡的人之间存在着一种逆关系。

由于支持度-置信度框架的局限性，各种客观度量已经用来评估关联模式。下面，简略介绍这些度量并解释它们的优点和局限性。

兴趣因子 茶与咖啡的例子表明，由于置信度度量忽略了规则后件中出现的项集的支持度，高置信度的规则有时可能出现误导。解决这个问题的一种方法是使用称作**提升度**（lift）的度量：

$$lift(A \rightarrow B) = \frac{c(A \rightarrow B)}{s(B)} \tag{6-4}$$

它计算规则置信度和规则后件中项集的支持度之间的比率。对于二元变量，提升度等价于另一种称作**兴趣因子**（interest factor）的客观度量，其定义如下：

$$I(A,B) = \frac{s(A,B)}{s(A) \times s(B)} = \frac{Nf_{11}}{f_{1+}f_{+1}} \tag{6-5}$$

兴趣因子比较模式的频率与统计独立假定下计算的基线频率。对于相互独立的两个变量，基线频率为：

$$\frac{f_{11}}{N}=\frac{f_{1+}}{N}\times\frac{f_{+1}}{N}\text{，或等价地 }f_{11}=\frac{f_{1+}f_{+1}}{N} \tag{6-6}$$

该式从使用简单比例作为概率估计的标准方法得到。分数 f_{11}/N 是联合概率 $P(A, B)$的估计，而 f_{1+}/N 和 f_{+1}/N 分别是概率 $P(A)$和 $P(B)$的估计。如果 A 和 B 是相互独立的，则 $P(A, B)=P(A)\times P(B)$，从而产生公式（6-6）。使用公式（6-5）和（6-6），该度量可以解释如下：

$$I(A,B)\begin{cases}=1 & \text{如果 } A \text{ 和 } B \text{ 是独立的}\\ >1 & \text{如果 } A \text{ 和 } B \text{ 是正相关的}\\ <1 & \text{如果 } A \text{ 和 } B \text{ 是负相关的}\end{cases} \tag{6-7}$$

对于表 6-8 中所显示的例子，$I=\dfrac{0.15}{0.2\times0.8}=0.9375$，这表明在饮茶者和喝咖啡的人之间稍微负相关。

兴趣因子的局限性　这里以一个文本挖掘领域的例子解释兴趣因子的局限性。在文本挖掘领域，假定一对词之间的关联依赖于同时包含这两个词的文档的数量是合理的。例如，由于二者之间的较强关联，预计在计算机文献中词“数据”和“挖掘”同时出现的频率高于“编译”和“挖掘”同时出现的频率。

表 6-9 显示了两对词$\{p, q\}$和$\{r, s\}$出现的频率。使用公式（6-5），$\{p, q\}$和$\{r, s\}$的兴趣因子分别为 1.02 和 4.08。由于下面的原因，这些结果多少有点问题：虽然 p 和 q 同时出现在 88%的文档中，但是它们的兴趣因子接近于 1，表明二者是相互独立的；另一方面，$\{r, s\}$的兴趣因子比$\{p, q\}$的高，尽管 r 和 s 很少同时出现在同一个文档中。在这种情况下，置信度可能是一个更好的选择，因为置信度表明 p 和 q 之间的关联（94.6%）远远强于 r 和 s 之间的关联（28.6%）。

表 6-9　词对$\{p,q\}$和$\{r,s\}$的相依表

	p	$\bar{p}$	
q	880	50	930
$\bar{q}$	50	20	70
	930	70	1000

	r	$\bar{r}$	
s	20	50	70
$\bar{s}$	50	880	930
	70	930	1000

相关分析　相关分析是分析一对变量之间关系的基于统计学的技术。对于连续变量，相关度用皮尔森相关系数定义（参看 2.4.5 节公式（2-10））。对于二元变量，相关度可以用ϕ系数度量。ϕ系数定义如下：

$$\phi=\frac{f_{11}f_{00}-f_{01}f_{10}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}} \tag{6-8}$$

相关度的值从−1（完全负相关）到+1（完全正相关）。如果变量是统计独立的，则$\phi=0$。例如，在表 6-8 中给出的饮茶者和喝咖啡者之间的相关度为−0.0625。

相关分析的局限性　相关性的缺点通过表 6-9 所给出词的关联可以看出。虽然词 p 和 q 同时

出现的次数比 r 和 s 更多，但是它们的ϕ系数是相同的，即$\phi(p, q) = \phi(r, s) = 0.232$。这是因为，$\phi$系数把项在事务中同时出现和同时不出现视为同等重要。因此，它更适合分析对称的二元变量。这种度量的另一个局限性是，当样本大小成比例变化时，它不能够保持不变。该问题将在稍后介绍客观度量的性质时更详细地讨论。

IS 度量 *IS* 是另一种度量，用于处理非对称二元变量。该度量定义如下：

$$IS(A,B) = \sqrt{I(A,B) \times s(A,B)} = \frac{s(A,B)}{\sqrt{s(A)s(B)}} \tag{6-9}$$

注意，当模式的兴趣因子和模式支持度都很大时，*IS* 也很大。例如，表 6-9 中显示的词对$\{p, q\}$和$\{r, s\}$的 *IS* 值分别是 0.946 和 0.286。与兴趣因子和ϕ系数给出的结果相反，*IS* 度量暗示$\{p, q\}$之间的关联强于$\{r,s\}$，这与期望的文档中词的关联一致。

可以证明 *IS* 数学上等价于二元变量的余弦度量（参见 2.4.5 节公式（2-7））。在这一点上，将 **A** 和 **B** 看作一对位向量，$\mathbf{A} \cdot \mathbf{B} = s(A, B)$表示两个向量的点积，$|\mathbf{A}| = \sqrt{s(A)}$ 表示向量 **A** 的大小。因此，

$$IS(A,B) = \frac{s(A,B)}{\sqrt{s(A) \times s(B)}} = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| \times |\mathbf{B}|} = cosine(\mathbf{A}, \mathbf{B}) \tag{6-10}$$

IS 度量也可以表示为从一对二元变量中提取出的关联规则的置信度的几何均值：

$$IS(A,B) = \sqrt{\frac{s(A,B)}{s(A)} \times \frac{s(A,B)}{s(B)}} = \sqrt{c(A \rightarrow B) \times c(B \rightarrow A)} \tag{6-11}$$

由于两个数的几何均值总是接近于较小的数，所以只要规则 $p \rightarrow q$ 或 $q \rightarrow p$ 中的一个具有较低的置信度，项集$\{p, q\}$的 *IS* 值就较低。

IS 度量的局限性 一对相互独立的项集 A 和 B 的 *IS* 值是：

$$IS_{\text{indep}}(A, B) = \frac{s(A,B)}{\sqrt{s(A) \times s(B)}} = \frac{s(A) \times s(B)}{\sqrt{s(A) \times s(B)}} = \sqrt{s(A) \times s(B)}$$

因为 *IS* 值取决于 $s(A)$和 $s(B)$，所以 *IS* 存在与置信度度量类似的问题——即使是不相关或负相关的模式，度量值也可能相当大。例如，尽管表 6-10 中所显示的项 p 和 q 之间的 *IS* 值相当大（0.889），当项统计独立时它仍小于期望值（$IS_{\text{indep}} = 0.9$）。

表 6-10 项 p 和 q 的相依表的例子

	q	$\bar{q}$	
p	800	100	900
$\bar{p}$	100	0	100
	900	100	1000

1. 其他客观兴趣度度量

除了迄今为止介绍的度量外，仍有另外一些分析二元变量之间联系的度量方法。这些度量可

以分为两类：**对称的**和**非对称的**度量。度量 M 是对称的，如果 $M(A\rightarrow B)=M(B\rightarrow A)$。例如，兴趣因子是对称的度量，因为规则 $A\rightarrow B$ 和 $B\rightarrow A$ 的兴趣因子的值相等；相反，置信度是非对称度量，因为规则 $A\rightarrow B$ 和 $B\rightarrow A$ 的置信度可能不相等。对称度量常常用来评价项集，而非对称度量方法更适合于分析关联规则。表 6-11 和表 6-12 用 2×2 相依表的频度计数，给出了这些度量的部分定义。

表 6-11　项集{A,B}的对称的客观度量

度量（符号）	定　义
相关性（ϕ）	$\dfrac{Nf_{11}-f_{1+}f_{+1}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$
几率（α）	$(f_{11}f_{00})\ /\ (f_{10}f_{01})$
κ（k）	$\dfrac{Nf_{11}+Nf_{00}-f_{1+}f_{+1}-f_{0+}f_{+0}}{N^2-f_{1+}f_{+1}-f_{0+}f_{+0}}$
兴趣因子（I）	$(Nf_{11})\ /\ (f_{1+}f_{+1})$
余弦（IS）	$(f_{11})\Big/\left(\sqrt{f_{1+}f_{+1}}\right)$
Piatetsky-Shapiro（PS）	$\dfrac{f_{11}}{N}-\dfrac{f_{1+}f_{+1}}{N^2}$
集体强度（S）	$\dfrac{f_{11}+f_{00}}{f_{1+}f_{+1}+f_{0+}f_{+0}}\times\dfrac{N-f_{1+}f_{+1}-f_{0+}f_{+0}}{N-f_{11}-f_{00}}$
Jaccard（ζ）	$f_{11}/\ (f_{1+}+f_{+1}-f_{11})$
全置信度（h）	$\min\left[\dfrac{f_{11}}{f_{1+}},\ \dfrac{f_{11}}{f_{+1}}\right]$

表 6-12　规则 A→B 的非对称的客观度量

度量（符号）	定　义
Goodman-Kruskal（λ）	$\left(\sum_j \max_k f_{jk}-\max_k f_{+k}\right)\Big/\left(N-\max_k f_{+k}\right)$
互信息（M）	$\left(\sum_i\sum_j \dfrac{f_{ij}}{N}\log\dfrac{Nf_{ij}}{f_{i+}f_{+j}}\right)\Big/\left(-\sum_i \dfrac{f_{i+}}{N}\log\dfrac{f_{i+}}{N}\right)$
J 度量（J）	$\dfrac{f_{11}}{N}\log\dfrac{Nf_{11}}{f_{1+}f_{+1}}+\dfrac{f_{10}}{N}\log\dfrac{Nf_{10}}{f_{1+}f_{+0}}$
Gini 指标（G）	$\dfrac{f_{1+}}{N}\times\left[\left(\dfrac{f_{11}}{f_{1+}}\right)^2+\left(\dfrac{f_{10}}{f_{1+}}\right)^2\right]-\left(\dfrac{f_{+1}}{N}\right)^2+\dfrac{f_{0+}}{N}\times\left[\left(\dfrac{f_{01}}{f_{0+}}\right)^2+\left(\dfrac{f_{00}}{f_{0+}}\right)^2\right]-\left(\dfrac{f_{+0}}{N}\right)^2$
拉普拉斯（L）	$(f_{11}+1)/(f_{1+}+2)$
信任度（V）	$(f_{1+}f_{+0})/(Nf_{10})$
可信度因子（F）	$\left(\dfrac{f_{11}}{f_{1+}}-\dfrac{f_{+1}}{N}\right)\Big/\left(1-\dfrac{f_{+1}}{N}\right)$
Added Value（AV）	$\dfrac{f_{11}}{f_{1+}}-\dfrac{f_{+1}}{N}$

2. 客观度量的一致性

给定各种各样的可用度量后，产生的一个合理问题是：当这些度量应用到一组关联模式时是

否会产生类似的有序结果。如果这些度量是一致的，那么就可以选择它们中的任意一个作为评估度量。否则的话，为了确定哪个度量更适合分析某个特定类型的模式，了解这些度量之间的不同点是非常重要的。

假设使用对称和非对称度量确定表 6-13 中的 10 个相依表的秩。这些相依表用来解释已有度量之间的差异。这些度量产生的序分别显示在表 6-14 和表 6-15 中（1 是最有趣的，10 是最无趣的）。虽然某些度量值看上去是一致的，但是仍有某些度量产生十分不同的次序结果。例如，ϕ 系数与κ和集体强度产生的秩是一致的，但是与兴趣因子和几率产生的秩有些不同。此外，相依表 E_{10} 根据ϕ系数具有最低秩，而根据兴趣因子却具有最高秩。

表 6-13 相依表的例子

实例	f_{11}	f_{10}	f_{01}	f_{00}
E_1	8123	83	424	1370
E_2	8330	2	622	1046
E_3	3954	3080	5	2961
E_4	2886	1363	1320	4431
E_5	1500	2000	500	6000
E_6	4000	2000	1000	3000
E_7	9481	298	127	94
E_8	4000	2000	2000	2000
E_9	7450	2483	4	63
E_{10}	61	2483	4	7452

表 6-14 使用表 6-11 中的对称度量对相依表定秩

	ϕ	α	κ	I	IS	PS	S	ζ	h
E_1	1	3	1	6	2	2	1	2	2
E_2	2	1	2	7	3	5	2	3	3
E_3	3	2	4	4	5	1	3	6	8
E_4	4	8	3	3	7	3	4	7	5
E_5	5	7	6	2	9	6	6	9	9
E_6	6	9	5	5	6	4	5	5	7
E_7	7	6	7	9	1	8	7	1	1
E_8	8	10	8	8	8	7	8	8	7
E_9	9	4	9	10	4	9	9	4	4
E_{10}	10	5	10	1	10	10	10	10	10

表 6-15 使用表 6-12 中的非对称度量对相依表定秩

	λ	M	J	G	L	V	F	AV
E_1	1	1	1	1	4	2	2	5
E_2	2	2	2	3	5	1	1	6
E_3	5	3	5	2	2	6	6	4
E_4	4	6	3	4	9	3	3	1
E_5	9	7	4	6	8	5	5	2
E_6	3	8	6	5	7	4	4	3
E_7	7	5	9	8	3	7	7	9
E_8	8	9	7	7	10	8	8	7
E_9	6	4	10	9	1	9	9	10
E_{10}	10	10	8	10	6	10	10	8

3. 客观度量的性质

表 6-14 中的结果数据表明很多度量对同一个模式的质量提供了互相矛盾的信息。为了了解它们之间的差异，需要考察这些度量性质。

反演性　考虑图6-28中显示的位向量，每个列向量中的0/1位表示一个事务（行）是否包含某个特定的项（列）。例如，向量**A**表示项a属于第一个和最后一个事务，而向量**B**表示项b只在第五个事务中出现。事实上，向量**C**和**E**与向量**A**有一定的关系——它们的位由0（不出现）反转为1（出现），反之亦然。同理，向量**D**与向量**B**和**F**也存在着同样的位反转关系。这种反转位向量的过程称为**反演**（inversion）。如果度量在反演操作下是不变的，则向量对(**C**, **D**)的度量值和向量对(**A**, **B**)的度量值应当相等。度量的反演性可以用如下方法检验。

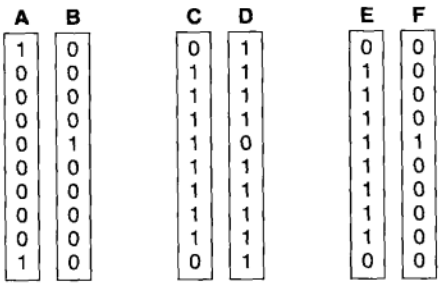


图 6-28　反演操作的结果。向量 **C** 和 **E** 是向量 **A** 的反演，而向量 **D** 是向量 **B** 和 **F** 的反演

定义 6.6　反演性　客观度量 M 在反演操作下是不变的，如果交换频度计数 f_{11} 和 f_{00}、f_{10} 和 f_{01} 它的值保持不变。

在反演操作下保持不变的度量有ϕ系数、几率、κ和集体强度。这些度量可能不适合分析非对称的二元数据。例如，向量**C**和**D**之间的ϕ系数与向量**A**和**B**之间的ϕ系数相等，尽管项c和d同时出现比项a和b同时出现更加频繁。此外，向量**C**和**D**之间的ϕ系数小于向量**E**和**F**之间的ϕ系数，虽然项e和f仅有一次同时出现。前面讨论ϕ系数的局限性时，已经提到该问题。对于非对称的二元数据，使用非反演不变的度量更可取。一些非反演不变的度量包括兴趣因子、*IS*、*PS*和Jaccard系数。

零加性　假定对分析文档集中的一对词（如“*数据*”和“*挖掘*”）之间的联系感兴趣。如果向数据集中添加有关冰下捕鱼的文章，对分析词“*数据*”和“*挖掘*”之间的关联有影响吗？这种向数据集（在此情况下为文档）中添加不相关数据的过程就是所谓的**零加**（null addition）操作。

定义 6.7　零加性　客观度量 M 在零加操作下是不变的，如果增加 f_{00} 而保持相依表中所有其他的频度不变并不影响 M 的值。

对文档分析或购物篮分析这样的应用，期望度量在零加操作下保持不变。否则的话，当添加足够多的不包含所分析词的文档时，被分析词语之间的联系可能会完全消失。满足零加性的度量包括余弦（*IS*）和 Jaccard（ξ）度量，而不满足该性质的度量包括兴趣因子、Piatetsky-Shapiro（*PS*）、几率（α）和 ϕ 系数。

缩放性 表6-16显示了1993年和2004年注册某课程的学生的性别和成绩的相依表。表中的数据表明自1993年以来男生的数量翻了一番，而女生则是原来的3倍。然而，2004年的男生并不比1993年的表现得更好，因为高分和低分男同学的比率保持不变，即3∶4。与之类似，2004年的女同学也并不比1993的表现得更好。尽管抽样分布发生了变化，但是成绩和性别之间的关联预期保持不变。

表 6-16 成绩和性别例子

	男	女	
高	30	20	50
低	40	10	50
	70	30	100

(a) 1993 年的样本数据

	男	女	
高	60	60	120
低	80	30	110
	140	90	230

(b) 2004 年的样本数据

定义 6.8 缩放不变性 客观度量 M 在行/列缩放操作下是不变的，如果 $M(T) = M(T')$，其中，T 是频度计数为$[f_{11}; f_{10}; f_{01}; f_{00}]$的相依表，$T'$是频度计数为$[k_1k_3f_{11}; k_2k_3f_{10}; k_1k_4f_{01}; k_2k_4f_{00}]$的相依表，而 k_1, k_2, k_3, k_4 是正常量。

由表 6-17 可知，只有几率(α)在行和列缩放操作下是不变的。所有其他的度量，例如 ϕ 系数、κ、IS、兴趣因子和集体强度(S)，当相依表的行和列缩放时，它们的值也发生变化。虽然没有讨论非对称度量（如置信度、J 度量、Gini 指标和信任度）的性质，但很明显，在反演和行/列缩放操作下，这些度量不可能保持相同的值，不过它们在零加操作下是不变的。

表 6-17 对称度量的性质

符号	度 量	反演	零加	缩放
ϕ	ϕ系数	Yes	No	No
α	几率	Yes	No	Yes
κ	Cohen 度量	Yes	No	No
I	兴趣因子	No	No	No
IS	余弦	No	Yes	No
PS	Piatetsky-Shapiro 度量	Yes	No	No
S	集体强度	Yes	No	No
ζ	Jaccard	No	Yes	No
h	全置信度	No	No	No
s	支持度	No	No	No

6.7.2 多个二元变量的度量

表 6-11 和表 6-12 显示的度量都是针对一对二元变量定义的，例如，2-项集或关联规则。然而，其中某些也可以应用于较大的项集，如支持度和全置信度（all-confidence）。其他度量（如兴趣因子、IS、PS 和 Jaccard 系数）使用多维相依表中的频率，可以扩展到多个变量。例如，表 6-18 显示了 a, b 和 c 的 3 维相依表。表中每个表目 f_{ijk} 都表示包含项 a, b 和 c 的某种组合的事务数。例如，f_{101} 表示包含 a 和 c 但不包含 b 的事务数。另一方面，边缘频率 f_{i+1} 表示包含项 a 和 c 而不管是否包含项 b 的事务数。

表 6-18　一个三维相关性表的例子

c	b	$\bar{b}$	
a	f_{111}	f_{101}	f_{1+1}
$\bar{a}$	f_{011}	f_{001}	f_{0+1}
	f_{+11}	f_{+01}	f_{++1}

$\bar{c}$	b	$\bar{b}$	
a	f_{110}	f_{100}	f_{1+0}
$\bar{a}$	f_{010}	f_{000}	f_{0+0}
	f_{+10}	f_{+00}	f_{++0}

给定一个 k-项集$\{i_1, i_2, \cdots, i_k\}$，统计独立性条件可以定义如下：

$$f_{i_1 i_2 \cdots i_k} = \frac{f_{i_1+\cdots+} \times f_{+i_2 \cdots+} \times \cdots \times f_{++\cdots i_k}}{N^{k-1}} \tag{6-12}$$

利用该定义，可以扩展基于背离统计独立性的客观度量（如兴趣因子（I）和 PS）到多个变量：

$$I = \frac{N^{k-1} \times f_{i_1 i_2 \cdots i_k}}{f_{i_1+\cdots+} \times f_{+i_2 \cdots+} \times \cdots \times f_{++\cdots i_k}}$$

$$PS = \frac{f_{i_1 i_2 \cdots i_k}}{N} - \frac{f_{i_1+\cdots+} \times f_{+i_2 \cdots+} \times \cdots \times f_{++\cdots i_k}}{N^k}$$

另一种方法是，将客观度量定义为模式中项对之间关联的最大值、最小值或平均值。例如，给定 k-项集 $X = \{i_1, i_2, \cdots, i_k\}$，可以将 X 的ϕ系数定义为 X 中所有项对(i_p, i_q)之间的ϕ系数的平均值。然而，由于该度量只考虑逐对之间的关联，所以它可能不能捕获模式中的隐含联系。

由于数据中存在部分关联，多维相依表的分析更加复杂。例如，根据特定变量的值，某些关联可能出现或不出现。这个问题就是**辛普森悖论**（Simpson's paradox），将在下一节中介绍。可以使用更复杂的统计技术（如对数线性模型）来分析这种联系，但是这些技术已经超出了本书的范围。

6.7.3　辛普森悖论

解释变量之间的关联时要特别小心，因为观察到的联系可能受其他混淆因素的影响，这些因素，即没有包括在分析中的隐藏变量。在某些情况下，隐藏的变量可能会导致观察到的一对变量之间的联系消失或逆转方向，这种现象就是所谓的辛普森悖论。用下面的例子解释这种悖论的性质。

考虑高清晰度电视（HDTV）销售和健身器销售之间的联系，如表 6-19 所示。规则{买 HDTV =是}→{买健身器 = 是}的置信度是 99/180 = 55%，而规则{买 HDTV = 否}→{买健身器 = 是}的置信度是 54/120 = 45%。这些规则暗示，购买了高清晰度电视的顾客比那些没有购买高清晰度电视的顾客更有可能购买健身器。

表 6-19　高清晰度电视和健身器销售之间的 2 路相依表

买 HDTV	买健身器		
	是	否	
是	99	81	180
否	54	66	120
	153	147	300

然而，进一步深入分析表明这些商品的销售取决于顾客是大学生或还是在职人员。表6-20汇总了大学生和在职人员购买高清晰度电视和健身器之间的联系。注意，表中给出的大学生和在职人员的支持度计数的总和等于表6-19中显示的频度。而且，更多是在职人员而不是大学生购买了这些商品的。对于大学生：

$$c(\{买\ HDTV=是\}\rightarrow\{买健身器=是\}) = 1/10 = 10\%$$

$$c(\{买\ HDTV=否\}\rightarrow\{买健身器=是\}) = 4/34 = 11.8\%$$

对于在职人员：

$$c(\{买\ HDTV=是\}\rightarrow\{买健身器=是\}) = 98/170 = 57.7\%$$

$$c(\{买\ HDTV=否\}\rightarrow\{买健身器=是\}) = 50/86 = 58.1\%$$

这些规则暗示，对于每一组顾客，不买高清晰度电视的顾客更可能购买健身器，这与先前由包含两组顾客的数据得到的结论恰好相反。即使采用其他度量（如相关性、几率或兴趣因子）仍然发现在组合数据情况下购买 HDTV 和健身器之间存在正相关，但是在分层数据情况下却存在负相关（参见本章习题 20）。这种关联方向上的逆转就是辛普森悖论。

表 6-20 3 路相依表的例子

顾客组	买 HDTV	买健身器		总数
		是	否	
大学生	是	1	9	10
	否	4	30	34
在职人员	是	98	72	170
	否	50	36	86

这种悖论可以用下面的方法解释。注意，买高清晰度电视的顾客大部分都是在职人员，而且在职人员也是购买健身器的最大人群。由于接近 85%的顾客是在职人员，所以在组合数据情况下观察到的 HDTV 和健身器之间的联系要强于分层情况下的联系。这也可以数学地解释如下。假设

$$a/b < c/d \text{ 并且 } p/q < r/s$$

其中 a/b 和 p/q 是规则 $A\rightarrow B$ 在两个不同层下的置信度，c/d 和 r/s 是规则 $\overline{A}\rightarrow B$ 在这两个层中的置信度。当数据汇集在一起时，在组合数据集中这些规则的置信度值分别是$(a+p)/(b+q)$和$(c+r)/(d+s)$。当$(a+p)/(b+q) > (c+r)/(d+s)$时，辛普森悖论出现，从而导致变量间联系的错误结论。这里的教训是，需要适当的分层才能避免因辛普森悖论产生虚假的模式。例如，大型连锁超市的购物篮数据应该依据商店的位置分层，而不同病人的医疗记录应当按照混杂因素（如年龄和性别等）分层。

6.8 倾斜支持度分布的影响

许多关联分析算法的性能受输入数据的性质的影响。例如，*Apriori*算法的计算复杂度取决于数据中的项数和事务的平均长度等性质。本节讨论另一种重要性质，该性质对关联分析算法的性能和提取模式的质量具有重要影响。更具体地说，关注具有倾斜支持度分布的数据集，其中大多

数项具有较低或中等频率，但是少数项具有很高的频率。

图6-29显示了一个呈现这种分布的实际数据集的例子。该数据取自PUMS（Public Use Microdata Sample）人口普查数据，它包含49046条记录和2113个非对称的二元变量。本节的剩余部分，把非对称二元变量作为项，把记录作为事务。尽管数据集中超过80%的项的支持度小于1%，但是少数项的支持度大于90%。为了解释倾斜支持度分布对频繁项集挖掘的影响，将所有的项按照支持度分为3组，G_1, G_2和G_3。表6-21显示了每一组中包含项的数量。

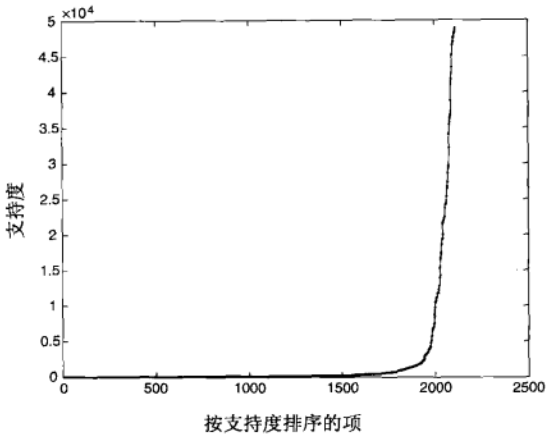


图 6-29　人口普查数据集中项的支持度分布

表 6-21　按照项的支持度将人口普查数据集中的项分组

组	G_1	G_2	G_3
支持度	<1%	1%～90%	>90%
项的数量	1735	358	20

选择合适的支持度阈值挖掘这样的数据集可能相当棘手。如果阈值太高（如 20%），则可能遗漏涉及 G_1 中较低支持度项的模式。在购物篮数据分析中，这种低支持度的项可能对应那些顾客很少买的昂贵商品（如珠宝），但是它们的模式仍然是零售商十分感兴趣的。相反，如果支持度阈值太低，由于下面的原因，挖掘关联模式将变得非常困难：首先，由于支持度阈值太低，已有的关联分析算法所需的计算量和内存需求都将显著增加；其次，由于支持度阈值太低，提取出的关联模式的数量大幅度增加；再次，可能会提取出大量的高频率项（如“牛奶”）与低频率项（如“鱼子酱”）相关联的虚假模式，这样的模式就是所谓的**交叉支持**（cross-support）模式，由于它们的相关性往往太弱，这些模式多半是虚假的。例如，当支持度阈值等于 0.05%时，将会挖掘出 18847 个涉及 G_1 和 G_3 中的项的频繁项对，其中 93%的是交叉支持模式，即包含来自 G_1 和 G_3 的项的模式。从这些交叉支持模式得到的最大相关度是 0.029，远远小于从同一个组中挖掘出的其他频繁模式（高达 1.0）。对前面讨论的其他兴趣度度量也可以做出类似的结论。这个例子表明，当支持度阈值足够低时，可能产生大量弱相关的交叉支持模式。在介绍排除这些模式的方法之前，首先形式地定义交叉支持模式。

定义 6.9　交叉支持模式　交叉支持模式是一个项集 $X=\{i_1, i_2, \cdots, i_k\}$，它的支持度比率

$$r(X)=\frac{\min[s(i_1),s(i_2),\cdots,s(i_k)]}{\max[s(i_1),s(i_2),\cdots,s(i_k)]} \tag{6-13}$$

小于用户指定的阈值 h_c。

例 6.4 假设牛奶的支持度是 70%，糖的支持度是 10%，鱼子酱的是 0.04%。给定 $h_c=0.01$，频繁项集{牛奶，糖，鱼子酱}是一个交叉支持模式，因为它的支持度比率为：

$$r=\frac{\min[0.7,0.1,0.0004]}{\max[0.7,0.1,0.0004]}=\frac{0.0004}{0.7}=0.00058<0.01 \qquad \square$$

现有的度量（如支持度和置信度），都不足以消除交叉支持模式，如图 6-30 显示的数据集所示。假定 $h_c=0.3$，项集$\{p, q\}$、$\{p, r\}$和$\{p, q, r\}$是交叉支持模式，因为它们的支持度比率等于 0.2，小于阈值 0.3。虽然可以采用较高的支持度阈值（如 20%）来消除交叉支持模式，但是，这样却损失了其他有趣的模式，如强关联项集$\{q, r\}$，它的支持度为 16.7%。

p	q	r
0	1	1
1	1	1
1	1	1
1	1	1
1	1	1
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
0	0	0
0	0	0
0	0	0
0	0	0

图 6-30 一个包含 3 个项 p，q 和 r 的事务数据集，其中 p 是高支持度项，q 和 r 是低支持度项

置信度剪裁也无济于事，因为由交叉支持模式提取的规则的置信度可能很高。例如，虽然$\{p, q\}$是一个交叉支持模式，但是规则$\{q\}\rightarrow\{p\}$的置信度却是 80%。交叉支持模式能够产生高置信度的规则并不奇怪，因为其中的项(p)在数据集中频繁出现。因此，p 在许多包含 q 的事务中出现是意料之中的事。同时，即使$\{q, r\}$不是交叉支持模式，规则$\{q\}\rightarrow\{r\}$也具有高置信度。这个例子表明，使用置信度度量很难区别从交叉支持模式或非交叉支持模式中提取的规则。

回到前面的例子，注意到由于包含 p 的大部分事务不包含 q，所以规则$\{p\}\rightarrow\{q\}$的置信度很低。相反，由模式$\{q, r\}$导出的规则$\{r\}\rightarrow\{q\}$却有很高的置信度。这一观察暗示，可以通过检查由给定项集提取的最低置信度规则来检测交叉支持模式。这一论断的证明可以从以下讨论中理解。

(1) 回忆置信度的如下反单调性：

$$conf(\{i_1, i_2\} \rightarrow \{i_3, i_4, \cdots, i_k\}) \leqslant conf(\{i_1 i_2 i_3\} \rightarrow \{i_4, i_5, \cdots, i_k\})$$

该性质表明，把关联规则左边的项移到右边后不会增加规则的置信度。根据这一性质，由频繁项集提取的最低置信度规则的左边仅包含一个项。把左边只有一个项的所有规则的集合用 R_1 表示。

(2) 给定一个频繁项集$\{i_1, i_2, \cdots, i_k\}$，如果 $s(i_j) = \max[s(i_1), s(i_2), \cdots, s(i_k)]$，则规则

$$\{i_j\} \rightarrow \{i_1, i_2, \cdots, i_{j-1}, i_{j+1}, \cdots, i_k\}$$

是 R_1 中具有最小置信度的规则。这一结论直接由置信度是规则的支持度与规则前件支持度的比得到。

(3) 总结以上各点，可以从频繁项集$\{i_1, i_2, \cdots, i_k\}$中得到的最低置信度为：

$$\frac{s(\{i_1, i_2, \cdots, i_k\})}{\max[s(i_1), s(i_2), \cdots, s(i_k)]}$$

这个表达式又称 **h 置信度**（h-confidence）或**全置信度**（all-confidence）度量。由于支持度的反单调性，h 置信度度量的分子受限于频繁项集所有项中最小的支持度。换句话说，项集 $X = \{i_1, i_2, \cdots, i_k\}$的 h 置信度不超过下面表达式：

$$\text{h-confidence}(X) \leqslant \frac{\min[s(i_1), s(i_2), \cdots, s(i_k)]}{\max[s(i_1), s(i_2), \cdots, s(i_k)]}$$

注意 h 置信度的上界与公式（6-13）中支持度比率(r)的等价性。因为交叉支持模式的支持度比率总是小于 h_c，因此这类模式的 h 置信度也一定小于 h_c。

因此，通过确保模式的 h 置信度值超过 h_c 就可以消除交叉支持模式。最后，值得一提的是，使用 h 置信度的好处不仅是消除交叉支持模式。这种度量也是反单调的，即

$$\text{h-confidence}(\{i_1, i_2, \cdots, i_k\}) \geqslant \text{h-confidence}(\{i_1, i_2, \cdots, i_{k+1}\})$$

从而可以将它直接并入挖掘算法。此外，h 置信度能够确保项集中的项之间是强关联的。例如，假定一个项集 X 的 h 置信度是 80%。如果 X 中的一个项出现在某个事务中，则 X 中其他的项至少有 80%的几率属于同一个事务。这种强关联模式又称**超团模式**（hyperclique pattern）。

文献注释

关联规则的挖掘首先是由Agrawal等人在[228, 229]中提出，用来发现购物篮数据事务中各项之间的有趣联系。从那以后，人们进行了广泛的研究，以解决关联分析任务的概念、实现和应用问题。图 6-31 中汇总了该领域各种各样的研究活动。

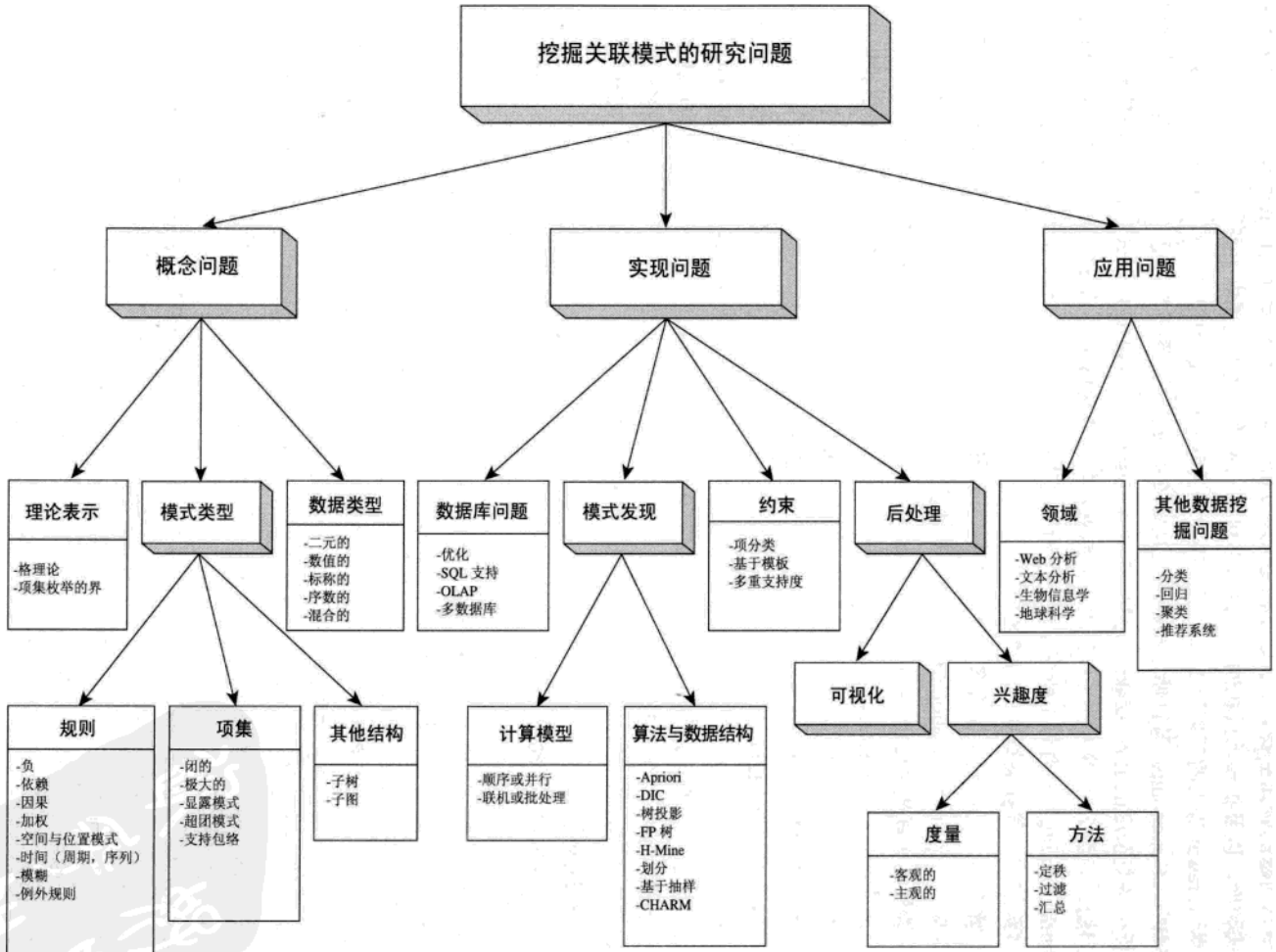


图 6-31　关联分析各种研究活动的汇总

1. 概念问题

概念问题的研究主要集中在建立描述关联分析的理论基础的框架，扩展形式机制，以处理新的模式类型，以及扩展形式机制，以纳入非对称二元数据之外的属性类型。

继 Agrawal 等人的开创性工作之后，在发展关联分析问题的理论方面已有大量的研究成果。在[254]中，Gunopoulos 等证明了挖掘极大频繁项集问题和超图遍历问题之间的关系，并推出关联分析任务复杂度的上界。Zaki 等[334，336]和 Pasquier 等[294]使用形式概念分析研究频繁项集产生问题。随后 Zaki 等人的工作提出了闭频繁项集的理论[336]。Friedman 等在多维空间**凸点搜索**（bump hunting）的背景下研究了关联分析的问题[252]。更具体地说，他们把频繁项集产生看作在多维空间中寻找高概率的稠密区域。

许多年来，已经定义了许多新类型的模式，如轮廓关联规则（profile association rule）[225]、环关联规则（cyclic association rule）[290]、模糊关联规则[273]、例外规则[316]、负关联规则[238, 304]、加权的关联规则[240, 300]、依赖规则[308]、罕见规则（peculiar rule）[340]、事务间关联规则（inter-transaction association rule）[250, 323]和局部分类规则[231, 285]。其他类型的模式包括闭项集[294，336]、极大项集[234]、超团模式[330]、支持包络（support envelope）[314]、显露模式[246]和对比集（contrast set）[233]。关联分析也成功地应用于序列数据[230, 312]、空间数据[266]和基于图的数据[268, 274, 293, 331, 335]。交叉支持模式的概念首先由 Hui 等[330]提出。作者还提出了一种有效的自动删除交叉支持模式的算法，称为超团挖掘（Hyperclique Miner）。

已经做了大量的研究，将最初的关联规则分析扩展到了标称属性[311]、序数属性[281]、区间属性[284]和比率属性[253, 255, 311, 325, 339]。一个关键性的问题是如何定义这些属性的支持度。Steinbach 等[315]提出了一种方法，将传统的支持度概念扩展到更一般的模式和属性类型。

2. 实现问题

这一领域的研究活动主要涉及：(1)将挖掘能力集成到现有的数据库技术中；(2)产生高效的可伸缩的挖掘算法；(3)处理用户指定的或领域特定的约束；(4)提取模式的后处理。

将关联分析集成到现有的数据库技术中有着许多优点。首先，可以利用数据库系统的索引和查询处理机制；其次，可以利用 DBMS 对可伸缩性、检查点和并行性的支持[301]。Houtsma 等[265]提出的 SETM 算法是早期通过 SQL 查询支持关联规则挖掘的算法之一。从那时起，产生了许多的算法，用以在数据库系统中提供关联规则挖掘能力。例如，DMQL[258]和 M-SQL[267]查询语言用新的关联规则挖掘操作扩展了基本 SQL。挖掘规则操作（Mine Rule operator）[283]是一种表达能力很强的 SQL 操作，可以处理聚集属性和项分层结构。Tsur 等[322] 提出了称作**查询群**（query flock）的挖掘关联规则的产生-测试的方法。Chen 等[241]提出了分布的、基于 OLAP 的挖掘多层关联规则的框架。

Dunkel 和 Soparkar[248]研究了 *Apriori* 算法的时间和存储复杂度。Han 等[259]提出了 FP 增长算法。挖掘频繁项集的其他算法包括 Park 等[292]提出的 DHP 算法和 Savasere 等[303]提出的划分算法。Toivonen[320]提出了基于抽样的频繁项集产生算法，这种算法只需要扫描一次数据集，但是它产生相对较多的候选项集。动态项集计数（dynamic itemset counting, DIC）算法[239]只需要扫描数据集 1.5 次，并且它产生的候选项集少于基于抽样的算法。其他著名的算法包括树投影算法[223]和 H-Mine [295]。关于频繁项集产生算法的综述可以在文献[226, 262]中找到。频繁项集挖掘实现库 FIMI（http://fimi.cs.helsinki.fi）提供了有用的数据集和挖掘算法。许多作者都提出了挖掘关联模式的并行算法[224, 256, 287, 306, 337]。这些算法的综述可以在[333]中找到。

Hidber[260]和 Cheung 等[242]还提出了挖掘关联规则算法的联机和增量版本。

Srikant等[313]考虑了在布尔约束下挖掘关联规则的问题。例如，给定诸如((饼干∧牛奶)∨(descendents(饼干) ∧ ¬ancestors(小麦面包))的约束，算法寻找包含饼干和牛奶的规则，或包含饼干的后代而不包含小麦面包的祖先的规则。Singh等[310]和Ng等[288]提出了另外一种基于约束的关联规则挖掘技术。也可以对不同项集的支持度施加约束。Wang等[324]、Liu等[279]和Seno等[305]研究了这个问题。

关联分析的潜在问题是现在的算法可能产生大量的模式。为了解决这个问题，提出了模式定秩、汇总和模式过滤方法。Toivonen 等[321]提出使用**结构规则覆盖**（structural rule cover）删除冗余规则、并使用聚类对剩下规则分组的思想。Liu 等[280]使用统计χ^2检验排除虚假模式，并采用一种称作**方向设置规则**（direction setting rule）的模式子集汇总剩下的模式。许多研究者都考察了使用客观度量过滤模式的方法，包括 Brin 等[238]、Bayardo 和 Agrawal [235]、Aggarwal 和 Yu[227]、DuMouchel 和 Pregibon[247]。Piatetsky-Shapiro[297]、Kamber 和 Singhal[270]、Hilderman 和 Hamilton[261]、Tan 等[318]分析了这些度量的性质。“成绩-性别”例子用于强调行、列缩放不变性的重要性，该例很大程度上是受 Mosteller 在[286]中的讨论的影响。同时，“喝茶-喝咖啡”的例子用以解释置信度的局限性，该例是受 Brin 等[238]给出的例子的启发。由于置信度的局限性，Brin 等[238]提出使用兴趣因子作为兴趣度度量的思想。Omiecinski[289]提出了全置信度度量观点。Xiong 等[330]引进交叉支持性质，并表明全置信度度量可以用来删除交叉支持模式。使用支持度之外的客观度量的主要困难在于它们不具有单调性，这使得它们很难直接应用到挖掘算法中。Xiong 等[328]通过引进ϕ系数的上界函数，提出了一种高效的挖掘相关性的方法。虽然ϕ系数是非单调的，但是它有一个上界表达式，可以用来有效地挖掘强相关的项对。

Fabris 和 Freitas[249]提出了一种方法，通过检测辛普森悖论[309]发现有趣的关联。Megiddo 和 Srikant[282]也介绍了一种方法，采用假设检验来验证提取的模式。为了避免因多重比较问题而产生虚假模式，提出了一种基于再抽样的技术。Bolton 等[237]使用 Benjamini - Hochberg[236]和 Bonferroni 校正方法调整从购物篮数据中挖掘出的模式的 p 值。Webb[326]和 Zhang 等[338]提出了另外一种多重比较问题的方法。

许多研究者都研究了主观度量在关联分析中的应用。Silberschatz 和 Tuzhilin[307]提出了从主观角度判断一个规则是否是有趣的两条原则。Liu 等[277]提出了非期望条件规则的概念。Cooley 等[243]使用 Dempster-Shafer 理论分析组合软置信集的思想，并使用这种方法识别 Web 数据中相反或新颖的关联模式。其他方法包括使用贝叶斯信念网络[269]和基于近邻的信息[245]识别主观上有趣的模式。

可视化也有助于用户快速地掌握发现模式的基本结构。许多商业数据挖掘工具把规则的完全集（满足支持度和置信度阈值）以二维图的形式显示，其中每个轴对应于这个规则的前件或后件项集。Hofmann 等[263]提出使用 Mosaic 图和双层图显示关联规则。这种方法不仅仅能够显示一条特定的规则，而且还显示规则的前件项集和后件项集之间的相依表。然而，这种技术假定规则的后件只有一个属性。

3. 应用问题

关联分析已经应用于各种各样的应用领域，如 Web 挖掘[296, 317]、文档分析[264]、通信警告分析[271]、网络入侵检测[232, 244, 275]和生物信息学[302, 327]。文献[298, 299, 319]考察了关联和相关模式分析在地球科学的研究中的应用。

关联模式也已经应用到其他学习问题，如分类[276, 278]、回归[291]和聚类[257, 329, 332]。Freitas 在他的意见书[251]中对分类和关联规则挖掘进行了比较。许多作者研究了将关联模式应用于聚类，包括 Han 等[257]、Kosters 等[272]、Yang 等[332]和 Xiong 等[329]。

参考文献

[223] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. A Tree Projection Algorithm for Generation of Frequent Itemsets. *Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining)*, 61(3):350 - 371, 2001.

[224] R. C. Agarwal and J. C. Shafer. Parallel Mining of Association Rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):962 - 969, March 1998.

[225] C. C. Aggarwal, Z. Sun, and P. S. Yu. Online Generation of Profile Association Rules. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 129 - 133, New York, NY, August 1996.

[226] C. C. Aggarwal and P. S. Yu. Mining Large Itemsets for Association Rules. *Data Engineering Bulletin*, 21(1):23 - 31, March 1998.

[227] C. C. Aggarwal and P. S. Yu. Mining Associations with the Collective Strength Approach. *IEEE Trans. on Knowledge and Data Engineering*, 13(6):863 - 873, January/February 2001.

[228] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5:914 - 925, 1993.

[229] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Intl. Conf. Management of Data*, pages 207 - 216, Washington, DC, 1993.

[230] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *Proc. of Intl. Conf. On Data Engineering*, pages 3 - 14, Taipei, Taiwan, 1995.

[231] K. Ali, S. Manganaris, and R. Srikant. Partial Classification using Association Rules. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 115 - 118, Newport Beach, CA, August 1997.

[232] D. Barbará, J. Couto, S. Jajodia, and N. Wu. ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection. *SIGMOD Record*, 30(4):15 - 24, 2001.

[233] S. D. Bay and M. Pazzani. Detecting Group Differences: Mining Contrast Sets. *Data Mining and Knowledge Discovery*, 5(3):213 - 246, 2001.

[234] R. Bayardo. Efficiently Mining Long Patterns from Databases. In *Proc. of 1998 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 85 - 93, Seattle, WA, June 1998.

[235] R. Bayardo and R. Agrawal. Mining the Most Interesting Rules. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 145 - 153, San Diego, CA, August 1999.

[236] Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal Royal Statistical Society B*, 57 (1):289 - 300, 1995.

[237] R. J. Bolton, D. J. Hand, and N. M. Adams. Determining Hit Rate in Pattern Search. In *Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, pages 36 - 48, London, UK, September 2002.

[238] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proc. ACM SIGMOD Intl. Conf. Management of Data*, pages 265 - 276, Tucson, AZ, 1997.

[239] S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic Itemset Counting and Implication Rules for market basket data. In *Proc. of 1997 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 255 - 264, Tucson, AZ, June 1997.

[240] C. H. Cai, A. Fu, C. H. Cheng, and W. W. Kwong. Mining Association Rules with Weighted Items. In *Proc. of IEEE Intl. Database Engineering and Applications Symp.*, pages 68 - 77, Cardiff, Wales, 1998.

[241] Q. Chen, U. Dayal, and M. Hsu. A Distributed OLAP infrastructure for E-Commerce. In *Proc. of the 4th IFCIS Intl. Conf. on Cooperative Information Systems*, pages 209 - 220, Edinburgh, Scotland, 1999.

[242] D. C. Cheung, S. D. Lee, and B. Kao. A General Incremental Technique for Maintaining Discovered Association Rules. In *Proc. of the 5th Intl. Conf. on Database Systems for Advanced Applications*, pages 185 - 194, Melbourne, Australia, 1997.

[243] R. Cooley, P. N. Tan, and J. Srivastava. Discovery of Interesting Usage Patterns from Web Data. In M. Spiliopoulou and B. Masand, editors, *Advances in Web Usage Analysis and User Profiling*, volume 1836, pages 163 - 182. Lecture Notes in Computer Science, 2000.

[244] P. Dokas, L. Ertöz, V. Kumar, A. Lazarevic, J. Srivastava, and P. N. Tan. Data Mining for Network Intrusion Detection. In *Proc. NSF Workshop on Next Generation Data Mining*, Baltimore, MD, 2002.

[245] G. Dong and J. Li. Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. In *Proc. of the 2nd Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 72 - 86, Melbourne, Australia, April 1998.

[246] G. Dong and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 43 - 52, San Diego, CA, August 1999.

[247] W. DuMouchel and D. Pregibon. Empirical Bayes Screening for Multi-Item Associations. In *Proc. of the 7th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 67 - 76, San Francisco, CA, August 2001.

[248] B. Dunkel and N. Soparkar. Data Organization and Access for Efficient Data Mining. In *Proc. of the 15th Intl. Conf. on Data Engineering*, pages 522 - 529, Sydney, Australia, March 1999.

[249] C. C. Fabris and A. A. Freitas. Discovering surprising patterns by detecting occurrences of Simpson's paradox. In *Proc. of the 19th SGES Intl. Conf. on Knowledge-Based Systems and Applied Artificial Intelligence*, pages 148 - 160, Cambridge, UK, December 1999.

[250] L. Feng, H. J. Lu, J. X. Yu, and J. Han. Mining inter-transaction associations with templates. In *Proc. of the 8th Intl. Conf. on Information and Knowledge Management*, pages 225 - 233, Kansas City, Missouri, Nov 1999.

[251] A. A. Freitas. Understanding the crucial differences between classification and discovery of association rules—a position paper. *SIGKDD Explorations*, 2(1):65 - 69, 2000.

[252] J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123 - 143, April 1999.

[253] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining Optimized Association Rules for Numeric Attributes. In *Proc. of the 15th Symp. on Principles of Database Systems*, pages 182 - 191, Montreal, Canada, June 1996.

[254] D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data Mining, Hypergraph Transversals, and Machine Learning. In *Proc. of the 16th Symp. on Principles of Database Systems*, pages 209 - 216, Tucson, AZ, May 1997.

[255] E.-H. Han, G. Karypis, and V. Kumar. Min-Apriori: An Algorithm for Finding Association Rules in Data with Continuous Attributes. http://www.cs.umn.edu/~han, 1997.

[256] E.-H. Han, G. Karypis, and V. Kumar. Scalable Parallel Data Mining for Association Rules. In *Proc. of 1997 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 277 - 288, Tucson, AZ, May 1997.

[257] E.-H. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering Based on Association Rule Hypergraphs. In *Proc. of the 1997 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Tucson, AZ, 1997.

[258] J. Han, Y. Fu, K. Koperski, W. Wang, and O. R. Zaïane. DMQL: A data mining query language for relational databases. In *Proc. of the 1996 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Montreal, Canada, June 1996.

[259] J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In *Proc. ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00)*, pages 1 - 12, Dallas, TX, May

2000.

[260] C. Hidber. Online Association Rule Mining. In *Proc. of 1999 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 145 - 156, Philadelphia, PA, 1999.

[261] R. J. Hilderman and H. J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic Publishers, 2001.

[262] J. Hipp, U. Guntzer, and G. Nakhaeizadeh. Algorithms for Association Rule Mining—A General Survey. *SigKDD Explorations*, 2(1):58 - 64, June 2000.

[263] H. Hofmann, A. P. J. M. Siebes, and A. F. X. Wilhelm. Visualizing Association Rules with Interactive Mosaic Plots. In *Proc. of the 6th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 227 - 235, Boston, MA, August 2000.

[264] J. D. Holt and S. M. Chung. Efficient Mining of Association Rules in Text Databases. In *Proc. of the 8th Intl. Conf. on Information and Knowledge Management*, pages 234 - 242, Kansas City, Missouri, 1999.

[265] M. Houtsma and A. Swami. Set-oriented Mining for Association Rules in Relational Databases. In *Proc. of the 11th Intl. Conf. on Data Engineering*, pages 25 - 33, Taipei, Taiwan, 1995.

[266] Y. Huang, S. Shekhar, and H. Xiong. Discovering Co-location Patterns from Spatial Datasets: A General Approach. *IEEE Trans. on Knowledge and Data Engineering*, 16 (12):1472 - 1485, December 2004.

[267] T. Imielinski, A. Virmani, and A. Abdulghani. DataMine: Application Programming Interface and Query Language for Database Mining. In *Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 256 - 262, Portland, Oregon, 1996.

[268] A. Inokuchi, T. Washio, and H. Motoda. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In *Proc. of the 4th European Conf. of Principles and Practice of Knowledge Discovery in Databases*, pages 13 - 23, Lyon, France, 2000.

[269] S. Jaroszewicz and D. Simovici. Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 178 - 186, Seattle, WA, August 2004.

[270] M. Kamber and R. Shinghal. Evaluating the Interestingness of Characteristic Rules. In *Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 263 - 266, Portland, Oregon, 1996.

[271] M. Klemettinen. *A Knowledge Discovery Methodology for Telecommunication Network Alarm Databases*. PhD thesis, University of Helsinki, 1999.

[272] W. A. Kosters, E. Marchiori, and A. Oerlemans. Mining Clusters with Association Rules. In *The 3rd Symp. on Intelligent Data Analysis (IDA99)*, pages 39 - 50, Amsterdam, August 1999.

[273] C. M. Kuok, A. Fu, and M. H. Wong. Mining Fuzzy Association Rules in Databases. *ACM SIGMOD Record*, 27(1):41 - 46, March 1998.

[274] M. Kuramochi and G. Karypis. Frequent Subgraph Discovery. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 313 - 320, San Jose, CA, November 2001.

[275] W. Lee, S. J. Stolfo, and K. W. Mok. Adaptive Intrusion Detection: A Data Mining Approach. *Artificial Intelligence Review*, 14(6):533 - 567, 2000.

[276] W. Li, J. Han, and J. Pei. CMAR: Accurate and Efficient Classification Based on Multiple Class-association Rules. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 369 - 376, San Jose, CA, 2001.

[277] B. Liu, W. Hsu, and S. Chen. Using General Impressions to Analyze Discovered Classification Rules. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 31 - 36, Newport Beach, CA, August 1997.

[278] B. Liu, W. Hsu, and Y. Ma. Integrating Classification and Association Rule Mining. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 80 - 86, New York, NY, August 1998.

[279] B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 125 - 134, San Diego, CA, August 1999.

[280] B. Liu, W. Hsu, and Y. Ma. Pruning and Summarizing the Discovered Associations. In *Proc. of the 5th*

Intl. Conf. on Knowledge Discovery and Data Mining, pages 125 - 134, San Diego, CA, August 1999.

[281] A. Marcus, J. I. Maletic, and K.-I. Lin. Ordinal association rules for error identification in data sets. In *Proc. of the 10th Intl. Conf. on Information and Knowledge Management*, pages 589 - 591, Atlanta, GA, October 2001.

[282] N. Megiddo and R. Srikant. Discovering Predictive Association Rules. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 274 - 278, New York, August 1998.

[283] R. Meo, G. Psaila, and S. Ceri. A New SQL-like Operator for Mining Association Rules. In *Proc. of the 22nd VLDB Conf.*, pages 122 - 133, Bombay, India, 1996.

[284] R. J. Miller and Y. Yang. Association Rules over Interval Data. In *Proc. of 1997 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 452 - 461, Tucson, AZ, May 1997.

[285] Y. Morimoto, T. Fukuda, H. Matsuzawa, T. Tokuyama, and K. Yoda. Algorithms for mining association rules for binary segmentations of huge categorical databases. In *Proc. of the 24th VLDB Conf.*, pages 380 - 391, New York, August 1998.

[286] F. Mosteller. Association and Estimation in Contingency Tables. *Journal of the American Statistical Association*, 63:1 - 28, 1968.

[287] A. Mueller. Fast sequential and parallel algorithms for association rule mining: A comparison. Technical Report CS-TR-3515, University of Maryland, August 1995.

[288] R. T. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory Mining and Pruning Optimizations of Constrained Association Rules. In *Proc. of 1998 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 13 - 24, Seattle, WA, June 1998.

[289] E. Omiecinski. Alternative Interest Measures for Mining Associations in Databases. *IEEE Trans. on Knowledge and Data Engineering*, 15(1):57 - 69, January/February 2003.

[290] B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic Association Rules. In *Proc. of the 14th Intl. Conf. on Data Eng.*, pages 412 - 421, Orlando, FL, February 1998.

[291] A. Ozgur, P. N. Tan, and V. Kumar. RBA: An Integrated Framework for Regression based on Association Rules. In *Proc. of the SIAM Intl. Conf. on Data Mining*, pages 210 - 221, Orlando, FL, April 2004.

[292] J. S. Park, M.-S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. *SIGMOD Record*, 25(2):175 - 186, 1995.

[293] S. Parthasarathy and M. Coatney. Efficient Discovery of Common Substructures in Macromolecules. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 362 - 369, Maebashi City, Japan, December 2002.

[294] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. of the 7th Intl. Conf. on Database Theory (ICDT'99)*, pages 398 - 416, Jerusalem, Israel, January 1999.

[295] J. Pei, J. Han, H. J. Lu, S. Nishio, and S. Tang. H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 441 - 448, San Jose, CA, November 2001.

[296] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu. Mining Access Patterns Efficiently from Web Logs. In *Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 396 - 407, Kyoto, Japan, April 2000.

[297] G. Piatetsky-Shapiro. Discovery, Analysis and Presentation of Strong Rules. In G. Piatetsky-Shapiro and W. Frawley, editors, *Knowledge Discovery in Databases*, pages 229 - 248. MIT Press, Cambridge, MA, 1991.

[298] C. Potter, S. Klooster, M. Steinbach, P. N. Tan, V. Kumar, S. Shekhar, and C. Carvalho. Understanding Global Teleconnections of Climate to Regional Model Estimates of Amazon Ecosystem Carbon Fluxes. *Global Change Biology*, 10(5):693 - 703, 2004.

[299] C. Potter, S. Klooster, M. Steinbach, P. N. Tan, V. Kumar, S. Shekhar, R. Myneni, and R. Nemani. Global Teleconnections of Ocean Climate to Terrestrial Carbon Flux. *J. Geophysical Research*, 108(D17), 2003.

[300] G. D. Ramkumar, S. Ranka, and S. Tsur. Weighted Association Rules: Model and Algorithm.

http://www.cs.ucla.edu/~czdemo/tsur/, 1997.

[301] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating Mining with Relational Database Systems: Alternatives and Implications. In *Proc. of 1998 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 343 - 354, Seattle, WA, 1998.

[302] K. Satou, G. Shibayama, T. Ono, Y. Yamamura, E. Furuichi, S. Kuhara, and T. Takagi. Finding Association Rules on Heterogeneous Genome Data. In *Proc. of the Pacific Symp. on Biocomputing*, pages 397 - 408, Hawaii, January 1997.

[303] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *Proc. of the 21st Int. Conf. on Very Large Databases (VLDB'95)*, pages 432 - 444, Zurich, Switzerland, September 1995.

[304] A. Savasere, E. Omiecinski, and S. Navathe. Mining for Strong Negative Associations in a Large Database of Customer Transactions. In *Proc. of the 14th Intl. Conf. on Data Engineering*, pages 494 - 502, Orlando, Florida, February 1998.

[305] M. Seno and G. Karypis. LPMiner: An Algorithm for Finding Frequent Itemsets Using Length-Decreasing Support Constraint. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 505 - 512, San Jose, CA, November 2001.

[306] T. Shintani and M. Kitsuregawa. Hash based parallel algorithms for mining association rules. In *Proc of the 4th Intl. Conf. on Parallel and Distributed Info. Systems*, pages 19 - 30, Miami Beach, FL, December 1996.

[307] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. on Knowledge and Data Engineering*, 8(6):970 - 974, 1996.

[308] C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39 - 68, 1998.

[309] E.-H. Simpson. The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society*, B(13):238 - 241, 1951.

[310] L. Singh, B. Chen, R. Haight, and P. Scheuermann. An Algorithm for Constrained Association Rule Mining in Semi-structured Data. In *Proc. of the 3rd Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 148 - 158, Beijing, China, April 1999.

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

[312] R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proc. of the 5th Intl Conf. on Extending Database Technology (EDBT'96)*, pages 18 - 32, Avignon, France, 1996.

[313] R. Srikant, Q. Vu, and R. Agrawal. Mining Association Rules with Item Constraints. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 67 - 73, Newport Beach, CA, August 1997.

[314] M. Steinbach, P. N. Tan, and V. Kumar. Support Envelopes: A Technique for Exploring the Structure of Association Patterns. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 296 - 305, Seattle, WA, August 2004.

[315] M. Steinbach, P. N. Tan, H. Xiong, and V. Kumar. Extending the Notion of Support. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 689 - 694, Seattle, WA, August 2004.

[316] E. Suzuki. Autonomous Discovery of Reliable Exception Rules. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 259 - 262, Newport Beach, CA, August 1997.

[317] P. N. Tan and V. Kumar. Mining Association Patterns in Web Usage Data. In *Proc. of the Intl. Conf. on Advances in Infrastructure for e-Business, e-Education, e-Science and e-Medicine on the Internet*, L' Aquila, Italy, January 2002.

[318] P. N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. In *Proc. of the 8th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 32 - 41, Edmonton, Canada, July 2002.

[319] P. N. Tan, M. Steinbach, V. Kumar, S. Klooster, C. Potter, and A. Torregrosa. Finding Spatio-Temporal Patterns in Earth Science Data. In *KDD 2001 Workshop on Temporal Data Mining*,

San Francisco, CA, 2001.

[320] H. Toivonen. Sampling Large Databases for Association Rules. In *Proc. of the 22nd VLDB Conf.*, pages 134 - 145, Bombay, India, 1996.

[321] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hatonen, and H. Mannila. Pruning and Grouping Discovered Association Rules. In *ECML-95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, pages 47 - 52, Heraklion, Greece, April 1995.

[322] S. Tsur, J. Ullman, S. Abiteboul, C. Clifton, R. Motwani, S. Nestorov, and A. Rosenthal. Query Flocks: A Generalization of Association Rule Mining. In *Proc. of 1998 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 1 - 12, Seattle, WA, June 1998.

[323] A. Tung, H. J. Lu, J. Han, and L. Feng. Breaking the Barrier of Transactions: Mining Inter-Transaction Association Rules. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 297 - 301, San Diego, CA, August 1999.

[324] K. Wang, Y. He, and J. Han. Mining Frequent Itemsets Using Support Constraints. In *Proc. of the 26th VLDB Conf.*, pages 43 - 52, Cairo, Egypt, September 2000.

[325] K. Wang, S. H. Tay, and B. Liu. Interestingness-Based Interval Merger for Numeric Association Rules. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 121 - 128, New York, NY, August 1998.

[326] G. I. Webb. Preliminary investigations into statistically valid exploratory rule discovery. In *Proc. of the Australasian Data Mining Workshop (AusDM03)*, Canberra, Australia, December 2003.

[327] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook. Identification of Functional Modules in Protein Complexes via Hyperclique Pattern Discovery. In *Proc. of the Pacific Symposium on Biocomputing, (PSB 2005)*, Maui, January 2005.

[328] H. Xiong, S. Shekhar, P. N. Tan, and V. Kumar. Exploiting a Support-based Upper Bound of Pearson's Correlation Coefficient for Efficiently Identifying Strongly Correlated Pairs. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 334 - 343, Seattle, WA, August 2004.

[329] H. Xiong, M. Steinbach, P. N. Tan, and V. Kumar. HICAP: Hierarchial Clustering with Pattern Preservation. In *Proc. of the SIAM Intl. Conf. on Data Mining*, pages 279 - 290, Orlando, FL, April 2004.

[330] H. Xiong, P. N. Tan, and V. Kumar. Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution. In *Proc. of the 2003 IEEE Intl. Conf. on Data Mining*, pages 387 - 394, Melbourne, FL, 2003.

[331] X. Yan and J. Han. gSpan: Graph-based Substructure Pattern Mining. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 721 - 724, Maebashi City, Japan, December 2002.

[332] C. Yang, U. M. Fayyad, and P. S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In *Proc. of the 7th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 194 - 203, San Francisco, CA, August 2001.

[333] M. J. Zaki. Parallel and Distributed Association Mining: A Survey. *IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining*, 7(4):14 - 25, December 1999.

[334] M. J. Zaki. Generating Non-Redundant Association Rules. In *Proc. of the 6th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 34 - 43, Boston, MA, August 2000.

[335] M. J. Zaki. Efficiently mining frequent trees in a forest. In *Proc. of the 8th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 71 - 80, Edmonton, Canada, July 2002.

[336] M. J. Zaki and M. Orihara. Theoretical foundations of association rules. In *Proc. of the 1998 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Seattle, WA, June 1998.

[337] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association Rules. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 283 - 286, Newport Beach, CA, August 1997.

[338] H. Zhang, B. Padmanabhan, and A. Tuzhilin. On the Discovery of Significant Statistical Quantitative Rules. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 374 - 383,

Seattle, WA, August 2004.

[339] Z. Zhang, Y. Lu, and B. Zhang. An Effective Partioning-Combining Algorithm for Discovering Quantitative Association Rules. In *Proc. of the 1st Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Singapore, 1997.

[340] N. Zhong, Y. Y. Yao, and S. Ohsuga. Peculiarity Oriented Multi-database Mining. In *Proc. of the 3rd European Conf. of Principles and Practice of Knowledge Discovery in Databases*, pages 136 - 146, Prague, Czech Republic, 1999.

习　题

1. 对于下面每一个问题，请在购物篮领域举出一个满足下面条件的关联规则的例子。此外，指出这些规则是否是主观上有趣的。
   (a) 具有高支持度和高置信度的规则。
   (b) 具有相当高的支持度却有较低置信度的规则。
   (c) 具有低支持度和低置信度的规则。
   (d) 具有低支持度和高置信度的规则。
2. 考虑表 6-22 中显示的数据集。

表 6-22　购物篮事务的例子

顾客 ID	事务 ID	购买项
1	0001	{a,d,e}
1	0024	{a,b,c,e}
2	0012	{a,b,d,e}
2	0031	{a,c,d,e}
3	0015	{b,c,e}
3	0022	{b,d,e}
4	0029	{c,d}
4	0040	{a,b,c}
5	0033	{a,d,e}
5	0038	{a,b,e}

   (a) 将每个事务 ID 视为一个购物篮，计算项集{e}, {b, d}和{b, d, e}的支持度。
   (b) 使用(a)的计算结果，计算关联规则{b, d}→{e}和{e}→{b, d}的置信度。置信度是对称的度量吗？
   (c) 将每个顾客 ID 作为一个购物篮，重复(a)。应当将每个项看作一个二元变量（如果一个项在顾客的购买事务中至少出现了一次，则为 1；否则，为 0）。
   (d) 使用(c)的计算结果，计算关联规则{b, d}→{e}和{e}→{b, d}的置信度。
   (e) 假定 s_1 和 c_1 是将每个事务 ID 作为一个购物篮时关联规则 r 的支持度和置信度，而 s_2 和 c_2 是将每个顾客 ID 作为一个购物篮时关联规则 r 的支持度和置信度。讨论 s_1 和 s_2 或 c_1 和 c_2 之间是否存在某种关系？
3. (a) 规则∅→A 和 A→∅的置信度是多少？
   (b) 令 c_1, c_2 和 c_3 分别是规则{p}→{q}, {p}→{q, r}和{p, r}→{q}的置信度。如果假定 c_1, c_2 和 c_3 有不同的值，那么 c_1, c_2 和 c_3 之间可能存在什么关系？哪个规则的置信度最低？
   (c) 假定(b)中的规则具有相同的置信度，重复(b)的分析。哪个规则的置信度最高？
   (d) 传递性：假定规则A→B和B→C的置信度都大于某个阈值*minconf*。规则A→C可能具有

小于*minconf*的置信度吗？

4. 对于下列每种度量，判断它是单调的、反单调的或非单调的（即既不是单调的，也不是反单调的）。

例如：支持度 $s=\sigma(X)/|T|$是反单调的，因为只要 $X\subset Y$，就有 $s(X)\geqslant s(Y)$。

(a) 特征规则是形如$\{p\}\rightarrow\{q_1, q_2,\cdots, q_n\}$的规则，其中规则的前件只有一个项。一个大小为 k 的项集能够产生 k 个特征规则。令ζ是由给定项集产生的所有特征规则的最小置信度：

$$\zeta(\{p_1, p_2,\cdots, p_k\})=\min[c(\{p_1\}\rightarrow\{p_2, p_3,\cdots, p_k\}),\cdots,c(\{p_k\}\rightarrow\{p_1, p_3,\cdots, p_{k-1}\})]$$

ζ是单调的、反单调的或非单调的？

(b) 区分规则是形如$\{p_1, p_2,\cdots, p_n\}\rightarrow\{q\}$的规则，其中规则的后件只有一个项。一个大小为 k 的项集能够产生 k 个区分规则。令η是由给定项集产生的所有区分规则的最小置信度：

$$\eta(\{p_1,p_2,\cdots,p_k\})=\min[c(\{p_2,p_3,\cdots,p_k\}\rightarrow\{p_1\}),\cdots, c(\{p_1,p_2,\cdots, p_{k-1}\}\rightarrow\{p_k\})]$$

η是单调的、反单调的或非单调的？

(c) 将最小值函数改为最大值函数，重做(a)和(b)的分析。

5. 证明公式（6-3）。（提示：首先，计算创建形成规则左部项集的方法数；然后，对每个选定为规则左部的 k 项集，计算选择剩下的 $d-k$ 个项形成规则右部的方法数。）

6. 考虑表 6-23 中显示的购物篮事务。

表 6-23　购物篮事务

事务 ID	购买项
1	{牛奶，啤酒，尿布}
2	{面包，黄油，牛奶}
3	{牛奶，尿布，饼干}
4	{面包，黄油，饼干}
5	{啤酒，饼干，尿布}
6	{牛奶，尿布，面包，黄油}
7	{面包，黄油，尿布}
8	{啤酒，尿布}
9	{牛奶，尿布，面包，黄油}
10	{啤酒，饼干}

(a) 从这些数据中，能够提取出的关联规则的最大数量是多少（包括零支持度的规则）？

(b) 能够提取的频繁项集的最大长度是多少（假定最小支持度>0）？

(c) 写出从该数据集中能够提取的 3-项集的最大数量的表达式。

(d) 找出一个具有最大支持度的项集（长度为 2 或更大）。

(e) 找出一对项 a 和 b，使得规则$\{a\}\rightarrow\{b\}$和$\{b\}\rightarrow\{a\}$具有相同的置信度。

7. 考虑下面的频繁 3-项集的集合：

$$\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{3, 4, 5\}$$

假定数据集中只有 5 个项。

(a) 列出采用 $F_{k-1}\times F_1$ 合并策略，由候选产生过程得到的所有候选 4-项集。

(b) 列出由 *Apriori* 算法的候选产生过程得到的所有候选 4-项集。

(c) 列出 *Apriori* 算法候选剪枝步骤后剩下的所有候选 4-项集。

8. *Apriori*算法使用产生-计数的策略找出频繁项集。通过合并一对大小为k的频繁项集得到一个大小为k+1的候选项集（称作候选产生步骤）。在候选项集剪枝步骤中，如果一个候选项集的任何一个子集是不频繁的，则该候选项集将被丢弃。假定将*Apriori*算法用于表6-24所示数据集，最小支持度为30%，即任何一个项集在少于3个事务中出现就被认为是非频繁的。

表 6-24　购物篮事务的例子

事务 ID	购买项
1	{*a*, *b*, *d*, *e*}
2	{*b*, *c*, *d*}
3	{*a*, *b*, *d*, *e*}
4	{*a*, *c*, *d*, *e*}
5	{*b*, *c*, *d*, *e*}
6	{*b*, *d*, *e*}
7	{*c*, *d*}
8	{*a*, *b*, *c*}
9	{*a*, *d*, *e*}
10	{*b*, *d*}

(a) 画出表示表 6-24 所示数据集的项集格。用下面的字母标记格中每个结点。

- **N**：如果该项集被 *Apriori* 算法认为不是候选项集。一个项集不是候选项集有两种可能的原因：它没有在候选项集产生步骤产生，或它在候选项集产生步骤产生，但是由于它的一个子集是非频繁的而在候选项集剪枝步骤被丢掉。
- **F**：如果该候选项集被 *Apriori* 算法认为是频繁的。
- **I**：如果经过支持度计数后，该候选项集被发现是非频繁的。

(b) 频繁项集的百分比是多少？（考虑格中所有的项集）

(c) 对于该数据集，*Apriori* 算法的剪枝率是多少？（剪枝率定义为由于如下原因不认为是候选的项集所占的百分比：在候选项集产生时未被产生，或在候选剪枝步骤被丢掉。）

(d) 假警告率是多少？（假警告率是指经过支持度计算后被发现是非频繁的候选项集所占的百分比。）

9. *Apriori* 算法使用Hash树数据结构，有效地计算候选项集的支持度。考虑图6-32所示的候选3-项集的Hash树。

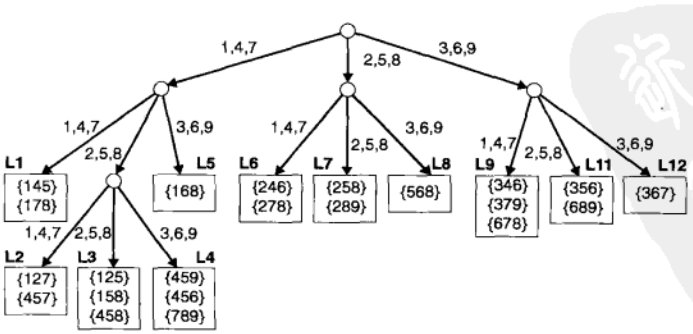


图 6-32　Hash 树结构的例子

(a) 给定一个包含项{1,3,4,5,8}的事务，在寻找该事务的候选项集时，访问了 Hash 树的哪些叶结点？

(b) 使用(a)中访问的叶结点确定事务{1,3,4,5,8}包含的候选项集。

10. 考虑下面的候选 3-项集的集合：{1,2,3}, {1,2,6}, {1,3,4}, {2,3,4}, {2,4,5}, {3,4,6}, {4,5,6}

(a) 构造以上候选 3-项集的 Hash 树。假定 Hash 树使用这样一个 Hash 函数：所有的奇数项都被散列到结点的左子女，所有偶数项被散列到右子女。一个候选 k-项集按如下方法插入到 Hash 树中：散列候选项集中的每个相继项，然后再按照散列值到相应的分支。一旦到达叶结点，候选项集将按照下面的条件插入。

- **条件 1**：如果该叶结点的深度等于 k（假定根结点的深度为 0），则不管该结点已经存储了多少个项集，将该候选插入该结点。
- **条件 2**：如果该叶结点的深度小于 k，则只要该结点存储的项集数不超过 *maxsize*，就把它插入到该叶结点。这里，假定 *maxsize* 为 2。
- **条件 3**：如果该叶结点的深度小于 k 且该结点已存储的项集数量等于 *maxsize*，则这个叶结点转变为内部结点，并创建新的叶结点作为老的叶结点的子女。先前老叶结点中存放的候选项集按照散列值分布到其子女中。新的候选项集也按照散列值存储到相应的叶结点。

(b) 候选 Hash 树中共有多少个叶结点、多少个内部结点？

(c) 考虑一个包含项集{1,2,3,5,6}的事务。使用(a)所创建的 Hash 树，则该事务要检查哪些叶结点？该事务包含哪些候选 3-项集？

11. 给定图 6-33 所示的格结构和表 6-24 给定的事务，用如下字母标记每一个结点。

- *M*：如果结点是极大频繁项集。
- *C*：如果结点是闭频繁项集。
- *N*：如果结点是频繁的，但既不是极大的也不是闭的。
- *I*：如果结点是非频繁的。

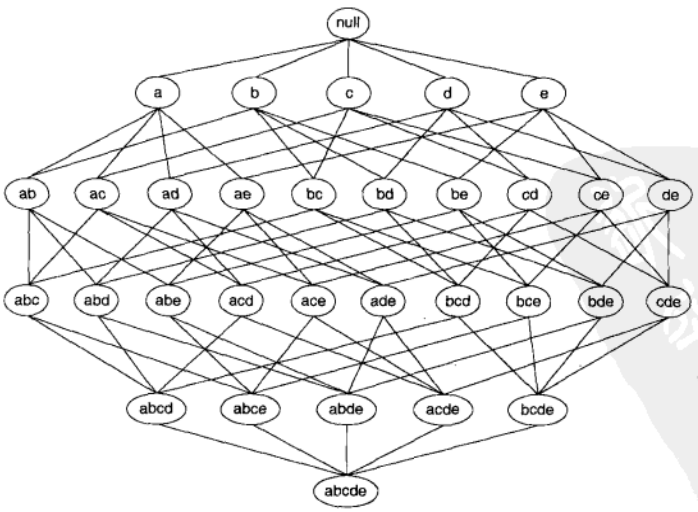


图 6-33　项集的格

假定支持度阈值等于 30%。

12. 传统的关联规则挖掘方法使用支持度和置信度度量来剪裁没有兴趣的规则。

(a) 使用表 6-25 中的事务数据，绘制出下面每个规则对应的相依表。

规则：$\{b\}\rightarrow\{c\}$, $\{a\}\rightarrow\{d\}$, $\{b\}\rightarrow\{d\}$, $\{e\}\rightarrow\{c\}$, $\{c\}\rightarrow\{a\}$

表 6-25　购物篮事务示例

事务 ID	购买项
1	$\{a, b, d, e\}$
2	$\{b, c, d\}$
3	$\{a, b, d, e\}$
4	$\{a, c, d, e\}$
5	$\{b, c, d, e\}$
6	$\{b, d, e\}$
7	$\{c, d\}$
8	$\{a, b, c\}$
9	$\{a, d, e\}$
10	$\{b, d\}$

(b) 利用(a)的相依表，按照下面的度量计算并依递减序确定规则的秩。

i. 支持度。

ii. 置信度。

iii. Interest$(X\rightarrow Y)=\dfrac{P(X,Y)}{P(X)}P(Y)$。

iv. IS$(X\rightarrow Y)=\dfrac{P(X,Y)}{\sqrt{P(X)P(Y)}}$。

v. Klosgen$(X\rightarrow Y)=\sqrt{P(X,Y)}\times(P(Y\mid X)-P(Y))$，其中 $P(Y\mid X)=\dfrac{P(X,Y)}{P(X)}$。

vi. 几率$(X\rightarrow Y)=\dfrac{P(X,Y)P(\bar{X},\bar{Y})}{P(X,\bar{Y})P(\bar{X},Y)}$。

13. 给定习题 12 中得到的秩，计算置信度的秩与其他五种度量之间的相关性。哪种度量与置信度相关性最强？哪种最弱？

14. 使用图 6-34 中所显示的数据集回答下列问题。注意，每个数据集包括 1000 个项和 10000 个事务。图中黑色单元表示项在事务中出现，白色表示不出现。假定使用 *Apriori* 算法提取频繁项集，最小支持度为 10%（即项集至少要包含在 1000 个事务中）。

(a) 哪些数据集产生的频繁项集数量最多？

(b) 哪些数据集产生的频繁项集数量最少？

(c) 哪些数据集产生最长的频繁项集？

(d) 哪些数据集产生具有最大支持度的频繁项集？

(e) 哪些数据集产生的频繁项集包含更广泛支持度（即所包含项的支持度由小于 20%到大于 70%）的项？

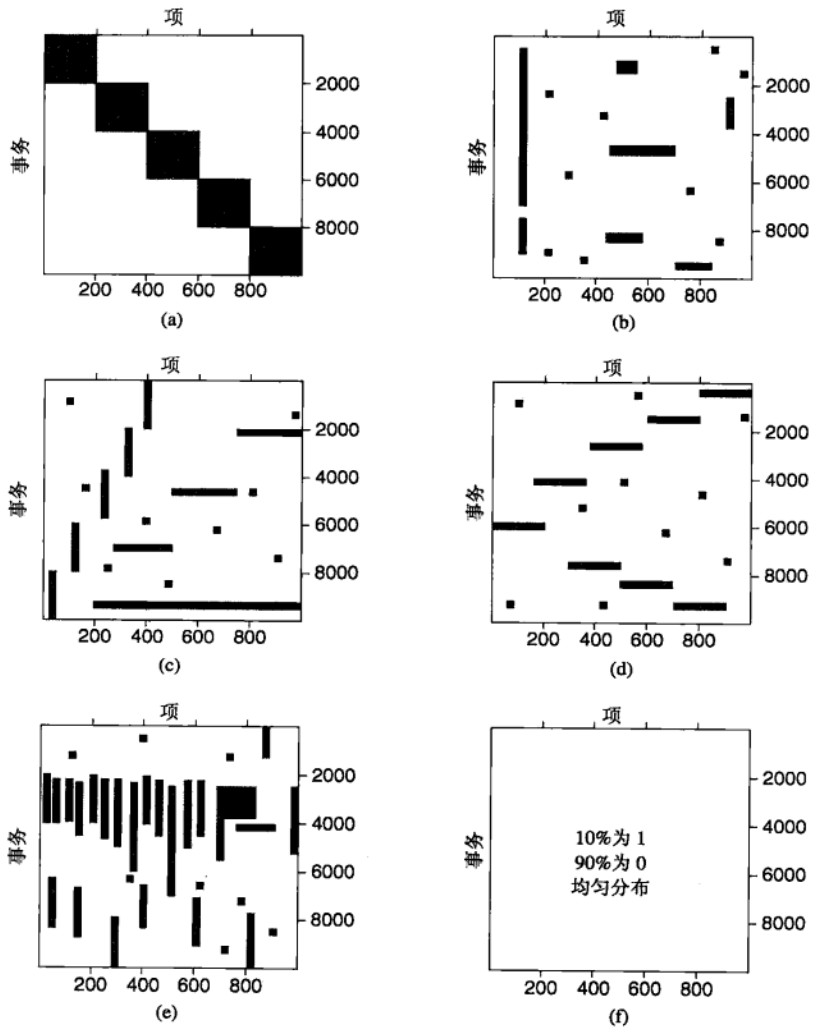


图 6-34　习题 14 的图

15. (a) 证明：当且仅当 $f_{11}=f_{1+}=f_{+1}$ 时，ϕ系数等于 1。

(b) 证明：如果 A 和 B 是相互独立的，则 $P(A,B)\times P(\bar{A},\bar{B})=P(A,\bar{B})\times P(\bar{A},B)$。

(c) 说明：Yule 的 Q 和 Y 系数是几率的规范化版本。

$$Q=\left[\frac{f_{11}f_{00}-f_{10}f_{01}}{f_{11}f_{00}+f_{10}f_{01}}\right]$$

$$Y=\left[\frac{\sqrt{f_{11}f_{00}}-\sqrt{f_{10}f_{01}}}{\sqrt{f_{11}f_{00}}+\sqrt{f_{10}f_{01}}}\right]$$

(d) 假定变量是统计独立的，写出表 6-11 和表 6-12 中所列出的各种度量值的简化表达式。

16. 对于关联规则 $A \rightarrow B$，考虑兴趣度度量 $M = \dfrac{P(B \mid A) - P(B)}{1 - P(B)}$。

    (a) 该度量的取值范围是什么？什么时候取最大值和最小值？

    (b) 当 $P(A, B)$增加，$P(A)$和 $P(B)$保持不变时，M 如何变化？

    (c) 当 $P(A)$增加，$P(A, B)$和 $P(B)$保持不变时，M 如何变化？

    (d) 当 $P(B)$增加，$P(A, B)$和 $P(A)$保持不变时，M 如何变化？

    (e) 该度量在变量置换下对称吗？

    (f) 若 A 和 B 是统计独立的，该度量的值是多少？

    (g) 该度量是零加不变的吗？

    (h) 在行或列缩放操作下，该度量保持不变吗？

    (i) 在反演操作下，该度量如何变化？

17. 假定有一个购物篮数据集，包含 100 个事务和 20 个项。假设项 a 的支持度为 25%，项 b 的支持度为 90%，且项集$\{a, b\}$的支持度为 20%。令最小支持度阈值和最小置信度阈值分别为 10%和 60%。

    (a) 计算关联规则$\{a\} \rightarrow \{b\}$的置信度。根据置信度度量，这条规则是有趣的吗？

    (b) 计算关联模式$\{a, b\}$的兴趣度度量。根据兴趣度度量，描述项 a 和项 b 之间联系的特点。

    (c) 由(a)和(b)的结果，能得出什么结论？

    (d) 证明：如果规则$\{a\} \rightarrow \{b\}$的置信度小于$\{b\}$的支持度，则

    i. $c(\{\bar{a}\} \rightarrow \{b\}) > c(\{a\} \rightarrow \{b\})$。

    ii. $c(\{\bar{a}\} \rightarrow \{b\}) > s(\{b\})$。

    其中，$c(\cdot)$表示规则置信度，$s(\cdot)$表示项集的支持度。

18. 表 6-26 显示了二元变量 A 和 B 在控制变量 C 的不同值上的 2×2×2 的相依表。

表 6-26　一个相依表

			A	
			1	0
$C=0$	B	1	0	15
		0	15	30
$C=1$	B	1	5	0
		0	0	15

    (a) 分别计算当 $C = 0$，$C = 1$ 和 $C = 0$ 或 1 时 A 和 B 的ϕ系数。注意：$\phi(\{A, B\}) = \dfrac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$。

    (b) 由上面的结果可以得出什么结论？

19. 考虑表 6-27 中显示的相依表。

表 6-27　习题 19 的相依表

	B	$\overline{B}$
A	9	1
$\overline{A}$	1	89

(a) 表 I

	B	$\overline{B}$
A	89	1
$\overline{A}$	1	9

(b) 表 II

(a) 对于表I，计算关联模式$\{A, B\}$的支持度、兴趣度和ϕ相关系数，并计算规则$A \rightarrow B$和$B \rightarrow A$的置信度。

(b) 对于表II，计算关联模式$\{A, B\}$的支持度、兴趣度和ϕ相关系数，并计算规则$A \rightarrow B$和$B \rightarrow A$的置信度。

(c) 由(a)和(b)的结果可以得出什么结论？

20. 考虑表 6-19 和表 6-20 中显示的购买高清晰度电视和购买健身器的顾客之间的联系。

(a) 计算两个表的几率。

(b) 计算两个表的ϕ系数。

(c) 计算两个表的兴趣因子。

对于上述每一个度量，描述当汇总数据取代分层数据后，关联方向的变化。

第7章

关联分析：高级概念

上一章介绍的关联规则挖掘假定输入数据由称作项的二元属性组成。还假定项在事务中出现比不出现更重要。这样，项被看作非对称的二元属性，并且只有频繁模式才被认为是有趣的。

本章将这种表示扩展到具有对称二元属性、分类属性和连续属性的数据集。这种表示还将进一步扩充到包含诸如序列和图形的更复杂的实体。尽管关联分析算法的总体结构保持不变，但是算法的某些方面必须加以修改，以便处理非传统的实体。

7.1 处理分类属性

许多应用包含对称二元属性和标称属性。表 7-1 显示的因特网调查数据包含对称二元属性，如性别、家庭计算机、网上聊天、网上购物和关注隐私；还包括标称属性，如文化程度和州。使用关联分析，我们可能发现关于因特网用户特征的有趣信息，如

$$\{网上购物 = 是\} \rightarrow \{关注隐私 = 是\}$$

这条规则暗示大部分在网上购物的因特网用户都关心个人隐私。

表 7-1 具有分类属性的因特网调查数据

性别	文化程度	州	家庭计算机	网上聊天	网上购物	关注隐私
女	研究生	伊利诺伊	是	是	是	是
男	大学	加利福尼亚	否	否	否	否
男	研究生	密歇根	是	是	是	是
女	大学	弗吉尼亚	否	否	是	是
女	研究生	加利福尼亚	是	否	否	是
男	大学	明尼苏达	是	是	是	是
男	大学	阿拉斯加	是	是	是	否
男	高中	俄勒冈	是	否	否	否
女	研究生	得克萨斯	否	是	否	否
…	…	…	…	…	…	…

为了提取这样的模式，首先将分类属性和对称二元属性转换成“项”，目的是使用已有的关联规则挖掘算法。这种类型的变换可以通过为每个不同的属性-值对创建一个新的项来实现。例如，标称属性文化程度可以用三个二元项取代：文化程度=大学，文化程度=研究生，文化程度=高中。类似地，对称二元属性性别可以转换成一对二元项：男和女。表 7-2 显示因特网调查数据二元化后的结果。

表 7-2　二元化分类属性和对称二元属性后的因特网调查数据

男	女	文化程度 = 研究生	文化程度 = 大学	…	关注隐私 = 是	关注隐私 = 否
0	1	1	0	…	1	0
1	0	0	1	…	0	1
1	0	1	0	…	1	0
0	1	0	1	…	1	0
0	1	1	0	…	1	0
1	0	0	1	…	1	0
1	0	0	1	…	0	1
1	0	0	0	…	0	1
0	1	1	0	…	0	1
…	…	…	…	…	…	…

将关联分析用于二元化后的数据时，需要考虑如下问题。

(1) 有些属性值可能不够频繁，不能成为频繁模式的一部分。对于具有许多可能属性值的标称属性（如州名），这个问题更为明显。降低支持度阈值不起作用，因为发现的频繁模式（许多可能是不真实的）将以指数增长，计算开销更高。更实际的做法是，将相关的属性值分组，形成少数类别。例如，每个州名都可以用对应的地理区域如中西部、太平洋西北部、西南部和东海岸取代。另一种可能性是，将不太频繁的属性值聚合成一个称作其他的类别，如图 7-1 所示。

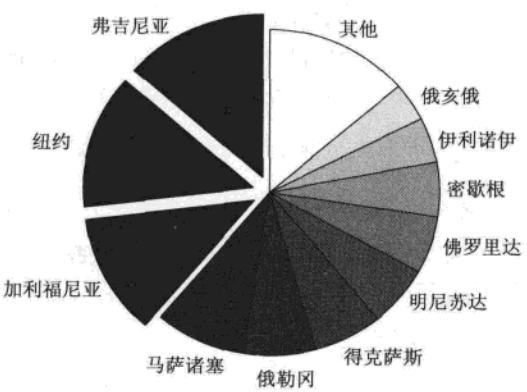


图 7-1　具有合并的“其他”类别的饼图

(2) 某些属性值的频率可能比其他属性高很多。例如，假定 85%的被调查人都有家庭计算机。如果为每个频繁出现在数据中的属性值创建一个二元项，我们可能产生许多冗余模式，如下面的例子所示：

$$\{\text{家庭计算机=是，网上购物=是}\} \rightarrow \{\text{关注隐私=是}\}$$

该规则是冗余的，因为它被归入本节开始给出的更一般的规则。由于高频繁度的项对应于属性的典型值，因此它们很少携带有助于更好地理解模式的新信息。因此，在使用标准的关联分析算法之前，删除这样的项可能是有好处的。另一种可能的做法是，使用 6.8 节提供的技术，处理具有宽支持度值域的数据集。

(3) 尽管每个事务的宽度与原始数据中属性的个数相同，但是计算时间可能增加，特别是当新创建的项变成频繁项时。这是因为需要更多时间处理由这些项产生的候选项集（见习题 1）。

减少计算时间的一种方法是，避免产生包含多个来自同一属性的项的候选项集。例如，我们不必产生诸如{州 $=X$, 州 $=Y$, …}的候选项集，因为该项集的支持度计数为零。

7.2 处理连续属性

上一节介绍的因特网调查数据可能还包含连续属性，如表 7-3 所示。挖掘连续属性可能揭示数据的内在联系，如“年收入超过\$120K 的用户属于 45～60 年龄组”，或“拥有超过 3 个 e-mail 账号并且每周上网超过 15 小时的用户通常关注个人隐私”。包含连续属性的关联规则通常称作**量化关联规则**（quantitative association rule）。

表 7-3 具有连续属性的因特网调查数据

性别	…	年龄	年收入	每周上网小时数	e-mail 账号数	关注隐私
女	…	26	90K	20	4	是
男	…	51	135K	10	2	否
男	…	29	80K	10	3	是
女	…	45	120K	15	3	是
女	…	31	95K	20	5	是
男	…	25	55K	25	5	是
男	…	37	100K	10	1	否
男	…	41	65K	8	2	否
女	…	26	85K	12	1	否
…	…	…	…	…	…	…

本节介绍对连续数据进行关联分析的各种方法。具体地说，我们讨论三类方法：(1)基于离散化的方法，(2)基于统计学的方法，(3)非离散化方法。使用这些方法导出的量化关联规则本质上差别很大。

7.2.1 基于离散化的方法

离散化是处理连续属性最常用的方法。这种方法将连续属性的邻近值分组，形成有限个区间。例如，年龄属性可以划分成如下区间：

$$\text{年龄}\in[12, 16),\ \text{年龄}\in[16, 20),\ \text{年龄}\in[20, 24),\ \cdots,\ \text{年龄}\in[56, 60)$$

其中，$[a,b)$ 代表包含 a 但不包含 b 的区间。离散化可以使用 2.3.6 节介绍的任意技术（等区间宽度、等频率、基于熵或聚类）实现。离散的区间可以映射到非对称的二元属性，使得可以使用已有的关联分析算法。表 7-4 显示离散化和二元化后的因特网调查数据。

属性离散化的一个关键参数是用于划分每个属性的区间个数。通常，这个参数由用户提供，用区间宽度（对于等区间宽度方法）、每个区间的平均事务个数（对于等频率方法）或所希望的聚类数（对于基于聚类的方法）来表示。确定正确的区间数的困难性可以用表 7-5 中的数据解释。该表汇总参加调查的 250 个用户的回答。数据中隐含两个强规则。

R_1：年龄$\in$ [16, 24) →网上聊天=是 （$s = 8.8\%$, $c = 81.5\%$）

R_2：年龄$\in$ [44, 60) →网上聊天=否 （$s = 16.8\%$, $c = 70\%$）

表 7-4　二元化分类属性和连续属性后的因特网调查数据

男	女	…	年龄<13	年龄∈[13, 21)	年龄∈[21, 30)	…	关注隐私=是	关注隐私=否
0	1	…	0	0	1	…	1	0
1	0	…	0	0	0	…	0	1
1	0	…	0	0	1	…	1	0
0	1	…	0	0	0	…	1	0
0	1	…	0	0	0	…	1	0
1	0	…	0	0	1	…	1	0
1	0	…	0	0	0	…	0	1
1	0	…	0	0	0	…	0	1
0	1	…	0	0	1	…	0	1
…	…	…	…	…	…	…	…	…

这些规则暗示 16～24 岁年龄组的大部分用户通常参加网上聊天，而 44～60 岁的多半不会参加网上聊天。在这个例子中，我们认为某个规则是有趣的，仅当它的支持度（s）超过 5%，并且它的置信度（c）超过 65%。当我们对年龄属性离散化时，遇到的问题之一是如何确定区间宽度。

表 7-5　根据参加网上聊天的因特网用户的年龄组划分因特网用户

年龄组	网上聊天=是	网上聊天=否
[12, 16)	12	13
[16, 20)	11	2
[20, 24)	11	3
[24, 28)	12	13
[28, 32)	14	12
[32, 36)	15	12
[36, 40)	16	14
[40, 44)	16	14
[44, 48)	4	10
[48, 52)	5	11
[52, 56)	5	10
[56, 60)	4	11

(1) 如果区间太宽，则可能因为缺乏置信度而丢失某些模式。例如，当区间宽度为 24 岁时，R_1 和 R_2 被如下规则所取代。

$$R_1'：\text{年龄}\in[12,\ 36)\rightarrow\text{网上聊天=是}\quad(s=30\%,\ c=57.7\%)$$
$$R_2'：\text{年龄}\in[36,\ 60)\rightarrow\text{网上聊天=否}\quad(s=28\%,\ c=58.3\%)$$

尽管它们有较高的支持度，但是较宽的区间导致两个规则的置信度都低于最小置信度阈值。其结果是，离散化之后，两个模式都失去了。

(2) 如果区间太窄，则可能因为缺乏支持度而丢失某些模式。例如，如果区间宽度为 4 岁，则 R_1 被分裂成如下两个子规则。

$$R_{11}^{(4)}：\text{年龄}\in[16,\ 20)\rightarrow\text{网上聊天=是}\quad(s=4.4\%,\ c=84.6\%)$$
$$R_{12}^{(4)}：\text{年龄}\in[20,\ 24)\rightarrow\text{网上聊天=是}\quad(s=4.4\%,\ c=78.6\%)$$

由于两个子规则的支持度都低于最小支持度阈值，离散化后 R_1 丢失了。同理，规则 R_2 被分

裂成 4 个子规则，也因 4 个子规则的支持度都低于最小支持度阈值而丢失。

(3) 如果区间宽度是 8 岁，则规则 R_2 被分裂成如下两个子规则。

$R_{21}^{(8)}$：年龄∈ [44, 52) →网上聊天=否 （s =8.4%, c = 70%）

$R_{22}^{(8)}$：年龄∈ [52, 60) →网上聊天=否 （s =8.4%, c = 70%）

由于 $R_{21}^{(8)}$ 和 $R_{22}^{(8)}$ 都有足够的支持度和置信度，R_2 可以通过聚合两个子规则而恢复。与此同时，R_1 被分裂成如下两个子规则。

$R_{11}^{(8)}$：年龄∈ [12, 20) →网上聊天=是 （s = 9.2%, c = 60.5%）

$R_{12}^{(8)}$：年龄∈ [20, 28) →网上聊天=是 （s = 9.2%, c = 60.0%）

不像 R_2，我们不能通过聚合这两个子规则来恢复 R_1，因为两个子规则的置信度都低于阈值。

处理这些问题的一个方法是，考虑邻近区间的每种可能的分组。例如，我们可以以宽度 4 岁开始，将近邻的区间合并成较宽的区间，年龄∈ [12, 16)，年龄∈ [12, 20)，…，年龄∈ [12, 60)，年龄∈ [16, 20)，年龄∈ [16, 24) 等等。这种方法能够检测出 R_1 和 R_2 是强规则。然而，这也导致如下计算问题。

(1) **计算开销非常大**。如果值域被划分成 k 个区间，则必须创建 $k(k-1)/2$ 个二元项来代表所有可能的区间。此外，如果对应于区间 $[a,b)$ 的项是频繁的，则包含 $[a,b)$ 的区间对应的所有项也必然是频繁的。因此，这种方法可能产生过多的候选和频繁项集。为了处理这些问题，可以使用最大支持度阈值，防止创建对应于非常宽的区间的项，并减少项集的数量。

(2) **提取许多冗余规则**。例如，考虑下面的规则对：

R_3：{年龄∈ [16, 20)，性别=男}→{网上聊天=是}

R_4：{年龄∈ [16, 24)，性别=男}→{网上聊天=是}

R_4 是 R_3 的泛化（R_3 是 R_4 的特化），因为对于年龄属性，R_4 有更宽的区间。如果两个规则的置信度值相同，则 R_4 应当更有趣，因为它涵盖了更多的例子——包括 R_3 涵盖的那些。因此，R_3 是冗余的。

7.2.2 基于统计学的方法

量化关联规则可以用来推断总体的统计性质。例如，假定我们希望根据表 7-1 和表 7-3 提供的数据，找出因特网用户特定组群的平均年龄。使用本节介绍的基于统计学的方法，可以提取如下形式的量化关联规则：

{年收入 > $100K, 网上购物=是 }→年龄：均值=38

该规则表明年收入超过$100K 并且定期在网上购物的因特网用户的平均年龄为 38 岁。

1. 规则产生

为了产生基于统计学的量化关联规则，必须指定用于刻画有趣总体段特性的目标属性。保留目标属性，使用上一节介绍的方法对数据中的其余分类属性和连续属性二元化。然后，可以使用已有的算法，如 *Apriori* 算法或 FP 增长，从二元化数据中提取频繁项集。每个频繁项集确定一个有趣总体段。使用诸如均值、中位数、方差或绝对偏差等统计量，可以对目标属性在每个段内的

分布进行汇总。例如，前面的规则就是通过对支持频繁项集{年收入 > $100K, 网上购物 = 是}的因特网用户的年龄求平均值得到的。

使用这个方法得到的量化关联规则的数量与提取的频繁项集相同。由于量化关联规则的定义方法，对于这种规则，不能使用置信度。确认关联规则的可选方法在下面给出。

2. 规则确认

某个量化关联规则是有趣的，仅当由规则覆盖的事务计算的统计量不同于由未被规则覆盖的事务计算的统计量。例如，本节开始给出的规则是有趣的，仅当不支持频繁集{年收入 >$100K, 网上购物 = 是}的因特网用户的平均年龄显著地大于或小于 38 岁。为了确定该平均年龄差是否具有统计意义，应当使用统计假设检验方法进行检验。

考虑量化关联规则 $A \rightarrow t:\mu$，其中 A 是频繁项集，t 是连续的目标属性，而 μ 是被 A 覆盖的事务的 t 的平均值。此外，设 μ' 是未被 A 覆盖的事务的 t 的平均值。目标是检验 μ 和 μ' 之间的差是否大于用户指定的某个阈值 Δ。在统计假设检验中，给定两个相反的假设分别称作**原假设**（null hypothesis）和**备择假设**（alternative hypothesis）。根据从数据收集的证据，进行假设检验，确定两个假设中的哪一个被接受。

在这种情况下，假定 $\mu < \mu'$，则原假设是 $H_0: \mu' = \mu + \Delta$，而备择假设是 $H_1: \mu' > \mu + \Delta$。为了确定应当接受哪个假设，计算下面的 Z 统计量：

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{7-1}$$

其中，n_1 是支持 A 的事务个数，n_2 是不支持 A 的事务个数，s_1 是支持 A 的事务的 t 的标准差，而 s_2 是不支持 A 的事务的 t 的标准差。在原假设下，Z 具有标准正态分布，均值为 0，方差为 1。然后，将使用公式（7-1）计算的 Z 值与临界值 Z_α 比较，其中 Z_α 是依赖于期望置信水平的阈值。如果 $Z > Z_\alpha$，则原假设被拒绝，并且我们可以断言该量化关联规则是有趣的。否则，数据中没有足够的证据证明均值之差具有统计意义。

例 7.1　考虑量化关联规则

$$\{收入 > 100K,\ 网上购物 = 是\} \rightarrow 年龄 : \mu = 38$$

假定有 50 个因特网用户支持该规则的前件。他们年龄的标准差是 3.5。另一方面，不支持该规则前件的 200 个用户的平均年龄是 30，标准差是 6.5。假定量化关联规则是有趣的，仅当 μ 与 μ' 之间的差大于 5 岁。使用公式（7-1），我们得到

$$Z = \frac{38 - 30 - 5}{\sqrt{\frac{3.5^2}{50} + \frac{6.5^2}{200}}} = 4.4414$$

对于一个置信水平为 95%的单侧假设检验，拒绝原假设的临界值是 1.64。由于 $Z > 1.64$，原假设被拒绝。因此，我们断言该量化关联规则是有趣的，因为支持和不支持规则前件的用户的平均年龄之差大于 5 岁。 □

7.2.3 非离散化方法

有一些应用，令分析者更感兴趣的是发现连续属性之间的关联，而不是连续属性的离散区间之间的关联。例如，考虑如下问题：找出表 7-6 所示文本文档中词的关联。文档-词矩阵中的每个表值代表词在给定文档中出现的规范化频率。用每个词的频率除以所有文档词频之和对数据进行规范化。这种规范化的理由之一是确保所得到的支持度值是 0 和 1 之间的数。然而，更重要的理由是确保数据在相同的尺度上，以相同方式变化的词的集合可以具有相似的支持度值。

表 7-6 规范化的文档-词矩阵

文档	$word_1$	$word_2$	$word_3$	$word_4$	$word_5$	$word_6$
d_1	0.3	0.6	0	0	0	0.2
d_2	0.1	0.2	0	0	0	0.2
d_3	0.4	0.2	0.7	0	0	0.2
d_4	0.2	0	0.3	0	0	0.1
d_5	0	0	0	1.0	1.0	0.3

在文本挖掘中，分析者更感兴趣的是发现词（例如，*数据*和*挖掘*）之间的关联，而不是词频区间（例如，*数据*∈[1, 4]，*挖掘*∈[2, 3]）之间的关联。一种做法是，将数据变换成 0/1 矩阵；其中，如果规范化词频超过某个阈值 t，则值为 1，否则为 0。尽管该方法使得分析者可以对二元化数据集使用已有的频繁模式产生算法，但是为二元化找到合适的阈值却很棘手。如果阈值设得太大，则可能丢失有趣的关联。反之，如果阈值设得太小，则可能产生大量谬误的关联。

本节提供另一种发现词关联的方法，称作 min-*Apriori*。类似于传统的关联分析，项集是词的汇集，而其支持度用来度量词之间的关联程度。项集的支持度可以根据对应词的规范化频率计算。例如，考虑表 7-6 中的文档 d_1。词 $word_1$ 和 $word_2$ 的规范化频率分别为 0.3 和 0.6。有人可能认为，计算这两个词之间关联的一个合理的方法是取它们的规范化频率的平均值，即(0.3+0.6)/2 = 0.45。然后，对所有文档的平均规范化频率求和，就可以计算项集的支持度：

$$s(\{word_1, word_2\}) = \frac{0.3+0.6}{2}+\frac{0.1+0.2}{2}+\frac{0.4+0.2}{2}+\frac{0.2+0}{2}=1$$

这个结果一点也不意外。因为每个词频都规范化到 1，对规范化频率取平均值使得每个项集的支持度等于 1。这样，使用该方法，所有的项集都是频繁的。该方法对于识别有趣的模式毫无用处。

在 min-*Apriori* 中，给定文档中词之间的关联通过取它们的规范化频率的最小值得到，即 $\min(word_1, word_2) = \min(0.3, 0.6) = 0.3$。项集的支持度通过在所有文档上聚集它的支持度得到。

$$s(\{word_1, word_2\}) = \min(0.3, 0.6)+ \min(0.1, 0.2)+ \min(0.4, 0.2)+ \min(0.2, 0)=0.6$$

min-*Apriori* 中定义的支持度具有如下期望性质，使它适合用来发现文档中词的关联。

(1) 支持度随词的规范化频率增加而单调递增。

(2) 支持度随包含该词的文档个数增加而单调递增。

(3) 支持度具有反单调性。例如，考虑一对项集$\{A, B\}$和$\{A, B, C\}$。由于 $\min(\{A, B\}) \geqslant \min(\{A, B, C\})$，从而 $s(\{A, B\}) \geqslant s(\{A, B, C\})$。因此，支持度随项集中词数的增加而单调递减。

使用新的支持度定义，可以修改标准 *Apriori* 算法，来发现词之间的关联。

7.3　处理概念分层

概念分层是定义在一个特定的域中的各种实体或概念的多层组织。例如，在购物篮分析中，概念分层具有如下形式：项的分类法描述商店销售的商品之间的“is-a”联系。例如，牛奶是一种食品，而 DVD 是一种电子设备（见图 7-2）。通常，概念分层根据领域知识，或者基于特定组织的标准分类方案来定义（例如，国会图书馆的分类方案用来根据主题组织图书资料）。

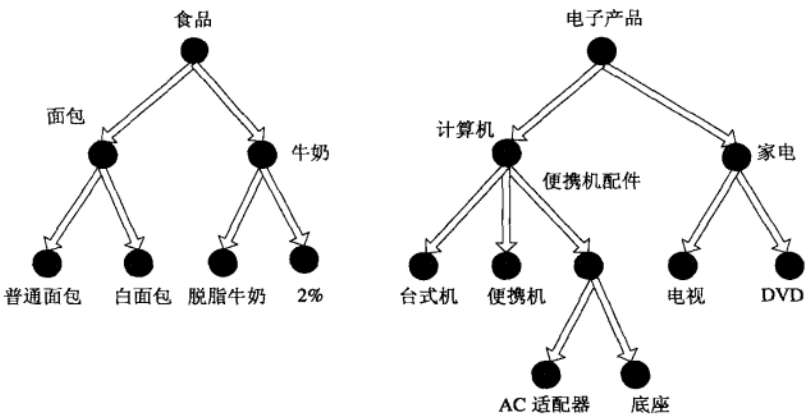


图 7-2　商品分类的例子

概念分层可以用**有向无环图**（directed acyclic graph）表示，如图 7-2 所示。如果图 7-2 中存在一条从结点 p 到另一个结点 c 的边，则称 p 是 c 的**父母**，c 是 p 的**子女**。例如，*牛奶*是*脱脂牛奶*的父母，因为从结点*牛奶*到结点*脱脂牛奶*存在一条有向边。$\hat{X}$ 称作 X 的**祖先**（X 是 $\hat{X}$ 的**后代**），如果有向图中存在一条从 $\hat{X}$ 到 X 的路径。在图 7-2 中*食品*是*脱脂牛奶*的祖先，而 AC *适配器*是*电子产品*的后代。

将概念分层纳入关联分析的主要优点如下。

(1) 位于层次结构较下层的项可能没有足够的支持度，从而不在任何频繁项集中出现。例如，尽管 AC 适配器和底座的销售量可能很低，但是，作为概念分层结构中它们的父母结点，便携机配件的销售量可能很高。不使用概念分层就可能丢失涉及便携机配件的有趣模式。

(2) 在概念分层的较低层发现的规则过于特殊，可能不如较高层的规则令人感兴趣。例如，诸如牛奶和面包等大宗商品趋向于产生许多低层规则，如，*脱脂牛奶→普通面包*，*2%牛奶→普通面包*，*脱脂牛奶→白面包*。使用概念分层结构，它们可以汇总为一条规则：*牛奶→面包*。仅考虑分层结构顶部的商品可能也不好，因为这样的规则可能没有任何实际应用价值。例如，尽管规则*电子产品→食品*可能满足支持度和置信度阈值，但是它并不提供什么信息，因为顾客经常一起购买电子产品和食品是已知的事实。如果牛奶和电池才是经常同时销售的商品，则模式{*食品*，*电子产品*}可能过分泛化了这种情况。

可以用以下方法扩充标准的关联分析，使其包括概念分层。初始，每个事务 t 用它的扩**展事务**（extended transaction）t'取代，其中，t'包含 t 中所有项和它们的对应祖先。例如，事务{DVD，*普通面包*}可以扩展为{DVD，*普通面包*，*家电*，*电子产品*，*面包*，*食品*}，其中，*家电*和*电子产品*是 DVD 的祖先，而*面包*和*食品*是*普通面包*的祖先。使用这种方法，可以对扩展的数据库使用

诸如 *Apriori* 等已有的算法来发现跨越多个概念层的规则。这种方法有一些明显的局限性。

(1) 处于较高层的项比处于较低层的项趋向于具有较高的支持度计数。这样，如果支持度阈值设得太高，则只能提取涉及较高层项的模式。另一方面，如果阈值设得太低，则算法可能产生太多模式（其中大部分可能是不真实的），使得计算效率极低。

(2) 概念分层的引入增加了关联分析的计算时间，因为项的个数更多，事务宽度更大。算法产生的候选模式和频繁模式的个数可能随事务变宽而指数增加。

(3) 概念分层的引入可能产生冗余规则。规则 $X \rightarrow Y$ 是冗余的，如果存在一个更一般的规则 $\hat{X} \rightarrow \hat{Y}$，其中 $\hat{X}$ 是 X 的祖先，$\hat{Y}$ 是 Y 的祖先，并且两个规则具有非常相似的置信度。例如，假定{面包}→{牛奶}，{白面包}→{2%牛奶}，{白面包}→{脱脂牛奶}和{普通面包}→{脱脂牛奶}具有非常相似的置信度。涉及较低层中项的规则是冗余的，因为它们可以被涉及其祖先的规则所概括。诸如{脱脂牛奶，牛奶，食品}的项集也是冗余的，因为食品和牛奶都是脱脂牛奶的祖先。幸而，给定分层结构，在频繁模式产生时容易删除这类冗余项集。

7.4 序列模式

购物篮数据常常包含关于商品何时被顾客购买的时间信息。可以使用这种信息，将顾客在一段时间内的购物拼接成事务序列。同样，从管理科学实验或对诸如通信网络、计算机网络和无线遥感网络等的物理系统中收集的基于事件的数据都具有固有的序列特征。也就是说在这种数据代表的事件之间存在某种序关系，通常基于时间或空间的先后次序。然而，迄今为止所讨论的关联模式概念都只强调同时出现关系，而忽略数据中的序列信息。对于识别动态系统的重现特征，或预测特定事件的未来发生，序列信息可能是非常有价值的。本节给出序列模式的基本概念和发现序列模式的算法。

7.4.1 问题描述

发现序列模式的问题输入是一个序列数据集，如图 7-3 左部所示。每一行记录与一个特定的对象相关联的一些事件在给定时刻的出现。例如，第一行包含在时间戳 $t = 10$ 时出现的对象 A 的事件集。将与对象 A 有关的所有事件按时间戳增序排序，就得到对象 A 的一个**序列**（sequence），如图 7-3 右部所示。

序列数据库

对象	时间戳	事件
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 8, 7

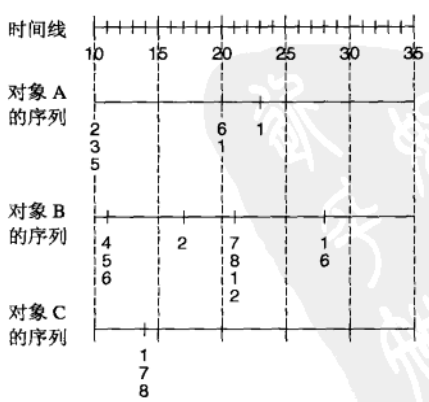


图 7-3 一个序列数据库的例子

一般地，序列是**元素**（element）的有序列表，可以记作 $s = \langle e_1 e_2 e_3 \cdots e_n \rangle$，其中每个 e_j 是一个或多个事件的集族，即 $e_j = \{i_1, i_2, \cdots, i_k\}$。下面是一些序列的例子。

- Web 站点访问者访问的 Web 页面序列：

  ⟨{主页}{电子产品}{照相机和摄像机}{数码像机}{购物车}{订购确认}{返回购物}⟩

- 导致三里岛核事故的事件序列：

  ⟨{树脂堵塞}{出口阀关闭}{失去给水}{冷凝器出口阀关闭}{增压泵跳闸}{主水泵跳闸}{主涡轮机跳闸}{反应堆压力上升}⟩

- 计算机科学主修课程序列：

  ⟨{算法与数据结构，操作系统引论}{数据库系统，计算机体系结构}{计算机网络，软件工程}{计算机图形学，并行程序设计}⟩

序列可以用它的长度和出现事件的个数刻画。序列的长度对应于出现在序列中的元素个数，而 k-序列是包含 k 个事件的序列。上面例子中的 Web 序列包含 7 个元素和 7 个事件；三里岛事件序列包含 8 个元素和 8 个事件；而课程序列包含 4 个元素和 8 个事件。

图 7-4 提供了一些应用领域定义的序列、元素和事件的例子。除最后一行外，与前三个领域相关的序数属性对应于日历时间。对于最后一行，序数属性对应于基（A、C、G、T）在基因序列中的位置。尽管关于序列模式的讨论主要考虑时间事件，但是可以将它推广到事件具有空间次序的情况。

序列数据库	序　列	元素（事务）	事件（项）
顾客	给定顾客的购物历史	顾客在时刻 t 购买的商品的集合	书、日常用品、CD 等
Web 数据	特定 Web 访问者的浏览活动	一次鼠标点击后 Web 访问者观看的文件的集合	主页、索引页、联系信息等
事件数据	给定的传感器产生的事件历史	传感器在时刻 t 触发的事件	传感器产生的警报类型
基因组序列	一个特定物种的 DNA 序列	DNA 序列的元素	基 A、T、G、C

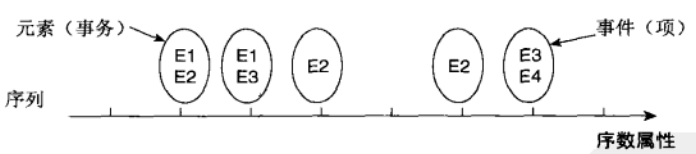


图 7-4　序列数据集中元素和事件的例子

子序列

序列 t 是另一个序列 s 的**子序列**（subsequence），如果 t 中每个有序元素都是 s 中一个有序元素的子集。形式化表述为，序列 $t = \langle t_1 t_2 \cdots t_m \rangle$ 是序列 $s = \langle s_1 s_2 \cdots s_n \rangle$ 的子序列，如果存在整数 $1 \leqslant j_1 < j_2 < \cdots < j_m \leqslant n$，使得 $t_1 \subseteq s_{j_1}$，$t_2 \subseteq s_{j_2}$，…，$t_m \subseteq s_{j_m}$。如果 t 是 s 的子序列，则称 t 包含在 s 中。下表的例子解释了子序列的概念。

序列 s	序列 t	t 是 s 的子序列吗？
⟨{2,4} {3,5,6} {8}⟩	⟨{2} {3,6} {8}⟩	是

（续）

序列 s	序列 t	t 是 s 的子序列吗？
⟨{2, 4} {3, 5, 6} {8}⟩	⟨{2} {8}⟩	是
⟨{1, 2} {3, 4}⟩	⟨{1} {2}⟩	否
⟨{2, 4} {2, 4} {2, 5}⟩	⟨{2} {4}⟩	是

7.4.2 序列模式发现

设 D 是包含一个或多个**数据序列**（data sequence）的数据集。术语数据序列是指与单个数据对象相关联的事件的有序列表。例如，图 7-3 中显示的数据集包含三个数据序列，对象 A、B 和 C 各一个。

序列 s 的支持度是包含 s 的所有数据序列所占的比例。如果序列 s 的支持度大于或等于用户指定的阈值 *minsup*，则称 s 是一个序列模式（或频繁序列）。

定义 7.1 序列模式发现 给定序列数据集 D 和用户指定的最小支持度阈值 *minsup*，序列模式发现的任务是找出支持度大于或等于 *minsup* 的所有序列。

图 7-5 给出了一个包含 5 个数据序列的数据集的例子。序列⟨{1}{2}⟩的支持度等于 80%，因为它出现在 5 个数据序列的 4 个中（除 D 之外的每个对象）。假定最小支持度阈值是 50%，则至少出现在 3 个数据序列中的任何序列都被视为序列模式。从给定的数据集中提取的序列模式的例子包括⟨{1}{2}⟩，⟨{1, 2}⟩，⟨{2, 3}⟩，⟨{1, 2}{2, 3}⟩等。

对象	时间戳	事件
A	1	1, 2, 4
A	2	2, 3
A	3	5
B	1	1, 2
B	2	2, 3, 4
C	1	1, 2
C	2	2, 3, 4
C	3	2, 4, 5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

Minsup = 50%

序列模式的例子：

<{1,2}>	s=60%
<{2,3}>	s=60%
<{2,4}>	s=80%
<{3} {5}>	s=80%
<{1} {2}>	s=80%
<{2} {2}>	s=60%
<{1} {2,3}>	s=60%
<{2} {2,3}>	s=60%
<{1,2} {2,3}>	s=60%

图 7-5 由包含 5 个数据序列的数据集导出的序列模式

序列模式的发现是一项具有挑战性的计算任务，因为在给定的数据序列中的序列有指数多个。例如，数据序列⟨{a, b}{c, d, e}{f}{g, h, i}⟩包含的序列有⟨{a}{c, d}{f}{g}⟩，⟨{c, d, e}⟩，⟨{b}{g}⟩等。容易证明，出现在具有 n 个事件的数据序列中的 k-序列总数为 C_n^k。因此，具有 9 个事件的数据序列包含

$$C_9^1 + C_9^2 + \cdots + C_9^9 = 2^9 - 1 = 511$$

个不同的序列。

产生序列模式的一种蛮力方法是枚举所有可能的序列，并统计它们各自的支持度。给定 n 个事件的集族，首先产生候选 1-序列，然后是候选 2-序列，候选 3-序列，等等。

1-序列：$\langle i_1\rangle, \langle i_2\rangle, \cdots, \langle i_n\rangle$

2-序列：$\langle\{i_1, i_2\}\rangle, \langle\{i_1, i_3\}\rangle, \cdots, \langle\{i_{n-1}, i_n\}\rangle, \langle\{i_1\}\{i_1\}\rangle, \langle\{i_1\}\{i_2\}\rangle, \cdots, \langle\{i_{n-1}\}\{i_n\}\rangle$

3-序列：$\langle\{i_1, i_2, i_3\}\rangle, \langle\{i_1, i_2, i_4\}\rangle, \cdots, \langle\{i_1, i_2\}\{i_1\}\rangle, \cdots,$
$\langle\{i_1\}\{i_1, i_2\}\rangle, \cdots, \langle\{i_1\}\{i_1\}\{i_1\}\rangle, \cdots, \langle\{i_n\}\{i_n\}\{i_n\}\rangle$

注意，候选序列的个数比候选项集的个数大得多。产生更多候选的原因有下面两个。

(1) 一个项在项集中最多出现一次，但一个事件可以在序列中出现多次。给定两个项 i_1 和 i_2，只能产生一个候选 2-项集$\{i_1, i_2\}$，但却可以产生许多候选 2-序列，如$\langle\{i_1, i_2\}\rangle$，$\langle\{i_1\}\{i_2\}\rangle$，$\langle\{i_2, i_1\}\rangle$和$\langle\{i_1, i_1\}\rangle$。

(2) 次序在序列中是重要的，但在项集中不重要。例如，{1, 2}和{2, 1}表示同一个项集，而$\langle\{i_1\}\{i_2\}\rangle$和$\langle\{i_2\}\{i_1\}\rangle$对应于不同的序列，因此必须分别产生。

先验原理对序列数据成立，因为包含特定 k-序列的任何数据序列必然包含该 k-序列的所有 (k–1)-子序列。可以开发类 *Apriori* 算法，从序列数据集中提取序列模式。算法的基本结构在算法 7.1 中给出。

算法 7.1　序列模式发现的类 *Apriori* 算法

1: $k = 1$
2: $F_k = \{i \mid i \in I \wedge \sigma(\{i\})/N \geqslant minsup\}$。　{找出所有的频繁 1-序列。}
3: **repeat**
4: $k = k + 1$
5: C_k = apriori-gen(F_{k-1})。　{产生候选 k-序列。}
6: **for** 每个数据序列 $t \in T$ **do**
7: C_t = subsequence(C_k, t)。　{识别包含在 t 中的所有候选。}
8: **for** 每个候选 k-序列 $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$。　{支持度计数增值。}
10: **end for**
11: **end for**
12: $F_k = \{c \mid c \in C_k \wedge \sigma(c)/N \geqslant minsup\}$。　{提取频繁 k-序列。}
13: **until** $F_k = \varnothing$。
14: Answer = $\cup F_k$。

注意，该算法的结构几乎与算法 6.1 完全一样。该算法将迭代地产生新的候选 k-序列，剪掉那些其(k–1)-序列非频繁的候选，然后对留下的候选计数，识别序列模式。这些步骤的细节在下面给出。

候选产生　一对频繁 (k–1)-序列合并，产生候选 k-序列。为了避免重复产生候选，传统的 *Apriori* 算法仅当前 k–1 项相同时才合并一对频繁 k-项集。类似的方法可以用于序列。序列合并的原则在以下过程中给出。

图 7-6 给出了一个例子，通过合并成对的频繁 3-序列得到候选 4-序列。第一个候选$\langle\{1\}\{2\}\{3\}\{4\}\rangle$通过合并$\langle\{1\}\{2\}\{3\}\rangle$和$\langle\{2\}\{3\}\{4\}\rangle$得到。由于事件 3 和事件 4 属于第二个序列的不同元素，它们在合并后序列中也属于不同的元素。另一方面，将$\langle\{1\}\{5\}\{3\}\rangle$与$\langle\{5\}\{3, 4\}\rangle$合并产生候选 4-序列

⟨{1}{5}{3, 4}⟩。在这种情况下，事件 3 和事件 4 属于第二个序列的相同元素，4 被合并到第一个序列的最后一个元素中。最后，序列⟨{1}{2}{3}⟩与⟨{1}{2, 5}⟩不必合并，因为去掉第一个序列的第一个事件与去掉第二个序列的最后一个事件并不产生相同的子序列。尽管⟨{1}{2,5}{3}⟩是一个可行的候选，但是它是通过合并另外一对序列⟨{1}{2, 5}⟩和⟨{2, 5}{3}⟩产生的。该例表明序列合并过程是完备的；即，它不会丢失任何可行的候选，与此同时，它能避免产生重复的候选序列。

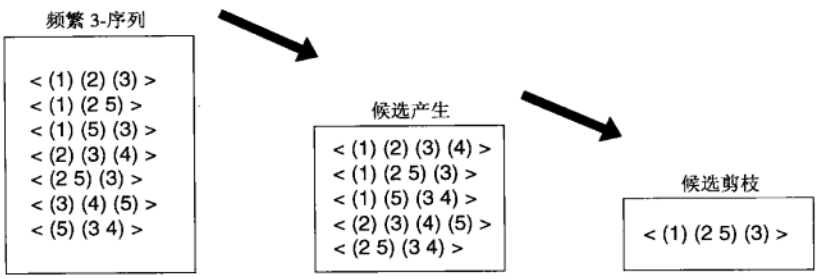


图 7-6 序列模式挖掘算法的候选产生和剪枝步骤的例子

序列合并过程

序列 $s^{(1)}$与另一个序列 $s^{(2)}$合并，仅当从 $s^{(1)}$中去掉第一个事件得到的子序列与从 $s^{(2)}$中去掉最后一个事件得到的子序列相同。结果候选是序列 $s^{(1)}$与 $s^{(2)}$的最后一个事件的连接。$s^{(2)}$的最后一个事件可以作为最后一个事件合并到 $s^{(1)}$的最后一个元素中，也可以作为一个不同的元素，取决于如下条件。

(1) 如果 $s^{(2)}$的最后两个事件属于相同的元素，则 $s^{(2)}$的最后一个事件在合并后的序列中是 $s^{(1)}$的最后一个元素的一部分。

(2) 如果 $s^{(2)}$的最后两个事件属于不同的元素，则 $s^{(2)}$的最后一个事件在合并后的序列中成为连接到 $s^{(1)}$的尾部的单独元素。

候选剪枝 如果候选 k-序列的(k–1)-序列致少有一个是非频繁的，那么它将被剪掉。例如，假定⟨{1}{2}{3}{4}⟩是一个候选 4-序列。我们需要检查⟨{1}{2}{4}⟩和⟨{1}{3}{4}⟩是否是频繁 3-序列。由于它们都不是频繁的，因此可以删除候选⟨{1}{2}{3}{4}⟩。读者可以验证，候选剪枝后，图 7-6 中剩下的唯一候选 4-序列是⟨{1}{2,5}{3}⟩。

支持度计数 在支持度计数期间，算法将枚举属于特定数据序列的所有候选 k-序列。这些候选的支持度将增值。计数之后，算法将识别出频繁 k-序列，并可以丢弃其支持度计数小于最小支持度阈值 *minsup* 的候选。

7.4.3 时限约束

本节提出一种序列模式，其中模式的事件和元素都施加时限约束。为了诱导对时限约束的需要，考虑如下被两个注册数据挖掘课程的学生选修的课程序列：

学生 A: ⟨{统计学}{数据库系统}{数据挖掘}⟩

学生 B: ⟨{数据库系统}{统计学}{数据挖掘}⟩

感兴趣的序列模式是⟨{统计学，数据库系统}{数据挖掘}⟩，意思是说注册数据挖掘课程的学生必须先选修数据库系统和统计学方面的课程。显然，该模式被这两个学生支持，尽管他们都没有同时选修统计学和数据库系统。相比之下，不能认为某个 10 年之前选修了统计学课程的学生支持该模式，因为这些课程的时间间隔太长了。由于上一节提供的表示并未体现时限约束，因此需要定义新的序列模式。

图 7-7 解释了可以施加在模式上的某些时限约束。这些约束的定义和它们对序列模式发现算法的影响将在下面讨论。注意，序列模式的每个元素都与一个时间窗口[l, u]相关联，其中 l 是该时间窗口内事件的最早发生时间，而 u 是该时间窗口内事件的最晚发生时间。

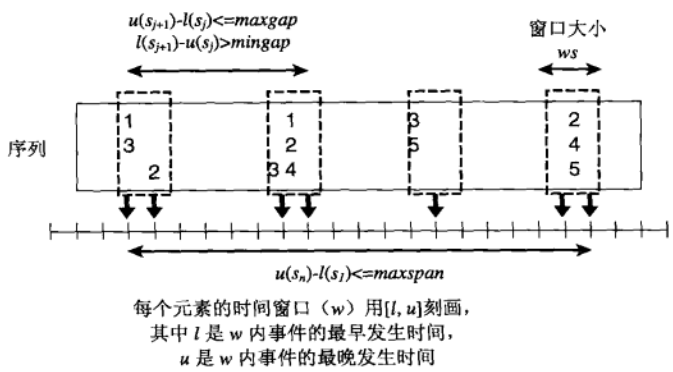


图 7-7　序列模式的时限约束

1. 最大跨度约束

最大跨度约束指定整个序列中所允许的事件的最晚和最早发生时间的最大时间差。例如，假定下面的数据序列包含的事件发生在相继的时间戳（1, 2, 3,…）。假定最大时间跨度 $maxspan = 3$，下面的表包含了给定的数据序列支持和不支持的序列模式。

数据序列 s	序列模式 t	s 支持 t?
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{3} {4}⟩	是
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{3} {6}⟩	是
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{1, 3} {6}⟩	否

一般地，*maxspan* 越长，在数据序列中检测到模式的可能性就能大。然而，较长的 *maxspan* 也可能捕获不真实的模式，因为这增加了两个不相关的事件成为时间相关事件的可能性。此外，模式也可能涉及陈旧事件。

最大跨度约束影响序列模式发现算法的支持度计数。如前面的例子所示，施加最大时间跨度约束之后，有些数据序列就不再支持候选模式。如果我们简单地使用算法 7.1，则有些模式的支持度计数就可能过分估计。为了避免该问题，必须修改算法，忽略给定模式中事件的第一次和最后一次发生的时间间隔大于 *maxspan* 的情况。

2. 最小间隔和最大间隔约束

时限约束也可以通过限制序列中两个相继元素之间的时间差来指定。如果最大时间差

(*maxgap*) 是一周，则元素中的事件必须在前一个元素的事件出现后的一周之内出现。如果最小时间差 (*mingap*) 是零，则元素中的事件必须在前一个元素的事件出现之后出现。假定 *maxgap* = 3，*mingap* = 1，下表给出了模式通过或未通过最大间隔和最小间隔约束的例子。

数据序列 s	序列模式 t	*maxgap*	*mingap*
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{3} {6}⟩	通过	通过
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{6} {8}⟩	通过	未通过
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{1, 3} {6}⟩	未通过	通过
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{1} {3} {8}⟩	未通过	未通过

与最大跨度一样，这些约束也影响序列模式发现算法的支持度计数，因为当最小间隔和最大间隔约束存在时，有些数据序列就不再支持候选模式。必须修改算法，确保对模式进行支持度计数时不会违反时限约束。否则的话，可能将某些非频繁的序列误认为频繁序列。

使用最大间隔约束的一个旁效是可能违反先验原理。为了解释这一点，考虑图7-5中的数据集。由于没有最小间隔或最大间隔约束，⟨{2}{5}⟩和⟨{2}{3}{5}⟩的支持度都是60%。然而，如果 *mingap* = 0，*maxgap* = 1，则⟨{2}{5}⟩的支持度下降至40%，而⟨{2}{3}{5}⟩的支持度仍然是60%。换句话说，当序列中的事件个数增加时，支持度增加了——这与先验原理相违背。出现这种违背的原因是，事件2和事件5之间的时间间隔大于*maxgap*，因而对象D不支持模式⟨{2}{5}⟩。使用邻接子序列的概念可以避免这一问题。

定义 7.2 邻接子序列 序列 s 是序列 $w = \langle e_1 e_2 \cdots e_k \rangle$的邻接子序列（contiguous subsequence），如果下列条件之一成立。

(1) s 是从 e_1 或 e_k 中删除一个事件后由 w 得到。

(2) s 是从至少包含两个事件的任意 $e_i \in w$ 中删除一个事件后由 w 得到。

(3) s 是 t 的邻接子序列，而 t 是 w 的邻接子序列。

下面的例子解释了邻接子序列概念：

数据序列 s	序列模式 t	t 是 s 的邻接子序列?
⟨{1} {2, 3}⟩	⟨{1} {2}⟩	是
⟨{1, 2} {2} {3}⟩	⟨{1} {2}⟩	是
⟨{3, 4} {1, 2} {2, 3} {4}⟩	⟨{1} {2}⟩	是
⟨{1} {3} {2}⟩	⟨{1} {2}⟩	否
⟨{1, 2} {1} {3} {2}⟩	⟨{1} {2}⟩	否

使用邻接子序列概念，可以用如下方法修改先验原理，来处理最大间隔约束。

定义 7.3 修订的先验原理 如果一个 k-序列是频繁的，则它的所有邻接 (k–1)-子序列也一定是频繁的。

只需少量改动，就可以将修订的先验原理用于序列模式发现算法。在候选剪枝阶段，并非所有的k-序列都需要检查，因为它们之中的一些可能违反最大间隔约束。例如，如果*maxgap* = 1，则不必检查候选⟨{1}{2, 3}{4}{5}⟩的子序列⟨{1}{2, 3}{5}⟩是否是频繁的，因为元素{2, 3}和{5}之间的时间差大于一个时间单位。我们只需要考察⟨{1}{2, 3}{4}{5}⟩的邻接子序列，包括⟨{1}{2,

3}{4}⟩，⟨{2, 3}{4}{5}⟩，⟨{1}{2}{4}{5}⟩和⟨{1}{3}{4}{5}⟩。

3. 窗口大小约束

最后，元素 s_j 中的事件不必同时出现。可以定义一个窗口大小阈值（*ws*）来指定序列模式的任意元素中事件最晚和最早出现之间的最大允许时间差。窗口大小为 0 表明模式同一元素中的所有事件必须同时出现。

下面的例子使用 $ws = 2$，确定数据序列是否支持给定的序列（假定 $mingap = 0$，$maxgap = 3$，$maxspan = \infty$）。

数据序列 *s*	序列模式 *t*	*s* 支持 *t*?
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{3, 4} {5}⟩	是
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{4, 6} {8}⟩	是
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{3, 4, 6} {8}⟩	否
⟨{1, 3} {3, 4} {4} {5} {6, 7} {8}⟩	⟨{1, 3, 4} {6, 7, 8}⟩	否

在上一个例子中，尽管模式⟨{1, 3, 4} {6, 7, 8}⟩满足窗口大小约束，但是它违反最大间隔约束，因为两个元素中事件的最大时间差是 5 个时间单位。窗口大小约束也影响序列模式发现算法的支持度计数。如果直接使用算法 7.1 而不施加窗口大小约束，则某些候选模式的支持度计数可能过低估计，从而可能丢掉某些有趣的模式。

7.4.4　可选计数方案

有一些方法可以用来由序列数据库对候选 *k*-序列的支持度计数。为了解释，考虑序列⟨{*p*}{*q*}⟩的支持度计数问题，如图 7-8 所示。假定 $ws = 0$，$mingap = 0$，$maxgap = 1$，$maxspan = 2$。

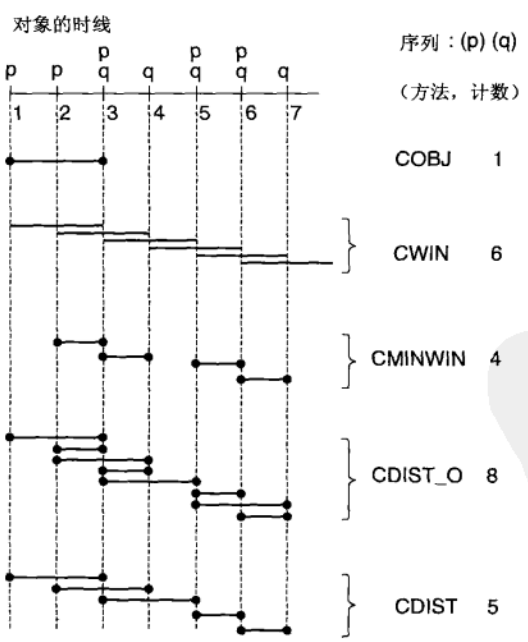


图 7-8　比较不同的支持度计数方法

- COBJ：每个对象出现一次。该方法在对象时线中查找给定序列的至少一次出现。在图 7-8 中，尽管序列⟨{p}{q}⟩在对象的时线中出现多次，但是只对它计数一次——p 出现在 $t=1$，而 q 出现在 $t=3$。
- CWIN：每个滑动窗口出现一次。在该方法中，将一个固定长度（*maxspan*）的滑动时间窗口移过时线，一次移动一个时间单位。序列在滑动窗口中遇到一次，支持度计数就增值一次。在图 7-8 中，使用该方法，序列⟨{p}{q}⟩被观察到 6 次。
- CMINWIN：最小出现窗口数。最小出现窗口是给定时限约束下序列出现的最小窗口。换言之，最小出现窗口是这样的时间间隔，使得序列在该时间间隔中出现，但不在其任何真子间隔中出现。该定义可以视为 CWIN 的限制版本，因为其效果是收缩或坍塌被 CWIN 计数的某些窗口。例如，序列⟨{p}{q}⟩有 4 个最小出现窗口：(1)对(p: $t=2$, q: $t=3$)，(2)对(p: $t=3$, q: $t=4$)，(3)对(p: $t=5$, q: $t=6$)，(4)对(p: $t=6$, q: $t=7$)。事件 p 在 $t=1$ 且事件 q 在 $t=3$ 的出现不是最小出现窗口，因为它包含一个更小的窗口(p: $t=2$, q: $t=3$)，该子窗口才是最小出现窗口。
- CDIST_O：允许事件-时间戳重叠的不同出现（distinct occurrence）。序列的不同出现定义为事件-时间戳对的集合，使得至少有一个新的事件-时间戳对不同于以前统计过的出现。对这样的不同出现计数就产生了 CDIST_O 方法。如果事件 p 和 q 的出现时间表示为元组($t(p)$, $t(q)$)，则该方法产生序列⟨{p}{q}⟩的 8 个不同出现，分别在时间(1,3)、(2,3)、(2,4)、(3,4)、(3,5)、(5,6)、(5,7)和(6,7)。
- CDIST：不允许事件-时间戳重叠的不同出现。在上面的 CDIST_O 中，允许序列的两次出现具有重叠的事件-时间戳对，如(1,3)和(2,3)。CDIST 方法不允许重叠。当一个事件-时间戳对在计数时用过之后，将它标记为已使用，并且在对相同的序列计数时不再使用。例如，在图 7-8 中，序列⟨{p}{q}⟩的不同的、不重叠的出现有 5 次。这些出现的发生时间分别为(1,3)、(2,4)、(3,5)、(5,6)和(6,7)。可以看出，这些出现是 CDIST_O 观察到的出现的子集。

关于计数方法最后要说的是，需要确定计算支持度度量的基线。对于频繁项集挖掘，基线由事务总数给定。对于序列模式挖掘，基线依赖于计数方法。对于 COBJ 方法，可以用输入数据中对象的总数作为基线。对于 CWIN 和 CMINWIN 方法，基线由所有对象中可能的时间窗口数之和给定。对于诸如 CDIST 和 CDIST_O 方法，基线由每个对象输入数据中出现的不同的时间戳个数之和确定。

7.5 子图模式

本节将关联分析方法应用到远比项集和序列更加复杂实体。例子包括化学化合物、3-D 蛋白质结构、网络拓扑和树结构的 XML 文档。这些实体可以用图形表示建模，如表 7-7 所示。

在这种类型的数据上进行数据挖掘的任务是，在图的集合中发现一组公共子结构。这样的任务称作**频繁子图挖掘**（frequent subgraph mining）。频繁子图挖掘的潜在应用可以在计算化学领域看到。每年，为了研制药物、农药、化肥等，都要构造新的化合物。尽管我们知道化合物的化学性质主要取决于其结构，但是建立它们之间的确切联系却很困难。通过识别与已知化合物的特定性质相关联的常见子结构，频繁子图挖掘可以为这项工作提供支持。这样的信息可以帮助科学家构造具有特定性质的新化学化合物。

表 7-7　不同应用领域中实体的图形表示

应　用	图　形	顶　点	边
Web 挖掘	Web 浏览模式	Web 页面	页面之间的超链接
计算化学	化学化合物的结构	原子或离子	原子或离子之间的键
网络计算	计算机网络	计算机和服务器	机器之间的互联
语义 Web	XML 文档的集合	XML 元素	元素之间的父子联系
生物信息学	蛋白质结构	氨基酸	接触残基

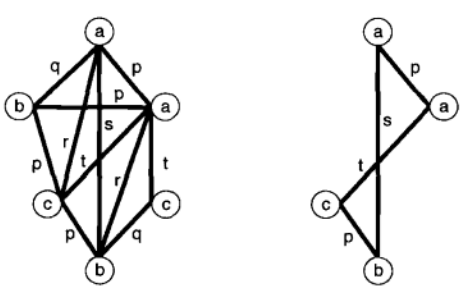
本节提供一些方法，将关联分析用于基于图的数据。首先，我们回顾一些与图有关的基本概念和定义；然后引入频繁子图挖掘问题；接下来介绍如何扩展传统的 *Apriori* 算法来发现这些模式。

7.5.1　图与子图

图是一种可以用来表示实体集之间联系的数据结构。从数学上讲，图由顶点集 V 和连接顶点对的边集 E 构成。每条边用顶点对(v_i, v_j)表示，其中 $v_i, v_j \in V$。可以给每个顶点 v_i 赋予一个标号 $l(v_i)$，代表实体的名字。同理，每条边(v_i, v_j)也可以关联到一个标号 $l(v_i, v_j)$，描述实体对之间的联系。表 7-7 显示了与不同类型的图相关联的顶点和边。例如，在一个 Web 图中，顶点对应于 Web 页面，而边表示 Web 页面之间的超链接。

定义 7.4　子图　图 $G' = (V', E')$是另一个图 $G = (V, E)$的子图，如果它的顶点集 V'是 V 的子集，并且它的边集 E'是 E 的子集。子图关系记作 $G' \subseteq_s G$。

图 7-9 显示了一个包含 6 个顶点和 11 条边的图，以及它的一个可能的子图。该子图显示在图 7-9b 中，只包含原图 6 个顶点中的 4 个，11 条边中的 4 条。



(a) 标记的图　　(b) 子图

图 7-9　子图的例子

定义 7.5　支持度　给定图的集族ς，子图 g 的支持度定义为包含它的所有图所占的百分比，即

$$s(g) = \frac{\left|\{G_i \mid g \subseteq_s G_i, G_i \in \varsigma\}\right|}{|\varsigma|} \tag{7-2}$$

例 7.2　考虑 5 个图 G_1 到 G_5，如图 7-10 所示。右上角的图 g_1 是 G_1、G_3、G_4 和 G_5 的子图，因此 $s(g_1) = 4/5 = 80\%$。同理，我们有 $s(g_2) = 60\%$，因为 g_2 是 G_1、G_2 和 G_3 的子图；而 $s(g_3) = 40\%$，因为 g_3 是 G_1 和 G_3 的子图。　□

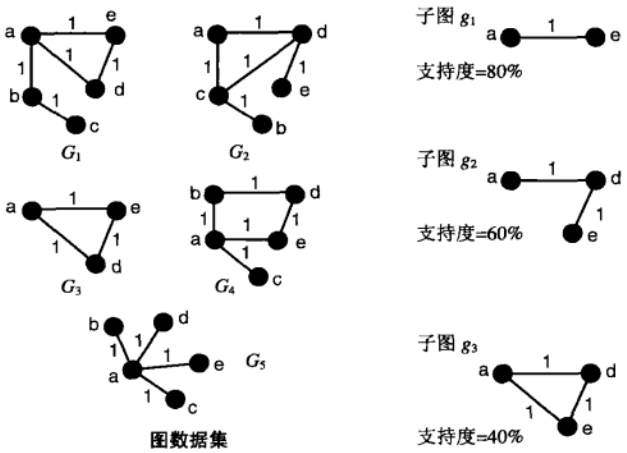


图 7-10 由图集计算子图的支持度

7.5.2 频繁子图挖掘

本节给出频繁子图挖掘问题的定义，并解释该任务的复杂性。

定义 7.6 频繁子图挖掘 给定图的集合ς和支持度阈值*minsup*，频繁子图挖掘的目标是找出所有使得$s(g) \geqslant minsup$的子图g。

尽管该定义适用于所有类型的图，但是本章主要关注**无向连通图**（undirected, connected graph）。这种图的定义如下。

(1) 一个图是连通的，如果图中每对顶点之间都存在一条路径；其中，路径是顶点的序列$<v_1 v_2 \cdots v_k>$，使得序列中每对相邻的顶点(v_i, v_{i+1})之间都有一条边。

(2) 一个图是无向的，如果它只包含无向边。一条边(v_i, v_j)是无向的，如果它与(v_j, v_i)无区别。

处理其他类型（有向的或不连通的）子图的方法留给读者作为习题（见本章习题 15）。

挖掘频繁子图是一项计算量很大的任务，因为搜索空间是指数的。为了解释这项任务的复杂性，考虑包含 d 个实体的数据集。在频繁项集挖掘中，每个实体是一个项，待考察的搜索空间大小是 2^d，这是可能产生的候选项集的个数。在频繁子图挖掘中，每个实体是一个顶点，并且最多可以有 d–1 条到其他顶点的边。假定顶点的标号是唯一的，则子图的总数是

$$\sum_{i=1}^{d} C_d^i \times 2^{i(i-1)/2}$$

其中，C_d^i 是选择 i 个顶点形成子图的方法数，而 $2^{i(i-1)/2}$ 是子图的顶点之间边的最大值。表 7-8 对不同的 d 比较了项集和子图的个数。

表 7-8 对于不同的维度 d，项集和子图的个数比较

实体个数 d	1	2	3	4	5	6	7	8
项集个数	2	4	8	16	32	64	128	256
子图个数	2	5	18	113	1 450	40 069	2 350 602	28 619 2513

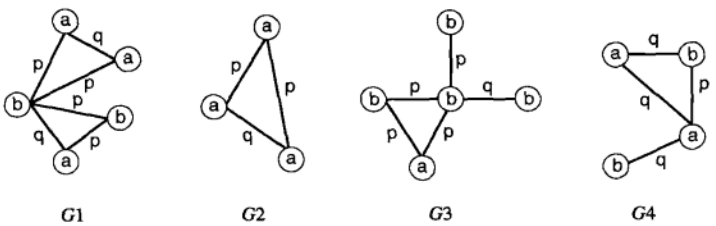
候选子图的个数实际上少得多，因为表 7-8 给出的个数包含了不连通的子图。不连通的子图通常被忽略，因为它们没有连通子图令人感兴趣。

挖掘频繁子图的一种蛮力方法是，产生所有的连通子图作为候选，并计算它们各自的支持度。例如，考虑图 7-11a 中显示的图。假定顶点标号选自集合$\{a, b\}$，而边的标号选自集合$\{p, q\}$，则具有一个到三个顶点的连通子图列在图 7-11b 中。候选子图的个数比传统的关联规则挖掘中的候选项集的个数大得多，其原因如下。

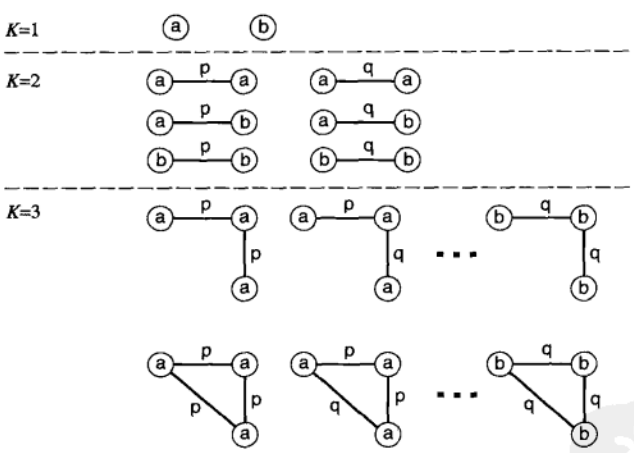
(1) 项在某个项集中至多出现一次，而某个顶点标号可能在一个图中出现多次。

(2) 相同的顶点标号对可以有多种边标号选择。

给定大量候选子图，即使对于规模适度的图，蛮力方法也可能垮掉。



(a) 图数据集的例子



(b) 连通子图的列表

图 7-11　挖掘频繁子图的蛮力方法

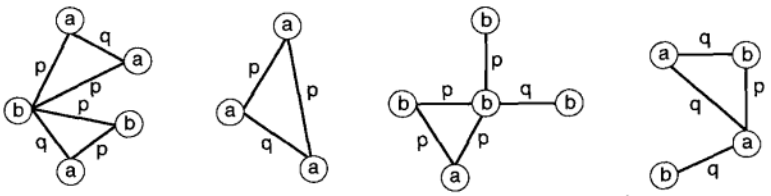
7.5.3　类 *Apriori* 方法

本节考察如何开发一种类 *Apriori* 算法来找出频繁子图。

1. 数据变换

一种可行的方法是将图变换成类似于事务的形式，使得我们可以使用诸如 *Apriori* 等已有的算法。图 7-12 解释了如何将图的集族变换成等价的购物篮表示。在这种表示下，边标号 $l(e)$与对

应的顶点标号($l(v_i)$, $l(v_j)$)组合被映射到一个“项”。“事务”的宽度由图的边数决定。尽管很简单，但是仅当图中每一条边都具有唯一的顶点和边标号组合时，该方法才可行。否则，就不能使用这种表示对图正确建模。



	(a,b,p)	(a,b,q)	(a,b,r)	(b,c,p)	(b,c,q)	(b,c,r)	...	(d,e,r)
G1	1	0	0	0	0	1	...	0
G2	1	0	0	0	0	0	...	0
G3	0	0	1	1	0	0	...	0
G4	0	0	0	0	0	0	...	0

图 7-12 映射图的集族到购物篮事务

2. 频繁子图挖掘算法的一般结构

挖掘频繁子图的类 *Apriori* 算法由以下步骤组成。

(1) **候选产生**：合并频繁(k–1)-子图对，得到候选 k-子图。

(2) **候选剪枝**：丢弃包含非频繁的(k–1)-子图的所有候选 k-子图。

(3) **支持度计数**：统计ς中包含每个候选的图的个数。

(4) **候选删除**：丢弃支持度小于 *minsup* 的所有候选子图。

这些步骤的具体细节在本节的余下部分讨论。

7.5.4 候选产生

在候选产生阶段，将一对频繁(k–1)-子图合并成一个候选 k-子图。首要问题是如何定义子图的大小 k。在图 7-11 显示的例子中，k 是图中的顶点个数。通过添加一个顶点，迭代地扩展子图的方法称作**顶点增长**（vertex growing）。k 也可以是图中边的个数。添加一条边到已有的子图中来扩展子图的方法称作**边增长**（edge growing）。

为了避免产生重复的候选，我们可以对合并施加附加的条件：两个(k–1)-子图必须共享一个共同的(k–2)-子图。共同的(k–2)-子图称作**核**（core）。下面，我们简要描述顶点增长和边增长策略的候选产生过程。

1. 通过顶点增长产生候选

顶点增长是通过添加一个新的顶点到一个已经存在的频繁子图上，产生新候选的过程。在介绍该方法之前，我们首先考虑图的邻接矩阵表示。矩阵的每一项 $M(i, j)$或者包含连接顶点 v_i和 v_j 的边标号，或者为 0（顶点之间没有边）。顶点增长方法可以看作合并一对(k–1)× (k–1)的邻接矩阵，产生 k× k 邻接矩阵的过程，如图 7-13 所示。$G1$ 和 $G2$ 是两个图，其邻接矩阵分别为 $M(G1)$和 $M(G2)$。图中虚线框指出了图的核。通过顶点增长产生候选子图的过程在下面给出。

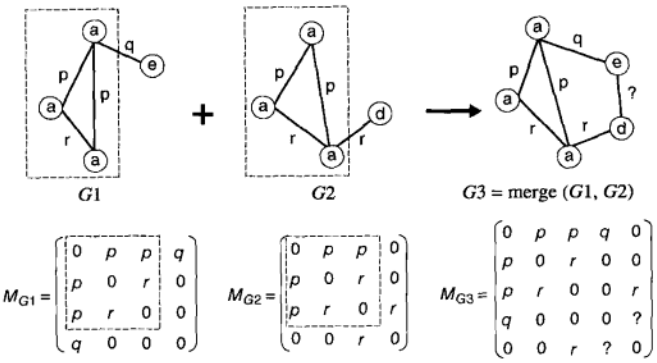


图 7-13　顶点增长策略

通过顶点增长合并子图过程

邻接矩阵 $M^{(1)}$与另一个邻接矩阵 $M^{(2)}$合并，如果删除 $M^{(1)}$和 $M^{(2)}$的最后一行和最后一列得到的子矩阵相同。结果矩阵是 $M^{(1)}$，添加上 $M^{(2)}$的最后一行和最后一列。新矩阵的其余项或者为 0，或者用连接顶点对的合法的边标号替换。

结果图包含的边比原来的图多一条或两条。在图 7-13 中，*G*1 和 *G*2 都包含 4 个顶点和 4 条边。合并之后，结果图 *G*3 有 5 个顶点。*G*3 中边的数目取决于顶点 *d* 和 *e* 是否相连。如果 *d* 和 *e* 是不相连的，则 *G*3 有 5 条边，并且(*d*, *e*)对应的矩阵项为 0；否则，*G*3 有 6 条边，并且(*d*, *e*)的矩阵项对应于新创建的边的标号。由于该边的标号未知，我们需要对(*d*, *e*)考虑所有可能的边标号，从而大大增加了候选子图的个数。

2. 通过边增长产生候选

在候选产生期间，边增长将一个新的边插入一个已经存在的频繁子图中。与顶点增长不同，结果子图的顶点个数不一定增加。图 7-14 显示了通过边增长策略合并 *G*1 和 *G*2 得到的两个可能的候选子图。第一个候选子图 *G*3 多了一个顶点，而第二个候选子图 G4 的顶点个数与原来的图一样。图中虚线框指出了图的核。

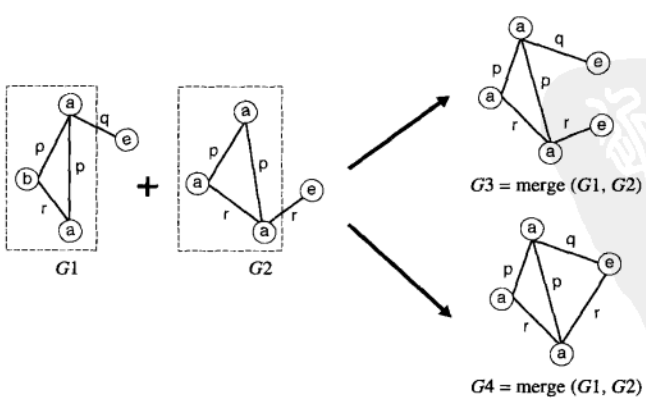


图 7-14　边增长策略

通过边增长产生候选子图的过程概括如下。

通过边增长合并子图过程

一个频繁子图 $g^{(1)}$与另一个频繁子图 $g^{(2)}$合并，仅当从 $g^{(1)}$删除一条边得到的子图与从 $g^{(2)}$删除一条边得到的子图拓扑等价。合并后，结果子图是 $g^{(1)}$，添加 $g^{(2)}$的那条额外的边。

待合并的图可能包含许多互相**拓扑等价**（topologically equivalent）的顶点。为了解释顶点拓扑等价的概念，考虑图 7-15 中所示的图。图 *G*1 包含 4 个顶点，具有相同的标号“*a*”。如果一条新边附着到 4 个顶点的任何一个上，结果图看上去都一样。因此，*G*1 的顶点都相互拓扑等价。

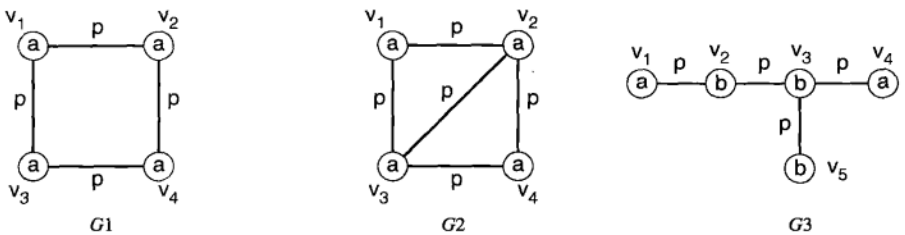


图 7-15 顶点拓扑等价的解释

图 *G*2 有两对拓扑等价的顶点：v_1 与 v_4，v_2 与 v_3，尽管顶点和边的标号都相同。容易看出 v_1 不拓扑等价于 v_2，因为它们附着的边数不同。因此，附着一条新边到 v_1 将导致与附着同样的边到 v_2 不同的图。然而，图 *G*3 没有任何拓扑等价的顶点。尽管 v_1 和 v_4 具有相同的顶点标号和相等的附着边数，但是附着一条新边到 v_1 与附着同样的边到 v_4 将导致不同的图。

顶点拓扑等价的概念能够帮助我们理解，在边增长时为什么能够产生多个候选子图。考虑图 7-16 中(*k*–1)-子图 *G*1 和 *G*2。为简单起见，它们的核（包含两个图的 *k*–2 条共同边）画成了一个矩形框。*G*1 剩下的一条不在核中的边显示为连接顶点 *a* 和 *b* 的悬挂边。类似地，*G*2 剩下的一条不在核中的边显示为连接顶点 *c* 和 *d* 的悬挂边。尽管 *G*1 和 *G*2 的核相同，但是 *a* 和 *c* 可能拓扑等价也可能不等价。如果 *a* 和 *c* 拓扑等价，我们将它们记作 $a = c$。对于核外边的点，如果它们的标号相同，我们将它们记作 $b = d$。

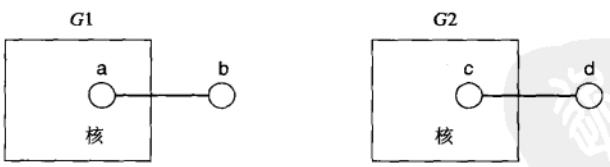


图 7-16 通过边增长合并一对子图的一般方法

下面的规则可以用来确定候选产生得到的候选子图。

(1) 如果 $a \neq c$ 并且 $b \neq d$，则只有一个可能的结果子图，如图 7-17a 所示。

(2) 如果 $a = c$ 但是 $b \neq d$，则有两个可能的结果子图，如图 7-17b 所示。

(3) 如果 $a \neq c$ 但是 $b = d$，则有两个可能的结果子图，如图 7-17c 所示。

(4) 如果 $a = c$ 并且 $b = d$，则有三个可能的结果子图，如图 7-17d 所示。

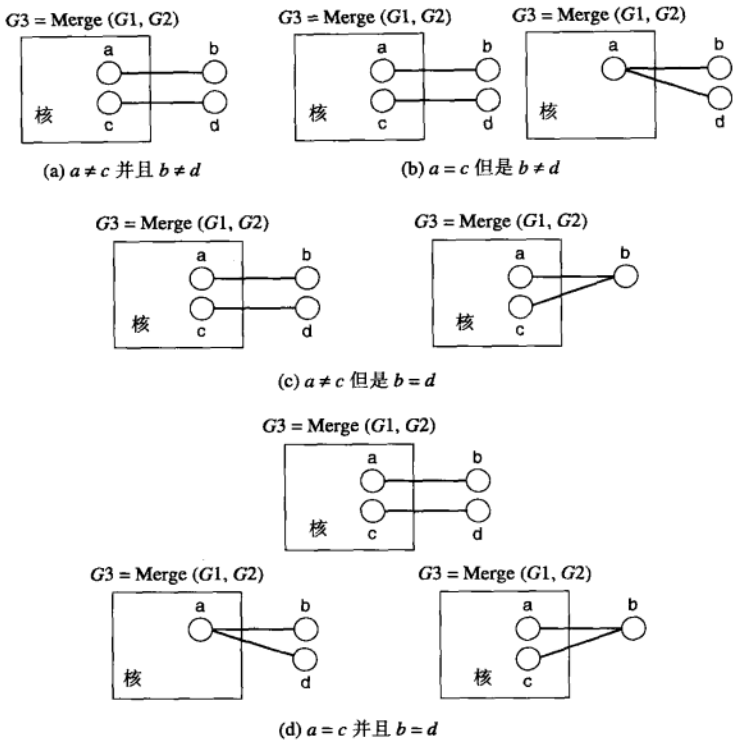


图 7-17　通过边增长产生的候选子图

当与一对(*k*–1)-子图相关联的核有多个时，还可能产生多个候选子图，如图 7-18 所示。带阴影的顶点对应于合并时形成核的顶点。每一个核都可能导致不同的候选子图集。原则上，如果合并一对频繁(*k*–1)-子图，则最多可以有 *k*–2 个核，每个核通过从被合并的图中删除一条边得到。尽管边增长过程可能产生多个候选子图，但是候选子图的数量趋向于比顶点增长策略产生的少。

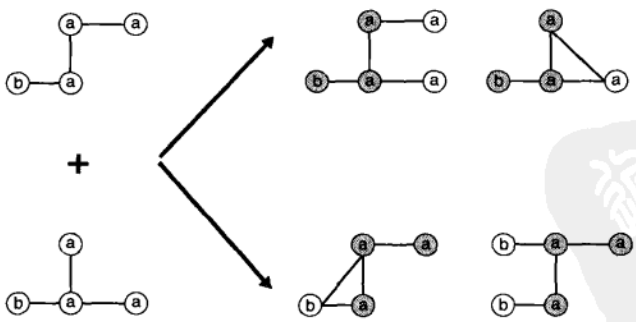


图 7-18　候选产生过程中候选的多重性

7.5.5　候选剪枝

产生候选 *k*-子图后，需要剪去 (*k*–1)-子图非频繁的候选。候选剪枝可以通过如下步骤实现：

相继从 k-子图删除一条边，并检查对应的(k–1)-子图是否连通且频繁。如果不是，则该候选 k-子图可以丢弃。

为了检查(k–1)-子图是否频繁，需要将它与其他频繁(k–1)-子图匹配。判定两个图是否拓扑等价（或同构）称为**图同构**（graph isomorphism）问题。为了解释解决图同构问题的困难性，考虑图7-19中的两个图。尽管这两个图看上去不同，但是它们是同构的，因为在这两个图的顶点之间存在一个1-1映射。

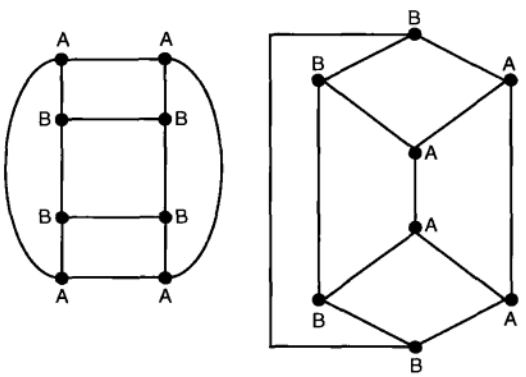


图 7-19 图同构

处理图同构

处理图同构问题的标准方法是，将每一个图都映射到一个唯一的串表达式，称作**代码**（code）或**规范标号**（canonical label）。规范标号具有如下性质：如果两个图是同构的，则它们的代码一定相同。这个性质使得我们可以通过比较图的规范标号来检查图同构。

构造图的规范标号的第一步是找出图的邻接矩阵表示。图 7-20 给出了一个给定图的邻接矩阵的例子。原则上，图可以有多种邻接矩阵表示，因为存在多种确定顶点次序的方法。在图 7-20 中，第一行和第一列对应于具有 3 条边的顶点 a，第二行和第二列对应于另一个具有 2 条边的顶点 a，如此等等。为了导出图的所有邻接矩阵表示，我们需要考虑矩阵行（和对应列）的所有可能的排列。

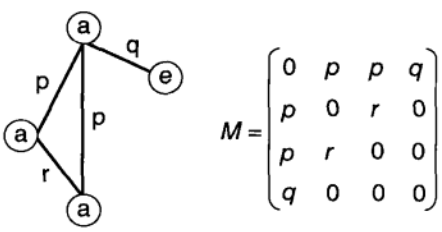


图 7-20 图的邻接矩阵表示

数学上讲，每个排列都对应于初始邻接矩阵与一个对应的排列矩阵的乘积，如下面的例子所示。

例 7.3　考虑下面的矩阵：

$$M=\begin{pmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\\ 13 & 14 & 15 & 16\end{pmatrix}$$

下面的排列矩阵可以用来交换 M 的第一行（和列）和第三行（和列）：

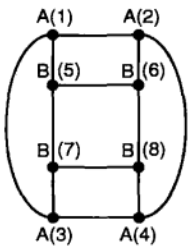
$$P_{13}=\begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

其中，P_{13} 是通过交换单位矩阵的第一行和第三行得到的。为了交换 M 的第一和第三行（和列），排列矩阵与 M 相乘：

$$M'=P_{13}^{T}\times M\times P_{13}=\begin{pmatrix}11 & 10 & 9 & 12\\ 7 & 6 & 5 & 8\\ 3 & 2 & 1 & 4\\ 15 & 14 & 13 & 16\end{pmatrix}$$

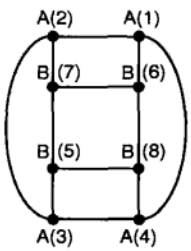
注意，M 右乘 P_{13} 交换 M 的第一列和第三列，而 M 左乘 P_{13}^{T} 交换 M 的第一行和第三行。如果三个矩阵相乘，则产生矩阵 M'，其第一、三行交换，第一、三列交换。　□

第二步是确定每个邻接矩阵的串表示。由于邻接矩阵是对称的，因此只需要根据矩阵的上三角部分构造串表示就足够了。在图7-21所示的例子中，代码是通过逐列连接矩阵的上三角元素得到的。最后一步是比较图的所有串表达式，并选出具有最小（最大）字典次序值的串。



	A(1)	A(2)	A(3)	A(4)	B(5)	B(6)	B(7)	B(8)
A(1)	0	1	1	0	1	0	0	0
A(2)	1	0	0	1	0	1	0	0
A(3)	1	0	0	1	0	0	1	0
A(4)	0	1	1	0	0	0	0	1
B(5)	1	0	0	0	0	1	1	0
B(6)	0	1	0	0	1	0	0	1
B(7)	0	0	1	0	1	0	0	1
B(8)	0	0	0	1	0	1	1	0

Code = 11001110000100100101000010 11



	A(1)	A(2)	A(3)	A(4)	B(5)	B(6)	B(7)	B(8)
A(1)	0	1	0	1	0	1	0	0
A(2)	1	0	1	0	0	0	1	0
A(3)	0	1	0	1	1	0	0	0
A(4)	1	0	1	0	0	0	0	1
B(5)	0	0	1	0	0	0	1	1
B(6)	1	0	0	0	0	0	1	1
B(7)	0	1	0	0	1	1	0	0
B(8)	0	0	0	1	1	1	0	0

Code = 10110100101000001001100011 10

图 7-21　邻接矩阵的串表达式

前面的方法开销很大，因为它需要考察图的所有可能的邻接矩阵，并计算它们的串表达式，以便找出规范标号。具体地说，对于包含 k 个顶点的图，需要考虑 $k!$种排列。已经开发了一些方法来降低该任务的复杂度，包括存放先前计算的规范标号（这样，当对同一个图进行同构测试时，我们就不必重新计算它）以及通过加入诸如顶点标号和顶点的度数等附加信息来减少确定规范标号所需要的排列数。后一种方法已经超出了本书的范围，但是有兴趣的读者可以参考本章结尾的文献注释。

7.5.6 支持度计数

支持度计数也可能是开销很大的操作，因为对于每个 $G \in \varsigma$，必须确定包含在 G 中的所有候选子图。加快该操作的一种方法是，维护一个与每个频繁$(k-1)$-子图相关联的图 ID 表。如果新的候选 k-子图通过合并一对频繁$(k-1)$-子图而产生，就对它们的对应图 ID 表求交集。最后，子图同构检查就在表中的图上进行，确定它们是否包含特定的子图。

7.6 非频繁模式

迄今为止，关联分析都基于这样的前提：项在事务中出现比不出现更重要。因此，数据库中很少出现的模式不是令人感兴趣的，并使用支持度度量将其删除。这种模式称为非频繁模式。

定义 7.7 非频繁模式 非频繁模式是一个项集或规则，其支持度小于阈值 *minsup*。

尽管绝大部分非频繁模式都不是令人感兴趣的，但是其中的一些可能对于分析是有用的，特别是涉及到数据中的负相关时。例如，DVD 和 VCR 一起销售的情况很少，因为购买 DVD 的顾客多半不会购买 VCR，反之亦然。这种负相关模式有助于识别**竞争项**（competing item），即可以相互替代的项。竞争项的例子包括茶与咖啡、黄油与人造黄油、普通与节食苏打以及台式与便携式计算机。

某些非频繁模式也可能暗示数据中出现了某些有趣的罕见事件或例外情况。例如，如果{火灾=Yes}是频繁的，但{火灾=Yes，警报=On}是非频繁的，则后者是有趣的非频繁模式，因为它可能指出警报系统的故障。为了检测这种不寻常情况，必须确定模式的期望支持度，以便当模式的支持度明显低于期望支持度时，可以声明它是一个有趣的非频繁模式。

挖掘非频繁模式是一个挑战，因为可以从给定的数据集导出大量这种模式。具体地说，挖掘非频繁模式的关键问题是：(1)如何识别有趣的非频繁模式，(2)如何在大型数据集中有效地发现它们。为了获得对各种类型的有趣的非频繁模式的不同看法，7.6.1 节和 7.6.2 节分别介绍两个相关的概念——负模式和负相关模式。这些模式之间的关系在 7.6.3 节阐述。最后，7.6.5 节和 7.6.6 节提供两类技术，用来挖掘有趣的非频繁模式。

7.6.1 负模式

设 $I=\{i_1, i_2, \cdots, i_d\}$是项的集合。**负项** $\overline{i_k}$ 表示项 i_k 不在给定的事务中出现。例如，如果事务中不包含咖啡，则 $\overline{\text{咖啡}}$ 是一个值为 1 的负项。

定义 7.8 负项集 负项集 X 是一个具有如下性质的项集：(1) $X=A \cup \overline{B}$，其中 A 是正项的集合，而 $\overline{B}$ 是负项的集合，$|\overline{B}| \geqslant 1$；(2) $s(X) \geqslant minsup$。

定义 7.9　负关联规则　负关联规则是具有如下性质的关联规则：(1)规则是从负项集提取的，(2)规则的支持度大于或等于 *minsup*，(3)规则的置信度大于或等于 *minconf*。

本章中，负项集和负关联规则统称**负模式**（negative pattern）。负关联规则的一个例子是茶→$\overline{\text{咖啡}}$，它暗示喝茶的人倾向于不喝咖啡。

7.6.2　负相关模式

6.7.1 节介绍了如何使用相关分析来分析两个分类变量之间的联系。已经证明，对于发现正相关的项集，诸如兴趣因子公式（6-5）和ϕ系数公式（6-8）等度量是有用的。本节将这些讨论扩展到负相关模式。

用 $X = \{x_1, x_2, \cdots, x_k\}$表示 k-项集，$P(X)$表示事务包含 X 的概率。在关联分析中，这个概率通常用项集的支持度 $s(X)$估计。

定义 7.10　负相关项集　项集 X 是负相关的，如果

$$s(X) < \prod_{j=1}^{k} s(x_j) = s(x_1) \times s(x_2) \times \cdots \times s(x_k) \tag{7-3}$$

其中，$s(x_j)$是项 x_j 的支持度。

上面表达式的右端 $\prod_{j=1}^{k} s(x_j)$ 给出了 X 中的所有项统计独立的概率估计。定义 7.10 暗示，项集是负相关的，如果它的支持度小于使用统计独立性假设计算出的期望支持度。$s(X)$越小，模式就越负相关。

定义 7.11　负相关关联规则　关联规则 $X \rightarrow Y$ 是负相关的，如果

$$s(X \cup Y) < s(X)s(Y) \tag{7-4}$$

其中，X 和 Y 是不相交的项集，即 $X \cap Y = \varnothing$。

上面的定义只提供了 X 中的项与 Y 中的项之间负相关的部分条件。负相关的完全条件可以表述如下：

$$s(X \cup Y) < \prod_{i} s(x_i) \prod_{j} s(y_j) \tag{7-5}$$

其中，$x_i \in X$ 而 $y_i \in Y$。由于 X 中（Y 中）的项通常是正相关的，因此使用部分条件而不是完全条件来定义负相关关联规则更实际。例如，尽管根据不等式（7-4），规则

{眼镜，镜头清洁剂} → {隐形眼镜，盐溶液}

是负相关的，但是眼镜与镜头清洁剂是正相关的，隐形眼镜与盐溶液是正相关的。如果使用不等式（7-5），这样的规则可能发现不了，因为它可能不满足负相关的完全条件。

负相关条件也可以用正项集和负项集的支持度表示。设 $\overline{X}$ 和 $\overline{Y}$ 分别表示 X 和 Y 的对应负项集，由于

$$
\begin{aligned}
& s(X \cup Y) - s(X)\, s(Y) \\
&= s(X \cup Y) - [s(X \cup Y) + s(X \cup \overline{Y})][\, s(X \cup Y) + s(\overline{X} \cup Y)] \\
&= s(X \cup Y)[1 - s(X \cup Y) - s(X \cup \overline{Y}) - s(\overline{X} \cup Y)] - s(X \cup \overline{Y})\, s(\overline{X} \cup Y) \\
&= s(X \cup Y)\, s(\overline{X} \cup \overline{Y}) - s(X \cup \overline{Y})\, s(\overline{X} \cup Y)
\end{aligned}
$$

负相关条件可以表述如下：

$$s(X \cup Y)\, s(\overline{X} \cup \overline{Y}) < s(X \cup \overline{Y})\, s(\overline{X} \cup Y) \tag{7-6}$$

本章中，负相关项集和负相关关联规则统称**负相关模式**（negatively correlated pattern）。

7.6.3 非频繁模式、负模式和负相关模式比较

非频繁模式、负模式和负相关模式是三个密切相关的概念。尽管非频繁模式和负相关模式只涉及包含正项的项集或模式，而负模式涉及包含正项和负项的项集或模式，但是这三个概念之间存在一定的共性，如图 7-22 所示。

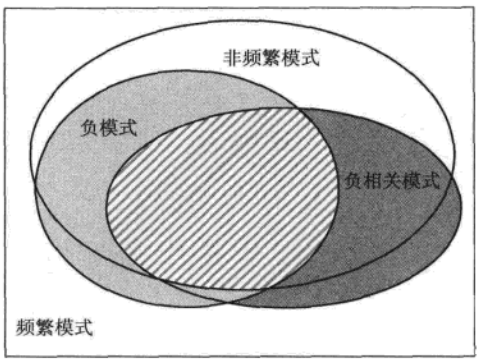


图 7-22 非频繁模式、负模式和负相关模式比较

首先，许多非频繁模式有对应的负模式。为了解释为什么出现这种情况，考虑表 7-9 显示的列联表。如果 $X \cup Y$ 是非频繁的，则除非 *minsup* 太高，否则它很可能有对应的负项集。例如，假定 *minsup*$\leqslant$0.25，如果 $X \cup Y$ 是非频繁的，则项集 $X \cup \overline{Y}$、$\overline{X} \cup Y$ 和 $\overline{X} \cup \overline{Y}$ 中至少有一个的支持度肯定大于 *minsup*，因为列联表中的支持度之和为 1。

表 7-9 关联规则 $X \nrightarrow Y$ 的二路列联表

	Y	$\overline{Y}$	
X	$s(X \cup Y)$	$s(X \cup \overline{Y})$	$s(X)$
$\overline{X}$	$s(\overline{X} \cup Y)$	$s(\overline{X} \cup \overline{Y})$	$s(\overline{X})$
	$s(Y)$	$s(\overline{Y})$	1

其次，许多负相关模式也具有对应的负模式。考虑表 7-9 显示的列联表和不等式（7-6）所示的负相关条件。如果 X 和 Y 具有很强的负相关性，则

$$s(X \cup \overline{Y}) \times s(\overline{X} \cup Y) \gg s(X \cup Y) \times s(\overline{X} \cup \overline{Y})$$

因此，当 X 和 Y 负相关时，$X \cup \overline{Y}$ 或者 $\overline{X} \cup Y$ 或者二者必然具有相对较高的支持度。这些项集对应于负模式。

最后，由于 $X \cup Y$ 的支持度越低，该模式就越负相关，因此非频繁的负相关模式趋向于比频繁的负相关模式更令人感兴趣。如图7-22所示，非频繁的、负相关的模式是这两类模式的重叠区域。

7.6.4　挖掘有趣的非频繁模式的技术

原则上讲，非频繁项集是未被标准化的频繁项集产生算法（如 *Apriori* 和 FP 增长）提取的所有项集。这些项集对应于图 7-23 所示的频繁项集边界之下的那些项集。

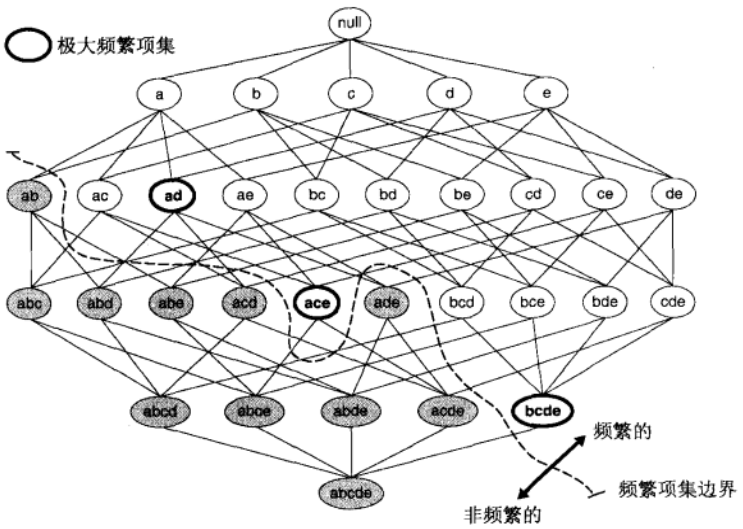


图 7-23　频繁与非频繁项集

由于非频繁模式的数量可能是指数的（特别是对于稀疏的、高维数据），因此，为挖掘非频繁模式而开发的技术着力于仅发现有趣的非频繁模式。这类模式的例子包括 7.6.2 节讨论过的负相关模式。这些模式可以通过删除那些不满足负相关条件（公式（7-3））的非频繁项集得到。这种方法可能是计算密集的，因为必须计算所有非频繁项集的支持度才能确定它们是否是负相关的。与挖掘频繁项集使用的支持度度量不同，挖掘负相关项集使用的基于相关性的度量不具有可以用于指数搜索空间剪枝的反单调性。尽管难以找到有效的解决方案，但是正如本章文献注释所述，已经开发了一些新颖算法。

本章余下部分提供两类技术来挖掘有趣的非频繁模式。7.6.5节介绍挖掘数据中负模式的方法，而7.6.6节介绍基于支持度期望发现有趣的非频繁模式的方法。

7.6.5　基于挖掘负模式的技术

为挖掘非频繁模式开发的第一类技术将每个项看作对称的二元变量。使用7.1 节介绍的方法，通过用负项增广，将事务数据二元化。图 7-24 显示了一个例子，将原始数据变换成具有正项和

负项的事务。对增广的事务使用已有的频繁项集产生算法（如 *Apriori*），可以推导出所有的负项集。

TID	项集
1	{A,B}
2	{A,B,C}
3	{C}
4	{B,C}
5	{B,D}

原来的事务

TID	A	$\overline{A}$	B	$\overline{B}$	C	$\overline{C}$	D	$\overline{D}$
1	1	0	1	0	0	1	0	1
2	1	0	1	0	1	0	0	1
3	0	1	0	1	1	0	0	1
4	0	1	1	0	1	0	0	1
5	0	1	1	0	0	1	1	0

包含负项的事务

图 7-24 用负项增广事务

仅当只有少量变量被视为对称的二元变量时（即负模式仅涉及少量负项），该方法才是可行的。如果每个项都必须视为对称的二元变量，则由于如下原因，该问题就成为难处理的计算问题。

(1) 当每个项都用对应的负项增广时，项的个数就加倍。待探测的项集格比 2^d 大得多（d 是原数据集中项的个数），见本章习题 21。

(2) 当增加进负项后，根据支持度的剪枝将不再有效。对于每个变量x，x或$\overline{x}$的支持度大于等于 50%。因此，即使支持度阈值达到 50%，仍然有一半的项是频繁的。对于较低的阈值，更多的项和包含它们的可能项集都是频繁的。*Apriori*使用的基于支持度的剪枝策略仅当大部分项集的支持度较低时才有效；否则的话，频繁项集的数量呈指数增长。

(3) 当增加进负项后，每个事务的宽度增加。假定原数据集中有 d 个项。对于像购物篮事务那样的稀疏数据集，每个事务的宽度趋向于远小于 d。这样，频繁项集的最大长度 $w_{\max}$ 受限于最大事务的宽度，也趋向于相对较小。当包含负项时，事务的宽度增加到 d，因为一个项要么在事务中，要么不在，而不会既在又不在。由于事务的宽度从 $w_{\max}$ 增加到 d，这将指数地增加频繁项集的数量。其结果是，许多已有的算法用于扩展数据集时都将失败。

前面的蛮力方法计算代价较高，因为它迫使我们确定大量正模式和负模式的支持度。另一种方法不是用负项增广数据集，而是根据对应的正项集计算负项集的支持度。例如，$\{p, \overline{q}, \overline{r}\}$的支持度可以用如下方法计算：

$$s(\{p,\overline{q},\overline{r}\}) = s(\{p\}) - s(\{p,q\}) - s(\{p,r\}) + s(\{p,q,r\})$$

通常，项集 $X \cup \overline{Y}$ 的支持度可以用下式得到：

$$s(X \cup \overline{Y}) = s(X) + \sum_{i=1}^{n} \sum_{Z \subset Y, |Z|=i} \{(-1)^i \times s(X \cup Z)\} \tag{7-7}$$

为了使用公式（7-7），必须对 Y 的每个子集 Z 确定 $s(X \cup Z)$。如果 X 和 Z 的组合的支持度超过最小支持度阈值 *minsup*，则其支持度可以使用 *Apriori* 算法得到，其他组合的支持度必须明确计算。例如，通过扫描整个事务数据集计算。另一种可能的方法是忽略非频繁项集 $X \cup Z$ 的支持度，或用支持度阈值近似。

可以用若干优化策略来进一步提高挖掘算法的性能。首先，可以限制被视为对称二元变量的变量数。具体地说，仅当 y 频繁时才认为负项 $\overline{y}$ 是有趣的。该策略的理由是，稀有项易于产生大

量的非频繁模式，并且其中许多都不是令人感兴趣的。将公式（7-7）中的集合 $\overline{Y}$ 限制于其正项频繁的变量，挖掘算法考虑的候选负项集的个数可能大幅度减少。另一种策略是限制负模式的类型。例如，算法考虑负模式 $X \cup \overline{Y}$，如果它至少包含一个正项（即 $|X| \geqslant 1$）。该策略的理由是，如果数据集包含少量支持度大于 50%的正项，则大部分形如 $\overline{X} \cup \overline{Y}$ 的负模式都将是频繁的，这样就会降低挖掘算法的性能。

7.6.6　基于支持度期望的技术

另一类技术仅当非频繁模式的支持度显著小于期望支持度时，才认为它是有趣的。对于负相关模式，期望支持度根据统计独立性假设计算。本节介绍两种计算期望支持度的方法，使用概念分层和基于近邻的方法，称作**间接关联**（indirect association）。

1. 基于概念分层的支持度期望

仅用客观度量还不足以删除不感兴趣的非频繁模式。例如，假设面包和台式计算机是频繁项。即使项集{面包，台式计算机}是非频繁的，并且可能是负相关，它也不是有趣的，因为对于领域专家，它们的支持度低是显然的。因此，需要确定期望支持度的主观方法，避免产生这种非频繁模式。

在上面的例子中，面包和台式计算机属于两个完全不同的产品类，因此发现他们的支持度低就毫不奇怪。这个例子也解释了使用领域知识剪裁不感兴趣的项的优点。对于购物篮数据，领域知识可以从诸如图 7-25 所显示的概念分层中推断。该方法的基本假设是，预期来自同一类产品的项与其他项具有类似的相互作用。例如，由于火腿和熏肉属于相同的产品族，我们预期火腿和薄片食物之间的关联与熏肉和薄片食物之间的关联类似。如果任何一对的支持度小于其期望支持度，则非频繁模式是有趣的。

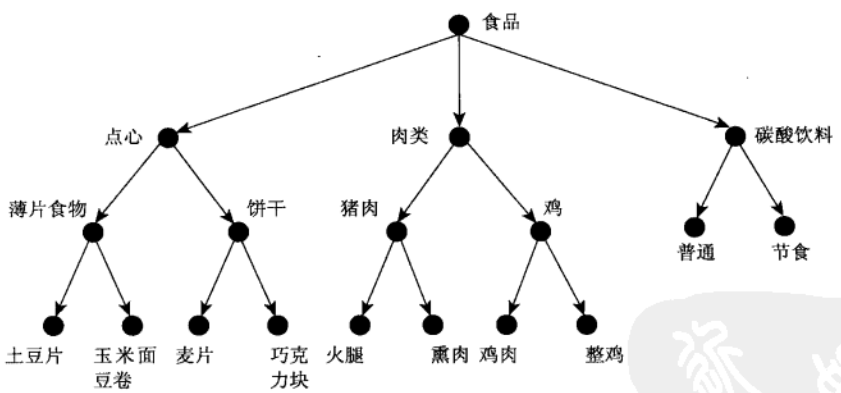


图 7-25　概念分层的例子

为了解释如何计算期望支持度，考虑图 7-26。假定项集{C, G}是频繁的。用 $s(\cdot)$表示模式的实际支持度，而$\varepsilon(\cdot)$表示期望支持度。C 和 G 的子女或兄弟的期望支持度可以用如下公式计算：

$$\varepsilon(s(E,J)) = s(C,G) \times \frac{s(E)}{s(C)} \times \frac{s(J)}{s(G)} \tag{7-8}$$

$$\varepsilon(s(C,J)) = s(C,G)\times\frac{s(J)}{s(G)} \tag{7-9}$$

$$\varepsilon(s(C,H)) = s(C,G)\times\frac{s(H)}{s(G)} \tag{7-10}$$

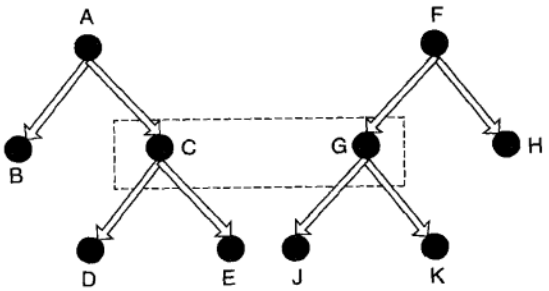


图 7-26 使用概念分层挖掘有趣的非频繁模式

例如，碳酸饮料和点心是频繁的，则节食碳酸饮料和薄片食物的期望支持度可以使用公式（7-8）计算，因为这两个项分别是碳酸饮料和点心的子女。如果节食碳酸饮料和薄片食物的实际支持度明显低于它们的期望值，则节食碳酸饮料和薄片食物形成一个有趣的非频繁模式。

2. 基于间接关联的支持度期望

考虑商品对(a, b)，它们很少被顾客同时购买。如果 a 和 b 是不相关的商品，如面包和 DVD 播放机，则它们的支持度预期较低；如果 a 和 b 是相关的商品，则它们的支持度预期较高。前面使用概念分层计算期望支持度。本节提供一种确定两个商品之间的期望支持度的方法：考察通常与这两个商品一起购买的其他商品。

例如，假定购买睡袋的顾客更有可能也购买其他野营设备，而买台式计算机的顾客也更有可能购买其他计算机附件，如光电鼠标或打印机。假定没有其他商品频繁地与睡袋和台式计算机一起购买，这些不相关的商品的支持度预期较低。另一方面，假定节食和普通碳酸饮料都经常与薄片食物和点心一起购买。即使不使用概念分层，这两种商品可望是相关的，并且它们的支持度应当较高。因为他们的实际支持度低，节食和普通碳酸饮料形成了一个有趣的非频繁模式。这样的模式称作**间接关联**（indirect association）模式。

间接关联的高层解释见图 7-27。项 a 和 b 对应于节食和普通碳酸饮料，而 Y 称作**中介集**（mediator set），包含诸如薄片食物和点心等商品。间接关联的形式定义在下面给出。

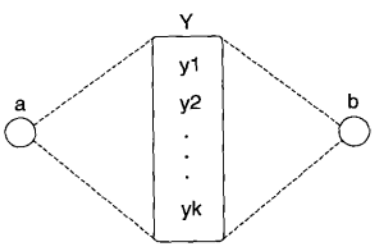


图 7-27 一对项之间的间接关联

定义 7.12 间接关联 一对项 a, b 是通过中介集 Y 间接关联的，如果下列条件成立。

(1) $s(\{a, b\}) < t_s$（项对支持度条件）。

(2) $\exists Y \neq \varnothing$使得

(a) $s(\{a\} \cup Y) \geqslant t_f$并且 $s(\{b\} \cup Y) \geqslant t_f$（中介支持度条件）；

(b) $d(\{a\}, Y) \geqslant t_d, d(\{b\}, Y) \geqslant t_d$，其中 $d(X, Z)$是 X 和 Z 之间关联的客观度量（中介依赖条件）。

注意，中介支持度和依赖条件用来确保 Y 中的项形成 a 和 b 的近邻。可以使用 6.7.1 节介绍的兴趣因子、余弦或 IS、Jaccard 和其他依赖度量。

间接关联有许多可能的应用。在购物篮分析中，a 和 b 可以是竞争商品，如台式计算机和便携式计算机。在文本挖掘中，间接关联可以用来识别同义词、反义词，或用于不同上下文的词。例如，给定一个文档集族，词数据和黄金可以通过中介挖掘间接关联。该模式暗示，词挖掘可以用于两种不同的上下文——数据挖掘与黄金挖掘。

间接关联可以用如下方法产生。首先，使用诸如 *Apriori* 和 FP 增长等标准算法产生频繁项集。然后，合并每对频繁 k-项集得到候选间接关联(a, b, Y)，其中 a 和 b 是一对项，而 Y 是它们的公共中介。例如，若$\{p, q, r\}$和$\{p, q, s\}$是频繁 3-项集，则通过合并这对频繁项集得到候选间接关联$(r, s, \{p, q\})$。产生候选之后，就要验证它是否满足定义 7.12 中的项对支持度和中介依赖条件。然而，中介支持度条件不必验证，因为候选间接关联是通过合并一对频繁项集得到的。该算法汇总在算法 7.2 中。

算法 7.2　挖掘间接关联的算法

```
1: 产生频繁项集的集合 F_k
2: for k = 2 to k_max do
3:    C_k = {(a, b, Y) | {a} ∪ Y ∈ F_k, {b} ∪ Y ∈ F_k, a ≠ b}
4:    for 每个候选(a, b, Y) ∈ C_k do
5:       if s({a, b}) < t_s ∧ d({a}, Y) ≥ t_d ∧ d({b}, Y) ≥ t_d then
6:          I_k = I_k ∪ {(a, b, Y)}
7:       end if
8:    end for
9: end for
10: Result = ∪ I_k
```

文献注释

从分类和连续数据中挖掘关联规则的问题由 Srikant 和 Agrawal 提出[363]。他们的策略是二元化分类属性，使用等频离散化连续属性。他们还提出了**部分完备性**（partial completeness）度量，来确定离散化导致的信息损失量。然后使用该度量来确定所需要的离散区间个数，以确保损失的信息量可以保持在一个期望水平。沿着这一工作，学者们提出了许多挖掘量化关联规则的方法。Aumann 和 Lindell[343]开发了基于统计学的方法，识别在某些定量属性上展示有趣性质的总体段。该形式化方法被其他人进一步扩展，包括 Webb[368]和 Zhang 等[372]。min-*Apriori* 算法由 Han 等[349]提出，用来发现连续数据中的关联规则，而不进行离散化。还有许多研究者也在书中写过挖掘连续数据中关联规则的问题，他们是 Fukuda 等[347]、Lent 等[355]、Wang 等[367]以及 Miller 和 Yang[357]。

7.3 节介绍的使用扩展的事务处理概念分层的方法由 Srikant 和 Agrawal 提出[362]。另一种可

供选择的算法由 Han 和 Fu[350]提出，该算法一次产生一层频繁模式。具体地说，他们的算法在概念分层的顶层产生所有的频繁 1-项集。频繁 1-项集的集合记作 $L(1, 1)$。使用 $L(1, 1)$中的频繁 1-项集，算法进一步产生第一层的所有频繁 2-项集 $L(1, 2)$。重复该过程，直到提取了最高概念层中的所有频繁项集 $L(1, k)$（$k>1$）。然后，基于 $L(1,1)$中的频繁项集，算法继续提取下一个概念层中的频繁项集 $L(2, 1)$。继续该过程，直到处理完用户指定的最低概念层后停止。

7.4节介绍的序列模式的形式化描述和算法由 Agrawal 和 Srikant[341, 364]提出。同样，Mannila 等[356]引进了频繁周期模式的概念，用来从长事件流中挖掘序列模式。另一种形式的序列模式挖掘基于正则表达式，由 Garofalakis 等[348]提出。Joshi 等试图统一各种不同的序列模式表示[352]，其结果是带有不同计数方案（7.4.4节介绍）的序列模式的通用表示。挖掘序列模式的其他可供选择的算法由 Pei 等[359]、Ayres 等[344]、Cheng 等[346]和 Seno 等[361]提出。

频繁子图挖掘问题最早由 Inokuchi 等提出[351]。他们使用了顶点增长方法，由图数据集产生频繁的归约子图。边增长方法由 Kuramochi 和 Karypis 开发[353]。他们还提出了一种类 *Apriori* 算法 FSG，处理诸如多重候选、规范标号、顶点变体等问题。另一种频繁子图挖掘算法称作 gSpan，由 Yang 和 Han 开发[370]。这种算法试图使用最小化的 DFS 码对各种子图编码。频繁子图挖掘的其他变形由 Zaki[371]、Parthasarathy 和 Coatney[358]以及 Kuramochi 和 Karypis[354]提出。

许多研究者都考察了挖掘非频繁模式的问题。Savasere 等[360]考察了使用概念分层挖掘负相关规则。Tan 等[365]提出了挖掘序列和非序列数据的间接关联的思想。挖掘负模式的有效算法由 Boulicaut 等[345]、Teng 等[366]、Wu 等[369]以及 Antonie 和 Zaïane[342]提出。

参考文献

[341] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *Proc. of Intl. Conf. on Data Engineering*, pages 3 - 14, Taipei, Taiwan, 1995.

[342] M.-L. Antonie and O. R. Zaïane. Mining Positive and Negative Association Rules: An Approach for Confined Rules. In *Proc. of the 8th European Conf. of Principles and Practice of Knowledge Discovery in Databases*, pages 27 - 38, Pisa, Italy, September 2004.

[343] Y. Aumann and Y. Lindell. A Statistical Theory for Quantitative Association Rules. In *KDD99*, pages 261 - 270, San Diego, CA, August 1999.

[344] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential Pattern mining using a bitmap representation. In *Proc. of the 8th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 429 - 435, Edmonton, Canada, July 2002.

[345] J.-F. Boulicaut, A. Bykowski, and B. Jeudy. Towards the Tractable Discovery of Association Rules with Negations. In *Proc. of the 4th Intl. Conf on Flexible Query Answering Systems FQAS'00*, pages 425 - 434, Warsaw, Poland, October 2000.

[346] H. Cheng, X. Yan, and J. Han. IncSpan: incremental mining of sequential patterns in large database. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 527 - 532, Seattle, WA, August 2004.

[347] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining Optimized Association Rules for Numeric Attributes. In *Proc. of the 15th Symp. on Principles of Database Systems*, pages 182 - 191, Montreal, Canada, June 1996.

[348] M. N. Garofalakis, R. Rastogi, and K. Shim. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. In *Proc. of the 25th VLDB Conf.*, pages 223 - 234, Edinburgh, Scotland, 1999.

[349] E.-H. Han, G. Karypis, and V. Kumar. Min-Apriori: An Algorithm for Finding Association Rules in Data with Continuous Attributes. http://www.cs.umn.edu/~han, 1997.

[350] J. Han and Y. Fu. Mining Multiple-Level Association Rules in Large Databases. *IEEE Trans. on Knowledge and Data Engineering*, 11(5):798 - 804, 1999.

[351] A. Inokuchi, T. Washio, and H. Motoda. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In *Proc. of the 4th European Conf. of Principles and Practice of Knowledge Discovery in Databases*, pages 13 - 23, Lyon, France, 2000.

[352] M. V. Joshi, G. Karypis, and V. Kumar. A Universal Formulation of Sequential Patterns. In *Proc. of the KDD'2001 workshop on Temporal Data Mining*, San Francisco, CA, August 2001.

[353] M. Kuramochi and G. Karypis. Frequent Subgraph Discovery. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 313 - 320, San Jose, CA, November 2001.

[354] M. Kuramochi and G. Karypis. Discovering Frequent Geometric Subgraphs. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 258 - 265, Maebashi City, Japan, December 2002.

[355] B. Lent, A. Swami, and J. Widom. Clustering Association Rules. In *Proc. of the 13th Intl. Conf. on Data Engineering*, pages 220 - 231, Birmingham, U.K, April 1997.

[356] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, 1(3):259 - 289, November 1997.

[357] R. J. Miller and Y. Yang. Association Rules over Interval Data. In *Proc. of 1997 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 452 - 461, Tucson, AZ, May 1997.

[358] S. Parthasarathy and M. Coatney. Efficient Discovery of Common Substructures in Macromolecules. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 362 - 369, Maebashi City, Japan, December 2002.

[359] J. Pei, J. Han, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. PrefixSpan: Mining Sequential Patterns efficiently by prefix-projected pattern growth. In *Proc of the 17th Intl. Conf. on Data Engineering*, Heidelberg, Germany, April 2001.

[360] A. Savasere, E. Omiecinski, and S. Navathe. Mining for Strong Negative Associations in a Large Database of Customer Transactions. In *Proc. of the 14th Intl. Conf. on Data Engineering*, pages 494 - 502, Orlando, Florida, February 1998.

[361] M. Seno and G. Karypis. SLPMiner: An Algorithm for Finding Frequent Sequential Patterns Using Length-Decreasing Support Constraint. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 418 - 425, Maebashi City, Japan, December 2002.

[362] R. Srikant and R. Agrawal. Mining Generalized Association Rules. In *Proc. of the 21st VLDB Conf.*, pages 407 - 419, Zurich, Switzerland, 1995.

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

[364] R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proc. of the 5th Intl Conf. on Extending Database Technology (EDBT'96)*, pages 18 - 32, Avignon, France, 1996.

[365] P. N. Tan, V. Kumar, and J. Srivastava. Indirect Association: Mining Higher Order Dependencies in Data. In *Proc. of the 4th European Conf. of Principles and Practice of Knowledge Discovery in Databases*, pages 632 - 637, Lyon, France, 2000.

[366] W. G. Teng, M. J. Hsieh, and M.-S. Chen. On the Mining of Substitution Rules for Statistically Dependent Items. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 442 - 449, Maebashi City, Japan, December 2002.

[367] K. Wang, S. H. Tay, and B. Liu. Interestingness-Based Interval Merger for Numeric Association Rules. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 121 - 128, New York, NY, August 1998.

[368] G. I. Webb. Discovering associations with numeric variables. In *Proc. of the 7th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 383 - 388, San Francisco, CA, August 2001.

[369] X.Wu, C. Zhang, and S. Zhang. Mining Both Positive and Negative Association Rules. *ACM Trans. on Information Systems*, 22(3):381 - 405, 2004.

[370] X. Yan and J. Han. gSpan: Graph-based Substructure Pattern Mining. In *Proc. of the 2002 IEEE Intl.*

Conf. on Data Mining, pages 721 - 724, Maebashi City, Japan, December 2002.

[371] M. J. Zaki. Efficiently mining frequent trees in a forest. In *Proc. of the 8th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 71 - 80, Edmonton, Canada, July 2002.

[372] H. Zhang, B. Padmanabhan, and A. Tuzhilin. On the Discovery of Significant Statistical Quantitative Rules. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 374 - 383, Seattle, WA, August 2004.

习　题

1. 考虑表 7-10 所示交通事故数据。

表 7-10　交通事故数据

天气条件	驾驶员状况	交通违章	安全带	损毁程度
好	饮酒	超速	无	较大
坏	清醒	无	有	较小
好	清醒	不遵守停车指示	有	较小
好	清醒	超速	有	较大
坏	清醒	不遵守交通信号	无	较大
好	饮酒	不遵守停车指示	有	较小
坏	饮酒	无	有	较大
好	清醒	不遵守交通信号	有	较大
好	饮酒	无	无	较大
坏	清醒	不遵守交通信号	无	较大
好	饮酒	超速	有	较大
坏	清醒	不遵守停车指示	有	较小

(a) 给出数据集的二元化版本。

(b) 在二元化数据中，每个事务的最大宽度是多少？

(c) 假定支持度阈值是 30%，将产生多少候选和频繁项集？

(d) 创建一个数据集，只包含如下非对称二元属性：（天气条件=坏，驾驶员状况=饮酒，交通违章=是，安全带=无，损毁程度=较大）。对于交通违章，无违章取值 0，其余情况属性值均为 1。假定支持度阈值是 30%，将产生多少候选和频繁项集？

(e) 比较(c)和(d)产生的候选和频繁项集的数量。

2. (a) 考虑表 7-11 所示数据集。假定对数据集的连续属性使用如下离散化策略。

D1：将每个连续属性的值域划分成 3 个等宽的箱。

D2：将每个连续属性的值域划分成 3 个箱，每个箱包含的事务个数相同。

对于每种策略，回答如下问题。

i. 构造数据集的二元化版本。

ii. 导出支持度大于或等于 30%的所有频繁项集。

(b) 连续属性也可以使用聚类方法进行离散化。

i. 为表 7-11 所示的数据点绘制温度与气压图。

ii. 从该图可以看出多少个自然聚类？对图中每个聚类赋予一个标号（C_1、C_2等）。

iii. 你认为可以使用何种类型的聚类算法来识别这些聚类？陈述你的理由。

iv. 使用非对称的二元属性 C_1、C_2 等置换表 7-11 中的温度和气压属性。使用新的属性（连同警报 1、警报 2 和警报 3）构造一个变换矩阵。

v. 从二元化数据导出支持度大于或等于 30%的频繁项集。

表 7-11　习题 2 的数据集

TID	温度	气压	警报 1	警报 2	警报 3
1	95	1 105	0	0	1
2	85	1 040	1	1	0
3	103	1 090	1	1	1
4	97	1 084	1	0	0
5	80	1 038	0	1	1
6	100	1 080	1	1	0
7	83	1 025	1	0	1
8	86	1 030	1	0	0
9	101	1 100	1	1	1

3. 考虑表 7-12 所示数据集。第一个属性是连续的，而其余两个属性是非对称二元的。一个规则是强规则，如果它的支持度超过 15%且置信度超过 60%。表 7-12 给出的数据支持如下两个强规则。

(i) $\{(1 \leqslant A \leqslant 2), B = 1\} \rightarrow \{C = 1\}$

(ii) $\{(5 \leqslant A \leqslant 8), B = 1\} \rightarrow \{C = 1\}$

表 7-12　习题 3 的数据集

A	B	C
1	1	1
2	1	1
3	1	0
4	1	0
5	1	1
6	0	1
7	0	0
8	1	1
9	0	0
10	0	0
11	0	0
12	0	1

(a) 计算这两个规则的支持度和置信度。

(b) 为了使用传统的 *Apriori* 算法找出这些规则，我们需要离散化连续属性 *A*。假定我们使用等宽分箱方法离散化该数据，其中 *bin-width* = 2, 3, 4。对于每个 *bin-width*，上面两个规则是否能够被 *Apriori* 算法发现？（注意，由于属性 *A* 可能具有较宽或较窄的区间，规则不一定与前面的规则完全相同。）对于每个与前面规则对应的规则，计算其支持度和置信度。

(c) 评述使用等宽分箱方法对上述数据集分类的有效性。是否有合适的箱宽度，以便很好地发现上面两个规则？如果没有，可以使用何种其他方法，以确保能够同时发现以上两个规则？

4. 考虑表 7-13 所示数据集。

表 7-13　习题 4 的数据集

年龄(A)	每周上网时数(B)				
	0-5	5-10	10-20	20-30	30-40
10-15	2	3	5	3	2
15-25	2	5	10	10	3
25-35	10	15	5	3	2
35-50	4	6	5	3	2

(a) 对于下面的每组规则，确定具有最高置信度的规则。

i. $15 < A < 25 \rightarrow 10 < B < 20$, $10 < A < 25 \rightarrow 10 < B < 20$ 和 $15 < A < 35 \rightarrow 10 < B < 20$.

ii. $15 < A < 25 \rightarrow 10 < B < 20$, $15 < A < 25 \rightarrow 5 < B < 20$ 和 $15 < A < 25 \rightarrow 5 < B < 30$.

iii. $15 < A < 25 \rightarrow 10 < B < 20$ 和 $10 < A < 35 \rightarrow 5 < B < 30$.

(b) 假定我们希望找出年龄在 15 岁到 25 岁之间的因特网用户每周的平均上网小时数。写一个基于统计学的关联规则，来刻画这个年龄段的用户。为了计算平均上网小时数，用中点近似值来表示每个区间（例如，使用 $B = 7.5$ 来表示区间 $5 < B < 10$）。

(c) 通过将(b)中的平均上网小时数与不属于该年龄段的其他用户的平均上网小时数进行比较，检查(b)的量化关联规则是否具有统计意义。

5. 对于具有下面给出的属性的数据集，描述如何将它转换成适合于关联分析的二元事务数据集。具体地，指出原数据集中的每个属性

(a) 对应于事务数据集中多少个二元属性；

(b) 原属性的值如何映射到二元属性的值；

(c) 数据属性值中是否有分层结构可以用来将数据分组，形成少量二元属性。

下面是该数据集的属性列表以及它们的可能值。假定所有的属性都在每个学生上收集。

- **年级**：一年级、二年级、三年级、四年级、硕士研究生、博士研究生、专业人员。
- **邮政编码**：美国学生的家庭邮政编码，非美国学生的住处邮政编码。
- **院**：农学、建筑学、继续教育、教育、文学、工程、自然科学、商学、法律、医学、牙科、药学、护理学、兽医学。
- **住校**：如果学生住校为 1，否则为 0。
- 以下每个项是一个属性，如果学生说对应的语言，则取 1，否则取 0。
  - 阿拉伯语
  - 孟加拉语
  - 汉语
  - 英语
  - 葡萄牙语
  - 俄语
  - 西班牙语

6. 考虑表 7-14 所示数据集。假定我们对提取如下形式的关联规则感兴趣：

$$\{\alpha_1 \leqslant \text{年龄} \leqslant \alpha_2，\ \text{弹钢琴}=\text{是}\} \rightarrow \{\text{喜欢古典音乐}=\text{是}\}$$

表 7-14　习题 6 的数据集

年龄	弹钢琴	喜欢古典音乐
9	是	是
11	是	是
14	是	否
17	是	否
19	是	是
21	否	否
25	否	否
29	是	是
33	否	否
39	否	是
41	否	否
47	否	是

为了处理连续属性，我们使用等频方法，区间个数为 3、4 和 6。分类属性通过引进与分类值个数一样多的新的非对称二元属性来处理。假定支持度阈值是 10%，置信度阈值是 70%。

(a) 假定我们将年龄属性离散化成3个等频区间。找出满足最小支持度和最小置信度的α_1和α_2。

(b) 将年龄属性离散化成 4 个等频区间，重复(a)。将得到的规则与(a)得到的规则进行比较。

(c) 将年龄属性离散化成 6 个等频区间，重复(a)。将得到的规则与(a)得到的规则进行比较。

(d) 由(a)、(b)和(c)的结果，讨论离散化区间的选择对关联规则挖掘算法所提取的规则的影响。

7. 考虑表 7-15 所示的事务，其中商品分类由图 7-25 给出。

表 7-15　购物篮事务的例子

事务 ID	购买的商品
1	薄片食物，饼干，普通碳酸饮料，火腿
2	薄片食物，火腿，鸡肉，节食碳酸饮料
3	火腿，熏肉，整鸡，普通碳酸饮料
4	薄片食物，火腿，鸡肉，节食碳酸饮料
5	薄片食物，熏肉，鸡肉
6	薄片食物，火腿，熏肉，整鸡，普通碳酸饮料
7	薄片食物，饼干，鸡肉，节食碳酸饮料

(a) 挖掘带产品分类的关联规则的主要挑战是什么？

(b) 考虑下面的方法：每个事务 t 用扩展的事务 t'替换，t'包含 t 中所有商品和它们的祖先。例如，事务 t = {薄片食物，饼干}用 t' = {薄片食物，饼干，点心，食品}替换。使用该方法导出所有支持度大于或等于 70%的频繁项集（长度不超过 4）。

(c) 考虑另一种方法，其中频繁项集逐层产生。开始，产生分层结构顶层的所有频繁项集。然后，使用较高层发现的频繁项集，产生涉及较低层中项的候选项集。例如，仅当{点心，碳酸饮料}频繁时，才产生候选项集{薄片食物，节食碳酸饮料}。使用该方法导出

所有支持度大于或等于70%的频繁项集（长度不超过4）。

(d) 比较(b)和(c)找出的频繁项集。评述算法的有效性和完全性。

8. 下面的问题考察关联规则的支持度和置信度，它们可以因概念分层而变化。

(a) 考虑给定概念分层中的项 x。令 $\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_k$ 表示概念分层中 x 的 k 个子女。证明 $s(x) \leqslant \sum_{i=1}^{k} s(\bar{x}_i)$，其中 $s(\cdot)$是项的支持度。在什么条件下，不等式取等号？

(b) 设 p 和 q 是一对项，而 $\hat{p}$ 和 $\hat{q}$ 是它们在概念分层中的对应父母。如果 $s(\{p, q\}) > minsup$，下面哪些项集肯定是频繁的？（i）$s(\{\hat{p}, q\})$，（ii）$s(\{p, \hat{q}\})$，（iii）$s(\{\hat{p}, \hat{q}\})$。

(c) 考虑关联规则 $\{p\} \rightarrow \{q\}$。假定规则的置信度超过 *minconf*。下面哪些规则的置信度肯定高于 *minconf* ？（i）$\{p\} \rightarrow \{\hat{q}\}$，（ii）$\{\hat{p}\} \rightarrow \{q\}$，（iii）$\{\hat{p}\} \rightarrow \{\hat{q}\}$。

9. (a) 假定没有时限约束，列举包含在下面数据序列中的所有4-子序列：

$$<\{1, 3\}\ \{2\}\ \{2, 3\}\ \{4\}>$$

(b) 假定未施加任何时限约束，列举包含在(a)的数据序列中的所有3-子序列。

(c) 列举包含在(a)的数据序列中的所有4-子序列（假定时限约束是灵活的）。

(d) 列举包含在(a)的数据序列中的所有3-子序列（假定时限约束是灵活的）。

10. 给定表 7-16 所示的序列数据库，找出支持度大于等于 50%的所有频繁子序列。假定序列上没有施加时限约束。

表 7-16　各种传感器产生的事件序列的例子

传感器	时间戳	事件
S1	1	A, B
	2	C
	3	D, E
	4	C
S2	1	A, B
	2	C, D
	3	E
S3	1	B
	2	A
	3	B
	4	D, E
S4	1	C
	2	D, E
	3	C
	4	E
S5	1	B
	2	A
	3	B, C
	4	A, D

11. (a) 对于下面给定的每个序列 $w = <e_1 e_2 \cdots e_i e_{i+1} \cdots e_{last}>$，确定它们是否是序列<{1, 2, 3} {2, 4} {2, 4, 5} {3, 5} {6}>的子序列，时限约束为：

mingap = 0　（e_i 中最后一个事件和 e_{i+1} 中第一个事件之间的间隔大于 0）

maxgap =3　（e_i 中第一个事件和 e_{i+1} 中最后一个事件之间的间隔小于等于 3）

maxspan = 5　（e_1 中第一个事件和 e_{last} 中最后一个事件之间的间隔小于等于 5）

ws = 1　（e_i 中第一个事件和最后一个事件之间的间隔小于等于 1）

- $w = <\{1\}\ \{2\}\ \{3\}>$
- $w = <\{1, 2, 3, 4\}\ \{5, 6\}>$
- $w = <\{2, 4\}\ \{2, 4\}\ \{6\}>$
- $w = <\{1\}\ \{2, 4\}\ \{6\}>$
- $w = <\{1, 2\}\ \{3, 4\}\ \{5, 6\}>$

(b) 确定上面每个子序列 w 是否是下面序列 s 的邻接子序列。

- $s = \langle\{1, 2, 3, 4, 5, 6\}\ \{1, 2, 3, 4, 5, 6\}\ \{1, 2, 3, 4, 5, 6\}\rangle$
- $s = \langle\{1, 2, 3, 4\}\ \{1, 2, 3, 4, 5, 6\}\ \{3, 4, 5, 6\}\rangle$
- $s = \langle\{1, 2\}\ \{1, 2, 3, 4\}\ \{3, 4, 5, 6\}\ \{5, 6\}\rangle$
- $s = \langle\{1, 2, 3\}\ \{2, 3, 4, 5\}\ \{4, 5, 6\}\rangle$

12. 对于下面给定的每个序列 $w = <e_1 \cdots e_{last}>$，确定它们是否是数据序列$\langle\{A, B\}\ \{C, D\}\ \{A, B\}\ \{C, D\}\ \{A, B\}\ \{C, D\}\rangle$的子序列，时限约束为：

$mingap = 0$　（e_i 中最后一个事件和 e_{i+1} 中第一个事件之间的间隔大于 0）

$maxgap = 2$　（e_i 中第一个事件和 e_{i+1} 中最后一个事件之间的间隔小于等于 2）

$maxspan = 6$　（e_1 中第一个事件和 e_{last} 中最后一个事件之间的间隔小于等于 6）

$ws = 1$　（e_i 中第一个事件和最后一个事件之间的间隔小于等于 1）

(a) $w = \langle\{A\}\ \{B\}\ \{C\}\ \{D\}\rangle$

(b) $w = \langle\{A\}\ \{B, C, D\}\ \{A\}\rangle$

(c) $w = \langle\{A\}\ \{A, B, C, D\}\ \{A\}\rangle$

(d) $w = \langle\{B, C\}\ \{A, D\}\ \{B, C\}\rangle$

(e) $w = \langle\{A, B, C, D\}\ \{A, B, C, D\}\rangle$

13. 考虑下面各频繁 3-序列：

$\langle\{1, 2, 3\}\rangle$、$\langle\{1, 2\}\ \{3\}\rangle$、$\langle\{1\}\ \{2, 3\}\rangle$、$\langle\{1, 2\}\ \{4\}\rangle$、$\langle\{1, 3\}\ \{4\}\rangle$、$\langle\{1, 2, 4\}\rangle$、$\langle\{2, 3\}\ \{3\}\rangle$、$\langle\{2, 3\}\ \{4\}\rangle$、$\langle\{2\}\ \{3\}\ \{3\}\rangle$和$\langle\{2\}\ \{3\}\ \{4\}\rangle$。

(a) 列出 GSP 算法的候选产生步骤产生的所有候选 4-序列。

(b) 列出 GSP 算法的候选剪枝步骤剪掉的所有候选 4-序列（假定没有时限约束）。

(c) 列出 GSP 算法的候选剪枝步骤剪掉的所有候选 4-序列（假定 $maxgap = 1$）。

14. 考虑表 7-17 所示的给定对象的数据序列。根据如下计数方法，对序列$\langle\{p\}\ \{q\}\ \{r\}\rangle$的出现次数计数。

表 7-17　习题 14 事件序列数据的例子

时间戳	事件
1	p, q
2	r
3	s
4	p, q
5	r, s
6	p
7	q, r
8	q, s
9	p
10	q, r, s

(a) COBJ（每个对象出现一次）。

(b) CWIN（每个滑动窗口出现一次）。

(c) CMINWIN（最小出现窗口数）。

(d) CDIST_O（允许事件-时间戳重叠的不同出现）。

(e) CDIST（不允许事件-时间戳重叠的不同出现）。

15. 为了使频繁子图挖掘算法能够处理如下类型的图，讨论算法应做的必要修改。

(a) 有向图。

(b) 无标号图。

(c) 无环图。

(d) 非连通图。

上面给定的图类型影响算法的哪些步骤（候选产生、候选剪枝和支持度计数），是否有进一步的优化有助于提高算法的性能。

16. 画出连接图 7-28 中的图对得到的所有候选子图。假定使用边增长算法扩展子图。

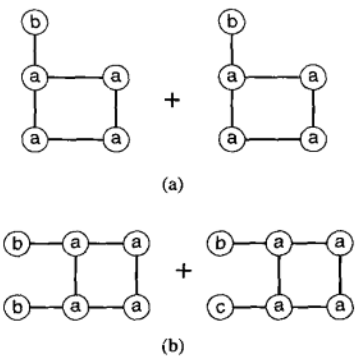


图 7-28　习题 16 的图

17. 画出连接图 7-29 中的图对得到的所有候选子图。假定使用边增长算法扩展子图。

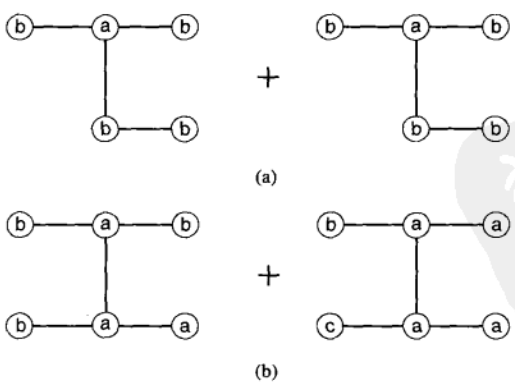


图 7-29　习题 17 的图

18. (a) 如果用归纳子图联系定义支持度，证明如果允许 g_1 和 g_2 具有重叠的顶点集，则规

则 $g_1 \rightarrow g_2$ 的置信度可能大于 1。

(b) 确定具有 $|V|$ 个顶点的图的规范标号的时间复杂性是多少？

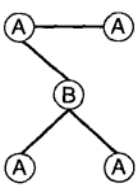
(c) 子图的核可能是多重自同构的。这可能增加合并两个具有相同核的频繁子图后得到的候选子图的个数。确定由于 k 个顶点的核的自同构得到的候选子图的最大个数。

(d) 两个大小为 k 的频繁子图可能共享多个核。确定被两个频繁子图共享的核的最大个数。

19. 考虑一个图挖掘算法，它使用边增长方法合并如下所示两个无向、无权子图。

(a) 绘制合并两个子图时得到的不同的核。

(b) 使用下面的核可以产生多少候选？



20. 原来的关联规则挖掘框架只考虑项在事务中同时出现。有时候非频繁的项集可能也是富含信息的。例如，项集{TV，DVD，¬VCR}暗示，许多购买电视和 DVD 的顾客不购买 VCR。本题要求将关联规则框架扩充到负项集（即包含项的出现和不出现）。我们使用符号“¬”表示缺失项。

(a) 一种导出负项集的朴素方法是扩充每个事务，使之包含缺失项，如表 7-18 所示。

表 7-18　数值数据集的例子

TID	TV	¬TV	DVD	¬DVD	VCR	¬VCR	···
1	1	0	0	1	0	1	···
2	1	0	0	1	0	1	···

i. 假定事务数据库包含 1 000 个不同的项。由这些项可能产生的正项集的总数是多少？（注意：正项集不包含任何负项。）

ii. 由这些事务产生的频繁项集的最大个数是多少？（假定频繁项集可以包含正项、负项或二者。）

iii. 解释为什么这种用负项扩充每个事务的朴素方法对于导出负项集不切实际。

(b) 考虑表 7-15 所示的数据库。如下涉及普通和节食碳酸饮料的负关联规则的支持度和置信度是多少？

i. ¬普通碳酸饮料 → 节食碳酸饮料。

ii. 普通碳酸饮料 → ¬节食碳酸饮料。

iii. ¬节食碳酸饮料 → 普通碳酸饮料。

iv. 节食碳酸饮料 → ¬普通碳酸饮料。

21. 假定我们想从包含 d 个项的数据集中提取正项集和负项集。

(a) 考虑一种方法，引进一个新的变量来表示每个负项。使用这种方法，项的个数从 d 增加到 $2d$。假定项集可以包含同一个变量的正项和负项，项集格的大小是多少？

(b) 假定项集必须包含不同变量的正项和负项。例如，项集$\{a, \bar{a}, b, \bar{c}\}$是不合法的，因为它同时包含了变量 a 的正项和负项。项集格的大小是多少？

22. 对于下面定义的每种类型的模式，确定支持度度量随着项集大小的增长，是否是单调的、反单调的或非单调的（即既不单调，也不反单调）。

(a) 包含正项和负项的项集，如$\{a, b, \bar{c}, \bar{d}\}$。用于这样的模式，支持度度量是否是单调的、反单调的或非单调的？

(b) 诸如$\{(a \vee b \vee c), d, e\}$的布尔逻辑模式，可能包含项的析取与合取。用于这样的模式，支持度度量是否是单调的、反单调的或非单调的？

23. 许多关联分析算法依赖类 *Apriori* 方法找出频繁模式。算法的总体结构如下。

算法 7.3　类 *Apriori* 算法

```
1: k = 1
2: F_k = {i | i ∈ I ∧ σ({i})/N ≥ minsup}      {找出频繁 1-模式}
3: repeat
4:    k = k + 1
5:    C_k = genCandidate(F_{k-1})              {候选产生}
6:    C_k = pruneCandidate(C_k, F_{k-1})       {候选剪枝}
7:    C_k = count(C_k, D)                      {支持度计数}
8:    F_k = {c | c ∈ C_k ∧ σ(c)/N ≥ minsup}    {提取频繁模式}
9: until F_k = ∅
10: Answer = ∪ F_k
```

假定我们对发现形如$\{a \vee b\} \rightarrow \{c, d\}$的布尔逻辑规则感兴趣。其中，规则可能涉及项的析取与合取。对应的项集可以写成$\{(a \vee b), c, d\}$。

(a) 对于这样的项集，先验原理是否依然成立？

(b) 如何修改候选产生步骤，以便发现这样的模式？

(c) 如何修改候选剪枝步骤，以便发现这样的模式？

(d) 如何修改支持度计数步骤，以便发现这样的模式？

第8章

聚类分析：基本概念和算法

聚类分析将数据划分成有意义或有用的组（簇）。如果目标是划分成有意义的组，则簇应当捕获数据的自然结构。然而，在某种意义下，聚类分析只是解决其他问题（如数据汇总）的起点。无论是旨在理解还是实用，聚类分析都在广泛的领域扮演着重要角色。这些领域包括：心理学和其他社会科学、生物学、统计学、模式识别、信息检索、机器学习和数据挖掘。

聚类分析在许多实际问题上都有应用。我们给出一些具体的例子，按聚类目的是为了理解还是实用来组织。

旨在理解的聚类 在对世界的分析和描述中，类，或在概念上有意义的具有公共特性的对象组，扮演着重要的角色。的确，人类擅长将对象划分成组（聚类），并将特定的对象指派到这些组（分类）。例如，即使很小的孩子也能很快地将图片上的对象标记为建筑物、车辆、人、动物、植物等。就理解数据而言，簇是潜在的类，而聚类分析是研究自动发现这些类的技术。下面是一些例子。

- **生物学**。生物学家花了许多年来创建所有生物体的系统分类学（层次结构的分类）：界（kingdom）、门（phylum）、纲（class）、目（order）、科（family）、属（genus）和种（species）。这样，或许并不奇怪，聚类分析早期的大部分工作都是在寻求创建可以自动发现分类结构的数学分类方法。最近，生物学家使用聚类分析大量的遗传信息。例如，聚类已经用来发现具有类似功能的基因组。
- **信息检索**。万维网包含数以亿计的 Web 页面，网络搜索引擎可能返回数以千计的页面。可以使用聚类将搜索结果分成若干簇，每个簇捕获查询的某个特定方面。例如，查询“电影”返回的网页可以分成诸如评论、电影预告片、影星和电影院等类别。每一个类别（簇）又可以划分成若干子类别（子簇），从而产生一个层次结构，帮助用户进一步探索查询结果。
- **气候**。理解地球气候需要发现大气层和海洋的模式。聚类分析已经用来发现对陆地气候具有显著影响的极地和海洋大气压力模式。
- **心理学和医学**。一种疾病或健康状况通常有多种变种，聚类分析可以用来发现这些子类别。例如，聚类已经用来识别不同类型的抑郁症。聚类分析也可以用来检测疾病的时间和空间分布模式。
- **商业**。商业点收集当前和潜在顾客的大量信息。可以使用聚类将顾客划分成若干组，以便进一步分析和开展营销活动。

旨在实用的聚类 聚类分析提供由个别数据对象到数据对象所指派的簇的抽象。此外，一些聚类技术使用簇原型（即代表簇中其他对象的数据对象）来刻画簇特征。这些簇原型可以用作大

量数据分析和数据处理技术的基础。因此，就实用而言，聚类分析是研究发现最有代表性的簇原型的技术。

- **汇总**。许多数据分析技术，如回归和 PCA，都具有 $O(m^2)$或更高的时间或空间复杂度（其中 m 是对象的个数）。因此，对于大型数据集，这些技术不切实际。然而，可以将算法用于仅包含簇原型的数据集，而不是整个数据集。依赖分析类型、原型个数和原型代表数据的精度，汇总结果可以与使用所有数据得到的结果相媲美。
- **压缩**。簇原型可以用于数据压缩。例如，创建一个包含所有簇原型的表，即每个原型赋予一个整数值，作为它在表中的位置（索引）。每个对象用与它所在的簇相关联的原型的索引表示。这类压缩称作**向量量化**（vector quantization），并常常用于图像、声音和视频数据，此类数据的特点是：(1)许多数据对象之间高度相似，(2)某些信息丢失是可以接受的，(3)希望大幅度压缩数据量。
- **有效地发现最近邻**。找出最近邻可能需要计算所有点对点之间的距离。通常，可以更有效地发现簇和簇原型。如果对象相对地靠近簇的原型，则我们可以使用簇原型减少发现对象最近邻所需要计算的距离的数目。直观地说，如果两个簇原型相距很远，则对应簇中的对象不可能互为近邻。这样，为了找出一个对象的最近邻，只需要计算到邻近簇中对象的距离，其中两个簇的邻近性用其原型之间的距离度量。这种思想的更详细描述见第 2 章习题 25。

本章是聚类分析导论。我们从概述聚类分析开始，包括将对象划分成簇的集合的各种方法和聚类的不同类型的讨论。然后介绍三种专门的聚类技术：K 均值、凝聚的层次聚类和 DBSCAN。它们代表一大类算法，并用于解释各种概念。本章的最后一节专门讨论簇的有效性——评估聚类算法产生的簇的方法。更高级的聚类概念和算法将在第 9 章讨论。我们尽可能地讨论不同方案的优点和缺点。此外，文献注释提供更深入考察聚类分析的相关书籍和文章。

8.1　概述

在讨论具体的聚类技术之前，我们先提供必要的背景知识。首先，我们进一步定义聚类分析，解释它的困难所在，并阐述它与其他数据分组技术之间的关系。然后，考察两个重要问题：(1)将数据对象集划分成簇集合的不同方法，(2)簇的类型。

8.1.1　什么是聚类分析

聚类分析仅根据在数据中发现的描述对象及其关系的信息，将数据对象分组。其目标是，组内的对象相互之间是相似的（相关的），而不同组中的对象是不同的（不相关的）。组内的相似性（同质性）越大，组间差别越大，聚类就越好。

在许多应用中，簇的概念都没有很好地加以定义。为了理解确定簇构造的困难性，考虑图 8-1。该图显示了 20 个点和将它们划分成簇的 3 种不同方法。标记的形状指示簇的隶属关系。图 8-1b 和图 8-1d 分别将数据划分成两部分和六部分。然而，将 2 个较大的簇都划分成 3 个子簇可能是人的视觉系统造成的假象。此外，说这些点形成 4 个簇（如图 8-1c 所示）可能也不无道理。该图表明簇的定义是不精确的，而最好的定义依赖于数据的特性和期望的结果。

聚类分析与其他将数据对象分组的技术相关。例如，聚类可以看作一种分类，它用类（簇）标号创建对象的标记。然而，只能从数据导出这些标号。相比之下，第4章的分类是**监督分类**

(supervised classification)，即使用由类标号已知的对象开发的模型，对新的、无标记的对象赋予类标号。为此，有时称聚类分析为**非监督分类**（unsupervised classification）。在数据挖掘中，不附加任何条件使用术语分类时，通常是指监督分类。

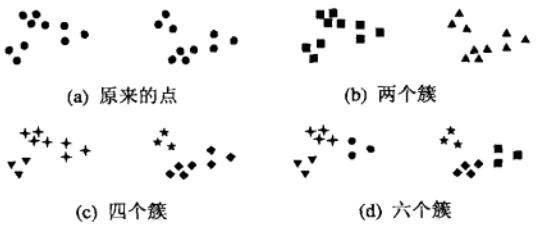


图 8-1 相同点集的不同聚类方法

此外，尽管术语**分割**（segmentation）和**划分**（partitioning）有时也用作聚类的同义词，但是这些术语通常用来表示传统的聚类分析之外的方法。例如，术语划分通常用在与将图分成子图相关的技术，与聚类并无太大联系。分割通常指使用简单的技术将数据分组；例如，图像可以根据像素亮度或颜色分割，人可以根据他们的收入分组。尽管如此，图划分、图像分割和市场分割的许多工作都与聚类分析有关。

8.1.2 不同的聚类类型

整个簇集合通常称作聚类，本节我们将区分不同类型的聚类：层次的（嵌套的）与划分的（非嵌套的），互斥的、重叠的与模糊的，完全的与部分的。

层次的与划分的 不同类型的聚类之间最常讨论的差别是：簇的集合是嵌套的，还是非嵌套的；或者用更传统的术语，是层次的还是划分的。**划分聚类**（partitional clustering）简单地将数据对象集划分成不重叠的子集（簇），使得每个数据对象恰在一个子集中。例如，图 8-1（b～d）中每个簇集都是一个划分聚类。

如果允许簇具有子簇，则我们得到一个**层次聚类**（hierarchical clustering）。层次聚类是嵌套簇的集族，组织成一棵树。除叶结点外，树中每一个结点（簇）都是其子女（子簇）的并，而树根是包含所有对象的簇。通常（但并非总是），树叶是单个数据对象的单元素簇。如果允许簇嵌套，则图 8-1a 的一种解释是：它有两个子簇（图 8-1b），其中每个子簇又各自具有 3 个子簇（图 8-1d）。图 8-1（a～d）中显示的簇也依次形成一个层次聚类，每层分别具有 1，2，4 和 6 个簇。最后，层次聚类可以看作划分聚类的序列，划分聚类可以通过取序列的任意成员得到，即通过在一个特定层剪断层次树得到。

互斥的、重叠的与模糊的 图8-1显示的簇都是**互斥的**（exclusive），因为每个对象都指派到单个簇。在有些情况下，可以合理地将一个点放到多个簇中，这种情况可以被非互斥聚类更好地处理。在最一般的意义下，**重叠的**（overlapping）或**非互斥的**（non-exclusive）聚类用来反映一个对象同时属于多个组（类）这一事实。例如，在大学里，一个人可能既是学生，又是雇员。当对象在两个或多个簇“之间”，并且可以合理地指派到这些簇中的任何一个时，也常常可以使用非互斥聚类。想象一个点在图8-1的两个簇中间。将它放到所有“同样好”的簇中，而不是任意地将它指派到单个簇中。

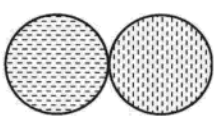
在**模糊聚类**（fuzzy clustering）中，每个对象以一个 0（绝对不属于）和 1（绝对属于）之间的隶属权值属于每个簇。换言之，簇被视为模糊集。（从数学上讲，在模糊集中，每个对象以 0 和 1 之间的权值属于任何一个集合。在模糊聚类中，通常施加一个约束条件：每个对象的权值之和必须等于 1。）同理，概率聚类技术计算每个点属于每个簇的概率，并且这些概率的和必须等于 1。由于任何对象的隶属权值或概率之和等于 1，因此模糊和概率聚类并不能真正地解决一个对象属于多个类的多类问题，如学生雇员。这些方法最适合如下情况：当对象接近多个簇时，避免将对象随意地指派到一个簇。实践中，通常通过将对象指派到具有最高隶属权值或概率的簇，将模糊或概率聚类转换成互斥聚类。

完全的与部分的　**完全聚类**（complete clustering）将每个对象指派到一个簇，而**部分聚类**（partial clustering）不是这样。促进部分聚类的因素是，数据集中某些对象可能不属于明确定义的组。数据集中的一些对象可能代表噪声、离群点或“不感兴趣的背景”。例如，一些报刊报导可能涉及公共主题，如全球变暖，而其他报导更一般或一类一事。这样，为了发现上月报导的最重要的主题，我们可能希望只搜索与公共主题紧密相关的文档簇。在其他情况下，需要对象的完全聚类。例如，使用聚类组织用于浏览的文档的应用，必需保证能够浏览所有的文档。

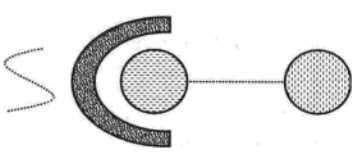
8.1.3　不同的簇类型

聚类旨在发现有用的对象组（簇），这里有用性由数据挖掘目标定义。毫无疑问，有许多不同的簇概念，实践证明都是有用的。为了以可视方式说明这些簇类型之间的差别，我们使用二维数据点（见图 8-2）作为我们的数据对象。然而，我们强调的是，这里介绍的簇类型同样适用于其他数据。

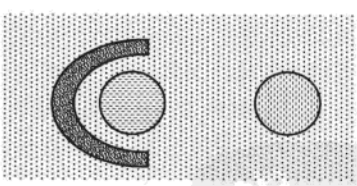
(a) 明显分离的簇。每个点到同簇中任意点的距离比到不同簇中所有点的距离更近。



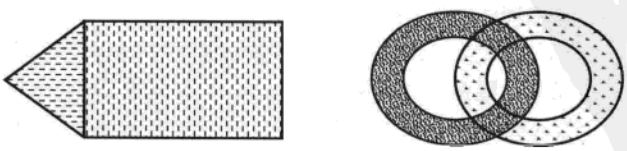
(b) 基于中心的簇。每个点到其簇中心的距离比到任何其他簇中心的距离更近。



(c) 基于邻近的簇。每个点到该簇中至少一个点的距离比到不同簇中任意点的距离更近。



(d) 基于密度的簇。簇是被低密度区域分开的高密度区域。



(e) 概念簇。簇中的点具有由整个点集导出的某种一般共同性质。（在两个环相交处的点属于两个环。）

图 8-2　用二维点集图示的不同簇类型

明显分离的 簇是对象的集合，其中每个对象到同簇中每个对象的距离比到不同簇中任意对象的距离都近（或更加相似）。有时，使用一个阈值来说明簇中所有对象相互之间必须充分接近（或相似）。仅当数据包含相互远离的自然簇时，簇的这种理想定义才能满足。图 8-2a 给出一个由二维空间的两组点组成的明显分离的簇的例子。不同组中的任意两点之间的距离都大于组内任意两点之间的距离。明显分离的簇不必是球形的，可以具有任意形状。

基于原型的 簇是对象的集合，其中每个对象到定义该簇的原型的距离比到其他簇的原型的距离更近（或更加相似）。对于具有连续属性的数据，簇的原型通常是质心，即簇中所有点的平均值。当质心没有意义时（例如当数据具有分类属性时），原型通常是中心点，即簇中最有代表性的点。对于许多数据类型，原型可以视为最靠近中心的点；在这种情况下，通常把基于原型的簇看作**基于中心的簇**（center-based cluster）。毫无疑问，这种簇趋向于呈球状。图 8-2b 给出一个基于中心的簇的例子。

基于图的 如果数据用图表示，其中结点是对象，而边代表对象之间的联系（见2.1.2节），则簇可以定义为**连通分支**（connected component），即互相连通但不与组外对象连通的对象组。基于图的簇的一个重要例子是**基于邻近的簇**（contiguity-based cluster），其中两个对象是相连的，仅当它们的距离在指定的范围之内。也就是说在基于邻近的簇中，每个对象到该簇某个对象的距离比到不同簇中任意点的距离更近。图8-2c对二维点给出这种簇的一个例子。当簇不规则或缠绕时，簇的这种定义是有用的。但是，当数据具有噪声时就可能出现问题，因为如图8-2c的两个球形簇所示，一个小的点桥就可能合并两个不同的簇。

也存在其他类型的基于图的簇。一种方法（见 8.3.2 节）是定义簇为**团**（clique），即图中相互之间完全连接的结点的集合。具体地说，如果我们按照对象之间的距离添加连接，当对象集形成团时就形成一个簇。与基于原型的簇一样，这样的簇也趋向于呈球形。

基于密度的 簇是对象的稠密区域，被低密度的区域环绕。图 8-2d 显示某些基于密度的簇，该图的数据是通过对图 8-2c 的数据添加噪声创建的。两个圆形簇没有合并，因为它们之间的桥消失在噪声中。图 8-2c 中的曲线也消失在噪声中，在图 8-2d 中并未形成簇。当簇不规则或互相盘绕，并且有噪声和离群点时，常常使用基于密度的簇定义。相比之下，对于图 8-2d 的数据，簇的基于邻近的定义就行不通，因为噪声将形成簇间的桥。

共同性质的（概念簇） 通常，我们可以把簇定义为有某种共同性质的对象的集合。这个定义包括前面的所有簇定义。例如，基于中心的簇中的对象都具有共同的性质：它们都离相同的质心或中心点最近。然而，共享性质的方法还包含新的簇类型。考虑图 8-2e 所示的簇。三角形区域（簇）邻近于矩形区域（簇），并且存在两个缠绕的环（簇）。在这两种情况下，聚类算法都需要非常具体的簇概念来成功地检测出这些簇。发现这样的簇的过程称作概念聚类。然而，过于复杂的簇概念将涉及模式识别领域。因此，本书只考虑较简单的簇类型。

线路图

本章我们使用如下三种简单但重要的技术来介绍聚类分析涉及的一些概念。

- **K 均值**。K 均值是基于原型的、划分的聚类技术。它试图发现用户指定个数（K）的簇（由质心代表）。
- **凝聚的层次聚类**。这种聚类方法涉及一组密切相关的聚类技术，它们通过如下步骤产生

层次聚类：开始，每个点作为一个单点簇；然后，重复地合并两个最靠近的簇，直到产生单个的、包含所有点的簇。其中某些技术可以用基于图的聚类解释，而另一些可以用基于原型的方法解释。

- DBSCAN。这是一种产生划分聚类的基于密度的聚类算法，簇的个数由算法自动地确定。低密度区域中的点被视为噪声而忽略，因此 DBSCAN 不产生完全聚类。

8.2　K 均值

基于原型的聚类技术创建数据对象的单层划分。存在许多这样的技术，但是两个最突出的是 K 均值和 K 中心点。K 均值用质心定义原型，其中质心是一组点的均值。通常，K 均值聚类用于 n 维连续空间中的对象。K 中心点使用中心点定义原型，其中中心点是一组点中最有代表性的点。K 中心点聚类可以用于广泛的数据，因为它只需要对象之间的邻近性度量。尽管质心几乎从来不对应于实际的数据点，但是根据定义，中心点必须是一个实际数据点。本节，我们只关注 K 均值，一种最老的、最广泛使用的聚类算法。

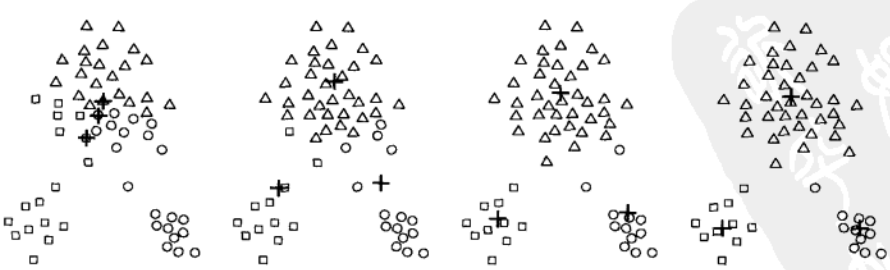
8.2.1　基本 K 均值算法

K 均值算法比较简单，我们从介绍它的基本算法开始。首先，选择 K 个初始质心，其中 K 是用户指定的参数，即所期望的簇的个数。每个点指派到最近的质心，而指派到一个质心的点集为一个簇。然后，根据指派到簇的点，更新每个簇的质心。重复指派和更新步骤，直到簇不发生变化，或等价地，直到质心不发生变化。

K 均值的形式描述参见算法 8.1。K 均值的操作解释参见图 8-3。该图显示如何从 3 个质心出发，通过 4 次指派和更新，找出最后的簇。在这些和其他显示 K 均值聚类的图中，每个子图显示(1)迭代开始时的质心，(2)点到质心的指派。质心用符号“+”指示；属于同一个簇的所有点具有相同形状的标记。

算法 8.1　基本 K 均值算法

1：选择 K 个点作为初始质心。
2：**repeat**
3：　将每个点指派到最近的质心，形成 K 个簇。
4：　重新计算每个簇的质心。
5：**until** 质心不发生变化。



(a) 迭代 1　(b) 迭代 2　(c) 迭代 3　(d) 迭代 4

图 8-3　使用 K 均值算法找出样本数据中的三个簇

在图 8-3a 所示的第一步，将点指派到初始质心。这些质心都在点的较大组群中。对于这个例子，我们用均值作为质心。把点指派到质心后，更新质心。每一步的图都显示该步开始时的质心和点到质心的指派。在第二步，点指派到更新后的质心，并且再次更新质心。在步骤 2、3 和 4（分别在图 8-3b、图 8-3c 和图 8-3d 中显示）中，两个质心移向图下部点的两个较小的组群。当 K 均值算法终止于图 8-3d 时（因为不再发生变化），质心标识出点的自然分组。

对于邻近性函数和质心类型的某些组合，K 均值总是收敛到一个解，即 K 均值到达一种状态，其中所有点都不会从一个簇转移到另一个，因此质心不再改变。然而，由于大部分收敛都发生在早期阶段，因此通常用较弱的条件替换算法 8.1 的第 5 行。例如，用“直到仅有 1%的点改变簇”。

我们将更详细地考虑基本 K 均值算法的每个步骤，并分析算法的时间和空间复杂度。

1. 指派点到最近的质心

为了将点指派到最近的质心，我们需要邻近性度量来量化所考虑的数据的“最近”概念。通常，对欧氏空间中的点使用欧几里得距离（L_2），对文档用余弦相似性。然而，对于给定的数据类型，可能存在多种适合的邻近性度量。例如，曼哈顿距离（L_1）可以用于欧几里得数据，而 Jaccard 度量常常用于文档。

通常，K 均值使用的相似性度量相对简单，因为算法要重复地计算每个点与每个质心的相似度。然而，在某些情况下，如数据在低维欧几里得空间时，许多相似度的计算都有可能避免，因此显著地加快了 K 均值算法的速度。二分 K 均值（见 8.2.3 节）是另一种通过减少相似度计算量来加快 K 均值速度的方法。

2. 质心和目标函数

K 均值算法第 4 步一般可以陈述为“重新计算每个簇的质心”，因为质心可能随数据邻近性度量和聚类目标不同而改变。聚类的目标通常用一个目标函数表示，该函数依赖于点之间，或点到簇的质心的邻近性；如，最小化每个点到最近质心的距离的平方。我们用两个例子解释这一点。然而，关键点是：一旦我们选定了邻近性度量和目标函数，则应当选择的质心可以从数学上确定。我们将在 8.2.6 节给出数学推导的细节，这里只提供非数学的讨论。

欧几里得空间中的数据 考虑邻近性度量为欧几里得距离的数据。我们使用**误差的平方和**（Sum of the Squared Error，SSE）作为度量聚类质量的目标函数。SSE 也称散布（scatter）。换言之，我们计算每个数据点的误差，即它到最近质心的欧几里得距离，然后计算误差的平方和。给定由两次运行 K 均值产生的两个不同的簇集，我们更喜欢误差的平方和最小的那个，因为这说明聚类的原型（质心）可以更好地代表簇中点。使用表 8-1 中的记号，SSE 形式地定义如下：

$$\mathrm{SSE}=\sum_{i=1}^{K}\sum_{x\in C_i} dist(c_i,x)^2 \tag{8-1}$$

其中，*dist* 是欧几里得空间中两个对象之间的标准欧几里得距离（L_2）。

给定这些假设，可以证明（见 8.2.6 节）：使簇的 SSE 最小的质心是均值。使用表 8-1 中的记号，第 i 个簇的质心（均值）由公式（8-2）定义。

$$\mathbf{c}_i=\frac{1}{m_i}\sum_{x\in C_i}\mathbf{x} \tag{8-2}$$

例如，3 个二维点(1, 1)、(2, 3) 和 (6, 2) 的质心是 ((1 + 2 +6) / 3, (1 + 3 + 2) / 3) = (3, 2)。

表 8-1　符号表

符　号	描　写
$\mathbf{x}$	对象
C_i	第 i 个簇
$\mathbf{c}_i$	簇 C_i 的质心
$\mathbf{c}$	所有点的质心
m_i	第 i 个簇中对象的个数
m	数据集中对象的个数
K	簇的个数

K 均值算法的步骤 3 和步骤 4 试图直接最小化 SSE（或更一般地，目标函数）。步骤 3 通过将点指派到最近的质心形成簇，最小化关于给定质心集的 SSE；而步骤 4 重新计算质心，进一步最小化 SSE。然而，K 均值的步骤 3 和步骤 4 只能确保找到关于 SSE 的局部最优，因为它们是对选定的质心和簇，而不是对所有可能的选择来优化 SSE。稍后，我们将用一个例子说明这将导致次最优聚类。

文档数据　为了解释K均值并不局限于欧几里得空间数据，我们考虑文档数据和余弦相似性度量。这里，我们假定文档数据用文档—词矩阵表示，如图 2-2 所示。我们的目标是最大化簇中文档与簇的质心的相似性；该量称作簇的**凝聚度**（cohesion）。对于该目标，可以证明，与欧几里得数据一样，簇的质心是均值。总SSE的类似量是总凝聚度（total cohesion），由公式（8-3）给出。

$$\text{Total Cohesion} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} cosine(\mathbf{x}, \mathbf{c}_i) \tag{8-3}$$

一般情况　一些邻近性函数、质心和目标函数可以用于基本 K 均值算法，并且确保收敛。表 8-2 列举了一些组合，包括我们刚讨论的两种。注意：对于曼哈顿距离（L_1）和最小化距离和的目标，合适的质心是簇中各点的中位数。

表 8-2　K 均值：常见的邻近度、质心和目标函数组合

邻近度函数	质　心	目标函数
曼哈顿距离（L_1）	中位数	最小化对象到其簇质心的 L_1 距离和
平方欧几里得距离（L_2^2）	均值	最小化对象到其簇质心的 L_2 距离的平方和
余弦	均值	最大化对象与其簇质心的余弦相似度和
Bregman 散度	均值	最小化对象到其簇质心的 Bregman 散度和

表的最后一项，Bregman散度（见 2.4.5 节）实际上是一类邻近性度量，包括平方欧几里得距离 L_2^2，Mahalanobis距离和余弦相似度。Bregman散度函数的重要性在于，任意这类函数都可以用作以均值为质心的K均值类型的聚类算法的基础。具体地说，如果我们用Bregman散度作为邻近度函数，则聚类算法的收敛性、局部最小等性质与通常的K均值相同。此外，对于所有可能的Bregman散度函数，都可以开发具有这样性质的聚类算法。事实上，使用余弦相似度或平方欧几里得距离的K均值算法是基于Bregman散度的一般聚类算法的特例。

在接下来的 K 均值讨论中，我们使用二维数据，因为对于这种类型的数据，容易解释 K 均值及其性质。但是，正如前面几段所述，K 均值是非常一般的聚类算法，可以用于许多类型的数据，如文档和时间序列。

3. 选择初始质心

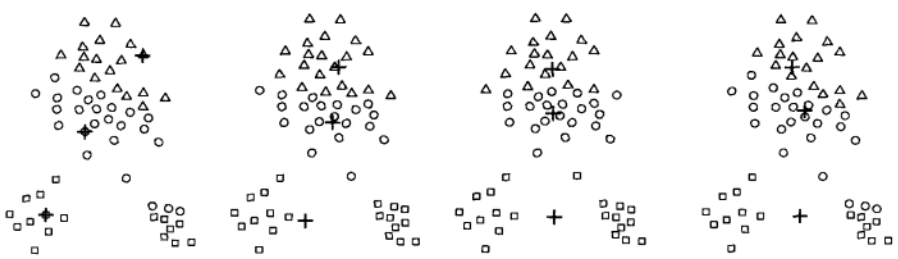
当质心随机初始化时，K均值的不同运行将产生不同的总SSE。我们用图8-3中的二维点集解释这一点。该二维点集具有三个自然点簇。图8-4a显示了一个聚类结果，三个簇的SSE是全局最小的，而图8-4b显示了一个次最优聚类，它只有局部最小。

(a) 最优聚类　　(b) 次最优聚类

图8-4 三个最优和非最优簇

选择适当的初始质心是基本K均值过程的关键步骤。常见的方法是随机地选取初始质心，但是簇的质量常常很差。

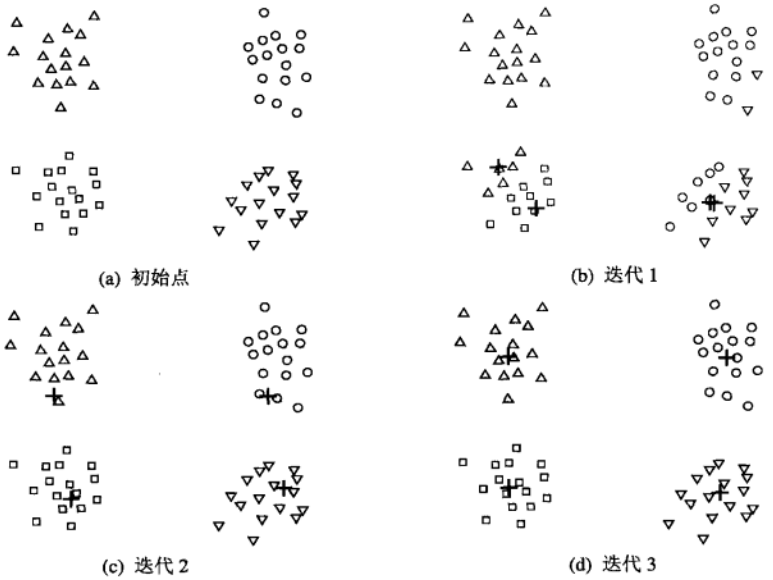
例8.1 拙劣的初始质心 随机地选取初始质心可能很糟糕。我们提供一个例子，使用与图8-3和图8-4相同的数据集。图8-3和图8-5显示了由两种选定的初始质心获得的簇。（对于这两个图，各次迭代的簇质心位置由“+”指出。）在图8-3中，尽管所有的初始质心都在自然簇中，但是仍然找到了最小SSE聚类。而在图8-5中，尽管初始质心的分布看上去较好，但是我们仅得到了一个次最优聚类，具有较高的平方误差。 □



(a) 迭代1　(b) 迭代2　(c) 迭代3　(d) 迭代4

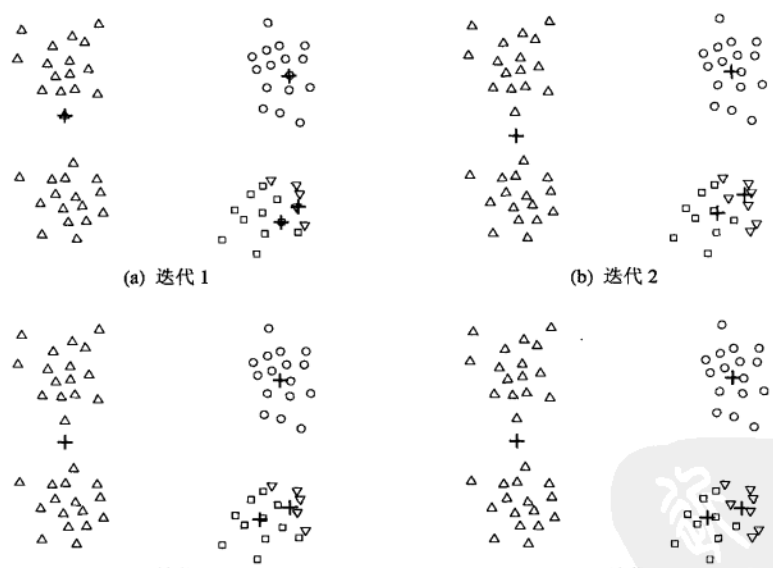
图8-5 K均值的拙劣的初始质心

例8.2 随机初始化的局限 处理选取初始质心问题的一种常用技术是：多次运行，每次使用一组不同的随机初始质心，然后选取具有最小SSE的簇集。该策略虽然简单，但是效果可能不好，这取决于数据集和寻找的簇的个数。我们使用图8-6a所示的数据集进行解释。该数据由两个簇对组成，其中，每个簇对（上、下）中的簇更靠近，而离另一对中的簇较远。图8-6b～图8-6d表明，如果我们对每个簇对用两个初始质心，则即使两个质心在一个簇中，质心也会自己重新分布，从而找到“真正的”簇。而图8-7表明，如果一个簇对只用一个初始质心，而另一对有三个，则两个真正的簇将合并，而一个真正的簇将分裂。



(a) 初始点　(b) 迭代 1　(c) 迭代 2　(d) 迭代 3

图 8-6　两个簇对，每个簇对有一对初始质心



(a) 迭代 1　(b) 迭代 2　(c) 迭代 3　(d) 迭代 4

图 8-7　两个簇对，簇对中的初始质心多于或少于两个

注意：只要两个初始质心落在簇对的任何位置，就能得到最优聚类，因为质心将自己重新分布，每个簇一个。不幸的是，随着簇的个数增加，至少一个簇对只有一个初始质心的可能性也逐步增大（见本章习题 4）。在这种情况下，由于簇对相距较远，K 均值算法不能在簇对之间重新分布质心，这样就只能得到局部最优。 □

随机选择初始质心存在的问题即使重复运行多次也不能克服，因此常常使用其他技术进行初始化。一种有效的方法是，取一个样本，并使用层次聚类技术对它聚类。从层次聚类中提取 K 个簇，并用这些簇的质心作为初始质心。该方法通常很有效，但仅对下列情况有效：(1)样本相对较小，例如数百到数千（层次聚类开销较大）；(2)K 相对于样本大小较小。

下面的过程是另一种选择初始质心的方法。随机地选择第一个点，或取所有点的质心作为第一个点。然后，对于每个后继初始质心，选择离已经选取过的初始质心最远的点。使用这种办法，我们得到初始质心的集合，确保不仅是随机的，而且是散开的。然而，这种方法可能选中离群点，而不是稠密区域（簇）中的点。此外，求离当前初始质心集最远的点开销也非常大。为了克服这些问题，通常将该方法用于点样本。由于离群点很少，它们多半不会在随机样本中出现。相比之下，除非样本非常小，否则来自稠密区域中的点很可能包含在样本中。此外，找出初始质心所需要的计算量也大幅度减少，因为样本的大小通常远小于点的个数。

稍后，我们将讨论另外两种产生较高质量（较低 SSE）聚类的方法：使用对初始化问题不太敏感的 K 均值的变种（二分 K 均值）、使用后处理来“修补”所产生的簇集。

时间复杂性和空间复杂性

K 均值的空间需求是适度的，因为只需要存放数据点和质心。具体地说，所需要的存储量为 $O((m+K)n)$，其中 m 是点数，n 是属性数。K 均值的时间需求也是适度的——基本上与数据点个数线性相关。具体地说，所需要的时间为 $O(I\times K\times m\times n)$，其中 I 是收敛所需要的迭代次数。如前所述，I 通常很小，可以是有界的，因为大部分变化通常出现在前几次迭代。因此，只要簇个数 K 显著小于 m，则 K 均值的计算时间与 m 线性相关，并且是有效的和简单的。

8.2.2 K均值：附加的问题

1. 处理空簇

前面介绍的基本K均值算法存在的问题之一是：如果所有的点在指派步骤都未分配到某个簇，就会得到空簇。如果这种情况发生，则需要某种策略来选择一个替补质心，否则的话，平方误差将会偏大。一种方法是选择一个距离当前任何质心最远的点。这将消除当前对总平方误差影响最大的点。另一种方法是从具有最大SSE的簇中选择一个替补质心。这将分裂簇并降低聚类的总SSE。如果有多个空簇，则该过程重复多次。

2. 离群点

使用平方误差标准时，离群点可能过度影响所发现的簇。具体地说，当存在离群点时，结果簇的质心（原型）可能不如没有离群点时那样有代表性，并且SSE也比较高。正因为如此，提前发现离群点并删除它们是有用的。然而，应当意识到有一些聚类应用，不能删除离群点。当聚类用来压缩数据时，必须对每个点聚类。在某些情况下（如财经分析），明显的离群点（如不寻常的有利可图的顾客）可能是最令人感兴趣的点。

一个明显的问题是如何识别离群点。第 10 章将讨论一些识别离群点的技术。如果我们使用的方法在聚类前就删除离群点，则我们就避免了对不能很好聚类的点进行聚类。当然也可以在后处理时识别离群点。例如，我们可以记录每个点对SSE的影响，删除那些具有异乎寻常影响的点（尤其是多次运行算法时）。此外，我们还可能需要删除那些很小的簇，因为它们常常代表离群点的组。

3. 用后处理降低 SSE

一种明显降低 SSE 的方法是找出更多簇，即使用较大的 K。然而，在许多情况下，我们希望降低 SSE，但并不想增加簇的个数。这是可能的，因为 K 均值常常收敛于局部极小。可以使用多种技术来“修补”结果簇，以便产生具有较小 SSE 的聚类。策略是关注每一个簇，因为总 SSE 只不过是每个簇的 SSE 之和。（为了避免混淆，我们将分别使用术语总 SSE 和簇 SSE。）通过在簇上进行诸如分裂和合并等操作，我们可以改变总 SSE。一种常用的方法是交替地使用簇分裂和簇合并。在分裂阶段将簇分开，而在合并阶段将簇合并。用这种方法，常常可以避开局部极小，并且仍然能够得到具有期望个数簇的聚类。下面是一些用于分裂和合并阶段的技术。

通过增加簇个数来降低总 SSE 的两种策略如下。

- **分裂一个簇**：通常选择具有最大 SSE 的簇，但是我们也可以分裂在特定属性上具有最大标准差的簇。
- **引进一个新的质心**：通常选择离所有簇质心最远的点。如果我们记录每个点对 SSE 的贡献，则可以容易地确定最远的点。另一种方法是从所有的点或者具有最高 SSE 的点中随机地选择。

减少簇个数，而且试图最小化总 SSE 的增长的两种策略如下。

- **拆散一个簇**：删除簇的对应质心，并将簇中的点重新指派到其他簇。理想情况下，被拆散的簇应当是使总 SSE 增加最少的簇。
- **合并两个簇**：通常选择质心最接近的两个簇，尽管另一种方法（合并两个导致总 SSE 增加最少的簇）或许更好。这两种合并策略与层次聚类使用的方法相同，分别称作质心方法和 Ward 方法。这两种方法将在 8.3 节讨论。

4. 增量地更新质心

可以在点到簇的每次指派之后，增量地更新质心，而不是在所有的点都指派到簇中之后才更新簇质心。注意，每步需要零次或两次簇质心更新，因为一个点或者转移到一个新的簇（两次更新），或者留在它的当前簇（零次更新）。使用增量更新策略确保不会产生空簇，因为所有的簇都从单个点开始；并且如果一个簇只有单个点，则该点总是被重新指派到相同的簇。

此外，如果使用增量更新，则可以调整点的相对权值；例如，点的权值通常随聚类的进行而减小。尽管这可能产生更好的准确率和更快的收敛性，但是在千变万化的情况下，选择好的相对权值可能是困难的。这些更新问题类似于人工神经网络的权值更新。

增量更新的另一个优点是使用不同于“最小化 SSE”的目标。假设给定一个度量簇集的目标函数。当我们处理某个点时，我们可以对每个可能的簇指派计算目标函数的值，然后选择优化目标的簇指派。可选的目标函数的具体例子在 8.5.2 节给出。

缺点方面，增量地更新质心可能导致次序依赖性。换言之，所产生的簇可能依赖于点的处理次序。尽管随机地选择点的处理次序可以解决该问题，但是，基本K均值方法在把所有点指派到簇中之后才更新质心并没有次序依赖性。此外，增量更新的开销也稍微大一些。然而，K均值收敛相当快，因此切换簇的点数很快就会变小。

8.2.3　二分 K 均值

二分 K 均值算法是基本 K 均值算法的直接扩充，它基于一种简单想法：为了得到 K 个簇，将所有点的集合分裂成两个簇，从这些簇中选取一个继续分裂，如此下去，直到产生 K 个簇。二

分 K 均值的细节由算法 8.2 给出。

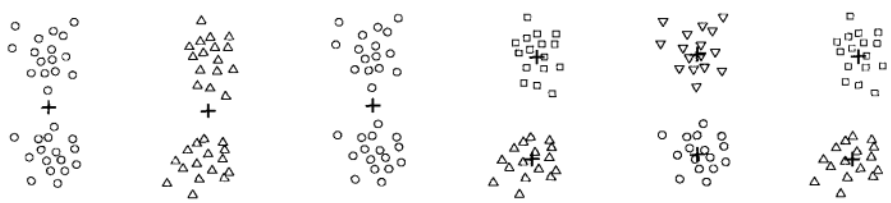
算法 8.2 二分 K 均值算法

```
1: 初始化簇表，使之包含由所有的点组成的簇。
2: repeat
3:    从簇表中取出一个簇。
4:    {对选定的簇进行多次二分“试验”。}
5:    for i = 1 to 试验次数 do
6:       使用基本 K 均值，二分选定的簇。
7:    end for
8:    从二分试验中选择具有最小总 SSE 的两个簇。
9:    将这两个簇添加到簇表中。
10: until 簇表中包含 K 个簇。
```

待分裂的簇有许多不同的选择方法。可以选择最大的簇，选择具有最大 SSE 的簇，或者使用一个基于大小和 SSE 的标准进行选择。不同的选择导致不同的簇。

我们通常使用结果簇的质心作为基本 K 均值的初始质心，对结果簇逐步求精。这是必要的，因为尽管 K 均值算法可以确保找到使 SSE 局部最小的聚类，但是在二分 K 均值算法中，我们“局部地”使用了 K 均值算法，即二分个体簇。因此，最终的簇集并不代表使 SSE 局部最小的聚类。

例 8.3 二分 K 均值和初始化 为了说明二分 K 均值不太受初始化问题的影响，在图 8-8 中，我们展示二分 K 均值如何找到图 8-6a 所示数据集中的 4 个簇。迭代 1 找到了两个簇对，迭代 2 分裂了最右边的簇对，迭代 3 分裂了最左边的簇对。二分 K 均值不太受初始化的困扰，因为它执行了多次二分试验并选取具有最小 SSE 的试验结果，还因为每步只有两个质心。 □



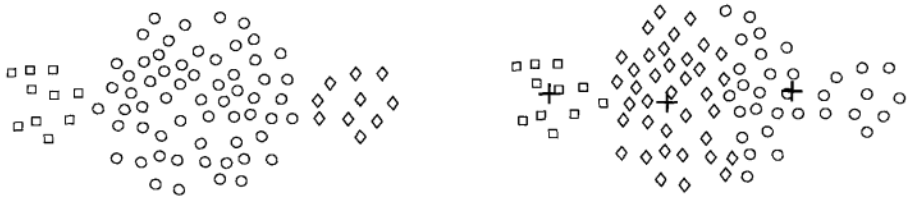
(a) 迭代 1　　(b) 迭代 2　　(c) 迭代 3

图 8-8 4 个簇的例子上的二分 K 均值

最后，通过记录 K 均值二分簇所产生的聚类序列，我们还可以使用二分 K 均值产生层次聚类。

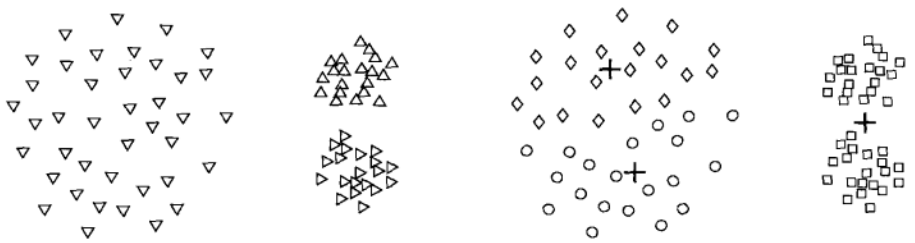
8.2.4 K 均值和不同的簇类型

对于发现不同的簇类型，K均值和它的变种都具有一些局限性。具体地说，当簇具有非球形形状或具有不同尺寸或密度时，K均值很难检测到“自然的”簇，如图 8-9、图 8-10 和图 8-11 所示。在图 8-9 中，K均值不能发现那三个自然簇，因为其中一个簇比其他两个大得多，因此较大的簇被分开，而一个较小的簇与较大簇的一部分合并到一起。在图 8-10 中，K均值未能发现那三个自然簇，因为两个较小的簇比较大的簇稠密得多。最后，在图 8-11 中，K均值发现了两个簇（两个自然簇的混合体），因为两个自然簇的形状不是球形的。



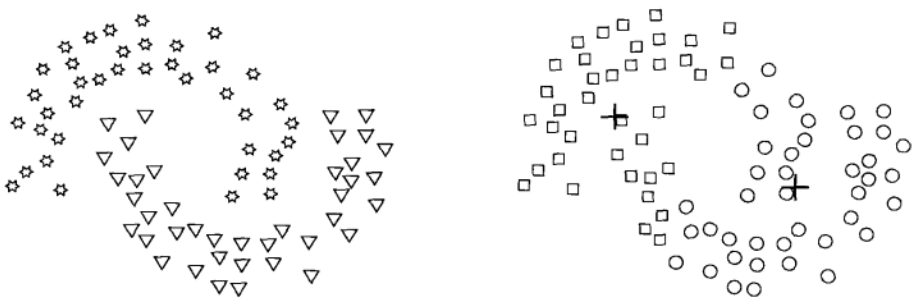
(a)原来的点　　(b)三个 K 均值簇

图 8-9　K 均值：具有不同尺寸的簇



(a) 原来的点　　(b) 三个 K 均值簇

图 8-10　K 均值：具有不同密度的簇



(a) 原来的点　　(b) 两个 K 均值簇

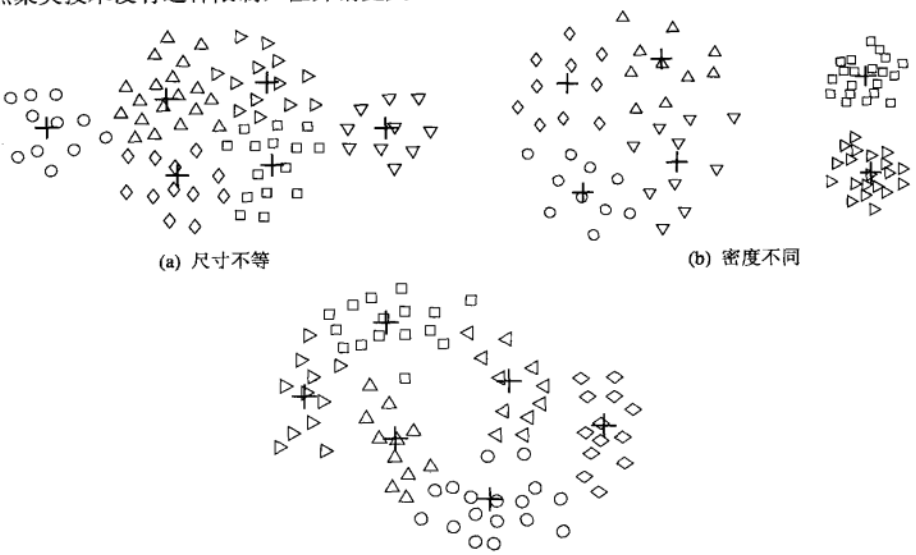
图 8-11　K 均值：非球形的簇

这三种情况的问题在于 K 均值的目标函数与我们试图发现的簇的类型不匹配，因为 K 均值目标函数是最小化等尺寸和等密度的球形簇，或者明显分离的簇。然而，如果用户愿意接受将一个自然簇分割成若干子簇的聚类的话，这些局限性在某种意义下可以克服。图 8-12 显示如果我们找 6 个簇，而不是 2 个或 3 个的话，前面 3 个数据集所发生的情况。在仅包含一个自然簇的点这种意义下，每个较小的簇都是纯的。

8.2.5　优点与缺点

K 均值简单并且可以用于各种数据类型。它也相当有效，尽管常常多次运行。K 均值的某些变种（包括二分 K 均值）甚至更有效，并且不太受初始化问题的影响。然而，K 均值并不适合所有的数据类型。它不能处理非球形簇、不同尺寸和不同密度的簇，尽管指定足够大的簇个数时它通常可以发现纯子簇。对包含离群点的数据进行聚类时，K 均值也有问题。在这种情况下，离

群点检测和删除大有帮助。最后，K 均值仅限于具有中心（质心）概念的数据。一种相关的 K 中心点聚类技术没有这种限制，但开销更大。



(a) 尺寸不等　(b) 密度不同　(c) 非球形形状

图 8-12　使用 K 均值发现自然簇的子簇

8.2.6 K 均值作为优化问题

这里，我们深入研究支撑 K 均值的数学问题。本节需要包括偏导数在内的微积分知识，不熟悉这方面知识的读者可以跳过本节，也不会失去学习的连贯性。熟悉优化技术，特别是基于梯度下降的优化技术，可能也有帮助。

正如前面提到的，给定一个诸如“最小化 SSE”这样的目标函数，可以把聚类视为优化问题。解决该问题（找出全局最优）的一种方法是：枚举将点划分成簇的所有可能方法，然后选择最好地满足目标函数（例如，最小化总 SSE）的簇集。当然，这种穷举的策略不是计算可行的，因此需要更实际的方法，即使这样的方法发现的解不能保证是最优的。一种称作**梯度下降**（gradient descent）的技术选择一个初始解，然后重复如下两步：计算最好地优化目标函数的解的改变，然后更新解。

我们假定数据是一维的，即 $dist(x, y) = (x - y)^2$。这本质上没有改变任何东西，但是大大简化了论证过程。

1. 作为最小化 SSE 的算法推导 K 均值

本节，我们说明，当邻近函数是欧几里得距离并且目标是最小化 SSE 时，K 均值的质心如何从数学上推导出来。具体地说，我们考察如何最好地更新簇质心，使得簇 SSE 最小化。使用数学术语，我们试图最简化公式（8-1）。对于一维数据，公式（8-1）可以写成：

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \qquad (8\text{-}4)$$

这里，C_i 是第 i 个簇，x 是 C_i 中的点，c_i 是第 i 个簇的均值。记号的完整列表见表 8-1。

我们可以对第 k 个质心 c_k 求解，最小化公式（8-4）；即对 SSE 求导，令导数等于 0，并求解 c_k，如下所示：

$$\begin{aligned}\frac{\partial}{\partial c_k}\mathrm{SSE} &= \frac{\partial}{\partial c_k}\sum_{i=1}^{K}\sum_{x\in C_i}(c_i - x)^2 \\ &= \sum_{i=1}^{K}\sum_{x\in C_i}\frac{\partial}{\partial c_k}(c_i - x)^2 \\ &= \sum_{x\in C_k} 2(c_k - x_k) = 0\end{aligned}$$

$$\sum_{x\in C_k} 2(c_k - x_k) = 0 \Rightarrow m_k c_k = \sum_{x\in C_k} x_k \Rightarrow c_k = \frac{1}{m_k}\sum_{x\in C_k} x_k$$

这样，正如前面所指出的，簇的最小化 SSE 的最佳质心是簇中各点的均值。

2. 为 SAE 推导 K 均值

为了表明 K 均值可以用于各种不同的目标函数，我们考虑如何将数据划分成 K 个簇，使得点到其簇中心的曼哈顿距离（L_1）之和最小。我们寻求最小化下式给出的 L_1 绝对误差和（SAE）：

$$\mathrm{SAE} = \sum_{i=1}^{K}\sum_{x\in C_i} dist_{L_1}(c_i, x)^2 \tag{8-5}$$

其中 $dist_{L_1}$ 是 L_1 距离。为了简单起见，我们再次使用一维数据，即 $dist_{L_1} = |c_i - x|$。

我们可以对第 k 个质心 c_k 求解，最小化公式（8-5），即对 SAE 求导，令导数等于 0，并求解 c_k。

$$\begin{aligned}\frac{\partial}{\partial c_k}\mathrm{SAE} &= \frac{\partial}{\partial c_k}\sum_{i=1}^{K}\sum_{x\in C_i}|c_i - x| \\ &= \sum_{i=1}^{K}\sum_{x\in C_i}\frac{\partial}{\partial c_k}|c_i - x| \\ &= \sum_{x\in C_k}\frac{\partial}{\partial c_k}|c_k - x| = 0\end{aligned}$$

$$\sum_{x\in C_k}\frac{\partial}{\partial c_k}|c_k - x| = 0 \Rightarrow \sum_{x\in C_k} sign(x - c_k) = 0$$

如果对 c_k 求解，可以发现 $c_k = median\{x \in C_k\}$，即簇中各点的中位数。一组点的中位数的计算是直截了当的，并且较少受离群点扰动的影响。

8.3　凝聚层次聚类

层次聚类技术是第二类重要的聚类方法。与K均值一样，与许多聚类方法相比，这些方法相对较老，但是它们仍然被广泛使用。有两种产生层次聚类的基本方法。

- **凝聚的**：从点作为个体簇开始，每一步合并两个最接近的簇。这需要定义簇的邻近性概念。
- **分裂的**：从包含所有点的某个簇开始，每一步分裂一个簇，直到仅剩下单点簇。在这种情况下，我们需要确定每一步分裂哪个簇，以及如何分裂。

到目前为止，凝聚层次聚类技术最常见，本节我们只关注这类方法。分裂的层次聚类技术将

在 9.4.2 节介绍。

层次聚类常常使用称作**树状图**（dendrogram）的类似于树的图显示。该图显示簇-子簇联系和簇合并（凝聚）或分裂的次序。对于二维点的集合（如我们将用作例子的那些），层次聚类也可以使用嵌套簇图（nested cluster diagram）表示。图 8-13 对 4 个二维点的集合，显示了这两种图的例子。这些点使用 8.3.2 节介绍的单链技术聚类。

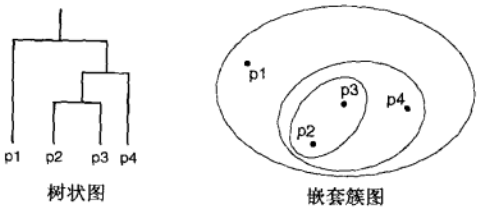


图 8-13 以树状图和嵌套簇图显示的 4 个点的层次聚类

8.3.1 基本凝聚层次聚类算法

许多凝聚层次聚类技术都是这个方法的变种：从个体点作为簇开始，相继合并两个最接近的簇，直到只剩下一个簇。该方法更形式地表达在算法 8.3 中。

算法 8.3 基本凝聚层次聚类算法

1：如果需要，计算邻近度矩阵
2：**repeat**
3：　合并最接近的两个簇
4：　更新邻近性矩阵，以反映新的簇与原来的簇之间的邻近性
5：**until** 仅剩下一个簇

1. 定义簇之间的邻近性

算法 8-3 的关键操作是计算两个簇之间的邻近度，并且正是簇的邻近性定义区分了我们将讨论的各种凝聚层次技术。簇的邻近性通常用特定的簇类型定义，见 8.1.3 节。例如，许多凝聚层次聚类技术，如 MIN、MAX 和组平均，都源于簇的基于图的观点。MIN 定义簇的邻近度为不同簇的两个最近的点之间的邻近度，或者使用图的术语，不同的结点子集中两个结点之间的最短边。这产生了图 8-2c 所示的基于邻近的簇。MAX 取不同簇中两个最远的点之间的邻近度作为簇的邻近度，或者使用图的术语，不同的结点子集中两个结点之间的最长边。（如果我们的邻近度是距离，则 MIN 和 MAX 这两个名字短小并且有提示作用。然而，对于相似度，值越大点越近，名字看上去是相反的。因此，我们通常使用替换的名字，分别为**单链**（single link）和**全链**（complete link）。）另一种基于图的方法是**组平均**（group average）技术。它定义簇邻近度为取自不同簇的所有点对邻近度的平均值（平均边长）。图 8-14 图示了这三种方法。

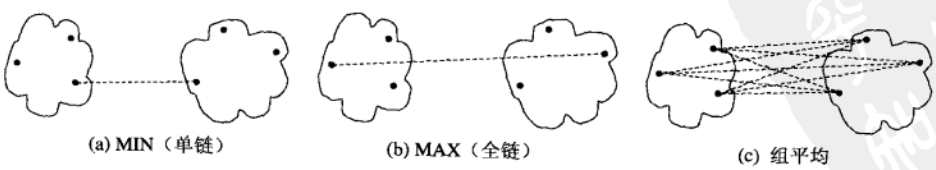


图 8-14 簇的邻近度的基于图的定义

如果我们取基于原型的观点，簇用质心代表，则不同的簇邻近度定义就更加自然。使用质心时，簇的邻近度一般定义为簇质心之间的邻近度。另一种技术，Ward 方法，也假定簇用其质心代表，但它使用合并两个簇导致的 SSE 增加来度量两个簇之间的邻近性。像 K 均值一样，Ward 方法也试图最小化点到其簇质心的距离的平方和。

2. 时间和空间复杂性

基本凝聚层次聚类算法使用邻近度矩阵。这需要存储 $m^2/2$ 个邻近度（假定邻近度矩阵是对称的），其中 m 是数据点的个数。记录簇所需要的空间正比于簇的个数为 $m-1$，不包括单点簇。因此总的空间复杂度为 $O(m^2)$。

基本凝聚层次聚类算法的计算复杂度分析也是很明确的，即需要$O(m^2)$时间计算邻近度矩阵。之后，步骤 3 和 4 涉及$m-1$ 次迭代，因为开始有m个簇，而每次迭代合并两个簇。如果邻近度矩阵采用线性搜索，则对于第i次迭代，步骤 3 需要$O((m-i+1)^2)$时间，这正比于当前簇个数的平方。步骤 4 只需要$O(m-i+1)$时间，在合并两个簇后更新邻近度矩阵。（对于我们考虑的技术，簇合并只影响$O(m-i+1)$个邻近度。）不作修改，时间复杂度将为$O(m^3)$。如果某个簇到其他所有簇的距离存放在一个有序表或堆中，则查找两个最近簇的开销可能降低到$O(m-i+1)$。然而，由于维护有序表或堆的附加开销，基于算法 8.3 的层次聚类所需要的总时间为$O(m^2\log m)$。

层次聚类的空间和时间复杂度严重地限制了它所能够处理的数据集的大小。我们将在 9.5 节讨论聚类算法的可伸缩方法，包括层次聚类技术。

8.3.2　特殊技术

1. 样本数据

为了解释各种层次聚类算法，我们将使用包含6个二维点的样本数据，如图8-15。点的 x 和 y 坐标，以及点之间的欧几里得距离分别在表8-3和表8-4中。

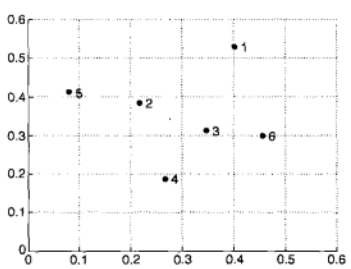


图 8-15　6 个二维点的集合

表 8-3　6 个点的 xy 坐标

点	x 坐标	y 坐标
p1	0.400 5	0.530 6
p2	0.214 8	0.385 4
p3	0.345 7	0.315 6
p4	0.265 2	0.187 5
p5	0.078 9	0.413 9
p6	0.454 8	0.302 2

表 8-4　6 个点的欧几里得距离矩阵

	p1	p2	p3	p4	p5	p6
p1	0.0000	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0.0000	0.1483	0.2042	0.1388	0.2540
p3	0.2218	0.1483	0.0000	0.1513	0.2843	0.1100
p4	0.3688	0.2042	0.1513	0.0000	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0.0000	0.3921
p6	0.2347	0.2540	0.1100	0.2216	0.3921	0.0000

2. 单链或 MIN

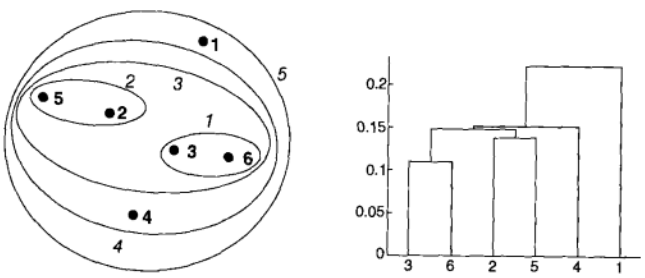
对于层次聚类的单链或 MIN 版本，两个簇的邻近度定义为两个不同簇中任意两点之间的最

短距离（最大相似度）。使用图的术语，如果我们从所有点作为单点簇开始，每次在点之间加上一条链，最短的链先加，则这些链将点合并成簇。单链技术擅长于处理非椭圆形状的簇，但对噪声和离群点很敏感。

例 8.4 单链 图 8-16 显示了将单链技术用于 6 个点数据集例子的结果。图 8-16a 用嵌套的椭圆序列显示嵌套的簇，其中与椭圆相关联的数指示聚类的次序。图 8-16b 显示了同样的信息，但使用树状图表示。树状图中两个簇合并处的高度反映两个簇的距离。例如，由表 8-4，我们看到点 3 和 6 的距离是 0.11，这就是它们在树状图里合并处的高度。作为另一个例子，簇{3, 6}和{2, 5}之间的距离是

$$\begin{aligned} dist(\{3,6\},\{2,5\}) &= \min(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\ &= \min(0.15, 0.25, 0.28, 0.39) \\ &= 0.15 \end{aligned}$$

□



(a) 单链聚类　　(b) 单链树状图

图 8-16 图 8-15 中 6 个点的单链聚类

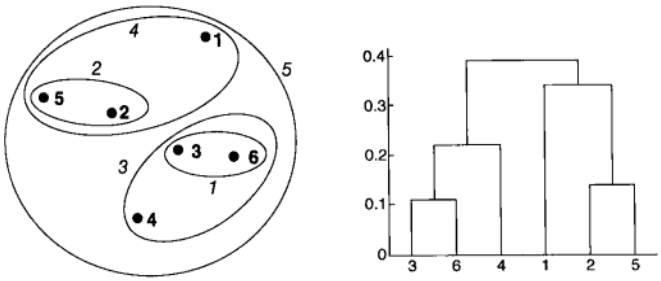
3. 全链或 MAX 或团

对于层次聚类的全链或 MAX 版本，两个簇的邻近度定义为两个不同簇中任意两点之间的最长距离（最小相似度）。使用图的术语，如果我们从所有点作为单点簇开始，每次在点之间加上一条链，最短的链先加，则一组点直到其中所有的点都完全被连接（即形成团）才形成一个簇。完全连接对噪声和离群点不太敏感，但是它可能使大的簇破裂，并且偏好球形。

例 8.5 全链 图 8-17 显示了将 MAX 用于 6 个点样本数据集的结果。与单链一样，点 3 和 6 首先合并。然而，{3, 6}与{4}合并，而不是与{2, 5}或{1}合并，因为

$$\begin{aligned} dist(\{3,6\},\{4\}) &= \max(dist(3,4), dist(6,4)) \\ &= \max(0.15, 0.22) \\ &= 0.22 \\ dist(\{3,6\},\{2,5\}) &= \max(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\ &= \max(0.15, 0.25, 0.28, 0.39) \\ &= 0.39 \\ dist(\{3,6\},\{1\}) &= \max(dist(3,1), dist(6,1)) \\ &= \max(0.22, 0.23) \\ &= 0.23 \end{aligned}$$

□



(a) 全链聚类　　(b) 全链树状图

图 8-17　图 8-15 中 6 个点的全链聚类

4. 组平均

对于层次聚类的组平均版本，两个簇的邻近度定义为不同簇的所有点对邻近度的平均值。这是一种界于单链和全链之间的折中方法。对于组平均，簇 C_i 和 C_j 的邻近度 $proximity(C_i, C_j)$ 由下式定义：

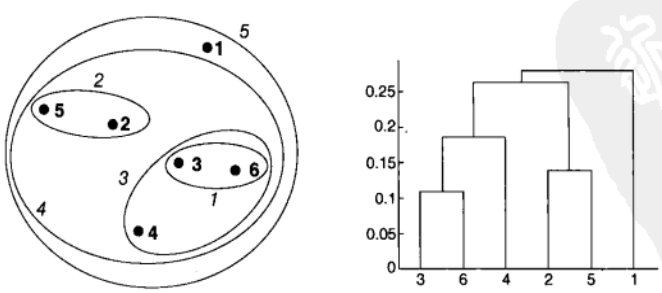
$$proximity(C_i, C_j) = \frac{\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_j}} proximity(\mathbf{x}, \mathbf{y})}{m_i * m_j} \tag{8-6}$$

其中，m_i 和 m_j 分别是簇 C_i 和 C_j 的大小。

例 8.6　组平均　图 8-18 显示了将组平均用于 6 个点样本数据集的结果。为了解释组平均如何工作，我们计算某些簇之间的距离

$$\begin{aligned} dist(\{3,6,4\},\{1\}) &= (0.22 + 0.37 + 0.23)/(3*1) \\ &= 0.28 \\ dist(\{2,5\},\{1\}) &= (0.2357 + 0.3421)/(2*1) \\ &= 0.2889 \\ dist(\{3,6,4\},\{2,5\}) &= (0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)/(3*2) \\ &= 0.26 \end{aligned}$$

因为 $dist(\{3,6,4\},\{2,5\})$ 比 $dist(\{3,6,4\},\{1\})$ 和 $dist(\{2,5\},\{1\})$ 小，簇{3, 6, 4}和{2, 5}在第 4 阶段合并。 □



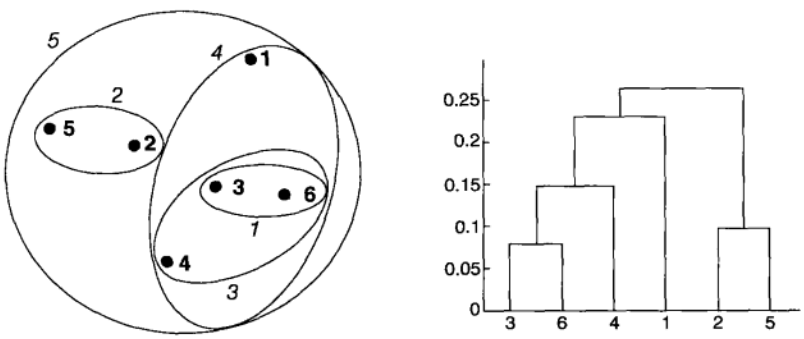
(a) 组平均聚类　　(b) 组平均树状图

图 8-18　图 8-15 中 6 个点的组平均聚类

5. Ward 方法和质心方法

对于 Ward 方法，两个簇的邻近度定义为两个簇合并时导致的平方误差的增量。这样一来，该方法使用的目标函数与 K 均值相同。尽管看上去这一特点使得 Ward 方法不同于其他层次聚类技术，但是可以从数学上证明：当两个点之间的邻近度取它们之间距离的平方时，Ward 方法与组平均非常相似。

例 8.7 Ward 方法 图 8-19 显示了将 Ward 方法用于 6 个点样本数据集的结果。所产生的聚类与单链、全链、组平均不同。 □



(a) Ward 聚类 (b) Ward 树状图

图 8-19 图 8-15 中 6 个点的 Ward 聚类

质心方法通过计算簇质心之间的距离来计算两个簇之间的邻近度。这种技术看上去与 K 均值类似，但是，正如我们论述的那样，Ward 方法才真正与它类似。

质心方法还具有一种我们讨论过的其他层次聚类技术不具备的特性（常被认为是坏的）：倒置(inversion)的可能性。具体地说，合并的两个簇可能比前一步合并的簇对更相似。对于其他方法，被合并的簇之间的距离随层次聚类进展单调地增加（或者，在最坏情况下不增加）。

8.3.3 簇邻近度的 Lance-Williams 公式

本节我们讨论过的任何簇邻近度都可以看作簇 Q 和 R 之间邻近度的不同参数（下面公式（8-7）显示的 Lance-Williams 公式）的一种选择，其中 R 是通过合并簇 A 和 B 形成的。在公式（8-7）中，$p(.\ ,\ .)$是邻近度函数，而 m_A、m_B 和 m_Q 分别是簇 A、B 和 Q 的点数。换言之，合并簇 A 和 B 形成簇 R 之后，新簇 R 与原簇 Q 的邻近度是 Q 与原来的簇 A 和 B 的邻近度的线性函数。表8-5对我们讨论过的技术显示了这些系数的值。

$$p(R, Q) = \alpha_A p(A, Q) + \alpha_B p(B, Q) + \beta p(A, B) + \gamma |p(A, Q) - p(B, Q)| \qquad (8\text{-}7)$$

任何可以使用 Lance-Williams 公式表示的层次聚类技术都不需要保留原来的数据点。也就是说，邻近度矩阵随聚类而更新。尽管通用公式是吸引人的，特别是对于实现。但是，通过直接考察每种方法使用的簇邻近度定义，更加容易理解层次聚类之间的不同。

表 8-5　常见层次聚类方法的 Lance-Williams 系数

聚类方法	α_A	α_B	β	γ
单链	1/2	1/2	0	-1/2
全链	1/2	1/2	0	1/2
组平均	$\frac{m_A}{m_A+m_B}$	$\frac{m_B}{m_A+m_B}$	0	0
质心	$\frac{m_A}{m_A+m_B}$	$\frac{m_B}{m_A+m_B}$	$\frac{-m_A m_B}{(m_A+m_B)^2}$	0
Ward	$\frac{m_A+m_Q}{m_A+m_B+m_Q}$	$\frac{m_B+m_Q}{m_A+m_B+m_Q}$	$\frac{-m_Q}{m_A+m_B+m_Q}$	0

8.3.4　层次聚类的主要问题

1. 缺乏全局目标函数

如前所述，凝聚层次聚类不能视为全局优化目标函数。也就是说，凝聚层次聚类技术使用各种标准，在每一步局部地确定哪些簇应当合并（或分裂，对于分裂方法）。这种方法产生的聚类算法避开了解决困难的组合优化问题。（可以证明：对于诸如“最小化 SSE”这样的目标函数，一般聚类问题不是计算可行的。）此外，这样的方法没有局部极小问题或很难选择初始点的问题。当然，在许多情况下，$O(m^2 \log m)$的时间复杂度和 $O(m^2)$的空间复杂度也阻碍了它们的应用。

2. 处理不同大小簇的能力

一个我们尚未讨论的凝聚层次聚类问题是：如何处理待合并的簇对的相对大小。（该讨论仅适用于涉及求和的簇邻近性方案，如质心、Ward 方法和组平均。）有两种方法：**加权**（weighted）方法平等地对待所有簇，**非加权**（unweighted）方法考虑每个簇的点数。注意：术语加权和非加权是对数据点而言，而不是对簇。换言之，平等地对待不同大小的簇表示赋予不同簇中的点不同的权值，而考虑簇的大小则赋予不同簇中的点相同的权值。

我们使用 8.3.2 节讨论的组平均技术解释这一点，它是组平均技术的非加权版本。在聚类文献中，该方法的全称是使用算术平均的、非加权的对组方法（Unweighted Pair Group Method using Arithmetic averages，UPGMA）。表 8-5 给出了更新簇相似度的公式，UPGMA 的系数涉及每个被合并簇的大小：$\alpha_A = \frac{m_A}{m_A+m_B}$，$\alpha_B = \frac{m_B}{m_A+m_B}$，$\beta=0$，$\gamma=0$。对于组平均的加权版本（称作 WPGMA），这些系数是常数：$\alpha_A = 1/2$，$\alpha_B = 1/2$，$\beta=0$，$\gamma=0$。通常，非加权的方法更可取，除非出于某种原因个体点具有不同的权值。例如，或许对象类非均匀地抽样。

3. 合并决策是最终的

对于合并两个簇，凝聚层次聚类算法趋向于作出好的局部决策，因为它们可以使用所有点的逐对相似度信息。然而，一旦作出合并两个簇的决策，以后就不能撤销。这种方法阻碍了局部最优标准变成全局最优标准。例如，尽管Ward方法使用K均值的“最小化平方误差”来决定合并哪些簇，但是每一层的簇并不代表总SSE局部最小。事实上，簇甚至是不稳定的，簇中的点可能离其他某个簇的质心更近，而离当前簇的质心更远。尽管如此，Ward方法还是经常作为一种初始化K均值聚类的鲁棒方法使用，表明局部“最小化平方误差”目标函数与全局“最小化平方误差”目标函数有关联。

有一些技术试图克服“合并是最终的”这一限制。一种方法试图通过如下方法来修补层次聚

类：移动树的分支以改善全局目标函数。另一种方法使用划分聚类技术（如K均值）来创建许多小簇，然后从这些小簇出发进行层次聚类。

8.3.5 优点与缺点

具体的凝聚层次聚类算法的优缺点上面已经讨论过。通常，使用这类算法是因为基本应用（如创建一种分类法）需要层次结构。此外，有些研究表明，这些算法能够产生较高质量的聚类。然而，就计算量和存储需求而言，凝聚层次聚类算法是昂贵的。所有合并都是最终的，对于噪声、高维数据（如文档数据），这也可能造成问题。先使用其他技术（如 K 均值）进行部分聚类，这两个问题都可以在某种程度上加以解决。

8.4 DBSCAN

基于密度的聚类寻找被低密度区域分离的高密度区域。DBSCAN 是一种简单、有效的基于密度的聚类算法，它解释了基于密度的聚类方法的许多重要概念。本节中，在考虑密度的主要概念之后，我们仅关注 DBSCAN。其他基于密度的聚类算法将在下一章介绍。

8.4.1 传统的密度：基于中心的方法

尽管定义密度的方法没有定义相似度的方法多，但仍存在几种不同的方法。本节中，我们讨论 DBSCAN 使用的基于中心的方法。密度的其他定义将在第 9 章提供。

在基于中心的方法中，数据集中特定点的密度通过对该点 *Eps* 半径之内的点计数（包括点本身）来估计，如图 8-20 所示。点 *A* 的 *Eps* 半径内点的个数为 7，包括 *A* 本身。

该方法实现简单，但是点的密度取决于指定的半径。例如，如果半径足够大，则所有点的密度都等于数据集中的点数 *m*。同理，如果半径太小，则所有点的密度都是 1。对于低维数据，一种确定合适半径的方法在讨论 DBSCAN 算法时给出。

根据基于中心的密度进行点分类

密度的基于中心的方法使得我们可以将点分类为(1)稠密区域内部的点（核心点），(2)稠密区域边缘上的点（边界点），(3)稀疏区域中的点（噪声或背景点）。图 8-21 使用二维点集图示了核心点、边界点和噪声点的概念。下文给出更详尽的描述。

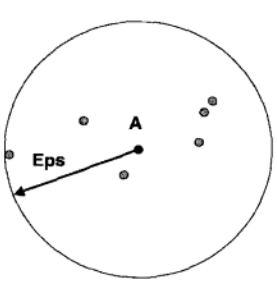


图 8-20 基于中心的密度

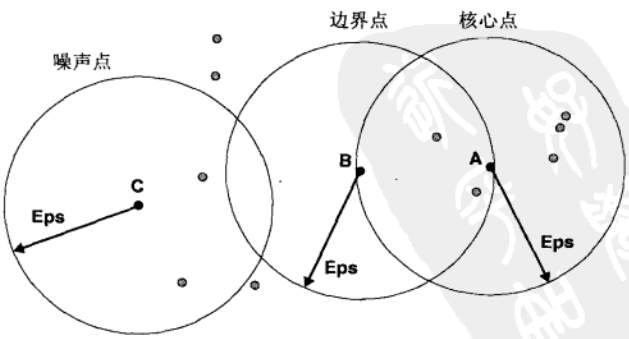


图 8-21 核心点、边界点和噪声点

- **核心点**（core point）：这些点在基于密度的簇内部。点的邻域由距离函数和用户指定的距离参数 *Eps* 决定。核心点的定义是，如果该点的给定邻域内的点的个数超过给定的阈值 *MinPts*，其中 *MinPts* 也是一个用户指定的参数。在图 8-21 中，如果 *MinPts*≤7，则对于给定的半径（*Eps*），点 *A* 是核心点。
- **边界点**（border point）：边界点不是核心点，但它落在某个核心点的邻域内。在图 8-21 中，点 *B* 是边界点。边界点可能落在多个核心点的邻域内。
- **噪声点**（noise point）：噪声点是既非核心点也非边界点的任何点。在图 8-21 中，点 *C* 是噪声点。

8.4.2　DBSCAN 算法

给定核心点、边界点和噪声点的定义，DBSCAN 算法可以非正式地描述如下。任意两个足够靠近（相互之间的距离在 *Eps* 之内）的核心点将放在同一个簇中。同样，任何与核心点足够靠近的边界点也放到与核心点相同的簇中。（如果一个边界点靠近不同簇的核心点，则可能需要解决平局问题。）噪声点被丢弃。算法的细节在算法 8.4 中给出。该算法与原来的 DBSCAN 算法使用了相同概念并发现相同的簇，但是为了简洁，而不是为了有效，做了一些优化。

算法 8.4　DBSCAN 算法

1：将所有点标记为核心点、边界点或噪声点。
2：删除噪声点。
3：为距离在 *Eps* 之内的所有核心点之间赋予一条边。
4：每组连通的核心点形成一个簇。
5：将每个边界点指派到一个与之关联的核心点的簇中。

1. 时间复杂性和空间复杂性

DBSCAN 的基本时间复杂度是 $O(m\times$找出 *Eps* 邻域中的点所需要的时间$)$，其中 m 是点的个数。在最坏情况下，时间复杂度是 $O(m^2)$。然而，在低维空间，有一些数据结构，如 kd 树，可以有效地检索特定点给定距离内的所有点，时间复杂度可以降低到 $O(m \log m)$。即便对于高维数据，DBSCAN 的空间也是 $O(m)$，因为对每个点，它只需要维持少量数据，即簇标号和每个点是核心点、边界点还是噪声点的标识。

2. 选择 DBSCAN 的参数

当然，还有如何确定参数 *Eps* 和 *MinPts* 的问题。基本方法是观察点到它的 k 个最近邻的距离（称为 k-距离）的特性。对于属于某个簇的点，如果 k 不大于簇的大小的话，则 k-距离将很小。注意，尽管因簇的密度和点的随机分布不同而有一些变化，但是如果簇密度的差异不是很极端的话，在平均情况下变化不会太大。然而，对于不在簇中的点（如噪声点），k-距离将相对较大。因此，如果我们对于某个 k，计算所有点的 k-距离，以递增次序将它们排序，然后绘制排序后的值，则我们会看到 k-距离的急剧变化，对应于合适的 *Eps* 值。如果我们选取该距离为 *Eps* 参数，而取 k 的值为 *MinPts* 参数，则 k-距离小于 *Eps* 的点将被标记为核心点，而其他点将被标记为噪声或边界点。

图 8-22 显示了一个样本数据集，而该数据的 k-距离图在图 8-23 给出。用这种方法决定的 *Eps* 值取决于 k，但并不随 k 改变而剧烈变化。如果 k 的值太小，则少量邻近点的噪声或离群点将可能不正确地标记为簇。如果 k 的值太大，则小簇（尺寸小于 k 的簇）可能会标记为噪声。最初的

DBSCAN 算法取 $k=4$，对于大部分二维数据集，看来是一个合理的值。

图 8-22 样本数据

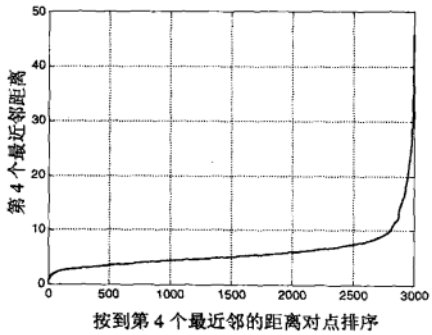


图 8-23 样本数据的 *k*-距离图

3. 变密度的簇

如果簇的密度变化很大，DBSCAN 可能会有问题。考虑图 8-24，它包含 4 个埋藏在噪声中的簇。簇和噪声区域的密度由它们的明暗度指出。较密的两个簇 *A* 和 *B* 周围的噪声的密度与簇 *C* 和 *D* 的密度相同。如果 *Eps* 阈值足够低，使得 DBSCAN 可以发现簇 *C* 和 *D*，则 *A*、*B* 和包围它们的点将变成单个簇。如果 *Eps* 阈值足够高，使得 DBSCAN 可以发现簇 *A* 和 *B*，并且将包围它们的点标记为噪声，则 *C*、*D* 和包围它们的点也将被标记为噪声。

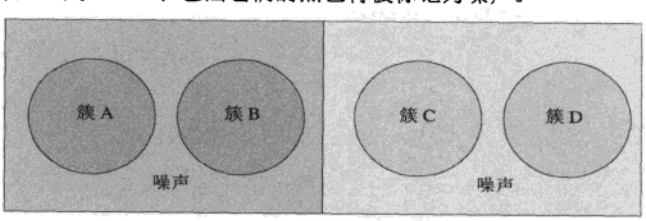


图 8-24 埋藏在噪声中的 4 个簇

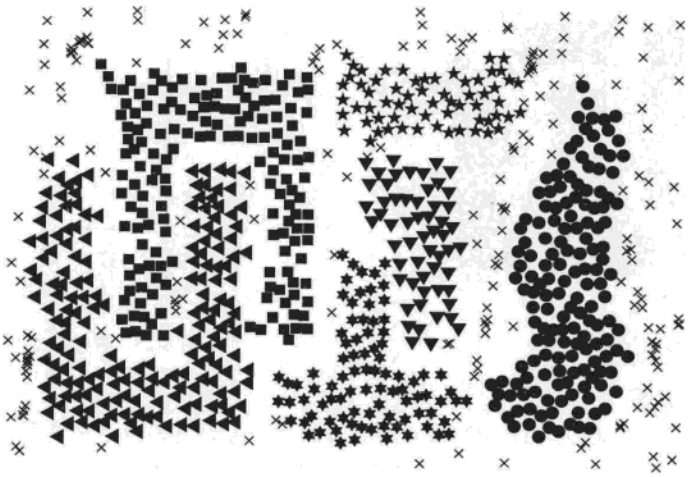
4. 例子

为了解释 DBSCAN 的使用，我们展示图 8-22 显示的相对复杂的二维数据集中发现的簇。该数据集包含 3 000 个二维点。该数据的 *Eps* 阈值通过对每个点到其第 4 个最近邻的距离排序绘图（图 8-23），并识别急剧变化处的值来确定。我们选取 $Eps=10$，对应于曲线的拐点。使用这些参数（$MinPts=4$，$Eps=10$），DBSCAN 发现的簇显示在图 8-25a 中。核心点、边界点和噪声点显示在图 8-25b 中。

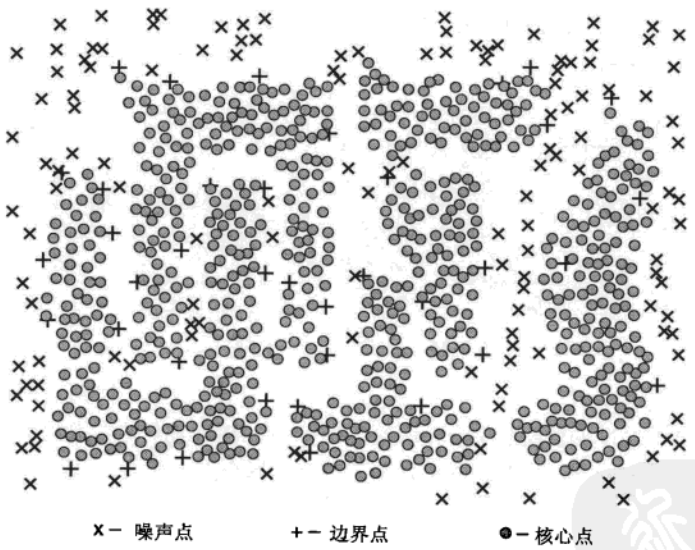
8.4.3 优点与缺点

因为 DBSCAN 使用簇的基于密度的定义，因此它是相对抗噪声的，并且能够处理任意形状和大小的簇。这样，DBSCAN 可以发现使用 K 均值不能发现的许多簇，如图 8-22 中的那些簇。然而，如前所述，当簇的密度变化太大时，DBSCAN 就会有麻烦。对于高维数据，它也有问题，因为对于这样的数据，密度定义更困难。处理这些问题的一种可行的方法在 9.4.8 节给出。最

后，当近邻计算需要计算所有的点对邻近度时（对于高维数据，常常如此），DBSCAN 的开销可能是很大的。



(a) DBSCAN 发现的簇



(b) 核心点、边界点和噪声点

图 8-25　3 000 个二维点的 DBSCAN 聚类

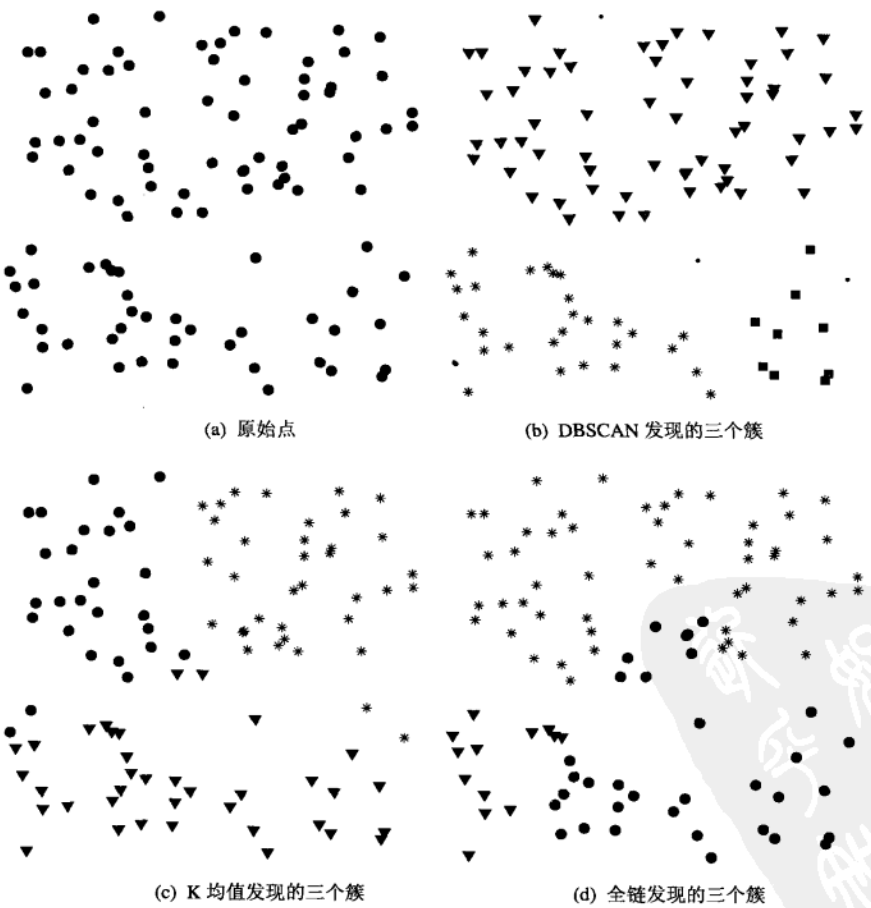
8.5　簇评估

对于监督分类，结果分类模型的评估是分类模型开发过程中必不可少的部分；并且存在广泛接受的评估度量和过程，如准确率和交叉确认。然而，由于簇的特性，簇评估技术未很好开发，或者说不是聚类分析普遍使用的。尽管如此，簇评估，或者使用更传统的称呼，**簇确认**（cluster

validation），是重要的。本节将回顾一些最常用和容易使用的方法。

对于簇评估的必要性可能存在一些疑惑。在许多情况下，聚类分析作为试探性数据分析的一部分来实施。因此，评估似乎不必要地使得一个本来是非形式化的过程复杂化。此外，由于存在大量不同的簇类型（在某种意义下，每种聚类算法都定义了自己的簇类型），似乎每种情况都可能需要一种不同的评估度量。例如，K 均值簇可能需要用 SSE 来评估；但是有些基于密度的簇不是球形的，SSE 全然不起作用。

尽管如此，簇评估应当是聚类分析的一部分。它的主要目的是，几乎每种聚类算法都会在数据集中发现簇，即便该数据集根本没有自然的簇结构。例如，考虑图 8-26，它显示了单位正方形上随机（均匀）分布的 100 个点的聚类结果。原始点显示在图 8-26a 中，而被 DBSCAN、K 均值和全链发现的簇分别显示在图 8-26b、图 8-26c 和图 8-26d 中。由于 DBSCAN 发现了三个簇（在我们通过观察第 4 个最近邻的距离设定 *Eps* 之后），我们让 K 均值和全链也去找三个簇（在图 8-26b 中，噪声用较小的标记显示）。然而，三种方法发现的簇看上去都不引人注目。在高维空间，这样的问题不容易检测到。



(a) 原始点　(b) DBSCAN 发现的三个簇

(c) K 均值发现的三个簇　(d) 全链发现的三个簇

图 8-26　100 个均匀分布的点的聚类

8.5.1　概述

能够识别数据中是否存在非随机结构正是簇确认的重要任务之一。下面列举了簇确认的一些重要问题。

(1) 确定数据集的**聚类趋势**（clustering tendency），即识别数据中是否实际存在非随机结构。

(2) 确定正确的簇个数。

(3) 不引用附加的信息，评估聚类分析结果对数据拟合情况。

(4) 将聚类分析结果与已知的客观结果（如，外部提供的类标号）比较。

(5) 比较两个簇集，确定哪个更好。

注意，第 1、2、3 项不使用任何外部信息（它们是非监督技术），而第 4 项使用外部信息。第 5 项可以用监督或非监督方式执行。第 3、4、5 项还可以进一步区分是评估整个聚类还是个别的簇？

尽管可以开发各种数值度量来评估上面提到的簇的有效性的不同方面，但是存在许多问题。首先，簇的有效性度量可能受限于它的可用范围。例如，聚类趋势度量方面的大部分工作都是针对二、三维空间数据。其次，我们需要框架来解释任意度量。对于评估簇标号与外部提供的类标号的匹配情况的度量，如果我们得到一个值 10，那么这个值表示匹配是好、可以、还是差？匹配的优良度通常可以通过考察该值的统计分布来度量，即这样的值偶然出现的几率多大。最后，如果度量太复杂，难以使用或难以理解，则很少有人愿意使用它。

用于评估簇的各方面的评估度量或指标一般分成如下三类。

- **非监督的**。聚类结构的优良性度量，不考虑外部信息。例如，SSE。簇的有效性的非监督度量常常可以进一步分成两类：**簇的凝聚性**（cluster cohesion）（紧凑性，紧致性）度量确定簇中对象如何密切相关，**簇的分离性**（cluster separation）（孤立性）度量确定某个簇不同于其他簇的地方。非监督度量通常称为**内部指标**（internal index），因为它们仅使用出现在数据集中的信息。
- **监督的**。度量聚类算法发现的聚类结构与某种外部结构的匹配程度。例如，监督指标的熵，它度量簇标号与外部提供的标号的匹配程度。监督度量通常称为**外部指标**（external index），因为它们使用了不在数据集中出现的信息。
- **相对的**。比较不同的聚类或簇。相对簇评估度量是用于比较的监督或非监督评估度量。因而，相对度量实际上不是一种单独的簇评估度量类型，而是度量的一种具体使用。例如，两个 K 均值聚类可以使用 SSE 或熵进行比较。

在本节的剩余部分，我们介绍关于簇有效性的具体内容。我们首先介绍关于非监督簇评估的主题：(1)基于凝聚性和分离性的度量，(2)两种基于邻近度矩阵的技术。由于这些方法仅用于部分簇集合，因此我们也介绍流行的共性分类相关系数。共性分类相关系数可以用于层次聚类的非监督评估，之后简略讨论找出正确的簇个数和评估聚类趋势结束非监督评估的主题。然后，我们考虑簇有效性的监督方法，如熵、纯度和 Jaccard 度量。最后，我们简略讨论如何解释（非监督或监督的）有效性度量值。

8.5.2　非监督簇评估：使用凝聚度和分离度

对于划分的聚类方案，簇有效性的许多内部度量都基于凝聚度和分离度概念。本节中，我们对基于原型和基于图的聚类技术，使用簇有效性度量来详细研究这些概念。在此过程中，我们也

将看到基于原型和基于图的聚类技术之间的一些有趣联系。

通常，将 K 个簇的集合的总体簇有效性表示成个体簇有效性的加权和：

$$overall\ validity = \sum_{i=1}^{K} w_i\ validity(C_i) \tag{8-8}$$

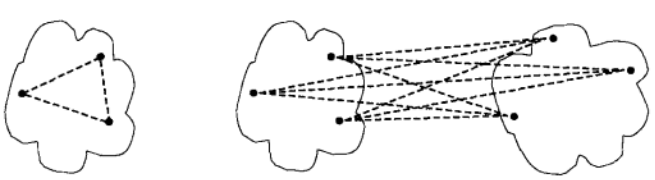
其中，*validity* 函数可以是凝聚度、分离度，或者这些量的某种组合。权值将因簇有效性度量而异。在某些情况下，权值可以简单地取 1 或者簇的大小；而在其他情况下，它们反映更复杂的性质，如凝聚度的平方根。见表 8-6。如果有效性函数是凝聚度，则值越高越好。如果是分离度，则值越低越好。

表 8-6 基于图的簇评估度量表

名称	簇 度 量	簇 权 值	类 型
$\mathcal{I}_1$	$\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_j}} proximity(\mathbf{x},\mathbf{y})$	$\frac{1}{m_i}$	基于图的凝聚度
$\mathcal{I}_2$	$\sum_{\mathbf{x}\in C_i} proximity(\mathbf{x},\mathbf{c}_i)$	1	基于原型的凝聚度
$\mathcal{E}_1$	$proximity(\mathbf{c}_i,\mathbf{c})$	m_i	基于原型的分离度
$\mathcal{G}_1$	$\sum_{j=1}^{k}\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_j}} proximity(\mathbf{x},\mathbf{y})$	$\frac{1}{\sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_i}} proximity(\mathbf{x},\mathbf{y})}$	基于图的凝聚度和分离度

1. 凝聚度和分离度的基于图的观点

对于基于图的簇，簇的凝聚度可以定义为连接簇内点的邻近度图中边的加权和。见图 8-27a。（回想一下，邻近度图以数据对象为结点，每对数据对象之间一条边，并且每条边指派一个权值，它是边所关联的两个数据对象之间的邻近度。）同样，两个簇之间的分离度可以用从一个簇的点到另一个簇的点的边的加权和来度量，如图 8-27b 所示。



(a) 凝聚度　　(b) 分离度

图 8-27 凝聚度和分离度的基于图的观点

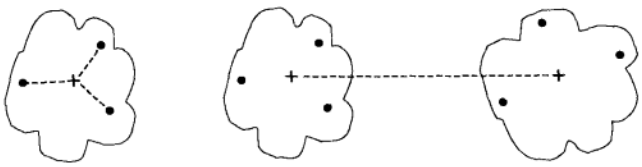
从数学上讲，基于图的簇的凝聚度和分离度可以分别用公式（8-9）和公式（8-10）表示。其中，*proximity* 函数可以是相似度、相异度，或者是这些量的简单函数。

$$cohesion(C_i) = \sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_i}} proximity(\mathbf{x},\mathbf{y}) \tag{8-9}$$

$$separation(C_i, C_j) = \sum_{\substack{\mathbf{x}\in C_i \\ \mathbf{y}\in C_j}} proximity(\mathbf{x},\mathbf{y}) \tag{8-10}$$

2. 凝聚度和分离度的基于原型的观点

对于基于原型的簇，簇的凝聚度可以定义为关于簇原型（质心或中心点）的邻近度的和。同理，两个簇之间的分离度可以用两个簇原型的邻近性度量。图8-28给出了图示，其中簇的质心用“+”标记。



(a) 凝聚度　　(b) 分离度

图 8-28　凝聚度和分离度的基于原型的观点

基于原型的凝聚度由公式（8-11）给出，而两个分离性度量分别由公式（8-12）和公式（8-13）给出，其中 $\mathbf{c}_i$ 是簇 C_i 的原型（质心），而 $\mathbf{c}$ 是总体原型（质心）。对于分离性，存在两种度量（稍后就会看到），这是因为簇原型与总原型的分离度有时与簇原型之间的分离度直接相关。注意，如果我们取邻近度为平方欧几里得距离，则公式（8-11）是簇的 SSE。

$$cohesion(C_i) = \sum_{\mathbf{x} \in C_i} proximity(\mathbf{x}, \mathbf{c}_i) \tag{8-11}$$

$$separation(C_i, C_j) = proximity(\mathbf{c}_i, \mathbf{c}_j) \tag{8-12}$$

$$separation(C_i) = proximity(\mathbf{c}_i, \mathbf{c}) \tag{8-13}$$

3. 凝聚度和分离度的总度量

前面的凝聚度和分离度定义给出了簇的有效性的简单而严格定义的度量。通过使用加权和可以将它们组合成簇的有效性的总度量，如公式（8-8）所示。然而，我们需要决定使用什么权值。毫无疑问，尽管通常权值是簇大小的某种度量，但是可用的权值变化范围很大。

表 8-6 提供了基于凝聚性和分离性的有效性度量的例子。$\mathcal{I}_1$ 用簇中对象逐对邻近度除以簇的大小来度量凝聚性。$\mathcal{I}_2$ 基于簇中对象与簇质心的邻近度之和度量凝聚性。$\mathcal{E}_1$ 是一种分离性度量，定义为簇质心与总质心的邻近度乘以簇中对象的个数。$\mathcal{G}_1$ 是一种基于凝聚性和分离性的度量，是簇中所有对象与簇外所有对象的邻近度之和（邻近度图中将簇分开必须切断的边的总权值）除以簇内对象的逐对邻近度之和。

注意，簇的有效性的任何非监督度量都可以作为聚类算法的目标函数使用，反之亦然。聚类工具箱（CLUstering TOolkit，CLUTO）（见文献注释）使用表 8-6 中的簇评估度量，以及这里未提及的其他评估度量来驱动聚类过程。它使用一种类似于 8.2.2 节讨论的增量 K 均值的算法。具体地说，每个点指派到产生最优簇评估函数值的簇。簇评估度量 $\mathcal{I}_2$ 对应于传统的 K 均值，并产生具有较好 SSE 值的簇。其他度量产生的簇关于 SSE 不好，但是关于特定的簇有效性度量更优。

4. 基于原型的凝聚度和基于图的凝聚度之间的联系

尽管度量簇的凝聚性和分离性的基于图的方法与基于原型的方法看上去截然不同，但是对于某些邻近性度量它们是等价的。例如，对于 SSE 和欧几里得空间的点，可以证明（公式（8-14））簇中逐对点的平均距离等于簇的 SSE（见本章习题 27）。

$$\text{Cluster SSE} = \sum_{\mathbf{x}\in C_i} dist(\mathbf{c}_i, \mathbf{x})^2 = \frac{1}{2m_i}\sum_{\mathbf{x}\in C_i}\sum_{\mathbf{y}\in C_i} dist(\mathbf{x}, \mathbf{y})^2 \tag{8-14}$$

5. 两种基于原型的分离性度量方法

当邻近度用欧几里得距离度量时，簇之间分离性的传统度量是组平方和（SSB），即簇质心 $\mathbf{c}_i$ 到所有数据点的总均值 $\mathbf{c}$ 的距离的平方和。通过在所有簇上对 SSB 求和，我们得到总 SSB，由公式（8-15）给出，其中 $\mathbf{c}_i$ 是第 i 个簇的均值，而 $\mathbf{c}$ 是总均值。总 SSB 越高，簇之间的分离性越好。

$$\text{总 SSB} = \sum_{i=1}^{K} m_i dist(\mathbf{c}_i, \mathbf{c})^2 \tag{8-15}$$

可以直接证明，总 SSB 与质心之间的逐对距离有直接关系。特殊地，如果簇的大小相等，即 $m_i = m/K$，则该关系取公式（8-16）给出的简单形式（见本章习题 28）。正是这类等价性诱导了公式（8-12）和公式（8-13）的原型分离度定义。

$$\text{总 SSB} = \frac{1}{2K}\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{m}{K} dist(\mathbf{c}_i, \mathbf{c}_j)^2 \tag{8-16}$$

6. 凝聚度和分离度之间的联系

在某些情况下，凝聚度和分离度之间也存在很强的联系。具体地说，可以证明总 SSE 和总 SSB 之和是一个常数，它等于总平方和（TSS）——每个点到数据的总均值的距离的平方和。这个结果的重要性在于：最小化 SSE（凝聚度）等价于最大化 SSB（分离度）。

下面，我们给出该事实的证明，该证明所用的方法也适用于证明前两节陈述的联系。为了简化证明过程，我们假定数据是一维的，即 $dist(x, y) = (x-y)^2$。证明中还使用了交叉项 $\sum_{i=1}^{K}\sum_{x\in C_i}(x-c_i)(c-c_i)$ 为 0 的事实（见本章习题 29）。

$$\begin{aligned}
\text{TSS} &= \sum_{i=1}^{K}\sum_{x\in C_i}(x-c)^2 \\
&= \sum_{i=1}^{K}\sum_{x\in C_i}\left((x-c_i)-(c-c_i)\right)^2 \\
&= \sum_{i=1}^{K}\sum_{x\in C_i}(x-c_i)^2 - 2\sum_{i=1}^{K}\sum_{x\in C_i}(x-c_i)(c-c_i) + \sum_{i=1}^{K}\sum_{x\in C_i}(c-c_i)^2 \\
&= \sum_{i=1}^{K}\sum_{x\in C_i}(x-c_i)^2 + \sum_{i=1}^{K}\sum_{x\in C_i}(c-c_i)^2 \\
&= \sum_{i=1}^{K}\sum_{x\in C_i}(x-c_i)^2 + \sum_{i=1}^{K}|C_i|(c-c_i)^2 \\
&= \text{SSE+SSB}
\end{aligned}$$

7. 评估个体簇和对象

迄今为止，我们一直关注使用凝聚度和分离度对一组簇进行总评估。许多簇的有效性度量也能用来评估个体簇和对象。例如，我们可以根据簇的有效性（即凝聚度和分离度）的具体值确定个体簇的秩。可以认为具有较高凝聚度值的簇比具有较低凝聚度值的簇好。这种信息通常可以用来提高聚类的质量。例如，如果簇凝聚性不好，则我们可能希望将它分裂成若干个子簇。另一方

面，如果两个簇相对凝聚，但分离性不好，则我们可能需要将它们合并成一个簇。

我们也可以使用对象对簇的总凝聚度或分离度的贡献，来评估簇中对象。对凝聚度和分离度贡献越大的对象就越靠近簇的“内部”，反之，对象可能离簇的“边缘”很近。下一节我们考虑一种簇评估度量，它使用基于这些思想的方法来评估点、簇和整个簇集合。

8. 轮廓系数

流行的轮廓系数（silhouette coefficient）方法结合了凝聚度和分离度。下面的步骤解释如何计算个体点的轮廓系数。此过程由如下三步组成。我们使用距离，但是类似的方法可以使用相似度。

(1) 对于第 i 个对象，计算它到簇中所有其他对象的平均距离。该值记作 a_i。

(2) 对于第 i 个对象和不包含该对象的任意簇，计算该对象到给定簇中所有对象的平均距离。关于所有的簇，找出最小值；该值记作 b_i。

(3) 对于第 i 个对象，轮廓系数是 $s_i = (b_i - a_i)/\max(a_i, b_i)$。

轮廓系数的值在–1 和 1 之间变化。我们不希望出现负值，因为负值表示点到簇内点的平均距离 a_i 大于点到其他簇的最小平均距离 b_i。我们希望轮廓系数是正的（$a_i < b_i$），并且 a_i 越接近 0 越好，因为当 $a_i = 0$ 时轮廓系数取其最大值 1。

我们可以简单地取簇中点的轮廓系数的平均值，计算簇的平均轮廓系数。通过计算所有点的平均轮廓系数，可以得到聚类优良性的总度量。

例 8.8　轮廓系数　图 8-29 显示了 10 个簇中点的轮廓系数图。较黑的阴影指示较小的轮廓系数。 □

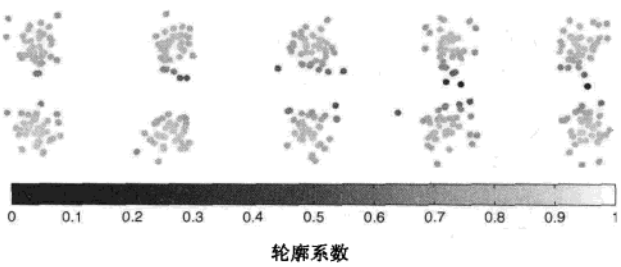


图 8-29　10 个簇中点的轮廓系数

8.5.3　非监督簇评估：使用邻近度矩阵

本节我们考察两种基于邻近度矩阵评估簇的有效性的非监督方法。第一种比较实际的邻近度矩阵和理想的邻近度矩阵，而第二种使用可视化技术。

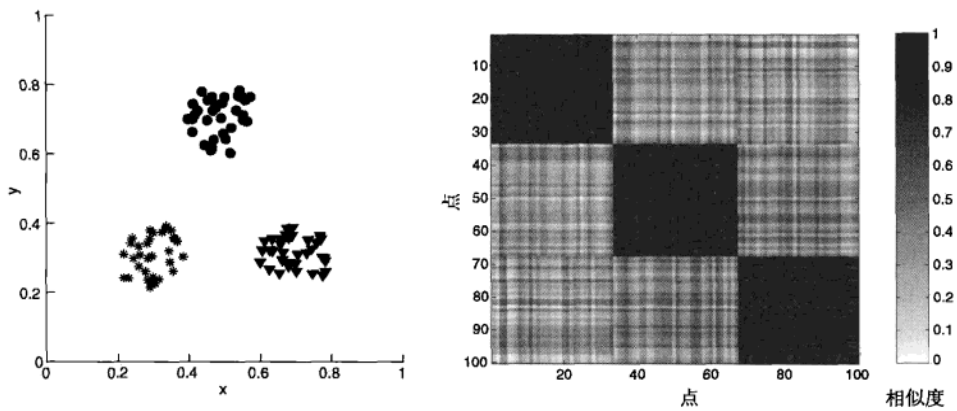
1. 通过相关性度量簇的有效性

如果给定数据集的相似度矩阵和数据集聚类分析得到的簇标号，则我们可以通过考察相似度矩阵和基于簇标号的相似度矩阵的理想版本之间的相关性来评估聚类的“优良性”。（稍加修改，下面内容也可用于邻近度矩阵，但是简单起见，我们只讨论相似度矩阵。）更具体地说，理想的簇是这样的簇，它的点与簇内所有点的相似度为 1，而与其他簇中的所有点的相似度为 0。这样，如果我们将相似度矩阵的行和列排序，使得属于相同簇的对象在一起，则理想的相似度矩阵具有

块对角（block diagonal）结构。换言之，在相似度矩阵中代表簇内相似度的项的块内部相似度非零（为 1），而其他地方为 0。理想的相似度矩阵可以通过如下方法构造：创建一个矩阵，每个数据点一行一列（与实际的相似度矩阵类似），矩阵的一个项为 1，如果它所关联的一对点属于同一个簇。其他项均为 0。

理想和实际相似度矩阵之间高度相关表明属于同一个簇的点相互之间很接近，而低相关性表明相反情况。（由于实际和理想相似度矩阵都是对称的，因此只需要对矩阵对角线下方或上方的 $n(n-1)/2$ 个项计算相关度。）对于许多基于密度和基于近邻的簇，这不是好的度量，因为它们不是球形的，并且常常与其他簇紧密地盘绕在一起。

例 8.9 实际的和理想的相似度矩阵 为了解释这种度量，我们对图8-26c（随机数据）和图8-30a（具有3个明显分离簇的数据）显示的 K 均值簇，计算理想和实际相似度矩阵之间的相关度。相关度分别为0.581 0和0.923 5，反映了期望的结果——K 均值在随机数据上发现的簇比在具有明显分离的簇的数据上发现的簇差。 □



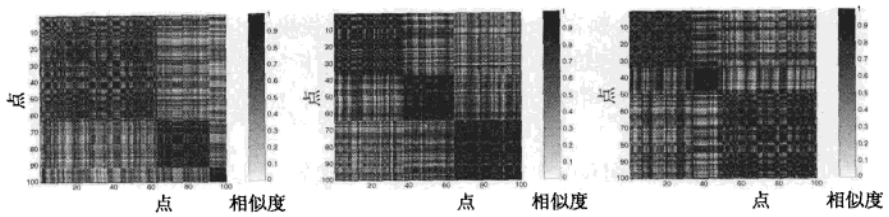
(a) 明显分离的簇　　(b) 按 K 均值簇标号排序的相似度矩阵

图 8-30 明显分离的簇的相似度矩阵

2. 通过相似度矩阵可视地评价聚类

前面的技术使人联想起一种评价簇集合的一般的、定性的方法：按照簇标号调整相似度矩阵的行列次序，然后画出它。从理论上讲，如果有明显分离的簇，则相似度矩阵应当粗略地是块对角的。如果不是，则相似度矩阵所显示的模式可能揭示了簇之间的联系。所有这些也可以用于相异度矩阵，但为了简单起见，我们只讨论相似度矩阵。

例 8.10 可视化相似度矩阵 考虑图 8-30a 中的点，它们形成 3 个明显分离的簇。如果我们使用 K 均值将这些点划分成 3 个簇，则我们应当不成问题地发现这 3 个簇，因为它们是明显分离的。这些簇的分离性由图 8-30b 显示的重新排序的相似度矩阵图示。（为了一致，我们使用公式 $s=1-(d-min\_d)/(max\_d-min\_d)$ 将距离变换成相似度。）图 8-31 显示了 DBSCAN、K 均值和全链在图 8-26 的随机数据集中发现的簇的相似度矩阵。



(a) 按 DBSCAN 簇标号排序的相似度矩阵　(b) 按 K 均值簇标号排序的相似度矩阵　(c) 按全链簇标号排序的相似度矩阵

图 8-31　随机数据的簇的相似度矩阵

图 8-30 中明显分离的簇在重新排序的相似度矩阵中显示了很强的块对角模式。然而，在随机数据上，DBSCAN、K 均值和全链发现的簇的重新排序的相似度矩阵中也存在弱块对角模式（见图 8-31）。正如人可以发现云中的模式一样，数据挖掘算法也可以发现随机数据中的簇。尽管发现云中的模式是一种娱乐，但是发现噪声中的簇毫无意义，并且可能会妨碍我们的工作。　□

对于大型数据集，该方法看来毫无希望，开销太大，因为相似度计算需要 $O(m^2)$时间，其中 m 是对象个数。但是，使用抽样，该方法仍然可以使用。我们可以从每个簇抽取数据点样本，计算这些点之间的相似度，然后对结果绘图。可能需要对小簇多抽样，对大簇少抽样，以得到所有簇的足够代表。

8.5.4　层次聚类的非监督评估

前面的簇评估方法是为划分聚类设计的。这里，我们讨论一种用于层次聚类的流行的评估度量——共性分类相关。两个对象之间的**共性分类距离**（cophenetic distance）是凝聚层次聚类技术首次将对象放在同一个簇时的邻近度。例如，如果在凝聚层次聚类进程的某个时刻，两个合并的簇之间的最小距离是 0.1，则一个簇中的所有点关于另一个簇中各点的共性分类距离都是 0.1。在共性分类距离矩阵中，项是每对对象之间的共性分类距离。点集的每个层次聚类的共性分类距离不同。

例 8.11　共性分类距离矩阵　表 8-7 显示了图 8-16 显示的单链聚类的共性分类距离矩阵。（图 8-16 中的数据由表 8-3 给出的 6 个二维点组成。）

表 8-7　单链和表 8-3 中数据的共性分类距离矩阵

点	p1	p2	p3	p4	p5	p6
p1	0	0.222	0.222	0.222	0.222	0.222
p2	0.222	0	0.148	0.151	0.139	0.148
p3	0.222	0.148	0	0.151	0.148	0.110
p4	0.222	0.151	0.151	0	0.151	0.151
p5	0.222	0.139	0.148	0.151	0	0.148
p6	0.222	0.148	0.110	0.151	0.148	0

□

共性分类相关系数（cophenetic correlation coefficient，CPCC）是该矩阵与原来的相异度矩阵的项之间的相关度，是（特定类型的）层次聚类对数据拟合程度的标准度量。该度量的最常见应用是评估对于特定的数据类型，哪种类型的层次聚类最好。

例 8.12 共性分类相关系数 我们对图 8-16～图 8-19 显示的层次聚类计算 CPCC。这些值在表 8-8 中。由单链技术产生的层次聚类看来不如由全链、组平均和 Ward 方法产生的聚类。

表 8-8 表 8-3 中的数据和 4 种凝聚层次聚类技术的共性分类相关系数

技 术	CPCC
单链	0.44
全链	0.63
组平均	0.66
Ward 方法	0.64

□

8.5.5 确定正确的簇个数

多种非监督簇评估度量都可以用来近似地确定正确的或自然的簇个数。

例 8.13 簇的个数 图 8-29 的数据集有 10 个自然簇。图 8-32 显示了该数据集的（二分）K 均值聚类发现的簇个数的 SSE 曲线，而图 8-33 显示了相同数据的簇个数的平均轮廓系数曲线。当簇个数等于 10 时，SSE 有一个明显的拐点，而轮廓系数有一个明显的尖峰。 □

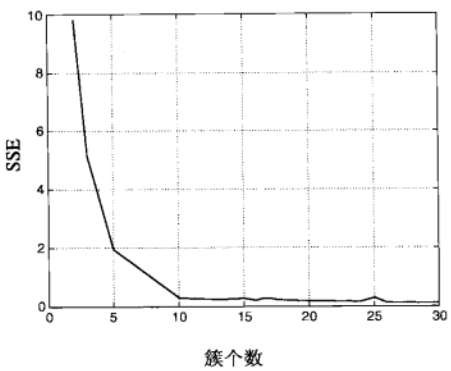


图 8-32 图 8-29 的数据簇个数的 SSE 曲线

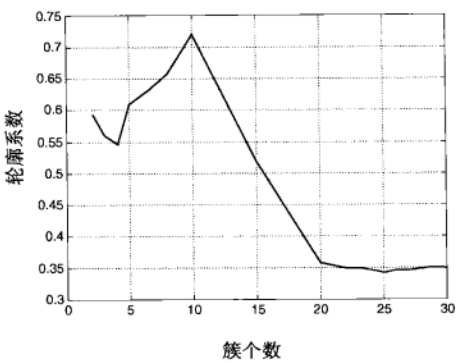


图 8-33 图 8-29 的数据簇个数的平均轮廓系数曲线

这样，我们可以尝试通过寻找簇个数的评估度量曲线图中的拐点、尖峰或下降点发现簇的自然个数。当然，这种方法并不总是有效的。与图 8-29 相比，簇可能盘绕得或交叠得更厉害。此外，数据中也可能包含嵌套的簇。事实上，图 8-29 中的簇也有点嵌套，即有 5 对簇，因为上下的簇比左右的簇更靠近。SSE 曲线有一个拐点，指明了这一点，但是轮廓系数曲线没有这么清楚。总而言之，尽管需要小心，刚才讨论的技术还是可以帮助我们洞察数据中簇的个数。

8.5.6 聚类趋势

确定数据集中是否包含簇的一种显而易见的方法是试着对它聚类。然而，给定数据集，几乎所有的聚类算法都责无旁贷地发现簇。为了处理这一问题，我们可以评估结果簇，至少有些簇具有好的质量，才能说数据集包含簇。然而，事实是数据集中可能存在不同于我们的聚类算法所能发现的簇类型。如果出现这种情况，该方法就不能处理。为了处理这样的问题，我们可以使用多

种算法，并评估结果簇的质量。如果簇都很差，则可能表明数据中确实没有簇。

换一种方式，我们可以关注聚类趋势度量——试图评估数据集中是否包含簇，而不进行聚类。最常用的方法（特别是对于欧几里得空间数据）是使用统计检验来检验空间随机性。然而，选择正确的模型、估计参数、评估数据是非随机的假设的统计数据，这一切可能非常具有挑战性。尽管如此，人们已经开发了许多方法，其中大部分都是针对低维欧几里得空间中的点。

例 8.14　Hopkins 统计量　对于该方法，我们产生 p 个随机地分布在数据空间上的点，并且也抽取 p 个**实际**数据点。对于这两个点集，我们找出每个点到原数据集的最近邻距离。设 u_i 是人工产生的点的最近邻距离，而 w_i 是样本点到原数据集的最近邻距离。Hopkins（霍普金斯）统计量 H 由公式（8-17）定义：

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i} \tag{8-17}$$

如果随机产生的点与样本点具有大致相同的最近邻距离，则 H 将在 0.5 左右。H 值接近 0 或 1 分别表明数据是高度聚类的和数据在数据空间是有规律分布的。为了举例说明，对于 $p = 20$ 和 100 的不同实验，我们计算了图 8-26 中数据的 Hopkins 统计量。H 的平均值为 0.56，标准差为 0.03。对图 8-30 中明显分离的数据点做相同实验。H 的平均值为 0.95，标准差为 0.006。　□

8.5.7　簇有效性的监督度量

当我们有关于数据的外部信息时，通常是从外部导出的数据对象的类标号形式。在这种情况下，惯常的做法是度量簇标号与类标号的对应程度。但是，这样做的目的是什么？归根结底，如果我们有了类标号，进行聚类分析的目的何在？这种分析的动机是比较聚类技术与“基本事实”，或评估人工分类过程可以在多大程度上被聚类分析自动地实现。

考虑两类不同的方法。第一组技术使用分类的度量，如熵、纯度和 F 度量。这些度量评估簇包含单个类的对象的程度。第二组方法涉及二元数据的相似性度量，如我们在第 2 章介绍的 Jaccard 度量。这些方法评估度量的程度，同一个类的两个对象在同一个簇中，或相反。为方便起见，我们分别称这两类度量为**面向分类的**（classification-oriented）和**面向相似性的**（similarity-oriented）。

1. 簇有效性的面向分类的度量

有许多度量（如熵、纯度、精度、召回率和 F 度量）普遍用来评估分类模型的性能。对于分类，我们度量预测的类标号与实际类标号的对应程度，但是对于上面提到的度量，通过使用簇标号而不是预测的类标号，不需要做重大改变。下面，我们简略地回顾这些度量的定义，这些曾在第 4 章讨论过。

- **熵**：每个簇由单个类的对象组成的程度。对于每个簇，首先计算数据的类分布，即对于簇 i，计算簇 i 的成员属于类 j 的概率 $p_{ij} = m_{ij}/m_i$，其中 m_i 是簇 i 中对象的个数，而 m_{ij} 是簇 i 中类 j 的对象个数。使用类分布，用标准公式 $e_i = -\sum_{j=1}^{L} p_{ij} \log_2 p_{ij}$ 计算每个簇 i 的熵，其中 L 是类的个数。簇集合的总熵用每个簇的熵的加权和计算，即 $e = \sum_{i=1}^{K} \frac{m_i}{m} e_i$，其中 K 是簇的个数，而 m 是数据点的总数。

- **纯度**：簇包含单个类的对象的另一种度量程序。使用前面的术语，簇 i 的纯度是 $p_i = \max_j p_{ij}$，而聚类的总纯度是 $purity = \sum_{i=1}^{K} \frac{m_i}{m} p_i$。
- **精度**：簇中一个特定类的对象所占的比例。簇 i 关于类 j 的精度是 $precision(i, j) = p_{ij}$。
- **召回率**：簇包含一个特定类的所有对象的程度。簇 i 关于类 j 的召回率是 $recall(i, j) = m_{ij}/m_j$，其中 m_j 是类 j 的对象个数。
- **F 度量**：精度和召回率的组合，度量在多大程度上，簇只包含一个特定类的对象和包含该类的所有对象。簇 i 关于类 j 的 F 度量是 $F(i, j) = (2 \times precision(i, j) \times recall(i, j))/(precision(i, j) + recall(i, j))$。

例 8.15 监督评估度量 我们提供一个例子解释这些度量。具体地说，我们以余弦相似性度量使用 K 均值，对取自《洛杉矶时报》的3 204篇报道文章进行聚类。这些文章取自6个不同的类：娱乐、财经、国外、都市、国内和体育。表8-9显示了 K 均值聚类发现6个簇的结果。第一列指示簇，接下来的六列形成混淆矩阵，即这些列指出每个类的文档在这些簇中如何分布。最后两列分别是熵和纯度。

表 8-9 《洛杉矶时报》文档数据集 K 均值聚类结果

簇	娱乐	财经	国外	都市	国内	体育	熵	纯度
1	3	5	40	506	96	27	1.227 0	0.747 4
2	4	7	280	29	39	2	1.147 2	0.775 6
3	1	1	1	7	4	671	0.181 3	0.979 6
4	10	162	3	119	73	2	1.748 7	0.439 0
5	331	22	5	70	13	23	1.397 6	0.713 4
6	5	358	12	212	48	13	1.552 3	0.552 5
合计	354	555	341	943	273	738	1.145 0	0.720 3

理想情况下，每个簇仅包含来自一个类的文档。事实上，每个簇包含来自多个类的文档。尽管如此，许多簇包含的文档主要来自一个类。具体地说，簇 3 包含的文档大部分来自体育版，纯度和熵都异常好。其他簇的纯度和熵没有这么好，但是如果数据被划分成更多的簇，则可能大幅度提高。

可以对每个簇计算精度、召回率和 F 度量。为了给出一个具体的例子，考虑表 8-9 的簇 1 和都市类。精度是 506/677 = 0.75，召回率是 506/943 =0.26，因而 F 值是 0.39。相比之下，簇 3 和体育的 F 值是 0.94。 □

2. 簇有效性的面向相似性的度量

我们本节讨论的度量都基于这样一个前提：同一个簇的任意两个对象也应当在同一个类，反之亦然。我们可以把这种簇有效性方法看作涉及两个矩阵的比较：(1)前面讨论过的**理想的簇相似度矩阵**，其第 ij 项为 1，如果两个对象 i 和 j 在同一个簇，否则为 0。(2)关于类标号定义的**理想的类相似度矩阵**（ideal class similarity matrix），其第 ij 项为 1，如果两个对象 i 和 j 在同一个类，否则为 0。与前面一样，我们可以取这些矩阵的相关度作为簇有效性的度量。在聚类确认文献中，该度量称作Γ统计量。

例 8.16 簇和类矩阵之间的相关性 为了更具体地解释这一思想，我们给出一个例子，涉及5个数据点 p_1、p_2、p_3、p_4、p_5，2个簇 $C_1 = \{p_1, p_2, p_3\}$、$C_2 = \{p_4, p_5\}$，以及2个类 $L_1 = \{p_1, p_2\}$，

$L_2 = \{p_3, p_4, p_5\}$。理想的簇和类相似度矩阵分别在表8-10和表8-11中给出。这两个矩阵项之间的相关度是0.359。 □

表 8-10　理想的簇相似度矩阵

点	p_1	p_2	p_3	p_4	p_5
p_1	1	1	1	0	0
p_2	1	1	1	0	0
p_3	1	1	1	0	0
p_4	0	0	0	1	1
p_5	0	0	0	1	1

表 8-11　理想的类相似度矩阵

点	p_1	p_2	p_3	p_4	p_5
p_1	1	1	0	0	0
p_2	1	1	0	0	0
p_3	0	0	1	1	1
p_4	0	0	1	1	1
p_5	0	0	1	1	1

更一般地，可以使用我们在 2.4.5 节看到的任何二元相似性度量。（例如，我们可以将这两个矩阵转换成二元向量。）我们重述用于定义这些相似性度量的 4 个量，但是稍加修改，以适合当前情况。具体地说，我们需要对所有的不同对象对，计算如下 4 个量。（如果 m 是对象的个数，则这样的对象对有 $m(m-1)/2$ 个。）

f_{00} =具有不同的类和不同的簇的对象对的个数
f_{01} =具有不同的类和相同的簇的对象对的个数
f_{10} =具有相同的类和不同的簇的对象对的个数
f_{11} =具有相同的类和相同的簇的对象对的个数

特殊地，在这种情况下，称作 Rand 统计量的简单匹配系数和 Jaccard 系数是两种最常使用的簇有效性度量。

$$\text{Rand 统计量} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \tag{8-18}$$

$$\text{Jaccard 系数} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \tag{8-19}$$

例 8.17　Rand 和 Jaccard 度量　根据这些公式，我们可以立即计算基于表 8-10 和表 8-11 的例子的 Rand 统计量和 Jaccard 系数。注意，$f_{00} = 4$，$f_{01} = 2$，$f_{10} = 2$，$f_{11} = 2$，Rand 统计量 $= (2 + 4)/10 = 0.6$，而 Jaccard 系数 $= 2/(2 + 2 + 2) = 0.33$。 □

还要注意，这 4 个量 f_{00}、f_{01}、f_{10} 和 f_{11} 定义了相依表，如表 8-12 所示。

表 8-12　确定对象对是否在相同的类和相同的簇的二路相依表

	相同的簇	不同的簇
相同的类	f_{11}	f_{10}
不同的类	f_{01}	f_{00}

前面，在关联分析的背景下（见 6.7.1 节），我们广泛地讨论了关联度量，可以用于这类相依表。（比较表 8-12 和表 6-7。）这些度量也能用于簇有效性。

3. 层次聚类的簇有效性

本节迄今为止，我们仅对划分聚类讨论了簇有效性的监督度量。由于各种原因（包括先前存在的层次结构常常不再存在），层次聚类的监督评估更加困难。这里，我们给出一个根据类标号集评估层次聚类方法的例子。类标号集可能比先前存在的簇结构更容易得到。

该方法的关键思想是，评估层次聚类是否对于每个类，至少有一个簇相对较纯，并且包含了该类的大部分对象。为了根据此目标评估层次聚类，我们对每个类，计算簇层次结构中每个簇的F度量。对于每个类，取最大的F度量。最后，通过计算每类的F度量的加权平均，计算层次聚类的总F度量，其中，权值基于类的大小。该层次F度量的形式上的定义如下：

$$F = \sum_j \frac{m_j}{m} \max_i F(i, j)$$

其中，最大值在所有层的所有簇 i 上取，m_j 是类 j 中对象的个数，而 m 是对象的总数。

8.5.8 评估簇有效性度量的显著性

簇有效性度量旨在帮助我们度量所得到的簇的优良性，通常用单个的数字作为这种优良性的度量。然而，我们也因此面临解释该数显著性的问题——一项可能更加困难的任务。

在许多情况下，簇评估度量的最小和最大值可能提供某种指导。例如，如果我们相信我们的类标号，并且希望我们的簇结构反映类结构，则根据定义，纯度0是坏的，而纯度1是好的。同理，与SSE为0一样，熵为0是好的。

然而，有时可能没有最小或最大值，或者数据的尺度也可能影响解释。此外，即使存在具有明显解释的最小和最大值，中间值也需要解释。在某些情况下，我们可以使用绝对标准。例如，如果我们为了实用而进行聚类，则在用质心近似对象点时，我们或许只能容忍一定程度的误差。

但是，如果不是这种情况，则我们必须做一些别的事情。一种常用的方法是用统计学术语解释有效性度量值。具体地说，我们试图确定观测值随机得到的可能性有多大。值是好的，如果它是不寻常的。即它不像是随机结果。这种方法的动机是，我们只对反映数据中非随机结构的簇感兴趣，并且这样的结构应当产生异常高（低）的簇有效度量值，至少在有效性度量旨在反映强簇结构的存在性时应当如此。

例8.18 SSE的显著性 我们用一个基于K均值和SSE的例子加以解释。假定我们想要度量相对于随机数据，图8-30中明显分离的簇为什么好。我们产生多个100个点的随机数据集，它们与三个簇中的点具有相同的值域；使用K均值在每个数据集找出三个簇，然后收集这些聚类的SSE分布。使用这一分布，我们可以估计原来簇的SSE值的概率。图8-34显示了500次随机运行的SSE的直方图。图8-34显示的最低SSE是0.017 3，对于图8-30的三个簇，SSE是0.005 0。因此，我们可以保守地认为，像图8-30那样的聚类随机出现的可能性不超过1%。 □

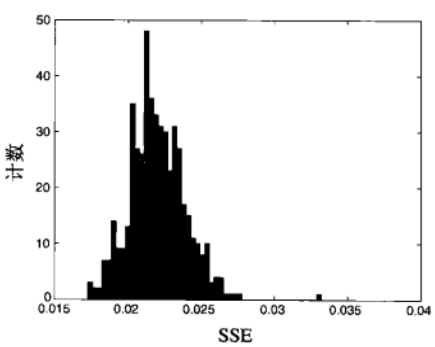


图8-34 500个随机数据集的SSE的直方图

最后，我们强调簇评估（监督或非监督）绝对不仅仅是为了得到簇有效性的数值度量。除非根据度量的定义，该值具有自然的解释，否则我们需要以某种方法解释它。如果我们的簇评估度量定义为较低的值预示较强的簇，当我们知道评估度量的分布时，就可以使用统计学方法评估我们得到的值是否异乎寻常地低。我们提供了如何发现这种分布的例子，但是该主题远非如此简单。为得到更多要点，我们建议读者阅读文献注释。

即使评估度量用作相对度量（即比较两个聚类），我们仍然需要评估比较两个聚类评估度量之间差别的显著性。尽管一个值几乎总是比另一个好，确定差别是否显著可能仍然是困难的。注意，这种显著性有两个方面：差别是否是统计显著的（可重复的），差别的量级对于应用是否有意义。许多人都不认为1%的差别是显著的，尽管它是相容可再产生的。

文献注释

本章的讨论深受Jain和Dubes[396]、Anderberg[374]、Kaufman和Rousseeuw[400]所著的簇分析书籍的影响。其他令人感兴趣的聚类书包括Aldenderfer和Blashfield[373]、Everitt等[388]、Hartigan[394]、Mirkin[405]、Murtagh[407]、Romesburg[409]、Späth [413]。更面向统计学的聚类方法的介绍在Duda等[385]的模式识别、Mitchell[406]的机器学习和Hastie等[395]的统计学习书中。Jain等[397]给出了聚类的一般综述，而Han等[393]提供了空间数据挖掘技术的综述。Behrkin [379]提供了数据挖掘聚类技术的综述。一个很好的数据挖掘领域之外的聚类参考文献源是Arabie和Hubert[376]的文章。Kleinberg[401]讨论了聚类算法的一些折中，并且证明一个聚类算法不可能同时具有三个简单性质。

K均值算法具有很长的历史，但是仍然是当前研究的课题。最早的K均值算法由MacQueen [403]提出。Ball和Hall[377]的ISODATA算法是K均值算法早期的复杂版本，它使用了各种预处理和后处理技术来改进基本算法。Anderberg[374]、Jain和Dubes[396]的书详细地介绍了K均值算法和它的一些变形。Steinbach等[414]的文章介绍了本章讨论的二分K均值算法，该算法与其他聚类方法的实现放在Karypis[382]创建的CLUTO（CLUstering TOolkit，聚类工具箱）软件包中，向学术研究免费提供。Boley[380]创建了基于发现数据主方向（主成分）的划分聚类算法（PDDP），而Savaresi和Boley[411]考察了它与二分K均值之间的关系。K均值的最新变形是K均值的新的增量版本（Dhillon等[383]）、X均值（Pelleg和Moore[408]）、K调和均值（Zhang等[416]）。Hamerly和Elkan[392]讨论了某些聚类算法，它们产生比K均值更好的结果。尽管前面提到的一些方法以某种方式处理了K均值的初始化问题，但是改进K均值初始化的其他方法也可以在Bradley和Fayyad[381]的工作中找到。Dhillon和Modha[384]提供了K均值的一般化，称作球形K均值，使用常见的相似度函数。Banerjee等[378]构建了使用基于Bregman散度的相异度函数的K均值聚类的一般框架。

层次聚类技术也有很长的历史。早期的活动大部分在分类学领域，Jardine和Sibson[398]、Sneath和Sokal[412]的书包含了这些研究。层次聚类的一般性讨论也可以在上面提到的大部分聚类书籍中找到。尽管层次聚类的大部分工作都关注凝聚层次聚类，但是分裂的层次聚类也受到一些关注。例如，Zahn[415]介绍了一种使用图的最小生成树的分裂的层次聚类技术。凝聚和分裂方法通常都将合并（分裂）作为最终决定，Fisher[389]、Karypis等[399]的一些工作试图突破这些限制。

Ester等提出了DBSCAN[387]，后来被Sander等[410]拓广成GDBSCAN，以处理更一般的数据

类型和距离度量，如邻近性用相交度来度量的多边形。Kriegel等[386]开发了DBSCAN的增量版本。DBSCAN的一个有趣派生物是OPTICS（Ordering Points To Identify the Clustering Structure，对点排序以识别聚类结构）（Ankerst等[375]），它使得簇结构可视化，并且也可以用于层次聚类。

Jain 和 Dubes 的聚类书[396]的第 4 章提供了簇有效性的权威讨论，对本章的讨论具有重要影响。聚类有效性的最近综述见 Halkidi 等[390, 391]和 Milligan[404]。Kaufman 和 Rousseeuw 的聚类书[400]介绍了轮廓系数。表 8-6 中的凝聚度与分离度度量源于 Zhao 和 Karypis[417]的文章，该文还包含了熵、纯度和层次 F 度量的讨论。层次 F 度量源于 Larsen 和 Aone[402]的文章。

参考文献

[373] M. S. Aldenderfer and R. K. Blashfield. *Cluster Analysis*. Sage Publications, Los Angeles, 1985.

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

[375] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering Points To Identify the Clustering Structure. In *Proc. of 1999 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 49 - 60, Philadelphia, Pennsylvania, June 1999. ACM Press.

[376] P. Arabie, L. Hubert, and G. D. Soete. An overview of combinatorial data analysis. In P. Arabie, L. Hubert, and G. D. Soete, editors, *Clustering and Classification*, pages 188 - 217. World Scientific, Singapore, January 1996.

[377] G. Ball and D. Hall. A Clustering Technique for Summarizing Multivariate Data. *Behavior Science*, 12:153 - 155, March 1967.

[378] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. In *Proc. of the 2004 SIAM Intl. Conf. on Data Mining*, pages 234 - 245, Lake Buena Vista, FL, April 2004.

[379] P. Berkhin. Survey Of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[380] D. Boley. Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*, 2(4):325 - 344, 1998.

[381] P. S. Bradley and U. M. Fayyad. Refining Initial Points for K-Means Clustering. In *Proc. of the 15th Intl. Conf. on Machine Learning*, pages 91 - 99, Madison, WI, July 1998. Morgan Kaufmann Publishers Inc.

[382] CLUTO 2.1.1: Software for Clustering High-Dimensional Datasets. /www.cs.umn.edu/ ~karypis, November 2003.

[383] I. S. Dhillon, Y. Guan, and J. Kogan. Iterative Clustering of High Dimensional Text Data Augmented by Local Search. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 131 - 138. IEEE Computer Society, 2002.

[384] I. S. Dhillon and D. S. Modha. Concept Decompositions for Large Sparse Text Data Using Clustering. *Machine Learning*, 42(1/2):143 - 175, 2001.

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

[386] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental Clustering for Mining in a Data Warehousing Environment. In *Proc. of the 24th VLDB Conf.*, pages 323 - 333, New York City, August 1998. Morgan Kaufmann.

[387] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 226 - 231, Portland, Oregon, August 1996. AAAI Press.

[388] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Arnold Publishers, London, fourth edition, May 2001.

[389] D. Fisher. Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research*, 4:147 - 179, 1996.

[390] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: part I. *SIGMOD Record*

(ACM Special Interest Group on Management of Data), 31(2):40 - 45, June 2002.

[391] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity checking methods: part II. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 31 (3):19 - 27, Sept. 2002.

[392] G. Hamerly and C. Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proc. of the 11th Intl. Conf. on Information and Knowledge Management*, pages 600 - 607, McLean, Virginia, 2002. ACM Press.

[393] J. Han, M. Kamber, and A. Tung. Spatial Clustering Methods in Data Mining: A review. In H. J. Miller and J. Han, editors, *Geographic Data Mining and Knowledge Discovery*, pages 188 - 217. Taylor and Francis, London, December 2001.

[394] J. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

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

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

[397] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264 - 323, September 1999.

[398] N. Jardine and R. Sibson. *Mathematical Taxonomy*. Wiley, New York, 1971.

[399] G. Karypis, E.-H. Han, and V. Kumar. Multilevel Refinement for Hierarchical Clustering. Technical Report TR 99-020, University of Minnesota, Minneapolis, MN, 1999.

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

[401] J. M. Kleinberg. An Impossibility Theorem for Clustering. In *Proc. of the 16th Annual Conf. on Neural Information Processing Systems*, December, 9 - 14 2002.

[402] B. Larsen and C. Aone. Fast and Effective Text Mining Using Linear-Time Document Clustering. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 16 - 22, San Diego, California, 1999. ACM Press.

[403] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, pages 281 - 297. University of California Press, 1967.

[404] G. W. Milligan. Clustering Validation: Results and Implications for Applied Analyses. In P. Arabie, L. Hubert, and G. D. Soete, editors, *Clustering and Classification*, pages 345 - 375. World Scientific, Singapore, January 1996.

[405] B. Mirkin. *Mathematical Classification and Clustering*, volume 11 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, August 1996.

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

[407] F. Murtagh. *Multidimensional Clustering Algorithms*. Physica-Verlag, Heidelberg and Vienna, 1985.

[408] D. Pelleg and A. W. Moore. *X*-means: Extending *K*-means with Efficient Estimation of the Number of Clusters. In *Proc. of the 17th Intl. Conf. on Machine Learning*, pages 727 - 734. Morgan Kaufmann, San Francisco, CA, 2000.

[409] C. Romesburg. *Cluster Analysis for Researchers*. Life Time Learning, Belmont, CA, 1984.

[410] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications. *Data Mining and Knowledge Discovery*, 2(2):169 - 194, 1998.

[411] S. M. Savaresi and D. Boley. A comparative analysis on the bisecting K-means and the PDDP clustering algorithms. *Intelligent Data Analysis*, 8(4):345 - 362, 2004.

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

[413] H. Späth. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*, volume 4 of *Computers and Their Application*. Ellis Horwood Publishers, Chichester, 1980. ISBN 0-85312-141-9.

[414] M. Steinbach, G. Karypis, and V. Kumar. A Comparison of Document Clustering Techniques. In *Proc. of KDD Workshop on Text Mining, Proc. of the 6th Intl. Conf. on Knowledge Discovery and*

Data Mining, Boston, MA, August 2000.

[415] C. T. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, C-20(1):68 - 86, Jan. 1971.

[416] B. Zhang, M. Hsu, and U. Dayal. K-Harmonic Means—A Data Clustering Algorithm. Technical Report HPL-1999-124, Hewlett Packard Laboratories, Oct. 29 1999.

[417] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311 - 331, 2004.

习　题

1. 考虑一个由 2^{20} 个数据向量组成的数据集，其中每个向量具有 32 个分量，而每个分量是 4 字节值。假定向量量化用于压缩，并且使用 2^{16} 个原型向量。压缩前后该数据集各需要多少字节的存储空间，压缩率是多少？
2. 找出图 8-35 所示点集中的所有明显分离的簇。

图 8-35　习题 2 的点

3. 许多自动地确定簇个数的划分聚类算法都声称这是它们的优点。列举两种情况，表明事实并非如此。
4. 给定 K 个等大小的簇，随机选取的初始质心来自一个给定的簇的概率是 $1/K$，但是每个簇恰好包含一个初始质心的概率要低得多。（应当清楚，每个簇有一个初始质心对于 K 均值是一个很好的开端。）一般地说，如果有 K 个簇，而每个簇有 n 个点，则在一个大小为 K 的样本中，由每个簇选取一个初始质心的概率 p 由公式（8-20）给出。（假定采用有放回抽样。）例如，由该公式我们可以计算 4 个簇每个具有一个初始质心的可能性是 $4!/4^4 = 0.093\,8$。

$$p = \frac{\text{从每个簇选取一个质心的选法}}{\text{选取}K\text{个质心的选法}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K} \tag{8-20}$$

   (a) 对于 2 和 100 之间的 K 值，绘制从每个簇得到一个点的概率。

   (b) 对于 K 个簇，$K = 10$、100 和 1 000，找出大小为 $2K$ 的样本至少包含来自每个簇中的一个点的概率。可以使用数学方法或统计估计确定答案。
5. 使用基于中心、邻近性和密度的方法，识别图 8-36 中的簇。对于每种情况指出簇个数，并简要给出你的理由。注意，明暗度或点数指明密度。如果有帮助的话，假定基于中心即 K 均值，基于邻近性即单链，而基于密度为 DBSCAN。
6. 对于下面的二维点集，(1)简略描述对于给定的簇个数，如何使用 K 均值将它们划分成簇，(2)指出结果质心大约在何处。假定使用平方误差目标函数。如果你认为存在多于一个解，则指出每个解是全局最小还是局部最小。注意，图 8-37 中，每个图的标记与本题的对应部分匹配；例如，图 8-37a 与(a)问题匹配。

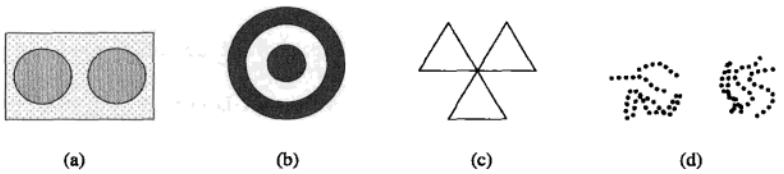


图 8-36　习题 5 的簇

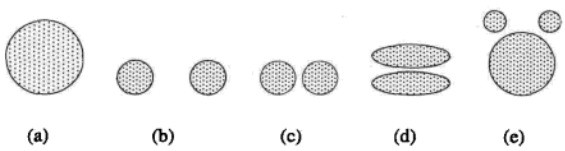


图 8-37　习题 6 的图

(a) $K = 2$。假定点均匀分布在圆中，（理论上）有多少种方法能将这些点划分成两个簇？两个质心在何处？（不必提供质心的准确位置，只需要定性描述。）

(b) $K = 3$。两个圆的边之间的距离略大于圆的半径。

(c) $K = 3$。两个圆的边之间的距离比圆的半径小得多。

(d) $K = 2$。

(e) $K = 3$。提示：利用对称性，并且记住我们只是寻找粗略的结果。

7. 假定一个数据集：

- 有 m 个点，K 个簇；
- 一半的点和簇在“较稠密的”区域；
- 一半的点和簇在“不太稠密的”区域；
- 两个区域之间是明显分离的。

对于给定的数据集，下面哪种情况可以最小化寻找 K 个簇时的平方误差？

(a) 在较稠密和不太稠密的区域质心分布应当相同。

(b) 不太稠密的区域应当分配更多的质心。

(c) 较稠密的区域应当分配更多的质心。

注意：不要被特殊情况转移视线，也不要引进除密度之外的因素。然而，如果你感到使用上面给定的条件很难得到答案，阐明你的理由。

8. 考虑取自二元事务数据集的对象的簇均值。均值分量的最小值和最大值是什么？簇均值分量如何解释？哪个分量最准确地刻画簇中的对象？

9. 给出一个数据集的例子，它包含三个自然簇。对于该数据集，K 均值（几乎总是）能够发现正确的簇，但是二分 K 均值不能。

10. 对于使用 K 均值对时间序列数据聚类，余弦度量是合适的相似性度量吗？为什么？如果不是，哪种相似性度量更合适？

11. 总 SSE 是每个属性的 SSE 之和。如果对于所有的簇，某变量的 SSE 都很低，这意味什么？如果只对一个簇很低呢？如果对所有簇都高呢？如果仅对一个簇高呢？如何使用每个变量的 SSE 信息改进聚类？

12. 领导者算法（Hartigan[394]）用一个点（称作领导者）代表一个簇，并将每个点指派到

最近的领导者对应的簇，除非距离大于用户指定的阈值。在那样的情况下，该点成为一个新簇的领导者。

(a) 与 K 均值比较，领导者算法的优点和缺点是什么？

(b) 提出可以改进领导者算法的方法。

13. 平面上 K 个点集合的 Voronoi 图是将平面上所有点分成 K 个区域的一个划分，使得（平面上）每个点都指派到 K 个指定点中最近的一个（见图 8-38）。Voronoi 图与 K 均值簇之间的关系是什么？关于 K 均值簇的可能形状，Voronoi 图能告诉我们什么？

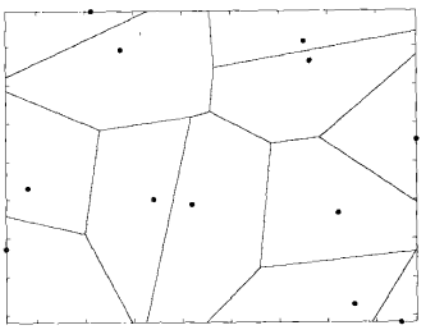


图 8-38　习题 13 的 Voronoi 图

14. 给定具有 100 个记录的数据集，要求对数据聚类。使用 K 均值对数据聚类，但是对所有的 K 值（$1 \leqslant K \leqslant 100$），K 均值算法都只返回一个非空簇。再用 K 均值的增量版本，但得到的结果完全相同。这怎么回事？用单链或 DBSCAN 处理该数据，结果如何？

15. 传统的凝聚层次聚类过程每步合并两个簇。这样的方法能够正确地捕获数据点集的（嵌套的）簇结构吗？如果不能，解释如何对结果进行后处理，以得到簇结构更正确的视图。

16. 使用表 8-13 中的相似度矩阵进行单链和全链层次聚类。绘制树状图显示结果。树状图应当清楚地显示合并的次序。

表 8-13　习题 16 的相似度矩阵

	p_1	p_2	p_3	p_4	p_5
p_1	1.00	0.10	0.41	0.55	0.35
p_2	0.10	1.00	0.64	0.47	0.98
p_3	0.41	0.64	1.00	0.44	0.85
p_4	0.55	0.47	0.44	1.00	0.76
p_5	0.35	0.98	0.85	0.76	1.00

17. 有时，层次聚类用来产生 K 个簇，$K > 1$。方法是取树状图的第 K 层（根在第一层）的簇。通过观察这种方法产生的簇，我们可以评估不同数据和簇类型的层次聚类行为，并且将层次聚类与 K 均值进行比较。

下面是一维点的集合：{6, 12, 18, 24, 30, 42, 48}。

(a) 对于下列每组初始质心，将每个点指派到最近的质心，创建两个簇，然后对两个簇的每组质心分别计算总平方误差。对每组质心，给出这两个簇和总平方误差。

i. {18, 45}

ii. {15, 40}

(b) 两组质心代表稳定解吗，即如果在该数据集上，使用给定的质心作为初始质心运行 K 均值，所产生的簇会有改变吗？

(c) 单链产生的簇是什么？

(d) 在此情况下，哪种技术（K 均值或单链）能够产生“最自然的”簇？（对于 K 均值，用最小平方误差产生聚类。）

(e) 这个自然聚类对应于哪种（些）簇定义？（明显分离的、基于中心的、基于邻近的或基于密度的。）

(f) K 均值算法的哪个著名特性解释了前面的行为？

18. 假定使用 Ward 方法、二分 K 均值和一般的 K 均值找到了 K 个簇。这些解中的哪些代表局部或全局最小？解释你的结论。

19. 层次聚类算法需要 $O(m^2 \log m)$时间，因此直接用于大型数据集是不现实的。一种减少所需要时间的技术是对数据集抽样。例如，如果期望 K 个簇，并且从 m 个点中抽取 $\sqrt{m}$ 个点作为样本，则层次聚类算法将在大约 $O(m)$时间产生一个层次聚类。取树状图第 K 层中的簇，便可以从层次聚类提取 K 个簇。使用各种策略，可以在线性时间内将其余的点指派到簇中。具体地说，可以计算这 K 个簇的质心，然后将剩余的 $m-\sqrt{m}$ 个点指派到最近的质心所关联的簇中。

对于下面每种数据和簇类型，简略讨论(1)对于这种方法，抽样是否会导致问题。(2)可能导致的问题有哪些。假定抽样技术随机地从 m 的点的数据集中选择点，并且数据或簇的未提及的特性都尽可能是最优的。换言之，只关注提到的特定性质导致的问题。最后，假定 K 比 m 小得多。

(a) 数据具有大小很不同的簇。

(b) 高维数据。

(c) 具有离群点（即非常见点）的数据。

(d) 具有高度不规则区域的数据。

(e) 具有球形簇的数据。

(f) 具有很不相同的密度的数据。

(g) 具有少量噪声点的数据。

(h) 非欧几里得数据。

(i) 欧几里得数据。

(j) 具有许多属性和混合属性类型的数据。

20. 考虑图 8-39 中显示的 4 张脸。明暗度或点数仍然表示密度。线用来区分区域，并不代表点。

(a) 对于每个图，可以使用单链找出鼻子、眼睛和嘴代表的模式吗？解释原因。

(b) 对于每个图，可以使用 K 均值找出鼻子、眼睛和嘴代表的模式吗？解释原因。

(c) 对于检测图 8-39c 中的点形成的所有模式，聚类存在什么局限性？

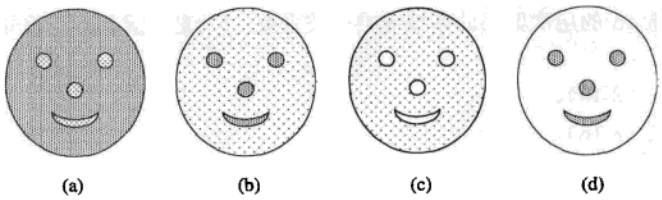


图 8-39　习题 20 的图

21. 计算表 8-14 的混淆矩阵的熵和纯度。

表 8-14　习题 21 的混淆矩阵

簇	娱乐	财经	国外	都市	国内	体育	合计
#1	1	1	0	11	4	676	693
#2	27	89	333	827	253	33	1562
#3	326	465	8	105	16	29	949
合计	354	555	341	943	273	738	3204

22. 给定两个点集，每个点集包含 100 个落在单位正方形中的点。一个点集这样安排，使得点在空间中均匀地分布。另一个点集由单位正方形上的均匀分布产生。
    (a) 这两个点集之间有差别吗？
    (b) 如果有，对于 $K = 10$ 个簇，哪一个点集通常具有较小的 SSE？
    (c) DBSCAN 在均匀数据集上的行为如何？在随机数据集上呢？
23. 使用习题 24 的数据计算每个点、每个簇和整个聚类的轮廓系数。
24. 给定分别由表 8-15 和表 8-16 显示的簇标号集和相似度矩阵，计算该相似度矩阵与理想的相似度矩阵之间的相关度。理想的相似度矩阵的第 ij 项为 1，如果两个对象属于同一个簇，否则为 0。

表 8-15　习题 24 的簇标号表

点	簇标号
p_1	1
p_2	1
p_3	2
p_4	2

表 8-16　习题 24 的相似度矩阵

点	p_1	p_2	p_3	p_4
p_1	1	0.8	0.65	0.55
p_2	0.8	1	0.7	0.6
p_3	0.65	0.7	1	0.9
p_4	0.55	0.6	0.9	1

25. 对于 8 个对象{p1, p2, p3, p4, p5, p6, p7, p8}和图 8-40 显示的层次聚类，计算层次 F 度量。类 A 包含点 p1、p2 和 p3，而 p4、p5、p6、p7 和 p8 属于类 B。

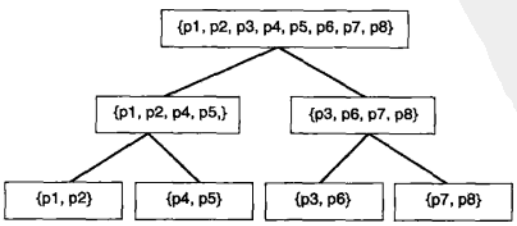


图 8-40　习题 25 的层次聚类

26. 计算习题 16 的层次聚类的共性分类相关系数。（可能需要将相似度矩阵转换成相异度矩阵。）
27. 证明公式（8-14）。
28. 证明公式（8-16）。
29. 证明 $\sum_{i=1}^{K}\sum_{x\in C_i}(x-m_i)(m-m_i)=0$。该事实用于证明 8.5.2 节的 TSS = SSE + SSB。
30. 文档簇可以通过发现簇中文档的最高项（词）概括。例如，通过取最频繁的 k 个项（其中，k 是常数，如 10），或者通过取出现频率超过指定阈值的所有项来概括。假定使用 K 均值来发现文档数据集中文档的簇和词的簇。
    (a) 文档簇中的最高项定义的项簇的集合与使用 K 均值对项聚类找到的词簇有何不同？
    (b) 如何使用项聚类来定义文档簇？
31. 我们可以将一个数据集表示成对象结点的集合和属性结点的集合，其中每个对象与每个属性之间有一条边，该边的权值是对象在该属性上的值。对于稀疏数据，如果权值为 0，则忽略该边。双划分聚类（Bipartite）试图将该图划分成不相交的簇，其中每个簇由一个对象结点集和一个属性结点集组成。该聚类的目标是最大化簇中对象结点和属性结点之间的边的权值，并且最小化不同簇的对象结点和属性结点之间的边的权值。这种聚类称作**协同聚类**（co-clustering），因为对象和属性同时聚类。
    (a) 双划分聚类（协同聚类）与对象集和属性集分别聚类有何不同？
    (b) 是否存在某些情况，这些方法产生相同的结果？
    (c) 与一般聚类相比，协同聚类的优点和缺点是什么？
32. 在图 8-41 中，相似度矩阵按簇标号存放。将相似度矩阵与点集匹配。不同的颜色深浅和标记形状区分不同的簇，并且每个点集包含 100 个点和 3 个簇。在标号为 2 的点集中，存在 3 个紧致的、大小相等的簇。

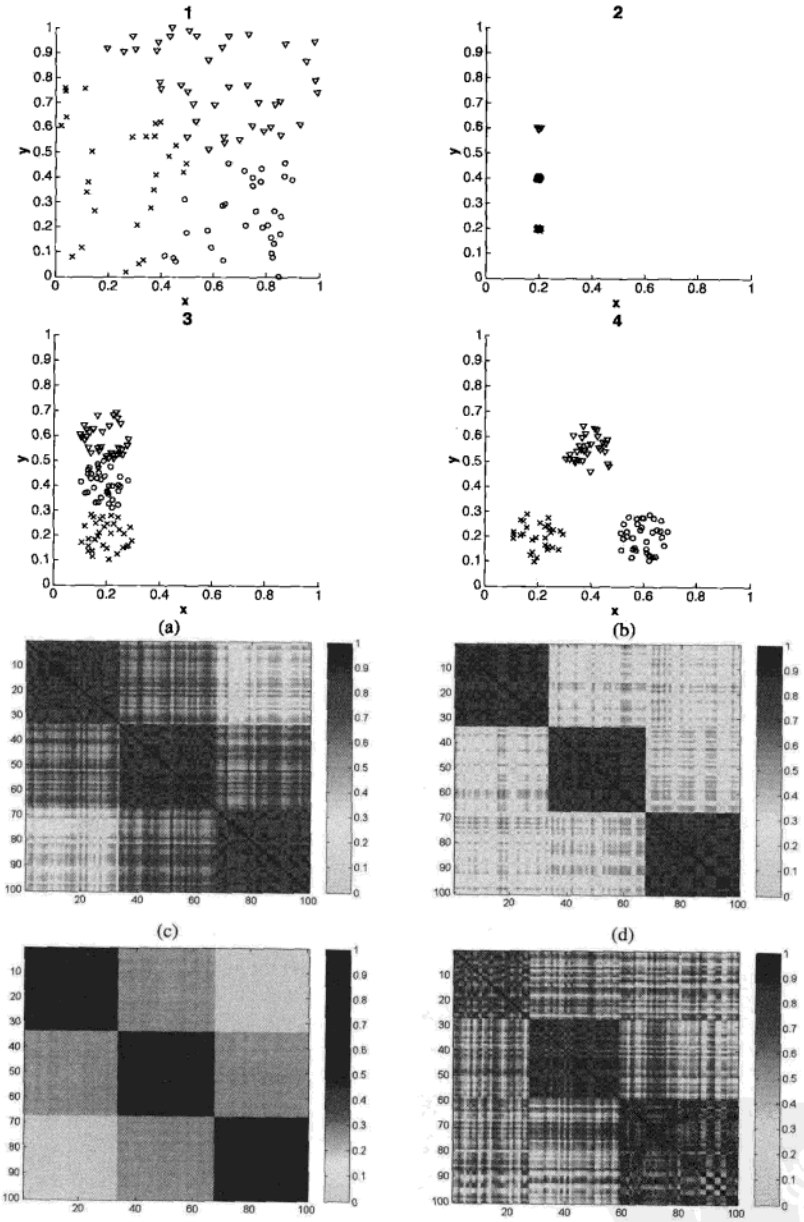


图 8-41　习题 32 的点集和相似度矩阵

第9章

聚类分析：其他问题与算法

在各个领域，针对不同的应用类型，人们已经开发了大量聚类算法。在这些算法中，没有一种算法能够适应所有的数据类型、簇和应用。事实上，对于更加有效或者更适合特定数据类型、簇或应用的新的聚类算法，看来总是有进一步的开发空间。我们只能说我们已经有了一些技术，对于某些情况运行良好。其原因是，在许多情况下，对于什么是一个好的簇集，仍然凭主观解释。此外，当使用客观度量精确地定义簇时，发现最优聚类问题常常是计算不可行的。

本章关注聚类分析的重要问题，并研究业已开发的处理它们的概念和技术。首先，我们讨论聚类分析的关键问题，即数据、簇和对聚类分析具有重要影响的算法的特性。这些问题对于理解、描述和比较聚类技术是重要的，并且提供在特定的情况下如何选择聚类技术的基础知识。例如，许多聚类算法具有 $O(m^2)$的时间或空间复杂度（m 是对象的个数），因此不适合大型数据集。然后，我们讨论其他聚类技术。对于每种技术，我们介绍算法，包括所处理的问题和处理这些问题所使用的方法。最后，我们用为给定的应用选择聚类算法提供某些一般准则结束本章。

9.1 数据、簇和聚类算法的特性

本节研究与数据、簇和聚类算法的特性相关的问题，这些对于全面理解聚类分析非常重要。其中某些问题具有挑战性，如处理噪声和离群点。其他问题涉及算法的期望特征，如无论数据对象以何种次序处理，都能产生相同结果的能力。本节的讨论，连同 8.1.2 节关于不同的聚类类型和 8.1.3 节关于不同的簇类型的讨论，可以确定一些可以用来描述和比较各种聚类算法及其聚类结果的“尺度”。为了解释这一点，我们从一个例子开始，此例比较上一章介绍的两种聚类算法 DBSCAN 和 K 均值。随后，更详细地介绍数据、簇和对聚类分析具有重要影响的算法的特性。

9.1.1 例子：比较 K 均值和 DBSCAN

为了简化比较，我们假定对于 K 均值和 DBSCAN 都没有距离的限制，并且 DBSCAN 总是将与若干个核心点相关联的边界点指派到最近的核心点。

- DBSCAN 和 K 均值都是将每个对象指派到单个簇的划分聚类算法，但是 K 均值一般聚类所有对象，而 DBSCAN 丢弃被它识别为噪声的对象。
- K 均值使用簇的基于原型的概念，而 DBSCAN 使用基于密度的概念。
- DBSCAN 可以处理不同大小和不同形状的簇，并且不太受噪声和离群点的影响。K 均值很难处理非球形的簇和不同大小的簇。当簇具有很不相同的密度时，两种算法的性能都很差。
- K 均值只能用于具有明确定义的质心（如均值或中位数）的数据。DBSCAN 要求密度定义（基于传统的欧几里得密度概念）对于数据是有意义的。

- K 均值可以用于稀疏的高维数据，如文档数据。DBSCAN 通常在这类数据上性能很差，因为对于高维数据，传统的欧几里得密度定义不能很好处理它们。
- K 均值和 DBSCAN 的最初版本都是针对欧几里得数据设计的，但是它们都被扩展，以便处理其他类型的数据。
- DBSCAN 不对数据的分布做任何假定。基本 K 均值算法等价于一种统计聚类方法（混合模型），假定所有的簇都来自球形高斯分布，具有不同的均值，但具有相同的协方差矩阵。见 9.2.2 节。
- DBSCAN 和 K 均值都寻找使用所有属性的簇，即它们都不寻找可能只涉及某个属性子集的簇。
- K 均值可以发现不是明显分离的簇，即便簇有重叠（见图 8-2b）也可以发现，但是 DBSCAN 会合并有重叠的簇。
- K 均值算法的时间复杂度是 $O(m)$，而 DBSCAN 的时间复杂度是 $O(m^2)$，除非用于诸如低维欧几里得数据这样的特殊情况。
- DBSCAN 多次运行产生相同的结果，而 K 均值通常使用随机初始化质心，不会产生相同的结果。
- DBSCAN 自动地确定簇个数；对于 K 均值，簇个数需要作为参数指定。然而，DBSCAN 必须指定另外两个参数：*Eps*（邻域半径）和 *MinPts*（最少点数）。
- K 均值聚类可以看作优化问题，即最小化每个点到最近的质心的误差的平方和，并且可以看作一种统计聚类（混合模型）的特例。DBSCAN 不基于任何形式化模型。

9.1.2　数据特性

下面是一些对聚类分析具有很强影响的数据特性。

高维性　在高维数据集中，传统的欧几里得密度定义（单位体积中点的个数）变得没有意义。为了看清这一点，考虑随着维数的增加，体积迅速增加，并且除非点的个数也随维数指数增加，否则密度将趋向于 0。（体积随维数指数增长。例如，半径为 r 维数为 d 的超球的体积正比于 r^d。）在高维空间中，邻近度也变得更加一致。看待这一事实的另一个角度是，存在更多确定两个点的邻近度的维（属性），而这会使邻近度更加一致。由于大部分聚类算法都基于邻近度或密度，因此处理高维数据时它们常常面临困难。处理该问题的一种方法是使用维归约技术。另一种方法，如 9.4.5 节和 9.4.7 节所讨论的，是重新定义邻近度和密度概念。

规模　许多聚类算法对于小规模和中等规模的数据集运行良好，但是不能处理大型数据集。这一问题将在讨论聚类算法的特性（可伸缩性就是这样的特性）时和 9.5 节进一步处理。9.5 节讨论可规模化的聚类算法。

稀疏性　稀疏数据通常由非对称的属性组成，其中零值没有非零值重要。因此，一般使用适合于非对称属性的相似性度量。然而，将会出现其他相关问题。例如，非零项的量级重要吗，或者它们会扭曲聚类吗？换言之，当只有两个值 0 和 1 时，聚类能够最好地处理吗？

噪声和离群点　非常见点（离群点）可能严重地降低聚类算法的性能，特别是 K 均值这样的基于原型的算法。另一方面，噪声也可能导致单链等技术合并两个不应当合并的簇。在某些情

况下，在使用聚类算法之前，先使用删除噪声和离群点的算法。还有些算法可以在聚类过程中检测代表噪声和离群点的点，然后删除它们或者消除它们的负面影响。例如，在前一章，我们看到DBSCAN 自动地将低密度的点分类成噪声，并把它们排除在聚类过程之外。Chameleon（9.4.4 节）、基于 SNN 密度的聚类（9.4.8 节）和 CURE（9.5.3 节）是本章介绍的三种算法，它们在聚类过程中显式地处理噪声和离群点。

属性和数据集类型 正如第 2 章所述，数据集可以有不用的类型，如结构化的、图形的或有序的，而属性也可以是分类的（标称的或序数的）、定量的（区间的或比率的）、二元的、离散的或连续的。不同的邻近性和密度度量适合于不同类型的数据。在某些情况下，数据可能需要离散化或二元化，以便可以使用期望的邻近性度量或聚类算法。当属性具有很多不同的类型（如连续的和标称的）时，另一种复杂情况将会出现。在这些情况下，邻近性和密度更难定义，并且常常更特殊。最后，可能需要特殊的数据结构和算法，来有效地处理特定类型的数据。

尺度 不同的属性，如高度和重量，可能用不同的尺度度量。这些差别可能严重影响两个对象之间的距离或相似性，从而影响聚类分析的结果。考虑根据身高和体重对一群人聚类，其中身高用米度量，而体重用千克度量。如果使用欧几里得距离作为邻近性度量，则身高的影响很小，人将主要根据体重属性聚类。然而，如果我们通过减去均值，再除以标准差，将每个属性都标准化，则我们将消除因尺度不同而造成的影响。更一般地，人们会使用诸如 2.3.7 节所讨论的那些规范化技术来处理这些问题。

数据空间的数学性质 有些聚类技术计算数据点集合的均值时，可能使用在欧几里得空间或其他具体数据空间有意义的其他数学运算。另一些算法要求密度的定义对于数据是有意义的。

9.1.3 簇特性

在 8.1.3 节，我们介绍了不同的簇类型，如基于原型的、基于图的和基于密度的。这里，我们介绍簇的其他重要特性。

数据分布 某些聚类技术假定数据具有特定的分布。更具体地说，它们常常假定可以用混合分布对数据建模，其中每个簇对应于一个分布。基于混合模型的聚类在 9.2.2 节讨论。

形状 有些簇具有规则的形状，如矩形和球形。但是，更一般地，簇可以具有任意形状。诸如 DBSCAN 和单链等技术可以处理任意形状的簇，但是基于原型的方法和诸如全链和组平均这样一些层次聚类技术不能进行这样的处理。Chameleon（9.4.4 节）和 CURE（9.5.3 节）提供专门用来处理这一问题的技术的例子。

不同大小 许多聚类方法，如 K 均值，当簇具有不同的大小时不能很好地完成任务（见 8.2.4 节）。这个主题将在 9.6 节进一步讨论。

不同密度 具有很不相同的密度的簇可能对诸如 DBSCAN 和 K 均值等算法造成问题。9.4.8 节提供的基于 SNN 密度的聚类技术处理这个问题。

无明显分离的簇 当簇接触或重叠时，有些聚类技术将应当分开的簇合并。有些发现不同簇的技术甚至随意地将点指派到这个或那个簇。9.2.1 节讨论的模糊聚类是一种旨在处理未形成明

显分离的簇的数据的技术。

簇之间的联系　在大部分聚类技术中，都不明显地考虑簇之间的联系，如簇的相对位置。9.2.3 节讨论的自组织映射（SOM）是一种在聚类期间直接考虑簇之间联系的聚类技术。例如，点到簇的指派影响邻近簇的定义。

子空间簇　簇可能只在维（属性）的一个子集中存在，并且使用一个维集合确定的簇可能与使用另一个维集合确定的簇很不相同。虽然这个问题在两个维时就可能出现，但是随着维度的增加，问题将变得越来越严重，因为维的可能子集数以总维数的指数增加。因此，除非维数相对很小，否则简单地在所有可能的维的子集中寻找簇是不可行的。

一种方法是使用 2.3.4 节讨论过的特征选择。然而，这种方法假定只有一个维子集存在簇。实际上，簇可能存在于多个不同的子空间（维的子集），其中一些是重叠的。9.3.2 节考虑处理子空间聚类的一般问题，即发现簇和它们生成的维。

9.1.4　聚类算法的一般特性

聚类算法各式各样。这里，我们提供对聚类算法重要特性的一般讨论，并在讨论特定的技术时做更具体的评论。

次序依赖性　对于某些算法，所产生的簇的质量和个数可能因数据处理的次序不同而显著地变化。尽管看起来要尽量避免这种算法，但是有时次序依赖性相对次要，或者算法可能具有其他期望的特性。SOM（9.2.3 节）是次序依赖算法的一个例子。

非确定性　像K均值这样的聚类算法不是次序依赖的，但是它们每次运行都产生不同的结果，因为它们取决于需要随机选择的初始化步骤。因为簇的质量可能随运行而变化，因此可能需要多次运行。

可伸缩性　包含数以百万计对象的数据集并不罕见，而用于这种数据集的聚类算法应当具有线性或接近线性的时间和空间复杂度。对于大型数据集，即使具有 $O(m^2)$复杂度的算法也不切实际。此外，数据集聚类技术不能总是假定数据放在内存，或者数据元素可以随机地访问。这样的算法对于大型数据集是不可行的。9.5 节专门讨论可伸缩问题。

参数选择　大部分聚类算法需要用户设置一个或多个参数。选择合适的参数值可能是困难的；因此，通常的态度是“参数越少越好”。如果参数值的很小改变就会显著改变聚类结果，则选择参数值就变得更加具有挑战性。最后，除非提供一个过程（可能涉及用户的输入）来确定参数值，否则算法的用户就不得不通过试探法找到合适的参数值。

或许，最著名的参数选择问题是为划分聚类算法（如 K 均值）“选择正确的簇个数”。处理这个问题的一种方法在 8.5.5 节给出，而其他方法的参考文献在文献注释中给出。

变换聚类问题到其他领域　一种被某些聚类技术使用的方法是将聚类问题映射到一个不同的领域。例如，基于图的聚类将发现簇的任务映射成将邻近度图划分成连通分支。

将聚类作为最优化问题处理　聚类常常被看作优化问题：将点划分成簇，根据用户指定的目标函数度量，最大化结果簇集合的优良度。例如，K 均值聚类算法（8.2 节）试图发现簇的集合，

使每个点到最近的簇质心距离的平方和最小。理论上讲，这样的问题可以通过枚举所有可能的簇集合，并选择具有最佳目标函数值的那个簇集合来解决。但是，这种穷举的方法在计算上是不可行的。因此，许多聚类技术都基于启发式方法，产生好的但并非最佳的聚类。另一种方法是在贪心的或局部基上使用目标函数。例如，8.3 节讨论的层次聚类技术在聚类过程的每一步都是做局部最优的（贪心的）决策。

题材安排

我们用类似于上一章的方式安排聚类算法的讨论，主要根据是否是基于原型的、基于密度的或基于图的方法将技术分组。对于可伸缩的聚类技术，专门用一节进行讨论。最后，讨论如何选择聚类算法。

9.2 基于原型的聚类

在基于原型的聚类中，簇是对象的集合，其中任何对象离定义该簇的原型比离定义其他簇的原型更近。8.2 节介绍的 K 均值是一种简单的基于原型的聚类算法，它使用簇中对象的质心作为簇的原型。本节讨论的聚类方法以一种或多种方式扩展基于原型的概念，如下所述。

- 允许对象属于多个簇。具体地说，对象以某个权值属于每一个簇。这样的方法针对这样的事实，即某些对象与多个簇原型一样近。
- 用统计分布对簇进行建模，即对象通过一个随机过程，由一个被若干统计参数（如均值和方差）刻画的统计分布产生。这种观点推广原型概念，并且可以使用牢固建立的统计学技术。
- 簇被约束为具有固定的联系。通常，这些联系是指定近邻关系的约束，即两个簇互为邻居的程度。约束簇之间的联系可以简化对数据的解释和可视化。

我们考虑三种特定的聚类算法，来解释这些基于原型的聚类的扩展。模糊 c 均值使用模糊逻辑和模糊集合论的概念，提出一种聚类方案，它很像 K 均值，但是不需要硬性地将对象指派到一个簇中。混合模型聚类采取这样的方法，簇集合可以用一个混合分布建模，每个分布对应一个簇。基于自组织映射（SOM）的聚类方法在一个框架（例如二维网格结构）内进行聚类，该框架要求簇具有预先指定的相互联系。

9.2.1 模糊聚类

如果数据对象分布在明显分离的组中，则把对象明确分类成不相交的簇看来是一种理想的方法。然而，在大部分情况下，数据集中的对象不能划分成明显分离的簇，指派一个对象到一个特定的簇也具有一定的任意性。考虑一个靠近两个簇边界的对象，它离其中一个稍微近一点。在大多数这种情况下，下面的做法更合适：对每个对象和每个簇赋予一个权值，指明该对象属于该簇的程度。从数学上讲，w_{ij} 是对象 $\mathbf{x}_i$ 属于簇 C_j 的权值。

正如下一节将介绍的，概率方法也可以提供这样的权值。尽管概率方法在许多情况下都是有用的，但是有时很难确定一个合适的统计模型。在这种情况下，就需要用非概率的聚类技术提供类似的能力。模糊聚类技术基于模糊集合论，并且提供一种产生聚类的自然技术，其中隶属权值（w_{ij}）具有自然的（但非概率的）解释。本节介绍模糊聚类的一般方法，并用模糊 c 均值（模糊 K 均值）给出一个具体的例子。

1. 模糊集合

1965年，Lotfi Zadeh引进**模糊集合论**（fuzzy set theory）和**模糊逻辑**（fuzzy logic）作为一种处理不精确和不确定性的方法。简要地说，模糊集合论允许对象以0和1之间的某个隶属度属于一个集合，而模糊逻辑允许一个陈述以0和1之间的确定度为真。传统的集合论和逻辑是对应的模糊集合论和模糊逻辑的特殊情况，它们限制集合的隶属度或确定度或者为0，或者为1。模糊概念已经用于许多不同的领域，包括控制系统、模式识别和数据分析（分类和聚类）。

考虑如下模糊逻辑的例子。陈述“天空多云”的为真程度可以定义为天空被云覆盖的百分比。例如，天空的50%被云覆盖，则“天空多云”为真的程度是0.5。如果我们有两个集合“多云天”和“非多云天”，则我们可以类似地赋予每一天隶属于这两个集合的程度。这样，如果一天25%多云，则它在“多云天”集合中具有0.25的隶属度，而在“非多云天”集合中具有0.75的隶属度。

2. 模糊簇

假定我们有一个数据点的集合$\mathcal{X}=\{\mathbf{x}_1,\cdots,\mathbf{x}_m\}$，其中每个点$\mathbf{x}_i$是一个$n$维点，即$\mathbf{x}_i=(x_{i1},\cdots,x_{in})$。模糊簇集$C_1, C_2,\cdots, C_k$是$\mathcal{X}$的所有可能模糊子集的一个子集。（这简单地表示对于每个点$\mathbf{x}_i$和每个簇C_j，隶属权值（度）w_{ij}已经赋予0和1之间的值。）然而，我们还想将以下合理的条件施加在簇上，以确保簇形成**模糊伪划分**（fuzzy psuedo-partition）。

(1) 给定点$\mathbf{x}_i$的所有权值之和为1：

$$\sum_{j=1}^{k} w_{ij} = 1$$

(2) 每个簇C_j以非零权值至少包含一个点，但不以权值1包含所有的点：

$$0<\sum_{i=1}^{m} w_{ij} < m$$

3. 模糊c均值

尽管存在多种模糊聚类（事实上，许多数据分析算法都可以“模糊化”），我们只考虑K均值的模糊版本，称作模糊c均值。在聚类文献中，不使用簇质心增量更新的K均值版本有时称作**c均值**，这个术语被模糊界接纳，用于K均值的模糊版本。模糊c均值算法有时称作FCM，由算法9.1给出。

算法9.1　基本模糊c均值算法

1：选择一个初始模糊伪划分，即对所有的w_{ij}赋值
2：**repeat**
3：　使用模糊伪划分，计算每个簇的质心
4：　重新计算模糊伪划分，即w_{ij}
5：**until** 质心不发生变化
（替换的终止条件是“如果误差的变化低于指定的阈值”或“如果所有w_{ij}的变化的绝对值都低于指定的阈值。”）

初始化之后，FCM重复地计算每个簇的质心和模糊伪划分，直到划分不再改变。FCM的结构类似于K均值。K均值在初始化之后，交替地更新质心和指派每个对象到最近的质心。具体地说，计算模糊伪划分等价于指派步骤。与K均值一样，FCM可以解释为试图最小化误差的平

方和（SSE），尽管 FCM 是基于 SSE 的模糊版本。事实上，K 均值可以看作 FCM 的特例，并且两个算法的行为相当类似。FCM 的细节介绍如下。

计算 SSE 误差的平方和（SSE）的定义修改为：

$$\mathrm{SSE}(C_1, C_2, \cdots, C_k) = \sum_{j=1}^{k}\sum_{i=1}^{m} w_{ij}^{p}\, dist(\mathbf{x}_i, \mathbf{c}_j)^2 \tag{9-1}$$

其中 $\mathbf{c}_j$ 是第 j 个簇的质心，而 p 是确定权值影响的指数，在 1 和∞之间取值。注意，这个 SSE 只不过是公式（8-1）给出的传统 K 均值的 SSE 的加权版本。

初始化 通常使用随机初始化。特殊地，权值随机地选取，同时限定与任何对象相关联的权值之和必须等于 1。与 K 均值一样，随机初始化是简单的，但是常常导致聚类结果代表 SSE 的局部最小。8.2.1 节包含了为 K 均值选择初始质心的讨论，与 FCM 也有很大关系。

计算质心 公式（9-2）给出的质心定义可以通过发现最小化公式（9-1）给定的模糊 SSE 的质心推导出来（见 8.2.6 节的方法）。对于簇 C_j，对应的质心 $\mathbf{c}_j$ 由下式定义：

$$c_j = \sum_{i=1}^{m} w_{ij}{}^{p}\mathbf{x}_i \Big/ \sum_{i=1}^{m} w_{ij}{}^{p} \tag{9-2}$$

模糊质心的定义类似于传统的质心定义，不同之处在于所有点都要考虑（任意点至少在某种程度上属于任意一个簇），并且每个点对质心的贡献要根据它的隶属度加权。对于传统的明确集合，所有的 w_{ij} 或者为 0，或者为 1，该定义退化为传统的质心定义。

选择 p 的值有几种考虑。选取 $p = 2$ 简化权值更新公式——见公式（9-4）。然而，如果所选取的 p 值接近 1，则模糊 c 均值的行为很像传统的 K 均值。另一方面，随着 p 增大，所有的簇质心都趋向于所有数据点的全局质心。换言之，随着 p 增大，划分变得越来越模糊。

更新模糊伪划分 由于模糊伪划分由权值定义，因此这一步涉及更新与第 i 个点和第 j 个簇相关联的权值 w_{ij}。公式（9-3）给出的权值更新公式可以通过限定权值之和为 1，最小化公式（9-1）中的 SSE 导出。

$$w_{ij} = (1/dist(\mathbf{x}_i, \mathbf{c}_j)^2)^{\frac{1}{p-1}} \Big/ \sum_{q=1}^{k} (1/dist(\mathbf{x}_i, \mathbf{c}_q)^2)^{\frac{1}{p-1}} \tag{9-3}$$

该公式显得有点神秘。然而，如果 $p = 2$，则我们得到公式（9-4），它简单一些。我们给出公式（9-4）的直观解释。这种解释稍加修改也适用于公式（9-3）。

$$w_{ij} = 1/dist(\mathbf{x}_i, \mathbf{c}_j)^2 \Big/ \sum_{q=1}^{k} 1/dist(\mathbf{x}_i, \mathbf{c}_q)^2 \tag{9-4}$$

直观地，权值 w_{ij} 指明点 $\mathbf{x}_i$ 在簇 C_j 中的隶属度。如果 $\mathbf{x}_i$ 靠近质心 $\mathbf{c}_j$（$dist(\mathbf{x}_i, \mathbf{c}_j)$比较小），则 w_{ij} 应当相对较高；而如果 $\mathbf{x}_i$ 远离质心 $\mathbf{c}_j$（$dist(\mathbf{x}_i, \mathbf{c}_j)$比较大），则 w_{ij} 相对较低。如果，$w_{ij} = 1/dist(\mathbf{x}_i, \mathbf{c}_j)^2$，即 w_{ij} 等于公式（9-4）的分子，则 w_{ij} 确实反映了这种情况。然而，除非加以规范化（即，除以公式（9-4）的分母），否则一个点的隶属权值之和不等于 1。总而言之，点在簇中的隶属权

值是点与簇质心距离平方的倒数，除以该点所有隶属权值之和。

现在考虑公式（9-3）中指数 $1/(p-1)$的影响。如果 $p>2$，则该指数降低赋予离点最近的簇的权值。事实上，随着 p 趋向于无穷大，该指数趋向于 0，而权值趋向于 $1/k$。另一方面，随着 p 趋向于 1，该指数加大赋予离点最近的簇的权值。随着 p 趋向于 1，关于最近簇的隶属权值趋向于 1，而关于其他簇的隶属权值趋向于 0。这对应于 K 均值。

例 9.1　三个圆形簇上的模糊 c 均值　图 9-1 显示对于 100 点的二维数据集，使用模糊 c 均值发现其三个簇的结果。每个点指派到它具有最大隶属权值的簇。属于各个簇的点用不同的标记显示，而点在簇中的隶属度用明暗程度表示。点越黑，它在被指派的簇中隶属度越高。靠近簇中心的点的隶属度最高，而簇间点的隶属度最低。　□

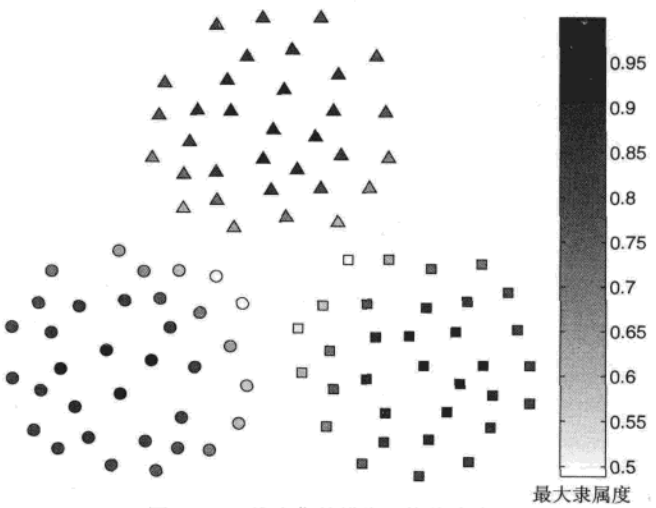


图 9-1　二维点集的模糊 c 均值聚类

4. 优点与局限性

FCM 的正面特征是，它产生指示任意点属于任意簇的程度的聚类。除此以外，它具有与 K 均值相同的优点和缺点，尽管它的计算密集程度更高一些。

9.2.2　使用混合模型的聚类

本节考虑基于统计模型的聚类。通常，一种方便而有效的做法是，假定数据是由一个统计过程产生的，并且通过找出最佳拟合数据的统计模型来描述数据，其中统计模型用分布和该分布的一组参数描述。在高层，该过程涉及确定数据的统计模型，并由数据估计该模型的参数。本节介绍一种特殊类型的统计模型，**混合模型**（mixture models），它使用若干统计分布对数据建模。每一个分布对应于一个簇，而每个分布的参数提供对应簇的描述，通常用中心和发散描述。

本节的讨论按如下次序进行。在描述混合模型之后，我们考虑如何估计统计数据模型的参数。首先介绍如何使用一个称作**最大似然估计**（Maximum Likelihood Estimation，MLE）的过程来估计简单统计模型的参数，然后讨论如何扩充该方法，来估计混合模型的参数。具体地说，我们介绍著名的**期望最大化**（Expectation-Maximization，EM）**算法**，它对参数做初始猜测，然后迭代

地改进这些估计。我们提供一些例子，展示如何通过估计混合模型的参数，使用 EM 算法对数据聚类，并讨论它的优点与局限性。

对于理解本节的内容，统计和概率的坚实基础是至关重要的。此外，为了方便起见，在下面的讨论中，我们使用术语概率表示概率和概率密度。

1. 混合模型

混合模型将数据看作从不同的概率分布得到的观测值的集合。概率分布可以是任何分布，但是通常是多元正态的，因为这种类型的分布已被人们完全理解，容易从数学上进行处理，并且已经证明在许多情况下都能产生好的结果。这种类型的分布可以对椭球簇建模。

概念上讲，混合模型对应于如下数据产生过程。给定几个分布（通常类型相同但参数不同），随机地选取一个分布并由它产生一个对象。重复该过程 m 次，其中 m 是对象的个数。

更形式地，假定有 K 个分布和 m 个对象 $\mathcal{X}=\{\mathbf{x}_1,\cdots,\mathbf{x}_m\}$。设第 j 个分布的参数为 θ_j，并设 Θ 是所有参数的集合，即 $\Theta=\{\theta_1,\cdots,\theta_K\}$。则 $prob(\mathbf{x}_i|\theta_j)$ 是第 i 个对象来自第 j 个分布的概率。选取第 j 个分布产生一个对象的概率由权值 $w_j(1\leqslant j\leqslant K)$ 给定，其中权值（概率）受限于其和为 1 的约束，即 $\sum_{j=1}^{K} w_j=1$。于是，对象 $\mathbf{x}$ 的概率由公式（9-5）给出。

$$prob(\mathbf{x}|\Theta)=\sum_{j=1}^{K} w_j p_j(\mathbf{x}|\theta_j) \tag{9-5}$$

如果对象以独立的方式产生，则整个对象集的概率是每个个体对象 $\mathbf{x}_i$ 的概率的乘积。

$$prob(\mathcal{X}|\Theta)=\prod_{i=1}^{m} prob(\mathbf{x}_i|\Theta)=\prod_{i=1}^{m}\sum_{j=1}^{K} w_j p_j(\mathbf{x}_i|\theta_j) \tag{9-6}$$

对于混合模型，每个分布描述一个不同的组，即一个不同的簇。通过使用统计方法，我们可以由数据估计这些分布的参数，从而描述这些分布（簇）。我们也可以识别哪个对象属于哪个簇。然而，混合建模并不产生对象到簇的明确指派，而是给出具体对象属于特定簇的概率。

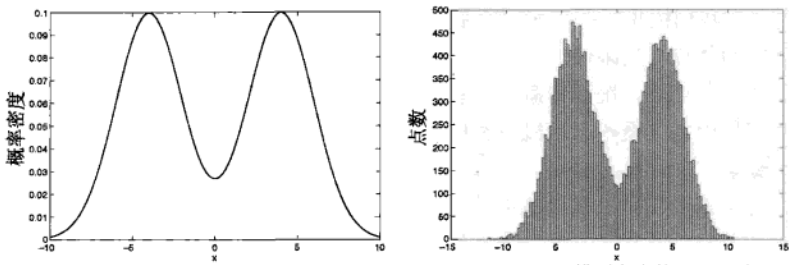
例 9.2 单变量的高斯混合分布 我们用高斯分布给出混合模型的具体解释。一维高斯分布在点 x 的概率密度函数是

$$prob(x_i|\Theta)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{9-7}$$

该高斯分布的参数是 $\theta=(\mu,\sigma)$，其中 μ 是分布的均值，而 σ 是标准差。假定有两个高斯分布，具有共同的标准差 2，均值分别为–4 和 4。还假定每个分布以等概率选取，即 $w_1=w_2=0.5$。于是，公式（9-5）变成

$$prob(x|\Theta)=\frac{1}{2\sqrt{2\pi}}e^{-\frac{(x+4)^2}{8}}+\frac{1}{2\sqrt{2\pi}}e^{-\frac{(x-4)^2}{8}} \tag{9-8}$$

图 9-2a 显示该混合模型的概率密度函数图，而图 9-2b 显示由该混合模型产生的 20 000 个点的直方图。 □



(a) 混合模型的概率密度函数　　(b) 由混合模型产生的 20 000 点

图 9-2　由两个正态分布组成的混合模型（两个分布的均值分别为–4 和 4，标准差都是 2）

2. 使用最大似然估计模型参数

给定数据的一个统计模型，必须估计该模型的参数。用于这类任务的标准方法是最大似然估计。现在我们对它进行解释。

首先，考虑由一维高斯分布产生的 m 个点的集合。假定点的产生是独立的，则这些点的概率是个体点概率的乘积（再次说明，我们处理的是概率密度，但是为了简化术语，我们称其为概率）。使用公式（9-7），我们可以将这个概率写成公式（9-9）。由于这个概率是一个非常小的数，因此我们一般使用对数概率，如公式（9-10）所示。

$$prob(\chi \mid \Theta) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}} \tag{9-9}$$

$$\log prob(\chi \mid \Theta) = -\sum_{i=1}^{m} \frac{(x_i-\mu)^2}{2\sigma^2} - 0.5m\log 2\pi - m\log\sigma \tag{9-10}$$

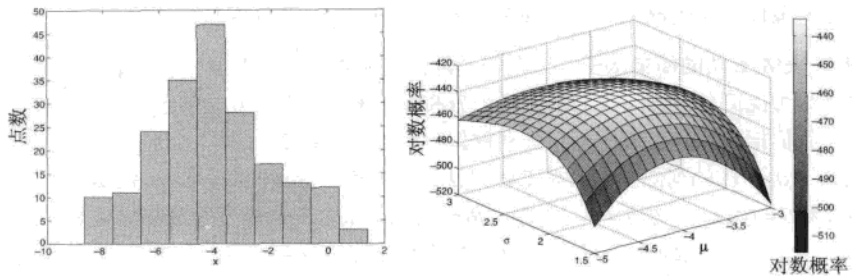
如果μ和σ的值未知，我们需要找到一个过程来估计它们。一种方法是选择合适的参数值使得数据是最可能的（最似然的）。换言之，选择最大化公式（9-9）的μ和σ。这种方法在统计学上称作**最大似然原理**（maximum likelihood principle），而使用该原理由数据估计统计分布参数的过程称作**最大似然估计**（Maximum Likelihood Estimation，MLE）。

之所以称该原理为最大似然原理，是因为给定一个数据集，数据的概率看作参数的函数，称作**似然函数**（likelihood function）。为了进行解释，我们将公式（9-9）写成公式（9-11），以强调我们把统计参数μ和σ看作变量，而把数据看作常量。考虑到实用性，对数似然更常用。从公式（9-10）的对数概率推导出来的对数似然显示在公式（9-12）中。注意，最大化对数似然的参数值也最大化该似然，因为 log 是单调增函数。

$$likelihood(\Theta \mid \chi) = L(\Theta \mid \chi) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}} \tag{9-11}$$

$$\log likelihood(\Theta \mid \chi) = \ell(\Theta \mid \chi) = -\sum_{i=1}^{m} \frac{(x_i-\mu)^2}{2\sigma^2} - 0.5m\log 2\pi - m\log\sigma \tag{9-12}$$

例 9.3　最大似然参数估计　我们给出使用 MLE 发现参数值的具体解释。假定我们有 200 个点的集合，其直方图显示在图 9-3a 中。图 9-3b 显示了所考虑的 200 点的最大对数似然图。使对数概率最大化的参数值是μ=–4.1 和σ=2.1，与基本高斯分布的参数值μ=–4.0 和σ=2.0 很接近。　□



(a) 符合高斯分布的 200 个点的直方图　　(b) 对于均值和标准差的不同值，200 个点的对数似然图

图 9-3　符合高斯分布的 200 个点及其在不同参数值下的对数概率

如果参数多于两个，对于参数的不同值绘制数据的似然图是不现实的。因而，标准的统计过程是，对似然函数关于参数求导，令结果等于 0 并求解，推导出统计参数的最大似然估计。特殊地，对于高斯分布，可以证明样本点的均值和标准差是基本分布对应参数的最大似然估计（见本章习题 9）。事实上，对于我们的例子所考虑的 200 个点，最大化对数似然的参数值恰好是这 200 个点的均值和标准差，即$\mu = -4.1$ 和$\sigma = 2.1$。

3. 使用最大似然估计混合模型参数：EM 算法

我们也可以使用最大似然方法来估计混合模型的参数。在最简单的情况下，我们知道哪个数据对象来自哪个分布，问题归结成：给定符合某分布的数据，估计单个分布的参数。对于大部分常见的分布，参数的最大似然估计由涉及数据的简单公式计算。

在更一般（并且更现实的）的情况下，我们不知道哪个点由哪个分布产生。这样，我们不能直接计算每一个点的概率，因此我们似乎不能使用最大似然原理来估计参数。该问题的解决方案是 EM 算法　显示在算法 9.2 中。简要地说，给定参数值的一个猜测，EM 算法计算每个点属于每个分布的概率，然后使用这些概率，计算参数的新的估计（这些参数是最大化该似然的参数）。该迭代继续下去，直到参数的估计不再改变或改变很小。这样，我们通过一个迭代搜索，仍然使用了最大似然估计。

算法 9.2　EM 算法

1：选择模型参数的初始集。
　（与 K 均值一样，可以随机地做，也可以用各种方法。）
2：**repeat**
3：　**期望步**　对于每个对象，计算每个对象属于每个分布的概率，即计算 $prob(\text{分布 } j \mid \mathbf{x}_i, \Theta)$。
4：　**最大化步**　给定期望步得到的概率，找出最大化该期望似然的新的参数估计。
5：**until** 参数不再改变。
　（替换地，如果参数的改变低于预先指定的阈值则停止。）

EM 算法类似于 8.2.1 节的 K 均值算法。事实上，欧几里得数据的 K 均值算法是具有相同协方差矩阵，但具有不同均值的球形高斯分布的 EM 算法的特殊情况。期望步对应于 K 均值将每个对象指派到一个簇的步骤，但将每个对象以某一概率指派到每个簇（分布）。最大化步对应于计算簇质心，但是选取分布的所有参数以及权值参数来最大化似然。这一过程常常是直截了当的，因为参数一般使用由最大似然估计推导出来的公式进行计算。例如，对于单个高斯分布，均值的 MLE 估计是分布中对象的均值。在混合分布和 EM 算法的背景下，均值的计算需要修改，

以说明每个对象以一定的概率属于某分布。下面的例子进一步解释这一点。

例 9.4　EM 算法的简单例子　这个例子解释 EM 算法用于图 9-2 的数据时如何执行。为了使这个例子尽可能简单，假定我们知道两个分布的标准差都是 2.0，并且点以相等的概率由两个分布产生。我们把左边和右边的分布分别称作分布 1 和分布 2。

我们从对μ_1和μ_2的初始猜测开始 EM 算法，比如说，取$\mu_1 = -2$，$\mu_2 = 3$。这样，对于两个分布，初始参数$\theta = (\mu, \sigma)$分别是$\theta_1 = (-2, 2)$和$\theta_2 = (3, 2)$。整个混合模型的参数集是$\Theta = \{\theta_1, \theta_2\}$。对于 EM 的期望步，我们要计算某个点取自一个特定分布的概率；即，我们要计算 $prob(\text{分布 } 1 | x_i, \Theta)$和 $prob(\text{分布 } 2 | x_i, \Theta)$。这些值可以用下式表示，它是贝叶斯规则的直接应用。

$$prob(\text{分布}j | x_i, \theta) = \frac{0.5\, prob(x_i | \theta_j)}{0.5\, prob(x_i | \theta_1) + 0.5\, prob(x_i | \theta_2)} \tag{9-13}$$

其中，0.5 是每个分布的概率（权），而 j 是 1 或 2。

例如，假定其中一个点是 0。使用公式（9-7）的高斯密度函数，我们计算 $prob(0 | \theta_1) = 0.12$，$prob(0 | \theta_2) = 0.06$（实际计算的是概率密度）。使用这些值和公式（9-13），我们发现 $prob(\text{分布 } 1 | 0, \Theta) = 0.12/(0.12 + 0.06) = 0.66$，$prob(\text{分布 } 2 | 0, \Theta) = 0.06/(0.12 + 0.06) = 0.33$。根据对参数值的当前假设，这意味点 0 属于分布 1 的可能性是属于分布 2 的可能性的两倍。

计算了 20 000 个点的簇隶属概率之后，我们在 EM 算法的最大化步，计算μ_1和μ_2的新的估计（使用公式（9-14）和公式（9-15））。注意，分布的均值的新的估计是点的加权平均，其中权值是点属于该分布的概率，即值 $prob(\text{分布 } j | x_i)$。

$$\mu_1 = \sum_{i=1}^{20\,000} x_i \frac{prob(\text{分布1} | x_i, \Theta)}{\sum_{i=1}^{20\,000} prob(\text{分布1} | x_i, \Theta)} \tag{9-14}$$

$$\mu_2 = \sum_{i=1}^{20\,000} x_i \frac{prob(\text{分布2} | x_i, \Theta)}{\sum_{i=1}^{20\,000} prob(\text{分布2} | x_i, \Theta)} \tag{9-15}$$

重复这两步，直到μ_1和μ_2的估计不再改变或变化很小。表 9-1 显示 EM 算法用于 20 000 个点的集合的前几次迭代。对于该数据，我们知道哪个分布产生哪些点，因此我们也可以由每个分布计算均值。这些均值是$\mu_1 = -3.98$ 和$\mu_2 = 4.03$。 □

表 9-1　简单例子 EM 算法的前几次迭代

迭代	μ_1	μ_2
0	−2.00	3.00
1	−3.74	4.10
2	−3.94	4.07
3	−3.97	4.04
4	−3.98	4.03
5	−3.98	4.03

例 9.5　样本数据集上的 EM 算法　我们给出三个例子，解释如何使用 EM 算法发现混合模型的簇。第一个例子基于用于解释模糊 c 均值算法（见图 9-1）的数据集。我们用三个具有不同均值和相同协方差矩阵的二维高斯分布对该数据建模。然后，使用 EM 算法对数据进行聚类。结果显示在图 9-4 中。每个点指派到它具有最大隶属权值的簇中。属于每个簇的点用不同形状的标

记显示，簇中的隶属度用明暗程度显示。在两个簇边界上的那些点的隶属度相对较低，而其他地方较高。比较图 9-4 和图 9-1 中的这些隶属权值和概率是很有趣的（见本章习题 11）。

对于第二个例子，我们使用混合模型对包含不同密度簇的数据进行聚类。该数据由两个自然簇组成，每个大约 500 个点。该数据通过合并两个高斯数据集而创建，其中一个数据集的中心在(–4, 1)，标准差为 2；而另一个的中心在(0, 0)，标准差为 0.5。图 9-5 显示了 EM 算法产生的聚类。尽管密度不同，但 EM 算法还是相当成功地识别出了原来的簇。

对于第三个例子，我们使用混合模型对 K 均值不能很好处理的数据集进行聚类。图 9-6a 显示混合模型算法产生的聚类，而图 9-6b 显示相同的 1 000 个点的集合上的 K 均值聚类。对于混合模型聚类，每个点已经指派到它具有最高概率的簇。两个图中都使用不同的标记来区分不同的簇。不要混淆图 9-6a 中的标记“+”和“x”。 □

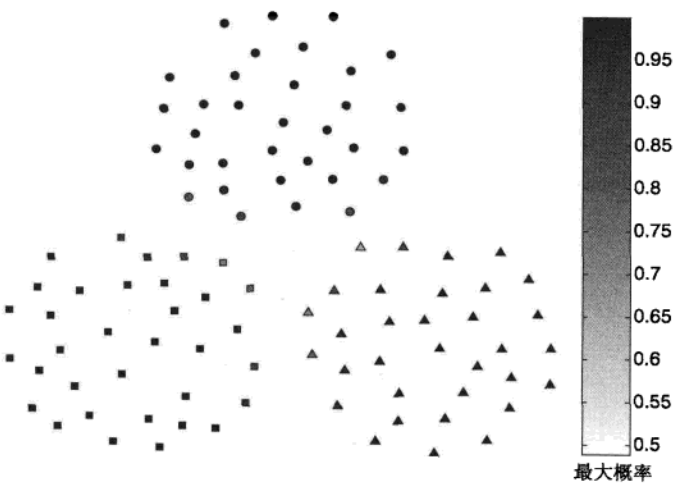


图 9-4 具有三个簇的二维数据点集的 EM 聚类

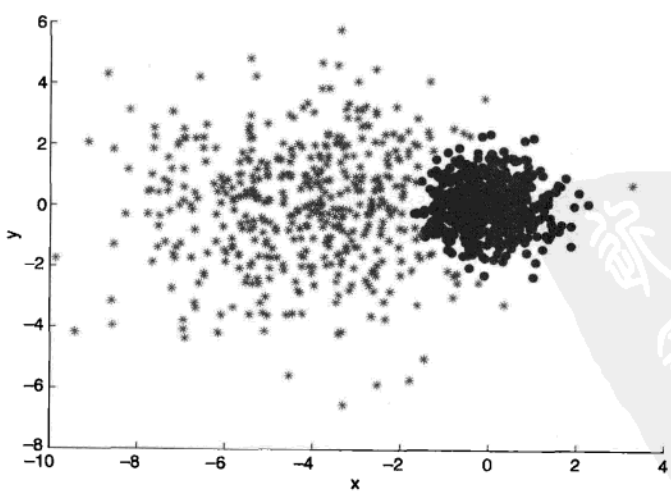
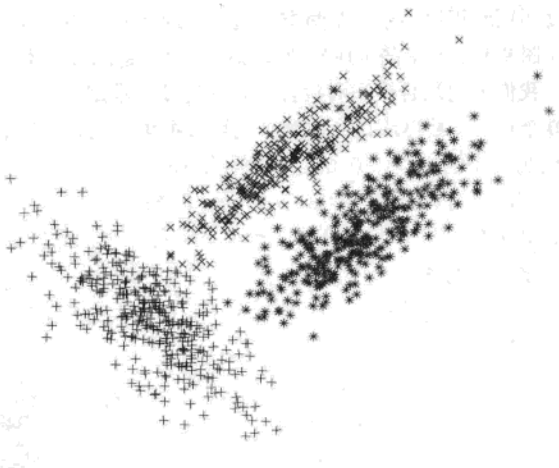
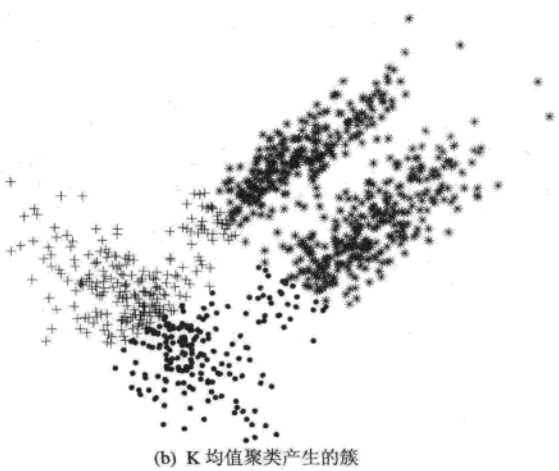


图 9-5 具有两个不同密度的簇的二维数据点集的 EM 聚类



(a) 混合模型聚类产生的簇



(b) K 均值聚类产生的簇

图 9-6　二维数据点集上的混合模型聚类和 K 均值聚类

4. 使用 EM 算法的混合模型聚类的优点和局限性

使用混合模型对数据建模，并使用 EM 算法估计这些模型的参数，从而会发现簇的方法有许多优点和缺点。缺点方面，EM 算法可能很慢，对于具有大量分量的模型可能不切实际；当簇只包含少量数据点，或者数据点近似协线性时，它也不能很好处理。在估计簇的个数，或更一般地，在选择正确的模型形式方面也存在问题。这个问题通常使用贝叶斯方法处理。粗略地说，贝叶斯方法基于由数据得到的估计，给出一个模型相对于另一个模型的几率。混合模型在有噪声和离群点时也可能有问题，尽管已经做了一些工作来处理该问题。

优点方面，混合模型比 K 均值或模糊 c 均值更一般，因为它可以使用各种类型的分布。这样，混合模型（基于高斯分布）可以发现不同大小和椭球形状的簇。此外，基于模型的方法提供了一种消除与数据相关联的复杂性的方法。为了看出数据中的模式，常常需要简化数据。如果模

型是数据的一个好的匹配的话，用数据拟合一个模型是一种简化数据的好办法。更进一步，模型更容易刻画所产生的簇，因为它们可以用少量参数描述。最后，许多数据集实际上是随机处理的结果，因此应当满足这些模型的统计假设。

9.2.3 自组织映射

Kohonen 自组织特征映射（SOFM 或 SOM）是一种基于神经网络观点的聚类和数据可视化技术。尽管 SOM 源于神经网络，但是它更容易（至少在本章的背景下）表示成一种基于原型的聚类的变形。与其他基于质心的聚类一样，SOM 的目标是发现质心的集合（用 SOM 的术语叫**参考向量**（reference vector）），并将数据集中的每个对象指派到提供该对象最佳近似的质心。用神经网络的术语，每个质心都与一个神经元相关联。

与增量 K 均值一样，每次处理一个数据对象并更新最近的质心。与 K 均值不同，SOM 赋予质心地形序（topographic ordering），也更新附近的质心。此外，SOM 不记录对象的当前簇隶属情况；并且不像 K 均值，如果对象转移簇，并不明确地更新旧的簇质心。当然，旧的簇质心可能是新的簇质心的近邻，这样它可能因此而更新。继续处理点，直到到达某个预先确定的界限，或者质心变化不大为止。SOM 的最终输出是一个隐式定义簇的质心的集合。每个簇由最靠近某个特定质心的点组成。下面考察该过程的细节。

1. SOM 算法

SOM 的显著特征是它赋予质心（神经元）一种地形（空间）组织。图 9-7 显示了一个二维 SOM 的例子，其中质心用安排在矩形格中的结点表示。每个质心分配一对坐标(i, j)。有时，这样的网络用邻接结点之间的链绘制，但这样可能误导，因为一个质心对另一个的影响是通过按坐标（而不是链）定义的邻域。有多种类型的 SOM 神经网络，但是我们只讨论具有矩形或六边形质心的二维 SOM。

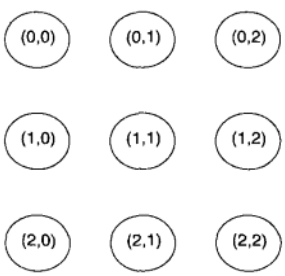


图 9-7 二维 3×3 矩形 SOM 神经网络

尽管 SOM 类似于 K 均值或其他基于原型的方法，但是存在根本上的差异。SOM 使用的质心具有预先确定的地形序关系。在训练过程中，SOM 使用每个数据点更新最近的质心和在地形序下邻近的质心。以这种方式，对于任意给定的数据集，SOM 产生一个有序的质心集合。换言之，在 SOM 网格中互相靠近的质心比远离的质心更加密切相关。由于这种约束，可以认为二维 SOM 质心在一个尽可能好地拟合 n 维数据的二维曲面上。SOM 质心也可以看作关于数据点的非线性回归的结果。

在高层，使用 SOM 技术聚类包含算法 9.3 中的步骤。

算法 9.3　基本 SOM 算法

1：初始化质心。
2：**repeat**
3：　选择下一个对象。
4：　确定到该对象最近的质心。
5：　更新该质心和附近的质心，即在一个特定邻域内的质心。
6：**until** 质心改变不多或超过某个阈值。
7：指派每个对象到最近的质心，并返回质心和簇。

初始化　这一步（行 1）可以用多种方法执行。一种方法是，对每个分量，从数据中观测到的值域随机地选择质心的分量值。尽管该方法可行，但是不一定是最好的，特别是对于快速收敛。另一种方法是从数据点中随机地选择初始质心。这非常像 K 均值随机地选择质心。

选择对象　循环的第一步（行 3）是选择下一个对象。这相当直接，但是存在一些困难。由于可能需要许多步才收敛，每个数据对象可能使用多次，特别是对象较少时。然而，如果对象很多，则并非需要使用每个对象。通过提高某些对象组在训练集中的出现频率，也可以增强这些对象组的影响。

指派　确定最近的质心（行 4）也是很简单的，尽管它需要具体的距离度量。通常使用欧几里得距离或点积度量。使用点积距离时，数据向量通常要预先规范化，并且要在每一步对参考向量进行规范化。在这种情况下，使用点积度量等价于使用余弦度量。

更新　更新步（行 5）最复杂。设 $\mathbf{m}_1,\cdots,\mathbf{m}_k$ 是质心（对于矩形网格，k 是行数与列数的乘积）。对于时间步 t，设 $\mathbf{p}(t)$是当前对象（点），并假定到 $\mathbf{p}(t)$最近的质心是 $\mathbf{m}_j$。则对于时间 $t+1$，使用下式更新第 j 个质心。（稍后我们将会看到，更新实际上限于其神经元在 $\mathbf{m}_j$ 的小邻域中的质心。）

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + h_j(t)(\mathbf{p}(t) - \mathbf{m}_j(t)) \tag{9-16}$$

这样，在时刻 t，质心 $\mathbf{m}_j(t)$被更新，加上一项 $h_j(t)(\mathbf{p}(t) - \mathbf{m}_j(t))$。新增的项正比于当前对象 $\mathbf{p}(t)$与质心 $\mathbf{m}_j(t)$之间的差 $\mathbf{p}(t) - \mathbf{m}_j(t)$。$h_j(t)$决定差 $\mathbf{p}(t) - \mathbf{m}_j(t)$将具有的影响，它的选取使得(1)随时间减退；(2)增强邻域效果，即对象在最接近质心 $\mathbf{m}_j$ 的质心上影响最大。这里我们所谈的是网格中的距离，而不是数据空间中的距离。通常，$h_j(t)$从以下两种函数选取：

$$h_j(t) = \alpha(t)\exp(-dist(\mathbf{r}_j, \mathbf{r}_k)^2/(2\sigma^2(t)))　\quad （高斯函数）$$

$$h_j(t) = \alpha(t)　\text{如果 } dist(\mathbf{r}_j, \mathbf{r}_k) \leqslant \text{阈值，否则 } 0　\quad （阶梯函数）$$

这些函数需要更多的解释。$\alpha(t)$是学习率参数，$0 < \alpha(t) < 1$，随时间单调减少，并控制收敛率。$\mathbf{r}_k = (x_k, y_k)$是二维点，给出第 k 个质心的网格坐标。$dist(\mathbf{r}_j, \mathbf{r}_k)$是两个质心网格位置之间的欧几里得距离，即 $\sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}$。这样，对于网格位置远离质心 $\mathbf{m}_j$ 的质心，对象 $\mathbf{p}(t)$的影响将大幅度减弱或不存在。最后，σ是典型的高斯方差参数，控制邻域的宽度；即，较小的σ将产生较小的邻域，而较大的σ将产生较宽的邻域。阶梯函数使用的阈值也控制邻域的大小。

记住，正是这种邻域更新技术加强了与邻近神经元相关联的质心之间的联系。

终止　决定何时足够接近稳定的质心集是一个重要的问题。理想情况下，迭代应当一直继续

到收敛为止，即直到参考向量不发生变化或变化很小。收敛率依赖于许多因素，如数据和$\alpha(t)$。除了一般地提及收敛可能很慢并且没有保证之外，我们不进一步讨论这些问题。

例 9.6 文档数据 我们提供两个例子。在第一个例子中，我们以边长为 4 的六边形网格，将 SOM 用于文档数据。我们对取自《洛杉矶时报》的 3 204 篇文章进行聚类。这些文章取自 6 个不同的版块：娱乐、财经、国外、都市、国内和体育。图 9-8 显示了 SOM 网格。我们使用六边形网格，这样每个质心具有 6 个直接邻居，而不是 4 个。每个 SOM 网格单元（簇）用相关联的点的多数类标记来标记。每个特定类的簇形成邻近组，并且它们相对于簇的其他类的位置给我们提供了附加的信息，例如，都市版块包含了与其他所有版块有关的故事。 □

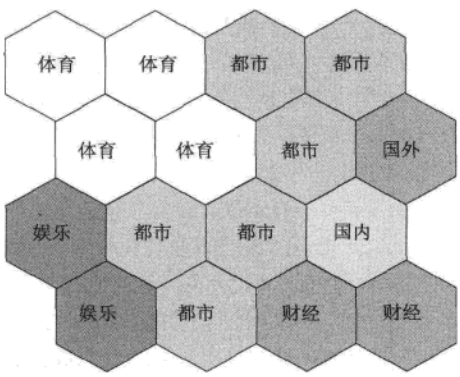
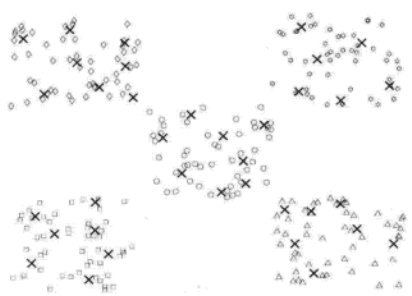


图 9-8 《洛杉矶时报》文档数据集的 SOM 簇之间的可视化联系

例 9.7 二维点 在第二个例子中，我们使用矩形 SOM 和一个二维数据点集。图 9-9a 显示点和 SOM 产生的 36 个参考向量（用“x”显示）的位置。点安排在棋盘模式中，并划分成 5 个类：圆形、三角形、正方形、菱形和六边形（星形）。使用 6 乘 6 的二维矩形质心网格并随机初始化。如图 9-9a 所示，质心趋向于分布在稠密区域。图 9-9b 指出与质心相关联的多数类。和与其他 4 种类型的点相关联的簇一样，与三角形点相关联的簇在一个连续的区域。这是 SOM 强加邻域约束的结果。尽管每组都有相同的点数，但是质心并非均匀地分布。原因一部分归结于点的总体分布，一部分归结于将每个质心放到单个簇中。 □



(a) 二维点集的 SOM 参考向量（“x”）的分布

菱形	菱形	菱形	六边形	六边形	六边形
菱形	菱形	菱形	圆形	六边形	六边形
菱形	菱形	圆形	圆形	圆形	六边形
正方形	正方形	圆形	圆形	三角形	三角形
正方形	正方形	圆形	圆形	三角形	三角形
正方形	正方形	正方形	三角形	三角形	三角形

(b) SOM 质心的类

图 9-9 用于二维数据点的 SOM

2. 应用

一旦找到 SOM 向量，就可以将它们用于聚类之外的许多目的。例如，使用二维 SOM，可以建立各种量与关联每个质心（簇）的网格点的联系，并通过各种类型的图显示结果。例如，绘制与每个簇相关联的点数将产生揭示点在簇之间分布的图。一个二维 SOM 是原概率分布函数到二维空间的非线性投影，该投影试图保持拓扑特征。这样，使用 SOM 捕获数据的结构被比作“压花”过程。

3. 优点与局限性

SOM 是一种聚类技术，它将相邻关系强加在结果簇质心上。正因为如此，互为邻居的簇之间比非邻居的簇之间更相关。这种联系有利于聚类结果的解释和可视化。事实上，SOM 的这一特点已经用在许多领域，如可视化 Web 文档或基因阵列数据。

SOM 也有许多局限性，列举如下。所列举的某些局限性仅当我们将 SOM 当作一种标准的聚类技术，旨在发现数据中真正的簇，而不是使用聚类帮助发现数据的结构时才是局限性。此外，其中某些局限性已经被扩充后的 SOM 或被受 SOM 影响的聚类算法所解决（见文献注释）。

- 用户必须选择参数、邻域函数、网格类型和质心个数。
- 一个 SOM 簇通常并不对应于单个自然簇。在某些情况下，一个 SOM 簇可能包含若干个自然簇，而在其他情况下，一个自然簇可能分解到若干个 SOM 簇中。这个问题部分地归结于簇网格的使用，部分地归结于如下事实：像其他基于原型的聚类技术一样，当自然簇的大小、形状和密度不同时，SOM 趋向于分裂或合并它们。
- SOM 缺乏具体的目标函数。SOM 试图找出最好地近似数据的质心的集合，受限于质心之间的地形约束；但是 SOM 的成功不能用一个函数来表达。这可能使得比较不同 SOM 聚类的结果是困难的。
- SOM 不能保证收敛，尽管实际中它通常收敛。

9.3　基于密度的聚类

在 8.4 节中，我们考虑了 DBSCAN，一种发现基于密度的簇的简单而有效的算法。基于密度的簇是对象的稠密区域，它们被低密度的区域所包围。本节将介绍其他基于密度的聚类技术，解决有效性、发现子空间中的簇和更准确的密度建模等问题。首先，我们考虑基于网格的聚类，它将数据空间划分成网格单元，然后由足够稠密的网格单元形成簇。这样的方法是有效的，至少对于低维数据如此。其次，我们考虑子空间聚类，它在所有维的子空间中寻找簇（稠密区域）。对于 n 维数据空间，需要搜索的潜在子空间有 $2^n - 1$ 个，因此需要有效的技术。CLIQUE 是一种基于网格的聚类算法，它基于如下观察提供了一种有效的子空间聚类方法：高维空间的稠密区域暗示低维空间稠密区域的存在性。最后，我们介绍一种聚类技术 DENCLUE，它使用核密度函数用个体数据对象影响之和对密度建模。尽管 DENCLUE 本质上不是基于网格的技术，但是它使用基于网格的方法提高性能。

9.3.1　基于网格的聚类

网格是一种组织数据集的有效方法，至少在低维空间中如此。其基本思想是，将每个属性的可能值分割成许多相邻的区间，创建网格单元的集合（对于这里和本节其余部分的讨论，我们假定属性值是序数的、区间的或连续的）。每个对象落入一个网格单元，网格单元对应的属性区间包含该对象的值。扫描一遍数据就可以把对象指派到网格单元中，并且还可以同时收集关于每个

单元的信息，如单元中的点数。

存在许多利用网格进行聚类的方法，但是大部分方法是基于密度的，至少部分地基于密度。因此，本节讨论的基于网格的聚类指的是使用网格的基于密度的聚类。算法 9.4 描述了基本的基于网格的聚类方法。该方法的各个步骤在下面介绍。

算法 9.4 基本的基于网格的聚类算法

1：定义一个网格单元集。
2：将对象指派到合适的单元，并计算每个单元的密度。
3：删除密度低于指定的阈值τ的单元。
4：由邻近的稠密单元组形成簇。

1. 定义网格单元

这是过程的关键步骤，但是定义也最不严格，因为存在许多方法将每个属性的可能值分割成许多相邻的区间。对于连续属性，一种常用的方法是将值划分成等宽的区间。如果该方法用于所有的属性，则结果网格单元都具有相同的体积，而单元的密度可以方便地定义为单元中点的个数。

然而，也可以使用更复杂的方法。例如，对于连续属性，通常用于离散化属性的任何技术都可以使用（见 2.3.6 节）。除已经提到的等宽方法之外，包括(1)将属性值划分成区间，使得每个区间包含的点数相等，即等频率离散化，或者(2)使用聚类。另一种方法被子空间聚类算法 MAFIA 使用，它初始地将属性值的集合划分成大量等宽区间，然后合并相近密度的区间。

无论采用哪种方法，网格的定义都对聚类的结果具有很大影响。我们稍后详细说明。

2. 网格单元的密度

定义网格单元密度的一种自然方法是：定义网格单元（或更一般形状的区域）的密度为该区域中的点数除以区域的体积。换言之，密度是每单位空间中的点数，而不管空间的维度。具体的、低维密度的例子是：每英里的路标个数（一维），每平方千米栖息地的鹰个数（二维），每立方厘米的气体分子个数（三维）。然而，正如所提到的，一种常用的方法是使用具有相同体积的网格单元，使得每个单元的点数直接度量单元的密度。

例 9.8 基于网格的密度 图 9-10 显示了两个二维点的集合，使用 7 乘 7 的网格划分成 49 个单元。第一个集合包含 200 个点，由圆心在(2, 3)、半径为 2 的圆上的均匀分布产生；而第二个集合包含 100 个点，由圆心在(6, 3)、半径为 1 的圆上的均匀分布产生。网格单元的计数显示在表 9-2 中。由于单元具有相等的体积（面积），因此我们可以将这些值看作单元的密度。 □

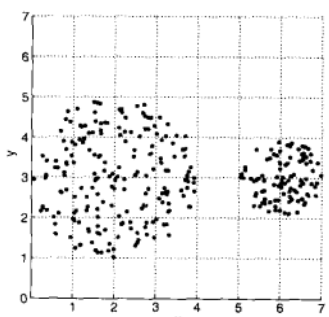


图 9-10 基于网格的密度

表 9-2 网格单元的点计数

0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

3. 由稠密网格单元形成簇

由邻接的稠密单元组形成簇是相对简单的（例如，在图 9-10 中，很明显存在两个簇）。然而，存在某些问题。我们需要定义邻接单元的含义。例如，二维网格单元有 4 个还是 8 个邻接单元？此外，我们需要有效的技术发现邻接单元，特别是当仅存放被占据的单元时更需要这种技术。

算法 9.4 定义的聚类方法有某些局限性，将算法改写得稍微复杂一点就可以处理。例如，在簇的边缘多半会有一些部分为空的单元。通常，这些单元不是稠密的。如果不稠密，它们将被丢弃，并导致簇的部分丢失。图 9-10 和表 9-2 显示，如果密度阈值为 9，则大簇的 4 个部分将丢失。可以修改聚类过程以避免丢弃这样的单元，尽管这需要附加的处理。

使用密度之外的信息也可以加强基本的基于网格的聚类算法。在许多情况下，数据具有空间和非空间属性。换言之，某些属性描述对象的时间或空间位置，而另一些属性描述对象的其他方面。一个常见的例子是房子，它具有位置和许多其他特性，如价格或占地面积。由于空间（或时间）的自相关性，一个特定单元中的对象通常在其他属性上也具有类似的值。在这些情况下，有可能基于一个或多个非空间属性的统计性质（如平均房价）对单元进行过滤，然后根据剩下的点的密度形成簇。

4. 优点与局限性

优点方面，基于网格的聚类可能是非常有效的。给定每个属性的划分，单遍数据扫描就可以确定每个对象的网格单元和每个网格单元的计数。此外，尽管潜在的网格单元数量可能很高，但是只需要为非空单元创建网格单元。这样，定义网格、将每个对象指派到一个单元并计算每个单元的密度的时间和空间复杂度仅为 $O(m)$，其中 m 是点的个数。如果邻接的、已占据的单元可以有效地访问（例如，通过使用搜索树），则整个聚类过程将非常高效，例如具有 $O(m \log m)$时间复杂度。正是由于这种原因，密度聚类的基于网格的方法形成了许多聚类算法的基础，如 STING、GRIDCLUS、WaveCluster、Bang-Clustering、CLIQUE 和 MAFIA。

缺点方面，像大多数基于密度的聚类方法一样，基于网格的聚类非常依赖于密度阈值τ的选择。如果τ太高，则簇可能丢失。如果τ太低，则本应分开的两个簇可能被合并。此外，如果存在不同密度的簇和噪声，则也许不能找到适用于数据空间所有部分的单个τ值。

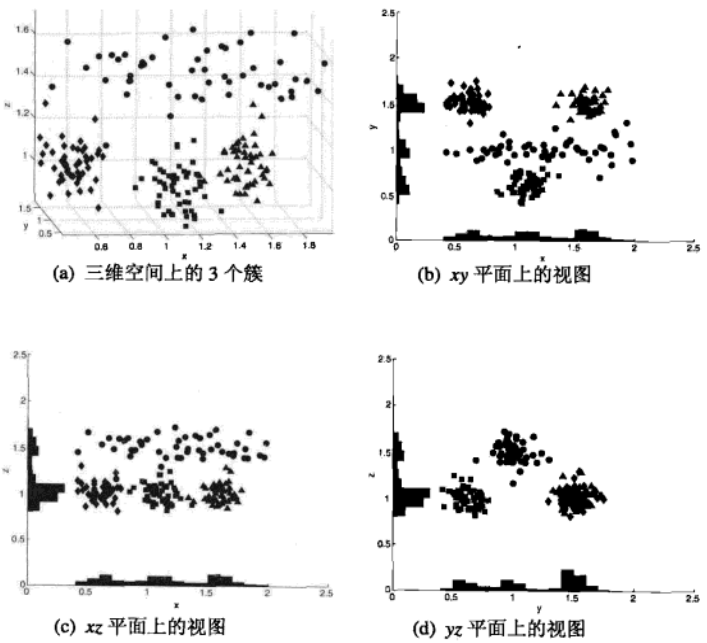
基于网格的方法还存在一些其他问题。例如，在图 9-10 中，矩形网格单元不能准确地捕获圆形边界区域的密度。我们可以试图通过将网格加细来减轻该问题，但是与一个簇相关联的网格单元中的点数可能更加波动，因为簇中的点不是均匀分布的。事实上，有些网格单元，包括簇内部的单元，甚至可能为空。另一个问题（依赖于单元的放置或大小）是一组点可能仅出现在一个单元中，或者分散在几个不同的单元中。在第一种情况下，同一组的点可能是簇的一部分，而在第二种情况下则可能被丢弃。最后，随着维度的增加，网格单元个数迅速增加——随维度指数增加。尽管不必明显地考虑空网格单元，但是大部分网格单元都只包含单个对象的情况很容易发生。换言之，对于高维数据，基于网格的聚类效果将会很差。

9.3.2　子空间聚类

迄今为止，所考虑的聚类技术都是使用所有的属性来发现簇。然而，如果仅考虑特征子集（即数据的子空间），则我们发现的簇可能因子空间不同而很不相同。有两个理由可以推断子空间的簇是有意义的。第一，数据关于少量属性的集合可能可以聚类，而关于其余属性是随机分布的。第二，在某些情况下，在不同的维集合中存在不同的簇。考虑记录不同时间、不同商品销售情况的数据集（时间是维，而商品是对象），某些商品对于特定的月份集（如夏季）可能表现出类似

行为，但是不同的簇可能被不同的月份（维）刻画。

例 9.9 子空间聚类 图 9-11a 显示一个三维空间点集。在整个空间有三个簇，分别用正方形、菱形和三角形标记。此外，有一个点集，用圆形标记，不是三维空间的簇。该数据集的每个维（属性）被划分成固定个数（η）的等宽区间。有$\eta = 20$ 个区间，每个宽度为 0.1。数据空间被划分成等体积的立方体单元，因此每个单元的密度是它所包含的点所占的比例。簇是稠密单元的邻接组。例如，如果稠密单元的阈值是$\xi = 0.06$，或 6%的点，则可以在图 9-12 中识别出 3 个一维簇。图 9-12 显示图 9-11a 的数据点关于 x 属性的直方图。



(a) 三维空间上的 3 个簇　(b) xy 平面上的视图

(c) xz 平面上的视图　(d) yz 平面上的视图

图 9-11 子空间聚类例子的图

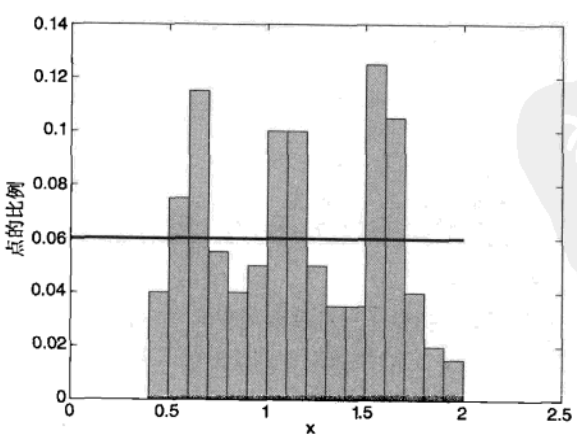


图 9-12 点关于 x 属性的分布直方图

图 9-11b 显示绘制在 xy 平面上的点（z 属性被忽略）。该图沿 x 和 y 轴也包含直方图，分别显示点关于其 x 和 y 坐标的分布（较高的条指明对应的区间包含相对较多的点，反之亦然）。当我们考虑 y 轴时，我们看到 3 个簇。一个来自在整个空间不形成簇的圆点，一个由正方形点组成，而另一个由菱形和三角形点组成。在 x 维上也有 3 个簇，它们对应于整个空间的 3 个簇（菱形、三角形和正方形）。这些点在 xy 平面上也形成不同的簇。图 9-11c 显示绘制在 xz 平面上的点。如果我们只考虑 z 属性，则存在两个簇。一个簇对应于圆表示的点，而另一个由菱形、三角形和正方形点组成。这些点在 xz 平面上也形成不同的簇。在图 9-11d 中，当我们考虑 y 和 z 时，存在 3 个簇。一个由圆组成，另一个由正方形标记的点组成。菱形和三角形形成 yz 平面上的单个簇。 □

这些图解释了两个重要事实。第一，一个点集（圆点）在整个空间可能不形成簇，但是在子空间却可能形成簇。第二，存在于整个数据空间（或者甚至子空间）的簇作为低维空间中的簇出现。第一个事实告诉我们可能需要在维的子集中发现簇，而第二个事实告诉我们许多在子空间中发现的簇可能只是较高维簇的“影子”（投影）。目标是发现簇和它们存在的维，但是我们通常对较高维簇的投影的那些簇并不感兴趣。

1. CLIQUE

CLIQUE（CLustering In QUEst）是系统地发现子空间簇的基于网格的聚类算法。检查每个子空间寻找簇是不现实的，因为这样的子空间的数量是维度的指数。CLIQUE 依赖如下性质。

基于密度的簇的单调性　如果一个点集在 k 维（属性）上形成一个基于密度的簇，则相同的点集在这些维的所有可能的子集上也是基于密度的簇的一部分。

考虑一个邻接的、形成簇的 k 维单元集，即其密度大于指定的阈值 ξ 的邻接单元的集合。对应的 $k-1$ 维单元集可以通过忽略 k 个维（属性）中的一个得到。这些较低维的单元也是邻接的，并且每个低维单元包含对应高维单元的所有点。它还可能包含附加的点。这样，低维单元的密度大于或等于对应高维单元的密度。结果，这些低维单元形成了一个簇，即点形成一个具有约减属性的簇。

算法 9.5 给出了一个 CLIQUE 的简化版本。从概念上讲，CLIQUE 算法类似于发现频繁项集的 *Apriori* 算法。见第 6 章。

算法 9.5　CLIQUE 算法

1：找出对应于每个属性的一维空间中的所有稠密区域。这是稠密的一维单元的集合。
2：$k \leftarrow 2$。
3：**repeat**
4：　由稠密的 $k-1$ 维单元产生所有的候选稠密 k 维单元。
5：　删除点数少于 ξ 的单元。
6：　$k \leftarrow k+1$。
7：**until** 不存在候选稠密 k 维单元。
8：通过取所有邻接的、高密度的单元的并发现簇。
9：使用一小组描述簇中单元的属性值域的不等式概括每一个簇。

2. CLIQUE 的优点与局限性

CLIQUE 的最有用的特征是，它提供了一种搜索子空间发现簇的有效技术。由于这种方法基于源于关联分析的著名的先验原理，它的性质能够被很好地理解。另一个有用特征是 CLIQUE 用一小组不等式概括构成一个簇的单元列表的能力。

CLIQUE 的许多局限性与前面讨论过的其他基于网格的密度方法相同。其他局限性类似于 *Apriori* 算法。具体地说，正如频繁项集可以共享项一样，CLIQUE 发现的簇也可以共享对象。允

许簇重叠可能大幅度增加簇的个数，并使得解释更加困难。另一个问题是 *Apriori*（和 CLIQUE）潜在地具有指数复杂度。例如，如果在较低的 k 值产生过多的稠密单元，则 CLIQUE 将遇到困难。提高密度阈值ξ可以减缓该问题。CLIQUE 的另一个潜在的局限性在本章习题 20 中考察。

9.3.3 DENCLUE：基于密度聚类的一种基于核的方案

DENCLUE（DENsity CLUstEring）是一种基于密度的聚类方法，它用与每个点相关联的影响函数之和对点集的总密度建模。结果总密度函数将具有局部尖峰（即局部密度最大值），并且这些局部尖峰用来以自然的方式定义簇。具体地说，对于每个数据点，爬山过程找出与该点相关联的最近的尖峰，并且与一个特定的尖峰（称作**局部密度吸引点**（local density attractor））相关联的所有数据点成为一个簇。然而，如果局部尖峰处的密度太低，则相关联的簇中的点将被视为噪声而丢弃。此外，如果一个局部尖峰通过一条数据点路径与另一个局部尖峰相连接，并且该路径上每个点的密度都高于最小密度阈值，则与这些局部尖峰相关联的簇合并在一起。这样就可以发现任意形状的簇。

例 9.10 DENCLUE 密度 我们用图 9-13 解释这些概念。该图显示了一维数据集的一个可能的密度函数。点 A～E 是该密度函数的尖峰，代表局部密度吸引点。垂直虚线描绘局部密度吸引点的局部影响区域。这些区域中的点将成为中心确定的簇。水平虚线显示密度阈值ξ。与密度小于ξ的局部密度吸引点相关联的所有点（如与 C 相关联的那些点）都将被丢弃。其他所有簇将留下。注意，留下的簇可能包括密度小于ξ的点，只要这些点与密度大于ξ的局部密度吸引点相关联。最后，通过密度大于ξ的点路径连接的簇合并在一起。簇 A 和 B 将保持分离，而簇 D 和 E 将合并。 □

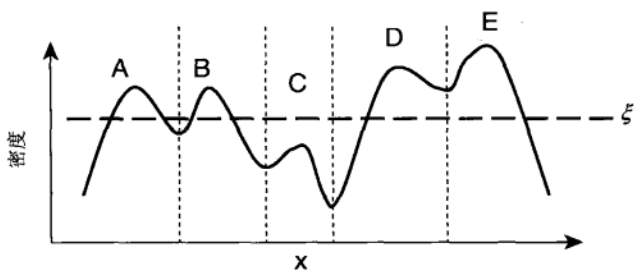


图 9-13 一维 DENCLUE 密度概念的解释

DENCLUE 算法的高层细节概括在算法 9.6 中。下面，我们更详细地讨论 DENCLUE 的各个方面。我们首先简略回顾核密度估计，然后提供 DENCLUE 用来近似密度的基于网格的方法。

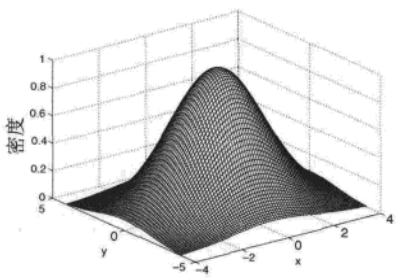
算法 9.6 DENCLUE 算法

1：对数据点占据的空间推导密度函数。
2：识别局部极大点。
（这些是密度吸引点。）
3：通过沿密度增长最大的方向移动，将每个点关联到一个密度吸引点。
4：定义与特定的密度吸引点相关联的点构成的簇。
5：丢弃密度吸引点的密度小于用户指定阈值ξ的簇。
6：合并通过密度大于或等于ξ的点路径连接的簇。

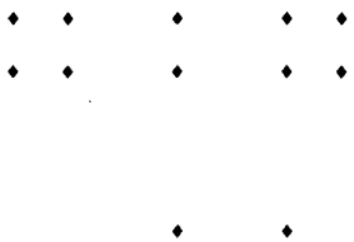
1. 核密度估计

DENCLUE 基于一个发展完善的统计学和模式识别领域，称作**核密度估计**（kernel density estimation）。这些技术（以及其他许多统计技术）的目标是用函数描述数据的分布。对于核密度估计，每个点对总密度函数的贡献用一个影响（influence）或**核函数**（kernel function）表示。总密度函数仅仅是与每个点相关联的影响函数之和。

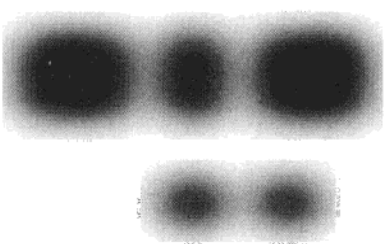
通常，影响函数或核函数是对称的（所有方向相同），并且它的值（贡献）随到点的距离增加而下降。例如，对于一个特定的点 **x**，高斯函数 $K(\mathbf{y}) = e^{-distance(x,y)^2/2\sigma^2}$ 常常用作核函数。（σ是参数，类似于标准差，它支配一个点的影响衰减的速度。）图 9-14a 显示单个二维点的高斯密度函数的形状，而图 9-14c 和图 9-14d 显示将该高斯影响函数用于图 9-14b 中的点集产生的总密度函数。



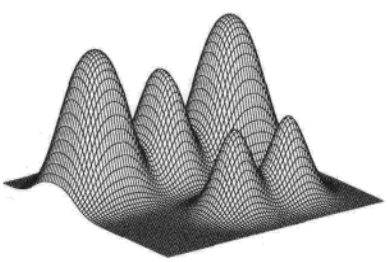
(a) 高斯核



(b) 12 个点的集合



(c) 总密度——灰度图



(d) 总密度——曲面图

图 9-14　高斯影响（核）函数和总密度函数的例子

2. 实现问题

核密度的计算开销可能相当高，DENCLUE 使用许多近似方法来有效地实现其基本算法。首先，它只在数据点显式地计算密度。然而，这仍然导致 $O(m^2)$时间复杂度，因为每个点的密度是所有点贡献的密度的函数。为了降低时间复杂度，DENCLUE 使用一种基于网格的实现来有效地定义近邻，并借此限制定义点的密度所需要考虑的点的数量。首先，预处理步创建网格单元集。仅创建被占据的单元，并且这些单元及其相关信息可以通过搜索树有效地访问。然后计算点的密度，并找出其最近的密度吸引点。DENCLUE 只考虑近邻中的点，即相同单元或者与该点所在单元相连接的单元中的点。尽管这种方法牺牲了一些密度估计的精度，但是计算复杂度大大降低。

3. DENCLUE 的优点与局限性

DENCLUE 具有坚实的理论基础，因为它基于统计学发展完善的领域——核密度函数和核密度估计。因为这种原因，DENCLUE 提供了比其他基于网格的聚类技术和 DBSCAN 更加灵活、更加精确的计算密度的方法（DBSCAN 是 DENCLUE 的特例）。基于核密度函数的方法本质上计算开销，但是 DENCLUE 使用基于网格的技术来处理该问题。尽管如此，DENCLUE 可能比其他基于密度的聚类技术的计算开销更大。此外，网格的使用对于密度估计的精度可能具有负面影响；并且这使得 DENCLUE 容易受基于网格的方法共同存在的问题的影响，例如，很难选择合适的网格尺寸。更一般地，DENCLUE 具有其他基于密度的方法的优点和局限性。例如，DENCLUE 擅长处理噪声和离群点，并且可以发现不同形状和不同大小的簇；但是对于高维数据和包含密度很不相同的簇的数据，DENCLUE 可能有问题。

9.4 基于图的聚类

8.3 节讨论了一些聚类技术，它们取数据的基于图的观点；其中，数据对象用结点表示，而两个数据对象之间的邻近度用对应结点之间边的权值表示。本节考虑其他一些基于图的聚类算法，它们利用图的许多重要性质和特性。下面是一些重要方法，算法利用这些方法的不同子集。

(1) 稀疏化邻近度图，只保留对象与其最近邻之间的连接。这种稀疏化对于处理噪声和离群点是有用的。稀疏化也使得我们可以利用为稀疏图开发的有效图划分算法。

(2) 基于共享的最近邻个数，定义两个对象之间的相似性度量。该方法基于这样一种观察，即对象和它的最近邻通常属于同一个类。该方法有助于克服高维和变密度簇的问题。

(3) 定义核心对象并构建环绕它们的簇。为了对基于图的聚类做这件事，需要引入邻近度图或稀疏化的邻近度图的基于密度概念。与 DBSCAN 一样，围绕核心对象构建簇将产生一种聚类技术，可以发现不同形状和大小的簇。

(4) 使用邻近度图中的信息，提供两个簇是否应当合并的更复杂的评估。具体地说，两个簇合并，仅当结果簇具有类似于原来的两个簇的特性。

我们从讨论邻近度图的稀疏化开始，提供两个例子，其聚类方法仅基于如下技术：MST（等价于单链聚类算法）和 Opossum。然后，我们讨论 Chameleon，一种使用自相似性（self-similarity）概念确定簇是否应当合并的层次聚类算法。接下来，我们定义一种新的相似性度量——共享的最近邻（Shared Nearest Neighbor，SNN）相似性，并介绍使用这种相似性度量的 Jarvis-Patrick 聚类算法。最后，我们讨论如何基于 SNN 相似性定义密度和核心对象，并介绍一种基于 SNN 密度的聚类算法（可以看作使用新的相似性度量的 DBSCAN）。

9.4.1 稀疏化

m 个数据点的 $m \times m$ 邻近度矩阵可以用一个稠密图表示，图中每个结点与其他所有结点相连接，任何一对结点之间边的权值反映它们之间的邻近性。尽管每个对象与其他每个对象都有某种程度的邻近性，但是对于大部分数据集，对象只与少量对象高度相似，而与大部分其他对象的相似性很弱。这一性质可以用来稀疏化邻近度图（矩阵）：在实际的聚类过程开始之前，将许多低相似度（高相异度）的值置 0。例如，稀疏化可以这样进行：断开相似度（相异度）低于（高于）

指定阈值的边，或仅保留连接到点的 k 个最近邻的边。后一种方法创建所谓 **k-最近邻图**（k-nearest neighbor graph）。

稀疏化具有下面一些有益效果。

- **压缩了数据量**。聚类所需要处理的数据量被大幅度压缩。稀疏化常常可以删除邻近度矩阵中99%以上的项。这样，可以处理的问题的规模就提高了。
- **可以更好地聚类**。稀疏化技术保持了对象与最近邻的连接，而断开了与较远对象的连接。这与**最近邻原理**（nearest neighbor principle）一致：对象的最近邻趋向于与对象在同一个类（簇）。这降低了噪声和离群点的影响，增强了簇之间的差别。
- **可以使用图划分算法**。在寻找稀疏图的最小切割划分启发式算法方面，特别是在并行计算和集成电路设计领域，研究人员已经做了大量工作。邻近度图的稀疏化使得使用图划分算法进行聚类成为可能。例如，Opossum 和 Chameleon 都使用图划分。

应当把邻近度图的稀疏化看成使用实际聚类算法之前的初始化步骤。理论上讲，完美的稀疏化应当将邻近度图划分成对应于期望簇的连通分支，但实际中这很难做到。很容易出现单条边连接两个簇，或者单个簇被分裂成若干个不相连接的子簇的情况。事实上，正如将在 9.4.6 节、9.4.7 节中讨论的那样，常常修改稀疏邻近度图，以便产生新的邻近度图。新的邻近度图还可以被稀疏化。聚类算法使用的邻近度图是所有这些预处理步骤的结果。这一过程汇总在图 9-15 中。

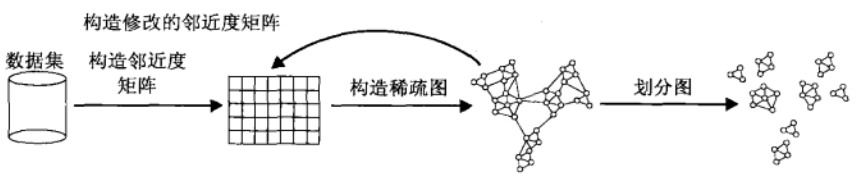


图 9-15　使用稀疏化聚类的理想过程

9.4.2　最小生成树聚类

在 8.3 节介绍凝聚层次聚类技术时，我们提到还存在分裂的层次聚类算法。在 8.2.3 节，我们看到了这种技术的一个例子，二分 K 均值。另一种分裂层次聚类技术 MST 从邻近度图的最小生成树开始，可以看作用稀疏化找出簇的应用。我们简略地讨论这个算法。有趣的是，这个算法也产生与单链凝聚聚类相同的聚类。见本章习题 13。

图的一棵**最小生成树**（Minimum Spanning Tree，MST）是一个子图，(1)它没有环，即是一棵树；(2)包含图的所有结点；(3)在所有可能的生成树中它的边的总权值最小。术语最小生成树假定我们只使用相异度或距离，我们将遵循这一约定。然而，这不是一种限制，因为我们可以将相似度转换成相异度，或者修改最小生成树的概念以使用相似度。某些二维点的最小生成树的一个例子显示在图 9-16 中。

MST 分裂层次聚类算法显示在算法 9.7 中。第一步是找出原相异度图的 MST。注意，最小生成树可以看作一种特殊类型的稀疏化图。步骤 3 也可以看作图的稀疏化。因此，MST 可以看作一种基于相异度图的稀疏化的聚类算法。

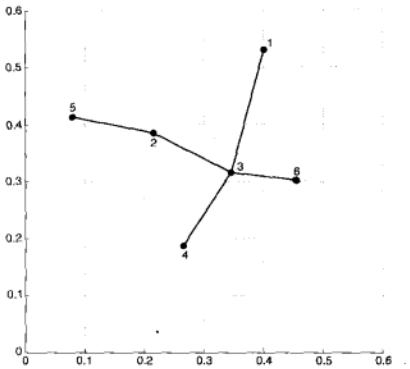


图 9-16 6 个二维点的集合的最小生成树

算法 9.7 MST 分裂层次聚类算法

1：计算相异度图的最小生成树。
2：**repeat**
3： 断开对应于最大相异度的边，创建一个新的簇。
4：**until** 只剩下单个簇。

9.4.3 OPOSSUM：使用 METIS 的稀疏相似度最优划分

OPOSSUM（Optimal Partitioning of Sparse Similarities Using METIS）是一种专门为诸如文档或购物篮数据等稀疏、高维数据设计的聚类技术。与 MST 一样，它基于邻近度图的稀疏化进行聚类。然而，OPOSSUM 使用 METIS 算法，该算法是专门为划分稀疏图设计的。OPOSSUM 的步骤在算法 9.8 中给出。

算法 9.8 OPOSSUM 聚类算法

1：计算稀疏化的相似度图。
2：使用 METIS，将相似度图划分成 k 个不同的分支（簇）。

所使用的相似性度量是适合于稀疏、高维数据的度量，如扩充的 Jaccard 度量或余弦度量。METIS 图划分程序将稀疏图划分成 k 个不同的分支，其中 k 是用户指定的参数，旨在(1)最小化分支之间边的权值（相似度），(2)实现平衡约束。OPOSSUM 使用如下两种平衡约束中的一种：(1)每个簇中的对象个数必须粗略相等，或(2)属性值的和必须粗略相等。第二种约束在有些情况下是有用的，例如当属性值表示商品价格时。

优点与缺点

OPOSSUM 简单、速度快。它将数据划分大小粗略相等的簇。根据聚类的目标，这可能看作优点或缺点。由于簇被约束为大小粗略相等，因此簇可能被分裂或合并。然而，如果使用 OPOSSUM 产生大量簇，则这些簇通常是更大簇的相对纯的片段。事实上，OPOSSUM 类似于 Chameleon 聚类过程的初始化步骤。Chameleon 在下面讨论。

9.4.4 Chameleon：使用动态建模的层次聚类

凝聚层次聚类技术通过合并两个最相似的簇来聚类，其中簇的相似性定义依赖于具体的算法。有些凝聚聚类算法，如组平均，将其相似性概念建立在两个簇之间的连接强度上（例如，两

个簇中点的逐对相似性），而其他技术，如单链方法，使用簇的接近性（例如，不同簇中点的最小距离）来度量簇的相似性。尽管有两种基本方法，但是仅使用其中一种方法可能导致错误的簇合并。考虑图 9-17，它显示了 4 个簇。如果我们使用簇的接近性（用不同簇的最近的两个点度量）作为合并标准，则我们将合并两个圆形簇(c)和(d)（它们几乎接触），而不是合并两个矩形簇(a)和(b)（它们被一个小间隔分开）。然而，直观地，我们应当合并(a)和(b)。习题 15 要求给出一个连接强度可能导致不直观结果的例子。

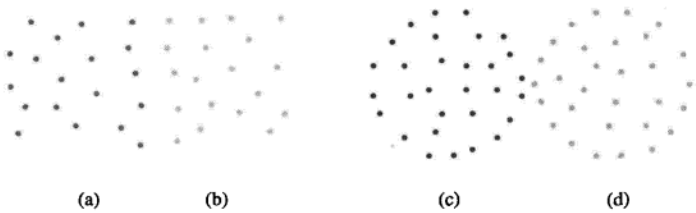


图 9-17　接近性不是适当的合并标准的情况（©1999, IEEE）

另一个问题是，大部分聚类技术都有一个全局（静态）簇模型。例如，K 均值假定簇是球形的，而 DBSCAN 基于单个密度阈值定义簇。使用这样一种全局模型的聚类方案不能处理诸如大小、形状和密度等簇特性在簇间变化很大的情况。作为簇的局部（动态）建模的重要性的一个例子，考虑图 9-18。如果我们使用簇的接近性来决定哪一对簇应当合并，例如，使用单链聚类算法，则我们将合并簇(a)和(b)。然而，我们并未考虑每个个体簇的特性。具体地说，我们忽略了个体簇的密度。对于簇(a)和(b)，它们相对稠密，两个簇之间的距离显著大于同一个簇内两个最近邻点之间的距离。对于簇(c)和(d)，就不是这种情况，它们相对稀疏。事实上，与合并簇(a)和(b)相比，簇(c)和(d)合并所产生的簇看上去与原来的簇更相似。

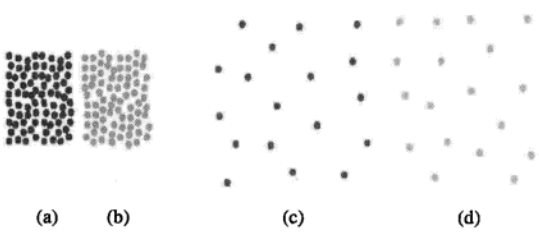


图 9-18　相对接近性概念的图示（©1999, IEEE）

Chameleon 是一种凝聚聚类技术，它解决前两段提到的问题。它将数据的初始划分（使用一种有效的图划分算法）与一种新颖的层次聚类方案相结合。这种层次聚类使用接近性和互连性概念以及簇的局部建模。它的关键思想是：仅当合并后的结果簇类似于原来的两个簇时，这两个簇才应当合并。我们首先介绍自相似性，然后提供 Chameleon 算法的其余细节。

1. 确定合并哪些簇

8.3 节考虑的凝聚层次聚类技术重复地合并两个最接近的簇，各具体技术之间的主要区别是簇的邻近度定义方式。相比之下，Chameleon 力求合并这样的一对簇，合并后产生的簇，用接近性和互连性度量，与原来的一对簇最相似。因为这种方法仅依赖于簇对而不依赖于全局模型，Chameleon 能够处理包含具有各种不同特性的簇的数据。

下面是接近性和互连性的更详细解释。为了理解这些性质，需要用邻近度图的观点，并且考虑簇内和簇间点之间的边数和这些边的强度。

- **相对接近度**（Relative Closeness，RC） 是被簇的内部接近度规范化的两个簇的绝对接近度。两个簇合并，仅当结果簇中的点之间的接近程度几乎与原来的每个簇一样。数学表述为：

$$RC(C_i,C_j)=\frac{\bar{S}_{EC}(C_i,C_j)}{\frac{m_i}{m_i+m_j}\bar{S}_{EC}(C_i)+\frac{m_j}{m_i+m_j}\bar{S}_{EC}(C_j)} \tag{9-17}$$

其中，m_i和m_j分别是簇C_i和C_j的大小；$\bar{S}_{EC}(C_i, C_j)$是连接簇C_i和C_j的（k-最近邻图的）边的平均权值；$\bar{S}_{EC}(C_i)$是二分簇C_i的边的平均权值；$\bar{S}_{EC}(C_j)$是二分簇C_j的边的平均权值；EC表示割边。图 9-18 解释了相对接近度的概念。如前所述，尽管簇(a)和(b)比簇(c)和(d)更绝对接近，但是如果考虑簇的特性，则情况并非如此。

- **相对互连度**（Relative Interconnectivity，RI） 是被簇的内部互连度规范化的两个簇的绝对互连度。如果结果簇中的点之间的连接几乎与原来的每个簇一样强，两个簇合并。数学表述为：

$$RI(C_i,C_j)=\frac{EC(C_i,C_j)}{\frac{1}{2}(EC(C_i)+EC(C_j))} \tag{9-18}$$

其中，$EC(C_i, C_j)$是连接簇C_i和C_j（k-最近邻图的）的边之和；$EC(C_i)$是二分簇C_i的割边的最小和；$EC(C_j)$是二分簇C_j的割边的最小和。图 9-19 解释了相对互连度的概念。两个圆形簇(c)和(d)比两个矩形簇(a)和(b)具有更多连接。然而，合并(c)和(d)产生的簇具有非常不同于(c)和(d)的连接性。相比之下，合并(a)和(b)产生的簇的连接性与簇(a)和(b)非常类似。

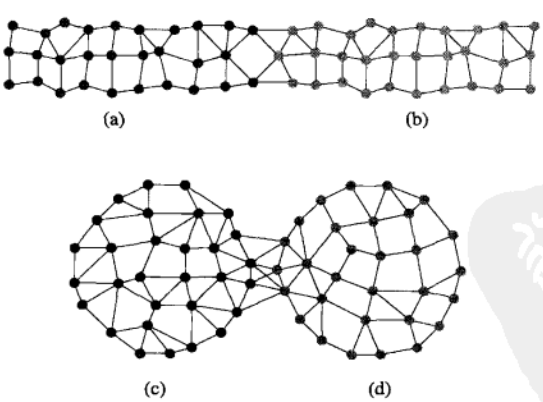


图 9-19 相对互连性概念的图示（©1999, IEEE）

RI 和 RC 可以用多种不同的方法组合，产生自相似性（self-similarity）的总度量。Chameleon 使用的一种方法是合并最大化$RI(C_i, C_j) \times RC(C_i, C_j)^{\alpha}$的簇对，其中$\alpha$是用户指定的参数，通常大于 1。

2. Chameleon 算法

Chameleon 算法由三个关键步骤组成：稀疏化、图划分和层次聚类。算法 9.9 和图 9-20 描述了这些步骤。

算法 9.9　Chameleon 算法

1：构造 k-最近邻图。
2：使用多层图划分算法划分图。
3：**repeat**
4：　合并关于相对互连性和相对接近性而言，最好地保持簇的自相似性的簇。
5：**until** 不再有可以合并的簇。

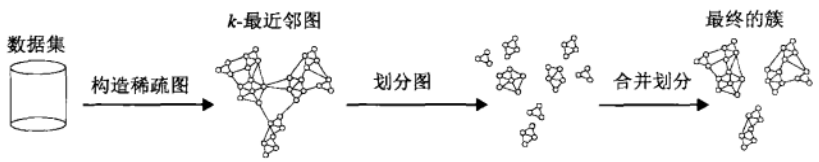


图 9-20　Chameleon 进行聚类的整个步骤（©1999, IEEE）

稀疏化　Chameleon 算法的第一步是产生 k-最近邻图。从概念上讲，这样的图由邻近度图导出，并且仅包含点和它的 k 个最近邻（即最近的点）之间的边。如前所述，使用稀疏化的邻近度图而不是完全的邻近度图可以显著地降低噪声和离群点的影响，提高计算的有效性。

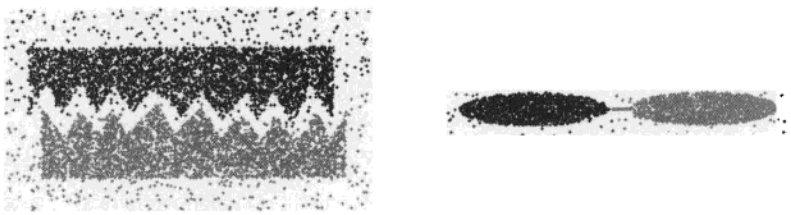
图划分　一旦得到稀疏化的图，就可以使用诸如 METIS（见文献注释）等有效的多层图划分算法来划分数据集。Chameleon 从一个全包含的图（簇）开始。然后，二分当前最大的子图（簇），直到没有一个簇多于 MIN_SIZE 个点，其中 MIN_SIZE 是用户指定的参数。这一过程导致大量大小大致相等的、良连接的顶点（高度相似的数据点）的集合。目标是确保每个划分包含的对象都大部分来自一个真正的簇。

凝聚层次聚类　如前所述，Chameleon 基于自相似性概念合并簇。可以用参数指定，让 Chameleon 一步合并多个簇对，并且在所有的对象都合并到单个簇之前停止。

3. 复杂性

假定 m 是数据点的个数，p 是划分的个数。在图划分得到的 p 个划分上进行凝聚层次聚类需要 $O(p^2 \log p)$时间（见 8.3.1 节）。划分图需要的时间总量是 $O(mp + m\log m)$。图稀疏化的时间复杂度取决于建立 k-最近邻图需要多少时间。对于低维数据，如果使用 k-d 树或类似的数据结构，则需要 $O(m\log m)$时间。然而，这种数据结构只适用于低维数据，因此，对于高维数据集，稀疏化的时间复杂度变成 $O(m^2)$。由于只需要存放 k-最近邻表，空间复杂度是 $O(km)$加上存放数据所需要的空间。

例 9.11　Chameleon 用于其他聚类算法（如 K 均值和 DBSCAN）很难聚类的两个数据集。聚类的结果在显示图 9-21 中。簇用点的明暗区分。在图 9-21a 中，两个簇具有不规则的形状，并且相当接近。此外，还有噪声。在图 9-21b 中，两个簇通过一个桥连接，并且也有噪声。尽管如此，Chameleon 还是识别出了大部分人认为自然的簇。这表明 Chameleon 对于空间数据聚类很有效。最后，注意与其他聚类方案不同，Chameleon 并不丢弃噪声点，而是把它们指派到簇中。　□



(a) (b)

图 9-21 使用 Chameleon 对两个二维点集进行聚类（©1999, IEEE）

4. 优点与局限性

Chameleon 能够有效地聚类空间数据，即便存在噪声和离群点，并且簇具有不同的形状、大小和密度。Chameleon 假定由稀疏化和图划分过程产生的对象组群是子簇，即一个划分中的大部分点属于同一个真正的簇。如果不是，则凝聚层次聚类将混合这些错误，因为它绝对不可能再将已经错误地放到一起的对象分开（见 8.3.4 节的讨论）。这样，当划分过程未产生子簇时，Chameleon 就有问题；对于高维数据，常常出现这种情况。

9.4.5 共享最近邻相似度

在某些情况下，依赖于相似度和密度的标准方法的聚类技术不能产生理想的聚类结果。本节考察这一问题的原因，并引入一种相似性的间接方法，它基于如下原理。

如果两个点都与一些相同的点相似，则即使直接的相似性度量不能指出，它们也相似。

为了进一步讨论，我们首先考察相似性的 SNN 版本解决的两个问题：低相似度和不同密度。

1. 传统的相似度在高维数据上的问题

在高维空间，相似度低并不罕见。例如，考虑如下文档集合，它包含取自报纸的不同版块的文章：娱乐、财经、国外、都市、国内和体育。正如第 2 章所讨论的，这些文档可以看作高维空间中的向量，其中向量的每个分量（属性）记录词汇表中每个词在文档中的出现次数。通常使用余弦相似性度量处理文档之间的相似性。对于这个例子（取自《洛杉矶时报》的文章的集合），表 9-3 给出了每个版块和整个文档集的平均余弦相似度。

表 9-3 报纸的不同版块文档之间的相似度

版块	平均余弦相似度
娱乐	0.032
财经	0.030
国外	0.030
都市	0.021
国内	0.027
体育	0.036
所有版块	0.014

每个文档与其最相似的文档（第一个最近邻）之间的相似性高一些，平均为 0.39。然而，同一类中对象之间低相似性的结果是，它们的最近邻也常常不在同一类。在产生表 9-3 的文档集合中，大约 20%的文档都有不同类中的最近邻。一般地说，如果直接相似度低，则对于聚类，特别是凝聚层次聚类（最近的点放在一起，并且不能再分开），相似度将成为不可靠的指导。尽管如

此，一个对象的大多数最近邻通常仍然属于同一个类；这一事实可以用来定义更适合聚类的邻近性度量。

2. 密度不同的问题

另一个问题涉及簇之间的密度不同。图 9-22 显示了一对具有不同密度点的二维簇。右边簇的较低密度反映在点之间的较低平均距离上。尽管不太稠密的簇中的点形成了同样合法的簇，但是常见的聚类技术发现这样的簇将产生更多的问题。此外，标准的凝聚度度量（如 SSE）将指出这样的簇不太凝聚。用一个实际的例子解释，与太阳系中的行星相比，银河系中的恒星更像一个恒星对象簇，尽管太阳系中的行星比银河系中的恒星的平均距离近得多。

图 9-22　200 个均匀分布的点形成的两个圆形簇

3. SNN 相似度计算

在上述两种情况下，关键思想是在定义相似性度量时考虑点的环境。这种思想可以按算法 9-10 所示的方式，用相似度的**共享最近邻**（Shared Nearest Neighbor，SNN）定义量化。本质上讲，只要两个对象都在对方的最近邻列表中，SNN相似度就是它们共享的近邻个数。注意，基本的邻近性度量可以是任何有意义的相似性或相异性度量。

算法 9.10　计算共享最近邻相似度

1：找出所有点的 *k*-最近邻。
2：**if** 两个点 **x** 和 **y** 不是相互在对方的 *k*-最近邻中 **then**
3：　*similarity*(**x**, **y**) ← 0
4：**else**
5：　*similarity*(**x**, **y**) ← 共享的近邻个数。
6：**end if**

SNN 相似度的计算在算法 9.10 中给出，而图形解释由图 9-23 给出。两个黑点都有 8 个最近邻，相互包含。这些最近邻中的 4 个（灰色点）是共享的。因此这两个点之间的 SNN 相似度为 4。

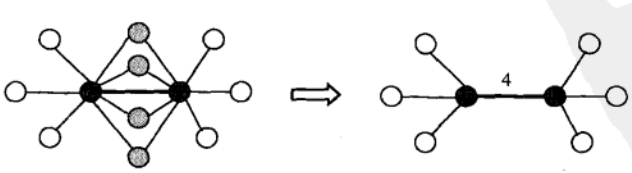


图 9-23　两个点之间 SNN 相似度的计算

对象之间 SNN 相似度的相似度图称作 **SNN 相似度图**（SNN similarity graph）。由于许多对

象对之间的 SNN 相似度为 0，因此 SNN 相似度图非常稀疏。

4. SNN 相似度与直接相似度

SNN 相似度是有用的，因为它解决了使用直接相似度出现的一些问题。首先，由于它通过使用共享最近邻的个数考虑了对象的环境，SNN 相似度可以处理如下情况：一个对象碰巧与另一个对象相对接近，但属于不同的类。在这种情况下，对象一般不共享许多近邻，并且它们的 SNN 相似度低。

SNN 相似度也能处理变密度簇的问题。在低密度区域，对象比高密度区域的对象分开得更远。然而，一对点之间的 SNN 相似度只依赖于两个对象共享的最近邻的个数，而不是这些近邻之间相距多远。这样一来，SNN 相似度关于点的密度自动进行缩放。

9.4.6 Jarvis-Patrick 聚类算法

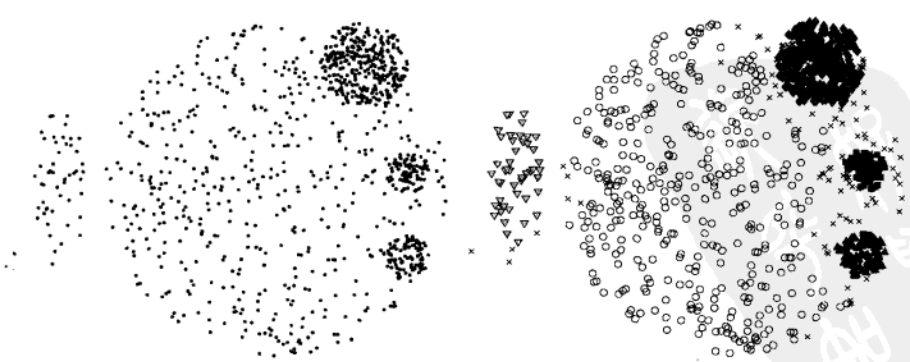
算法 9.11 给出了使用上一节概念的 Jarvis-Patrick（JP）聚类算法。JP 聚类算法用算法 9.10 计算的 SNN 相似度取代两个点之间的邻近度。然后使用一个阈值来稀疏化 SNN 相似度矩阵。使用图的术语就是，创建并稀疏化 SNN 相似度图。簇是 SNN 图的连通分支。

算法 9.11 Jarvis-Patrick 聚类算法

1：计算 SNN 相似度图。
2：使用相似度阈值，稀疏化 SNN 相似度图。
3：找出稀疏化的 SNN 相似度图的连通分支（簇）。

JP 聚类算法的存储需求仅为 $O(km)$，因为即便在初始阶段也不需要存放整个相似度矩阵。JP 聚类的基本时间复杂度是 $O(m^2)$，因为 k-最近邻列表的创建可能需要计算 $O(m^2)$个邻近度。然而，对于特定类型的数据，如低维欧几里得数据，可以使用专门的技术（如 k-d 树）来更有效地找出 k-最近邻，而不必计算整个相似度矩阵。这可以把时间复杂度从 $O(m^2)$降低到 $O(m\log m)$。

例 9.12 二维数据集的 JP 聚类 我们使用 JP 聚类算法对图 9-24a 显示的“鱼”数据集聚类，发现的簇显示在图 9-24b 中。最近邻列表的大小为 20，并且当两个点至少共享 10 个点时才将它们放到一个簇。不同的簇用不同的标记和不同的明暗度显示。标记为“x”的点被 Jarvis-Patrick 算法分类为噪声。它们大部分在不同密度的簇之间的过渡区域。 □



(a) 原数据　　(b) Jarvis-Patrick 聚类发现的簇

图 9-24 二维点集的 Jarvis-Patrick 聚类

优点与局限性

因为 JP 聚类基于 SNN 相似度概念，它擅长处理噪声和离群点，并且能够处理不同大小、形状和密度的簇。该算法对高维数据效果良好，尤其擅长发现强相关对象的紧致簇。

然而，JP 聚类把簇定义为 SNN 相似度图的连通分支。这样，一个对象集是分裂成两个簇还是作为一个簇留下，可能取决于一条链。因此 JP 聚类多少有点脆弱，即它可能分裂真正的簇，或者合并本应分开的簇。

另一个潜在的局限性是并非所有的对象都被聚类。然而，这些对象可以添加到已有的簇中，并且在某些情况下也不要求完全聚类。JP 聚类的基本时间复杂度为 $O(m^2)$，这是一般情况下计算对象集的最近邻列表所需要的计算时间。在特定情况下（例如低维数据），可以使用专门的技术将找出最近邻的时间复杂度降低到 $O(m\log m)$。最后，与其他聚类算法一样，选择好的参数值可能是一个问题。

9.4.7　SNN 密度

正如本章导论所讨论的，传统的欧几里得密度在高维空间变得没有意义。无论我们取基于网格的观点（如 CLIQUE 所采用的），基于中心的观点（如 DBSCAN 所采用的）还是核密度估计方法（如 DENCLUE 所采用的）情况都是如此。借助于在高维数据上也很成功的相似性度量（例如余弦或 Jaccard），使用基于中心的密度定义是可能的，但是正如 9.4.5 节所述，虽然这些度量也有问题。然而，由于 SNN 相似性度量反映了数据空间中点的局部结构，因此它对密度的变化和空间的维度都相对不太敏感，并且是新的密度度量的有优先的候选。

本节解释如何使用 SNN 相似度，并按照 8.4 节的 DBSCAN 方法定义 SNN 密度概念。为清楚起见，我们重复 8.4 节的定义，但是做了适当的修改，以反映我们正在使用 SNN 相似度。

- **核心点**。一个点是核心点，如果在该点给定邻域（由 SNN 相似度和用户提供的参数 *Eps* 确定）内的点数超过某个阈值 *MinPts*，其中 *MinPts* 也是用户提供的参数。
- **边界点**。边界点不是核心点（即它的邻域内没有足够的点使它成为核心点），但是它落在一个核心点的邻域内。
- **噪声点**。噪声点是既非核心点，也非边界点的任何点。

SNN 密度度量一个点被类似的点（关于最近邻）包围的程度。这样，在高密度和低密度区域的点一般具有相对较高的 SNN 密度，而在从低密度到高密度过渡的区域中的点（簇间的点）将趋向于具有低 SNN 密度。这样的方法可能更适合这样的数据集，其中密度变化很大，但是低密度的簇仍然是有趣的。

例 9.13　核心、边界和噪声点　为了更具体地讨论 SNN 密度概念，我们用一个例子说明如何使用 SNN 密度发现核心点并删除噪声和离群点。图 9-25a 显示的 2D 点的数据集包含 10 000 个点。图 9-25b～d 根据点的 SNN 密度区分了这些点。图 9-25b 显示具有高 SNN 密度的点，图 9-25c 显示具有中等 SNN 密度的点，而图 9-25d 显示具有低 SNN 密度的点。我们从这些图看到，具有高 SNN 密度（即 SNN 图中的高连接性）的点是候选代表点或核心点，因为它们大部分在簇的内部；而具有低连接性的点是候选噪声点或离群点，因为它们多半在环绕簇的区域中。　□

(a) 所有的点 (b) 高 SNN 密度 (c) 中等 SNN 密度 (d) 低 SNN 密度

图 9-25 二维点的 SNN 密度

9.4.8 基于 SNN 密度的聚类

可以将上面定义的 SNN 密度与 DBSCAN 算法结合在一起，创建一种新的聚类算法。该算法类似于 JP 聚类算法，都以 SNN 相似度图开始。然而，基于 SNN 密度的聚类算法简单地使用 DBSCAN，而不是使用阈值稀疏化 SNN 相似度图，然后取连通分支作为簇。

1. 基于 SNN 密度的聚类算法

基于 SNN 密度的聚类算法的步骤在算法 9.12 中给出。

算法 9.12 基于 SNN 密度的聚类算法

1：计算 SNN 相似度图。
2：以用户指定的参数 *Eps* 和 *MinPts*，使用 DBSCAN。

该算法自动地确定数据中的簇的个数。注意并非所有的点都被聚类。被丢弃的点包括噪声和离群点，以及没有很强地连接到一组点的那些点。基于 SNN 密度的聚类发现这样的簇，簇中的点相互之间都是强相关的。依据应用，我们可能需要丢弃许多点。例如，基于 SNN 密度的聚类对于发现文档组中的主题效果很好。

例 9.14 时间序列的基于 SNN 密度的聚类 本节提供的基于 SNN 密度的聚类算法比 Jarvis-Patrick 聚类或 DBSCAN 更加灵活。不像 DBSCAN，它可以用于高维数据和簇具有不同密度的情况。不像 Jarvis-Patrick 聚类简单地使用阈值，然后取连通分支作为簇，基于 SNN 密度的聚类使用基于 SNN 密度和核心点概念的方法。

为了表明基于 SNN 密度的聚类处理高维数据的能力，我们将它用于地球各点上的大气压时间序列数据。具体地说，该数据包含 41 年期间，在 2.5° 的经纬度网格的每一点上的月平均海平面气压（SLP）。基于 SNN 密度的聚类算法发现的簇（灰色区域）显示在图 9-26 中。注意，尽管它们可视化为二维区域，但是这些是长度为 492 个月的时间序列簇。白色区域是压力不均匀的区域。由于球面映射到矩形的扭曲，靠近两极的簇被拉长。

使用 SLP，地球科学家已经定义了时间序列，称作**气候指数**（climate indice），可以用来捕获与地球气候有关的现象的行为。例如，气候指数异常涉及世界不同地区异常低/高的降水量或气温。基于 SNN 密度的聚类发现的某些簇与地球科学家已知的某些气候指数具有很强的关联。

图 9-27 显示用于提取簇的数据的 SNN 密度结构。密度已经规范化到 0 和 1 之间。时间序列的密度看来可能是一个不寻常的概念，它测量时间序列与它的最近邻具有相同最近邻的程度。由于每个时间序列都与一个地点相关联，因此可以在二维图上绘制这些密度。由于时间的自相关性，这些密度形成了有意义的模式；例如，可以从视觉上识别图 9-27 中的簇。 □

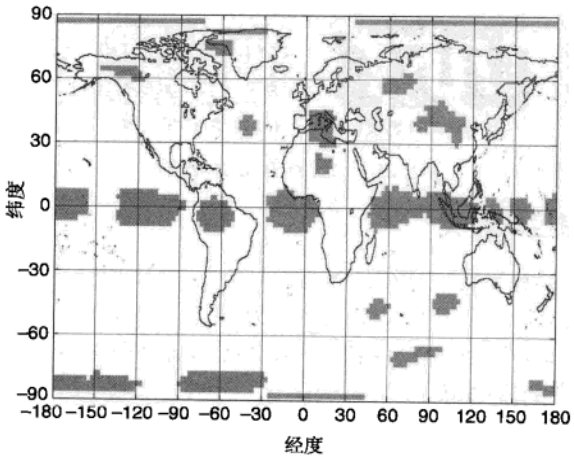


图 9-26　用基于 SNN 密度的聚类发现的气压时间序列簇

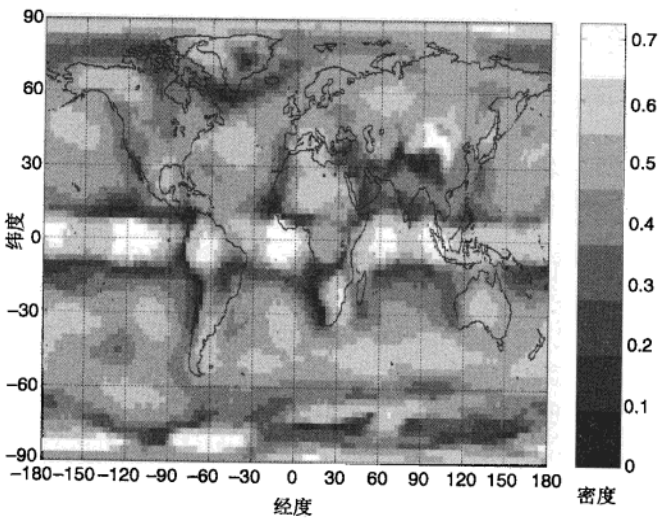


图 9-27　气压时间序列的 SNN 密度

2. 优点与局限性

基于 SNN 密度的聚类的优点与局限性类似于 JP 聚类。然而，核心点和 SNN 密度的使用大大增加了该方法的能力和灵活性。

9.5　可伸缩的聚类算法

如果运行时间长得不可接受，或者需要的存储量太大，即使最好的聚类算法也没有多大价值。本节考察着重强调可扩展到超大型数据集的聚类技术，这种超大型数据集正变得越来越常见。首先，我们讨论可伸缩的某些一般策略，包括降低邻近度计算数量的方法、数据抽样、数据划分和对数据的汇总表示聚类。然后，我们讨论两个具体的可伸缩聚类算法：CURE

和 BIRCH。

9.5.1 可伸缩：一般问题和方法

许多聚类算法所需要的存储量都是非线性的。例如，使用层次聚类，存储需求一般是 $O(m^2)$，其中 m 是对象的个数。例如，对于 10 000 000 个对象，所需要的存储量级是 10^{14}，远远超过当前系统的容量。注意，由于需要随机访问数据，许多聚类算法都很难修改，以便有效地利用二级存储器（磁盘）（对于磁盘，数据的随机访问太慢）。同样，某些聚类算法所需要的计算量也是非线性的。在本节的剩余部分，我们讨论减少聚类算法所需计算量和存储量的各种技术。CURE 和 BIRCH 使用其中某些技术。

多维或空间存取方法 许多聚类技术（K 均值、Jarvis-Patrick 聚类和 DBSCAN）需要找出最近的质心、点的最近邻或指定距离内的所有点。可以使用称作多维或空间存取方法的专门技术来更有效地执行这些任务，至少对于低维数据可以这样做。这些技术，如 k-d 树或 R*树，一般产生数据空间的层次划分，可以用来减少发现点的最近邻所需要的时间。注意，基于网格的聚类方法也划分数据空间。

邻近度界 另一种避免邻近度计算的方法是使用邻近度界。例如，使用欧几里得距离时，有可能使用三角不等式来避免许多距离的计算。例如，在传统K均值的每一阶段，需要评估点是应当留在它的当前簇，还是应当移动到一个新的簇。如果我们知道质心间的距离和点到当前所属簇的（刚更新的）质心的距离，则可以使用三角不等式来避免计算该点到其他质心的距离。见本章习题 21。

抽样 另一种降低时间复杂度的方法是抽样。在这种方法中，提取一个样本，对样本中的点进行聚类，然后将其余的点指派到已有的簇——通常是最近的簇。如果抽取的点数是 $\sqrt{m}$，则 $O(m^2)$时间复杂度的算法复杂度降低到 $O(m)$。不过，抽样的主要问题是小簇可能丢失。在讨论 CURE 时，我们将提供一种技术，考察这种问题出现的频繁程度。

划分数据对象 另一种降低时间复杂度的常用方法是，使用某种有效的技术，将数据划分成不相交的集合，然后分别对这些集合聚类。最终的簇的集合是这些分离的簇的集合的并，或者通过合并和/或对分离的簇的集合求精得到。本节，我们只讨论二分K均值（8.2.3 节），尽管许多其他基于划分的方法也能使用。在本节后面介绍CURE时，将介绍一种这样的方法。

如果使用 K 均值来找出 K 个簇，则在每次迭代时都需要计算每个点到每个簇质心的距离。如果 K 很大，则这种计算可能开销很大。二分 K 均值从整个点集合开始，使用 K 均值二次重分一个现有的簇，直到我们得到 K 个簇。在每一步，需要计算点到两个簇质心的距离。除了第一步（该步被二分的簇由所有的点组成），我们只需要计算点的一个子集到两个被考虑的质心的距离。正因为如此，二分 K 均值明显比一般的 K 均值快。

汇总 另一种聚类方法是：首先汇总数据（通常通过一遍扫描），然后在汇总数据上聚类。比如，领导者算法（见第 8 章习题 12）或者将一个数据对象放进最近的簇（如果该簇足够近），或者创建一个包含当前对象的新簇。这种方法关于对象个数是线性的，可以用来汇总数据，以便使用其他聚类技术。BIRCH 算法使用了类似的概念。

并行与分布式计算　如果不能利用前面介绍的技术，或者如果这些计算不能产生期望的精度或降低计算时间，则需要其他方法。一种高效的方法是将计算分布到多个处理机上。

9.5.2 BIRCH

BIRCH（Balanced Iterative Reducing and Clustering using Hierarchies）是一种非常有效的聚类技术，用于欧几里得向量空间数据，即平均值有意义的数据。BIRCH 能够用一遍扫描有效地对这种数据进行聚类，并可以使用附加的扫描改进聚类。BIRCH 还能够有效地处理离群点。

BIRCH 基于聚类特征（Clustering Feature，CF）和 CF 树的概念。其基本思想是，数据点（向量）的簇可以用三元组（N, LS, SS）表示，其中 N 是簇中点的个数，LS 是点的线性和，而 SS 是点的平方和。这些是常见的统计量，可以增量地更新，并且可以用来计算许多重要的量，如簇的质心及其方差（标准差）。方差用来度量簇的直径。

这些量也可以用来计算簇之间的距离。最简单的方法是计算质心之间的 L_1（城市块）或 L_2（欧几里得）距离。我们还可以用合并簇的直径（方差）作为距离。BIRCH 定义了许多不同的簇距离，但是所有的距离都可以使用这些汇总统计量来计算。

CF 树是一棵高度平衡的树。每个内部结点具有形如$[CF_i, child_i]$的项，其中 $child_i$ 是指向第 i 个子女结点的指针。每个项占用的空间和页面大小决定了内部结点的项个数。而每个项的空间由每个点的属性个数决定。

叶结点由一个聚类特征序列 CF_i 组成，其中每个聚类特征代表先前扫描过的若干点。叶结点受限于如下限制：叶结点的直径必须小于参数化的阈值 T。每个项占用的空间，连同页面大小，决定叶结点中项的个数。

通过调整阈值参数 T，可以控制树的高度。T 控制聚类的粒度，即原数据集中的数据被压缩的程度。目标是通过调整参数 T，将 CF 树保持在内存中。

CF 树在数据扫描时创建。每当遇到一个数据点，就从根结点开始遍历 CF 树，每层选择最近的结点。当识别出当前数据点的最近的叶结点（簇）时，就进行测试，检查将该数据项添加到候选簇中是否导致新簇的直径大于给定的阈值 T。如果不是，则通过更新 CF 信息将数据点添加到候选簇中。从该叶到根的所有结点的簇信息也都需要更新。

如果新簇的直径大于 T，若叶结点不满就创建一个新项，否则必须分裂叶结点。选择两个相距最远的项（簇）作为种子，而其余的项分布到两个新的叶结点中，分布基于哪个叶结点包含最近的种子簇。一旦分裂叶结点，就要更新父母结点，并且在必要时（即父母结点满时）分裂父母结点。这一过程可能继续，一直到根结点。

BIRCH 在每次分裂后跟随一个合并步。在分裂停止的内部结点，找出两个最近的项。如果这些项不对应于刚分裂产生的项，则试图合并这些项及其对应的子女结点。这一步的目的是提高空间利用率，并避免不对称的数据输入顺序带来的问题。

BIRCH 还有一个删除离群点的过程。当用尽内存而需要重建树时，可以将离群点写到磁盘。（离群点定义为包含的点远小于平均情况的结点）。在该过程的特定点，扫描离群点，看是否可以将它们吸收到树中，而不导致树增长。如果可以，则吸收它们。如果不可以，则删除它们。

除 CF 树的初始创建外，BIRCH 还包括其他许多阶段。BIRCH 的所有阶段都在算法 9.13 中简略描述。

算法 9.13 BIRCH

1：**通过创建汇总数据的 CF 树，将数据装入内存。**

2：**如果第 3 阶段需要，构造一棵较小的 CF 树。**T 增值，然后重新插入叶结点项（簇）。由于 T 已增加，某些簇将合并。

3：**进行全局聚类。**可以使用不同形式的全局聚类（使用所有簇之间的逐对距离的聚类）。然而，我们选取一种凝聚的层次技术。因为聚类特征存放了对于特定聚类类型很重要的汇总信息，可以使用全局聚类算法，就像它用于 CF 代表的簇中的所有点上一样。

4：**使用步骤 3 发现的簇质心，重新分布数据点，从而发现新的簇集合。**这克服了可能在 BIRCH 第一阶段出现的问题。由于页面大小的限制和参数 T 的缘故，应当在一个簇中的点有时可能被分裂，而应当在不同簇中的点有时可能被合并。此外，如果数据集包含重复点，则这些点根据出现次序的不同，有时可能被聚到不同的类。通过多次重复本阶段，过程将收敛到一个局部最优解。

9.5.3 CURE

CURE（Clustering Using REpresentative）是一种聚类算法，它使用各种不同的技术创建一种方法，该方法能够处理大型数据、离群点和具有非球形和非均匀大小的簇的数据。CURE 使用簇中的多个代表点来表示一个簇。理论上，这些点捕获了簇的几何形状。第一个代表点选择离簇中心最远的点，而其余的点选择离所有已经选取的点最远的点。这样，代表点自然地相对分散。选取的点的个数是一个参数，但是业已发现 10 或更大的值效果很好。

一旦选定代表点，它们就以因子 α 向簇中心收缩。这有助于减轻离群点的影响（离群点一般远离中心，因此收缩更多）。例如，一个到中心的距离为 10 个单位的代表点将移动 3 个单位（对于 $\alpha = 0.7$），而到中心距离为 1 个单位的代表点仅移动 0.3 个单位。

CURE 使用一种凝聚层次聚类方案进行实际的聚类。两个簇之间的距离是任意两个代表点（在它们向它们代表的中心收缩之后）之间的最短距离。尽管这种方案与我们看到的其他层次聚类方案不完全一样，但是如果 $\alpha = 0$，它等价于基于质心的层次聚类；而 $\alpha = 1$ 时它与单链层次聚类大致相同。注意，尽管使用层次聚类方案，但是 CURE 的目标是发现用户指定个数的簇。

CURE 利用层次聚类过程的特性，在聚类过程的两个不同阶段删除离群点。首先，如果一个簇增长缓慢，则这意味它主要由离群点组成，因为根据定义，离群点远离其他点，并且不会经常与其他点合并。在 CURE 中，离群点删除的第一个阶段一般出现在簇的个数是原来点数的 1/3 时。第二个离群点删除阶段出现在簇的个数达到 K（期望的簇个数）的量级时。此时，小簇又被删除。

由于 CURE 在最坏情况下的复杂度为 $O(m^2 \log m)$，它不能直接用于大型数据集。因此 CURE 使用了两种技术来加快聚类过程。第一种技术是取随机样本，并在抽样的数据点上进行层次聚类。随后是最终扫描，通过选择具有最近代表点的簇，将数据集中剩余的点指派到簇中。我们稍后更详细地讨论 CURE 的抽样方法。

在某些情况下，聚类所需要的样本仍然太大，需要第二种附加的技术。在这种情况下，CURE 划分样本数据，然后聚类每个划分中的点。这种预聚类步后通常紧随中间簇的聚类，以及将数据集中的每个点指派到一个簇的最终扫描。CURE 的划分方案稍后也将更详细地讨论。

算法 9.14 总结了 CURE。注意，K 是期望的簇个数，m 是点的个数，p 是划分的个数，而 q 是一个划分中的点的期望压缩，即一个划分中的簇的个数是 $\frac{m}{pq}$。因此，簇的总数是 $\frac{m}{q}$。例如，如果 $m = 10\,000$，$p = 10$ 并且 $q = 100$，则每个划分包含 10 000/10 = 1 000 个点，每个划分有

1 000/100 = 10 个簇，而总共有 10 000/100 = 100 个簇。

算法 9.14　CURE

1：**由数据集抽取一个随机样本**。值得注意的是，CURE 的文章中有明确的公式，指出为了以较高的概率确保所有的簇都被最少的点代表，样本应当多大。

2：**将样本划分成 p 个大小相等的划分。**

3：**使用 CURE 的层次聚类算法，将每个划分中的点聚类成 $\frac{m}{pq}$ 个簇，得到总共 $\frac{m}{q}$ 个簇**。注意，在此处理过程中将删除某些离群点。

4：**使用 CURE 的层次聚类算法对上一步发现的 $\frac{m}{q}$ 个簇进行聚类，直到只剩下 k 个簇。**

5：**删除离群点**。这是删除离群点的第二阶段。

6：**将所有剩余的数据点指派到最近的簇，得到完全聚类。**

1. CURE 的抽样

使用抽样的一个关键问题是样本是否具有代表性，即它是否捕获了感兴趣的特性。对于聚类，该问题是我们是否能够在样本中发现与在整个对象集中相同的簇。理想情况下，我们希望对于每个簇，样本都包含一些对象；并且对于整个数据集中属于不同簇的对象，在样本中也在不同的簇中。

一个更具体的和可达到的目标是（以较高的概率）确保每个簇至少有一些点。这样的样本所需要的点的个数因数据集而异，并且依赖于对象的个数和簇的大小。CURE 的创建者推导出了一个样本大小的界，指出为了（以较高的概率）确保我们从每个簇至少得到一定数量的点，样本应当多大。使用本书的记号，这个界由如下定理给出。

定理 9.1　设 f 是一个分数，$0 \leqslant f \leqslant 1$。对于大小为 m_i 的簇 C_i，我们将以概率 $1-\delta$（$0 \leqslant \delta \leqslant 1$）从簇 C_i 得到至少 $f \times m_i$ 个对象，如果样本的大小 s 由下式给出：

$$s = fm + \frac{m}{m_i} * \log\frac{1}{\delta} + \frac{m}{m_i}\sqrt{\log\frac{1^2}{\delta} + 2 * f * m_i * \log\frac{1}{\delta}} \tag{9-19}$$

其中，m 是对象的个数。

这个表达式看上去有点吓人，但是相当容易使用。假定有 100 000 个对象，我们的目标是以 80%的可能性得到 10%的 C_i 簇对象，其中 C_i 的大小是 1 000。在此情况下，$f = 0.1$，$\delta = 0.2$，$m =$ 100 000，这样 $s = 11\,962$。如果目标是得到 5%的 C_i 簇对象，其中 C_i 有 50 个对象，则大小为 6 440 的样本就足够了。

再次说明，CURE 以如下方式使用抽样。首先抽取一个样本，然后使用 CURE 对该样本进行聚类。找到簇之后，将每个未聚类的点指派到最近的簇。

2. 划分

当抽样不够时，CURE 还使用划分方法。其基本思想是，将点划分成 p 个大小为 m/p 的组，使用 CURE 对每个划分聚类，将对象的个数压缩一个因子 $q > 1$，其中 q 可以粗略地看作划分中的簇的平均大小。总共产生 m/q 个簇。（注意，由于 CURE 用多个代表表示一个簇，因此对象个数的压缩量不是 q）。然后，预聚类后随 m/q 个中间簇的最终聚类，产生期望的簇个数（K）。两遍聚类都使用 CURE 的层次聚类算法，而最后一遍将数据集中的每个点指派到一个簇。

关键的问题是如何选取 p 和 q。像 CURE 这样的算法时间复杂度为 $O(m^2)$或更高，并且还需要将所有的数据放在内存。因此，我们希望选择尽可能小的 p，使得整个划分可以以“合理的”时间在内存处理。当前，常见的台式计算机几秒内可以对几千个对象进行聚类。

选取 p 和 q 的另一个因素涉及聚类质量。具体地说，目标是选取 p 和 q 的值，使得同一基本簇的对象最终在一个簇中。为了解释这一点，假定有 1 000 个对象和一个大小为 100 的簇。如果我们随机地产生 100 个划分，则在平均情况下，每个划分只有一个点来自我们的簇。这些点很可能与来自其他簇的点放到一个簇中，或者被当作离群点丢弃。如果我们只产生 10 个 100 个对象的划分，但是 q 是 50，则每个簇的 10 个点（平均情况）仍然可能与其他簇的点合并，因为每个簇只有 10 个点（平均情况），并且我们要为每个划分产生两个簇。为了避免后一个涉及 q 的适当选择问题，我们建议如果簇过于不相似，就不合并簇。

9.6 使用哪种聚类算法

在确定使用哪种类型的聚类算法时，需要考虑各种各样的因素。其中许多因素已经在本章和前几章讨论过。本节的目的是简洁地总结这些因素，清楚地显示对于特定的聚类任务，哪种聚类算法更合适。

聚类的类型 确定聚类的类型与预期的使用相匹配的一个重要因素是算法产生的聚类类型。对于一些应用，如创建生物学分类法，层次是首选的。对于旨在汇总的聚类，划分聚类是常用的。对于其他应用，两种都可能是有用的。

大部分聚类应用要求所有（或几乎所有）对象的聚类。例如，如果使用聚类组织用于浏览的文档集，则我们希望大部分文档都属于一个组。然而，如果我们想要找出文档集合中的最重要的主题，则我们更愿意有一个只产生凝聚的簇的聚类方案，即使许多文档未被聚类也没有关系。

最后，大部分聚类应用都假定每个对象都被指派到一个簇（或层次方案某层上的一个簇）。然而，我们已经看到，概率和模糊方案提供了指明对象在各簇的概率或隶属度的权值。其他技术，如 DBSCAN 和基于 SNN 密度的聚类，具有核心点概念。核心点强属于一个簇。在特定应用中，这些概念可能是有用的。

簇的类型 另一个重要方面是，簇的类型是否与应用匹配。经常遇到的簇有三种类型：基于原型的、基于图的和基于密度的。基于原型的聚类方案以及某些基于图的聚类方案（全链、质心和 Ward）易于产生全局簇，其中每个对象都与簇的原型或簇中其他对象足够靠近。例如，如果我们想汇总数据以压缩它的大小，并且我们希望以最小的误差做这件事，则这些技术类型应当最适合。相比之下，基于密度的聚类技术和某些基于图的聚类技术（如单链）易于产生非全局的簇，因而包含许多相互之间不很相似的对象。如果使用聚类根据地表覆盖将地理区域划分成毗邻的区域，则这些技术比基于原型的方法（如 K 均值）更合适。

簇的特性 除一般的簇类型之外，簇的其他特性也很重要。如果我们想在原数据空间的子空间中发现簇，则必须选择像 CLIQUE 这样的算法，显式地寻找这样的簇。同样，如果我们对强化簇之间的空间联系感兴趣，则 SOM 或某些相关的方法更合适。此外，对于处理形状、大小和密度变化的簇，聚类算法的能力也有很大区别。

数据集和属性的特性 正如在导论中所讨论的，数据集和属性的类型可能决定所用算法的类

型。例如，K均值算法只能用于这样的数据：有合适的邻近性度量，使得簇质心的计算是有意义的。对于其他聚类技术，如许多凝聚层次聚类方法，只要可以创建邻近度矩阵，数据集和属性的基本性质就不那么重要。

噪声和离群点　噪声和离群点是数据特别重要的方面。我们一直试图指出噪声和离群点对我们讨论的各种聚类算法的影响。然而在实践中，估计数据集中的噪声量或离群点的个数可能是非常困难的。此外，对一个人而言是噪声或离群点的东西，对另一个人可能是有意义的。例如，如果我们使用聚类将一个区域划分成人口密度不同的区域，则我们不愿意使用诸如DBSCAN那样的基于密度的技术，因为它假定密度低于全局阈值的区域或点是噪声或离群点。作为另一个例子，诸如CURE这样的层次聚类方案通常丢弃增长缓慢的点簇，因为这样的簇更适合代表离群点。然而，在某些应用中，我们可能对相对小的簇最感兴趣；例如，在市场分割中，这样的组群可能代表最有利可图的顾客。

数据对象的个数　在前几节，我们已经非常详细地考虑了数据对象的个数对聚类的影响。我们重申，在决定使用聚类算法的类型时，这个因素起重要作用。假设我们想创建数据集的一个层次聚类，我们对一路扩展到每个对象的完全层次聚类不感兴趣，而只对将数据分裂成数百个簇的那些点感兴趣。如果该数据集非常大，则我们不能直接使用凝聚聚类技术。然而，我们可以使用分裂聚类技术，如最小生成树（MST）算法（类似于单链的分裂算法），但是这仅当数据集不是太大时才是可行的。二分K均值也可以处理许多数据集，但是如果数据集太大，不能完全放入内存，则这种方法可能会遇到问题。在这种情况下，像BIRCH这样的不要求数据都在内存的技术就变得更有用。

属性的个数　我们已讨论了某种长度的维度的影响。关键是要认识到，在低维和适度维上运行很好的算法在高维空间可能无法运行。正如其他不适当地使用聚类算法的情况那样，聚类算法可能运行并产生簇，但是这些簇可能并不代表数据的真实结构。

簇描述　聚类技术常常被忽视的一个方面是如何描述结果簇。原型簇由簇原型的一个小集合简洁地描述。对于混合模型，簇被少量参数（如均值向量和协方差矩阵）的集合描述。这也是非常紧凑和容易理解的表示。对于SOM，一般可以把簇之间的联系可视化地显示在一个如图9-8那样的二维图中。然而，对于基于图和基于密度的聚类方法，簇通常用簇成员的集合描述。尽管如此，在CURE中，也可以用（相对）较小的代表点的集合描述簇。此外，对于基于网格的方案（如CLIQUE），可以使用描述簇中网格单元的属性值上的条件，产生更紧凑的描述。

算法考虑　算法也有需要考虑的重要方面。算法是非确定性的还是次序依赖的？算法自动地确定簇的个数吗？是否存在某种技术确定各种参数的值？许多聚类算法试图通过最优化一个目标函数来解决聚类问题。该目标与应用目标匹配吗？如果不，即使算法做了很好的工作，发现了关于目标函数最优或接近最优的聚类，结果仍然没有意义。此外，大部分目标函数以牺牲较小的簇为代价，偏向于较大的簇。

小结　选择合适的聚类算法涉及对所有这些问题，以及特定领域问题的考虑。不存在确定合适技术的公式。尽管如此，可用的关于聚类技术类型的一般知识和对上述问题的考虑，连同对实际应用的密切关注，应当可以帮助数据分析者做出试用哪种（或哪些）聚类方法的决策。

文献注释

模糊聚类的广泛讨论，包括模糊c均值的描述和9.2.1节提供的公式的形式推导可以在Höppner等[441]关于模糊聚类分析的书中找到。尽管本章没有讨论，Cheeseman等[424]的AutoClass是最早的、最著名的混合模型聚类程序之一。混合模型导论可以在以下文献中找到：Bilmes[420]的指南，Mitchell[450]的书（它还介绍了如何从混合模型方法推导出K均值算法），Fraley和Raftery[429]的文章。

除数据探查之外，SOM 和它的监督学习版本，学习向量量化（Learning Vector Quantization, LVQ）已经用于许多任务：图像分割、文档文件的组织和语音处理。我们的 SOM 讨论使用了基于原型的聚类的术语。Kohonen 等[447]的书包含了 SOM 的广泛介绍，侧重它的神经网络起源，以及它的一些变形和应用。一种重要的 SOM 相关的聚类开发是 Bishop 等[421]的生成地形图（Generative Topographic Map，GTM）算法。该算法使用 EM 算法找出满足二维地形约束的高斯模型。

Chameleon 的介绍可以在 Karypis 等[445]的文章中找到。尽管不等价于 Chameleon，类似的功能已经在 Karypis 等[425]的 CLUTO 聚类软件包中实现。Karypis 和 Kumar[446]的 METIS 图划分软件包用于这两个程序进行图划分，同时还用于 Strehl 和 Ghosh[459]的 OPOSSUM 聚类算法。Jarvis 和 Patrick[442]引进 SNN 相似度概念。Gowda 和 Krishna[434]提出了一种基于共有最近邻的类似概念的层次聚类方案。Guha 等[437]创建了 ROCK——一种用于事务数据聚类的层次的、基于图的聚类算法。ROCK 也使用一种共享近邻的概念，非常像 Jarvis 和 Patrick 提出的 SNN 相似性。基于 SNN 密度的聚类技术的介绍可以参考 Ertöz 等[426, 427]的文章。Steinbach 等[457]使用基于 SNN 密度的聚类来发现气候指数。

基于网格的聚类算法的例子有 OptiGrid（Hinneburg 和 Keim[440]）、BANG 聚类系统（Schikuta 和 Erhart[455]）和 WaveCluster（Sheikholeslami 等[456]）。Guha 等[418]的文章介绍了 CLIQUE 算法。MAFIA（Nagesh 等[452]）是对 CLIQUE 的修改，目标是提高效率。Kailing 等[444]开发了 SUBCLU（density-connected SUBspace CLUstering）——一种基于 DBSCAN 的子空间聚类算法。DENCLUE 算法由 Hinneburg 和 Keim[439]提出。

我们的可伸缩讨论深受 Ghosh[432]的文章影响。用于大规模数据集聚类的专门技术的广泛讨论可以在 Murtagh[451]的文章中找到。CURE 是 Guha 等[436]设计的，而 BIRCH 的细节可参见 Zhang 等[460] 的文章。CLARANS（Ng 和 Han[453]）是一种把 K 中心点聚类伸缩到更大数据集的算法。将 EM 和 K 均值聚类扩展到更大数据集的讨论由 Bradley 等[422, 423]提供。

关于聚类，有许多问题我们没有提及。上一章文献注释提到的书和综述提供了附加的线索。这里，我们提及了四个领域，由于篇幅有限忽略了很多。事务数据聚类（Ganti 等[430]、Gibson 等[433]、Han 等[438]、Peters 和 Zaki[454]）是一个重要的领域，因为事务数据常见并在商业上具有重要意义。随着通信和传感器网络的普及，流数据也变得日趋普遍和重要。数据流聚类的两篇导论性文章是 Barbará[419]和 Guha 等[435]。概念聚类（Fisher 和 Langley[428]、Jonyer 等[443]、Mishra 等[449]，Michalski 和 Stepp[448]以及 Stepp 和 Michalski[458]）使用更复杂的簇定义，通常可以更好地应用于人类的簇概念。概念聚类是一个潜能或许还未被完全认识的聚类领域。最后，在向量量化领域，存在大量旨在压缩数据的聚类工作。Gersho 和 Gray[431]的书是该领域的标准教科书。

参考文献

[418] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of 1998 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 94 - 105, Seattle, Washington, June 1998. ACM Press.

[419] D. Barbará. Requirements for clustering data streams. *SIGKDD Explorations Newsletter*, 3(2):23 - 27, 2002.

[420] J. Bilmes. A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report ICSITR-97-021, University of California at Berkeley, 1997.

[421] C. M. Bishop, M. Svensen, and C. K. I. Williams. GTM: A principled alternative to the self-organizing map. In C. von der Malsburg, W. von Seelen, J. C. Vorbruggen, and B. Sendhoff, editors, *Artificial Neural Networks—ICANN96. Intl. Conf, Proc.*, pages 165 - 170. Springer-Verlag, Berlin, Germany, 1996.

[422] P. S. Bradley, U. M. Fayyad, and C. Reina. Scaling Clustering Algorithms to Large Databases. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 9 - 15, New York City, August 1998. AAAI Press.

[423] P. S. Bradley, U. M. Fayyad, and C. Reina. Scaling EM (Expectation Maximization) Clustering to Large Databases. Technical Report MSR-TR-98-35, Microsoft Research, October 1999.

[424] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. AutoClass: a Bayesian classification system. In *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*, pages 431 - 441. Morgan Kaufmann Publishers Inc., 1993.

[425] CLUTO 2.1.1: Software for Clustering High-Dimensional Datasets. /www.cs.umn.edu/ ~karypis, November 2003.

[426] L. Ertöz, M. Steinbach, and V. Kumar. A New Shared Nearest Neighbor Clustering Algorithm and its Applications. In *Workshop on Clustering High Dimensional Data and its Applications, Proc. of Text Mine'01, First SIAM Intl. Conf. on Data Mining, Chicago, IL, USA*, 2001.

[427] L. Ertöz, M. Steinbach, and V. Kumar. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. In *Proc. of the 2003 SIAM Intl. Conf. on Data Mining*, San Francisco, May 2003. SIAM.

[428] D. Fisher and P. Langley. Conceptual clustering and its relation to numerical taxonomy. *Artificial Intelligence and Statistics*, pages 77 - 116, 1986.

[429] C. Fraley and A. E. Raftery. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578 - 588, 1998.

[430] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS - Clustering Categorical Data Using Summaries. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 73 - 83. ACM Press, 1999.

[431] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*, volume 159 of *Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1992.

[432] J. Ghosh. Scalable Clustering Methods for Data Mining. In N. Ye, editor, *Handbook of Data Mining*, pages 247 - 277. Lawrence Ealbaum Assoc, 2003.

[433] D. Gibson, J. M. Kleinberg, and P. Raghavan. Clustering Categorical Data: An Approach Based on Dynamical Systems. *VLDB Journal*, 8(3 - 4):222 - 236, 2000.

[434] K. C. Gowda and G. Krishna. Agglomerative Clustering Using the Concept of Mutual Nearest Neighborhood. *Pattern Recognition*, 10(2):105 - 112, 1978.

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

[436] S. Guha, R. Rastogi, and K. Shim. CURE:An Efficient Clustering Algorithm for Large Databases. In

Proc. of 1998 ACM-SIGMOD Intl. Conf. on Management of Data, pages 73 - 84. ACM Press, June 1998.

[437] S. Guha, R. Rastogi, and K. Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. In *Proc. of the 15th Intl. Conf. on Data Engineering*, pages 512 - 521. IEEE Computer Society, March 1999.

[438] E.-H. Han, G. Karypis, V. Kumar, and B. Mobasher. Hypergraph Based Clustering in High-Dimensional Data Sets: A Summary of Results. *IEEE Data Eng. Bulletin*, 21 (1):15 - 22, 1998.

[439] A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proc. of the 4th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 58 - 65, New York City, August 1998. AAAI Press.

[440] A. Hinneburg and D. A. Keim. Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. In *Proc. of the 25th VLDB Conf.*, pages 506 - 517, Edinburgh, Scotland, UK, September 1999. Morgan Kaufmann.

[441] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. JohnWiley & Sons, New York, July 2 1999.

[442] R. A. Jarvis and E. A. Patrick. Clustering Using a Similarity Measure Based on Shared Nearest Neighbors. *IEEE Transactions on Computers*, C-22(11):1025 - 1034, 1973.

[443] I. Jonyer, D. J. Cook, and L. B. Holder. Graph-based hierarchical conceptual clustering. *Journal of Machine Learning Research*, 2:19 - 43, 2002.

[444] K. Kailing, H.-P. Kriegel, and P. Kröger. Density-Connected Subspace Clustering for High-Dimensional Data. In *Proc. of the 2004 SIAM Intl. Conf. on Data Mining*, pages 428 - 439, Lake Buena Vista, Florida, April 2004. SIAM.

[445] G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *IEEE Computer*, 32(8):68 - 75, August 1999.

[446] G. Karypis and V. Kumar. Multilevel k-way Partitioning Scheme for Irregular Graphs. *Journal of Parallel and Distributed Computing*, 48(1):96 - 129, 1998.

[447] T. Kohonen, T. S. Huang, and M. R. Schroeder. *Self-Organizing Maps*. Springer- Verlag, December 2000.

[448] R. S. Michalski and R. E. Stepp. Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):396 - 409, 1983.

[449] N. Mishra, D. Ron, and R. Swaminathan. A New Conceptual Clustering Framework. *Machine Learning Journal*, 56(1 - 3):115 - 151, July/August/September 2004.

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

[451] F. Murtagh. Clustering massive data sets. In J. Abello, P. M. Pardalos, and M. G. C. Reisende, editors, *Handbook of Massive Data Sets*. Kluwer, 2000.

[452] H. Nagesh, S. Goil, and A. Choudhary. Parallel Algorithms for Clustering High-Dimensional Large-Scale Datasets. In R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. Namburu, editors, *Data Mining for Scientific and Engineering Applications*, pages 335 - 356. Kluwer Academic Publishers, Dordrecht, Netherlands, October 2001.

[453] R. T. Ng and J. Han. CLARANS: A Method for Clustering Objects for Spatial Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003 - 1016, 2002.

[454] M. Peters and M. J. Zaki. CLICKS: Clustering Categorical Data using K-partite Maximal Cliques. In *Proc. of the 21st Intl. Conf. on Data Engineering*, Tokyo, Japan, April 2005.

[455] E. Schikuta and M. Erhart. The BANG-Clustering System: Grid-Based Data Analysis. In *Advances in Intelligent Data Analysis, Reasoning about Data, Second Intl. Symposium, IDA-97, London*, volume 1280 of *Lecture Notes in Computer Science*, pages 513 - 524. Springer, August 1997.

[456] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proc. of the 24th VLDB Conf.*, pages 428 - 439, New York City, August 1998. Morgan Kaufmann.

[457] M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, and C. Potter. Discovery of climate indices using

clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 446 - 455, New York, NY, USA, 2003. ACM Press.

[458] R. E. Stepp and R. S. Michalski. Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence*, 28(1):43 - 69, 1986.

[459] A. Strehl and J. Ghosh. A Scalable Approach to Balanced, High-dimensional Clustering of Market-Baskets. In *Proc. of the 7th Intl. Conf. on High Performance Computing (HiPC 2000)*, volume 1970 of *Lecture Notes in Computer Science*, pages 525 - 536, Bangalore, India, December 2000. Springer.

[460] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proc. of 1996 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 103 - 114, Montreal, Quebec, Canada, June 1996. ACM Press.

习 题

1. 对于稀疏数据，讨论为什么只考虑非零值的存在性给出的对象视图可能比考虑实际值的大小更准确。什么时候该方法不是所期望的？
2. 描述随着待发现的簇的个数增加，K 均值的时间复杂度的变化。
3. 考虑一个文档集。假定所有的文档已经规范化，具有单位长度 1。包含到质心的余弦相似度大于某个指定常数（即 $\cos(d, c) \geq \delta$，其中 $0 < \delta \leq 1$）的所有文档的簇是什么“形状”？
4. 讨论将聚类问题处理成最优化问题的优点和缺点。在其他因素中，考虑有效性、非确定性，以及基于最优化的方法是否捕获了所有感兴趣的聚类类型。
5. 模糊 c 均值的时间和空间复杂度是多少？SOM 呢？这些复杂度与 K 均值的复杂度相比较如何？
6. 传统的 K 均值具有许多局限性，例如对离群点敏感、难以处理不同大小和不同密度或具有非球形形状的簇。评述模糊 c 均值处理这些问题的能力。
7. 对于本书描述的模糊 c 均值算法，任何点在所有簇中的隶属度之和为 1。我们也可以只要求点在一个簇的隶属度在 0 和 1 之间。这种方法的优点和缺点是什么？
8. 解释似然与概率的区别。
9. 公式（9-12）将取自高斯分布的点集的似然作为均值μ和标准差σ的函数。从数学上证明μ和σ的最大似然估计分别是样本的均值和样本的标准差。
10. 我们取一个成年人的样本并度量他们的身高。如果我们记录每个人的性别，则我们可以分别计算男人和女人的平均身高和身高的方差。然而，如果没有记录性别信息，仍然有可能得到这一信息吗？解释原因。
11. 比较图 9-1 和图 9-4 的隶属权值和概率，它们分别来自对相同的数据点集使用模糊和 EM 聚类。你发现了什么差别，如何解释这些差别？
12. 图 9-28 显示具有两个簇的二维点集的聚类。左边的簇（点用星号标记）多少有点散开，而右边的簇（点用圆标记）是紧凑的。在紧凑簇的右边有一个单独的点（用箭头指出）属于散开的簇。该簇的中心比紧凑簇的中心远得多。解释为什么用 EM 聚类这是可能的，但是用 K 均值聚类不可能。

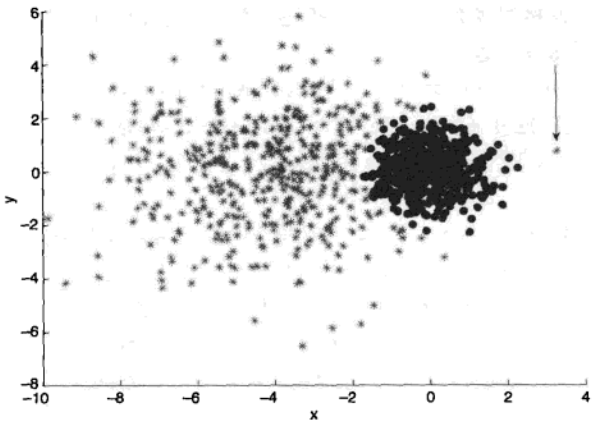


图 9-28　习题 12 的数据集。具有两个不同密度的簇的二维点集的 EM 聚类

13. 证明 9.4.2 节的 MST 聚类技术与单链产生相同的簇。为了避免复杂化和特殊情况，假定所有的逐对相似度都不相同。
14. 一种稀疏化邻近度矩阵的方法如下：对于每个对象（矩阵的行），除对应于对象的 k-最近邻的项之外，所有的项都设置为 0。然而，稀疏化之后的邻近度矩阵一般不是对称的。
    (a) 如果对象 a 在对象 b 的 k-最近邻中，为什么不能保证 b 在对象 a 的 k-最近邻中？
    (b) 至少建议两种方法，可以用来使稀疏化后的矩阵是对称的。
15. 给出一个簇集合的例子，其中基于簇的接近性的合并得到的簇集合比基于簇的连接强度（互连性）的合并得到的簇集合更自然。
16. 表 9-4 列出了 4 个点的两个最近邻。使用算法 9.10 定义的 SNN 相似度定义，计算每对点之间的 SNN 相似度。

表 9-4　4 个点的两个最近邻

点	第一个近邻	第二个近邻
1	4	3
2	3	4
3	4	2
4	3	1

17. 对于算法 9.10 提供的 SNN 相似度定义，SNN 距离的计算没有考虑两个最近邻表中共享近邻的位置。换言之，可能希望给予以相同或粗略相同的次序共享相同的最近邻的两个点以更高的相似度。
    (a) 描述如何修改 SNN 相似度定义，给予以粗略相同的次序共享近邻的两个点以更高的相似度。
    (b) 讨论这种修改的优点和缺点。
18. 至少列举一种你不想使用基于 SNN 相似度或密度的聚类的情况。
19. 网格聚类技术不同于其他聚类技术，它们划分空间而不是点的集合。
    (a) 这样的技术对于结果簇的描述和可以发现的簇的类型有何影响？

(b) 使用基于网格的聚类可以发现的哪些簇类型不能使用其他类型的聚类方法发现（提示：见第8章习题20）？

20. 在 CLIQUE 中，当维数增加时，用来发现簇密度的阈值仍为常数。这是一个潜在的问题，因为密度随维度增加而下降；即，为了发现较高维的簇，阈值应该设置到可能导致低维簇合并的水平。评论你是否认为这的确是一个问题，如果是，如何修改 CLIQUE 来解决该问题。

21. 给定欧几里得空间中一个点集，以欧几里得距离使用 K 均值对它进行聚类。在赋值时，可以使用三角不等式避免计算每个点到每个簇质心的距离。给出如何做的一般步骤。

22. 除了使用 CURE 推导出的公式（见公式（9-19）），我们可以运行蒙特卡洛模拟来直接估计大小为 s 的样本至少包含一个簇的一定比例的样本的概率。使用蒙特卡洛模拟计算大小为 s 的样本包含一个大小为 100 的簇中 50%元素的概率，其中总点数为 1 000，而 s 可以取 100、200 和 500。

第10章

异常检测

异常检测的目标是发现与大部分其他对象不同的对象。通常，异常对象被称作**离群点**（outlier），因为在数据的散布图中，它们远离其他数据点。异常检测也称**偏差检测**（deviation detection），因为异常对象的属性值明显偏离期望的或常见的属性值。异常检测也称**例外挖掘**（exception mining），因为异常在某种意义上是例外的。本章我们主要使用术语异常或离群点。

有各种各样的异常检测方法，这些方法来自多个领域，包括统计学、机器学习和数据挖掘。所有这些都试图捕获这样的思想：异常的数据对象是不寻常的，或者在某些方面与其他对象不一致。尽管根据定义，不寻常的对象或事件是相对罕见的，但是这并不表示它们绝对不常出现。例如，当所考虑的事件数多达数十亿时，可能性为“千分之一”的事件也可能出现数百万次。

在自然界、人类社会或数据集领域，大部分事件和对象，按定义都是平凡的或平常的。然而，我们应当敏锐地意识到不寻常或不平凡的对象存在的可能性。这包括异常干旱或多雨的季节，著名的运动员，或比其他值小得多或大得多的属性值。我们对异常事件或对象的兴趣源于如下事实：它们通常具有异乎寻常的重要性。干旱威胁农作物，运动员的异常能力可能最终取胜，实验结果的异常值可能指出实验中的问题或需要研究的新现象。

下面的例子阐明在一些应用中异常是相当有趣的。

- **欺诈检测**。盗窃信用卡的人的购买行为可能不同于信用卡持有者。信用卡公司试图通过寻找窃贼的购买模式，或通过注意不同于常见行为的变化来检测窃贼。类似的方法可以用于其他类型的欺诈检测。
- **入侵检测**。不幸的是，对计算机系统和网络系统的攻击已是常事。某些攻击是显而易见的，如旨在瘫痪或控制计算机和网络的攻击；但是其他攻击，如旨在秘密收集信息的攻击则很难检测。许多入侵只能通过监视系统和网络的异常行为来检测。
- **生态系统失调**。在自然界，存在一些非常见的事件，对人类具有重大影响。例子包括飓风、洪水、干旱、热浪和火灾。目标通常是预测这些事件的似然度和它们的成因。
- **公共卫生**。在许多国家，医院和医疗诊所向国家机构报告各种统计数据，以供进一步分析。例如，如果一座城市的所有孩子都接种某种特定疾病（如麻疹）的疫苗，则散布在城市各医院的少量病例是异常事件，可能指示该城市疫苗接种程序方面的问题。
- **医疗**。对于特定的患者，不寻常的症状或检查结果可能指出潜在的健康问题。然而，一个特定的检查结果是否异常可能依赖患者的其他特征，如性别和年龄。此外，对结果分类为异常与否会付出某种代价——如果患者是健康的，代价是不必要的进一步检查；如果病情未诊断出来和未予治疗，代价是对患者的潜在伤害。

尽管当前感兴趣的异常检测多半是由关注异常的应用驱动的，但是历史上异常检测（和消除）一直被视为一种旨在改进常见数据对象分析的技术。例如，相对少的离群点可能扭曲一组值的均值和标

准差，或者改变聚类算法产生的簇的集合。因此，异常检测（和消除）通常是数据预处理的一部分。

本章我们将集中讨论异常检测。先介绍少量预备知识，然后我们详细讨论一些重要的异常检测方法，并用具体技术的例子解释它们。

10.1　预备知识

在着手讨论具体的异常检测算法之前，我们提供某些附加的背景材料。具体地说，(1)我们考察异常的成因，(2)考虑各种异常检测方法，(3)根据是否使用类标号考察方法之间的差别，(4)介绍异常检测技术的常见问题。

10.1.1　异常的成因

下面是一些常见的异常成因：数据来源于不同的类，自然变异，以及数据测量或收集误差。

数据来源于不同的类　某个数据对象可能不同于其他数据对象（即异常），因为它属于一个不同的类型或类。例如，进行信用卡欺诈的人属于不同的信用卡用户类，不同于合法使用信用卡的那些人。本章开始提供的大部分例子，即欺诈、入侵、疾病暴发、不寻常的实验结果，都是代表不同类对象的异常的例子。这类异常通常都是相当有趣的，并且是数据挖掘领域异常检测的关注点。

异常对象来自于一个与大多数数据对象源（类）不同的源（类）的思想，是统计学家 Douglas Hawkins 在经常被引用的一个离群点的定义中提出的。

定义 10.1　Hawkins 的离群点定义　离群点是一个观测值，它与其他观测值的差别如此之大，以至于怀疑它是由不同的机制产生的。

自然变异　许多数据集可以用一个统计分布建模，如用正态（高斯）分布建模，其中数据对象的概率随对象到分布中心距离的增加而急剧减小。换言之，大部分数据对象靠近中心（平均对象），数据对象显著地不同于这个平均对象的似然性很小。例如，一个特别高的人，在来自一个单独对象类的意义下不是异常的，而仅在所有对象都具备的一个特性（身高）有一个极端值的意义下才是异常的。通常，代表极端的或未必可能变异的异常是有趣的。

数据测量和收集误差　数据收集和测量过程中的误差是另一个异常源。例如，由于人的错误、测量设备的问题或存在噪声，测量值可能被不正确地记录。我们目标是删除这样的异常，因为它们不提供有意义的信息，而只会降低数据和其后数据分析的质量。事实上，删除这类异常是数据预处理（尤其是数据清理）的关注点。

小结　异常可以是上述原因或我们未考虑的其他原因的结果。事实上，数据集中可能有多种异常源，并且任何特定的异常的底层原因常常是未知的。在实践中，异常检测技术着力于发现显著不同于其他对象的对象，而技术本身不受异常源的影响。这样一来，异常的底层原因仅对预期的应用是重要的。

10.1.2　异常检测方法

这里，我们提供一些异常检测技术和与之相关联的异常定义的高层描述。这些技术之间有些重叠，它们之间的关系在本章习题 1 中进一步考察。

基于模型的技术 许多异常检测技术首先建立一个数据模型。异常是那些同模型不能完美拟合的对象。例如，数据分布模型可以通过估计概率分布的参数来创建。如果一个对象不能很好地同该模型拟合，即如果它很可能不服从该分布，则它是一个异常。如果模型是簇的集合，则异常是不显著属于任何簇的对象。在使用回归模型时，异常是相对远离预测值的对象。

由于异常和正常对象可以看作定义两个不同的类，因此可以使用分类技术来建立这两个类的模型。当然，仅当某些对象存在类标号，使得我们可以构造训练数据集时才可以使用分类技术。此外，异常相对稀少，在选择分类技术和评估度量时需要考虑这一因素。

在某些情况下，很难建立模型，例如，因为数据的统计分布未知或没有训练数据可用。在这些情况下，可以使用如下所述的不需要模型的技术。

基于邻近度的技术 通常可以在对象之间定义邻近性度量，并且许多异常检测方法都基于邻近度。异常对象是那些远离大部分其他对象的对象。这一领域的许多技术都基于距离，称作**基于距离的离群点检测技术**。当数据能够以二维或三维散布图显示时，通过寻找与大部分其他点分离的点，可以从视觉上检测出基于距离的离群点。

基于密度的技术 对象的密度估计可以相对直接地计算，特别是当对象之间存在邻近性度量时。低密度区域中的对象相对远离近邻，可能被看作异常。一种更复杂的方法考虑到数据集可能有不同密度区域这一事实，仅当一个点的局部密度显著地低于它的大部分近邻时才将其分类为离群点。

10.1.3 类标号的使用

异常检测有三种基本方法：非监督的、监督的和半监督的。它们的主要区别至少对于某些数据而言是类标号（异常或正常）可以利用的程度。

监督的异常检测 监督的异常检测技术要求存在异常类和正常类的训练集（注意，可能存在多个正常类或异常类）。正如前面所提到的，处理所谓稀有类问题的分类技术至关重要，因为相对于正常类而言，异常相对稀少。见 5.7 节。

非监督的异常检测 在许多实际情况下，没有提供类标号。在这种情况下，目标是将一个得分（或标号）赋予每个实例，反映该实例是异常的程度。注意许多互相相似的异常的出现可能导致它们都被标记为正常，或具有较低的离群点得分。这样，对于成功的非监督的异常检测，异常必须相互不同，与正常对象也不同。

半监督的异常检测 有时，训练数据包含被标记的正常数据，但是没有关于异常对象的信息。在半监督的情况下，目标是使用有标记的正常对象的信息，对于给定的对象集合，发现异常标号或得分。注意，在这种情况下，被评分对象集中许多相关的离群点的出现并不影响离群点的评估。然而，在许多实际情况下，可能很难发现代表正常对象的小集合。

本章介绍的所有异常检测方案都可以用于监督或非监督方式。监督方案本质上与 5.7 节讨论的稀有类分类方案相同。

10.1.4 问题

在处理异常时，存在各种需要处理的重要问题。

用于定义异常的属性个数 一个对象是不是基于单个属性的异常问题也就是对象的那个属

性值是否异常的问题。然而，由于对象可以有许多属性，它可能在某些属性上具有异常值，而在其他属性上具有正常值。此外，即使一个对象的所有属性值都不是异常的，对象也可能是异常的。例如，身高2英尺（儿童）或体重300磅的人很常见，但是体重300磅的人身高2英尺是罕见的。异常的一般定义必须指明如何使用多个属性的值确定一个对象是否异常。当数据的维度很高时，这个问题特别重要。

全局观点与局部观点　一个对象可能相对于所有对象看上去不寻常，但是相对于它的局部近邻并非如此。例如，身高6英尺5英寸的人对于一般人群是不常见的，但是对于职业篮球运动员不算什么。

点的异常程度　某些技术以二元方式报告对象是否异常的评估：对象要么是异常，要么不是。通常，这不能反映某些对象比其他对象更加极端异常的基本事实。因此，需要有某种对象异常程度的评估，这种评估称作**异常或离群点得分**（outlier score）。

一次识别一个异常与多个异常　在某些技术中，一次删除一个异常，即识别并删除最异常的实例，然后重复这一过程。对于其他技术，异常集族一起识别。试图一次识别一个异常的技术常常遇到所谓**屏蔽**（masking）问题，其中若干异常的出现屏蔽其他异常。另一方面，一次检测多个异常的技术可能陷入**泥潭**（swamping），其中正常的对象被识别为离群点。在基于模型的方法中，这些情况可能发生，因为异常扰乱模型。

评估　如果可以使用类标号来识别异常和正常数据，则可以使用5.7节讨论的分类性能度量来评估异常检测方案的有效性。但是由于异常类通常比正常类小得多，因此诸如精度、召回率和假正率等度量比正确率更合适。如果不能使用类标号，则评估是困难的。然而，对于基于模型的方法，离群点检测的有效性可以通过删除异常后对模型的改进来评估。

有效性　各种异常检测方案的计算开销显著不同。基于分类的方案可能需要相当多的资源来创建分类模型，但是使用开销通常很小。同理，基于统计的方法创建一个统计模型，而后可以以常数时间对一个对象分类。基于邻近度的方法通常具有$O(m^2)$时间复杂度，其中m是对象的个数，因为它们需要的信息通常只能通过计算邻近度矩阵得到。这一时间复杂度在具体情况下（如低维数据）可以通过使用专门的数据结构和算法来降低。其他方法的时间复杂度考虑见本章习题6。

路线图

下面四节介绍异常检测方法的几种主要类型：统计学的、基于邻近度的、基于密度的和基于簇的。在每种类型中，都考虑一种或多种具体技术。在以下几节中，我们将遵循惯例，使用术语离群点，而不是异常。

10.2　统计方法

统计学方法是基于模型的方法，即为数据创建一个模型，并且根据对象拟合模型的情况来评估它们。大部分用于离群点检测的统计学方法都基于构建一个概率分布模型，并考虑对象有多大可能符合该模型。这一思想表达在定义10.2中。

定义10.2　离群点的概率定义　离群点是一个对象，关于数据的概率分布模型，它具有低概率。

概率分布模型通过估计用户指定的分布的参数，由数据创建。如果假定数据具有高斯分布，则基本分布的均值和标准差可以通过计算数据的均值和标准差来估计。然后可以估计每个对象在

该分布下的概率。

基于定义 10.2，研究人员设计了各种统计检验来检测离群点，或者使用统计学界常用的称呼，**不和谐的观测值**（discordant observation）。大部分不和谐检验都是高度专业化的，并且所需要的统计学知识水平已超出了本书范围。因此，我们只用一些例子来解释基本概念，而建议读者从文献注释中寻求进一步指导。

问题

这种离群点检测方法面临的重要问题如下。

识别数据集的具体分布 尽管许多类型的数据都可以用少量常见的分布（如高斯、泊松或二项式分布）来描述，但是具有非标准分布的数据集也很常见。当然，如果选择了错误的模型，则对象可能被错误地识别为离群点。例如，数据也许被建模成来自高斯分布，但是它实际可能来自另一种分布，它以（比高斯分布）更高的概率具有远离均值的值。具有这类行为的统计分布在实践中是常见的，并称作**重尾分布**（heavy-tailed distribution）。

使用的属性个数 尽管大部分基于统计学的离群点检测技术都使用单个属性，但是已经开发了一些技术用于多元数据。

混合分布 可以用混合分布对数据建模，并且基于这种模型开发离群点检测方案。尽管功能可能更强，但是这种模型更复杂，难以理解和使用。例如，需要在将对象分类为离群点之前识别分布。见 9.2.2 节关于混合模型和 EM 算法的讨论。

10.2.1 检测一元正态分布中的离群点

高斯（正态）分布是统计学最常使用的分布之一，我们将使用它介绍一种简单的统计学离群点检测方法。该分布用记号 $N(\mu,\sigma)$表示，它的两个参数μ和σ分别为均值和标准差。图 10-1 显示 $N(0,1)$的密度函数。

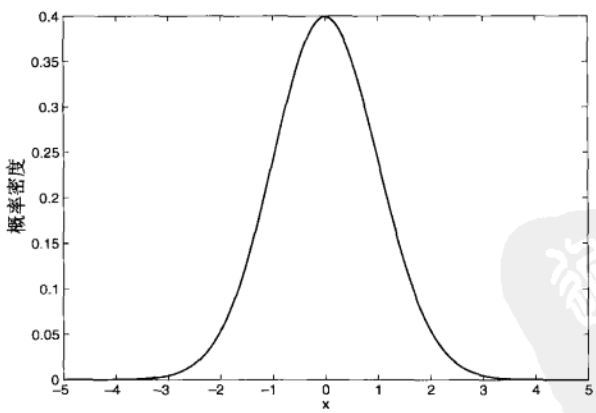


图 10-1 均值为 0，标准差为 1 的高斯分布的概率密度函数

来自 $N(0,1)$分布的对象（值）出现在该分布尾部的机会很小。例如，对象落在± 3 标准差的中心区域之外的概率仅有 0.0027。更一般地，如果 c 是常数，x 是属性值，则$|x| \geqslant c$ 的概率随 c 增加而迅速减小。设$\alpha = prob(|x| \geqslant c)$。表 10-1 显示当分布为 $N(0,1)$时 c 的某些样本值和对应的α值。

注意，离均值超过 4 个标准差的值出现的可能性是万分之一。

表 10-1　均值为 0，标准差为 1 的高斯分布的样本对(c, α)，$\alpha = prob(|x| \geqslant c)$

c	$N(0,1)$的α
1.00	0.3173
1.50	0.1336
2.00	0.0455
2.50	0.0124
3.00	0.0027
3.50	0.0005
4.00	0.0001

因为值到 $N(0,1)$分布中心的距离 c 直接与该值的概率相关，因此可以使用它作为检测对象（值）是否是定义 10.3 指出的离群点的基础。

定义 10.3　单个 $N(0,1)$高斯属性的离群点　设属性 x 取自具有均值 0 和标准差 1 的高斯分布。一个具有属性值 x 的对象是离群点，如果

$$|x| \geqslant c \tag{10-1}$$

其中，c 是一个选定的常量，满足 $prob(|x| \geqslant c) = \alpha$。

为了使用该定义，需要指定α值。从不寻常的值（对象）预示来自不同分布的值的观点来说，α表示我们错误地将来自给定分布的值分类为离群点的概率。从离群点是 $N(0,1)$分布的稀有值的观点来说，α表示稀有程度。

如果（正常对象的）一个感兴趣的属性的分布是具有均值μ和标准差σ的高斯分布，即 $N(\mu,\sigma)$分布，则为了使用定义 10.3，我们需要将属性 x 变换为新属性 z，z 具有 $N(0,1)$分布。具体地说，方法是令 $z = (x - \mu)/\sigma$（通常，z 称 z 得分）。然而，μ和σ通常是未知的，并使用样本均值$\bar{x}$和样本标准差 s_x 估计。实践中，当观测值很多时，这种估计的效果很好。然而，我们应当注意 z 的分布事实上并非 $N(0,1)$。一种更复杂的统计过程（Grubbs 检验）在本章习题 7 中考察。

10.2.2　多元正态分布的离群点

对于多元高斯观测，我们希望使用类似用于单变量高斯分布的方法。比如，如果点关于估计的数据分布具有低概率，则我们将把它们分类为离群点。此外，我们希望能够用简单的检验，例如点到分布中心的距离来进行判定。

然而，由于不同变量（属性）之间的相关性，多元正态分布并不关于它的中心对称。图 10-2 显示一个二维多元高斯分布的概率密度，该分布均值为(0,0)，协方差矩阵为

$$\Sigma = \begin{pmatrix} 1.00 & 0.75 \\ 0.75 & 3.00 \end{pmatrix}$$

如果我们打算使用一个简单的阈值来决定一个对象是否是离群点，则我们需要一种考虑数据分布形状的距离度量。Mahalanobis 距离就是这样一种距离，见公式（2-14）。点 $\mathbf{x}$ 与数据均值 $\bar{\mathbf{x}}$ 之间的 Mahalanobis 距离显示在公式（10-2）中。其中 $\mathbf{S}$ 是数据的协方差矩阵。

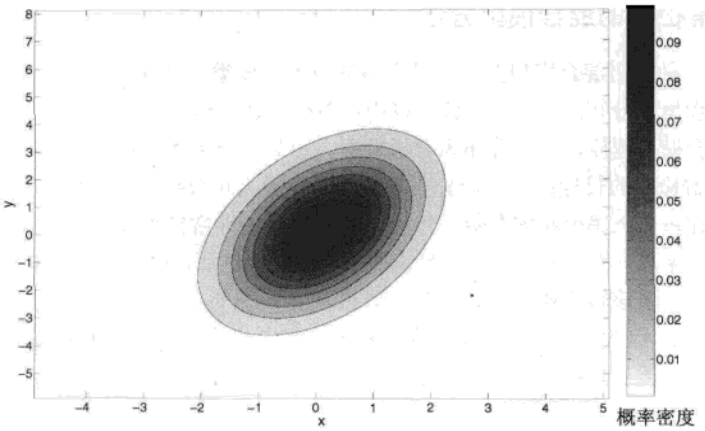


图 10-2 用于产生图 10-3 的高斯分布的概率密度

$$mahalanobis(\mathbf{x}, \overline{\mathbf{x}}) = (\mathbf{x} - \overline{\mathbf{x}})\mathbf{S}^{-1}(\mathbf{x} - \overline{\mathbf{x}})^T \tag{10-2}$$

容易证明，点到基础分布均值的 Mahalanobis 距离与点的概率直接相关。例如，Mahalanobis 距离等于点的概率密度的对数加上一个常数，见本章习题 9。

例 10.1 多元正态分布的离群点 图 10-3 显示二维数据集中点(到分布均值)的Mahalanobis 距离。点 A(–4, 4)和 B(5, 5)是添加到数据集中的离群点，它们的 Mahalanobis 距离显示在图中。数据集中的其他 2 000 个点使用图 10-2 的分布随机地产生。

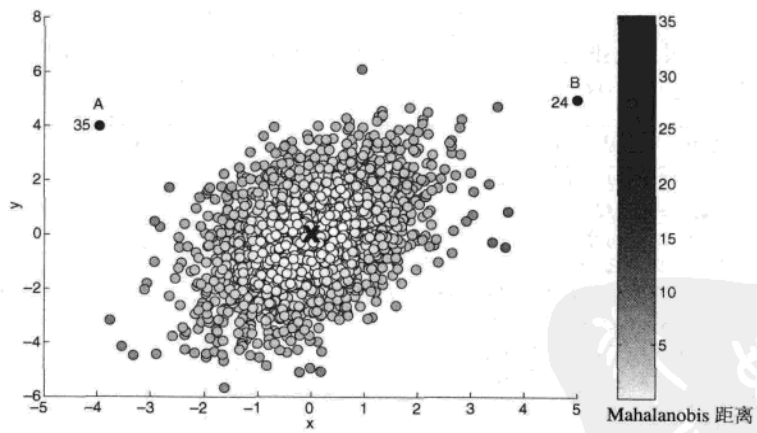


图 10-3 2 002 个点的二维数据集中点到中心的 Mahalanobis 距离

A 和 B 都具有很大的 Mahalanobis 距离。然而，尽管使用欧几里得距离度量 A 比 B 更靠近中心（(0,0)处的黑色“x”），但是按照 Mahalanobis 距离，它比 B 更远离中心，因为 Mahalanobis 距离考虑了分布的形状。具体地说，点 B 的欧几里得距离为 $5\sqrt{2}$，而 Mahalanobis 距离为 24，而点 A 的欧几里得距离为 $4\sqrt{2}$，而 Mahalanobis 距离为 35。 □

10.2.3　异常检测的混合模型方法

本节提供一种使用混合模型方法的异常检测技术。聚类（见9.2.2节）时，混合模型方法假定数据来自混合概率分布，并且每个簇可以用这些分布之一识别。同样，对于异常检测，数据用两个分布的混合模型建模，一个分布为普通数据，而另一个为离群点。

聚类和异常检测的目标都是估计分布的参数，以最大化数据的总似然（概率）。聚类时，使用EM算法估计每个概率分布的参数。然而，这里提供的异常检测技术使用一种更简单的方法。初始时将所有对象放入普通对象集，而异常对象集为空。然后，用一个迭代过程将对象从普通集转移到异常集，只要该转移能提高数据的总似然。

假定数据集 D 包含来自两个概率分布的对象：M 是大多数（正常）对象的分布，而 A 是异常对象的分布。数据的总概率分布可以记作

$$D(\mathbf{x}) = (1-\lambda)M(\mathbf{x}) + \lambda A(\mathbf{x}) \tag{10-3}$$

其中，$\mathbf{x}$ 是一个对象；λ 是0和1之间的数，给出离群点的期望比例。分布 M 由数据估计，而分布 A 通常取均匀分布。设 M_t 和 A_t 分别为时刻 t 正常和异常对象的集合。初始 $t=0$，$M_0 = D$，而 A_0 为空。在任意时刻 t，整个数据集的似然和对数似然分别以下两式给出：

$$L_t(D) = \prod_{\mathbf{x}_i \in D} P_D(\mathbf{x}_i) = \left((1-\lambda)^{|M_t|} \prod_{\mathbf{x}_i \in M_t} P_{M_t}(\mathbf{x}_i)\right)\left(\lambda^{|A_t|} \prod_{\mathbf{x}_i \in A_t} P_{A_t}(\mathbf{x}_i)\right) \tag{10-4}$$

$$LL_t(D) = |M_t|\log(1-\lambda) + \sum_{\mathbf{x}_i \in M_t} \log P_{M_t}(\mathbf{x}_i) + |A_t|\log\lambda + \sum_{\mathbf{x}_i \in A_t} \log P_{A_t}(\mathbf{x}_i) \tag{10-5}$$

其中 P_D、P_{M_t} 和 P_{A_t} 分别是 D、M_t 和 A_t 的概率分布函数。该式可以由公式（9-6）（9.2.2节）给出的混合模型的一般定义推出。为此，有必要做一些简化假定——对于以下两种情况概率为0：(1)A 中的对象是正常的，(2)M 中的对象是离群点。算法10.1给出了细节。

算法 10.1　基于似然的离群点检测

1：初始化：在时刻 $t=0$，令 M_t 包含所有对象，而 A_t 为空。
　令 $LL_t(D) = LL(M_t) + LL(A_t)$ 为所有数据的对数似然。
2：**for** 属于 M_t 的每个点 $\mathbf{x}$ **do**
3：　将 $\mathbf{x}$ 从 M_t 移动到 A_t，产生新的数据集合 A_{t+1} 和 M_{t+1}。
4：　计算 D 的新的对数似然 $LL_{t+1}(D) = LL(M_{t+1}) + LL(A_{t+1})$
5：　计算差 $\Delta = LL_t(D) - LL_{t+1}(D)$
6：　**if** $\Delta > c$，其中 c 是某个阈值 **then**
7：　　将 $\mathbf{x}$ 分类为异常。即 M_{t+1} 和 A_{t+1} 保持不变，并成为当前的正常和异常集。
8：　**end if**
9：**end for**

因为正常对象的数量比异常对象的数量大得多，因此，当一个对象移动到异常集后，正常对象的分布变化不大。在这种情况下，每个正常对象对正常对象的总似然的贡献保持相对不变。此外，如果假定异常服从均匀分布，则移动到异常集的每个对象对异常的似然贡献一个固定的量。这样，当一个对象移动到异常集时，数据总似然的改变粗略地等于该对象在均匀分布下的概率（用 λ 加权）减去该对象在正常数据点的分布下的概率（用 $1-\lambda$ 加权）。从而，异常集由这样一些对象组成，这些对象在均匀分布下的概率明显比在正常对象分布下的概率高。

在刚才讨论的情况下，算法 10.1 粗略地等价于把在正常对象的分布下具有低概率的对象分类为离群点。例如，当用于图 10-3 中的点时，该技术将把 A 和 B（以及其他远离均值的点）分类为离群点。然而，如果随着异常点的移出正常对象的分布显著改变，或者可以用更复杂的方法对异常的分布建模，则该方法产生的结果将不同于简单地将低概率对象分类为离群点的结果。此外，即使对象的分布是多峰的，该方法仍然能够处理。

10.2.4 优点与缺点

离群点检测的统计学方法具有坚实的基础，建立在标准的统计学技术（如分布参数的估计）之上。当存在充分的数据和所用的检验类型的知识时，这些检验可能非常有效。对于单个属性，存在各种统计离群点检测。对于多元数据，可用的选择少一些，并且对于高维数据，这些检验可能性能很差。

10.3 基于邻近度的离群点检测

尽管基于邻近度的异常检测的思想存在若干变形，但是其基本概念是很简单的。一个对象是异常的，如果它远离大部分点。这种方法比统计学方法更一般、更容易使用，因为确定数据集的有意义的邻近性度量比确定它的统计分布更容易。

度量一个对象是否远离大部分点的一种最简单的方法是使用到 k-最近邻的距离。定义 10.4 就是基于这种思想。离群点得分的最低值是 0，而最高值是距离函数的可能最大值——一般为无穷大。

定义 10.4 到 k 最近邻的距离 一个对象的离群点得分由到它的 k-最近邻的距离给定。

图 10-4 显示一个二维点集。使用 $k=5$，每个点的阴影指明它的离群点得分。注意，边远的点 C 被正确地赋予最高离群点得分。

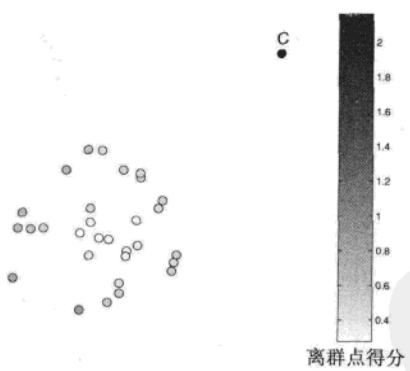


图 10-4 基于到第 5 个最近邻距离的离群点得分

离群点得分对 k 的取值高度敏感。如果 k 太小（例如 1），则少量的邻近离群点可能导致较低的离群点得分。例如，图 10-5 显示一个二维数据点集，其中另一个点靠近 C。阴影反映使用 $k=1$ 的离群点得分。注意，C 和它的近邻都具有低离群点得分。如果 k 太大，则点数少于 k 的簇中所有的对象可能都成了离群点。例如，图 10-6 显示一个二维数据集，除了一个 30 个点的较大的簇之外，该数据集还有一个 5 个点的自然簇。对于 $k=5$，较小簇中所有点的离群点得分都很高。为了使该方案对于 k 的选取更具有鲁棒性，可以修改定义 10.4，使用前 k 个最近邻的平均距离。

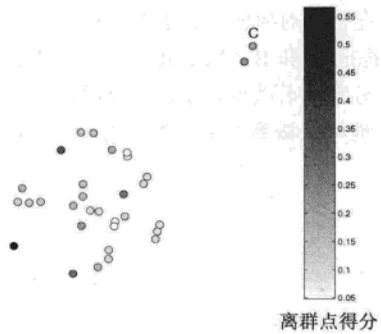


图 10-5　基于到第一个最近邻距离的离群点得分，邻近的离群点具有低离群点得分

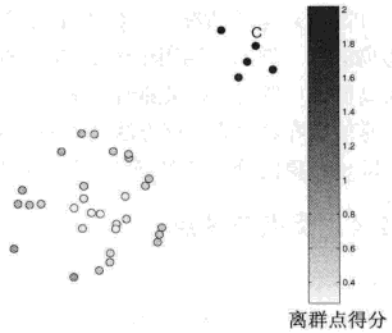


图 10-6　基于到第 5 个最近邻距离的离群点得分，一个小簇成了离群点

优点与缺点

与其他相关方案一样，上面介绍的基于距离的离群点检测方案是简单的。然而，基于邻近度的方法一般需要 $O(m^2)$时间。这对于大型数据集可能代价过高，尽管在低维情况下可以使用专门的算法来提高性能。该方法对参数的选择也是敏感的。此外，它不能处理具有不同密度区域的数据集，因为它使用全局阈值，不能考虑这种密度的变化。

为了解释这一点，考虑图 10-7 中的二维数据点的集合。该图有一个相当松散的点簇，一个稠密点簇，和两个点 C 和 D，它们离这两个簇相当远。根据定义 10.4，对于 $k=5$ 对点赋予离群点得分正确地识别出 C 为离群点，但是 D 表现出低的离群点得分。事实上，D 的离群点得分比松散簇中的许多点都低得多。

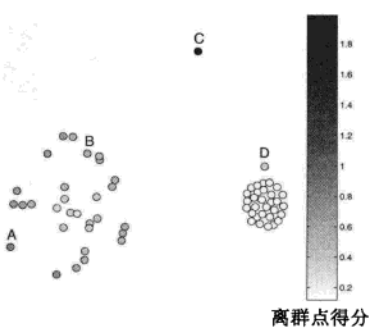


图 10-7　基于到第 5 个最近邻距离的离群点得分，不同密度的簇

10.4　基于密度的离群点检测

从基于密度的观点来说，离群点是在低密度区域中的对象。

定义 10.5　基于密度的离群点　一个对象的离群点得分是该对象周围密度的逆。

基于密度的离群点检测与基于邻近度的离群点检测密切相关，因为密度通常用邻近度定义。一种常用的定义密度的方法是，定义密度为到 k 个最近邻的平均距离的倒数。如果该距离小，则密度高，反之亦然。定义 10.6 体现了这种思想。

定义 10.6 逆距离

$$density(\mathbf{x},k)=\left(\frac{\sum_{\mathbf{y}\in N(\mathbf{x},k)}distance(\mathbf{x},\mathbf{y})}{|N(\mathbf{x},k)|}\right)^{-1} \tag{10-6}$$

其中，$N(\mathbf{x},k)$是包含 $\mathbf{x}$ 的 k-最近邻的集合，$|N(\mathbf{x},k)|$是该集合的大小，而 $\mathbf{y}$ 是一个最近邻。

另一种密度定义是使用 DBSCAN 聚类算法使用的密度定义，见 8.4 节。

定义 10.7 给定半径内的点计数 一个对象周围的密度等于该对象指定距离 d 内对象的个数。

需要小心地选择参数 d。如果 d 太小，则许多正常点可能具有低密度，从而具有高离群点得分。如果 d 太大，则许多离群点可能具有与正常点类似的密度（和离群点得分）。

使用任何密度定义检测离群点具有与 10.3 节讨论的基于邻近度的离群点方案类似的特点和局限性。例如，当数据包含不同密度的区域时，它们不能正确地识别离群点（见图 10-7）。为了正确地识别这种数据集中的离群点，我们需要与对象邻域相关的密度概念。例如，根据定义 10.6 和定义 10.7，图 10-7 中的点 D 比点 A 具有更高的绝对密度，但是相对于它的最近邻，它的密度较低。

有许多方法定义对象的相对密度。一种方法是 9.4.8 节讨论的基于 SNN 密度的聚类算法使用的方法。另一种方法是用点 $\mathbf{x}$ 的密度与它的最近邻 $\mathbf{y}$ 的平均密度之比作为相对密度，如下式：

$$average\ relative\ density(\mathbf{x},k)=\frac{density(\mathbf{x},k)}{\sum_{\mathbf{y}\in N(\mathbf{x},k)}density(\mathbf{y},k)/|N(\mathbf{x},k)|} \tag{10-7}$$

10.4.1 使用相对密度的离群点检测

本节，我们介绍一种基于相对密度概念的技术。该技术是局部离群点要素（Local Outlier Factor，LOF）技术（见文献注释）的简化版本，在算法 10.2 中给出。下面更详细地考察算法的细节。这里先概括地介绍操作步骤。首先，对于指定的近邻个数（k），基于对象的最近邻计算对象的密度 $density(\mathbf{x},k)$，由此计算每个对象的离群点得分。然后，计算点的近邻平均密度，并使用它们计算公式（10-7）定义的点的平均相对密度。这个量指示 $\mathbf{x}$ 是否在比它的近邻更稠密或更稀疏的邻域内，并取作 $\mathbf{x}$ 的离群点得分。

算法 10.2 相对密度离群点得分算法

```
1：{k 是最近邻个数}
2：for all 对象 x do
3：    确定 x 的 k-最近邻 N(x, k)。
4：    使用 x 的最近邻（即 N(x, k)中的对象），确定 x 的密度 density(x,k)。
5：end for
6：for all 对象 x do
7：    由公式（10-7），置 outlier score(x, k) = average relative density(x,k)。
8：end for
```

例 10.2 相对密度离群点检测 我们使用图 10-7 显示的示例数据集，解释相对密度离群点

检测方法的性能。这里，k=10。这些点的离群点得分显示在图 10-8 中。每个点的明暗由它的得分决定；即，具有高得分的点较黑。我们用值标出了点 A、C 和 D，它们具有最大离群点得分。这些点分别是最极端的离群点、关于紧致点集的最极端的点和松散点集中最极端的点。 □

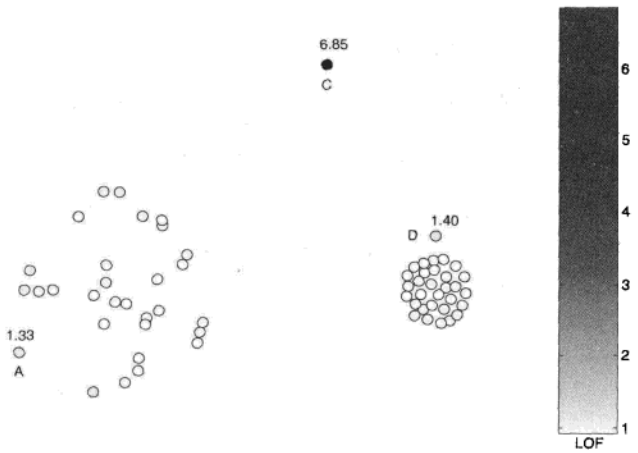


图 10-8　图 10-7 中二维点的相对密度（LOF）离群点得分

10.4.2　优点与缺点

基于相对密度的离群点检测给出了对象是离群点程度的定量度量，并且即使数据具有不同密度的区域也能够很好地处理。与基于距离的方法一样，这些方法必然具有 $O(m^2)$时间复杂度（其中 m 是对象个数），虽然对于低维数据，使用专门的数据结构可以将它降低到 $O(m\log m)$。参数选择也是困难的，虽然标准 LOF 算法通过观察不同的 k 值，然后取最大离群点得分来处理该问题。然而，仍然需要选择这些值的上下界。

10.5　基于聚类的技术

聚类分析发现强相关的对象组，而异常检测发现不与其他对象强相关的对象。因此毫无疑问，聚类可以用于异常检测。本节，我们将讨论一些这样的技术。

一种利用聚类检测离群点的方法是丢弃远离其他簇的小簇。这种方法可以与任何聚类技术一起使用，但是需要最小簇大小和小簇与其他簇之间距离的阈值。通常，该过程可以简化为丢弃小于某个最小尺寸的所有簇。这种方案对簇个数的选择高度敏感。此外，使用这一方案，很难将离群点得分附加在对象上。注意，把一组对象看作离群点，将离群点的概念从个体对象扩展到对象组，但是本质上没有任何改变。

一种更系统的方法是，首先聚类所有对象，然后评估对象属于簇的程度。对于基于原型的聚类，可以用对象到它的簇中心的距离来度量对象属于簇的程度。更一般地，对于基于目标函数的聚类技术，可以使用该目标函数来评估对象属于任意簇的程度。特殊情况下，如果删除一个对象导致该目标的显著改进，则我们可以将该对象分类为离群点。例如，对于K均值，删除远离其相关簇中心的对象能够显著地改进该簇的误差的平方和（SSE）。总而言之，聚类创建数据的模型，而异常扭曲该模型。该思想反映在定义 10.8 中。

定义 10.8 基于聚类的离群点 一个对象是基于聚类的离群点，如果该对象不强属于任何簇。

在与具有目标函数的聚类方法一起使用时，该定义是基于模型的异常定义的特殊情况。尽管定义 10.8 对于基于原型的或具有目标函数的方案更自然，但是它也可以包含用于检测离群点的基于密度和基于连接度的聚类方法。具体地说，对于基于密度的聚类，一个对象不强属于任何簇，如果它的密度太低；而对于基于连接度的聚类，一个对象不强属于任何簇，如果它不是强连接的。

下面，我们讨论任何基于聚类的离群点检测都需要处理的问题。我们的讨论集中在基于原型的聚类技术，如 K 均值。

10.5.1 评估对象属于簇的程度

对于基于原型的聚类，评估对象属于簇的程度的方法有多种。一种方法是度量对象到簇原型的距离，并用它作为该对象的离群点得分。然而，如果簇具有不同的密度，则我们可以构造一种离群点得分，度量对象到簇原型的相对距离（关于到该簇其他对象的距离）。另一种方法是使用 Mahalanobis 距离，只要簇可以准确地用高斯分布建模。

对于具有目标函数的聚类技术，我们可以将离群点得分赋予对象。该得分反映删除该对象后目标函数的改进。然而，基于目标函数评估点是离群点的程度可能是计算密集的。正因为如此，上一段的基于距离的方法更可取。

例 10.3 基于聚类的例子 这个例子基于图 10-7 显示的点集。基于原型的聚类使用 K 均值算法，而点的离群点得分用两种方法计算：(1)点到它的最近质心的距离，(2)点到它的最近质心的相对距离，其中相对距离是点到质心的距离与簇中所有点到质心的距离的中位数之比。后一种方法用于调整紧致簇与松散簇密度上的较大差别。

结果离群点得分显示在图 10-9 和图 10-10 中。与前面一样，本例用距离或相对距离度量的离群点得分用明暗度表示。在每种情况下，我们都使用两个簇。基于距离的方法在处理簇的不同密度方面存在问题；例如，D 未被视为离群点。对于基于相对距离的方法，先前使用 LOF 被识别为离群点的点（A、C 和 D）也作为离群点出现。 □

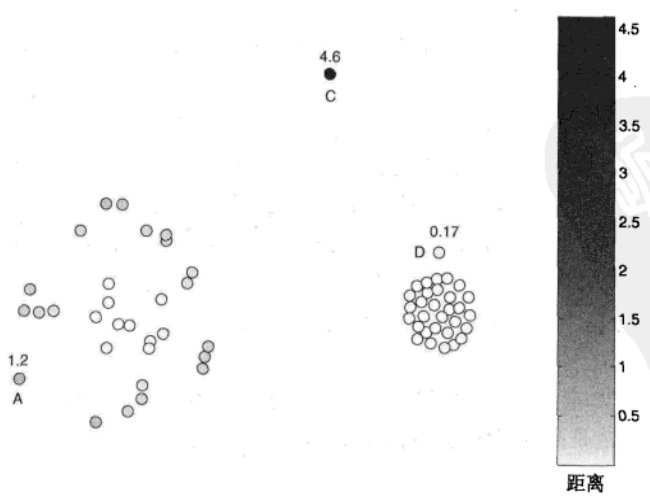


图 10-9 点到最近质心的距离

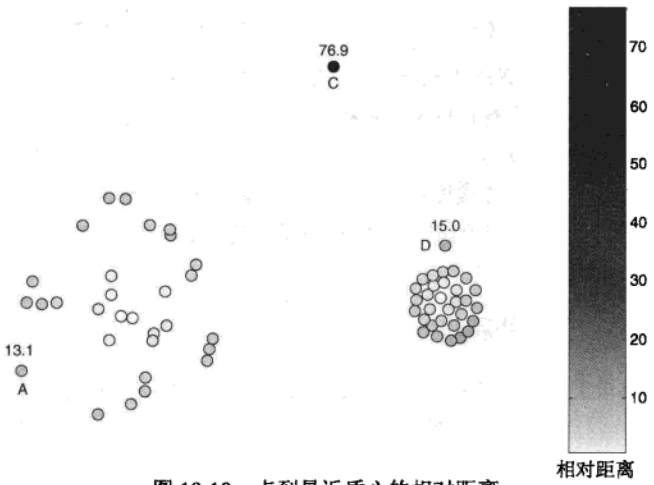


图 10-10　点到最近质心的相对距离

10.5.2　离群点对初始聚类的影响

如果通过聚类检测离群点，则由于离群点影响聚类，存在一个问题：结果是否有效。为了处理该问题，可以使用如下方法：对象聚类，删除离群点，对象再次聚类。尽管不能保证这种方法产生最优结果，但是该方法容易使用。一种更复杂的方法是取一组不能很好地拟合任何簇的特殊对象。这组对象代表潜在的离群点。随着聚类过程的进展，簇在变化。不再强属于任何簇的对象被添加到潜在的离群点集合；而当前在该集合中的对象被测试，如果它现在强属于一个簇，就可以将它从潜在的离群点集合移出。聚类过程结束时还留在该集合中的点被分类为离群点。但这样还是不能保证得到最优解，甚至不能保证该方法比前面的简单方法更好。例如，一个噪声点簇可能看上去像一个没有离群点的实际簇。如果使用相对距离计算离群点得分，这个问题特别严重。

10.5.3　使用簇的个数

诸如 K 均值等聚类技术并不能自动地确定簇的个数。在使用聚类进行离群点检测时这是一个问题，因为对象是否被认为是离群点可能依赖于簇的个数。例如，10个对象相互可能相对靠近，但是如果只找出几个大簇，则可能将它们作为某个较大簇的一部分。在这种情况下，10个点都可能被视为离群点。但是，如果指定足够多的簇个数，它们可能可以形成一个簇。

与其他某些问题一样，对于该问题也没有简单的答案。一种策略是对不同的簇个数重复该分析。另一种方法是找出大量小簇，其想法是(1)较小的簇趋向于更加凝聚，(2)如果在存在大量小簇时一个对象是离群点，则它多半是一个真正的离群点。不利的一面是一组离群点可能形成小簇而逃避检测。

10.5.4　优点与缺点

有些聚类技术（如 K 均值）的时间和空间复杂度是线性或接近线性的，因而基于这种算法的离群点检测技术可能是高度有效的。此外，簇的定义通常是离群点的补，因此可能同时发现簇和离群点。缺点方面，产生的离群点集和它们的得分可能非常依赖所用的簇的个数和数据中离群点的存在性。例如，基于原型的算法产生的簇可能因数据中存在离群点而扭曲。聚类算法产生的

簇的质量对该算法产生的离群点的质量影响非常大。正如在第 8 章和第 9 章所讨论的，每种聚类算法只适合特定的数据类型；因此，应当小心地选择聚类算法。

文献注释

异常检测有很长的历史，特别是在统计学领域，常称为离群点检测。与该主题相关的书有 Barnett 和 Lewis[464]、Hawkins[483]、Rousseeuw 和 Leroy[513]。Beckman 和 Cook[466]的文章提供了统计学家如何看待离群点检测这一主题的一般评述，并且介绍了该主题的历史，追溯到 1777 年 Bernoulli 的评论。另见相关文章[467，484]。另一篇关于离群点检测的一般评述是 Barnett [463]的文章。在多元数据中找离群点的文章包括 Davies 和 Gather[474]、Gnanadesikan 和 Kettenring [480]、Rocke 和 Woodruff[511]、Rousseeuw 和 van Zomerenand[515]以及 Scott [516]。Rosner[512]提供了同时找多个离群点的讨论。

Hodge 和 Austin[486]广泛综述离群点检测方法的。Markou 和 Singh[506，507]给出了分别涵盖统计学和神经网络技术的新颖检测技术的两部分评述。检测离群点的 Grubbs 过程最早在[481]中介绍。10.2.3 节讨论的混合模型离群点方法取自 Eskin[476]。Knorr 等[496～498]介绍了基于距离的离群点概念，并指出该定义可以包含离群点的许多统计学定义。LOF 技术（Breunig 等[468，469]）来自 DBSCAN。Ramaswamy 等[510]提出了一种基于距离的离群点检测过程，基于对象的 k-最近邻距离赋予每个对象一个离群点得分。有效性通过使用 BIRCH（9.5.2 节）的第一步划分数据获得。Chaudhary 等[470]使用 k-d 树提高离群点检测的有效性，而 Bay 和 Schwabacher[465]使用随机化和剪枝提高性能。Aggarwal 和 Yu[462]使用投影处理高维数据的离群点检测，而 Shyu 等[518]使用一种基于主成分的方法。高维空间离群点删除的理论讨论可以在 Dunagan 和 Vempala[475]的文章中找到。Lee 和 Xiang[504]介绍了异常检测中信息度量的使用，而 Ye 和 Chen[520]给出了一种基于χ^2度量的方法。

许多不同类型的分类技术都可以用于异常检测。Hawkins 等[485]、Ghosh 和 Schwartzbard[479]和 Sykacek[519]的文章讨论了神经网络领域的方法。稀有类检测的新近研究包括 Joshi 等[490～494]的研究。稀有类问题有时也称不平衡数据集问题。相关的文献有 AAAI 研讨会（Japkowicz[488]）、ICML 研讨会（Chawla 等[471]）和 SIGKDD 的专集（Chawla 等[472]）。

聚类和异常检测具有长期联系。在第 8 章和第 9 章，我们考虑了一些技术，如 BIRCH、CURE、DENCLUE、DBSCAN 和基于 SNN 密度的聚类，这些都包含专门处理异常的技术。Scott[516]、Hardin 和 Rocke[482]的文章介绍了讨论这种联系的统计学方法。

本章，我们关注基本异常检测方法。我们没有讨论考虑数据空间或时间特性的方法。Shekhar 等[517]详细讨论了空间离群点的问题，提出了一种统一的空间离群点检测方法。Fox[478]首次用严格的统计学方法讨论了时间序列中的离群点问题。Muirhead[508]讨论了时间序列中不同类型的离群点。Abraham 和 Chuang[461]提出了一种时间序列中检测离群点的贝叶斯方法，而 Chen 和 Liu[473]考虑了时间序列中不同类型的离群点，提出了一种检测它们并得到时间序列参数的较好估计的技术。Jagadish 等[487]和 Keogh 等[495]进行了发现时间序列数据库中的变异或意外模式的研究。Johnson 等[489]、Liu 等[505]和 Rousseeuw 等[514]的文章考察了基于几何学思想（如凸包的深度）的离群点检测。

异常检测的一个重要应用领域是入侵检测。Lee和Stolfo[502]、Lazarevic等[501]给出了数据挖掘在入侵检测方面应用的综述。在另一篇文章中，Lazarevic等[500]比较了专门用于网络入侵的

异常检测方法。Lee等[503]提供了使用数据挖掘技术进行入侵检测的框架。入侵检测领域的基于聚类的方法包括Eskin等[477]、Lane和Brodley[499]和Portnoy等[509]的工作。

参考文献

[461] B. Abraham and A. Chuang. Outlier Detection and Time Series Modeling. *Technometrics*, 31(2): 241 - 248, May 1989.

[462] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *Proc. of 2001 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 37 - 46. ACM Press, 2001.

[463] V. Barnett. The Study of Outliers: Purpose and Model. *Applied Statistics*, 27(3): 242 - 250, 1978.

[464] V. Barnett and T. Lewis. *Outliers in Statistical Data.* Wiley Series in Probability and Statistics. John Wiley & Sons, 3rd edition, April 1994.

[465] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. of the 9th Intl. Conf. on Knowledge Discovery and Data Mining,* pages 29 - 38. ACM Press, 2003.

[466] R. J. Beckman and R. D. Cook. 'Outlier.....s'. *Technometrics*, 25(2):119 - 149, May 1983.

[467] R. J. Beckman and R. D. Cook. ['Outlier.....s']: Response. *Technometrics*, 25(2): 161 - 163, May 1983.

[468] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. OPTICS-OF: Identifying Local Outliers. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pages 262 - 270. Springer-Verlag, 1999.

[469] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proc. of 2000 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 93 - 104. ACM Press, 2000.

[470] A. Chaudhary, A. S. Szalay, and A. W. Moore. Very fast outlier detection in large multidimensional data sets. In *Proc. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD),* 2002.

[471] N. V. Chawla, N. Japkowicz, and A. Kolcz, editors. *Workshop on Learning from Imbalanced Data Sets II, 20th Intl. Conf. on Machine Learning,* 2000. AAAI Press.

[472] N. V. Chawla, N. Japkowicz, and A. Kolcz, editors. *SIGKDD Explorations Newsletter, Special issue on learning from imbalanced datasets,* volume 6(1), June 2004. ACM Press.

[473] C. Chen and L.-M. Liu. Joint Estimation of Model Parameters and Outlier Effects in Time Series. *Journal of the American Statistical Association,* 88(421):284 - 297, March 1993.

[474] L. Davies and U. Gather. The Identification of Multiple Outliers. *Journal of the American Statistical Association,* 88(423):782 - 792, September 1993.

[475] J. Dunagan and S. Vempala. Optimal outlier removal in high-dimensional spaces. *Journal of Computer and System Sciences, Special Issue on STOC 2001,* 68(2):335 - 373, March 2004.

[476] E. Eskin. Anomaly Detection over Noisy Data using Learned Probability Distributions. In *Proc. of the 17th Intl. Conf. on Machine Learning,* pages 255 - 262, 2000.

[477] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. J. Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of Data Mining in Computer Security*, pages 78 - 100. Kluwer Academics, 2002.

[478] A. J. Fox. Outliers in Time Series. *Journal of the Royal Statistical Society. Series B (Methodological),* 34(3):350 - 363, 1972.

[479] A. Ghosh and A. Schwartzbard. A Study in Using Neural Networks for Anomaly and Misuse Detection.In *8th USENIX Security Symposium,* August 1999.

[480] R. Gnanadesikan and J. R. Kettenring. Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data. *Biometrics*, 28(1):81 - 124, March 1972.

[481] F. Grubbs. Procedures for Testing Outlying Observations. *Annal of Mathematical Statistics,* 21(1): 27 - 58, March 1950.

[482] J. Hardin and D. M. Rocke. Outlier Detection in the Multiple Cluster Setting using the Minimum Covariance Determinant Estimator. *Computational Statistics and Data Analysis*, 44:625 - 638, 2004.

[483] D. M. Hawkins. *Identification of Outliers*. Monographs on Applied Probability and Statistics. Chapman & Hall, May 1980.

[484] D. M. Hawkins. '[Outlier.....s]': Discussion. *Technometrics*, 25(2):155 - 156, May 1983.

[485] S. Hawkins, H. He, G. J. Williams, and R. A. Baxter. Outlier Detection Using Replicator Neural Networks. In *DaWaK 2000: Proc. of the 4th Intnl. Conf. on Data Warehousing and Knowledge Discovery*, pages 170 - 180. Springer-Verlag, 2002.

[486] V. J. Hodge and J. Austin. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22:85 - 126, 2004.

[487] H. V. Jagadish, N. Koudas, and S. Muthukrishnan. Mining Deviants in a Time Series Database. In *Proc. of the 25th VLDB Conf.*, pages 102 - 113, 1999.

[488] N. Japkowicz, editor. *Workshop on Learning from Imbalanced Data Sets I, Seventeenth National Conference on Artificial Intelligence, Published as Technical Report WS-00-05*, 2000. AAAI Press.

[489] T. Johnson, I. Kwok, and R. T. Ng. Fast Computation of 2-Dimensional Depth Contours. In *KDD98*, pages 224 - 228, 1998.

[490] M. V. Joshi. On Evaluating Performance of Classifiers for Rare Classes. In *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, pages 641 - 644, 2002.

[491] M. V. Joshi, R. C. Agarwal, and V. Kumar. Mining needle in a haystack: Classifying rare classes via two-phase rule induction. In *Proc. of 2001 ACM-SIGMOD Intl. Conf.on Management of Data*, pages 91 - 102. ACM Press, 2001.

[492] M. V. Joshi, R. C. Agarwal, and V. Kumar. Predicting rare classes: can boosting make any weak learner strong? In *Proc. of 2002 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 297 - 306. ACM Press, 2002.

[493] M. V. Joshi, R. C. Agarwal, and V. Kumar. Predicting Rare Classes: Comparing Two-Phase Rule Induction to Cost-Sensitive Boosting. In *Proc. of the 6th European Conf. of Principles and Practice of Knowledge Discovery in Databases*, pages 237 - 249. Springer-Verlag, 2002.

[494] M. V. Joshi, V. Kumar, and R. C. Agarwal. Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, pages 257 - 264, 2001.

[495] E. Keogh, S. Lonardi, and B. Chiu. Finding Surprising Patterns in a Time Series Database in Linear Time and Space. In *Proc. of the 8th Intl. Conf. on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 2002.

[496] E. M. Knorr and R. T. Ng. A Unified Notion of Outliers: Properties and Computation. In *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 219 - 222, 1997.

[497] E. M. Knorr and R. T. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *Proc. of the 24th VLDB Conf.*, pages 392 - 403, 24 - 27,1998.

[498] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3-4):237 - 253, 2000.

[499] T. Lane and C. E. Brodley. An Application of Machine Learning to Anomaly Detection. In *Proc. 20th NIST-NCSC National Information Systems Security Conf.*, pages 366 - 380, 1997.

[500] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava. A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In *Proc. of the 2003 SIAM Intl. Conf. on Data Mining*, 2003.

[501] A. Lazarevic, V. Kumar, and J. Srivastava. Intrusion Detection: A Survey. In *Managing Cyber Threats: Issues, Approaches and Challenges*, pages 19 - 80. Kluwer Academic Publisher, 2005.

[502] W. Lee and S. J. Stolfo. Data Mining Approaches for Intrusion Detection. In *7th USENIX Security Symposium*, pages 26 - 29, January 1998.

[503] W. Lee, S. J. Stolfo, and K. W. Mok. A Data Mining Framework for Building Intrusion Detection Models. In *IEEE Symposium on Security and Privacy*, pages 120 - 132, 1999.

[504] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *Proc. of the 2001*

IEEE Symposium on Security and Privacy, pages 130 - 143, May 2001.

[505] R. Y. Liu, J. M. Parelius, and K. Singh. Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Annals of Statistics*, 27(3):783 - 858, 1999.

[506] M. Markou and S. Singh. Novelty detection: A review - part 1: Statistical approaches. *Signal Processing*, 83(12):2481 - 2497, 2003.

[507] M. Markou and S. Singh. Novelty detection: A review - part 2: Neural network based approaches. *Signal Processing*, 83(12):2499 - 2521, 2003.

[508] C. R. Muirhead. Distinguishing Outlier Types in Time Series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(1):39 - 47, 1986.

[509] L. Portnoy, E. Eskin, and S. J. Stolfo. Intrusion detection with unlabeled data using clustering. In *In ACM Workshop on Data Mining Applied to Security*, 2001.

[510] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. of 2000 ACM-SIGMOD Intl. Conf. on Management of Data*, pages 427 - 438. ACM Press, 2000.

[511] D. M. Rocke and D. L. Woodruff. Identification of Outliers in Multivariate Data. *Journal of the American Statistical Association*, 91(435):1047 - 1061, September 1996.

[512] B. Rosner. On the Detection of Many Outliers. *Technometrics*, 17(3):221 - 227, 1975.

[513] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Statistics. John Wiley & Sons, September 2003.

[514] P. J. Rousseeuw, I. Ruts, and J. W. Tukey. The Bagplot: A Bivariate Boxplot. *The American Statistician*, 53(4):382 - 387, November 1999.

[515] P. J. Rousseeuw and B. C. van Zomeren. Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association*, 85(411):633 - 639, September 1990.

[516] D. W. Scott. Partial Mixture Estimation and Outlier Detection in Data and Regression. In M. Hubert, G. Pison, A. Struyf, and S. V. Aelst, editors, *Theory and Applications of Recent Robust Methods*, Statistics for Industry and Technology. Birkhauser, 2003.

[517] S. Shekhar, C.-T. Lu, and P. Zhang. A Unified Approach to Detecting Spatial Outliers. *GeoInformatica*, 7(2):139 - 166, June 2003.

[518] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A Novel Anomaly Detection Scheme Based on Principal Component Classifier. In *Proc. of the 2003 IEEE Intl. Conf. on Data Mining*, pages 353 - 365, 2003.

[519] P. Sykacek. Equivalent error bars for neural network classifiers trained by bayesian inference. In *Proc. of the European Symposium on Artificial Neural Networks*, pages 121 - 126, 1997.

[520] N. Ye and Q. Chen. Chi-square Statistical Profiling for Anomaly Detection. In *Proc. of the 2000 IEEE Workshop on Information Assurance and Security*, pages 187 - 193, June 2000.

习 题

1. 比较和对比 10.1.2 节介绍的不同的异常检测技术。具体地说，试确定用于不同技术的异常定义可能等价的情况；或一种定义有意义，而另一种无意义的情况。确保考虑不同的数据类型。
2. 考虑如下异常定义：异常是一个对象，它对数据模型的创建具有不寻常的影响。
   (a) 将该定义与标准的基于模型的异常定义进行比较。
   (b) 对于多大的数据集（小型、中型或大型），该定义是合适的？
3. 在一种异常检测方法中，对象用多维空间中的点表示，点被分组形成相继的壳（shell），其中每个壳代表点组周围的一个层，如凸包（convex hull）。一个对象是异常的，如果它落在一个外部的壳中。
   (a) 该定义与 10.1.2 节的哪个异常定义最相关？

(b) 指出该异常定义的两个问题。

4. 关联分析可以用来发现异常，方法如下。找出涉及对象最少的强关联模式。异常是不属于任何这种模式的对象。6.8 节讨论的超团关联模式特别适合这种方法。具体地说，给定用户选定的 h 置信水平，找出对象的最大超团模式。不在大小至少为 3 的最大超团模式中出现的所有对象都被分类为离群点。
   (a) 该技术属于本章讨论的某种类型吗？如果是，哪一种？
   (b) 指出该方法的一个潜在的优点和一个潜在的缺点。
5. 讨论结合多种异常检测技术，提高异常对象识别的技术。考虑监督和非监督两种情况。
6. 讨论基于如下方法的异常检测方法潜在的时间复杂度：使用聚类的基于模型的、基于邻近度和基于密度的。不需要专门技术的知识，而是关注每种方法的基本计算需求，如计算每个对象的密度的时间需求。
7. 算法 10.3 描述的 Grubbs 检验是比定义 10.3 更复杂的离群点检测的统计过程。它是迭代的，并且考虑到 z 得分不具有正态分布。该算法基于当前值集合的样本均值和标准差，计算每个值的 z 得分。丢弃具有最大 z 得分量级的值，如果它的 z 得分大于显著水平α下离群点检测的临界值 g_c。重复该过程，直到不能再删除任何对象。注意，每次迭代都更新样本均值、标准差和 g_c。

算法 10.3　离群点删除的 Grubbs 方法

1：输入值和α
{m 是值的个数，α是参数，t_c 是一个选定的值，使得对于具有 $m-2$ 个自由度的 t 分布，$\alpha = prob(x \geq t_c)$ }
2：**repeat**
3：　计算样本均值（$\bar{x}$）和标准差（s_x）。
4：　计算值 g_c，使得 $prob(|z| \geq g_c) = \alpha$。
（根据 t_c 和 m，$g_c = \frac{m-1}{\sqrt{m}} \sqrt{\frac{t_c^2}{m-2+t_c^2}}$。）
5：　计算每个值的 z 得分，即 $z = (x - \bar{x})/s_x$。
6：　令 $g = \max|z|$，即找出具有最大量级的 z 得分，并称它为 g。
7：　**if** $g > g_c$ **then**
8：　　删除对应于 g 的值。
9：　　$m \leftarrow m - 1$
10：　**end if**
11：**until** 没有对象被删除。

   (a) 当 m 趋向于无穷大时，用于 Grubbs 检验的值 $\frac{m-1}{\sqrt{m}} \sqrt{\frac{t_c^2}{m-2+t_c^2}}$ 的极限是什么？使用 0.05 的显著水平。
   (b) 描述前面结果的意义。
8. 许多用于离群点检测统计检验是在这样一种环境下开发的：数百个观测就是一个大数据集。我们考察这种方法的局限性。
   (a) 如果一个值与平均值的距离超过标准差的 3 倍，则检验称它为离群点。对于 1 000 000 个值的集合，根据该检验，有离群点的可能性多大？（假定正态分布）
   (b) 一种方法称离群点是具有不寻常低概率的对象。处理大型数据集时，该方法需要调整

吗？如果需要，如何调整？

9. 点 $\mathbf{x}$ 关于多元正态分布（均值为 μ，协方差矩阵为 Σ）的概率密度由下式给出

$$prob(\mathbf{x}) = \frac{1}{(\sqrt{2\pi})^m |\Sigma|^{1/2}} e^{-\frac{(\mathbf{x}-\mu)\Sigma^{-1}(\mathbf{x}-\mu)}{2}} \tag{10-8}$$

使用样本均值 $\bar{\mathbf{x}}$ 和协方差矩阵 S 分别作为均值 μ 和协方差矩阵 Σ 的估计，证明 $\log prob(\mathbf{x})$ 等于数据点 $\mathbf{x}$ 与样本均值 $\bar{\mathbf{x}}$ 之间的 Mahalanobis 距离，加上一个不依赖于 $\mathbf{x}$ 的常量。

10. 比较以下两种对象属于簇的程度的度量：(1)对象到它的最近簇的质心的距离，(2)8.5.2节介绍的轮廓系数。

11. 考虑 10.5 节介绍的离群点检测的（相对距离）K 均值方案和相应的图 10-10。
    (a) 图 10-10 紧致簇底部的点比该紧致簇顶部的点的离群点得分略高。为什么？
    (b) 假定我们选择的簇个数很大，例如 10。该技术仍然能够有效地找出该图顶部的最极端的离群点吗？为什么能或为什么不能？
    (c) 相对距离的使用旨在根据不同的密度调整。给出一个例子，说明这种方法可能导致错误的结论。

12. 假定正常对象被分类为异常的概率是 0.01，而异常对象被分类为异常的概率是 0.99。如果 99%的对象都是正常的，那么假警告率（false alarm rate）和检测率各是多少？（使用下面给出的定义。）

$$\text{检测率} = \frac{\text{检测出的异常的个数}}{\text{异常的总数}} \tag{10-9}$$

$$\text{假警告率} = \frac{\text{假异常的个数}}{\text{被分类为异常的对象个数}} \tag{10-10}$$

13. 当存在详尽的训练集时，如果使用诸如检测率和假警告率等度量评价性能的话，则监督的异常检测技术一般优于非监督的异常检测技术。然而，在某些情况下，如欺诈检测，总是出现新的异常类型。可以根据检测率和假警告率评价性能，因为通常可以根据观察决定对象（事务）是否是异常。讨论在此条件下，监督和非监督异常检测的相对优点。

14. 考虑一组文档，它们选自大量不同文档，使得被选中的文档尽可能相异。如果我们认为相互之间不高度相关（相连接、相似）的文档是异常，则我们选择的所有文档可能都被分类为异常。一个数据集仅由异常对象组成可能吗？或者，这是滥用术语吗？

15. 考虑一个点集，其中大部分点在低密度区域，少量点在高密度区域。如果我们定义异常为低密度区域中的点，则大部分点将被分类为异常。这是对基于密度的异常定义的适当使用吗？是否需要用某种方式修改该定义？

16. 考虑一个均匀地分布在[0，1]区间的点集。离群点是不被频繁观测到的值这一统计概念对于该数据集有意义吗？

17. 一个数据分析者使用一种异常检测算法发现了一个异常集。出于好奇，该分析者对这个异常集使用该异常检测算法。
    (a) 讨论本章介绍的每种异常检测技术的行为。（如果可能，使用实际数据和算法来做。）
    (b) 当用于异常对象的集合时，你认为异常检测算法将做何反应？

附录A

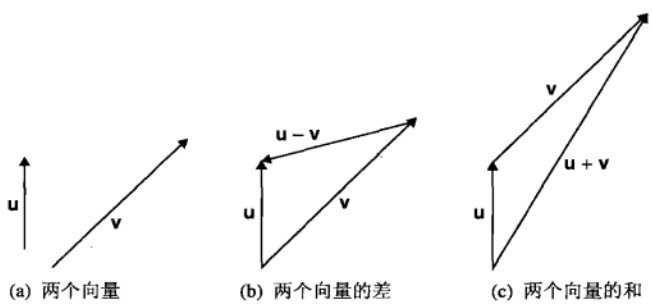
线 性 代 数

本附录简要介绍线性代数，主要介绍与本书相关的那些内容。先介绍可以用来表示数据对象和属性的向量，再讨论可以用来表示数据集和描述数据集上的变换的矩阵。

A.1 向量

A.1.1 定义

在欧几里得空间，如我们熟悉的二维空间和三维空间，**向量**（vector）是一个具有**量值**（magnitude）和**方向**（direction）的量。通常，向量用一个有向线段表示，其长度等于向量的量值，其指向由向量的方向给出。图 A-1a 给出了两个向量：向量 **u** 长度为 1、平行于 y 轴，向量 **v** 长度为 2、与 x 轴夹角为 45°。（我们将使用加粗的小写字母，如 **u** 和 **v**，表示向量。向量也常用斜体小写字母表示，如 u 和 v。）由于点可以看作由原点沿特定方向的位移，因此点可以用从原点到该点的向量表示。



(a) 两个向量　(b) 两个向量的差　(c) 两个向量的和

图 A-1　两个向量以及它们的和与差

A.1.2 向量加法和向量与标量乘法

向量可以进行多种运算。（下面我们假定向量都取自同一空间，即它们具有相同的维。）例如，向量可以加、减。向量的加、减最好用图示说明。图 A-1b 和图 A-1c 分别图示了向量的减法和加法。与数的加法一样，向量的加法也具有一些我们熟知的性质。如果 **u**、**v** 和 **w** 是 3 个向量，则向量的加法具有如下性质。

- **向量加法的交换律**。加的次序不影响结果：$\mathbf{u}+\mathbf{v}=\mathbf{v}+\mathbf{u}$。
- **向量加法的结合律**。相加时向量分组不影响结果：$(\mathbf{u}+\mathbf{v})+\mathbf{w}=\mathbf{u}+(\mathbf{v}+\mathbf{w})$。
- **向量加法单位元的存在性**。存在一个**零向量**（zero vector），简记为 **0**，是单位元。对于任

意向量 $\mathbf{u}$，有 $\mathbf{u}+\mathbf{0}=\mathbf{u}$。

- **向量加法逆元的存在性**。对于每个向量 $\mathbf{u}$，都存在一个逆向量 $-\mathbf{u}$，使得 $\mathbf{u}+(-\mathbf{u})=\mathbf{0}$。

另一个重要运算是用一个数乘以向量。用线性代数的术语，这个数通常称为**标量**（scalar）。标量乘改变向量的量值；如果标量为正则方向不变，如果标量为负则方向相反。如果 $\mathbf{u}$ 和 $\mathbf{v}$ 是向量，α和β是标量（数），则向量的标量乘法具有如下性质。

- **标量乘法的结合律**。被两个标量乘的次序不影响结果：$\alpha(\beta\mathbf{u})=(\alpha\beta)\mathbf{u}$。
- **标量加法对标量与向量乘法的分配律**。两个标量相加后乘以一个向量等于每个标量乘以该向量之后的结果向量相加：$(\alpha+\beta)\mathbf{u}=\alpha\mathbf{u}+\beta\mathbf{u}$。
- **标量乘法对向量加法的分配律**。两个向量相加之后的和与一个标量相乘等于每个向量与该标量相乘然后相加：$\alpha(\mathbf{u}+\mathbf{v})=\alpha\mathbf{u}+\alpha\mathbf{v}$。
- **标量单位元的存在性**。如果$\alpha=1$，则对于任何向量 $\mathbf{u}$，有$\alpha\mathbf{u}=\mathbf{u}$。

A.1.3　向量空间

向量空间（vector space）是向量的集合，连同一个相关联的标量集（如实数集），满足上述性质，并且关于向量加法和标量与向量乘法是封闭的。（封闭是指向量相加的结果、向量与标量相乘的结果都是原向量集中的向量。）向量空间具有如下性质：任何向量都可以用一组称作**基**（basis）的向量的**线性组合**（linear combination）表示。更明确地说，如果 $\mathbf{u}_1, \cdots, \mathbf{u}_n$ 是基向量，则对于任意向量 $\mathbf{v}$，都可以找到 n 个标量的集合$\{\alpha_1, \cdots, \alpha_n\}$使得 $\mathbf{v}=\sum_{i=1}^{n}\alpha_i\mathbf{u}_i$。我们称基向量**生成**（span）了该向量空间。向量空间的**维**（dimension）是形成基所需要的最少向量数。通常，我们选取具有单位长度的基向量。

基向量通常是**正交的**（orthogonal）。向量正交是直线垂直的二维概念的推广，稍后会准确定义。从概念上讲，正交向量是不相关的或独立的。如果基向量是相互正交的，则将向量表示成基向量的线性组合事实上把该向量分解成一些**独立分量**（independent component）。

因此，n 维空间的向量可以看作标量（数）的 n 元组。为了具体地解释，考虑二维欧几里得空间，那里每个点都与一个表示该点到原点的位移的向量相关联。到任意点的位移向量都可以用 x 方向和 y 方向的位移和表示，这些位移分别是该点的 x 和 y 坐标。

我们将使用记号 $\mathbf{v}=(v_1, v_2, \cdots, v_{n-1}, v_n)$引述向量 $\mathbf{v}$ 的分量。（关于等式 $\mathbf{v}=\sum_{i=1}^{n}\alpha_i\mathbf{u}_i$，$v_i=\alpha_i$。）注意，$v_i$是向量 $\mathbf{v}$ 的一个分量，而 $\mathbf{v}_i$是向量集中的一个向量。

从向量的分量角度看，向量的加法变得简单、易于理解。为了将两个向量相加，我们只需要简单地将对应的分量相加。例如，$(2, 3)+(4, 2)=(6, 5)$。为了计算标量乘以向量，我们只要用标量乘以每个分量，如 $3\times(2, 3)=(6, 9)$。

A.1.4　点积、正交性和正交投影

现在，我们定义何谓两个向量正交。为简单起见，我们只讨论欧几里得向量空间，这些定义和结果都很容易推广到一般情况。我们从定义两个向量的**点积**（dot product）开始。

定义 A.1　点积　两个向量 $\mathbf{u}$ 和 $\mathbf{v}$ 的点积 $\mathbf{u}\cdot\mathbf{v}$ 由下式给出：

$$\mathbf{u}\cdot\mathbf{v}=\sum_{i=1}^{n}u_iv_i \tag{A-1}$$

也就是说，两个向量的点积用向量对应分量的乘积的和来计算。例如，$(2, 3) \cdot (4, 1) = 2 \times 4 + 3 \times 1 = 11$。

在欧几里得空间，可以证明两个（非零）向量的点积为 0 当且仅当它们是垂直的。从几何角度，两个向量定义一个平面，并且它们的点积为 0 当且仅当这两个向量（在平面内）的夹角等于 90°。我们说这样的两个向量是**正交的**（orthogonal）。

点积也可以用来计算欧几里得空间中的向量长度：$\text{length}(\mathbf{u}) = \sqrt{\mathbf{u} \cdot \mathbf{u}}$。向量长度又称 L_2 **范数**（norm），并记作$||\mathbf{u}||$。给定一个向量 $\mathbf{u}$，我们可以通过用其长度除以 $\mathbf{u}$ 的每个分量，即通过计算 $\mathbf{u}/||\mathbf{u}||$，找到一个向量，它与 $\mathbf{u}$ 指向相同的方向，但是具有单位长度。这称作将该向量规范化，具有 L_2 范数 1。

给定向量范数，向量的点积也可以写成

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}|| \, ||\mathbf{v}|| \cos(\theta), \tag{A-2}$$

其中，θ 是两个向量之间的夹角。把项分组并重新排列，上式可以改写为

$$\mathbf{u} \cdot \mathbf{v} = (||\mathbf{v}|| \cos(\theta)) \, ||\mathbf{u}|| = \mathbf{v}_u \, ||\mathbf{u}||, \tag{A-3}$$

其中 $\mathbf{v}_u = ||\mathbf{v}|| \cos(\theta)$表示向量 $\mathbf{v}$ 在 $\mathbf{u}$ 的方向上的长度，如图 A-2 所示。如果 $\mathbf{u}$ 是单位向量，则该点积是 $\mathbf{v}$ 在 $\mathbf{u}$ 的方向上的分量。我们称它为 $\mathbf{v}$ 在 $\mathbf{u}$ 上的**正交投影**（orthogonal projection）。当然，如果 $\mathbf{v}$ 是单位向量，则该点积也是 $\mathbf{u}$ 在 $\mathbf{v}$ 方向上的投影。

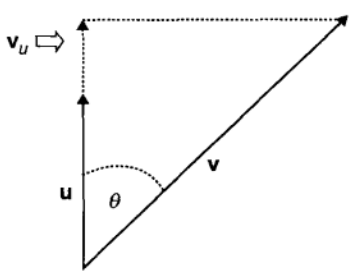


图 A-2 向量 $\mathbf{v}$ 在向量 $\mathbf{u}$ 方向的正交投影

这一结果的一个重要推论是，给定一个形成向量空间基的、范数为 1 的正交向量集，对于任意向量，我们可以通过求该向量与每个基向量的点积来计算该向量关于该基的分量。

一个与正交性密切相关的概念是**线性独立性**（linear independence）。

定义 A.2 线性独立性 如果一个向量集中的每个向量都不能表示成该集合中其他向量的线性组合，则该集合是**线性独立的**。

如果一个向量集不是线性独立的，则它们是**线性依赖的**（linearly dependent）。注意，我们希望基中每个向量都不线性依赖于其余的基向量，否则的话，我们可以删除线性依赖于其余基向量的向量，仍然有一个可以生成整个向量空间的基向量集。如果选择相互正交的（独立的）基向量，则我们自动得到一个线性独立的基向量集，因为任意两个向量都正交的向量集是线性独立的。

A.1.5 向量与数据分析

尽管最初引进向量是为了处理力、速度、加速度这样的量，但是实践证明它们也能用来表示和理解许多其他类型的数据。特别是，我们常常把一个数据对象或属性看作向量。例如，在第 2

章中，我们介绍了一个由150种鸢尾花组成的数据集，用萼片长度、萼片宽度、花瓣长度和花瓣宽度4个属性刻画。每种花可以看作一个4维向量，而每个属性可以看作一个150维的向量。另一个例子，文档可以用向量表示，其中每个分量对应一个术语（词），而每个分量的值是该术语在文档中出现的次数。这会产生非常稀疏、高维的向量。这里，稀疏是指向量的大多数分量为0。

一旦我们用向量表示数据对象，我们就可以在数据上执行各种向量运算。例如，使用各种向量运算，我们可以计算两个向量的相似性或距离。具体而言，两个向量的余弦相似性定义为

$$\cos(\mathbf{u},\mathbf{v})=\frac{\mathbf{u}}{\|\mathbf{u}\|}\cdot\frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{A-4}$$

这种相似性度量不考虑向量的量值（长度），而只考虑两个向量点在相同方向的程度。就文档而言，这意味如果两个文档以相同比例包含相同术语，则它们相同。两个文档中都不出现的术语在计算相似性时不起作用。

我们还可以简单地定义两个向量（点）之间的距离。如果**u**和**v**是向量，则这两个向量（点）之间的欧几里得距离简单地定义为

$$dist(\mathbf{u},\mathbf{v})=\sqrt{(\mathbf{u}-\mathbf{v})\cdot(\mathbf{u}-\mathbf{v})} \tag{A-5}$$

对于鸢尾花数据集，这种度量更合适，因为在考虑是否相似时，向量各分量的量值确实很重要。

对于向量数据，计算向量集的均值也有意义。向量集的均值通过计算每个分量的均值实现。确实，有些聚类方法，如K-均值（第8章）就是将数据对象划分成组（簇），并用数据对象（数据向量）的均值刻画每个簇。其基本思想是，簇中数据对象靠近均值的簇是好的簇。其中，对于像鸢尾花这样的数据集，靠近用欧几里得距离度量，而对于像文档这样的数据集用余弦相似性度量。

数据上的其他常用运算也可以看作向量上的运算。考虑维归约。在最简单的方法中，数据向量中的某些分量被删除，而保留其他分量不变。其他维归约技术产生数据向量的新的分量（属性）集，这些新分量是原分量的线性组合。还有其他方法，使用更复杂的方法改变分量。维归约在附录B中进一步讨论。

对于某些数据分析领域（如统计学），分析技术用数据向量和包含这些数据向量的数据矩阵上的运算数学地表达。这样，向量表示带动了可以用来表示、变换和分析数据的强有力的数学工具。

在本附录的其余部分，我们将讨论矩阵。

A.2　矩阵

A.2.1　矩阵：定义

矩阵（matrix）是把数集合汇聚成行和列的一种表表示。我们将使用大写加粗的字母（如**A**）表示矩阵。（也使用大写斜体字母，如A。）术语“$m\times n$矩阵”通常用来说明矩阵具有m行和n列。例如，下面所示的矩阵**A**是2×3矩阵。如果$m=n$，则我们称该矩阵为**方阵**（square matrix）。矩阵**A**的转置记作$\mathbf{A}^T$，它通过交换**A**的行和列得到。

$$\mathbf{A}=\begin{bmatrix}2 & 6 & 1\\ 7 & 5 & 2\end{bmatrix}\qquad \mathbf{A}^T=\begin{bmatrix}2 & 7\\ 6 & 5\\ 1 & 2\end{bmatrix}$$

矩阵的元素用带下标的小写字母表示。例如，对于矩阵**A**，a_{ij}是其第i行第j列的元素。行

自上而下编号，而列自左向右编号。例如，$a_{21}=7$ 是矩阵 **A** 的第 2 行第 1 列的元素。

矩阵的每一行或列定义一个向量。对于矩阵 **A**，其第 i 个**行向量**（row vector）可以用 $\mathbf{a}_{i*}$ 表示，而第 j 个**列向量**（column vector）用 $\mathbf{a}_{*j}$ 表示。使用前面的例子，$\mathbf{a}_{2*}=[7\ 5\ 2]$，而 $\mathbf{a}_{*3}=[1\ 2]^T$。注意：行向量和列向量都是矩阵，必须加以区分，即元素个数相同并且值相同的行向量和列向量代表不同的矩阵。

A.2.2 矩阵：加法和与标量乘法

与向量一样，矩阵也可以通过将对应元素（分量）相加来求和。（这里，我们假设矩阵具有相同的行数和列数。）更明确地说，如果 **A** 和 **B** 都是 $m\times n$ 的矩阵，**A** 和 **B** 的和定义如下。

定义 A.3 矩阵加法 两个 $m\times n$ 矩阵 **A** 和 **B** 的和是 $m\times n$ 矩阵 **C**，其元素由下式计算：

$$c_{ij}=a_{ij}+b_{ij} \tag{A-6}$$

例如：

$$\begin{bmatrix}3 & 1\\ 1 & 2\end{bmatrix}+\begin{bmatrix}5 & 4\\ 2 & 9\end{bmatrix}=\begin{bmatrix}8 & 5\\ 3 & 11\end{bmatrix}$$

矩阵加法具有如下性质。

- **矩阵加法的交换律**。加的次序不影响结果：$\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$。
- **矩阵加法的结合律**。相加时矩阵分组不影响结果：$(\mathbf{A}+\mathbf{B})+\mathbf{C}=\mathbf{A}+(\mathbf{B}+\mathbf{C})$。
- **矩阵加法单位元的存在性**。存在一个**零矩阵**（zero matrix），其元素均为 0 并简记为 **0**，是单位元。对于任意矩阵 **A**，有 $\mathbf{A}+\mathbf{0}=\mathbf{A}$。
- **矩阵加法逆元的存在性**。对于每个矩阵 **A**，都存在一个矩阵 $-\mathbf{A}$，使得 $\mathbf{A}+(-\mathbf{A})=\mathbf{0}$。$-\mathbf{A}$ 的元素为 $-a_{ij}$。

与向量一样，也可以用标量乘以矩阵。

定义 A.4 矩阵的标量乘法 标量 α 和矩阵 **A** 的乘积是矩阵 $\mathbf{B}=\alpha\mathbf{A}$，其元素由下式给出：

$$b_{ij}=\alpha a_{ij} \tag{A-7}$$

矩阵的标量乘法具有与向量的标量乘法非常相似的性质。

- **标量乘法的结合律**。被两个标量乘的次序不影响结果：$\alpha(\beta\mathbf{A})=(\alpha\beta)\mathbf{A}$。
- **标量加法对标量与矩阵乘法的分配律**。两个标量相加后乘以一个矩阵等于每个标量乘以该矩阵之后的结果矩阵相加：$(\alpha+\beta)\mathbf{A}=\alpha\mathbf{A}+\beta\mathbf{A}$。
- **标量乘法对矩阵加法的分配律**。两个矩阵相加之后的和与一个标量相乘等于每个矩阵与该标量相乘然后相加：$\alpha(\mathbf{A}+\mathbf{B})=\alpha\mathbf{A}+\alpha\mathbf{B}$。
- **标量单位元的存在性**。如果 $\alpha=1$，则对于任意矩阵 **A**，有 $\alpha\mathbf{A}=\mathbf{A}$。

具有这些性质并不奇怪，因为我们可以认为矩阵由行向量或列向量组成，因此矩阵相加或用标量乘以矩阵等于对应的行向量或列向量相加或用标量乘它们。

A.2.3 矩阵：乘法

我们可以定义矩阵的乘法运算。先定义矩阵与向量的乘法。

定义 A.5　矩阵与列向量的乘法　$m \times n$ 矩阵 **A** 乘以 $n \times 1$ 的列矩阵 **u** 的积是 $m \times 1$ 的列矩阵 $\mathbf{v} = \mathbf{Au}$，其元素由下式给出：

$$v_i = \mathbf{a}_{i*} \cdot \mathbf{u}^T \tag{A-8}$$

换言之，我们取 **A** 的每个行向量与 **u** 的转置的点积。注意，在下面的例子中，**u** 的行数必然与 **A** 的列数相等。

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 17 \\ 9 \end{bmatrix}$$

类似地，我们可以定义矩阵被行向量左乘。

定义 A.6　矩阵与行向量的乘法　$1 \times m$ 的行矩阵 **u** 乘以 $m \times n$ 矩阵 **A** 的积是 $1 \times n$ 的行矩阵 $\mathbf{v} = \mathbf{uA}$，其元素由下式给出：

$$v_i = \mathbf{u} \cdot (\mathbf{a}_{*j})^T \tag{A-9}$$

换言之，我们取该行向量与矩阵 **A** 的每个列向量的转置的点积。下面给出一个例子：

$$\begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 5 & 4 \\ 2 & 9 \end{bmatrix} = \begin{bmatrix} 9 & 22 \end{bmatrix}$$

我们定义两个矩阵的乘积，作为上述概念的推广。

定义 A.7　$m \times n$ 矩阵 **A** 与 $n \times p$ 矩阵 **B** 的积是 $m \times p$ 矩阵 $\mathbf{C} = \mathbf{AB}$，其元素由下式给出：

$$c_{ij} = \mathbf{a}_{i*} \cdot (\mathbf{b}_{*j})^T \tag{A-10}$$

换言之，**C** 的第 ij 个元素是 **A** 的第 i 个行向量与 **B** 的第 j 个列向量转置的点积。

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 5 & 4 \\ 2 & 9 \end{bmatrix} = \begin{bmatrix} 17 & 21 \\ 9 & 22 \end{bmatrix}$$

矩阵乘法具有如下性质。

- **矩阵乘法的结合律**。矩阵乘的次序不影响计算结果：$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$。
- **矩阵乘法的分配律**。矩阵乘法对矩阵加法是可分配的：$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$ 并且 $(\mathbf{B} + \mathbf{C})\mathbf{A} = \mathbf{BA} + \mathbf{CA}$。
- **矩阵乘法单位元的存在性**。如果 $\mathbf{I}_p$ 是 $p \times p$ 矩阵，仅在对角线上为 1，其余地方为 0，则对于任意 $m \times n$ 矩阵 **A**，$\mathbf{AI}_n = \mathbf{A}$ 并且 $\mathbf{I}_m\mathbf{A} = \mathbf{A}$。（注意，单位矩阵是**对角矩阵**（diagonal matrix）的特例。对角矩阵的非对角线元素均为 0，即如果 $i \neq j$，则 $a_{ij} = 0$。）

一般地，矩阵乘法不是可交换的，即 $\mathbf{AB} \neq \mathbf{BA}$。

A.2.4　线性变换与逆矩阵

如果我们有一个 $n \times 1$ 列向量 **u**，则我们可以把 $m \times n$ 矩阵 **A** 被该向量右乘看作 **u** 到 m 维列向量 $\mathbf{v} = \mathbf{Au}$ 的变换。类似地，如果我们用一个（行）向量 $\mathbf{u} = [u_1, \cdots, u_m]$ 左乘 **A**，则我们可以将它看作 **u** 到 n 维行向量 $\mathbf{v} = \mathbf{uA}$ 的变换。这样，我们可以把一个任意 $m \times n$ 矩阵 **A** 看作一个把一个向量空间映射到另一个向量空间的函数。

在许多情况下，可以用更容易理解的术语描述变换（矩阵）。

- **缩放矩阵**（scaling matrix）不改变向量的方向，而是改变向量的长度。这等价于乘以一个乘了标量的单位矩阵得到的矩阵。

- **旋转矩阵**（rotation matrix）改变向量的方向但不改变向量的量值。这相当于改变坐标系。
- **反射矩阵**（reflection matrix）将一个向量从一个或多个坐标轴反射。这等价于用−1 乘该向量的某些元素，而保持其他元素不变。
- **投影矩阵**（projection matrix）把向量置于较低维子空间。最简单的例子是修改单位矩阵，将对角线上的一个或多个 1 改为 0。这样的矩阵消除对应于 0 元素的向量分量，而保留其他分量。

当然，单个矩阵可能同时进行两种类型的变换，如缩放和旋转。

把矩阵看作将向量从一个空间映射到另一个空间的函数时，矩阵具有如下性质。

- 矩阵是**线性变换**（linear transformation），即 $\mathbf{A}(\alpha\mathbf{u}+\beta\mathbf{v}) = \alpha\mathbf{A}\mathbf{u}+\beta\mathbf{A}\mathbf{v}$ 并且$(\alpha\mathbf{u} + \beta\mathbf{v})\mathbf{A} = \alpha\mathbf{u}\mathbf{A} + \beta\mathbf{v}\mathbf{A}$。
- 矩阵 **A** 变换后的所有行向量的集合称作 **A** 的**行空间**（row space），因为该矩阵的行向量或它们的某个子集形成变换后行向量子空间的一个基。这可以从下面的等式看出来。该等式把 $1 \times m$ 的行向量 $\mathbf{u} = [u_1, \cdots, u_m]$ 与 $m \times n$ 矩阵 **A** 的积表示为矩阵 **A** 的行的线性组合：

$$\mathbf{v} = \mathbf{u}\mathbf{A} = \sum_{i=1}^{n} u_i \mathbf{a}_{i*} \tag{A-11}$$

  行空间的维告诉我们 **A** 的线性独立的行数。
- 矩阵 **A** 变换后的所有列向量的集合称作 **A** 的**列空间**（column space）。矩阵的列向量或它们的某个子集形成变换后列向量子空间的一个基。这从下面的等式可以清楚。该等式把 $n \times 1$ 的列向量 $\mathbf{u} = [u_1, \cdots, u_n]^T$ 与 $m \times n$ 矩阵 **A** 的积表示为矩阵 **A** 的列的线性组合：

$$\mathbf{v} = \mathbf{A}\mathbf{u} = \sum_{j=1}^{n} u_j \mathbf{a}_{*j} \tag{A-12}$$

  列空间的维告诉我们 **A** 的线性独立的列数。
- **左零空间**（left nullspace）是被该矩阵映射到 **0** 的行向量。
- **右零空间**（right nullspace）（或更一般地，零空间）是被该矩阵映射到 **0** 的列向量。

注意，**矩阵的秩**（rank of a matrix）是行空间和列空间的最小维度，常常用来刻画矩阵。例如，如果我们把一个 $1 \times n$ 的行向量复制 m 次，产生一个 $m \times n$ 矩阵，则我们只有一个秩为 1 的矩阵。

一个实际和理论问题是矩阵是否像实数一样具有乘法逆元。首先，由于矩阵乘法的性质（即维必须匹配），如果矩阵具有**逆矩阵**（inverse matrix），它必须是方阵。这样，对于一个 $m \times m$ 矩阵 **A**，我们会问是否可以找到一个矩阵 $\mathbf{A}^{-1}$ 使得 $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_m$。答案是某些方阵有逆矩阵，而有些没有。

更抽象地说，一个 $m \times m$ 矩阵有逆矩阵，当且仅当它的零空间只包含 **0** 向量，或等价地，矩阵的行空间和列空间都是 m 维的。（这等价于矩阵的秩为 m。）从概念上讲，一个 $m \times m$ 矩阵有逆矩阵，当且仅当它把每个非零 m 维行（列）向量都映射到一个唯一的非零 m 维行（列）向量。

在求解各种矩阵方程时，逆矩阵的存在性是很重要的。

A.2.5 本征值与奇异值分解

现在，我们讨论线性代数的一个非常重要的问题：本征值和本征向量。本征值和本征向量，连同相关的奇异值和奇异向量概念，捕获了矩阵的结构，使得我们可以分解矩阵，并用标准形式

表示它们。因此，这些概念可以用于数学方程求解、维归约和降低噪声。我们从本征值和本征向量的定义开始讨论。

定义 A.8　本征向量和本征值　$m \times n$ 矩阵 $\mathbf{A}$ 的本征值和本征向量分别是标量值 λ 和向量 $\mathbf{u}$，它们是如下方程的解：

$$\mathbf{Au} = \lambda\mathbf{u} \tag{A-13}$$

换言之，**本征向量**（eigenvector）是被 $\mathbf{A}$ 乘时除量值外并不改变的向量。**本征值**（eigenvalue）是缩放因子。该方程也可以写成$(\mathbf{A} - \lambda\mathbf{I})\mathbf{u} = \mathbf{0}$。

对于方阵，可以使用本征值和本征向量分解矩阵。

定理 A.1　假设 $\mathbf{A}$ 是 $n \times n$ 矩阵，具有 n 个独立的（正交的）本征向量 $\mathbf{u}_1, \cdots, \mathbf{u}_n$ 和 n 个对应的本征值 $\lambda_1, \cdots, \lambda_n$。设 $\mathbf{U}$ 是矩阵，它的列是这些本征向量，即 $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$；并设 $\mathbf{\Lambda}$ 是对角矩阵，它的对角线元素是 $\lambda_i, 1 \leqslant i \leqslant n$。则 $\mathbf{A}$ 可以表示为

$$\mathbf{A} = \mathbf{U\Lambda U}^{-1} \tag{A-14}$$

这样，$\mathbf{A}$ 可以分解成 3 个矩阵的乘积。$\mathbf{U}$ 称为**本征向量矩阵**（eigenvector matrix），而 $\mathbf{\Lambda}$ 称为**本征值矩阵**（eigenvalue matrix）。

更一般地，任意矩阵都可以用类似的方法分解。更具体地说，任何 $m \times n$ 矩阵 $\mathbf{A}$ 都可以分解成 3 个矩阵的乘积，如下面的定理所述。

定理 A.2　假设 $\mathbf{A}$ 是 $m \times n$ 矩阵，则 $\mathbf{A}$ 可以表示如下：

$$\mathbf{A} = \mathbf{U\Sigma V}^T \tag{A-15}$$

其中 $\mathbf{U}$ 是 $m \times m$ 矩阵，$\mathbf{\Sigma}$ 是 $m \times n$ 矩阵，$\mathbf{V}$ 是 $n \times n$ 矩阵。$\mathbf{U}$ 和 $\mathbf{V}$ 是标准正交矩阵，即它们的列向量都是单位长度，并且相互正交。这样，$\mathbf{UU}^T = \mathbf{I}_m$，$\mathbf{VV}^T = \mathbf{I}_n$。$\mathbf{\Sigma}$ 是对角矩阵，其对角线元素非负，并且被排好序，使得较大的元素先出现，即 $\sigma_{i,i} \geqslant \sigma_{i+1,i+1}$。

$\mathbf{V}$ 的列向量 $\mathbf{v}_1, \cdots, \mathbf{v}_n$ 是**右奇异向量**（right singular vector），$\mathbf{U}$ 的列向量是**左奇异向量**（left singular vector）。**奇异值矩阵**（singular value matrix）$\mathbf{\Sigma}$ 的对角线元素通常记作 $\sigma_1, \cdots, \sigma_n$，称为 $\mathbf{A}$ 的**奇异值**（singular value）。（σ 的这种用法不要与使用 σ 表示变量的标准差混淆。）最多存在 $rank(\mathbf{A}) \leqslant \min(m, n)$个非零奇异值。

可以证明 $\mathbf{A}^T\mathbf{A}$ 的本征向量是右奇异向量（即 $\mathbf{V}$ 的列），而 $\mathbf{AA}^T$ 的本征向量是左奇异向量（即 $\mathbf{U}$ 的列）。$\mathbf{A}^T\mathbf{A}$ 和 $\mathbf{AA}^T$ 的非零本征值是 σ_i^2，即奇异值的平方。的确，方阵的本征值分解可以看作奇异值分解的一个特例。

矩阵的奇异值分解（Singular Value Decomposition，SVD）也可以用下面的等式表示。注意，尽管看上去像点积，但它并不是点积，其结果是秩为 1 的 $m \times n$ 矩阵。

$$\mathbf{A} = \sum_{i=1}^{rank(\mathbf{A})} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{A-16}$$

这种表示的重要性是每个矩阵都可以表示成秩为 1 矩阵的以奇异值为权重的加权和。由于以非递增序排列的奇异值通常下降很快，因此有可能使用少量奇异值和奇异向量得到矩阵的很好的近似。这对于维归约是很有用的，这将在附录 B 中进一步讨论。

A.2.6 矩阵与数据分析

我们可以把数据集表示成数据矩阵，其中每一行存放一个数据对象，而每一列是一个属性。（同样，我们也可以用行表示属性，列表示对象。）矩阵表示为我们的数据提供了紧凑、结构良好的表示，使得我们可以很容易地通过各种矩阵运算对数据对象或属性进行操作。

线性方程组是使用数据的矩阵表示的很常见的例子。线性方程组可以写成一个矩阵方程 $\mathbf{Ax} = \mathbf{b}$，并使用矩阵运算求解。

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

特殊地，如果 $\mathbf{A}$ 有逆矩阵，则该方程组的解为 $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$。如果 $\mathbf{A}$ 没有逆矩阵，则该方程组或者没有解，或者有无穷多个解。注意，在这种情况下，行（数据对象）是方程，列是变量（属性）。

对于许多统计学和数据分析问题，我们希望解线性方程组，但是这些线性方程组不能使用刚才介绍的方法求解。例如，我们可能有一个数据矩阵，其中行代表病人，而列代表病人的特征（身高、体重和年龄）和他们对特定药物治疗的反应（如血压的变化）。我们想把血压（因变量）表示成其他（自）变量的线性函数，并且可以用上面的方法写一个矩阵方程。然而，如果我们的病人比变量多（通常如此），则矩阵的逆不存在。

在这种情况下，我们仍然想找出该方程组的最好解。这意味我们想找出自变量的最好线性组合来预测因变量。使用线性代数的术语，我们想找到尽可能接近向量 $\mathbf{b}$ 的向量 $\mathbf{Ax}$；换句话说，我们希望最小化向量 $\mathbf{b} - \mathbf{Ax}$ 的长度$||\mathbf{b} - \mathbf{Ax}||$。这称作**最小二乘**（least square）问题。许多统计学技术（例如，将在附录 D 中讨论的线性回归）都需要解最小二乘问题。可以证明，方程 $\mathbf{Ax} = \mathbf{b}$ 的最小二乘解是 $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$。

在分析数据时，特别是对于维归约（将在附录 B 中讨论），奇异值和本征向量分解也非常有用。注意，维归约还可以带来降低噪声的效果。

尽管我们给出了一些线性代数应用的例子，但是我们省略的更多。其他需要使用线性代数形式化和求解的领域包括微分方程组的求解、优化问题（如线性规划）和图分割。

A.3 文献注释

许多书都很好地讲述了线性代数，包括 Demmel [521]、Golub 和 Van Loan [522]以及 Strang [523]的书。

参考文献

[521] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM Press, September 1997.（中文版《应用数值线性代数》已由人民邮电出版社出版。）

[522] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, November 1996.

[523] G. Strang. *Linear Algebra and Its Applications*. Harcourt Brace & Company, Orlando, FL, 3rd edition, 1986.

附录B

维 归 约

本附录考虑各种维归约技术，目的是向读者揭示所涉及的问题，介绍一些较常用的方法。我们从讨论 PCA（Principal Components Analysis，主成分分析）和 SVD（Singular Value Decomposition，奇异值分解）开始。我们会详细介绍这些方法，因为它们是最常用的方法，并且可以建立在附录 A 的线性代数讨论的基础之上。但是，还有一些其他方法也能用于维归约，因此我们也简略讨论一些其他方法。最后，我们简要评述一些重要问题。

B.1 PCA 和 SVD

PCA 和 SVD 是两种密切相关的技术。PCA 不考虑数据均值，而 SVD 考虑。这些技术已经在诸多领域广泛使用了数十年。在下面的讨论中，我们假定读者熟悉附录 A 中讨论的线性代数内容。

B.1.1 PCA

PCA 的目标是找出一个更好地捕获数据变异性的、新的维（属性）集合。更明确地说，所选取的第一个维要尽可能多地捕获数据的变异性。第二个维与第一个正交，并且尽可能多地捕获剩余的变异性，如此下去。

PCA 具有一些引人注目的特性。首先，它趋向于确定数据中最强的模式。因此，PCA 可以用作模式发现技术。其次，数据的大部分变异性通常都可以被整个维集合的一小部分新维所捕获。这样，使用 PCA 进行维归约可以产生相对低维的数据，使得我们有可能使用在高维数据上不太有效的技术。再次，由于数据中的噪声比模式弱（希望如此），维归约可以去掉许多噪声。这有利于数据挖掘和其他数据分析算法。

我们简要讨论 PCA 的数学基础，然后给出一个例子。

数学细节

统计学家通过计算数据的协方差矩阵 **S** 汇总多元数据集（例如，具有多个连续属性的数据）的变异性。

定义 B.1 给定一个 $m \times n$ 的数据矩阵 **D**，其 m 个行是数据对象，其 n 个列是属性。**D** 的协方差矩阵是矩阵 **S**，其元素 s_{ij} 定义为

$$s_{ij} = covariance(\mathbf{d}_{*i}, \mathbf{d}_{*j}) \tag{B-1}$$

换言之，s_{ij} 是数据的第 i 和第 j 个属性（列）的协方差。

两个属性的协方差在附录 C 中定义，它度量两个属性一起变化的程度。如果 $i = j$（即两个属性相同），则协方差就是该属性的方差。如果数据矩阵 **D** 经过预处理，使得每个属性的均值都是

0，则 $\mathbf{S} = \mathbf{D}^T\mathbf{D}$。

PCA 的目标是找到一个满足如下性质的数据变换。

(1) 每对（不同的）新属性的协方差为 0。

(2) 属性按照每个属性捕获的数据方差的多少排序。

(3) 第一个属性捕获尽可能多的数据方差。

(4) 在满足正交性的前提下，每个后继属性捕获尽可能多的剩余方差。

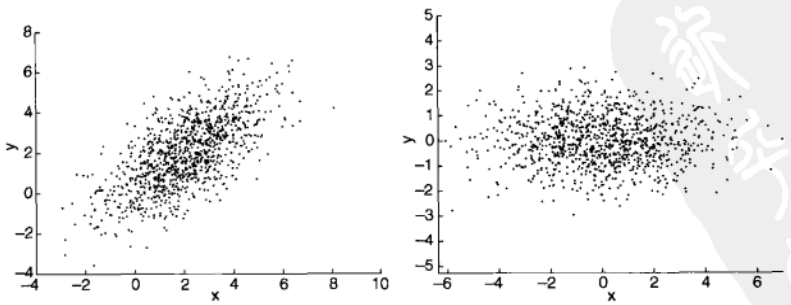
具有这些性质的数据变换可以通过协方差矩阵的本征值分析得到。令 $\lambda_1, \cdots, \lambda_n$ 是 $\mathbf{S}$ 的本征值。这些本征值都是非负的，并且可以排序，使得 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \lambda_{m-1} \geqslant \lambda_m$。（协方差矩阵是一种**半正定矩阵**（positive semidefinite matrix），其性质之一是具有非负本征值。）令 $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$ 是 $\mathbf{S}$ 的本征向量矩阵。这些本征向量已排序，使得第 i 个本征向量对应于第 i 个最大的本征值。最后，假设已经对数据矩阵 $\mathbf{D}$ 进行过预处理，使得每个属性（列）的均值均为 0。我们可以做如下陈述。

- 数据矩阵 $\mathbf{D}' = \mathbf{DU}$ 是变换后的数据集，满足以上条件。
- 每个新属性都是原属性的线性组合。更明确地说，第 i 个属性的线性组合权重是第 i 个本征向量的分量。这一结论由 $\mathbf{D}'$ 的第 j 列是 $\mathbf{D}\mathbf{u}_j$ 这一事实和公式（A-12）给出的矩阵-向量乘法定义推出。
- 第 i 个新属性的方差是 λ_i。
- 原属性的方差和等于新属性的方差和。
- 新属性称作**主成分**（principal component），也就是说，第一个新属性是第一个主成分，第二个新属性是第二个主成分，如此下去。

与最大本征值相关联的本征向量指示数据具有最大方差的方向。换言之，就所有可能的方向而言，如果所有数据投影到该向量定义的直线上，则结果值将具有最大方差。与次大本征值相关联的本征向量（正交于第一个本征向量）是具有最大剩余方差的数据的方向。

$\mathbf{S}$ 的本征向量定义了一个新的坐标系。确实，PCA 可以看作原坐标系到新坐标系的旋转变换。新坐标轴按数据的变异性排列。变换保持数据的总变异性，但是新属性是不相关的。

例 B.1　二维数据　我们用例子说明使用 PCA 把坐标轴校正到数据最大变异性的方向。图 B-1 显示了 PCA 变换前后的 1000 个二维数据点。原坐标系中数据的总方差为 x 和 y 属性上的方差和，等于 $2.84 + 2.95 = 5.79$。变换后，方差为 $4.81 + 0.98 = 5.79$。 □

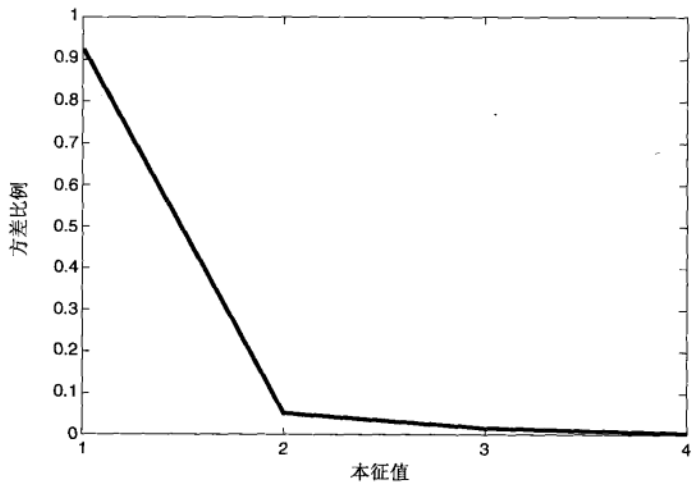


(a) 原数据点　　(b) 变换后的数据点

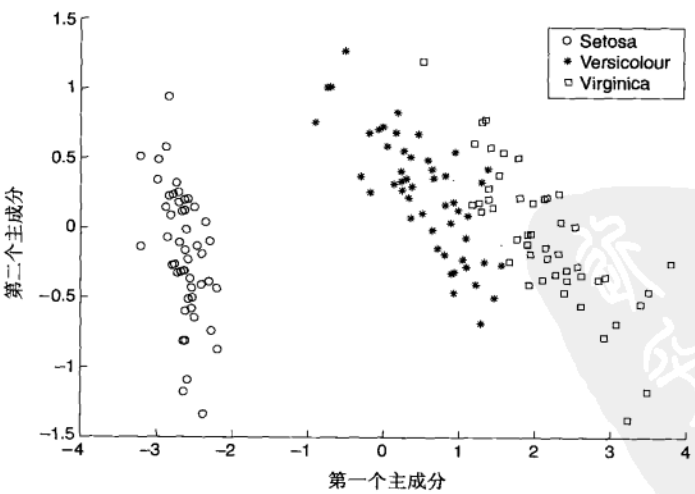
图 B-1　使用 PCA 变换数据

例 B.2 鸢尾花数据 这个例子使用鸢尾花数据集解释使用 PCA 进行维归约。该数据集包含 150 个数据对象（花）；有 50 种花，取自鸢尾花的 3 个不同品种，即 Setosa、Versicolour 和 Virginica。每种花用 4 个属性刻画，即萼片长度、萼片宽度、花瓣长度和花瓣宽度。更详细的介绍见第 3 章。

图 B-2a 显示协方差矩阵的每个本征值（主成分）导致的方差所占的比例。这种类型的图称作**斜坡图**（scree plot），可以用来确定需要多少主成分来捕获数据的大部分变异性。对于鸢尾花数据，第一个主成分捕获了方差的大部分（92.5%），第二个仅 5.3%，而其余两个仅仅为 2.2%。因此，只需要保留前两个主成分就能保持数据变异性的大部分。图 B-2b 显示鸢尾花数据基于前两个主成分的散布图。注意，Setosa 花与 Versicolour 和 Virginica 花很好地分开。尽管后两种花更靠近，但是仍然相对较好地分开。 □



(a) 每个主成分导致的方差所占的比例



(b) 鸢尾花数据前两个主成分上的散布图

图 B-2 PCA 用于鸢尾花数据

B.1.2 SVD

如果不考虑每个变量的均值，PCA 等价于 SVD 分析。尽管如此，从 SVD 角度考察维归约仍然是有意义的，因为我们并非总是希望排除数据的均值，特别是当数据相对稀疏时更是如此。

数学细节

从附录 A 可知，$m \times n$ 矩阵 **A** 可以表示为

$$\mathbf{A} = \sum_{i=1}^{rank(\mathbf{A})} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{B-2}$$

其中 σ_i 是 **A** 的第 i 个奇异值（$\mathbf{\Sigma}$ 的第 i 个对角线元素），$\mathbf{u}_i$ 是 **A** 的第 i 个左奇异向量（**U** 的第 i 列），而 $\mathbf{v}_i$ 是 **A** 的第 i 个右奇异向量（**V** 的第 i 列）。（参见 A.2.5 节。）数据矩阵的 SVD 分解具有如下性质。

- 属性中的模式被右奇异向量（即 **V** 的列）捕获。
- 对象中的模式被左奇异向量（即 **U** 的列）捕获。
- 矩阵 **A** 可以通过依次取公式（B-2）中的项，以最优的方式不断逼近。我们不解释“最优”的涵义，读者可以参阅文献注释。非形式地说，奇异值越大，该奇异值和其相关联的奇异向量决定矩阵的比例越大。
- 为了得到具有 k 个属性的新数据矩阵，我们计算矩阵 $\mathbf{D}' = \mathbf{D} * [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k]$。从前面的讨论看来，我们应该取公式（A-12）的前 k 项得到的矩阵。然而，尽管结果矩阵的秩为 k，但是它仍然有 n 个列（属性）。

例 B.3 文档数据 可以使用 SVD 分解分析文档数据。本例的数据包含取自《洛杉矶时报》的 3024 篇文章。这些文章取自 6 个不同的版块：娱乐、财经、国外、都市、国内和体育。数据矩阵是文档-术语矩阵，其中每行代表一个文档，每列代表一个术语（词）。第 ij 个元素是第 j 个术语在第 i 个文档中出现的次数。数据已经过处理，使用标准技术删除了常用词，调整了术语出现的频率，并且调整了文档长度。（更多细节，参见 2.3.7 节。）

该数据的 SVD 分析旨在找出前 100 个奇异值和向量。（对于许多数据集，找出全部 SVD 或 PCA 分解都开销太大，并且常常没有意义，因为只需要相对较少的奇异值或本征值就能捕获矩阵的结构。）最大的奇异值与常用术语有关，它们是频繁的但是未被预处理删除。（可能出现这种情况，最强的模式表示噪声或不感兴趣的模式。）

然而，与其他奇异值相关联的模式更有意义。例如，下面是与第二个右奇异向量的最强分量相关联的前 10 个术语（词）：

```
game, score, lead, team, play, rebound, season, coach, league, goal
```

所有这些术语都与体育有关。毫不奇怪，与第二个左奇异向量的最强分量相关联的文档主要取自体育版块。

与第三个右奇异向量的最强分量相关联的前 10 个术语如下：

```
earn, million, quarter, bank, rose, billion, stock, company, corporation, revenue
```

这些都是财经术语，并且毫不奇怪，与第三个左奇异向量的最强分量相关联的文档主要取自财经版块。

使用第二和第三个奇异向量，即使用 $\mathbf{D}' = \mathbf{D} * [\mathbf{v}_2, \mathbf{v}_3]$，我们降低了该数据的维度。换言之，所有的文档都用两个属性表示，一个涉及体育，另一个涉及财经。文档的一个散点图由图 B-3 给

出。为清晰起见，删除了非体育、非财经文档。体育文档用浅灰色表示，而财经文档用深灰色表示。两个不同范畴的文档大部分都完全分开。确实，体育文档关于财经变量（分量 3）变化不大，而财经文档关于体育变量（分量 2）变化不大。 □

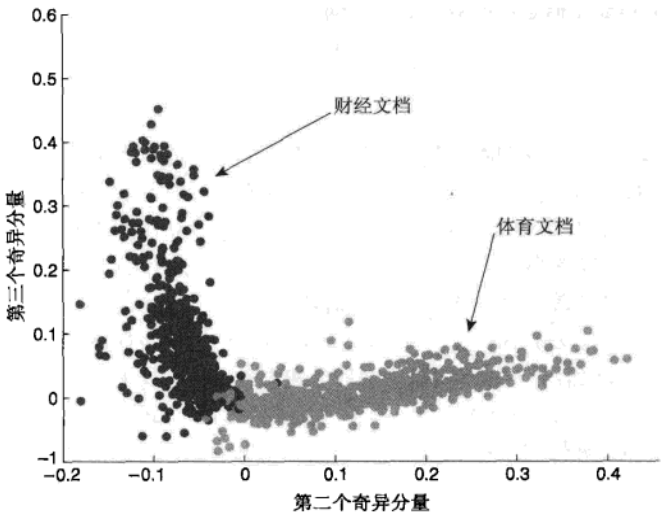


图 B-3 使用第二和第三个奇异值的《洛杉矶时报》的体育和财经文档的散点图

B.2 其他维归约技术

本节，我们考察几种其他维归约技术。这些技术的讨论将更简略，主要关注一般动机和方法。

B.2.1 因子分析

对于 PCA 和 SVD，产生的新属性是原变量的线性组合。进行因子分析的目标是将原变量表示成少数**隐藏**（hidden）或**潜在**（latent）属性的线性组合。这样做的动机是基于如下观察：数据对象常常有些特征很难直接测量，但是它们看上去与可测量的特征相关。一个常见的例子是智能和做各种 IQ（智商）测试成绩。另一个常见的例子是各种运动项目成绩与运动员的速度和力量之间的联系。如果能够找到少量属性来对原始属性进行分组和汇总，则我们可以实现维归约，并有助于对数据的理解。

有时，因子分析的动机也用数据的协方差或协相关矩阵解释。假设一组属性与其他属性不高度相关，但是它们之间强相关，或许因为它们度量相同的潜在量。在这种情况下，看起来需要开发一种技术，可以对每个这样的组找出汇总该组的单个潜在属性。

例如，考虑记录一组运动员十项全能的每个单项成绩的数据集。我们可能发现一个运动员在要求速度的所有项目中都大致呈现相同的表现，即速度慢的运动员总是慢，而速度快的运动员总是快。类似地，我们可能发现运动员在一个要求力量的项目中的表现预示着他在另一个要求力量的项目中的表现。因此，我们可以假设一个运动员在任何项目中的成绩实际上是由该项目的特性和速度与力量两个潜在因素决定的。因子分析试图发现这种联系。

更形式化地说，令 $\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_p$ 为**潜在因子**（latent factor），即潜在或隐藏属性。注意，这些

是新属性，每个对象都有一个值。如果原数据矩阵 **D** 是 $m \times n$ 矩阵，则新数据矩阵是 $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_p]$，为 $m \times p$ 矩阵。（注意，$\mathbf{f}_{*j} = \mathbf{f}_j$。）**F** 的第 ij 个元素是 f_{ij}，$\mathbf{f}_i$ 的第 j 个分量。

假设每个属性的均值均为 0。如果 $\mathbf{d}_{i*}$ 是原数据矩阵 **D** 的第 i 行，则 $\mathbf{f}_{i*}$ 是新数据矩阵 **F** 的对应行。标准因子分析模型假定新旧数据对象之间存在如下联系：

$$\mathbf{d}_{i*}^T = \mathbf{\Lambda}\mathbf{f}_{i*}^T + \boldsymbol{\varepsilon} \tag{B-3}$$

或等价地有

$$d_{ij} = \lambda_{j1} f_{i1} + \lambda_{j2} f_{i2}, \cdots, \lambda_{jp} f_{ip} + \varepsilon_i \tag{B-4}$$

Λ 的元素为 λ_{kl}，是 $n \times p$ 的**因子载荷**（factor loading）矩阵，指示原来的值对潜在因子（即新属性）的依赖程度。为了说明这一点，考虑十项全能的例子。这里有两个潜在因子：速度和力量，对应于的 **F** 的列。每个运动员用 **F** 的一行表示，记录运动员的速度和力量。**D** 的每列对应十项全能的每个单项，而每行也对应于一个运动员。**D** 的第 ij 个元素是第 i 个运动员第 j 个项目的成绩。**Λ** 将是 10×2 矩阵。如果 **D** 的第一列记录运动员的 100 米短跑成绩，则第 i 个运动员的 100 米短跑成绩写为 $d_{i1} = \lambda_{11} f_{i1} + \lambda_{12} f_{i2}$，其中 f_{i1} 是指示第 i 个运动员的速度值，而 f_{i2} 是指示第 i 个运动员的力量值。λ_{11} 和 λ_{12} 分别指示运动员的速度和力量如何加权来预测运动员的 100 米短跑成绩。我们预料与 λ_{12} 相比，λ_{11} 相对较大。注意，对于所有对象（运动员），这些权重相同。

由于在确定任意原属性的值时涉及所有的潜在因子，因此它们又称**公共因子**（common factor）。ε 是误差项，处理属性未被公共因子涵盖的部分，因此 ε 的分量称作**特殊因子**（specific factor）。

例 B.4　鸢尾花数据的因子分析　这个例子基于鸢尾花数据集。对于该数据，只能找到一个因子。鸢尾花数据集中的花这样组织：前 50 种花是 Setosa 种属，中间 50 种花是 Versicolour 种属，而最后 50 种花是 Virginica 种属。花的单个因子（属性）如图 B-4 所示。这个因子看上去捕获了三个种属之间的差别。 □

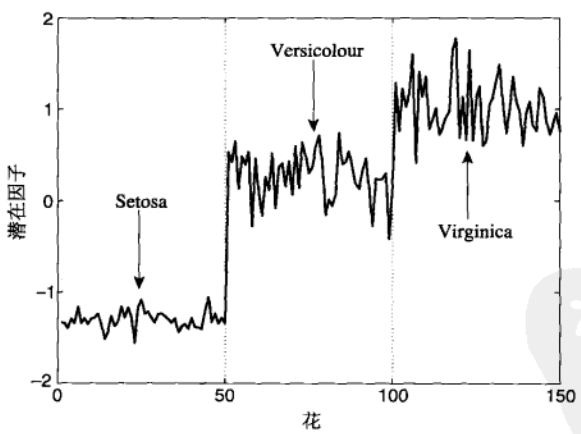


图 B-4　鸢尾花数据集中的花与单个潜在因子图示

B.2.2　LLE

LLE（Local Linear Embedding，局部线性嵌入）是一种维归约技术，它基于这样一种思想：分析重叠的局部邻域，以确定局部结构。LLE 算法在下面给出。

算法 B.1 LLE 算法

1：找出每个数据点的最近邻。
2：把每个点 $\mathbf{x}_i$ 表示成其他点的线性组合，即 $\mathbf{x}_i = \sum_j w_{ij}\mathbf{x}_j$，其中 $\sum_j w_{ij} = 1$，并且如果 $\mathbf{x}_j$ 不是 $\mathbf{x}_i$ 的最近邻，则 $w_{ij} = 0$。
3：使用步骤 2 得到的权重，找出每个点在指定维 p 的较低维空间的坐标。

在步骤 2 中，通过最小化下式给出的近似误差平方找出其元素为 w_{ij} 的权重矩阵 $\mathbf{W}$。可以通过求解最小二乘问题找出 $\mathbf{W}$。（这种问题在附录 A 中讨论过。）

$$error(\mathbf{W}) = \sum_i \left(\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_j \right)^2. \tag{B-5}$$

步骤 3 完成实际的维归约。给定权重矩阵和用户指定的维数 p，算法构造数据的“保持邻域嵌入”到较低维空间。如果 $\mathbf{y}_i$ 是低维空间中的向量，对应于 $\mathbf{x}_i$，而 $\mathbf{Y}$ 是新数据矩阵，其第 i 行是 $\mathbf{y}_i$，则这一步可以通过最小化下式找到 $\mathbf{Y}$ 来实现：

$$error(\mathbf{Y}) = \sum_i \left(\mathbf{y}_i - \sum_j w_{ij}\mathbf{y}_j \right)^2 \tag{B-6}$$

例 B.5 使用鸢尾花数据集来说明 LLE 用于维归约。具体地说，数据被投影到二个维上，使用 30 个点的邻域。投影后的数据的散点图在图 B-5 中。数据也可以投影到一个维。在这种情况下，看上去与图 B-4 很相似。 □

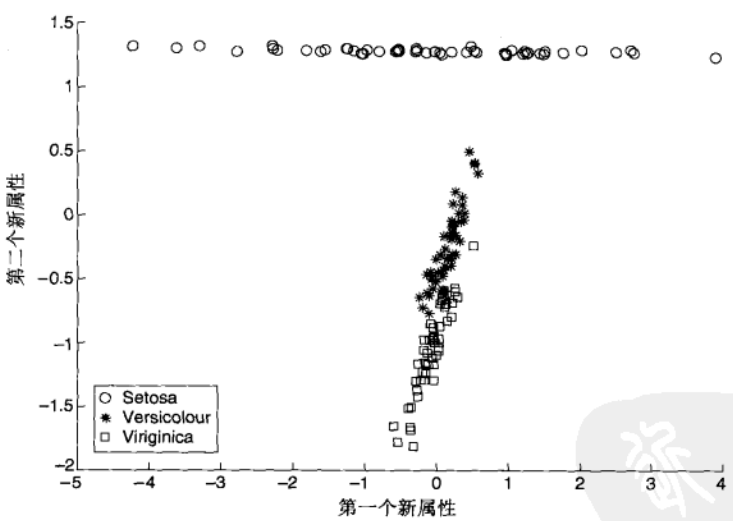


图 B-5 鸢尾花数据集中的花基于 LLE 的两个新属性的图

B.2.3 MDS、FastMap 和 ISOMAP

MDS（Multi Dimensional Scaling，多维缩放）是一种常用于维归约的技术。已经提出了这种技术的各种变形，但是这些技术的一般策略是相同的：找出数据到低维空间的投影，尽可能保持逐对距离（用一个目标函数度量）。正因为这种策略，MDS 从相异矩阵出发，因此甚至可以适

用于原先没有向量空间表示的数据，如字符串。

1. 标准 MDS 技术

我们从介绍把数据投影到 p 维空间的经典 MDS 方法开始。假设我们有距离矩阵 $\mathbf{D}$，其中 d_{ij} 是第 i 个与第 j 个对象之间的距离。令 d'_{ij} 是这两个对象经过变换后的距离。经典的 MDS 试图把每个对象指派到一个 p 维空间的点，使得一个称为**应力**（stress）的量最小。其中应力定义为

$$stress = \sqrt{\frac{\sum_{ij}(d'_{ij} - d_{ij})^2}{\sum_{ij} d_{ij}^2}} \tag{B-7}$$

MDS 的经典版本是**度量 MDS**（metric MDS）技术的一个实例，它假定相异度是连续变量（区间或比率）。**非度量 MDS**（non-metric MDS）技术假定数据是分类的（最好是序数的）。我们不讨论这些算法的细节，只指出典型的方法是以某种方式把对象初始指派到 p 维空间点，然后修正这些点，以降低应力。

当经典的 MDS 或 MDS 某种标准变形应用于鸢尾花数据集时，它们产生的结果将与图 B-2 所示结果几乎完全一样。实际上，对于欧几里得距离，经典的 MDS 等价于 PCA。

2. FastMap

MDS 领域的最近进展是 FastMap（快速映射）算法。它与其他 MDS 技术的目标相同，但有两个重要差别。

- 它比较快——线性复杂度。
- 它可以增量执行。

FastMap 算法识别一对对象，然后计算这一方向上其余对象的距离。这可以通过利用特定的几何事实（即余弦律），仅使用逐对距离来实现。取该距离为第一个属性的值，之后把对象投影到一个$(n-1)$维子空间。这也可以仅使用逐对距离来完成。然后重复这一过程。

最初，FastMap 算法应用于整个数据集。然而，如果我们记录每步选取的对象对，则可以对新对象增量地使用 FastMap。所需要的信息仅仅是新对象到被选取的对象对的距离。

3. ISOMAP

当点之间存在复杂的非线性关系时，MDS 和 PCA 的维归约效果并不好。（一个例外是核 PCA，参见文献注释。）ISOMAP 是传统 MDS 的扩展，用来处理这种数据集。它可以处理的这类数据集的一个例子在图 B-6 给出，这是一个“瑞士卷”曲面。具有这种结构的数据集形成三维空间的二维数据集，但不能被 PCA 或 MDS 有效处理。然而，ISOMAP 可以成功地分析这种数据集。

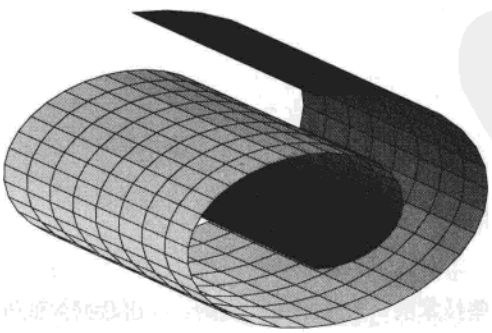


图 B-6　瑞士卷数据集的图示

算法 B.2 给出了基本 ISOMAP 算法的概要。最近邻可以取 k 个最近的点（其中 k 是参数），或者取指定半径内的所有点。第 2 步的目的是计算测地距离（geodesic distance），即两个点在所在面上的距离，而不是欧几里得距离。作为一个例子，地球两侧的两个城市之间的欧几里得距离是穿越地球的线段长度，而两个城市之间的测地距离是地球表面最短弧的长度。

算法 B.2 ISOMAP 算法

1：找出每个数据点的最近邻，并将每个点连接到它的最近邻，创建一个加权图。图的结点是数据点，而边的权重是点之间的距离。
2：重新定义点之间的距离为近邻图中两个点之间的最短路径长度。
3：对新的距离矩阵应用经典的 MDS。

例 B.6 使用 ISOMAP 把鸢尾花数据投影到二维空间。见图 B-7。结果与先前的技术类似。□

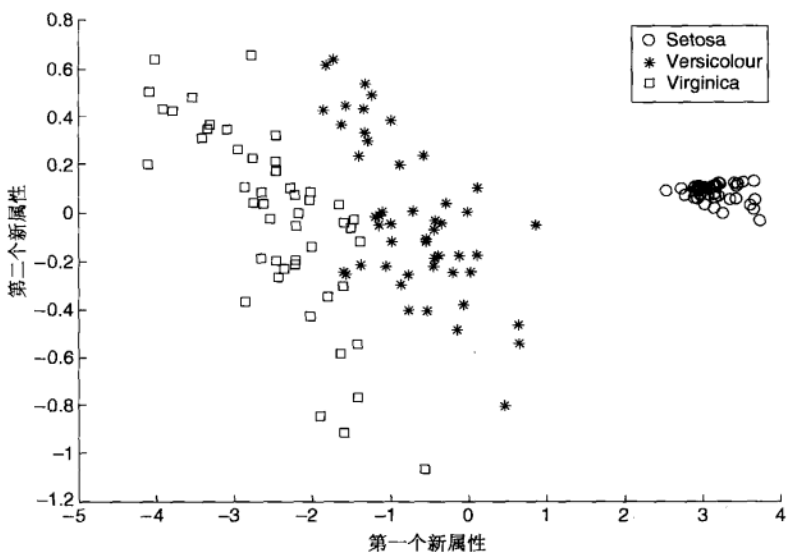


图 B-7 鸢尾花数据集中的花基于 ISOMAP 得到的两个新属性的图

B.2.4 常见问题

与其他数据分析技术一样，在许多领域我们都可以区别不同的维技术。一个关键问题是结果的质量：一种技术能够产生相当可靠的、数据的较低维空间的表示吗？这种表示能够捕获那些对预期的应用（例如聚类）很重要的数据特征，而删除不相关甚至有害的（例如噪声）那些方面吗？

从很大程度上讲，答案依赖于可以被维归约方法分析的数据类型和数据分布。像 PCA、SVD 和因子分析假定新旧属性集之间存在线性关系。尽管在许多情况下这种假定都近似成立，但是还有一些情况需要非线性方法。特别是，开发了像 ISOMAP 和 LLE 这样的算法来处理非线性关系。

维归约算法的时间和空间复杂度也是一个关键问题。我们讨论的大部分算法的时间或空间复杂度都是 $O(m^2)$或更高，其中 m 是对象数。对于大型数据集，尽管抽样有时可能相当有效，但是这仍然限制了它们可用性。FastMap 是本附录给出的唯一具有线性时间和空间复杂度的维归约算法。

维归约算法的另一个重要问题是：它们每次运行是否产生相同的结果。PCA、SVD 和 LLE 确实如此，而因子分析和 MDS 在不同的运行可能产生不同的结果。许多我们未讨论的技术也具有这种性质，因为它们试图优化某个目标，而这需要搜索，可能陷入局部极小。基于搜索的方法还可能具有很高的时间复杂度。

最后，一个关键问题是确定维归约的维数。我们已经考虑的这些技术通常可以把维归约到任意维数。归约的质量通常用可以用某种图示（如用斜坡图）的质量测量。在某些情况下，这些曲线清楚地指示了固有维度。在其他一些情况，选择需要在维数较少但近似误差较大和近似误差较小但维数更多之间权衡。

B.3 文献注释

维归约是一个很广泛的课题，相关文献散布在许多领域。PCA 的全面讨论可以在 Jolliffe [531] 的书中找到，而 SVD 的介绍由 Demmel[527]和其他线性代数教材给出。Schölkopf 等[534]描述了核 PCA。许多多元统计分析的书，如 Anderson[524]，也包含了 PCA 的讨论以及因子分析。MDS 的更多细节可以在 Kruskal 和 Wish[532]的书中找到。FastMap 算法是 Faloutsos 和 Lin [529]提出的。关于 LLE 的文章（Roweis 和 Saul [535]）和 ISOMAP 的文章（Tenenbaum et al.[533]）出现在《科学》（*Science*）同一专题上。ISOMAP 和 LLE 算法的 MATLAB 代码可以在 Web 上找到。其他可能感兴趣的文章包括 M. Belkin 和 P. Niyogi [525]、Donoho 和 Grimes[528]以及 Ye 等 [536, 537]的文章。

有许多其他技术常常用于维归约，或与维归约非常相关。这些领域包括主曲线、主曲面、非线性 PCA（包括神经网络方法）、向量量化、随机投影、独立成分分析（ICA）、自组织映射（SOM）、投影寻踪、基于回归的方法、遗传算法和诸如模拟或确定性退火等基于优化的方法。关于这些领域的介绍和附加的参考文献可以在 Fodor [530]和 Carreira-Perpinan [526]的关于维归约的两篇综述中找到。SOM 在 9.2.3 节讨论过。

参考文献

[524] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2nd edition, July 2003.

[525] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Technical Report TR 2002-01, Department of Computer Science and Statistics, University of Chicago, January 2002.

[526] M. A. Carreira-Perpinan. A Review of Dimension Reduction Techniques. Technical Report CS–96–09, Dept. of Computer Science, University of Sheffield, January 1997.

[527] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM Press, September 1997.

[528] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*, 100(10):5591–5596, 2003.

[529] C. Faloutsos and K.-I. Lin. FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In *Proc. of the 1995 ACM SIGMOD Intl. Conf. on Management of Data*, pages 163–174, San Jose, California, June 1995.

[530] I. K. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID- 148494, LLNL, June 2002.

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

[532] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. SAGE Publications, January 1978.

[533] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[534] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.

[535] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[536] J. Ye, R. Janardan, and Q. Li. GPCA: an efficient dimension reduction scheme for image compression and retrieval. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 354–363, Seattle, Washington, August 2004. ACM.

[537] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar. IDR/QR: an incremental dimension reduction algorithm via QR decomposition. In *Proc. of the 10th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 364–373, Seattle, Washington, 2004. ACM.

附录 C

概 率 统 计

本附录提供本书使用的一些概率论和统计学的基本概念。

C.1 概率

随机实验（random experiment）是测量其结果不确定的过程的实验。随机实验的例子包括掷一个色子，从一副牌中抽牌，监测网络路由器中的通信类型。随机实验的所有可能结果的集合称为**样本空间**（sample space）Ω。例如，对于掷一个色子，$\Omega = \{1, 2, 3, 4, 5, 6\}$是样本空间。**事件**（event）$E$ 对应于这些结果的一个子集，即 $E \subseteq \Omega$。例如，$E = \{2, 4, 6\}$是掷一个色子时观察到偶数点的事件。

概率 P 是定义在样本空间 Ω 上的实数值函数，满足如下性质。

(1) 对于任意事件 $E \subseteq \Omega$，$0 \leqslant P(E) \leqslant 1$。

(2) $P(\Omega) = 1$。

(3) 对于任意不相交的事件集 $E_1, E_2, \cdots, E_k \in \Omega$，

$$P(\bigcup_{i=1}^{k} E_i) = \sum_{i=1}^{k} P(E_i)$$

事件 E 的概率记作 $P(E)$，是在可能无穷多次实验中观测到 E 的次数所占的比例。

在随机实验中，通常有一个我们想测量的量。例如，统计掷 50 次硬币背面朝上的次数，或测量主题公园中坐转轮的游客的身高。因为这种量值依赖于随机实验的结果，所以这种感兴趣的量称为**随机变量**（random variable）。随机变量的值可能是离散的或连续的。例如，伯努利随机变量是离散随机变量，其可能的值只有 0 和 1。

对于离散随机变量 X，X 取特定值 ν 的概率是 $X(e) = \nu$ 的所有结果 e 的总概率。

$$P(X = \nu) = P(E = \{e | e \in \Omega, X(e) = \nu\}) \tag{C-1}$$

离散随机变量 X 的概率分布也称它的**概率质量函数**（probability mass function）。

例 C.1 考虑随机投一枚均匀硬币 4 次的随机实验。该实验有 16 种可能的结果，即 HHHH、HHHT、HHTH、HTHH、THHH、HHTT、HTHT、THHT、HTTH、THTH、TTHH、HTTT、THTT、TTHT、TTTH 和 TTTT，其中 H（T）表示观察到正面（背面）。设 X 是随机变量，度量在实验中观测到背面的次数。X 的 5 个可能值是 0, 1, 2 , 3, 4 和 5。X 的概率质量函数由下表给出。

X	0	1	2	3	4
$P(X)$	1/16	4/16	6/16	4/16	1/16

例如，$P(X = 2) = 6/16$，因为在 4 次投掷中观测到两次背面的结果有 6 种。 □

另一方面，如果 X 是连续随机变量，则 X 的值在 a 和 b 之间的概率为

$$P(a < x < b) = \int_a^b f(x)\mathrm{d}x \tag{C-2}$$

函数 $f(x)$称为**概率密度函数**（probability density function，pdf）。因为 f 是连续分布，所以 X 取特定值 x 的概率总是为 0。

表 C-1 显示了一些著名的离散和连续概率函数。概率（质量或密度）函数的概念可以推广到多个随机变量。例如，如果 X 和 Y 是随机变量，则 $p(X, Y)$表示**联合**（joint）概率函数。随机变量 X 和 Y 是相互独立的，如果 $P(X, Y) = P(X)\times P(Y)$。如果两个随机变量是独立的，则意味一个变量的值对另一个的值没有影响。

表 C-1　概率函数的例子 ($\Gamma(n+1) = n\Gamma(n)$ 并且 $\Gamma(1) = 1$)

	概率函数	参数
高斯	$P(x)=\frac{1}{\sqrt{2\pi\sigma}}\exp^{\frac{1(x-\mu)^2}{2\ \sigma^2}}$	μ, σ
伯努利	$P(x)=\binom{n}{x}p^x(1-p)^{n-x}$	n, p
泊松	$p(x)=\frac{1}{x!}\theta^x\exp^{-\theta}$	θ
指数	$p(x)=\theta\exp^{-\theta x}$	θ
Γ	$p(x)=\frac{\lambda^\alpha}{\Gamma(\alpha)}x^{\alpha-1}\exp^{-\lambda x}$	λ, α
卡方	$p(x)=\frac{1}{2^{k/2}\Gamma(k/2)}x^{k/2-1}\exp^{-x/2}$	k

对于理解随机变量之间的依赖性，**条件概率**（conditional probability）是另一个有用的概念。给定 X，变量 Y 的条件概率记作 $P(Y|X)$，定义为

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)} \tag{C-3}$$

如果 X 和 Y 是独立的，则 $P(Y|X) = P(Y)$。使用称作 **Bayes 定理**的公式，条件概率 $P(Y|X)$和 $P(X|Y)$都可以用另一个表示。Bayes 定理由下式给出：

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \tag{C-4}$$

如果$\{X_1, X_2, \cdots, X_k\}$是随机变量 X 的所有可能的结果集，则上式的分母可以用下式表示：

$$P(X) = \sum_{i=1}^{k} P(X, Y_i) = \sum_{i=1}^{k} P(X \mid Y_i)P(Y_i) \tag{C-5}$$

公式（C-5）称作**全概率律**（law of total probability）。

期望值

随机变量 X 的函数 g 的**期望值**（expected value）记作 $E[g(X)]$，是 $g(X)$的加权平均值，其中权重由 X 的概率函数给出。如果 X 是离散随机变量，则它的期望值可以用下式计算：

$$E[g(X)] = \sum_i g(x_i)P(X = x_i) \tag{C-6}$$

另一方面，如果 X 是连续随机变量，则

$$E[g(X)] = \int_{-\infty}^{\infty} g(X) f(X) \mathrm{d}X \tag{C-7}$$

其中 $f(X)$是 X 的概率密度函数。本节的其余部分只考虑离散随机变量的期望值，对应连续随机变量的期望值可以通过用积分取代求和得到。

在概率论中，有一些特别有用的期望值。首先，如果 $g(X) = X$，则

$$\mu_X = E[X] = \sum_i x_i P(X = x_i) \tag{C-8}$$

这个期望值对应于随机变量 X 的**均值**（mean）。另一个有用的期望值是 $g(X) = (X - \mu_X)$时的期望值。这个函数的期望值是

$$\sigma_X^2 = E[(X - \mu_X)^2] = \sum_i (x_i - \mu_X)^2 P(X = x_i) \tag{C-9}$$

这个期望值对应于随机变量 X 的**方差**（variance）。方差的平方根对应于随机变量 X 的**标准差**（standard deviation）。

例 C.2 考虑例 C.1 中的随机实验。掷 4 次均匀的硬币，期望看到背面朝上的平均次数为

$$\mu_X = 0\times 1/16 + 1 \times 4/16 + 2 \times 6/16 + 3 \times 4/16 + 4 \times 1/16 = 2 \tag{C-10}$$

期望看到背面朝上次数的方差为

$$\sigma_X^2 = (0-2)^2 \times 1/16 + (1-2)^2 \times 4/16 + (2-2)^2 \times 6/16$$
$$+(3-2)^2 \times 4/16 + (4-2)^2 \times 1/16 = 1$$ □

对于一对随机变量，要计算的一个有用的期望值是**协方差**（covariance）函数 Cov，它定义如下：

$$Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] \tag{C-11}$$

注意，随机变量 X 的方差等于 $Cov(X, X)$。函数的期望值还具有如下性质。

(1) 如果 a 是常量，则 $E[a] = a$。

(2) $E[aX] = aE[X]$。

(3) $E[aX + bY] = aE[X] + bE[Y]$。

根据这些性质，公式（C-9）和（C-11）可以改写成如下形式：

$$\sigma_X^2 = E[(X - \mu_X)^2] = E[X^2] - E[X]^2 \tag{C-12}$$

$$Cov(X, Y) = E[XY] - E[X]E[Y] \tag{C-13}$$

C.2 统计学

为了提取关于总体的结论，从整个总体收集数据通常是不可行的。相反，我们必须基于从样本数据收集的证据得到合理的结论。这种基于样本数据提取关于总体的可靠结论的过程称作**统计推理**（statistical inference）。

C.2.1 点估计

术语**统计量**（statistic）是指从样本数据推导出的数值量。两个有用的统计量是样本均值（$\bar{x}$）和样本方差（s_X^2）：

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} X_i \tag{C-14}$$

$$s_X^2 = \frac{1}{N-1}\sum_{i=1}^{1}(X_i - \bar{x})^2 \tag{C-15}$$

使用样本统计量估计总体参数的过程称为**点估计**（point estimation）。

例 C.3　设 $X_1, X_2, \cdots, X_N$ 是从均值为 μ_X、方差为 σ_X^2 的总体抽取的 N 个独立同分布观测的随机样本。令 $\bar{x}$ 为样本均值，则

$$E[\bar{X}] = E\left[\frac{1}{N}\sum_i X_i\right] = \frac{1}{N}\sum_i E[X_i] = \frac{1}{N}\times N\mu_X = \mu_X \tag{C-16}$$

其中 $E[X_i] = \mu_X$，因为所有的观测都来自均值为 μ_X 的相同分布。这一结果表明样本的均值 $\bar{x}$ 逼近总体均值 μ_X，特别是当 N 充分大时。用统计学的术语来说，样本均值称作总体均值的**无偏**（unbiased）估计。可以证明样本均值的方差为

$$E\left[(\bar{x} - E[\bar{x}])^2\right] = \sigma_X^2 / N \tag{C-17}$$

由于总体的方差通常是未知的，因此通常用样本方差 s_X^2 替换 σ_X^2 来近似样本均值的方差。量 $s_X / \sqrt{N}$ 称为均值的**标准误差**（standard error）。 □

C.2.2　中心极限定理

正态分布可能是最常用的概率分布，因为很多随机现象都可以用这种分布建模。这是称作**中心极限定理**（central limit theorem）的统计学原理的推论。

定理 C.1　中心极限定理　*考虑从均值为 μ_X、方差为 σ_X^2 的概率分布抽取的、大小为 N 的随机样本。如果 $\bar{x}$ 是样本均值，则当样本的规模增大时，$\bar{x}$ 的分布逼近均值为 μ_X、方差为 σ_X^2/N 的正态分布。*

无论随机变量从何种分布提取，中心极限定理都成立。例如，假设我们从具有某个未知分布的数据集随机地抽取 N 个独立实例。令 X_i 是一个随机变量，它指示第 i 个实例是否被给定的分类器正确预测，即如果该实例被正确分类则 $X_i = 1$，否则 $X_i = 0$。样本均值 $\bar{X}$ 表示分类器的期望准确率。中心极限定理表明，尽管抽取实例的分布可能不是正态的，但是期望准确率（即样本均值）往往是正态分布。

C.2.3　区间估计

在估计总体的参数时，指出估计的可靠性是有用的。例如，假设我们对由随机抽取的观测估计总体均值 μ_X 感兴趣。使用诸如样本均值 $\bar{x}$ 这样的点估计可能并不充分，特别是当样本的规模比较小时尤其如此。作为替代，给出一个以高概率包含总体均值的区间可能是有用的。这种估计可能找到总体参数的区间的估计任务称为**区间估计**（interval estimation）。令 θ 是需要估计的总体参数。如果

$$P(\theta_1 < \theta < \theta_2) = 1 - \alpha \tag{C-18}$$

则(θ_1, θ_2)是 θ 在**置信水平**（confidence level）$1-\alpha$上的**置信区间**（confidence interval）。图 C-1 显示了由均值为 0、方差为 1 的正态分布导出的参数的 95%置信区间。正态分布下的阴影区域的面积为 0.95。换言之，如果我们从该分布产生一个样本，则被估计的参数落在−2 和+2 之间的可能性为 95%。

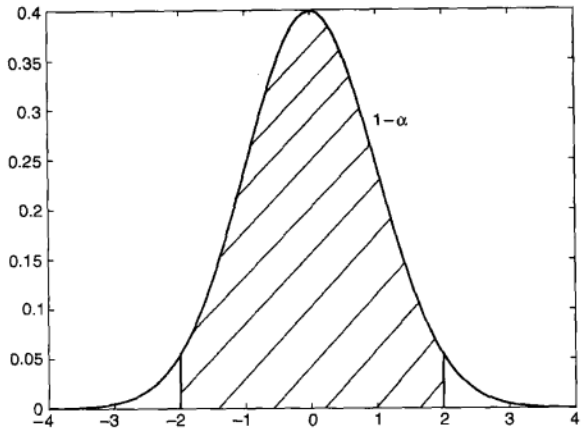


图 C-1 参数的置信区间

考虑一个随机抽取的观测序列 $X_1, X_2, \cdots, X_N$。我们想根据样本均值 $\bar{x}$，在 68%置信区间估计总体均值 μ_X。根据中心极限定理，当 N 充分大时，$\bar{x}$ 逼近均值为 μ_X、方差为 σ_x^2/N 的正态分布。可以用如下方法把这种分布变换为标准正态分布（即均值为 0、方差为 1 的正态分布）：

$$Z = \frac{\bar{x} - \mu_X}{\sigma_X / \sqrt{N}} \approx \frac{\bar{x} - \mu}{s_X / \sqrt{N}} \approx \aleph(0,1) \tag{C-19}$$

其中总体的标准差用样本均值的标准误差近似。查标准正态分布的概率表得到 $P(-1 < Z < 1) = 0.68$。该概率可以改写成

$$P(-s_X / \sqrt{N} < \bar{x} - \mu_X < s_X / \sqrt{N}) = 0.68$$

或等价地改成

$$P(\bar{x} - s_X / \sqrt{N} < \mu_X < \bar{x} + s_X / \sqrt{N}) = 0.68$$

因此，μ_X 的 68%置信区间为 $\bar{x} \pm s_X / \sqrt{N}$。

C.3 假设检验

假设检验是一种统计推理过程，它基于从数据中收集的证据，确定一个猜想或假设是应该被接受，还是被拒绝。假设检验的例子包括检验被数据挖掘算法提取的模式的质量、确定两个分类模型之间性能差别的显著性。

在假设检验，通常我们有两个矛盾的假设，分别称作**原假设**（null hypothesis）和**备择假设**（alternative hypothesis）。假设检验的一般过程包括如下 4 个步骤。

(1) 形式化需要检验的原假设和备择假设。

(2) 定义一个确定原假设是应该被接受还是被拒绝的检验统计量 θ。与该检验统计量相关联的分布应该是已知的。

(3) 由观测数据计算 θ 的值。使用概率分布知识确定称作 p 值的量值。

(4) 定义**显著水平** α，它控制原假设应该被拒绝的 θ 值域。θ 这个值域称为**拒绝域**（rejection region）。

考虑用第6章提供的算法得到的关联模式 X。假设我们对从统计学的角度评估该模式的质量感兴趣。判定一个关联模式是否有意义的标准依赖于一个称作模式支持度的量 $s(X)$（见公式(6-2)）。支持度测量实际观测到该模式的记录所占的比例。如果 $s(X) > minsup$，则 X 被认为是有意义的，其中 $minsup$ 是用户指定的最小阈值。

该问题可以用如下方法纳入假设检验框架。为了检验模式 X，我们需要确定是接受原假设 $H_0: s(X) = minsup$，还是接受备择假设 $H_1: s(X) > minsup$。如果原假设被拒绝，则 X 被认为是一个有意义的模式。为了进行检验，应该知道 $s(X)$的概率分布。我们可以使用二项分布对这个问题建模，因为确定模式 X 在 N 个记录中出现的次数类似于确定掷 N 次硬币面朝上的次数。前者可以用一个均值为 $s(X)$、方差为 $s(X) \times (1 - s(X))/N$ 的二项分布描述。如果 N 充分大（在大多数购物篮分析问题中通常如此），则该二项分布可以进一步用正态分布近似。

在原假设下，假设 $s(X)$是均值为 $minsup$、方差为 $minsup \times (1 - minsup)/N$ 的正态分布，为了检验是应该接受还是拒绝原假设可以使用如下 Z 统计量：

$$Z = \frac{s(X) - minsup}{\sqrt{minsup \times (1 - minsup) / N}} \tag{C-20}$$

Z 是均值为 0、方差为 1 的标准正态分布。该统计量本质上是以标准差为单位测量观测的 $s(X)$支持度与阈值 $minsup$ 的差。令 $N = 10000$，$s(X) = 11\%$，$minsup = 10\%$。在原假设下，$Z = (0.11 - 0.1) / \sqrt{0.09/10000} = 3.33$。查找标准正态分布概率表，$Z = 3.33$ 的单侧检验对应于 p 值 4.34×10^4。

假设 $\alpha = 0.001$ 是期望的显著水平。α控制尽管原假设为真但不正确地拒绝它（在统计学文献中这称为**第一类型**错误）的概率。例如，0.01 的α值暗示被发现的模式可疑的可能性为 1%。在每个显著水平，存在一个对应的阈值 Z_α，使得模式的 Z 值超过该阈值时，我们认为该模式是显著的。阈值 Z_α可以在标准正态分布的概率表中查找。例如，选取 $\alpha = 0.001$ 将设定 $Z_\alpha = 3.09$ 的拒绝域。由于 $p < \alpha$，即等价地 $Z > Z_\alpha$，因此原假设被拒绝，而模式被认为是有意义的。

附录 D

回　　归

回归是一种预测建模技术，其中被估计的目标变量是连续的。回归应用的例子包括使用其他经济学指标预测股市指数，基于高空气特征流预测一个地区的降水量，根据广告开销预测公司的总销售，按照有机物质中的碳 14 残留估计化石的年龄。

D.1　预备知识

令 D 指包含 N 个观测的数据集 $D = \{(\mathbf{x}_i, y_i) | i = 1, 2, \cdots, N\}$。每个 $\mathbf{x}_i$ 对应于第 i 个观测的属性集（又称**说明变量**（explanatory variable）），而 y_i 对应于**目标变量**（target variable）（或因变量）。回归任务的说明属性可以是离散的或连续的。

定义 D.1　回归　回归（regression）是一个任务，它学习一个把每个属性集 $\mathbf{x}$ 映射到一个连续值输出 y 的**目标函数**（target function）f。

回归的目标是找到一个可以以最小误差拟合输入数据的目标函数。回归任务的**误差函数**（error function）可以用绝对误差或平方误差和表示：

$$\text{绝对误差} = \sum_i | y_i - f(\mathbf{x}_i) | \tag{D-1}$$

$$\text{平方误差} = \sum_i \left(y_i - f(\mathbf{x}_i)\right)^2 \tag{D-2}$$

D.2　简单线性回归

考虑图 D-1 所示的生理学数据。该数据对应于热通量和一个人睡眠时皮肤温度的测量。假设我们希望根据热传感器收集的热通量测量值预测一个人的皮肤温度。二维散布图表明这两个变量之间存在很强的线性关系。

D.2.1　最小二乘方法

假设我们希望用下面的线性模型拟合观测数据：

$$f(x) = \omega_1 x + \omega_0 \tag{D-3}$$

其中 ω_1 和 ω_0 是该模型的参数，称作**回归系数**（regression coefficient）。做这件事的标准方法是使用**最小二乘方法**（method of least squares）。该方法试图找出参数 (ω_0, ω_1)，它们最小化误差的平方和为

$$SSE = \sum_{i=1}^{N} \left[y_i - f(x_i) \right]^2 = \sum_{i=1}^{N} \left[y_i - \omega_1 x - \omega_0 \right]^2 \tag{D-4}$$

它又称**残差平方和**（residual sum of square）。

热通量	皮肤温度
10.858	31.002
10.617	31.021
10.183	31.058
9.7003	31.095
9.652	31.133
10.086	31.188
9.459	31.226
8.3972	31.263
7.6251	31.319
7.1907	31.356
7.046	31.412
6.9494	31.468
6.7081	31.524

热通量	皮肤温度
6.3221	31.581
6.0325	31.618
5.7429	31.674
5.5016	31.712
5.2603	31.768
5.1638	31.825
5.0673	31.862
4.9708	31.919
4.8743	31.975
4.7777	32.013
4.7295	32.07
4.633	32.126
4.4882	32.164

热通量	皮肤温度
4.3917	32.221
4.2951	32.259
4.2469	32.296
4.0056	32.334
3.716	32.391
3.523	32.448
3.4265	32.505
3.3782	32.543
3.4265	32.6
3.3782	32.657
3.3299	32.696
3.3299	32.753
3.4265	32.791

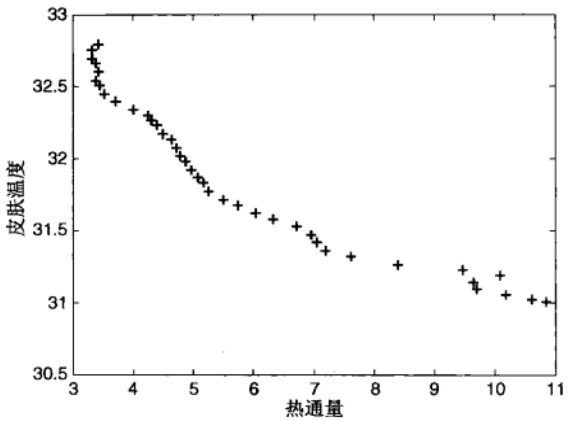


图 D-1　热通量测量值和人的皮肤温度

这个优化问题可以通过如下方法求解：求 E 关于 ω_0 和 ω_1 的偏导数，令它们等于零，并解对应的线性方程组：

$$\begin{aligned}\frac{\partial E}{\partial \omega_0} &= -2\sum_{i=1}^{N}[y_i - \omega_1 x_i - \omega_0] = 0 \\ \frac{\partial E}{\partial \omega_1} &= -2\sum_{i=1}^{N}[y_i - \omega_1 x_i - \omega_0]x_i = 0\end{aligned} \tag{D-5}$$

这些方程可以简写成如下矩阵方程，又称**正规方程**（normal equation）：

$$\begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix}\begin{pmatrix}\omega_0 \\ \omega_1\end{pmatrix} = \begin{pmatrix}\sum_i y_i \\ \sum_i x_i y_i\end{pmatrix} \tag{D-6}$$

由于$\sum_i x_i = 229.9$，$\sum_i x_i^2 = 1569.2$，$\sum_i y_i = 1242.9$，$\sum_i x_i y_i = 7279.7$，因此可以求解该正态方程，得到参数的如下估计：

$$\begin{aligned}\begin{pmatrix}\hat{\omega}_0 \\ \hat{\omega}_1\end{pmatrix} &= \begin{pmatrix} 39 & 229.9 \\ 229.9 & 1569.2 \end{pmatrix}^{-1}\begin{pmatrix}1242.9 \\ 7279.7\end{pmatrix} \\ &= \begin{pmatrix} 0.1881 & -0.0276 \\ -0.0276 & 0.0047 \end{pmatrix}\begin{pmatrix}1242.9 \\ 7279.7\end{pmatrix} \\ &= \begin{pmatrix}33.1699 \\ -0.2208\end{pmatrix}\end{aligned}$$

这样，最小化 SSE、最佳拟合数据的线性模型为

$$f(x) = 33.17 - 0.22x$$

图 D-2 显示了对应于该模型的直线。

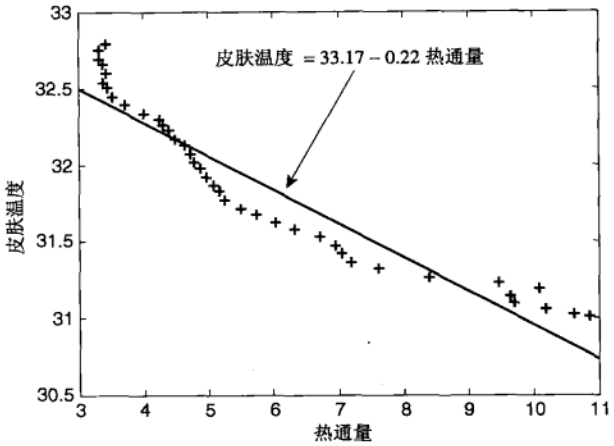


图 D-2 拟合图 D-1 中数据的线性模型

我们可以证明公式（D-6）的正规方程的一般解可以表示为

$$\hat{\omega}_0 = \bar{y} - \hat{\omega}_1 \bar{x}$$

$$\hat{\omega}_1 = \frac{\sigma_{xy}}{\sigma_{xx}} \tag{D-7}$$

其中，$\bar{x} = \sum_i x_i/N$，$\bar{y} = \sum_i y_i/N$，而

$$\sigma_{xy} = \sum_i (x_i - \bar{x})(y_i - \bar{y}) \tag{D-8}$$

$$\sigma_{xx} = \sum_i (x_i - \bar{x})^2 \tag{D-9}$$

$$\sigma_{yy} = \sum_i (y_i - \bar{y})^2 \tag{D-10}$$

这样，产生最小平方误差的线性模型由下式给出：

$$f(x) = \bar{y} + \frac{\sigma_{xy}}{\sigma_{xx}}[x - \bar{x}] \tag{D-11}$$

概括地说，最小二乘方法是一种系统方法，它通过最小化 y 的实际值与估计值之间的平方误差，用一个线性模型拟合因变量 y。尽管该模型相对简单，但是它看上去给出了相当精确的近似，因为线性模型是具有连续导数的任意函数的一阶泰勒级数近似。

D.2.2 分析回归误差

某些数据可能包含 $\mathbf{x}$ 和 y 的测量误差。此外，可能存在一些混杂因素影响因变量 y，但未包含在模型中。正因为如此，回归任务中的因变量 y 可能是非确定的；也就是说，即使提供相同属性集 $\mathbf{x}$，它也可能产生不同的值。

我们可以使用概率方法对这类情况建模，其中 y 被看作一个随机变量：

$$\begin{aligned} y &= f(\mathbf{x}) + [y - f(\mathbf{x})] \\ &= f(\mathbf{x}) + \varepsilon \end{aligned} \tag{D-12}$$

测量误差和模型误差都被一个随机噪声项ε所吸收。通常假定数据中的随机噪声出现是独立的，并且服从某种概率分布。

例如，如果随机噪声来自一个均值为0、方差为σ^2的正态分布，则

$$P(\varepsilon \mid \mathbf{x}, \Omega) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\left[-\frac{[y - f(\mathbf{x}, \Omega)]^2}{2\sigma^2}\right]} \tag{D-13}$$

$$\log[P(\varepsilon \mid \mathbf{x}, \Omega)] = -\frac{1}{2\sigma^2}(y - f(\mathbf{x}, \Omega))^2 \tag{D-14}$$

这一分析表明最小化*SSE*（$[y-f(\mathbf{x}, \Omega)]^2$）蕴含地假定随机噪声来自一个正态分布。此外，可以证明最能最小化这类误差的常数模型$f(\mathbf{x}, \Omega) = c$是均值，即$c = \bar{y}$。

另一种典型的噪声概率模型使用拉普拉斯分布：

$$P(\varepsilon \mid \mathbf{x}, \Omega) = c\,\mathrm{e}^{-c|y - f(\mathbf{x}, \Omega)|} \tag{D-15}$$

$$\log[P(\varepsilon|\mathbf{x}, \Omega)] = -c|y - f(\mathbf{x}, \Omega)| + \text{常量} \tag{D-16}$$

这表明最小化绝对误差$|y - f(\mathbf{x}, \Omega)|$蕴含地假定随机噪声服从拉普拉斯分布。这种情况下的最佳常量模型对应于$f(\mathbf{x}, \Omega) = \tilde{y}$，$y$的中位数。

除了公式（D-4）定义的*SSE*之外，我们还可以定义另外两种误差：

$$SST = \sum_i (y_i - \bar{y})^2 \tag{D-17}$$

$$SSM = \sum_i (f(x_i) - \bar{y})^2 \tag{D-18}$$

其中*SST*称为总平方和，而*SSM*称为回归平方和。在使用平均值$\bar{y}$估计因变量时，*SST*表示预测误差，而*SSM*代表回归模型的误差量。*SST*、*SSE*和*SSM*之间的关系推导如下：

$$\begin{aligned} SSE &= \sum_i [y_i - \bar{y} + \bar{y} - f(x_i)]^2 \\ &= \sum_i [y_i - \bar{y}]^2 + \sum_i [f(x_i) - \bar{y}]^2 + 2\sum_i (y_i - \bar{y})(\bar{y} - f(x_i)) \\ &= \sum_i [y_i - \bar{y}]^2 + \sum_i [f(x_i) - \bar{y}]^2 - 2\sum_i (y_i - \bar{y})\omega_1(x_i - \bar{x}) \\ &= \sum_i [y_i - \bar{y}]^2 + \sum_i [f(x_i) - \bar{y}]^2 - 2\sum_i \omega_1^2 (x_i - \bar{x})^2 \\ &= \sum_i [y_i - \bar{y}]^2 - \sum_i [f(x_i) - \bar{y}]^2 \\ &= SST - SSM \end{aligned} \tag{D-19}$$

其中，我们使用了如下关系：

$$\bar{y} - f(x_i) = -\omega_1(x_i - \bar{x})$$

$$\sum_i [y_i - \bar{y}][x_i - \bar{x}] = \sigma_{xy} = \omega_1 \sigma_{xx} = \omega_1 \sum_i [x_i - \bar{x}]^2$$

这样，我们有$SST = SSE + SSM$。

D.2.3　分析拟合的满意度

一种测量拟合满意度的方法是计算如下度量：

$$R^2 = \frac{SSM}{SST} = \frac{\sum_i [f(x_i) - \bar{y}]^2}{\sum_i [y_i - \bar{y}]^2} \tag{D-20}$$

回归模型的 R^2（即判决系数）可能在 0 和 1 之间取值。如果因变量中观察到的大部分变异性都能被回归模型解释，则它的值接近 1。

R^2 也与协相关系数 r 有关。r 度量自变量与因变量之间的线性关系强度

$$r = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}\sigma_{xy}}} \tag{D-21}$$

由公式（D-9）、（D-10）和（D-11）我们有

$$\begin{aligned} R^2 &= \frac{\sum_i [f(x_i) - \bar{y}]^2}{\sum_i [y_i - \bar{y}]^2} \\ &= \frac{\sum_i \left[\frac{\sigma_{xy}}{\sigma_{xx}}(x_i - \bar{x})\right]^2}{\sigma_{yy}} \\ &= \frac{\sigma_{xy}^2}{\sigma_{xx}^2 \sigma_{yy}} \sum_i (x_i - \bar{x})^2 \\ &= \frac{\sigma_{xy}^2}{\sigma_{xx}^2 \sigma_{yy}} \sigma_{xx} \\ &= \frac{\sigma_{xy}^2}{\sigma_{xx}\sigma_{yy}} \end{aligned} \tag{D-22}$$

上面的分析表明协相关系数等于判决系数的平方根（不考虑符号。符号与关系的方向有关，或者为正或者为负）。

值得注意的是，R^2 随着更多自变量添加到模型中而增大。一种校正添加到模型中的自变量数的方法是使用如下调整后的 R^2 度量：

$$\text{调整后的 } R^2 = 1 - \left(\frac{N-1}{N-d}\right)(1 - R^2) \tag{D-23}$$

其中 N 是数据点数，而 $d+1$ 是回归模型的参数个数。

D.3 多元线性回归

使用下面的矩阵表示法，正规方程可以写成更紧凑的形式。令 $\mathbf{X} = (\mathbf{1}\ \mathbf{x})$，其中 $\mathbf{1} = (1, 1, 1, \cdots)^T$，而 $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T$。于是，我们可以证明：

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T\mathbf{x} \\ \mathbf{x}^T\mathbf{1} & \mathbf{x}^T\mathbf{x} \end{pmatrix} = \begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \tag{D-24}$$

这等于正规方程左边的矩阵。类似地，如果 $\mathbf{y} = (y_1, y_2, \cdots, y_N)^T$，我们可以证明

$$(\mathbf{1}\ \mathbf{x})^T\mathbf{y} = \begin{pmatrix} \mathbf{1}^T\mathbf{y} \\ \mathbf{x}^T\mathbf{y} \end{pmatrix} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix} \tag{D-25}$$

这等于正规方程右边的矩阵。把公式（D-24）和（D-25）代入公式（D-6），我们得到如下方程：

$$\mathbf{X}^T\mathbf{X}\Omega = \mathbf{X}^T\mathbf{y} \tag{D-26}$$

其中 $\Omega = (\omega_0, \omega_1)^T$。我们可以用下式求解 Ω 中的参数：

$$\Omega = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{D-27}$$

上面的表示法是有用的，因为它可以让我们把线性回归方法推广到多元情况。更明确地说，如果属性集包含 d 个说明属性$(x_1, x_2, \cdots, x_d)$，则 $\mathbf{X}$ 变成 $N \times d$ **设计矩阵**（design matrix）：

$$\mathbf{X}=\begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{pmatrix} \tag{D-28}$$

而 $\Omega = (\omega_0, \omega_1, \cdots, \omega_{d-1})^T$ 是 d 维向量。可以通过解公式（D-26）中的矩阵方程来计算参数。

D.4 可选的最小二乘回归方法

最小二乘方法也可以用来找其他类型的最小化 SSE 的回归模型。更具体地说，如果模型是

$$y = f(\mathbf{x}, \Omega) + \varepsilon \tag{D-29}$$

$$= \omega_0 + \sum_i \omega_i g_i(\mathbf{x}) + \varepsilon \tag{D-30}$$

并且随机噪声是正态分布的，则我们可以使用与前面确定参数向量 Ω 相同的技术。g_i 可以是任何类型的基本函数，包括多项式、核和其他非线性函数。

例如，假设 $\mathbf{x}$ 是二维特征向量，回归模型是二次多项式函数

$$f(x_1, x_2, \Omega) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_1 x_2 + \omega_4 {x_1}^2 + \omega_5 {x_2}^2 \tag{D-31}$$

如果我们创建如下设计矩阵：

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{11}x_{12} & x_{11}^2 & x_{22}^2 \\ 1 & x_{21} & x_{22} & x_{21}x_{22} & x_{21}^2 & xx_{22}^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{N1} & x_{N2} & x_{N1}x_{N2} & x_{N1}^2 & x_{N2}^2 \end{pmatrix} \tag{D-32}$$

其中 x_{ij} 是第 i 个观测的第 j 个属性，则该回归问题变成等价于解公式（D-26）中的方程。参数向量 Ω 的最小二乘解由公式（D-27）给出。通过选择适当的设计矩阵，我们可以把这种方法推广到任意基本函数。

附录E

优　　化

优化是找出函数的最大值或最小值的方法。优化是数据挖掘的重要课题，因为许多数据挖掘任务都可以设计成优化问题，例如，8.2.1 节介绍的K均值聚类算法寻找最小化误差的平方和（SSE）的簇集合。类似地，D.2.1 节介绍的最小二乘方法旨在学习最小化模型SSE的回归系数。本附录简略回顾用于求解优化问题的各种技术。

E.1　无约束的优化

假设 $f(x)$是一元函数，具有连续一阶导数和二阶导数。在无约束的优化问题中，任务是找出最小化或最大化 $f(x)$的解 x^*，而不对 x^*施加任何约束。解 x^*称作**平稳点**（stationary point），可以通过取 f 的一阶导数，并令它等于零找到：

$$\left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x^*}=0$$

$f(x^*)$可以取极大或极小值，取决于该函数的二阶导数。

- 如果在 $x=x^*$有 $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}<0$，则 x^*是极大平稳点。
- 如果在 $x=x^*$有 $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}>0$，则 x^*是极小平稳点。
- 当在 $x=x^*$上 $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}=0$ 时，x^*是拐点。

图 E-1 图示了一个例子，函数包含三个平稳点（极大、极小和拐点）。

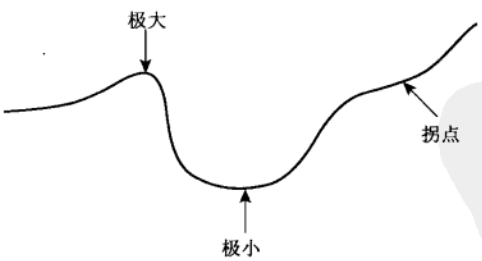


图 E-1　函数的平稳点

该定义可以推广到多元函数 $f(x_1, x_2, \cdots, x_d)$，这里找平稳点 $\mathbf{x}^*=[x_1^*, x_2^*, \cdots, x_d^*]^T$ 的条件为

$$\left.\frac{\partial f}{\partial x_i}\right|_{x_i=x_i^*}=0, \forall i=1,2,\cdots,d \qquad \text{(E-1)}$$

然而，不像一元函数，确定 $\mathbf{x}^*$是极大还是极小平稳点更困难。困难的原因在于我们需要对所有可能的一对 i 和 j，考虑偏导数 $\frac{\partial^2 f}{\mathrm{d}x_i \mathrm{d}x_j}$。二阶偏导数的完全集由黑森矩阵（Hessian matrix）给出：

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d} \\ \cdots & \cdots & & \cdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1} & \frac{\partial^2 f}{\partial x_d \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_d \partial x_d} \end{bmatrix} \tag{E-2}$$

- **黑森矩阵 H 是正定的**，当且仅当对于任意非零向量 $\mathbf{x}$，$\mathbf{x}^T\mathbf{H}\mathbf{x} > 0$。如果 $\mathbf{H}(\mathbf{x}^*)$是正定的，则 $\mathbf{x}^*$是极小平稳点。
- **黑森矩阵 H 是负定的**，当且仅当对于任意非零向量 $\mathbf{x}$，$\mathbf{x}^T\mathbf{H}\mathbf{x} < 0$。如果 $\mathbf{H}(\mathbf{x}^*)$是负定的，则 $\mathbf{x}^*$是极大平稳点。
- **黑森矩阵 H 是不定的**，如果 $\mathbf{x}^T\mathbf{H}\mathbf{x}$ 对于某些 $\mathbf{x}$ 值为正，而对于其它 $\mathbf{x}$ 值为负。具有不定黑森矩阵的平稳点是**鞍点**（saddle point），它在一个方向上具有极小值，在另一个方向上具有极大值。

例 E.1　设 $f(x, y) = 3x^2+2y^3-2xy$。图 E-2 显示了该函数的图像。找该函数平稳点的的条件为

$$\begin{aligned} \frac{\partial f}{\partial x} &= 6x - 2y = 0 \\ \frac{\partial f}{\partial y} &= 6y^2 - 2x = 0 \end{aligned} \tag{E-3}$$

它的解为 $x^* = y^* = 0$ 或 $x^* = 1/27, y^* = 1/9$。

f 的黑森矩阵为

$$\mathbf{H}(x, y) = \begin{bmatrix} 6 & -2 \\ -2 & 12y \end{bmatrix}$$

在 $x = y = 0$，

$$\mathbf{H}(0,0) = \begin{bmatrix} 6 & -2 \\ -2 & 0 \end{bmatrix}$$

由于$[x\ y]\ \mathbf{H}(0, 0)\ [x\ y]^T = 6x^2 - 4xy = 2x(3x - 2y)$，它可以是正的或负的，因此黑森矩阵是不定的，并且(0, 0)是一个鞍点。

在 $x = 1/27, y = 1/9$，

$$\mathbf{H}(1/27,1/9) = \begin{bmatrix} 6 & -2 \\ -2 & 12/9 \end{bmatrix}$$

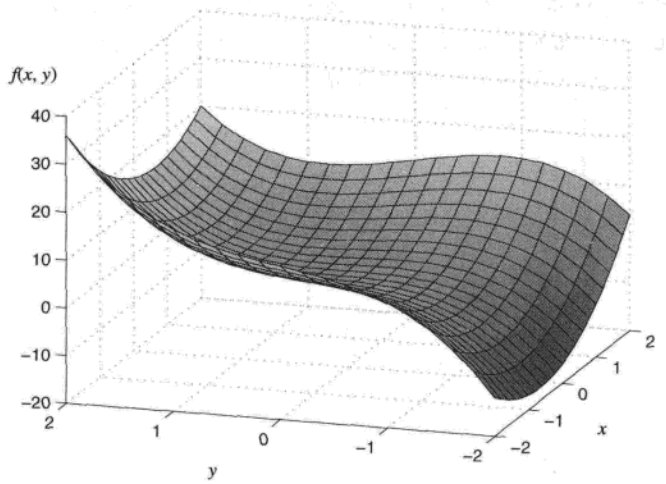


图 E-2 函数 $f(x, y) = 3x^2 + 2y^3 - 2xy$ 的图像

由于对于非零 x 和 y，$[x\ y]\ \mathbf{H}(1/27, 1/9)\ [x\ y]^T = 4x^2 - 2xy + 4y^2/3 = 4(x - y/4)^2 + 13y^2/4 > 0$，因此黑森矩阵是正定的。这样，(1/27, 1/9)是极小平稳点。f 的极小值为–0.0014。 □

数值方法

如果公式（E-1）可以对 $\mathbf{x}^*$解析地求解，则可以使用上面的方法。在许多情况下，找解析解是一个很困难的问题，这就迫使我们使用数值方法找近似解。找函数极小值的一些数值方法包括黄金搜索、牛顿法和梯度下降搜索。尽管这里提供的技术用于极小化目标函数 $f(\mathbf{x})$，但是它们也可以用于极大化问题，因为容易通过把函数 $f(\mathbf{x})$转换成$-f(\mathbf{x})$，把极大化问题转换成极小化问题。

黄金搜索 考虑图 E-3 所示的单峰分布，其极小值在区间 a 和 b 之间。黄金搜索方法迭代地找相继较小的、包含极小值的区间，直到区间的宽度足够小，可以近似平稳点。为了确定较小的区间，选择两个点 c 和 d，使得区间(a, c, d)和(c, d, b)具有相等的宽度。令 $c - a = b - d = \alpha \times (b - a)$，$d - c = \beta \times (b - a)$。因此

$$1 = \frac{(b-d)+(d-c)+(c-a)}{b-a} = a + \beta + \alpha$$

或等价地

$$\beta = 1 - 2\alpha \tag{E-4}$$

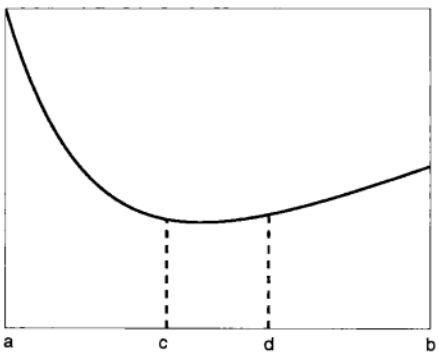
还要选择宽度，满足以下条件，使得我们可以使用递归过程：

$$\frac{d-c}{b-c} = \frac{c-a}{b-a}$$

或等价地

$$\frac{\beta}{1-\alpha} = \alpha \tag{E-5}$$

公式（E-4）和公式（E-5）中的方程可以一起求解，得到$\alpha = 0.382$，$\beta = 0.236$。通过比较$f(c)$和$f(d)$，可以确定极小值在区间(a, c, d)，还是在区间(c, d, b)。然后递归地划分包含最小值的区间，直到区间宽度足够小，可以近似极小值，如算法E.1所示。



图E-3　一个单峰函数

算法E.1　黄金搜索算法

```
1:  c = a + 0.382(b − a)
2:  while b − a > ε do
3:      d = b − 0.382(b − a)
4:      if f(d) > f(c) then
5:          b = d
6:      else
7:          a = c, c = d
8:      end if
9:  end while
10: return c
```

除了假定函数在初始区间$[a, b]$中连续并且是单峰的之外，黄金搜索方法不对函数做其他假定。它线性收敛于极小值解。

牛顿法　牛顿法基于使用函数$f(x)$的二次近似。通过使用f在x_0的泰勒级数展开式，得到如下表达式：

$$f(x) \approx f(x_0) + (x - x_0) f'(x_0) + \frac{(x - x_0)^2}{2} f''(x_0) \tag{E-6}$$

取该函数关于x的导数，并令它等于零，得到如下等式：

$$f'(x) = f'(x_0) + (x - x_0) f''(x_0) = 0$$
$$x = x_0 - \frac{f'(x_0)}{f''(x_0)} \tag{E-7}$$

可以使用公式（E-7）更新x，直到它收敛于极小值。可以证明牛顿法是二次收敛的，尽管它可能不收敛，特别是当初始点x_0远离极小值时。该方法的概要在算法E.2中给出。

算法 E.2 牛顿法

1：令 x_0 为初始点。
2：**while** $|f'(x_0)| > \varepsilon$ **do**
3：$x = x_0 - \dfrac{f'(x_0)}{f''(x_0)}$
4：$x_0 = x$
5：**end while**
6：**return** x

用梯度算子 $\nabla f(\mathbf{x})$ 替换一阶导数 $f'(x)$，用黑森矩阵 $\mathbf{H}$ 替换二阶导数 $f''(x)$，可以把牛顿法推广到多元数据：

$$\mathbf{x} = \mathbf{x} - \mathbf{H}^{1}\nabla f(\mathbf{x})$$

然而，更容易的办法不是计算黑森矩阵的逆，而是解如下方程：

$$\mathbf{Hz} = -\nabla f(\mathbf{x})$$

来得到向量 $\mathbf{z}$。找平稳点的迭代公式修改为 $\mathbf{x} = \mathbf{x} + \mathbf{z}$。

梯度下降法 牛顿法是使用如下更新方程渐近地寻找函数的平稳点的多种增量方法之一：

$$\mathbf{x} = \mathbf{x} + \lambda g(\mathbf{x})) \tag{E-8}$$

函数 $g(\mathbf{x})$ 确定搜索方向，而 λ 确定步长。

梯度下降法假定函数 $f(\mathbf{x})$ 是可微的，并按下式计算平稳点：

$$\mathbf{x} = \mathbf{x} - \lambda\nabla f(\mathbf{x}) \tag{E-9}$$

在这种方法中，$\mathbf{x}$ 的位置沿下降最陡的方向更新，这意味着 $\mathbf{x}$ 朝着减小 f 值的方向移动。5.4.2 节介绍了如何使用梯度下降法学习人工神经网络的权重参数。该方法的概要在算法 E.3 中给出。注意，除更新公式外，该算法看上去与算法 E.2 非常相似。

算法 E.3 梯度下降法

1：令 $\mathbf{x}_0$ 为初始点。
2：**while** $\|\nabla f(\mathbf{x}_0)\| > \varepsilon$ **do**
3：$x = x_0 - \lambda\nabla f(\mathbf{x})$
4：$x_0 = x$
5：**end while**
6：**return** x

E.2 约束优化

本节考察当变量受限于各种约束时如何解优化问题。

E.2.1 等式约束

考虑受限于如下形式的等式约束：

$$g_i(\mathbf{x}) = 0,\ i = 1, 2, \cdots, p$$

求 $f(x_1, x_2, \cdots, x_d)$ 的极小值问题。

一种称作拉格朗日乘子的方法可以用来解约束优化问题。该方法涉及如下步骤。

(1) 定义拉格朗日函数 $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{p} \lambda_i g_i(\mathbf{x})$，其中 λ_i 是哑变量，称作**拉格朗日乘子**（Lagrange multiplier）。

(2) 令拉格朗日函数关于 $\mathbf{x}$ 和拉格朗日乘子的一阶导数等于 0：

$$\frac{\partial L}{\partial x_i} = 0, \forall i = 1, 2, \cdots, d$$

并且

$$\frac{\partial L}{\partial \lambda_i} = 0, \forall i = 1, 2, \cdots, p$$

(3) 求解步骤 2 得到的 $(d+p)$个方程，得到平稳点 $\mathbf{x}^*$和对应的诸 λ_i 的值。

下面的例子解释如何使用拉格朗日乘子方法。

例 E.2　令 $f(x, y) = x + 2y$。假设我们受限于约束 $x^2 + y^2 - 4 = 0$，希望极小化函数 $f(x, y)$。可以用如下方式使用拉格朗日乘子方法求解这个约束优化问题。

首先，我们引入拉格朗日函数

$$L(x, y, \lambda) = x + 2y + \lambda(x^2 + y^2 - 4)$$

其中 λ 是拉格朗日乘子。为了确定它的极小值，我们需要对拉格朗日函数对它的参数求导：

$$\frac{\partial L}{\partial x} = 1 + 2\lambda x = 0 \tag{E-10}$$

$$\frac{\partial L}{\partial y} = 2 + 2\lambda y = 0 \tag{E-11}$$

$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 4 = 0$$

解这些方程，我们得到 $\lambda = \pm\sqrt{5}/4, x = \mp 2/\sqrt{5}, y = \mp 4/\sqrt{5}$。当 $\lambda = \sqrt{5}/4$ 时，$f(-2/\sqrt{5}, -4/\sqrt{5}) = -10/\sqrt{5}$。类似地，当 $\lambda = -\sqrt{5}/4$ 时，$f(2/\sqrt{5}, 4/\sqrt{5}) = 10/\sqrt{5}$。因此，函数 $f(x, y)$在 $x = -2/\sqrt{5}, y = -4/\sqrt{5}$ 取极小值。 □

E.2.2　不等式约束

考虑受限于如下形式的不等式约束：

$$h_i(\mathbf{x}) \leqslant 0, i = 1, 2, \cdots, q$$

求 $f(x_1, x_2, \cdots, x_d)$的最小值问题。

求解这类问题的方法与上面介绍的拉格朗日方法非常相似。然而，不等式约束把一些附加条件施加到优化问题上。特殊地，上述优化问题导致如下拉格朗日函数：

$$L = f(\mathbf{x}) + \sum_{i=1}^{q} \lambda_i h_i(\mathbf{x}) \tag{E-12}$$

和称作 Karush-Kuhn-Tucker (KKT)条件的约束：

$$\frac{\partial L}{\partial x_i} = 0, \forall i = 1, 2, \cdots, d \tag{E-13}$$

$$h_i(\mathbf{x}) \leqslant 0, \forall i = 1, 2, \cdots, q \tag{E-14}$$

$$\lambda_i \geqslant 0, \forall i = 1, 2, \cdots, q \tag{E-15}$$
$$\lambda_i h_i(\mathbf{x}) = 0, \forall i = 1, 2, \cdots, q \tag{E-16}$$

注意，拉格朗日乘子在不等式约束中出现，不再是不受限的。

例 E.3 假设我们想最小化函数 $f(x, y) = (x-1)^2 + (y-3)^2$，受限于如下约束：

$$x + y \leqslant 2 \text{ 并且 } y \geqslant x$$

该问题的拉格朗日函数为 $L = (x-1)^2 + (y-3)^2 + \lambda_1(x+y-2) + \lambda_2(x-y)$，受限于如下 KKT 约束：

$$\frac{\partial L}{\partial x} = 2(x-1) + \lambda_1 + \lambda_2 = 0 \tag{E-17}$$
$$\frac{\partial L}{\partial y} = 2(y-3) + \lambda_1 - \lambda_2 = 0 \tag{E-18}$$
$$\lambda_1(x + y - 2) = 0 \tag{E-19}$$
$$\lambda_2(x - y) = 0 \tag{E-20}$$
$$\lambda_1 \geqslant 0, \lambda_2 \geqslant 0, x+y \leqslant 2, y \geqslant x \tag{E-21}$$

为了求解以上方程，我们需要考察公式（E-19）和公式（E-20）的所有可能情况。

情况 1 $\lambda_1 = 0, \lambda_2 = 0$。在这种情况下，我们得到如下方程：

$$2(x-1) = 0,\ 2(y-3) = 0$$

其解为 $x = 1$，$y = 3$。由于 $x + y = 4$，这不是一个可行的解，因为它违反约束 $x + y \leqslant 2$。

情况 2 $\lambda_1 = 0, \lambda_2 \neq 0$。在这种情况下，我们得到如下方程：

$$x - y = 0, 2(x-1) + \lambda_2 = 0, 2(y-3) - \lambda_2 = 0,$$

其解为 $x = 2, y = 2, \lambda_2 = -2$。这不是一个可行的解，因为它违反约束 $\lambda_2 \geqslant 0$ 和 $x+y \leqslant 2$。

情况 3 $\lambda_1 \neq 0, \lambda_2 = 0$。在这种情况下，我们得到如下方程：

$$x + y - 2 = 0, 2(x-1) + \lambda_1 = 0, -2(x+1) + \lambda_1 = 0,$$

其解为 $x = 0, y = 2, \lambda_1 = 2$。这是一个可行的解。

情况 4 $\lambda_1 \neq 0, \lambda_2 \neq 0$。在这种情况下，我们得到如下方程：

$$x+y-2 = 0, x-y = 0, 2(x-1) + \lambda_1 + \lambda_2 = 0, 2(y-3) + \lambda_1 - \lambda_2 = 0,$$

其解为 $x = 1, y = 1, \lambda_1 = 2, \lambda_2 = -2$。这不是一个可行的解。

因此，该问题的解是 $x = 0$，$y = 2$。 □

求解 KKT 条件可能是一项相当艰巨的任务，当约束不等式的数量较大时尤其如此。在这种情况下，求闭型解不再可行，而需要使用诸如线性和二次规划这样的数值优化技术。