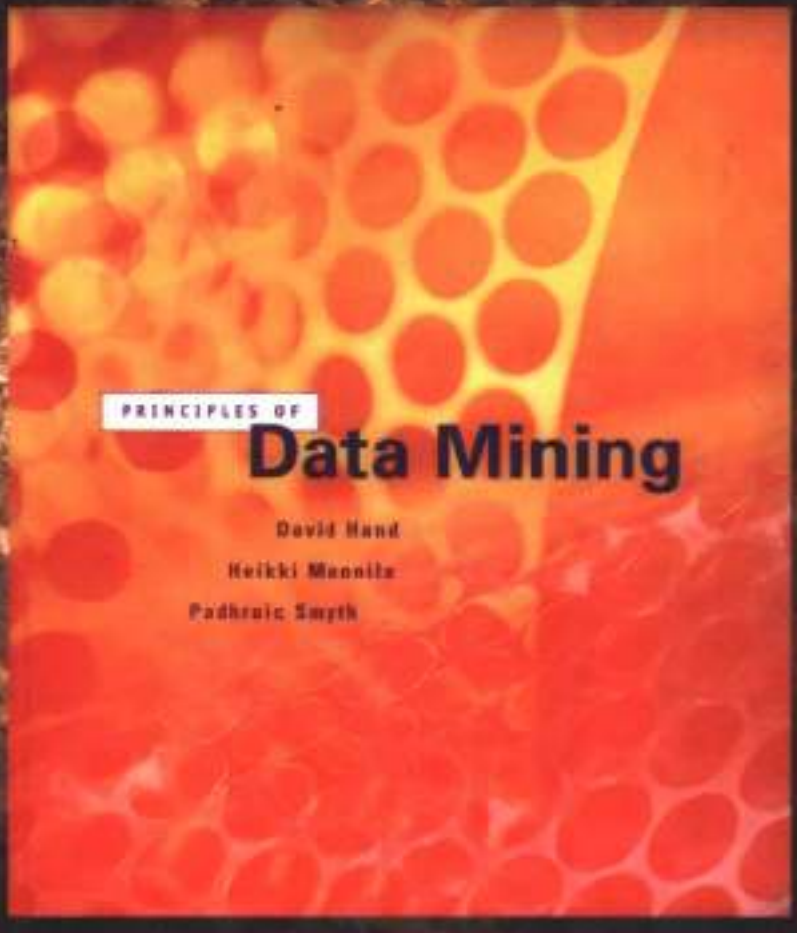
计 算 机 科 学 丛 书

数据挖掘原理

David Hand Heikki Mannila Padhraic Smyth 著 张银奎 廖丽 宋俊 等译



Principles of Data Mining

机械工业出版社 China Machine Press
中信出版社 CITIC PUBLISHING HOUSE

很多学科都面临着一个普遍问题：如何存储、访问异常庞大的数据集，并用模型来描述和理解它们？这些问题使得人们对数据挖掘技术的兴趣不断增强。长期以来，很多相互独立的不同学科分别致力于数据挖掘的各个方面。本书把信息科学、计算科学和统计学在数据挖掘方面的应用融合在一起，是第一本真正的跨学科教材。

本书由三部分构成。第一部分是基础，介绍了数据挖掘算法及其应用所依赖的基本原理。讨论方法直观易懂，深入浅出。第二部分是数据挖掘算法，系统讨论了如何构建求解特定问题的不同算法。讨论的内容包括用于分类和回归的树及规则、关联规则、信念网络、传统统计模型，以及各种非线性模型，比如神经网络和“基于记忆”的局部模型。第三部分介绍了如何应用前面讨论的算法和原理来解决现实世界中的数据挖掘问题。谈到的问题包括元数据的作用，如何处理残缺数据，以及数据预处理。

作者简介

David Hand 是伦敦帝国大学数学系统计学教授。Heikki Mannila是赫尔辛基工业大学计算科学与工程系的教授，诺基亚研究中心的研究员。Padhraic Smyth是加州大学Irvine分校信息与计算科学系的副教授

ISBN 7-111-11577-5

9 787111 115779

HZ BOOKS 华章图书

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037

购书热线：(010)68995259，8006100280 (北京地区)

读者服务信箱：hzedu@hzbook.com

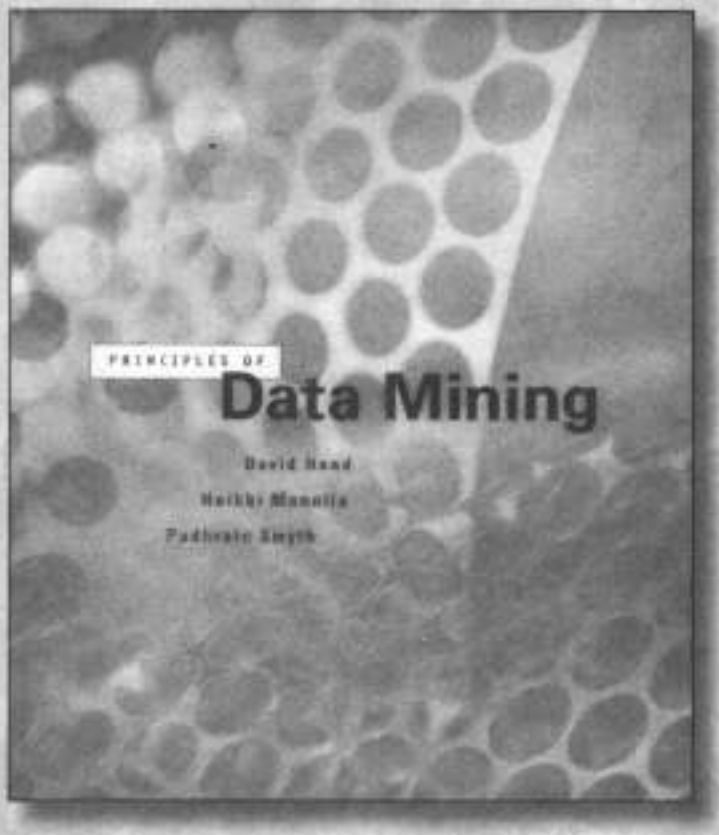
ISBN 7-111-11577-5/TP · 2778

定价：48.00 元

计 算 机 科 学 丛 书

数据挖掘原理

David Hand Heikki Mannila Padhraic Smyth 著 张银奎 廖丽 宋俊 等译



Principles of Data Mining

机械工业出版社 China Machine Press
中信出版社 CITIC PUBLISHING HOUSE

面对日益庞大的数据资源，人们迫切需要强有力的工具来“挖掘”其中的有用信息，数据挖掘就是针对这一需求而发展起来的一门汇集统计学、机器学习、数据库、人工智能等学科内容的新兴的交叉学科。本书深入探讨数据挖掘原理，把信息科学、计算科学和统计学对数据挖掘的贡献融合在一起，是一本真正的跨学科教材。

本书适合计算机专业、应用数学专业高年级本科生和研究生，以及致力于数据挖掘方向的研究和工作人员等阅读。

David Hand，et al：Principles of Data Mining (ISBN 0-262-08290-x).

Original English language edition copyright © 2001 by Massachusetts Institute of Technology.

All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字版由麻省理工学院出版社授权机械工业出版社和中信出版社共同出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

版权登记号：图字：01-2002-0607

图书在版编目（CIP）数据

数据挖掘原理/（英）汉德（Hand, D.）著；张银奎等译. -北京：机械工业出版社，2003.4
（计算机科学丛书）
书名原文：Principles of Data Mining
ISBN 7-111-11577-5

Ⅰ. 数… Ⅱ. ①汉…②张… Ⅲ. 数据采集 Ⅳ. TP274

中国版本图书馆 CIP 数据核字（2003）第 016122 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）
责任编辑：辛再甫 刘立卿

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2003 年 4 月第 1 版第 1 次印刷
787mm×1092mm 1/16 • 24 印张
印数：0 001-5 000册
定价：48.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“国外经典教材”是响应教育部提出的使用外版教材的号召，为国内高校的计算机本科教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这20多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995265
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

译 者 序

网络和存储技术的迅猛发展，使数据的传播和积累速度不断提高，但当我们为拥有极其详尽的数据而欣喜的同时，也发现新的数据处理和提炼技术非常匮乏。面对日益庞大的数据资源，人们迫切需要更强有力的工具来“挖掘”其中有用的信息。数据挖掘就是针对这一需求而发展起来的一门新兴学科。

本书是数据挖掘领域的三位专家的一本最新力作。全书共 14 章，从内容上可以分为三大部分。第一部分是基础，介绍了数据挖掘算法及其应用所依赖的基本原理。第二部分是数据挖掘算法，系统讨论了如何构建求解特定问题的不同算法。第三部分介绍了如何应用前面讨论的算法和原理来求解现实世界中的数据挖掘问题。该书虽然出版时间不长，但已经得到很多好评，被很多大学选为教材。该书的特色在于：

第一，以统计学家的视角系统解析了数据挖掘技术所依赖的统计原理。因为数据挖掘所针对的是不完整和存在“噪声”的庞大数据集，所以统计学中的概率分析和检验技术在数据挖掘中有着极其重要的作用。本书系统讨论了各种数据挖掘算法之中所蕴含的统计原理，为读者深入学习奠定了坚实的理论基础。

第二，应用面向组件的思想，把数据挖掘算法分解为既相互联系，又相对独立的几大组件，即模型结构、评分函数、搜索方法和数据管理技术。这样便可以把成熟的组件和分布式处理技术（如 COM、DCOM 和 CORBA 等）应用到数据挖掘领域之中，把各种数据挖掘算法封装为灵活的组件，从而可以加快数据挖掘算法的开发、重组、交流和工程化、商业化速度。

第三，全书既具有极强的理论性，又不脱离实践。既深入浅出地讲授了很多非常基本的理论，如数据、测量、概率分布、统计检验、算法的定义和复杂度等，又系统全面地介绍了各种模型（参数模型、非参数模型和混合模型）、模式和评分函数，同时，所有的理论介绍都与实践应用保持着密切的关系。既不空泛，也不僵化。书中还穿插了很多实例和图形，这进一步增强了理论的说服力。

第四，内容精练，分析独到。数据挖掘具有典型的多学科性，涉及的内容极其广泛。本书有的放矢，集中讨论基本的理论和各种算法中所蕴含的思想精华，可谓是授读者以“渔”。而且全书视角新颖，分析独到，可以说是一部用心良苦的作品。

根据以上特征，适合阅读本书的读者包括：应用数学、计算机科学等专业高年级本科学生和研究生，致力于数据挖掘方向的研究和工作人员，对数学建模、分类和回归算法、模式识别、图像和内容检索等感兴趣的其他读者。

本书是三位作者多年研究成果和教学实践的结晶。David Hand 是伦敦帝国大学数学系的统计学教授和消费者信誉研究组（Consumer Credit Research Group）的主席，他是统计和智能计算领域的一位资深专家，迄今已发表了大量著作，2002 年他被授予英国统计学会银质奖章。Heikki Mannila 是赫尔辛基工业大学计算科学与工程系的教授、HIIT（Helsinki Institute for Information Technology）基础研究部的主任、诺基亚研究中心的研究员。Padhraic Smyth

是加利福尼亚大学 Iryine 分校信息与计算科学系的副教授。

这个译本来自多人的共同努力，参加本书翻译的有张银奎（第 1、2、8～14 章及附录）、宋俊（第 3 章）、廖丽（第 5 章）、闫绍松（第 6 章）、张猛、郑靓（序言和第 7 章）和龙欣（中文版序），全书由张银奎统稿和审校。另外，曾华军对本书一些术语的译法提出了宝贵意见，并阅读了部分译稿；龙晓华、徐峰等人也对部分内容的翻译提出了很多宝贵建议。翻译一本书绝不像批评一本书译得不好那么容易，特别是这本书专业性很强，我们的水平也很有限，因此，错误和不当之处敬请各位读者批评指正。

译　者

2002 年 12 月 15 日

中文版序言（附原文）

非常高兴我们的书被译成了中文。在中国这样一个具有世界最多人口的国家发展分析庞大数据集的尖端技术是再合适不过的了。

张银奎先生为该书的翻译付出了很多劳动，在此我们对其表示真诚的谢意。尽管我们三人都不懂中文，但从他在翻译该书过程中向我们询问的问题中可以看出，他非常好地理解了我们想表达的内容，并领会了我们写作的目标。而且，他还为我们指出了第一版中的一些打字错误，所以他的翻译工作对本书的英文再版也起到了改进作用。

科学是没有国界的，像这样的翻译说明我们完全可以进行全球化的国际性合作。我们在写作本书时发现了许多乐趣，希望中国的读者在阅读本书时能够找到同样多的乐趣。

We are delighted that our book has been translated into Chinese. Indeed, it is entirely appropriate that the country with the largest population on Earth should develop expertise in the analysis of huge datasets.

We would like to express our appreciation to Raymond Yinkuizhang for the superb job he has done in translating the book. Although none of the three of us can read Chinese, we knew from the detailed questions he asked while translating the book that he had an excellent understanding of what we were trying to say and an appreciation of what we were trying to achieve in writing it. Indeed, he spotted several typographical errors in the first edition, so his translation will also lead to improvement of the second English edition.

The scientific enterprise is international and translations such as this mean that we are able to contribute to a properly global international community. We hope our Chinese readers find as much enjoyment from reading the book as we found in writing it.

David Hand
Heikki Mannila
Padhraic Smyth

前　言

我们把从庞大的数据集或数据库中提炼有用信息的科学称为数据挖掘。它汇集了统计学、机器学习、数据库、模式识别、人工智能等学科的内容，是一门新兴的交叉学科。这些学科都致力于数据分析的某一个方面，因此它们有很多共性——但是每一学科又有其独有的特色，分别针对不同的问题和求解的不同方式。

由于数据挖掘涵盖了计算机科学和统计学中的很多主题，所以要在一本书中覆盖所有的相关材料是不可能的。因此，我们把焦点集中在那些我们认为特别重要的主题上。

从教学的角度来讲，本书主要适合于希望学习数据挖掘基本原理的较高年级（最后一年）大学生，或者是一、二年级的研究生；本书对于那些旨在更好地了解数据挖掘方法和技术的研究者和实践者也是有价值的。本书假定读者已经熟悉了概率论、微积分、线性代数和优化等学科中的基本概念——也就是说，诸如工程学、计算机科学、数学和经济学等专业的大学学历背景会为阅读和理解本书提供一个很好的基础。

目前，已经出版了许多关于数据挖掘的书籍，但其中大多数都是直接面向商业应用的，着重于特定的方法和算法（例如决策树分类），而不是一般性原理（例如参数估计和计算复杂性）。这些书对于一般了解和实例研究是很有价值的，但对于课堂教学来说有很多不足，因为底层的基本原理经常被忽略掉了。也有一些数据挖掘方面的书具有很强的专业性，但迄今为止这些书绝大部分是从计算机科学的角度出发的，特别是从数据库角度（Han and Kamber, 2000，该书中译本《数据挖掘：概念与技术》已由机械工业出版社出版。）或从机器学习的角度（Witten and Franke, 2000，该书中译本即将由机械工业出版社出版）。

本书的侧重点有所不同。我们的目标是分析数据挖掘的最基本特征。我们没有用很长的篇幅来讨论特定的数据挖掘应用，比如协同过滤（collaborative filtering）[⊖]、信用评分（credit scoring）以及欺诈探查（fraud detection）等，而是把焦点集中在这些应用所依赖的基本原理和算法上。但这并不是说本书忽视了应用，因为从根本上讲数据挖掘就是一门应用性学科。我们始终记着这一点，在探讨基本理论的同时，也介绍了非常多的可以运用（或者已经运用了）该理论的具体应用和研究实例。

我们认为，要精通数据挖掘既需要理解统计学又要理解计算科学。要掌握这两个不同的专业领域，不论对学生还是对老师来说都是一个比较大的挑战。对于一般的计算机科学家来说，统计学著作是相当难以理解的：冗长而枯燥的专业术语、隐含的假定、渐近性的证明，而且缺乏这些理论和数学概念究竟是如何在实际数据分析算法中真正实现的细节。对统计学家来说情况恰好相反：关于机器学习和数据挖掘的计算机科学文献中充满了对算法、伪代码、计算效率等的讨论，但往往却很少提到潜在的模型或推理过程。尽管如此，这两个学科对于处理庞大数据集来说都是至关重要的。既可以从“数学模型”角度，又可以从“计算算法”角度理解数据挖掘是正确把握其复杂性的关键。

⊖ 译注：简单来说就是对有相同购物历史的顾客提供交叉推荐服务。

在本书中，我们试图架起一座沟通这两个世界的桥梁，目的是把统计建模的思想和“现实世界”中的实际计算方法和算法联系起来。

本着这一宗旨，我们以一种有些与众不同的方式组织了本书的结构。首先我们讨论了建模和推理的基本原理，然后介绍了数据挖掘算法的系统框架——通过各种计算方法和算法把模型与数据联系起来，最后结合诸如分类和回归这样的具体技术例释了这些思想。因此，本书可分为三大部分：

1. 基础 第 1 章到第 4 章着重讨论数据和数据分析的基本原理。介绍了数据挖掘（第 1 章）、测量（第 2 章）、可视化数据（第 3 章）、不确定性和推理（第 4 章）的基本原理。

2. 数据挖掘组件 第 5 章到第 8 章着重讨论用以系统地创建和分析数据挖掘算法的各个标准部件，即我们所称的数据挖掘算法“组件”。第 5 章主要讨论分析算法的系统方法，我们认为这种“分组件”的方法为那些刚刚接触数据挖掘这一学科的初学者提供了一种非常有用的视角，可以系统地透视数据分析算法中那些非常容易令人困惑的地方。而后在这一框架下，我们对每个组件进行了广泛深入的讨论，第 6 章讨论模型表示方法，第 7 章讨论用来拟合模型和数据的评分函数，第 8 章讨论优化和搜索技术（数据管理在第 12 章讨论）。

3. 数据挖掘任务和算法 本书的前 8 章已经对数据挖掘的基本原理和组件进行了讨论，余下的章节（第 9 章到第 14 章）则致力于特定的数据挖掘任务以及针对这些任务的算法。我们将基本的数据挖掘任务组织成以下几类：密度估计和聚类（第 9 章）、分类（第 10 章）、回归（第 11 章）、模式发现（第 13 章）以及根据内容检索（第 14 章）。在这些章节中我们使用了第二部分所建立的框架结构，讨论了针对每一项任务的具体算法。例如，在对分类的讨论中，我们回答了这些问题：哪些模型和表示是值得考虑和有价值的？我们可以使用或者应该用哪些评分函数来训练分类器？哪些优化和搜索技术是必要的？一旦我们使用了某种方法来实际实现算法时，这个算法的复杂度如何？我们希望这种通用的方法使读者认识到，数据挖掘算法是建立在一些非常通用的系统原理之上的，而不是简单地将一些看起来并不相关的生僻算法堆积在一起。

如果将本书用于教学的话，那么正如在前面所提到的，本书的目标读者是具有以下专业背景的大学生：计算机科学、工程学、数学、自然科学以及像经济学这样的面向商业的很多专业。从教师的角度来说，在课程中应该如何覆盖本书的内容主要依赖于课时长度（例如 10 周还是 15 周）和学生对统计学和机器学习等基本概念的熟悉程度。举例来说，如果是为具有统计学基本概念的一年级研究生开设的 10 周长度的课程，那么教师可以简单地讲述前面的章节，提纲挈领地讨论第 3 章、第 4 章、第 5 章和第 7 章；并将第 1 章、第 2 章、第 6 章和第 8 章作为背景/补充读物要求学生阅读；然后把 10 周中的大部分时间放在第 9 章到第 14 章的内容上，对这些内容进行深入的讨论。

然而，许多同学和读者可能只有很少的或根本没有正式的统计学背景。令人遗憾的是许多理工科专业（例如计算机科学）的本科生或研究生只有非常有限的统计学知识，他们对许多现代程序中的统计思想知之甚少。由于本书很大程度上是从统计学的角度来讨论数据挖掘的，所以我们在计算机系学生中使用本书草稿的经验告诉我们：对于许多学生来说，在 10 周或 15 周的课程时间中掌握本书是一个不小的挑战，因为要完全吸收所有内容，他们必须掌握第 2 章到第 8 章中提到的相当大范围的统计学、数学和算法概念。因此，在教学或第一遍阅读时，可以跳过本书的一些章节，以降低难度。例如，本书第 11 章中的回归可能是最

具数学挑战性的章节，而跳过这一内容也不会影响对其他内容的理解。同样，第 9 章中的某些内容（比如说有关混合模型的内容）也可以跳过；第 4 章中的贝叶斯估计框架也如此。那么哪些内容是阅读的关键呢？我们认为第 1 章到第 5 章和第 7 章、第 8 章和第 12 章中的绝大多数内容对学生来说是必须掌握的，这些内容是掌握后续章节中的模型和算法思想（第 6 章包含了很多关于一般建模概念的有价值的内容，但是篇幅相当长，所以可以跳过以缩短时间）的关键。第 9 章、第 10 章、第 11 章、第 13 章和第 14 章是“针对各种任务”的，这些章的内容是彼此相对独立的，所以可以任意选择其中的一些章节（但是前提是已经相当好地掌握了第 1 章到第 8 章中的内容）。

建议那些仅具有很少统计学知识的学生，在学习本书第 4 章（关于不确定性）之前，应该复习一下概率论和统计学中的一些基本概念。如果连诸如条件概率和期望这样的基本概念都还没有熟练掌握的话，那么就会在第 4 章及以后章节的学习中遇到相当大的困难。本书附录中简要介绍了常见分布的定义和基本的概率知识，不过许多学生可能喜欢在学习新东西之前再复习他们大学期间的概率论和统计学教材。

另一方面，对于那些具有坚实统计学背景的读者（例如统计专业学生或是对数据挖掘有兴趣的统计学家）来说，本书的绝大部分内容看起来相当熟悉，甚至有的统计学读者可能会倾向于说：“咳，这本数据挖掘的材料在很多方面与应用统计学的内容非常相似啊！”这句话确实多少有些道理，因为数据挖掘技术（在我们看来）在相当大的程度上依赖于统计模型和方法。然而，统计学者在本书的很多地方都会很容易地发现相当多的新内容：第 1 章的总括部分、第 5 章的算法观点、第 7 章的评分函数观点、从第 12 章到第 14 章的数据库原理、模式发现以及根据内容检索等。另外，我们还从数据挖掘的角度展示了许多传统的统计学概念（例如分类、聚类和回归等），以及在普通统计学教材中通常不包括的有关算法和计算复杂度的丰富内容。包括如何将各种技术运用到不同的数据挖掘应用中。虽然如此，统计学者还是会在本书中发现许多熟悉的材料。如果要从计算和数据管理的角度讨论数据挖掘，那么可以参阅参考文献中列出的 Han and Kamber（2000）；如果需要侧重于商业应用的材料，那么可以参阅参考文献中列出的 Berry and Linoff（2000）。这些教材可以作为课堂教学的补充读物。

总而言之，本书讨论了用于数据挖掘的各种工具，并将它们分解为不同的组成部分，以便看到各个组成部分间的相互关系和结构。本书不仅给出了如何设计这些工具的内幕，而且力图使读者在面临特定的问题时，能够独立设计出合适的数据挖掘工具。本书也阐释了为什说数据挖掘是一个过程——不是那些一蹴而就的任务，而是一种“发现——表示——再调查”的持续过程。本书也包含了大量针对现实数据的应用，其中很多是从作者本人所从事的科研和应用研究中选摘的。为了教学的方便，所有讨论的数据集合并非都是很大，因为这样解释起来更加简单。而且一旦领会了其中的思想，就可以很容易地把这些思想应用到现实大小的数据集中。

综上所述，数据挖掘技术的确是一门令人兴奋的学科。当然，和所有的科研事业都一样，许多努力将是没有回报的（做一项保证会成功的研究，这样的情况是罕见的，而且也是乏味的）。但是一旦一个令人兴奋的发现（信息的宝石）“出土”，这些努力也就获得了成倍的补偿。我们希望本书能够激励读者前进并发现自己的宝石！

我们衷心地感谢 Christine McLaren 允许我们使用红血球数据作为第 9 章和第 10 章的演示实例。Padhraic Smyth 在本书中的工作受到了美国国家科学基金会（Grant IRI-9703120）

的部分支持。

我们还要感谢 Niall Adams 帮助制作了本书的部分图表，Tom Benton 帮助完成了样稿的校对，以及 XianPing Ge 对参考书目的格式化。当然，本书中一定还存在很多错误，这是作者应该负责的（不过，我们三个中的每一个都保留了责备其他两个人的权利）。

最后，我们还要感谢我们各自的妻子和家庭，在写作本书的漫长而且看似没有终点的整个过程中，她们为我们提供了极大的鼓励和支持！

目　录

出版者的话
专家指导委员会名单
译者序
中文版序言
前言
第 1 章　绪论 …… 1
1.1　数据挖掘简介 …… 1
1.2　数据集属性 …… 3
1.3　结构类型：模型和模式 …… 5
1.4　数据挖掘任务 …… 7
1.5　数据挖掘算法的组件 …… 10
1.5.1　评分函数 …… 10
1.5.2　优化和搜索方法 …… 10
1.5.3　数据管理策略 …… 11
1.6　统计和数据挖掘的相互关系 …… 11
1.7　数据挖掘：打捞、探查还是垂钓 …… 13
1.8　本章归纳 …… 14
1.9　补充读物 …… 15
第 2 章　测量和数据 …… 17
2.1　简介 …… 17
2.2　测量类型 …… 17
2.3　距离尺度 …… 20
2.4　数据转化 …… 25
2.5　数据形式 …… 28
2.6　单个测量的数据质量 …… 29
2.7　数据群体的数据质量 …… 30
2.8　本章归纳 …… 33
2.9　补充读物 …… 33
第 3 章　可视化和探索数据 …… 35
3.1　简介 …… 35
3.2　总结数据：几个简单例子 …… 36
3.3　显示单个变量的一些工具 …… 37
3.4　显示两个变量间关系的工具 …… 41
3.5　显示两个以上变量间关系的工具 …… 46
3.6　主分量分析 …… 48
3.7　多维缩放 …… 54
3.8　补充读物 …… 58
第 4 章　数据分析和不确定性 …… 61
4.1　简介 …… 61
4.2　处理不确定性 …… 61
4.3　随机变量和它们的关系 …… 63
4.4　样本和统计推理 …… 66
4.5　估计 …… 69
4.5.1　估计量的理想属性 …… 69
4.5.2　最大似然估计 …… 70
4.5.3　贝叶斯估计 …… 76
4.6　假设检验 …… 81
4.6.1　古典假设检验 …… 82
4.6.2　数据挖掘中的假设检验 …… 85
4.7　采样方法 …… 87
4.8　本章归纳 …… 90
4.9　补充读物 …… 91
第 5 章　数据挖掘算法概览 …… 93
5.1　简介 …… 93
5.2　建立树分类器的 CART 算法 …… 95
5.3　数据挖掘算法的化约主义观点 …… 99
5.3.1　用于回归和分类的多层感知器 …… 99
5.3.2　关联规则学习的 A Priori 算法 …… 102
5.3.3　检索文本的向量空间算法 …… 104
5.4　讨论 …… 105

5.5 补充读物 ……106
第6章 模型和模式……107
6.1 概述 ……107
6.2 建模基础 ……108
6.3 用于预测的模型结构 ……109
6.3.1 具有线性结构的回归模型……109
6.3.2 用于回归的局部分段模型结构 ……112
6.3.3 “基于记忆”的非参数局部模型 ……113
6.3.4 模型结构的随机部分 ……114
6.3.5 用于分类的预测模型 ……116
6.3.6 选择适当复杂度的模型 ……117
6.4 概率分布和密度函数模型 ……118
6.4.1 一般概念 ……119
6.4.2 混合模型 ……119
6.4.3 无序范畴型数据的联合分布 ……121
6.4.4 因式分解和高维空间中的独立性 ……121
6.5 维度效应 ……124
6.5.1 高维数据的变量选择 ……125
6.5.2 高维数据的变换 ……126
6.6 用于结构化数据的模型 ……127
6.7 模式结构 ……130
6.7.1 数据矩阵中的模式 ……130
6.7.2 字符串模式 ……132
6.8 参考读物 ……133
第7章 数据挖掘算法的评分函数……135
7.1 简介 ……135
7.2 对模式进行评价……136
7.3 预测性评分函数和描述性评分函数 ……137
7.3.1 评价预测模型的评分函数……137
7.3.2 评价描述模型的评分函数……139
7.4 评价不同复杂度的模型 ……140
7.4.1 模型比较的一般概念 ……141
7.4.2 再谈偏差-方差 ……142
7.4.3 惩罚复杂模型的评分函数……144
7.4.4 使用外部验证的评分函数 ……145
7.5 模型和模式的评价 ……146
7.6 鲁棒方法 ……148
7.7 补充读物 ……148
第8章 搜索和优化方法 ……151
8.1 简介 ……151
8.2 搜索模型或模式 ……152
8.2.1 搜索背景 ……152
8.2.2 数据挖掘中的状态空间搜索 ……154
8.2.3 简单贪婪搜索算法 ……155
8.2.4 系统搜索和搜索启示 ……156
8.2.5 分支定界法 ……157
8.3 参数优化方法 ……158
8.3.1 参数优化：背景……158
8.3.2 闭合形式解和线性代数方法 ……159
8.3.3 优化平滑函数的基于梯度方法 …160
8.3.4 一元参数优化 ……160
8.3.5 多元参数优化 ……163
8.3.6 约束优化 ……165
8.4 存在残缺数据时的优化：EM算法……166
8.5 在线和单扫描算法 ……169
8.6 随机搜索和优化技术……170
8.7 补充读物 ……171
第9章 描述建模 ……173
9.1 简介 ……173
9.2 通过概率分布和密度描述数据 ……174
9.2.1 简介 ……174
9.2.2 用来估计概率分布和密度的评分函数 ……174
9.2.3 参数密度模型 ……175
9.2.4 混合分布和密度……178
9.2.5 混合模型的EM算法 ……179
9.2.6 非参数的密度估计 ……181
9.2.7 范畴型数据的联合分布 ……183
9.3 聚类分析背景 ……186
9.4 基于划分的聚类算法……188

9.4.1 基于划分聚类的评分函数……188
9.4.2 基于划分聚类的基本算法……191
9.5 层次聚类 ……196
9.5.1 凝聚方法 ……197
9.5.2 分裂方法 ……199
9.6 基于混合模型的概率聚类 ……200
9.7 补充读物 ……206
第 10 章 用于分类的预测建模 ……209
10.1 预测建模概览 ……209
10.2 分类建模简介 ……210
10.2.1 判别分类和决策边界 ……210
10.2.2 分类的概率模型……211
10.2.3 建立实际的分类器 ……213
10.3 感知器……216
10.4 线性判别式 ……217
10.5 树模型……219
10.6 最近邻方法 ……222
10.7 logistic 判别式分析 ……224
10.8 朴素贝叶斯模型 ……224
10.9 其他方法 ……226
10.10 分类器的评估和比较 ……228
10.11 高维分类的特征选取 ……230
10.12 补充读物 ……231
第 11 章 用于回归的预测建模 ……233
11.1 简介 ……233
11.2 线性模型和最小二乘法拟合 ……233
11.2.1 拟合模型的计算问题 ……235
11.2.2 线性回归的概率解释 ……236
11.2.3 拟合后模型的解释 ……238
11.2.4 推理和泛化 ……239
11.2.5 模型搜索和建模……240
11.2.6 模型诊断和审查……241
11.3 推广的线性模型 ……243
11.4 人工神经网络 ……247
11.5 其他高度参数化的模型……249
11.5.1 推广的相加模型……249
11.5.2 投影追踪回归 ……250
11.6 补充读物 ……251
第 12 章 数据组织和数据库 ……253
12.1 简介 ……253
12.2 存储器层次 ……253
12.3 索引结构 ……254
12.3.1 B-树 ……254
12.3.2 哈希索引……255
12.4 多维索引 ……256
12.5 关系数据库 ……256
12.6 操纵表格 ……259
12.7 结构化查询语言 ……261
12.8 查询的执行和优化 ……263
12.9 数据仓库和在线分析处理 ……264
12.10 OLAP 的数据结构……265
12.11 字符串数据库 ……266
12.12 海量数据集、数据管理和数据挖掘 ……266
12.12.1 把数据都放入主存储器 ……267
12.12.2 数据挖掘算法的可伸缩版本 …267
12.12.3 考虑磁盘访问的有针对性算法 ……268
12.12.4 伪数据集和充分统计量 ……268
12.13 补充读物 ……269
第 13 章 寻找模式和规则 ……271
13.1 简介 ……271
13.2 规则表示 ……272
13.3 频繁项集和关联规则 ……272
13.3.1 简介 ……272
13.3.2 寻找频繁集和关联规则……274
13.4 推广 ……276
13.5 寻找序列中的片段 ……277
13.6 选择发现的模式和规则 ……278
13.6.1 简介 ……278
13.6.2 寻找模式的启发式搜索……278
13.6.3 有趣度标准 ……279

13.7 从局部模式到全局模型 ……280
13.8 预测规则归纳 ……281
13.9 补充读物 ……283
第 14 章 根据内容检索 ……285
14.1 简介 ……285
14.2 检索系统的评价 ……286
14.2.1 评价检索性能的困难之处 ……286
14.2.2 查准率对查全率 ……287
14.2.3 查准率和查全率的实践应用 ……288
14.3 文本检索 ……289
14.3.1 文本的表示 ……289
14.3.2 匹配查询和文档 ……292
14.3.3 隐含语义索引 ……294
14.3.4 文档和文本分类 ……297
14.4 对个人偏好建模 ……297
14.4.1 相关性反馈 ……297
14.4.2 自动推荐系统 ……298
14.5 图像检索 ……299
14.5.1 图像理解 ……299
14.5.2 图像表示 ……299
14.5.3 图像查询 ……300
14.5.4 图像恒定性 ……301
14.5.5 图像检索的推广 ……301
14.6 时间序列和序列检索 ……301
14.6.1 时间序列数据的全局模型 ……302
14.6.2 时间序列的结构和形状 ……303
14.7 本章归纳 ……304
14.8 补充读物 ……305
附录 随机变量 ……307
参考文献 ……311
索引 ……340

第1章 绪 论

1.1 数据挖掘简介

数据采集和存储技术的进步导致庞大的数据库日益增多。这已经发生在人类耕耘的几乎所有领域，从普通的（比如超市业务数据、信用卡使用记录、电话呼叫清单以及政府统计数据）到不太普通的（比如天体图像、分子数据库和医疗记录）。那么，能否从这些数据中提取出对数据库拥有者有价值的信息呢？毫无疑问，人们对这个问题的兴趣在不断增长。而且已经形成了致力于这个任务的一门学科，称为“数据挖掘（data mining）”。

定义一门学科总是一件容易引起争论的事情，学者们经常反对给他们的研究领域划定精确的范围和界限。考虑到这一点，并且想到有些人可能不喜欢细枝末节，所以我们在本书中采用以下的数据挖掘定义：

> 数据挖掘就是对观测到的数据集（经常是很庞大的）进行分析，目的是发现未知的关系和以数据拥有者可以理解并对其有价值的新颖方式来总结数据。

通过数据挖掘过程所推导出的关系和摘要经常被称为模型（model）或模式（pattern）。例如线性方程、规则、聚类（cluster）、图、树结构以及用时间序列表示的循环模式。

上面定义中所说的“观测到的数据”，是与“实验得到的”数据相对而言的。一般来说数据挖掘所处理的数据是为了其他某个目的已经收集好的，而不是为了数据分析本身去收集的（例如，这些数据可能是银行中为正常业务所需而收集的）。这意味着数据挖掘的目标根本不在于数据采集策略。这是数据挖掘区别于大多数统计任务的一个特征，在统计中经常是利用高效率的策略来采集数据以回答特定的问题。由于这个原因，数据挖掘经常被称为“次级的”数据分析。

定义中还提到了数据挖掘所分析的经常是很庞大的数据集。如果仅涉及很小的数据集，那么我们就可以仅仅讨论统计学家们所使用的标准数据探测和分析方法了。当面对很庞大的数据集时，新的问题就产生了。有些问题是与如何存储或访问这些数据有关的，这可能很好解决；但是还有很多更重要的问题，比如如何表示数据，如何在合理的时间内分析数据，以及如何判定一个表面上的关系是否仅仅是偶然发生的，并不能反映任何潜在的事实。大多数情况下，现有的数据仅仅是整个总体（或者可能是一个猜想的超总体（superpopulation））的一个样本；最终的目的可能是从这个样本泛化（generalize）到总体。举例来说，我们可能希望预测未来客户的可能行为或判断我们未见过的蛋白质的结构特征。通过标准的统计方法可能无法实现这样的泛化，因为数据经常不是“随机样本”（这是经典统计方法必须的），而是“顺便的”或者说“机会的”样本。不过，有时我们可能想要用某种方式总结或压缩（compress）一个庞大的数据集，使得到的结果更容易让人理解，并不带有任何的泛化目的。例如，当我们面对某个国家的全部人口普查数据或数百万条的零售业务记录时，便会有这样的问题。

⊖ 边栏数字为原书页码，与索引中的页码相呼应。

当然，从数据集中发现的关系和结构必须是新颖的（novel）。“反刍”已经确立的关系（除非，目的是确认假设，以判定一个建立的模式是否也在新的数据集中存在）或必然的关系（例如，所有怀孕的患者都是女的）是没什么价值的。当然，新颖性是相对用户以前的知识而言的。不幸的是几乎所有数据挖掘算法都不考虑用户以前的知识。由于这个原因，我们在本书中不过多地讨论新颖性。那是一个正在研究的课题。

虽然新颖性是我们要寻找的关系的一个重要特征，但是它不足以确定是否值得搜索一个关系。需要指出的是，寻找到的关系还必须是易于理解的。例如简单的关系比繁琐的关系更
2 易于理解，所以如果其他方面都相同的话我们会优先选择简单的关系。

数据挖掘经常被置于更广阔的数据库知识发现（knowledge discovery in databases）也就是KDD的大背景下。KDD这个术语来源于人工智能（AI）领域。KDD过程包括几个阶段：选择目标数据、预处理数据、转化数据（如果需要）、进行数据挖掘以提取模式和关系、解释并评价发现的结构。为这个过程中的数据挖掘部分精确划定界限也不是简单的事，例如很多人认为数据转化是数据挖掘的一个必不可少的部分。在这本教材中我们主要集中讨论各种算法而不是整个过程。例如我们不会花很多时间来讨论数据预处理问题，比如数据净化、数据核对和定义变量等。相反我们将集中讨论一些基本的原理，包括对数据建模以及如何构造算法过程以把这些模型拟合到数据。

寻找数据集中的关系也就是寻找精确、方便并且有价值地总结了数据的某一特征的表示，这个过程包括很多个步骤：

- 决定要使用的表示的特征和结构；
- 决定如何量化和比较不同表示拟合数据的好坏（也就是选择一个“评分（score）”函数）；
- 选择一个算法过程使评分函数最优；
- 决定用什么样的数据管理原则以高效地实现算法。

这本教材的目标就是系统详细地讨论这些问题。书中既包括了基本的原理（第2章到第8章），又包括了如何应用这些原理来构造和评估特定的数据挖掘算法（第9到第14章）。

例1.1　回归分析是很多读者熟悉的工具。在最简单的回归形式中，通过$Y=aX+b$的形式把一个预报（predictor）变量X与一个响应（response）变量Y联系起来。举例来说，我们可以使用这种方法建立一个模型，通过这个模型我们可以根据一个人的年收入预测他每年的信用卡支出。当然这个模型不会很完美，但由于支出一般是随着收入增长的，所以这个模型足可以作出一个粗略的刻画。根据上面列出的步骤，我们可以这样设计这个任务的解决方案：

3
- 表示模型中，响应变量spending与预测变量income线性相关。
- 在本例这样的情况下，最普遍使用的评分函数是模型预测支出与观测到支出间差异的平方和。
- 线性回归的最优化算法是非常简单的：可以把a和b表示为观测到支出和收入值的函数，我们将在第11章描述其中的数学细节。
- 除非数据集非常庞大，否则对于线性回归算法来说没有什么数据管理问题。数据的简单汇总（求和，对平方求和，以及对X和Y值的乘积求和）就足以计算出a和b的估计值。这意味着只要遍历数据一次就可以得到预测。

数据挖掘是一门跨学科的技术。统计学、数据库技术、机器学习、模式识别、人工智能、可视化技术都在数据挖掘中起着作用。而且就像难以定义这些学科间的严格界限一样，也很难定义这些学科和数据挖掘间的界限。在边缘上，一个人的数据挖掘问题可能是其他人的统计、数据库或机器学习问题。

1.2 数据集属性

我们首先讨论数据集的基本特征。

数据集是从某个环境或过程中取得的一系列测量结果。对于最简单的情况，我们有一系列对象，每一个对象都有统一的 p 个测量结果。这时我们可以把这 n 个对象的一系列测量结果看作一个 $n \times p$ 的数据矩阵。矩阵中的 n 行表示被测量的 n 个对象（例如，医治中的患者、信用卡用户，或从夜空中观测到的天体，比如各种星星和星系）。根据不同的上下文环境可以把这样的行称为个体（individual）、实体（entity）、实例（case）、对象（object）或记录（record）。

数据矩阵的另一维包含对每个对象所作的 p 种测量。通常我们假定对每个个体使用同样的 p 个测量指标，不过这未必和实际情况一致（例如，对于不同的患者可能使用不同的检验方法）。可以把数据矩阵的 p 个列称为变量（variable）、特征（feature）、属性（attribute）或者字段（field），与前面一样到底使用哪一种说法要看研究的上下文。在所有情况下，思想是一样的，即这些名字是指每一列所代表的测量。在第 2 章中我们将更详细地讨论测量的概念。

4

例 1.2 美国人口普查局每隔十年调查一次美国人口信息。这些信息中的一部分是对公共使用开放的，但所有能够识别出某个个人的信息都被删除了。这些数据集被称为公用微观数据样本（Public Use Microdata Sample），或 PUMS。可以按 5% 和 1% 的采样率得到这些数据。注意即使是对美国人口按 1% 采样，那也有 270 万条记录。这样的数据集可能包含几十个变量，比如人的年龄、总收入、职业、资产损益、教育程度等等。下面考虑表 1-1 所示的简单数据矩阵。注意这里的数据包含不同类型的变量，有些是连续型的、有些是范畴型的（categorical）。也请注意有些值是空缺的，比如 ID 为 249 的人的年龄，和 ID 为 255 的人的婚姻状况。在现实情况下，庞大的数据集中缺少某些测量结果是很普遍的。更容易导致错误的是测量结果中的噪声。例如，ID 为 248 的人的收入真的是 100 000 美元还是这仅仅是一个粗略的估计？

表 1-1　公用微观数据样本中的数据示例

ID	年　龄	性　别	婚姻状况	文化程度	收　入
248	54	男	已婚	高中毕业	100 000
249	()	女	已婚	高中毕业	12 000
250	29	男	已婚	大专	23 000
251	9	男	未婚	儿童	0
252	85	女	未婚	高中毕业	19 798
253	40	男	已婚	高中毕业	40 100
254	38	女	未婚	低于一年级	2 691
255	7	男	()	儿童	0
256	49	男	已婚	十一年级	30 000
257	76	男	已婚	博士	30 686

对于这种类型的数据，一个典型的任务是发现不同变量间的关系。例如我们可能想看一看从其他变量预测一个人的收入有多准确。我们也可能想看一看是否存在独特的人群，或者对发现变量的频繁值感兴趣。可以从加利福尼亚大学 Irvine 分校的机器学习资料库中在线得到包含部分变量的一些记录，http://www.ics.uci. edu/~mlearn/MLSummary.html。

5 数据是以很多种形式出现的，而且本书的目的也不是要开发全面的数据分类系统。事实上，现在还不清楚是否能够开发出这样一个全面的分类系统，因为在一个条件下很重要的数据特征可能在另一个条件下并不重要。然而有一些基本的差别是我们该注意的。一点是数量值和范畴值的不同（有时使用不同的名字来称呼这两类值）。一个数量值变量是按照某个数字比例测量的，并且至少在理论上是可以取任意值的。表 1-1 中的**年龄**和**收入**列是数量值变量的例子。相反，像**性别**、**婚姻状况**和**文化程度**这样的范畴值变量仅能取确定个数的离散值。医学上使用的三档严重程度（轻微、中等和严重）是范畴值的另一个常见例子。范畴值可能是有顺序的（对应一个自然的排序，就像文化程度）也可能就是一种标称（仅仅是对这个范畴的命名，像婚姻状况那样）。适合一种数据度量的分析技术未必适合另一种（不过这确实要看分析的目标——参见 Hand（1996），其中有更详细的讨论）。例如如果把婚姻状况表示为整数（比如 1 表示单身，2 表示已婚，3 表示丧偶，依此类推），那么对于这个样本中的这个指标计算数学平均值通常是没有意义的或者说是不恰当的。类似的，简单线性回归（把一个数量值变量预测为其他变量的函数）通常适于数量值的数据，如果应用到范畴值变量则是不恰当的。针对相似目标的其他技术可能更适合于范畴值变量。

不管如何定义测量尺度，它总是位于数据分类系统的底层。由此向上，我们会发现数据是按不同的关系和结构产生的。数据可能是按时间序列方式连续产生的，在这种情况下，数据挖掘可以针对整个时间序列，也可以针对这个序列的特定片段。数据也可能描述空间的关系，因此对单个记录来说，仅当从其他记录的上下文环境来考虑时才能看出它的完全含义。

考虑一个关于医疗患者的数据集。它可能包括对同一个变量（例如**血压**）的多个测量结果，每个测量结果对应不同的时间。某些患者可能有进一步的图像数据（例如 *X* 射线或磁共振图像），而其他人没有。某个人可能还有文字形式的数据，记录了专家对他的病情的注释和诊断。此外，在患者和医生、医院、以及地理位置间还可能存在一层关系。数据结构越复杂，我们需要的数据挖掘模型、算法和工具也就越复杂。

6 由于上面讨论的各种原因，$n \times p$ 的数据矩阵经常是对实际情况的一种过度简化或者说理想化。很多数据集不适合这样的简单格式。尽管原则上很多信息可以“压平”成 $n \times p$ 的矩阵（通过适当定义的 p 个变量），但是这经常会丢失嵌入在数据中的大多数结构信息。然而，当讨论数据分析的基本原理时，假定观测的数据存在于一个 $n \times p$ 的数据矩阵中经常是非常方便的，所以我们除非特别说明也使用这种方法。应该记住对于数据挖掘应用 n 和 p 可能都非常大。有必要说明观测到的数据矩阵也可能被称为其他名字，比如数据集、训练数据、样本、数据库等（往往不同术语来自不同的学科）。

例 1.3 文本文档一种是重要的信息来源，数据挖掘方法可以帮助人们从一系列庞大的文档（例如网页）中检索有价值的文本。每篇文档可以被看作单词和标点的序列。挖掘文本数据库的典型任务包括把文档分类到预先定义的类目中，把相似的文档聚类到一起，以及寻找匹配查询要求的文档。一个典型的文档集合是

"Reuters-21578, Distribution 1.0", 位于 http://www. research.att.com/~lewis。这个集合中的每一篇文档都是一篇短小的新闻专线文章。

一系列文本文档也可以被看作一个矩阵，行表示文档，列表示单词。表项（d, w）对应文档 d 中单词 w 的出现情况，可以是 w 在 d 中出现的次数，或者干脆是如果 w 在 d 中出现了则为 1 否则就为 0。尽管使用这种方法我们丢失了单词在文档中的顺序信息（因此也同时失去了大部分上下文语义），但这仍是一个对文档内容的不错的表示。对于一个文档集合，矩阵的行数就是文档的篇数，列数就是独立的单词数。因此庞大的多语言文档集合可能包含上百万行和几百或几千列。注意这样的数据矩阵将是非常稀疏的，也就是说大多数表项是 0。我们将在第 14 章更加详细地讨论文本挖掘。

例 1.4　另一种常见的数据类型是**事务数据**（transaction data），例如商店的销售清单，通过日期、客户 ID 以及商品和价格列表描述每一笔销售（或交易）。一个类似的例子是网络事务日志，一系列三元组（用户 ID、网页、时间）表示用户在某个时间访问了某个网页。网站的设计者和拥有者经常对了解用户浏览他们网站的模式非常感兴趣。

就像对待文本文档一样，我们可以把一系列事务数据转化为矩阵的形式。想像一个庞大的稀疏矩阵，每一行对应某个用户，每一页对应某个网页或某种商品。这个矩阵的表项可以是二进制的值（例如表示一个用户是否已经访问了一个特定网页）或整数值（例如表示一个用户已经访问了某个网页多少次）。

图 1-1 显示了矩阵形式的一种可视化表示，描述的是一个很大的零售事务数据集中的一小部分数据。行对应一个客户个体，列对应商品的种类，每一个黑色的表项表示对应那一行的客户购买了对应那一列的商品。即使是在这种简单的显示中，我们也可以发现某些明显的模式。例如，客户购买商品的种类和购买数量方面都有相当大的差异。另外，某几类商品有不少客户购买（例如，列 3、5、11、26），某些列是所有人都没有购买的（例如，列 18 和 19）。我们也看到有些类商品经常是被一起购买的（例如，列 2 和 3）。

但是也该注意到，通过这种"平面表示"我们可能丢失了信息的某些重要部分，比如购买的顺序和时间信息（比如商品是按什么顺序和在什么时间被购买的），以及各个商品间的结构关系信息（比如产品类别层次，网页间的链接，等等）。然而，把这样的数据看作一个标准的 $n \times p$ 矩阵经常是有价值的。例如这允许我们通过比较 p 维网页访问向量来定义用户间的距离，从而根据网页模式对用户进行聚类分析。我们将在第 9 章更加详细地讨论聚类。

1.3　结构类型：模型和模式

可以从很多角度来对数据挖掘所探寻的不同表示进行分类。一种方法是分析全局模型（model）和局部模式（pattern）的差异。

这里我们把模型结构定义为对数据集的全局性总结，它对整个测量空间的每一点作出描述。从几何角度讲，如果我们把数据矩阵的各行看作 p 维向量（也就是 p 维空间中的点），

那么模型可以对这个空间中的每一点（也就是所有对象）作出描述。例如，它可以把一个点分配到一个聚类或者预测出某个其他变量的值。即使缺少一些测量结果（也就是 p 维向量的一些分量是未知的），模型一般也可以对这样的（不完全）向量所表示的对象作出某种论断。

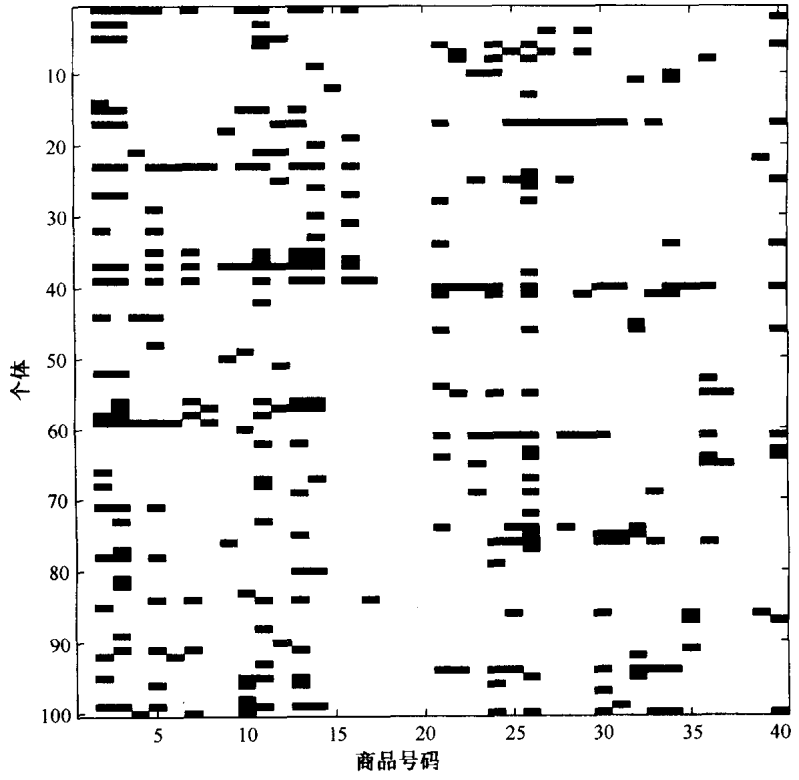


图1-1　显示为二进制图像的零售事务数据集的一部分，图中画出了100个客户个体（行）和40种商品（列）的情况

一个简单的模型可能取这样的形式：$Y = aX + c$，其中 Y 和 X 是变量，a 和 c 是模型的参数（数据挖掘过程中要决定的常数）。因为 Y 是 X 的线性函数，所以我们说这个模型的函数形式是线性的（linear）。这个术语在传统统计学中的含义与此略有不同。在统计学中，当一个模型是参数（parameter）的线性函数时说这个模型是线性的。在本书中我么将尽可能地明确指出我们使用的是哪一种线性概念，但当我们讨论模型的结构时（就像在这里）我们约定要考虑的线性特征是相对感兴趣变量的，而不是相对参数。例如，在传统统计范畴中，模型结构 $Y = aX^2 + bX + c$ 被认为是线性模型，但联系 Y 和 X 的模型的函数形式是非线性的（是一个二次多项式）。

8
~
9

与模型的全局性相反，模式结构（pattern structure）仅对变量变化空间的一个有限区域作出描述。一个例子是以下这种形式的简单概率性结论：如果 $X > x_1$，那么 $Y > y_1$ 的概率为 $p1$。这个结构由对变量 X 和变量 Y 的值的约束（constraint）组成，并以概率规则的形式将这两个变量联系起来。另一种方法是，我们可以把这个关系描述为条件概率 $p(Y > y_1 \mid X > x_1) = p1$，在语义上这与前面是等价的。另一个例子是，我们可能注意到事务记录中的某些类记录

没有显示出大多数记录所显示出的峰谷特征，因此需要一步分析寻找其中的原因。（利用这种方法，一家银行发现了那些属于已经去逝的人的账户。）

因此，与（全局的）模型不同，（局部的）模式描述的结构仅与数据或一小部分数据空间有关。或许仅有一部分记录具有某种特性，那么模式就是用来刻画这一部分数据的。例如，搜索通过邮件订购商品的数据库可能发现购买某个商品组合的人也可能购买其他的。还有，或许我们可以标识出和大多数记录（可以想像为是 p 维空间中的中央云团）完全不同的“孤立”记录。最后一个例子说明全局的模型和局部的模式有时是相互联系的（好比同一枚硬币的两个面）。为了探测异常的行为我们需要一种对正常行为的描述。局部模式的作用相当于统计分析中的信息诊断（diagnostic），局部模式探测方法已经应用于很多异常探测任务，例如工业生产中的故障探测，银行和其他商业活动中的欺诈行为探测。

注意上面描述的模型和模式都有参数与之相关，比如模型例子中的 a，b，c 和模式例子中的 x_1，y_1，$p1$。通常一旦我们已经建立了要寻找的结构形式，下一步就是要从现有的数据中估计出结构的参数。这个过程的细节将在第 4 章、第 7 章和第 8 章中讨论。一旦这些参数被赋值，我们便把这个特定的模型（比如 $y=3.2x+2.8$）称为“已经拟合的模型（fitted model）”，或简短起见就叫“模型”（类似地，对模式来说也如此）。把模型（或模式）结构和实际的（已拟合的）模型（或模式）区分开来是非常重要的。结构代表模型（或模式）的一般函数形式，还没有确定参数值。已拟合的模型或模式已经具有了特定的参数值。 10

在很多情况下，把模型和模式区别对待是有价值的。但与大多数为了便于人类理解而做的自然分类一样，二者的界限不是非常严格的：有时该把一个结构看作模型还是模式并不明确。这种情况下，最好不要过分关心哪一个更合适，区分它们仅是为了帮助我们讨论，而不是要强加一种约束。

1.4 数据挖掘任务

根据数据分析工作者的不同目标来划分数据挖掘任务的类型是很方便的。下面给出的分类不是唯一的，而且还可以进一步划分出更细致的任务，但它总结了数据挖掘活动的各个类型，并预览了本书后面将要描述的主要数据挖掘算法。

1. **探索性数据分析**（Exploratory Data Analysis，EDA）（第 3 章）：正像名字所暗示的，这种方法的宗旨就是对数据进行探索，在探索时我们对要寻找什么并没有明确的想法。通常，EDA 技术是交互式的（interactive）和可视化的（visual），对于维数比较低的数据集来说，有很多种有效的图形化显示方法。但随着维数（变量的个数 p）的增多，可视化变得越来越困难。当 p 大于 3 或 4 时，可以产生数据低维投影的投影技术（例如主要分量分析）是非常有价值的。数量很大的数据集可能不容易被有效的可视化，然而，可以使用缩放和明细数据的思想来显示或总结“较低分辨率”的数据样本（以可能丢失重要细节为代价）。以下是一些 EDA 应用的例子：

- 与饼图相似，锯齿图（coxcomb）也切分一个圆。然而在饼图中扇形的角度不同；而在锯齿图中扇形的半径不同。弗洛伦斯·南丁格尔（Florence Nightingale）在伦敦及其附近的军事医院中使用这种图来显示死亡率（Nightingale，1858）。

11

- 1856年，John Bennett Lawes 在英国洛桑实验站（Rothamsted Experimental Station）⊖附近投资购置了很多土地。至今这些地带依然没有接触化肥和受其他人工方式的影响。这些区域提供了不同植物物种在不受影响的情况下进化和竞争的丰富数据。有些科学家用主分量分析（principal components analysis）来显示反映不同作物相对产量的数据（Digby and Kempton，1987，p.59）。
- 最近，Becker、Eick 和 Wilks（1995）描述了使用复杂的立体显示来可视化随时间变化的长途电话网络模式（超过12 000个连接）。

2. 描述建模（descriptive modeling）（第9章）：描述模型的目标是描述数据（或产生数据的过程）的所有特征。这样的例子包括为数据的总体概率分布建模（密度估计（density estimation））；把 p 维空间划分成组（聚类分析和区隔（cluster analysis and segmentation））；以及描述变量间的关系（依赖建模（dependency modeling））。例如在区隔分析中，目标是把相似的记录分成一组，比如商业数据库的市场区隔。这样做的目的是把记录分成均匀同质的（homogeneous）小组，以便使相似的人（如果记录是指人的）被分到同一组。这可以使广告商或销售者可以把他们的促销策略指向最可能响应的人群，以提高效率。这里分成的组数是由研究者决定的，没有对错之分。这与聚类分析不同，在聚类分析中目标是发现数据（例如科研数据库）中的“自然”群体。描述建模已经被应用到很多领域。

- 区隔已经被广泛而且成功地应用于市场营销领域，根据购买模式和年龄、收入等人口统计数据把客户分成均匀同质的小组（Wedel and Kamakura，1998）。
- 聚类分析已经被广泛应用于精神病研究领域以建立精神病的疾病分类。例如，Everitt，Gourlay and Kendell（1971）应用这种方法对住院的精神病患者进行采样，他们的报告指出（很多发现之一）“所有四种分析都产生了一个主要由精神压抑患者组成的聚类。”
- 聚类技术已经被用于分析地球北极大气层的长期气候变化。根据从1948年开始每天记录的数据来看，这种变化主要受三个循环的空间压力（recurring spatial pressure）模式（聚类）支配（进一步的讨论请参见 Cheng and Wallace（1993）和 Smyth，Ide，and Ghil（1999））。

12

3. 预测建模（predictive modeling）：**分类和回归**（第10章和第11章）：预测建模的目标是建立一个模型，这个模型允许我们根据已知的变量值来预测其他某个变量值。在分类中，被预测的变量是范畴型的，而在回归中被预测的变量是数量型（quantitative）的。这里“预测”这个词是取它的一般含义，根本不带有任何时间延续性的暗示。所以，我们可以预测将来某一天股票的市值，或预测哪一匹马会赢得比赛；我们也可以预测患者的病情，或焊接的牢固程度。在统计和机器学习中人们已经开发出了大量的方法来解决预测建模问题，而且这一领域的工作已经取得了重大理论进展，并加深了对深层推理问题的理解。预测和描述间的关键区别是预测的目标是唯一的变量（例如市值、疾病分类、牢固程度），而描述问题的模型中并不以任何单一的变量为中心。预测模型的例子如下：

- Fayyad，Djorgovski，and Weir（1996）的 SKICAT 系统使用树结构表示建立了一个分

⊖ 译注：世界最著名和最古老的的农作物和农业研究机构，建立于1843年，现为英国农作物研究所（Institute of Arable Crops Research）的主要部分（简称为 IACR-Rothamasted）。1842年 John Bennett Lawes 在英国获得过磷酸盐专利，1843年他和另一位农业化学家一起创立此试验站。

类树，这个分类树可以根据 40 维的特征向量分类星体和星系，并且做的和人类的专家一样好。这个系统常年被用来对天空数字图像中的上百万星体和星系进行自动分类。

- AT&T 的研究人员开发了一个系统，用来跟踪美国的所有 35 000 万个电话号码（Cortes and Pregibon，1998）。他们使用回归技术建立了一个模型，这个模型可以估计出一个电话号码位于商业机构还是居民住宅的概率。

4. 寻找模式和规则（第 13 章）：上面列出的三类任务都致力于建立模型。还有一些数据挖掘应用是致力于模式探测的。一个例子是欺诈探测，做法是寻找明显不同于其他点的数据点，并查出这些数据点所属的不同交易类型，然后通过探测这些包含特殊交易的空间区域来查出欺诈行为。另一个应用是在天文方面探测异常的星体或星系，目的是发现以前未知的对象。还有一个应用就是在交易数据库中发现频繁出现的商品组合（比如日常用品经常被一起购买）。这个问题已经吸引了很多数据挖掘者的注意力，而且已经采用基于关联规则（association rule）的算法技术来解决这样的问题。 13

这里的一个重要问题是如何决定哪个因素真正导致了异常行为，也就是统计学家们所说的孤立点检测（outlier detection）问题。在高维情况下，这会变得更加困难。背景领域的知识和人类的解释可能是最宝贵的。利用模式和规则发现技术的数据挖掘系统的实例包括：

- 美国的职业篮球比赛会常规性地提供每场比赛的详细记录，包括在什么时间谁以何种姿势投篮，谁得分，谁传球给谁等等。Bhandari etal.（1997）的超级侦察（Advanced Scout）系统从这些记录中搜索类似规则的模式，目的是发现职业教练可能注意不到的有用信息片段（例如，“当选手 X 在场上时选手 Y 的投篮准确率从 75% 下降到 30%”。）1997 年这个系统被美国的多家职业篮球队使用。
- 在美国，盗用蜂窝电话估计使电话行业每年损失几亿美元。Fawcett and Provost（1997）描述了一个应用，该应用通过规则学习算法从庞大的用户事务数据库中发现盗用行为的特征。根据报告，利用这种方法建立的系统比现有的手工检测欺诈方法精度更高。

5. 根据内容检索（第 14 章）：这种情况下，用户有一种感兴趣的模式并且希望在数据集中找到相似的模式。这种任务对于文本和图像数据集合应用最普遍。对于文本，模式可能是一系列关键字，用户希望在庞大的可能相关的文档集合中（例如网页）寻找相关的文档。对于图像，用户可能有一幅样本图像、一幅图像的草图、或一幅图像的描述，然后希望从庞大的图像集合中发现类似的图像。无论对于两种情况中的哪一种，相似性的定义都非常关键，但搜索策略的细节也很重要。 14

检索系统有很多大规模应用的例子，包括：

- 在网络中，检索方法被用来定位文档，就像 Brin and Page（1998）的 Google 系统（www.google.com）那样。Google 系统使用了被称为“PageRank”的数学方法来基于链接模式估计各个网页的相对重要性。
- IBM 的研究人员开发了一个称为 QBIC（“根据图像内容查询（Query by Image Content）”）的系统，这个系统允许用户使用交互的方式搜索庞大的图像数据库，支持以像颜色、纹理和相对位置信息这样的内容描述提出查询（Flickner et al.，1995）。

尽管上面的五种任务彼此间有明显的差异，但它们也有很多共同的特征。例如，很多任务都具有“任意两个数据向量间的相似性或者距离”的概念。还有一个共同点是评分函数的思想（用来评估一个模型或模式拟合数据的好坏程度），不过对于不同类型的任务，具体的

函数形式往往有很大的差异。另外很明显的一点是，不同的任务需要不同的模型和模式结构，就像不同种类的数据需要不同的结构一样。

1.5 数据挖掘算法的组件

前面一节中我们已经列出了数据挖掘所针对的基本任务类型。现在我们开始考虑该如何完成这些任务。我们认为，针对这些任务的数据挖掘算法具有以下四个基本组件：

1. **模型或模式结构**：决定要从数据中寻找的潜在结构或函数形式（第6章）。
2. **评分函数**：鉴定一个已拟合模型的质量（第7章）。
3. **优化和搜索方法**：优化评分函数并对不同的模型和模式结构进行搜索（第8章）。
4. **数据管理策略**：在搜索和优化期间高效地处理数据访问问题（第12章）。

我们已经讨论了模型和模式间的差异。在这一节的余下部分我们讨论数据挖掘算法的其
15 他三个组件。

1.5.1 评分函数

*评分函数*对一个模型或参数结构拟合给定数据集的效果进行量化。在理想情况下，最佳的评分函数应该精确地反映出特定预测模型的效果（也就是期望模型所带来的真正效益）。然而在实践中，往往难以精确地确定预测模型的真正效果。所以，经常使用简单的、“通用的”评分函数，比如最小平方以及分类精度。

如果没有某种形式的评分函数，我们就无法说出一个模型是否比另一个更好，或者到底如何为模型的参数选择一套好的参数值。为了实现这个目的，广泛使用了以下几种*评分函数*：似然（likelihood）、误差平方和以及错误分类率（后者用于有指导的分类问题）。例如著名的误差平方评分和函数是这样定义的：

$$\sum_{i=1}^{n}(y(i)-\hat{y}(i))^2 \tag{1.1}$$

其中 $y(i)$为被预测的 n 个目标值之一，$1 \leqslant i \leqslant n$，$\hat{y}(i)$为我们作出的预测值（通常它是关于供预测的其他“输入”变量值和模型参数的函数）。

值得注意的是，不仅要考虑不同评分函数理论上的合适性，还应该通过应用实践来检验它们。打个比方来说，最有可能拟合数据的模型可能是很理想的，但如果估计它的参数需要几个月的计算时间，那么它也就没有什么价值了。同样，特别容易受数据中的微小变化影响的评分函数也不可能有很高的价值（它的用途依赖于研究的目标）。举例来说，如果几个极端情况值会导致对某个模型参数的估计疯狂变化，那么一定要提高警惕；一个数据集通常是从大量的可能数据集中选取的，那么在其他数据集中这些极端情况值就有可能会有所不同。而使用对极端情况不敏感的鲁棒（robust）方法就可以避免这样的问题。

1.5.2 优化和搜索方法

评分函数衡量了提出的模型或模式多好地匹配了数据的各种特征。通常这些模型或模式
16 是以各种形式的结构来描述的，有时还带有未知的参数。优化和搜索的目标就是决定这些结构和参数值，以使评分函数达到最小值（或最大值，取决于具体情况）。发现模型中的最佳

参数值的任务通常被称为优化（或估计）问题。从庞大的潜在模式族中发现感兴趣的模式（比如规则）的任务通常被当作组合搜索问题，而且经常利用启发式搜索技术来实现这类问题。在线性回归中，经常通过最小化误差平方评分函数（模型的预测值与被预测变量的观测值之间的误差平方和）来发现预测规则。这样的评分函数易于进行各种数学操作，而且可以用代数方法得到使它最小化的模型。相反，像错误分类率（用于有指导分类）这样的评分函数就难以用解析方法来最小化。举例来说，因为它本质上是不连续的，那么强大的微积分方法就无法发挥作用。

当然，尽管我们可以使用一个评分函数使一个模型或模式很好的拟合数据，但在很多时候这不是真正的目标。正像上面所指出的，我们的目的经常是要泛化（generalize）到可能出现的新数据（新的客户，新的化学制品等等），而且过度拟合数据库中的数据可能降低对新案例的预测精度。在本章的后面我们将讨论这个问题。

1.5.3 数据管理策略

数据挖掘算法的最后一个组件是数据管理策略：存储、索引和访问数据的方式。统计和机器学习中的大多数著名数据分析算法都是假定可以在内存（RAM）中迅速高效地访问到所有数据个体。尽管主存储器技术迅速提高，但第二级（磁盘）和第三级（磁带）存储技术也在以相同的速度提高，因此很多海量数据集仍然主要被存储在磁盘或磁带上，现有的 RAM 是容纳不下的。所以访问海量数据集必然要付出一定开销，因为不可能使所有的数据一下子都可以被中央处理器访问到。

已经开发出的很多数据分析算法并没有明确地对数据管理策略作出说明。对于过去的相对较小的数据集来说这样做还可以，但是如果现在把很多算法（例如分类和回归树算法）的 17
传统版本直接应用到主要存储在第二级存储器中的数据上，性能往往变得很差。

数据库领域所关心的是开发索引方法、数据结构以及如何既高效又可靠地检索数据的查询算法。他们已经开发出了很多技术支持在庞大的数据集上相当简单地进行计数（聚合）操作，以生成报表。然而，最近几年来，人们已经开始开发支持“原语（primitive）”数据访问操作的技术，这是实现高效率数据挖掘算法（例如用于在高维空间中检索相邻点的树结构索引算法）所必需的。

1.6 统计和数据挖掘的相互关系

单纯的统计技术已经不足以解决某些日益复杂的数据挖掘问题，特别是那些涉及海量数据集的问题。然而统计在数据挖掘中承担着非常重要的角色：在任何数据挖掘项目中它都是一个必要的部分。这一节我们讨论一下传统统计和数据挖掘的相互关系。

对于庞大的数据集（特别是非常庞大的数据集），我们可能无法轻易知道数据中的规律，即使是非常显而易见的。对数据进行简单的目测不是办法。这意味着对于很大的数据集，我们需要周密完善的搜索和分析方法来弄清对于小数据集可以立刻得到的特征。此外，正如我们前面所讲到的，很多情况下数据挖掘的目标是要得到针对现有数据之外的某种推理。例如，在一个天体数据库中，我们可能想要得到这样一个结论“类似这个天体的所有对象的行为是这样的”，或许附带一个概率限制。类似地，我们可以断定一个国家的某个地区的电话呼叫呈现某种特定的模式。当然，需要我们作出论断的不可能是数据库中的某个呼叫，而是希望

能够预测将来呼叫的模式。数据库提供了用来建立模型或搜索模式的对象集合，但最终的目的一般不是描述这些数据。在大多数情况下目标是描述数据产生的一般过程，以及描述可能由同样的过程产生的其他数据集。所有这些都意味着有必要避免模型或模式与现有的数据匹
18 配得太紧密：要知道现有的数据集仅仅是可能数据中的一部分，所以我们不希望模型与现有数据的特异性太接近。换句话来讲，就是必须避免过度拟合（overfitting）给定的数据，而是要发现可以很好地泛化到潜在将来数据的模型或模式。在选取用来选择模型或模式的评分函数时应该考虑这一点。在第 7 章和第 9 章到第 11 章我们将更详细地讨论这个问题。虽然我们是从数据挖掘角度讨论这个问题，但是对于统计这个问题也是很重要的；甚至一些人把它当作是统计学科的一个定义特征。

既然统计思想和方法对于数据挖掘如此重要，那么就有一个很自然的问题是这两者之间到底有什么差异。数据挖掘就是针对非常庞大数据集的探索性统计，还是除了探索性数据分析外还有更多的内容？回答是肯定的——数据挖掘有更多的内容。

经典的统计应用和数据挖掘的基本差异是数据集的大小。对于一个传统的统计学家，一个“大”的数据集可能包含几百或几千个数据点。然而对于致力于数据挖掘的人来讲几百万甚至几十亿的数据点并不意外——GB 甚至 TB 数量级的数据库也不少见。生活中很多地方都有这样的大数据库。例如，美国的零售商沃尔玛每天完成 2 千万笔交易（Babcock，1994），1998 年形成了一个 11TB 的客户交易数据库（Piatetsky-Shapiro，1999）。AT&T 有 1 亿个客户，它的长途网每天有 3 亿次的呼叫。每次呼叫的特征被更新到一个数据库，用以建立美国所有电话号码的模型（Cortes and Pregibon，1998）。Harrison（1993）报道说美孚石油公司（Mobil Oil）打算要存储超过 100TB 的有关石油探测的数据。Fayyad，Djorgovski，and Weir（1996）描述的“帕洛马天文台数字化天体调查（Digital Palomar Observatory Sky Survey）”中涉及 3TB 的数据。正在进行的 Sloan 天体数字化调查将产生大约 40T 字节数据，最终要缩减为含有 400GB 的包含 3×10^8 个天体的目录（Szalay et al.，1999）。美国国家航空和宇宙航行局（NASA）的地球观测系统设计为每小时产生几个 GB 的原始数据（Fayyad，Piatetsky-Shapiro and Smyth，1996）。人类基因工程要完成整个人体基因的测序可能要产生超过 3.3×10^9 个核苷酸的数据集（Salzberg，1999）。这样大容量的数据集带来了统计学家使用传统方法无法处理的一些问题。

19 可以通过采样来简化海量数据集（如果目标是建立模型是可以的，但是如果目标是模式探测就不合适了），也可以使用可适应方法（adaptive），或者用*充分统计量*（sufficient statistics）来总结记录。例如，在标准的最小平方回归问题中，我们可以用所有记录的和、平方和以及乘积的和来代替针对每个变量的大量评分——这样就足以计算出回归系数，而不管有多少条记录。随着记录或变量数量的上升，考虑以计算时间表示算法规模的变化是很重要的。例如，搜索最佳变量子集（根据某个评分函数）的穷举方法仅在一定限度内是可行的。如果有 p 个变量，那么就要考虑 2^p-1 个可能的变量子集。对前一节提到的高效搜索方法来说如何放宽这个限制是至关重要的。

当有很多变量时会产生更多的困难。很重要的问题之一是*维度效应*（curse of dimensionality）：空间中单元格（unit cell）的数量随着变量个数的上升按指数增长。例如，考虑一个二进制变量，要得到对两个单元格的合理估计精度我们可能希望对每个单元格有 10 个观测，那么共有 20 个。如果有两个二进制变量（也就是四个单元格），那么就需要 40 个观测。如果有 10 个二进制变量，那么就需要 10240 个观测，要是 20 个变量就是 10485760 个

了。维度过高的恶果是陷入如下困境：在高维空间中如果没有天文数字大小的数据库（事实上，需要的数据量非常大，以致于在这样的数据挖掘应用中 GB 量级的数据也显得苍白无力）就无法找到概率密度的精确估计。在高维空间中，相邻点可能离得很远。这不仅仅是操纵其中的大量变量的困难，而且关系到能否实现目标。在这种情况下，有必要在预先选取模型时增加一些额外的约束（例如，假定为线性模型）。

访问庞大的数据集会产生很多问题。统计学家们传统上理解的“平面”数据文件——行表示对象；列表式变量——可能和数据的实际存储方式大不相同（比如前面描述的文本和网络交易数据集合）。在很多种情况下，数据是分布存储在多台计算机上的。从这种分散的数据中获得一个随机样本不再是一件微不足道的事。如何定义采样框架以及访问数据需要多长时间都是很重要的问题。 20

还有更糟糕的是很多时候数据集是不停变化的——举例来说，就像电话呼叫记录或用电记录那样。分布的或者不断变化的数据可能成倍地增加数据集的大小并改变需要解决的问题的属性。

除了数据集的大小可能导致很多困难外，标准统计应用中不经常遇到的其他问题也可能如此。我们已经指出数据挖掘通常是数据分析的次级过程，也就是说数据本来是为了其他目的而收集的。相反，很多统计工作是本位分析（primary analysis）：带着特定的问题采集数据，然后分析数据回答这个问题。统计学中甚至包括试验设计和调查设计这样的子学科——整个领域的专家都致力于寻找最好的方式采集数据以回答特定的问题。当数据被用于搜集数据的本来目的之外的问题时，这些数据可能不能理想地适合这些问题。有时数据集是整个总体（例如，一类化学品中的所有化学品），所以标准统计中的推理思想已不适用了。即使数据集不是整个群体，也经常是顺便的（convenience）或机会的（opportunity）样本，而不是随机样本。（例如，问题中的记录很可能是因为它们最容易被测量或覆盖一个特定时期而被收集起来的。）

除了数据采集方式导致的问题，还有发生在庞大数据集中的失真问题——包括残缺值、污染和数据损坏。很少有哪个数据集不存在这些问题。以至于一些周密的建模方法在模型中包括一个部分来描述处理残缺值或数据失真问题的机制。也可以使用像 EM 算法（在第 8 章中讨论）这样的估计方法或者插补（imputation）方法来产生与可能使用的残缺值具有同样分布属性的人工模拟数据。当然这些问题在标准的统计应用中也存在（尽管对于小的、特别搜集的数据集来说这些问题的严重程度会小很多），但基本的统计教材倾向于掩饰它们。

概括地讲，尽管数据挖掘确实与标准统计中的探索性数据分析技术有相当大的重叠，但数据挖掘面临着很多新的问题，这主要是涉及的数据集大小和数据集的新属性所导致的。 21

1.7 数据挖掘：打捞、探查还是垂钓

作为数据挖掘这本书的绪论，如果不介绍一下历史上曾使用的对数据挖掘的称呼，那么就不完整了。这些称呼包括“数据挖掘（data mining）”、“打捞（dredging）”、“探查（snooping）”和“垂钓（fishing）”。在 20 世纪 60 年代，随着计算机不断地应用到数据分析领域，人们注意到，只要你搜索的时间足够长，就总能发现好的拟合数据集的某个模型。有两个因素对这个过程起作用：模型的复杂度和可能模型集合的大小。

不难理解，如果我们采用的模型灵活度足够高（相对于现有数据集的大小），那么我们很可能可以做到任意好（arbitrarily well）的拟合现有数据。然而，正像前面所指出的，我们的目标可能是泛化到现有数据之外，一个很好拟合现有数据的模型可能对泛化这个目的而言

并不理想。而且，即使目标就是拟合现有数据（例如，当我们希望产生一个描述完整群体的数据的最精确总结），那么通常更倾向于选择简单的模型来做到这一点。极端地讲，与原始数据复杂度等价的模型当然完美地拟合它自己，但这几乎没有任何意义和价值。

即使使用一种相当简单的模型结构，如果我们考虑具有这种基本结构的足够多的不同模型，那么我们也可以期望发现很好的拟合。例如，考虑从预报变量 X 来预测一个响应变量 Y，X 是从可供选择的非常庞大的变量集合 X_1，…，X_p 中选取的，且它们都与 Y 无关。由于数据产生过程的随机变化作用，尽管在 Y 和任意变量 X 之间不存在潜在的联系，但在现有的数据中仍会显示出某种关系。接下来搜索过程便会发现变量 X 和 Y 之间有紧密的联系。结果，庞大的搜索空间导致发现了本来不存在的虚假模式。当变量 X 的潜在可能个数 p 非常庞大而且样本尺寸 n 很小时这种情况尤其严重。这类错误的更熟悉的例子还有媒体中流行的虚假相关推论，比如“发现”在过去 30 年中，当美国橄榄球超级杯赛的冠军来自某个联盟时，一种股票指数在下一个月就会上涨。在很多领域都有大量的相似例子，比如像经济和社会科
22 学这些数据一般相对稀疏但匹配数据的模型或理论相对充足的领域。例如，在经济领域的时间序列预测中，可能仅有较短时间跨度的历史数据，但却有大量的经济指标（潜在的预测变量）。Leinweber 提供了这种类型预测的一个特别幽默的例子，他得到了对著名的标准普尔 500 种股票指数（Standard and Poor 500）年值几乎完美的预测，方法是把这个指数的年值定义为前一年孟加拉和美国的黄油产量、干酪产量和绵羊数量年值的函数。

这种“发现”的危险是为统计学家们所熟知的，过去他们把这种泛泛的搜索称为“数据挖掘”或“数据打捞”——使用这些术语来表示贬低的内涵。当数据集很庞大时，如果分析的潜在结构空间也足够大，那么这种问题的严重性会降低，不过即使这样仍有危险存在。在模式探测中这种风险比在模型拟合中更大，因为根据定义，模式涉及相对更少的实例（也就是样本尺寸很小）。因为如果我们为了搜索仅有 50 个点的异常结构而分析了 10 亿个数据点，那么我们很有可能探测到这个结构。

不存在可以解决这个问题的简单技术，尽管已经研究出了很多种策略，包括把数据分成子样本，然后使用一个部分来建立模型或探测模式，再用另一部分来验证。在后面的章节中我们将更多地讲述这样的方法。然而最终的答案是不要把数据挖掘当作脱离数据内涵的简单技术来运用。任何有潜力的模型或模式都该呈现给数据拥有者，让他们来评估它的有趣度、价值、有用性以及它的潜在真实性。

1.8 本章归纳

由于计算机和数据采集技术的进步，我们已经积累了而且还正在积累包含 G 字节或者甚至是 T 字节的庞大数据集合。这些堆积如山的数据包含了可能很有价值的信息。问题是如何把这些有价值的信息从包围它的大量枯燥的数字中提取出来，从而使数据拥有者可以从中取得收益。数据挖掘是一门新兴的学科，它所要做的就是：通过筛分这些数据库，对它们进行总结，并寻找其中的模式。

不应该把数据挖掘看作是简单的一次性操作。对于巨大的数据集合来说，考察和分析它
23 的方式是没有止境的。随着时间的推进，新的结构和模式类型可能引起我们的兴趣，并值得在数据中寻找它们。

数据挖掘已经受到了广泛的瞩目，这有很多原因：它是一门新的技术；针对新的问题；

对于寻找商业和科研中的有价值发现有很大的潜力。然而，我们不应该期望它可以回答所有的问题。就像所有的发现过程一样，数据挖掘的成功具有幸运（serendipity）的因素。尽管数据挖掘提供了有用的工具，但这并不意味着必然可以得到重要、有趣、而又有价值的结果。所以我们必须警惕对可能的成果的过分地夸大。不过潜力是有的。

1.9 补充读物

以下文献对数据挖掘作了简单扼要的介绍：Fayyad, Piatetsky-Shapiro and Smyth（1996）；Glymour et al.（1997）以及《ACM（美国计算机学会）通讯》的 Vol. 39, No. 11 特刊。Adriaans and Zantige（1996）以及 Weiss and Indurkhya（1998）总结了数据挖掘中有关预测的一些问题。Witten and Franke（2000）从机器学习（人工智能）的角度讨论了数据挖掘，该书面向应用，可读性非常强；Han and Kamber（2000）是从数据库角度编写的一本很容易理解的数据挖掘教材。针对商业用户的数据挖掘书籍非常多，特别值得一提的是 Berry and Linoff（1997，2000），其中对一些有潜力的数据挖掘商业应用提出了的很多实践性很强的建议。

Leamer（1978）广泛地讨论了数据打捞的危险，Lovell（1983）就这一主题发表了评论。Hendry（1995，15.1 节）从统计的角度给出了计量经济学家对数据挖掘的看法。Hand et al.（2000）以及 Smyth（2000）对数据挖掘和统计作了比较性讨论。Casti（1990，192-193 页和 439 页）简要地讨论了巧合性。

24

第 2 章　测量和数据

2.1　简介

我们的目标是发现存在于“真实世界”中的各种关系，这可能是物理世界、商业世界、科学世界、也可能是其他某个概念上的领域。然而在探索这样的关系时，我们并不要走出去直接观察这个领域，而是通过描述它的数据来进行研究。所以，首先我们需要明确数据的含义。

数据是通过把感兴趣领域里的实体以某种测量过程映射到符号表示得到的，测量就是把实体的一个给定属性与一个变量值联系起来。对象间的关系是通过变量间的数值关系表示的。这些数值表示——也就是数据项——是以数据集的形式存储的；这些数据项就是我们的数据挖掘活动的题材。

透彻理解测量过程是至关重要的。它是接下来所有数据分析和数据挖掘活动的基础。我们将在 2.2 节详细地讨论这个过程。

我们在第 1 章中提到，两个对象间距离的概念是很重要的。2.3 节突出讨论了两个对象间的距离尺度——基于对这些对象的测量向量。测量的原始结果可能适合也可能不适合直接用作数据挖掘，2.4 节简要地讨论了分析前如何转化数据。

我们已经指出，我们不希望数据挖掘活动所发现的关系就是对采集到数据的生搬硬套（artifact）。同样，我们也不希望我们的发现就是数据的定义属性：比如发现具有同样姓氏的人经常生活在同一个家庭算不上什么成就。2.5 节中我们简要介绍了数据图式（schema）——数据中预先存在的结构——的思想。

没有完美的数据集，庞大的数据集更是如此。测量误差、数据残缺、采样失真、人为错误以及其他一大堆因素都可能损坏数据。既然数据挖掘是致力于探测数据中的未知模式，那么警惕这些缺陷是非常重要的——我们要得到的结论不该是建立在那些反映了数据搜集或录制过程中的瑕疵的模式。2.6 节针对记录（或实例）和单个字段（或变量）的测量讨论了数据质量问题。2.7 节讨论了这些个体的集合（也就是样本）的总体质量。

2.8 节归纳了本章的要点，2.9 节推荐了一些更详细的读物。

2.2　测量类型

可以按照很多种方式来对测量分类。一些分类标准是从被测量属性的特征而来的，还有一些是根据测量的用途而来的。

为了阐明这一点，我们先考虑如何度量 WEIGHT 属性。在这个讨论中，我们使用大写字母代表属性，用小写字母代表它对应的变量（映射到测量操作所产生的数字的结果）。即测量 WEIGHT 得到 **weight** 的值。为了更加具体，设想我们有一堆石头。

首先我们可以根据 WEIGHT 属性来排列石头。例如我们可以这么做，在天平的每一个托盘上放一块石头，观察天平倾斜的方式。根据这个过程，我们可以赋给每一块石头一个数字，并使较重的石头对应较大的数字。注意这里这些数字只代表序号。一块石头被赋予数字

4 另一块石头被赋予数字 2 并不代表第一个就总是第二个的两倍重。我们完全可以选取某个其他数字表示第一块石头的 WEIGHT，只要它大于 2。通常，我们可以使用单调（保持顺序）变化的任意数字集合，它们都是等价的合理赋值。我们仅关心石头按 WEIGHT 属性的排列
26 顺序。

我们可以进一步探讨前面的例子。假定我们发现，当我们放一块大的石头在天平的一个托盘上，放两块石头在另一个托盘上使天平平衡了。从某个角度来讲两个小石头的 WEIGHT 属性合起来等价于一个大石头的 WEIGHT 属性。这说明（很自然地得出）我们可以用这种方式赋一个数字给石头，也就是说赋给石头的序号数字不仅对应于天平观测到的序号，同时使赋给两块小石头的数字之和与赋给大石头的数字相等。也就是说这两块较小石头的重量等于这块较大石头的重量。假定我们赋给较小的两块石头的数字是 2 和 3，而且赋给较大石头的数字是 5。这套赋值满足了顺序要求和属性可加性的要求，但如果分别赋值为 4、6 和 10 也可以做到这一点。因此如何定义对应属性 WEIGHT 的变量 **weight** 还存在一些自由度。

这个例子说明了我们的数字表示反映了我们所研究系统的试验（empirical）属性。按 WEIGHT 属性表示的石头之间的关系与测量到的变量 **weight** 的值之间的关系对应。这个表示的价值在于它允许我们通过这个数字系统来研究对应的物理系统。不必把一袋袋的石头弄来弄去就可以看到哪一袋中含有最大的石头，哪一袋的石头平均重量最重，等等。

石头的例子包括了两种实验关系：石头的次序，这是根据天平如何倾斜来决定的；以及它们的结合（concatenation）属性——两块石头一起与第三块石头平衡。其他的实验系统可能包括少于或多于两个的实验关系。次序关系是最普遍的，通常，如果一个实验系统仅有一个关系，那就是次序关系。次序关系的例子还有医学中的 SEVERITY（严重性）属性和心理学中的 PREFERENCE（爱好）属性。

当然，一些属性甚至没有次序关系，例如属性 HAIR COLOR（发色），RELIGION（宗教信仰），和 RESIDENCE OF PROGRAMMER（程序员的住所）都没有自然的次序。但也可以用数字来表示这些属性的“值”，例如（blond = 1，black = 2，brown = 3 等等），但是被表示的实验关系仅代表颜色的不同（因而被表示为不同的数字）。这里可能更加明显地看到
27 也可以使用其他的数字集合。只要是不同数字对应不同属性值的数字集合就可以。

既然数字赋值是不唯一的，我们就必须找到某种方法来限制这种自由性——否则如果不同的研究者使用不同的赋值就可能产生问题。解决的办法是采用某种约定。对于石头的例子，我们采用 WEIGHT 属性的一个“基本”值，对应于变量 **weight** 的一个基本值，并且根据需要多少个基本值的拷贝平衡被测量对象来定义测量值。例如可以使用克和磅作为 WEIGHT/**weight** 系统的基本值。

可以按照测量中所寻求的实验关系来分类测量类型。然而很重要的一种其他分类方法是按照测量结果所支持的变换方式来分类，也就是可以使用哪种（或哪些种）变换来产生其他等价的合理数字表示。例如，对于一个数字的严重性标度，因为它仅代表它所处的次序，所以可以使用任意保持此次序的数字来等价地表示它——通过单调或依次地转化原来的次序推导出新的数字。由于这个原因，这样的标度被称为顺序标度（ordinal scale）。

在石头的例子中，唯一的合法变换是乘一个常数（例如把磅转换为克）。任何其他的变换（对数字取平方，加一个常数等）都会破坏数字所表示的次序或它所具备的相加结合性。（当然，其他的变换可能使实验关系可以用其他数学运算来表示。例如，如果我们把石头例子中的值 2、3 和 5 变换为 e^2、e^3 和 e^5，那么我们可以用乘法来表示这个实验关系 $e^2e^3 = e^5$。

然而加法是最基本的运算，所以是被优先选择的。）因为使用一个常数来乘这种类型的标度仍然保持测量值原来的比例，所以这种标度被称为比例标度（ratio scale）。

在我们前面的另一个例子中（头发颜色的例子）任何变换都是合法的，只要保持每个唯一标识是用不同的数字表示的——不关心两个数哪一个大一些，而且属性的相加是没有意义的。更简单地讲，这里的数字就是用作标签或名字；因此这样的标度被称为标称标度（nominal scale）。

对应于不同的合理（或可接受的）变换方式，还存在其他的标度类型。其中之一是区间标度（interval scale）。这里的合理变换方式是允许对测量单位乘一个常数，再加一个任意常数。因此不仅测量单位是任意的，而且原点也是任意的。这种标度的经典例子是传统的温度 28
测量（华氏，摄氏等等）和日历时间。

理解不同种类测量标度的基础是很重要的，因为只有这样我们才能保证数据挖掘操作中发现的模式是名副其实的。为了举例说明这一重要性，假定有两个小组，每个小组有三个患者，现在用从 1（不痛）到 10（剧痛）的顺序标度来记录他们的疼痛情况。一个小组的三个患者的结果是 1，2 和 6；另一个小组的结果是 3，4 和 5。前三个人的平均值是（1 + 2 + 6）/3 = 3，而另三个人的平均值是 4。第二个小组的平均值较大。然而由于标度是纯顺序的，所以保持顺序的任意变换都会得到等价的合理数字表示。例如，可以变换标度使它的范围变为 1 到 20，把（1，2，3，4，5，6）变换为（1，2，3，4，5，12）将仍保持着不同等级疼痛间的顺序关系——如果使用第一套标度患者 A 比患者 B 疼痛得更加厉害，那么使用第二套标度患者 A 也还比患者 B 疼痛得更加厉害。然而现在第一组患者的平均结果是（1 + 2 + 12）/3 = 5，而第二组的平均值还是 4。这样，两个等价的合理数字表示就导致了相矛盾的结论。使用第一套标度观测到的模式（一组的均值大于另一组）是对所采用的数值表示的生搬硬套，并不与对象间任何真实的关系所对应（如果它反映了真实的关系，那么两个等价的合理表示不会得出相反的结论）。为了避免这个问题，我们必须保证仅在测量标度合理变换时真实值保持不变的情况下作出统计结论。在这个例子中，我们可以得出这样的结论，对第二组评价（score）的中值（median）大于对第一组的评价的中值；无论我们应用什么样的保持顺序的变换这都是成立的。

直到这里，我们一直集中讨论的是映射意义上的测量，在这种映射中，被研究的实验系统中的数字间关系对应于数字系统中的数字间关系。因为这种映射是用来表示实验系统中的关系，所以这种类型的测量被称为*表示性的*（representational）。

然而并非所有的测量过程都可以很容易地被纳入这个框架。在某些情况下，更自然的是会把测量过程当作定义问题中的属性，并赋一个数字给这个属性。例如，医学中的 QUALITY OF LIFE（生活质量）属性经常是这样衡量的：标识出人类生活中那些被认为很重要的部分，然后定义一种方法用来把对应于每一部分的分数合并起来（例如，加权求和）。软件工程中 29
的 EFFORT 有时也是以相似的方式定义的：把程序指令的数量、复杂度等级、内部和外部文档的数量等尺度联合起来。同时定义并测量一个属性的测量过程被称为*操作性的*（operational）或*非表示性的*（nonrepresentational）过程。关于测量的操作性观点是在物理学中形成的，时间大约在 20 世纪初，当时物理学界正处于对像原子概念这样的事实的不安之中。今天，这种方法在社会和行为科学中具有了更大的实践内涵。因为在这种方法中测量过程同时定义了属性，所以避免了合理变换所产生的问题。既然不存在其他可选的数字表示，

那么任何统计结论都是可容许的。

例 2.1 Halstead（1997）给出了一种测量编程工作量的早期尝试。在一个给定的程序中，如果 a 是独立的运算符的数量，b 是独立的操作数的数量，n 是全部运算符的数量，m 是全部操作数的数量，那么编写这个程序的工作量是：

$$e = am(n + m)\log(a + b)/2b$$

这是一个非表示性的测量，因为它既定义了编程工作量这个属性，又提供了一种测量它的方法。

一种描述表示性测量和操作性测量之间差异的方法是前者侧重于理解系统中发生什么，而后者侧重于预测发生什么。在本书的其他很多地方都提到了理解（或描述）一个系统与预测一个系统的行为间的差异。当然，这两个目标是相互重叠的，但是知道它们间的差异是有价值的。我们可以不必关心隐藏在测量过程底层的具体机制便构建出有效的而且有价值的预测系统。例如很多人成功地驾驶汽车或操作影碟机，但他们并不知道这些设备内部的任何工作过程。

原则上，不论是测量的表示性方法所定义的映射，还是操作性方法所赋予的数字都可能从一个连续区域取任何值。例如，一个映射可以告诉我们单位正方形的对角线长度是 2 的平方根。然而实践中记录的数据仅是这个数学理想值的近似。首先，在测量中经常存在
30 不可避免的误差（例如，如果你反复地测量某个人的身高，精确到毫米，那么你看到的是一个很多值的分布）。第二，数据总是记录到有限的小数位。我们可以记录单位正方形的对角线长度为 1.4，或 1.41，或 1.414，或 1.4142，等等，但这个测量值总是不精确的。偶尔这种近似会对分析造成影响。当近似太粗糙时（记录数据的小数位数太少），这种影响是非常明显的。

上面的讨论对测量问题提供了一个理论基础。然而，这并未覆盖已经提出的所有测量术语。有人已经提出了很多其他对测量标度分类的方法，有时并不是基于标度的抽象数学属性，而是基于用来操纵它们的数据分析技术的种类。这样的备选方法包括计数和测量；标称的、顺序的和数字的（numerical）标度；定性和定量测量；范畴型（categorical）测量和标距型测量（metrical）；分级（grade）、排位（rank）、计份额（counted fraction）、计数（count）、计量（amount）和计余额（balance）。多数情况下这些术语的含义是很清楚的。例如，排位就是以问题中属性的相对“大小”为基础赋给集合中的某些特定实体一个整数：排位是保持顺序属性的整数。

在数据挖掘应用中（也在本书中），最常见的标度类型有：允许任何一对一变换的范畴型标度（标称标度），有序的范畴型标度，以及数字（定量的或实数值的）标度。

2.3 距离尺度

很多数据挖掘技术（例如最近邻分类（nearest neighbor classification）方法、聚集分析、多维缩放（multidimensional scaling）方法）都是基于对象间的相似性尺度。主要有两种方法得到相似性尺度。第一种是直接从对象获得。例如市场调查可以让受访者根据成对对象间的相似性来鉴定它们的等级，或者在食品品尝试验中可以在各种滋味的冰激淋间说出相似性。

第二种方法是可以根据描述每一个对象的测量或特征向量，间接地得到相似性尺度。在第二种情况中有必要定义“相似”的含义，以便计算正式的相似尺度。

与其谈论两个对象间如何相似，还不如谈论它们如何不相似。一旦我们有了“相似”或 31
“不相似”中任一个的正式定义，那么我们就可以简单地通过一个适当的单调递减变换来定义另一个。例如，如果 $s(i, j)$ 表示对象 i 和 j 间的相似性，$d(i, j)$ 表示相异性，那么可行的变换包括 $d(i, j) = 1 - s(i, j)$ 和 $d(i, j) = \sqrt{2(1 - s(i, j))}$。术语“邻近度（proximity）”经常用作既可以表示相似性又可以表示相异性的一个通用提法。

关于相似性的另外两个经常使用的术语是距离（distance）和标距（metric）⊖。术语距离经常用来指非正式的相异性尺度，它是从描述对象的特征推导出的，比如下面定义的欧氏距离（Euclidean distance）。另一方面，标距是满足以下三个条件的不相似尺度：

1. 对于所有的 i 和 j，$d(i,j) \geq 0$，并且当且仅当 $i = j$ 时 $d(i,j)=0$；
2. 对于所有的 i 和 j，$d(i,j) = d(j,i)$；
3. 对于所有的 i，j 和 k，$d(i,j) \leq d(i,k) + d(k,j)$。

上面的第三个条件被称为三角不等式。

假定我们有 n 个数据对象，每个对象有 p 个实数的测量值。我们用以下方法表示第 i 个对象的观测向量，$\mathbf{x}(i) = (x_1(i), x_2(i), \cdots, x_p(i))$，$1 \leq i \leq n$，其中 $x_k(i)$ 是第 i 个对象的第 k 个变量。那么第 i 个对象和第 j 个对象间的欧氏距离（Euclidean distance）被定义为：

$$d_E(i,j) = \left(\sum_{k=1}^{p} (x_k(i) - x_k(j))^2 \right)^{\frac{1}{2}} \tag{2.1}$$

这个尺度是以变量间的一定的公度性（commensurability）为前提的。因此如果每个变量都是使用统一单位测量的长度（当维数 p 是 2 或 3 时，这个公式得到的是我们使用的标准物理距离尺度）或者重量，那么这个公式也是有效的。如果变量不是使用统一标准测量的，那么这个公式就没什么意义了。例如，如果一个变量是 **length**，另一个变量是 **weight**，那么没有明显的办法来选择单位；如果改变单位的选择，那么就我们所关心的距离而言的最重要变量就可能不是原来的那一个了。

既然我们经常要处理不是在同一公度下测量的变量，我们就必须找到某个办法来克服选择单位的任意性。一种普遍的策略是使数据标准化，即用样本的标准差除以每一个变量，以 32
便使所有变量都可以被看作具有同等的重要性。（但注意这又产生了一个问题——“把每个变量看作具有同等的重要性”还是做了一个很武断的假定。）第 k 个变量 X_k 的标准差可以通过下式来估计：

$$\hat{\sigma}_k = \left(\frac{1}{n} \sum_{i=1} (x_k(i) - \mu_k)^2 \right)^{\frac{1}{2}} \tag{2.2}$$

其中 μ_k 是 X_k 的均值，可以用样本均值（sample mean）$\bar{x}_k = \frac{1}{n} \sum_{i=1}^{n} x_k(i)$ 来估计 μ_k（如果其未知）。于是 $x_k' = x_k / \hat{\sigma}_k$ 通过 $\hat{\sigma}_k$ 消除了标度的影响。

⊖ 译注：目前这个单词的中文译法很多，比如“尺度”、“量度”、“测度”、“跳数”等等，本书中将其译为“标距”。

此外，如果我们希望区别每一个变量的相对重要性，那么我们可以对它们进行加权（标准化之后），于是便得到了加权的欧氏距离尺度：

$$d_{WE}(i,j)=\left(\sum_{k=1}^{p} w_k(x_k(i)-x_k(j))^2\right)^{\frac{1}{2}} \tag{2.3}$$

从变量独立地对距离尺度起作用这个角度来看，欧氏和加权欧氏距离都是加成的（additive）。这个特征有时是不合适的。考虑一个极端的情况，设想我们量很多杯子的高度和直径。利用同量纲（commensurate）单位，我们可以使用这两个测量来定义杯子间的相似性。现在假定我们测量每个杯子的高度 100 次，直径 1 次（那么对于任一个杯子我们有 101 个变量，其中 100 个几乎是相同的值）。如果我们把这些测量代入标准欧氏距离公式中计算，那么高度会支配杯子间的相似性。然而，99 个高度测量对于我们真正需要的尺度没有任何贡献；它们与第一个高度测量是高度相关的（事实上除了测量误差外是完全相关的）。为了消除这种冗余我们需要一种数据驱动（data-driven）方法。一种方法是使数据标准化，不仅是像加权欧氏距离中的那样在每个变量的方向上标准化；而且还考虑变量间的协方差（covariances）。

33

例 2.2　考虑两个变量 X 和 Y，并且假定我们有 n 个对象，变量 X 对这 n 个对象的取值是 $x(1)$，…，$x(n)$，变量 Y 的取值是 $y(1)$，…，$y(n)$。那么 X 和 Y 间的**样本协方差**（sample covariance）被定义为：

$$Cov(X,\ Y)=\frac{1}{n}\sum_{i=1}^{n}\ (x(i)-\bar{x})\,(y(i)-\bar{y}) \tag{2.4}$$

其中 $\bar{x}$ 是 X 的样本均值，$\bar{y}$ 是 Y 的样本均值。

协方差是衡量 X 和 Y 如何一起变化的尺度：如果 X 中的较大值趋向于和 Y 中的较大值关联而且 X 中的较小值和 Y 中较小值趋向关联，那么协方差是一个大正值。如果 X 中的较大值趋向于和 Y 中的较小值关联，那么协方差将是一个负值。

更一般地讲，对于 p 个变量，我们可以建立 $p\times p$ 的协方差矩阵，其中元素（k，l）是第 k 个和第 l 个变量间的协方差。从前面的协方差定义中我们可以看出这样的一个矩阵（协方差矩阵）一定是对称的。

协方差的值依赖于 X 和 Y 的范围。可以通过标准化方法来消除这种依赖性，用 X 值的标准偏差除以 X 值，用 Y 值的标准偏差除以 Y 的值。得到的结果是 X 和 Y 间的**样本相关系数**（sample correlation coefficient）$\rho(X, Y)$：

$$\rho(X,Y)=\frac{\sum_{i=1}^{n}(x(i)-\bar{x})(y(i)-\bar{y})}{\left(\sum_{i=1}^{n}(x(i)-\bar{x})^2\sum_{i=1}^{n}(y(i)-\bar{y})^2\right)^{\frac{1}{2}}} \tag{2.5}$$

如果有 p 个变量，那么可以用与建立协方差矩阵相同的方式建立一个 $p\times p$ 的相关矩阵（correlation matrix）。图 2-1 显示了一个 11 维数据集的相关矩阵的像素图像，其中的数据是波士顿的各个不同郊区和居住有关的变量。从这个矩阵我们可以

清楚地看出不同变量是如何相关的。例如，变量 3 和 4（与商业区面积和氮氧化物浓度有关）都与变量 2（郊区内大居民点的比重）高度负相关；而且变量 3 和 4 相互间是正相关的。变量 5（家庭平均房间数）和变量 11（客户中数（median home value））是正相关的（也就是房子越大往往越富有）。变量 8 和 9（动产纳税比率和高速公路可达性）也是高度正相关的。

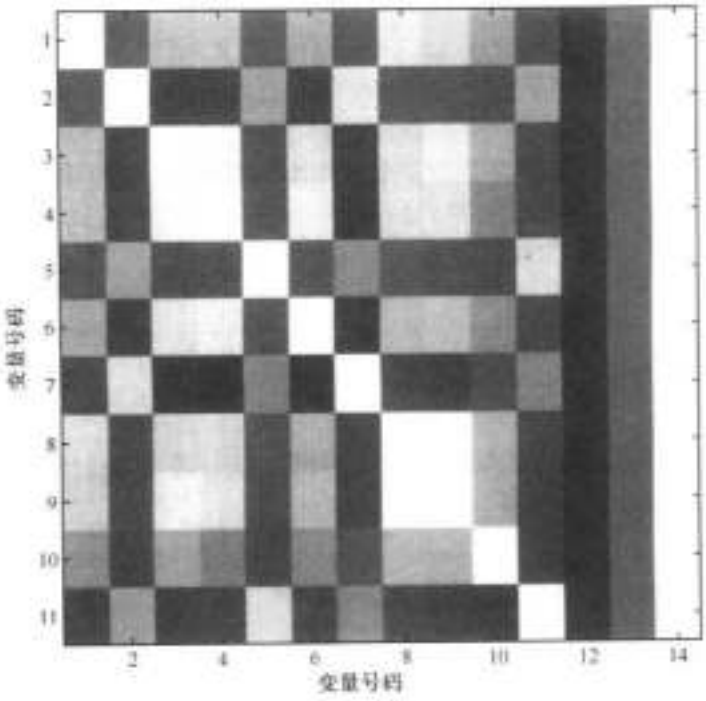


图2-1 一个显示为像素图像的样本相关矩阵。白色对应+1，黑色对应-1。最右边的三列分别包含-1，0和+1以提供一个像素亮度的参考。其余的11×11个像素表示11×11的相关矩阵。数据来自于回归研究文献中的一个著名数据集，其中每一个数据向量是波士顿的一个郊区，每个变量表示郊区的某一普通指标。变量名是（1）按人口计算的平均犯罪率，（2）大居住区的面积比例，（3）非零售商业面积的比例，（4）氮氧化物浓度，（5）每个住所（perdwelling）的平均房间数，（6）1940年前的家庭所占比例，（7）到零售中心的距离，（8）高速公路的可达性，（9）动产纳税比率（property tax rate），（10）儿童受教育率，（11）业主未失业家庭财产的中数

注意协方差和相关性体现了变量间的**线性依赖性**（linear dependency）（更精确地讲它们应被称为**线性协方差**和**线性相关**）。考虑在二维空间（X 和 Y）中以圆心为中心均匀分布的数据点。显然这两个变量是**依赖的**（dependent），但是是以非线性的方式依赖的，所以它们是 0 线性相关。因此独立意味着不相关，但反过来不总是正确的。在第 4 章我们将更多地探讨独立性。

34
/
35

再回忆一下前面关于咖啡杯的例子，其中有 100 个高度测量值和一个直径测量值。我们可以通过把我们的距离定义融入到协方差矩阵而削减 100 个相关变量的影响。这就是两个 p 维测量值 $\mathbf{x}(i)$和 $\mathbf{x}(j)$间的马氏（Mahalanobis）距离，具体定义为：

$$d_{MH}(i,j)=\left((\mathbf{x}(i)-\mathbf{x}(j))^T\Sigma^{-1}(\mathbf{x}(i)-\mathbf{x}(j))\right)^{\frac{1}{2}} \tag{2.6}$$

其中 T 表示转置矩阵，Σ是 $p\times p$ 的样本协方差矩阵，Σ^{-1} 相对Σ标准化了数据。注意尽管我们一直是把 p 维测量向量 $\mathbf{x}(i)$考虑为数据矩阵中的行，但是矩阵代数中的惯例是把它当作 $p\times 1$ 的列向量（column vector）（我们仍然可以把我们的数据矩阵想像为一个 $n\times p$ 的矩阵）。Σ的元素（k，l）是变量 X_k和 X_l间按公式 2.5 定义的样本相关系数。于是我们把向量 $p\times 1$ 转置（以给出一个 $1\times p$ 的向量），再乘以 $p\times p$ 的矩阵Σ^{-1}，再乘以一个 $p\times 1$ 的向量，得到一个标量的距离。当然在Σ的位置也可以用其他的矩阵。实际上，仍使用“典型”变量分析（canonical variates analysis）和判别分析的统计框架就是使用不同实例（cases）组的协方差均值矩阵。

也可以用其他方式来推广欧氏标距。例如，一种明显的泛化是闵可夫斯基（Minkowski）（译注：亦有人称“闵氏”）或 L_λ标距：

$$\left(\sum_{k=1}^{p}(x_k(i)-x_k(j))^{\lambda}\right)^{\frac{1}{\lambda}} \tag{2.7}$$

其中$\lambda\geqslant 1$。使用这种推广，欧氏距离是$\lambda=2$ 时的特例。L_1 标距（又被称为曼哈坦（Manhattan）或城市街区标距）可被定义为：

$$\sum_{k=1}^{p}|x_k(i)-x_k(j)| \tag{2.8}$$

当$\lambda\to\infty$时得到 L_∞标距

$$\max_k|x_k(i)-x_k(j)|$$

还有大量的其他标距可以用来量化相异度，所以与其说问题是如何定义标距，还不如说是如何判断哪一个标距最适于特定的问题。

36 对于多变量的二进制数据（binary data）我们可以数出（count）两个对象取相同值或取不同值的变量计数。考虑表 2-1，i 和 j 为两个对象，为它们定义的所有 p 个变量的取值范围都是{0, 1}；表格中当 $i=1$ 和 $j=1$ 时的表项 $n_{1,1}$ 表示 i 和 j 的值都为 1 时的变量有 $n_{1,1}$ 个。

表 2-1　二进数变量的交叉分类

	$j=1$	$j=0$
$i=1$	$n_{1,1}$	$n_{1,0}$
$i=0$	$n_{0,1}$	$n_{0,0}$

对于二进制数据，我们一般不再衡量对象间的相异性，而是衡量相似性。或许最明显的相似性尺度是简单匹配系数（simple matching coefficient），具体定义为两个对象取相同值的变量数占总变量数的比例：

$$\frac{n_{1,1}+n_{0,0}}{n_{1,1}+n_{1,0}+n_{0,1}+n_{0,0}} \tag{2.9}$$

其中 $n_{1,1}+ n_{1,0}+ n_{0,1}+ n_{0,0} = p$，即变量的总数。然而有时包括（0，0）情况（或（1，1）的情况，依赖于 1 和 0 的含义）是不适宜的。例如，如果变量所表示的是具有（为 1）或不具有（为 0）某种特定属性，那么我们可能不关心那些两个对象都不具有的无关属性。（例如，在文本文档的向量表示中，两篇文档中都不包含成千的特定术语，不过这一点可能是对我们的问题无关的。）这种考虑产生了一种改进的匹配系数，被称为 Jaccard 系数，具体定义为：

$$\frac{n_{1,1}}{n_{1,1}+n_{1,0}+n_{0,1}} \tag{2.10}$$

Dice 系数延伸了这一讨论。如果（0，0）匹配是无关的，那么从相关的角度来看，(0, 1) 和（1，0）的不匹配数应该在（1，1）匹配数和（0，0）匹配数之间。由于这个原因（0，1）和（1，0）的不匹配数量应该被折半。这样便得到了 $2n_{1,1}/(2\ n_{1,1}+\ n_{1,0}+\ n_{0,1})$。和前面定量（quantitative）数据的情况一样，对于多元二进制数据也有很多不同的尺度——同样，问题不是定义这样一个尺度，而是选取一个具有我们手头问题所期望特征的尺度。

对于变量具有两个以上范畴的范畴型数据，我们可以把两个对象取值一致的变量评为 1，否则评为 0，并对这些值求和，然后再除以总的变量数 p。如果我们知道范畴，那么我们 37
可以定义一个矩阵，给出各种不一致情况的值。

加成的距离尺度可以方便地升级到适合处理混合类型的数据（例如，一些是二进制的、一些是范畴值，一些是定量值），因为我们可以把每个变量的贡献相加。当然，标准化的问题也还是存在的。

2.4 数据转化

有时原始数据的形式并不是最方便的，因此在分析前对它们进行调整是有好处的。注意，数据的性质是直接影响模型形式的。例如，如果我们推测变量 Y 是变量 X 的平方的函数，那么我们既可以努力寻找 X^2 的合适的函数，又可以先对 X 乘方令 $U = X^2$，然后匹配一个对 U 的函数。在这个简单的例子中，很明显两种方法是等价的，但有时一种或另一种可能更加直截了当，易于理解。

例 2.3 很明显在图 2-2 中变量 V_1 和变量 V_2 是非线性相关的。然而如果我们取 V_2 的倒数，也就是定义 $V_3 = 1/V_2$，那么我们就得到了图 2-3 所示的线性关系。

有时，特别是如果我们在使用正式的统计推理，而且在这种推理中分布的形状是很重要的（因为要运行统计测试或计算置信区间），那么我们可能要转换数据以便使它们更接近所必需的分布。例如，对正向倾斜的（positively skewed）数据（例如银行账号里的金额或收入）取对数使其分布更均衡（symmetric）（以便它更好地逼近一个正态分布，很多推理过程都是基于正态分布的）是很常见的。

例 2.4 在图 2-4 中两个变量不仅不线性相关，而且变量 V_2 随着变量 V_1 的增长而增长。有时推理是基于固定变化率假定的（例如，回归分析中的基本模型）。对于这些数据（人工模拟的）的情况，对 V_2 取平方根得到了图 2-5 所示的转化后的数据。

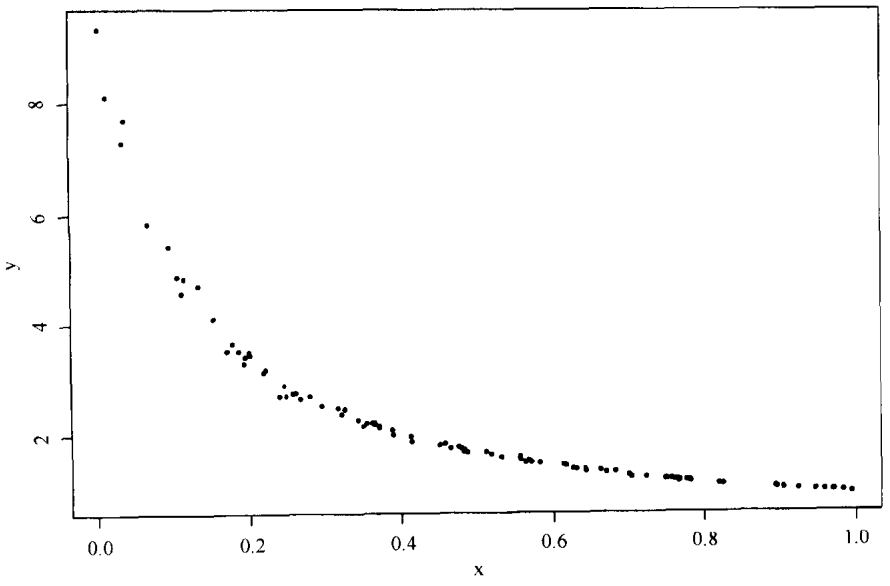


图2-2　变量V_1和V_2间的简单非线性关系（在这一幅和后边的几幅插图中V_1和V_2分别在X轴和Y轴上）。

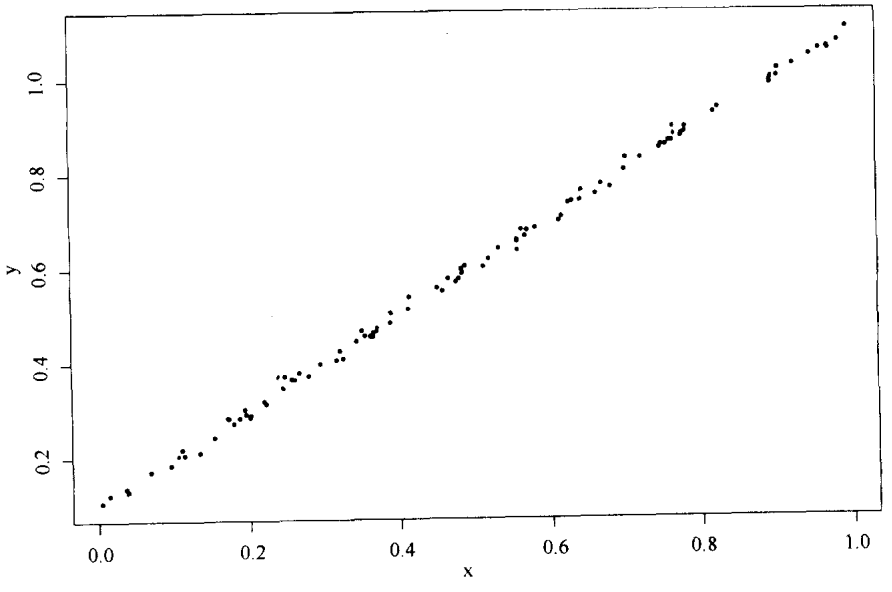


图2-3　进行简单变换，把V_2变换到$1/V_2$后的图2-2中的数据

既然数据挖掘的基本目标是探索新的发现，那么我们必须始终注视并搜索未知的情况。对数据进行某些转换可能发现本来并很不明显的结构。另一方面，过分地依赖数据转化也可能物极必反：我们必须提防完全由机械的数据变换产生的结构（参见 2.2 节中顺序的疼痛标度的例子）。一般，当数据挖掘中发生这种情况时，负责评估“新发现”的专家会很快推翻这样的结构。

也要注意到数据转化可能牺牲数据表示对象的方式。例如 2.2 节中描述的石头到重量的标准映射把物理结合映射到加法操作。如果我们对表示重量的数字进行非线性转换，例如使用对数或平方根操作，那么就不再保持物理上的结合操作了。因此必须在应用转化时注意这

一点。

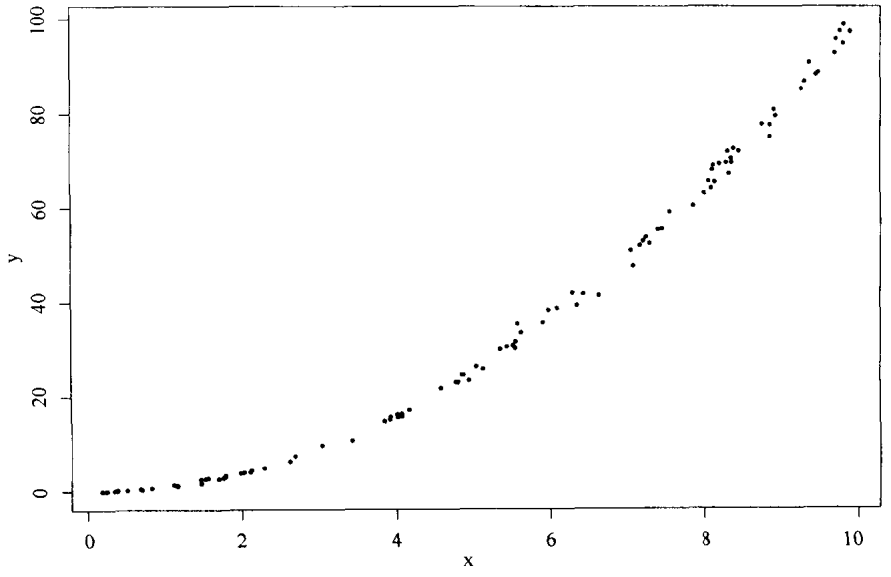


图2-4 另一种简单的非线性关系。这里变量V_2的变化率随着V_1的增长而增长

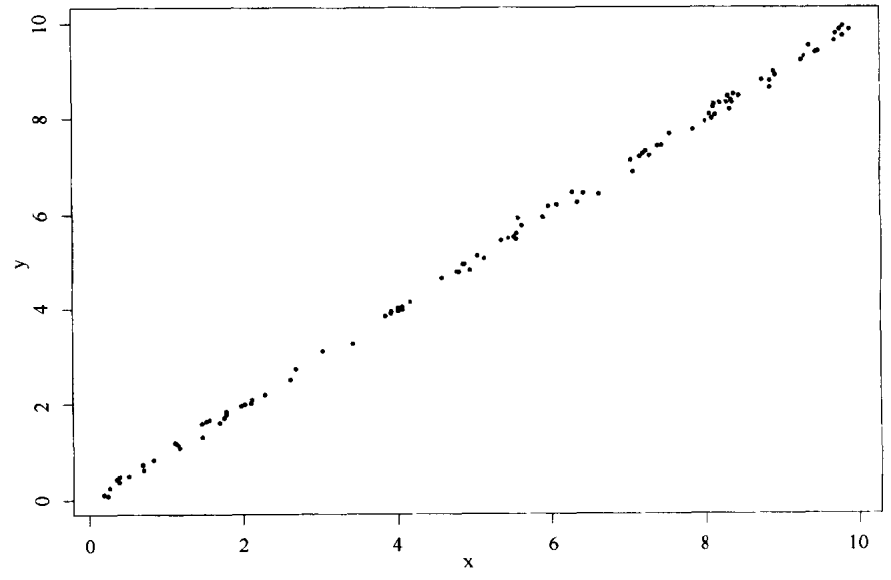


图2-5 对V_2进行简单的平方根变换后图2-4中的数据。此时V_2与V_1是线性相关的，随着V_1的增长V_2的相对变化率是固定的

普通的数据转换包括取平方根、倒数、对数和调整变量到正的整数乘方。对于表示为比例的数据经常使用分对数转换（logit transformation），$f(p)=\frac{p}{1-p}$。

某些技术假定变量是范畴型的——也就是说仅有几种（有序的）可能响应（response）。对于极端的情况，一些技术假定响应是二进制的，即仅有两种可能的分类结果。当然可以用多个不同的阈值对连续的变量（那些至少在理论上可以在给定区间中取任意值的变量）进行

分割，这样便把它们缩减为范畴型的。这牺牲了信息，而且信息的丢失随着范畴数的减少而增加，但在实践中这个损失可能非常小。

2.5 数据形式

在第一章中我们提到数据集具有不同的形式；这些形式被称为图式（schema）。最简单的数据形式（也是我们已经仔细讨论过的唯一形式）是关于对象 $o(1)$，…，$o(n)$的测量向量集。对于每一个对象我们有 p 个变量的测量值。我们称这种标准的数据形式为*数据矩阵*（data matrix），或就叫*标准数据*（standard data）。我们也可以把这样的数据集称为*表*（table）。

38
~
41

很多时候，我们要分析几种类型的对象。例如，在一个工资数据库中，我们可能既有关于雇员的数据，例如**姓名**（name）、**部门名**（department-name）、**年龄**（age）和**工资**（salary）；又有关于部门的数据，例如**部门名**（department-name）、**部门预算**（budget）和**部门经理**（manager）。这些数据矩阵是通过**部门名**字段以及**姓名**、**部门经理**字段的相同值相互连接的。由多个这样的矩阵或表组成的数据集被称为*多重关系数据*（multirelational data）。

在很多情况下，多重关系数据可以被映射到单一的数据矩阵或表中。例如，我可以使用变量**部门名**的值来连接（join）以上两个数据表。这样我们就得到一个包含变量**姓名**（name）、**部门名**（department-name）、**年龄**（age）、**工资**（salary）、**部门预算**（budget）、**部门经理**（manager）的数据矩阵。这种变换的可能性似乎提示我们根本没有必要考虑多重数据关系结构，因为原则上我们可以用一张大的表或矩阵来表示这样的数据。然而，这种连接数据集的方法不是唯一的可能办法：我们也可以创建一张表，在这张表中存在的所有部门都有对应的行（如果我们对取部门的信息感兴趣那么这种办法是有价值的，例如决定是否在部门预算和部门经理的年龄间存在依赖性）。一般来说，一张单一的表不可能完全捕捉多重数据集的所有信息。更加重要的是，从数据存储和数据访问的角度来看，把多重关系数据展开成一张单一的大表格可能带来大量值的不必要重复。

某些数据集合不能很好地适合矩阵和表格的形式。一个典型的例子是时间序列，在时间序列中一连串的值对应于连贯的多次测量（例如，波形的信号强度测量，或者一个患者接受治疗后一系列时间的反应）。我们可以使用两个变量来表示一个时间序列，一个变量表示时间，另一个表示在一定时间的测量。这确实是在数据库中存储时间序列的最自然表示。然而，把数据表示为一个二变量矩阵没有考虑数据的顺序性。在分析这样的数据时，认识到数据中确实存在自然的顺序是很重要的。例如，发现相邻观测的关系比相距较远的观测间的关系更密切（关联更紧密）是不足为奇的。没有很好地考虑这个因素可能导致建模的失败。

字符串（string）是来自某个有限字符表的符号序列。一个范畴型变量的值的序列构成一个字符串；标准的英语文本即是如此，在英文中值是字母或数字字符、空格和标点符号。其他的例子还有蛋白质和 DNA/RNA 序列。这里的字母是单个的蛋白质（注意蛋白质序列的串表示是三维结构的二维视图）。字符串是另一种有序的并且未必适合标准矩阵形式的数据类型。

一种有联系的有序数据类型是*事件序列*（event-sequence）。如果给定了一个有限的事件类型表（值是范畴型的）后，那么时间序列是一系列“{事件，发生时间}”形式的对。这与字符串非常相似，但这里序列中的每项附带一个发生时间。一个时间序列的例子是电信报警

记录，其中包括每个报警的发生时间。更复杂的时间序列包括事务型数据（比如零售或财务交易的记录），其中每一笔处理都有发生时间戳并且事件本身可能比较复杂（例如，包含购买的所有商品和价格、部门名称等等）。此外，没有理由把时间序列的概念限制在范畴型数据内；例如，我们可以把它扩展到异步发生的实数值事件，比如来自动物行为实验的数据和关于外层空间物体能量爆发的数据。 42 ~ 43

当然，有时顺序可能就是为了逻辑上的方便：把患者的记录按名字的字母排序有助于检索，因此 Jones 的记录在 Smith 的前面不可能对大多数数据挖掘算法造成任何影响。尽管如此，在数据挖掘中还是应该始终保持谨慎。例如，同一家庭成员（具有相同的姓氏）的记录可能在一个数据集中相邻的出现，而且它们可能具有相联系的属性。（我们可以发现传染性疾病趋向于感染数据集中名字相近的人群）

有序数据是沿一个单维连续区展开的（每个单一变量），但其他的数据经常是位于更高维空间的。空间的（spatial）、地理的（geographic）或图像的数据是位于两维或三维空间的。某些变量是数据图式定义的一部分，认识到这一点是很重要的；换句话来说，某些变量就是用来确定观测点在空间中的坐标。发现地理数据位于二维的连续区没有什么意义。

层次（hierarchical）结构是一种更复杂的数据图式。例如，一个关于儿童的数据集可以分组成班级，班级可以分组成各个年级，再可以分组成不同学校，再可以按国家分组，等等。这个结构明显是对数据的一种多重关系表示，但在单一的表格中很难发现这样的结构。在数据分析中忽视这个结构是相当错误的。近年来对针对这种多层次数据的统计模型的研究特别活跃。下面的情况是层次结构的一种特例：对问卷上特定项目的反应要视其他问题的答案而定；例如“你是否做过子宫切除手术？”这一问题是否适合被调查者回答依赖于其对问题“你是男的还是女的？”的回答。

概括一下，在任何数据挖掘应用中注意数据的图式都是很关键的。忽视这一点，很容易错过数据中的重要模式，甚或更严重的是，重复发现那些作为数据基础设计的一部分的模式。另外在采样时我们必须特别注意数据图式问题，在第 4 章中我们将对此进行更详细的讨论。

2.6 单个测量的数据质量

数据挖掘的有效性与数据质量密不可分。在计算领域有一个熟悉的首字母缩写词来表示这种思想：GIGO——*垃圾进，垃圾出*（Garbage In，Garbage Out）。因为数据挖掘是对庞大数据集的次级分析，所以危险性更增大了。我们在数据挖掘中发现的最有趣模式很有可能是由测量的不准确、采样的失真或某些其他对数据的误解而导致的结果。 44

我们可以从两个角度来刻画数据质量：个别记录和字段的质量，以及数据集合的总体质量。下面我们依次讨论它们。

任何测量过程都可能存在误差。误差的来源的无限的，从测量人员的不小心和仪器的缺欠，到我们对测量对象的认识不够。测量仪器可能在两方面导致误差：仪器不准确（inaccurate）或仪器的的精度不够（imprecise）。因为处理不同种类的误差需要采取不同的策略，所以区别这两种情况是很重要的。

一个精确的（precise）的测量过程具有较小的变化性（经常使用测量结果的方差来衡量）。对于一个精确的测量过程来说，在同一条件下对同一对象重复测量将得到非常相似的结果。有时精确一词意味着在记录中有很多的数字位。我们不采用这种解释，因为这样的“精确”

太容易伪造了，这一点任何熟悉现代数据分析包（有时这些包给出的计算结果是到小数点后第八位或更多）的人可能都知道。

相对而言，一个准确的（accurate）测量过程不仅具有很小的变化性，而且得到的结果更接近真实值。一个测量过程可能得到精度很高的结果，但并不准确。例如反复测量某个人的身高，可能是高精度的，但如果这些测量是当这个人穿鞋时测的，那么结果当然是不准确的。用统计的术语来讲，反复测量的结果和真实值间的差异是测量过程的偏差（bias）。准确的测量不仅具有很小的偏差而且具有很小的方差。

注意“真实值”的概念是“准确”概念的一个必要部分。但这个概念远比乍听起来更加耐人寻味。例如拿一个人的身高来讲。不仅不同时刻有所差异——由于这个人的呼吸或由于他或她的心脏跳动——而且在一天当中也有所变化（重力把我们向下拉）。从太空旅行返回的宇航员明显地比他出发时高（尽管他们会很快恢复以前的身高）。Mosteller（1968）指出：
45 “当代的一些科学家相信独立于所使用的测量过程的真实值是不存在的，而且大部分的社会科学理论也充分的支持这种观点。这种观点并不局限于社会科学中，在物理学中，对微观和宏观量（例如长度）的不同测量方法会使问题变得错综复杂。另一方面，因为真实值提出了改善测量方法的方向，所以真实值的概念是有价值的；因为某些方法比其他方法更接近理想情况，那么可以用较好方法给出真实值的代替值。”

也有其他的术语来表示这样的概念。测量过程的可靠性（reliability）与它的精确性是一致的。前一个术语通常用于社会科学中，而后者被用于物理学中。对同一概念使用两个不同的名字并非是没有根据的，因为决定可靠性与决定精确性的过程是根本不同的。在衡量一个仪器的精确性时，我们可以反复使用这个仪器：假定反复测量的过程中环境条件没有大的变化。而且，我们假定测量过程本身不会影响被测量的系统。（当然，这里存在一个 grey area：如同 Mosteller 所指出的，测量过程确实可能产生非常小的或细微的干扰现象。）然而在社会和行为科学中，这种干扰几乎是不可避免的：例如一个要求被测试者记忆一系列单词的测试接连进行两次是没有价值的。有效地重复测试需要更周密的技术，例如多种形式的测试（在测试中使用两种可选择的测试形式），分成两半测试（在这种测试中，把一个测试的多个项目分成两组），以及使用具备内在一致性（一致性给出了一个测试和它的其他包含同样数量项目的版本间的期望相关性）的多种方法。

前面我们描述了导致测量不准确的两个因素。一个是基本的精确性——反复测量同一对象得到结果的相似程度。另一个是测量结果的分布相对真实值的集中程度。既然精确性对应于可靠性，那么另一个就对应于有效性（validity）。有效性是一个测量过程反映它要度量对象的程度。在很多领域——包括软件工程和经济学——需要仔细地思考如何建立可以揭示我们想要度量的潜在概念的尺度。如果一个测量过程的有效性很差，那么我们从中吸取的关于
46 目标现象的结论在最好的情况下也是靠不住的，在最坏的情况下便是完全误导。在反馈系统中这个问题尤其严重，因为此时动作是根据测量结果决定的。如果测量不是在描述目标现象，那么这样的动作会导致系统与它的目标背道而驰。

2.7 数据群体的数据质量

除了观测个体的质量，我们还需要考虑观测群体的质量。大多数统计和数据挖掘方法的目标都是如何从样本推理到总体，也就是如何基于对群体中部分对象的分析作出对整个总体

的推理。统计学家使用术语参数（parameter）来指对对象群体或分布的描述性归纳（当然，更一般的情况下，参数是索引一族数学函数的一个值）。从对象样本计算出的值被称为统计量（statistics），可以选取适当的统计量作为对参数的估计。例如，我们可以用样本的平均值作为对整个总体或分布的均值（参数）的估计。

这样的估计只有准确才有价值。正如我们前面所指出的，两种方式会导致不准确。如果从不同样本得到的估计差异很大，那么这些估计是不可靠的：因为使用不同的样本可能就得到完全不同的估计；或者估计是有偏的，倾向于太大或太小。一般来说，估计的精确性（估计随着样本的不同而变化的程度）随着样本容量的增大而提高；所以只要资源允许，我们就可以把这种不确定性减小到一个可接受的值。另一方面，偏差不是这么容易减小的。

某些估计的偏差是固有的，但不会导致问题，因为这个偏差会随着样本的增大而减小。在数据挖掘中更该引起重视的是因为样本不适当所产生的偏差。如果我们希望计算纽约居民的平均体重，那么要是把样本局限在女性范围内显然是不妥当的。如果我们这样做了，那么我们很可能低估了这个平均值。很明显，在这个例子中，我们从中抽取样本的群体（纽约的女性）不是我们希望要泛化的群体（纽约的所有人）。我们的采样框架——我们将从中抽取样 47
本的人员名单——与我们要对其作出推理的总体不匹配。这个例子的情况是比较简单的——我们可以明确地鉴别出从中抽取样本的总体（纽约的女性）。如果错误采样框架的影响不太明显时便有困难了。例如，假定我们从在办公室工作的人中抽取样本，这会导致估计有偏差吗？或许办公室中的性别比例是不相称的，或许办公室工作者倾向于比平均值重因为他们的职业使他们习惯于坐着。有很多原因说明这样的样本无法代表我们要研究的群体。样本的典型性是能否作出有效推理的关键，这与随机样本的概念是一样的。我们将在第 4 章中讨论随机样本的必要性以及抽取这种样本的策略。

因为很多情况下数据挖掘对数据的采集方式没有任何控制，所以数据的质量问题更加重要。我们的数据集可能是我们希望描述的总体的失真样本。如果我们知道这种失真的特征，我们就有可能在推理中考虑到这个因素，但是一般来说实际并非如此，所以作出推理时必须谨慎。有时用术语“机会样本（opportunity sample）”和“顺便样本（convenience sample）”来描述对目标总体抽样不正确的样本。上面关于办公室工作人员的样本就是一个这样的例子——从他们中采样比从纽约的全部人群采样更方便。很多原因会导致样本的失真，当包含人的因素时这个风险更加严重。例如，在庞大的样本中年龄分布趋向于聚集在以 0 或 5 结尾的整数附近——正是数据挖掘会探测到并认为特别有趣的那种模式。这可能是有趣的，但很可能对我们的分析没有任何价值。

当通过一系列筛选步骤选择客户时会出现另一种失真。例如，就银行贷款来说，先要联系一个初始群体中的客户（某些回应了，某些没有），然后评估那些回应者的信誉度（某些得到较高的分数，某些没有），然后对那些得到较高分数的客户提供一笔贷款（有些接受了，有些没有），然后跟踪那些得到贷款的客户（有些客户很好，按期归还贷款的各个部分，其他的不是），等等。在任一阶段中抽取的样本都可能曲解了前一阶段的总体。

在这个银行贷款候选者的例子中，每一步的筛选标准很清晰而且有明确的规定，但就像前面所指出的，事实却不总是这样。例如，在临床试验的样本中，选择的患者来自不同的国家，已经具有了不同的诊断经历，而且或许以前曾在不同的初级医疗机构接受过不同的治疗。 48
在这里“从精确定义的总体中抽取随机样本”的概念行不通了。取而代之的是一些硬性的包含条件或排除条件：或许患者必须是男性，年龄在 18 到 50 岁间，两年内被初诊为患有所讨

论的疾病，等等。（从这里不难理解为什么临床试验记录的有效率通常比大范围应用时发现的有效率要高。另一方面这是为了确保一定要以这种方式来应用这种治疗方法。）

除了由于样本总体和目标总体不匹配造成的样本失真外，还有其他种类的失真。很多数据挖掘任务是为了对将来发生的事情作出预测。在这种情况下总体不是静态的，牢记这一点是非常重要的。例如，客户在某个商店购物的特征会随着时间变化，或许因为周围社会文化的变化，或者是对市场促销的反应，或者由于其他很多原因。很多对预测方法的研究因为没有考虑这种“总体漂移（population drift）”因素而失败。通常，是使用与建立模型的数据同时收集的数据来评估这些方法的未来性能的——隐含的假定用来建立模型对象的分布与未来对象的分布是一样的。理想的情况是有一种更加完善的模型，它可以随时间进化。从理论上讲是可以对总体漂移建模的，但在实践中这并不简单。

警惕使用失真样本的风险对保证数据挖掘的有效性是至关重要的，不过并非所有的数据集都是从感兴趣的总体抽出的样本。很多情况下数据集包含了整个总体，但是太庞大以至于我们希望工作在它的一个样本上。只要恰当地选取样本，那么我们可以产生任意准确度的有效描述来概括这个数据集所表示的总体。当然，当数据集具有复杂的结构而且可能分布在很多不同的数据库中时，可能产生一些技术问题，我们将在第 4 章中更详细地讨论这些问题。在那里，我们介绍了如何从这样的数据集中抽取样本，以便我们可以对数据集的整个总体作出准确推理。但是我们把讨论限制在样本的实际抽取过程非常简单的情况，我们重点是应该在样本中包括哪些实例。

49 可以把样本失真看作是数据不完整的一个特例，即缺少了典型样本必须的一些整条记录。数据残缺的方式还有很多。特别是记录中可能缺少整个字段。从某种程度来说这不如前面描述的那样严重。（至少这种情况下，我们可以看见数据是不完整的！）然而数据的不完整还是可能导致重大的问题。根本的问题是“数据为什么残缺？”是否在缺少的数据中存在已经记录的数据不具备的信息？如果是，那么基于这样的观测数据作出的推理很可能是有偏差的。在任何存在不完整数据的问题中，清楚所分析的目标是很关键的。值得一提的是，如果目标是仅对具有完整记录的案例进行推理，那么仅基于这些完整记录的推理是完全有效的。

孤立点和异常观测结果代表了另一类完全不同的数据质量问题。在很多情况下，数据挖掘的目的就是探测异常：在欺诈检测和故障检测中，那些与众不同的记录正是应该感兴趣的记录。这种情况下，我们将使用模式探测过程（参见第 6 章和第 13 章）。另一方面，如果目标是建模——建立一个全局模型，以辅助理解或从中预测，那么孤立点可能使模型的要点变得模糊。这种情况下，我们可能希望在建模前把它们标识出来并删除。但仅观察一个变量时，我们可以简单地通过画出数据图形来检测出孤立点，例如直方图。远离其他点的点会落在尾部。然而当面临多个变量时，情况就变得更加复杂了。这时，有可能对于单个记录来说每一个变量都具有完全正常的值，但总体模式不正常。考虑图 2-6 中各点的分布。可以清楚地看到这里有一个异常点，如果在实践中观察这样的一个分布那么这个点会马上引起怀疑。但是这个点很突出完全是因为我们产生的是二维点图。如果对这些数据进行一维分析，那么这一点根本不会表现出任何异常。

此外，对于有些特别异常的实例，仅当同时分析大量变量时它们的反常性才会显露出来。这种情况时，使用计算机检测是必须的。

所有庞大的数据集都含有值得怀疑的数据。所以应该充分重视数据不完整、采样失真、测量误差以及其他可能损害数据集质量的因素。只有认识到并理解数据的不足，我们才能采

取措施减小它们的影响。然后我们才能保证发现的结构和模式反映了客观世界的真实情况。既然数据挖掘者很少能对数据采集过程进行控制，那么就更应该警惕不良数据所导致的危险。Hunter（1980）简洁地指出了这种风险：

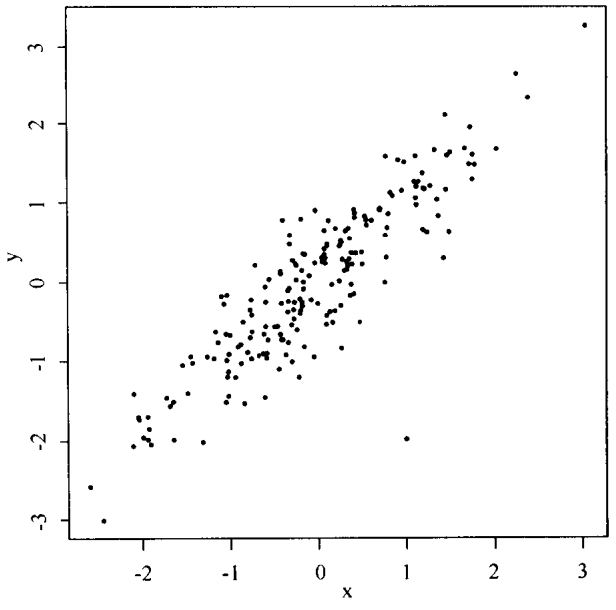


图2-6 数据中的孤立点。图中画出了正相关的二元数据（来自一个二元正态分布）的200个数据点，其中含有一个明显的孤立点

> 质量低劣的数据是清晰思考和合理决策的污染源。偏倚的数据以及从这些数据推导出的关系可能对法律法规的制定造成严重的后果。

而且我们还可以补充很多危害，比如它们还可能严重地影响科学理论的发展、重要商业信息的揭示、生活质量的提高，等等。

50
~
51

2.8 本章归纳

在这一章中，我们的讨论仅限于数字型数据。然而，其他类型的数据还有很多。例如文本数据是一类重要的非数字型数据，我们将在第 14 章讨论这种数据。有时数据项个体的定义（从而决定它是数字型的或非数字型的）依赖于分析的目标：在经济领域，数百数千个时间序列存储在数据库中，数据项可能就是整个时间序列，而不是这些序列中的各个数字。

即使对于非数字型数据，数字数据分析也具有重要的作用。大多时候，非数字数据项或它们间的关系被简化为数字描述，这是一些标准分析方法所必须的。例如，在文本处理中我们可以衡量某个单词在每一篇文档中出现的次数或某些单词对在文档中成对出现的概率。

2.9 补充读物

关于表示性测量理论（representational measurement theory）的巨著是三卷本的 Krantz et al.

（1971），Suppes et al.（1989）和 Luce et al.（1990）。Roberts（1979）也概述了这种方法。Dawes and Smith （1985）和 Michell（1986，1990）描述了其他一些方法，包括操作性的方法。Hand（1996）探讨了测量理论和统计学的关系。一些作者在基本的测量理论框架下探讨了用于软件工程的尺度——比如 Fenton（1991）。Anderberg（1973）深入地讨论了相似和不相似尺度。

在处理社会、行为和医药科学领域的问题时，经常离不开对可靠性和有效性的讨论——参见 Dunn（1989）和 Streiner and Norman（1995）。Carmines and Zeller （1979）也讨论了这一问题。关于数据不完整和不同类型的数据残缺机制的重要著作是 Little and Rubin（1987），用于说明失真样本的银行贷款实例来自 Hand，McConway，and Stanghellini（1997）。
52 Goldstein（1995）是一本关于多层次建模的重要著作。

第3章　可视化和探索数据

3.1　简介

本章将讨论如何运用可视化方法来发现数据中的结构。可视化方法之所以在数据探索中占有特别重要的地位是因为人类的眼睛和大脑具有强大的结构探测能力——这是长期进化的成果。可视化方法就是以各种可以发挥出人类在模式处理方面的特殊能力的方式来显示数据。这种方法与正规的建模方法和用来验证观察数据是否出自某种假设结构的方法是截然不同的。可视化方法在数据挖掘中具有非常重要的地位，它是筛选数据寻找未知数据关系的理想方法。另一方面正如下面将阐述的，它也具有局限性，尤其是对于非常庞大的数据集合。

可以把探索性数据分析理解为*以数据驱动的方式生成假设*（data-driven hypothesis generation）。我们从各个角度审查数据，目的是发现可以体现各个实例或变量间深层关系的结构。这一过程与假设检验（hypothesis testing）大不相同，后者是先提出一个模型或假设，然后进行各种统计处理以判定数据出自这一模型的可能性（参见第4章）。这里*"数据驱动"*的含义是为了说明是数据中的模式引发了假设——这不同于根据有关数据潜在机制的理论推导出假设的情形。这一特征暗示了对这些假设进行检验的必要性。这与第7章（第10章、第11章中会再次提到）讨论的过度拟合问题有着密切的关系。下面用一个简单事例来说明这个问题。

如果我们从同一总体中随机提取10组样本，每组的容量为20，然后测量某个变量的值，这些随机样本会有不同的均值（由于随机变化性）。我们可以用正式的检验来比较这些均值。假定无论如何我们只取产生最小和最大均值的两组样本，并忽略其他样本。对于这些均值差异的检验很可能是显著的。如果我们取100个样本而不是10个，那么就更有可能发现最大和最小均值间的显著差异。尽管这些样本是从同一总体产生的，但是如果我们忽略这些均值是100个样本集合的最大和最小均值这一事实，那么我们的分析就是偏向于发现差异的。

通常，当我们搜索模式时，如果不考虑搜索的规模——我们已分析的可能存在的模式数，那么我们就无法检验所发现的模式是否是潜在分布的真实属性（相对于样本的偶然属性）。探索性数据分析的非正式性使这一问题变得非常复杂——很多情况下要统计出已经审查了多少种模式是不可能的。由于这一原因，科研人员经常使用一个分离的数据集合（与前面的数据集合来自于同一数据源）并采用正式的检验方法来验证模式的存在性。（或者也可以使用某些复杂的方法，例如第7章介绍的交叉验证和样本复用方法。）

本章分析了用非正式的图形来探索数据的方法，该方法在数据分析中的广泛运用可以追溯到很多个时代以前。早期的统计书籍中包含了许多这样的方法。在计算机出现以前它通常比其他冗长的数字分析方法更实用。近年来某些领域的变革使这些方法的应用更加广泛。和本书中提到的大量其他方法一样，这些变革是由计算机所引发的：计算机使我们可以用许多不同的方式来观察数据，既快又方便，并且已经开发出了很多功能强大的可视化工具。

我们将在3.2节中讨论总结数据的简单统计方法。在3.3节中将讨论用可视化方法探索单个变量值的分布。这些工具已经应用了数个世纪（至少对于小数据集来说是这样的），近年来计算

机技术的进步使其有了更新的发展。此外，即便是使用单变量显示，我们通常也希望同时显示
54 出许多变量的单变量显示，所以我们需要的是易于表达数据分布主要特征的简洁表示。

3.4 节将转到如何显示变量对之间关系的方法。最基本的方法或许是散点图（scatter plot）。由于数据挖掘中所碰到的数据集合经常是很庞大的，所以散点图并不总是有效的——有时会被数据所淹没。当然，这一法则也适用于其他图形表示。

3.5 节将超越一对变量的情况，描述用于揭示多个变量间关系的工具。当然，所有方法都不是完美无缺的：除了极少数的数据关系外，二维显示根本无法完全表示出多个变量间的关系。

在 3.6 节中将举例说明主分量分析（principal components analysis）方法。这种方法可以被认为是多维缩放分析（multidimensional scaling analysis）方法的一种特殊形式（实际上是最基本的形式）。

关于数据可视化的书籍数不胜数（参见 3.8 节）。我们不能指望在短短一章中讨论所有的方法。也有一些用来使数据可视化的软件包，它们提供了很多非常强大而又灵活的图形化工具。

3.2 总结数据：几个简单例子

在前两章中我们提到过均值就是对一组数据的平均值的简单概括。假定 $x(1)$，…，$x(n)$ 组成了一个 n 个值的集合。那么样本均值就被定义为：

$$\hat{\mu} = \sum_i x(i)/n \tag{3.1}$$

注意我们用μ来表示总体的真正均值，用 $\hat{\mu}$ 来表示这个均值的样本估计。

样本均值的一个特征是：它与所有数据值的差异平方和是最小化的，从这个意义上来说它是这些样本值的“中心”。因此，如果有 n 个数据值，那么均值就是满足它的 n 个拷贝之和等于这些数据值之和这一条件的值。

均值是一种位置（location）尺度。另一种重要的位置尺度是中值（median），中值就是使在其上的数据点数和在其下的数据点数相等的值。（如果 n 是一个奇数，那么很简单。如
55 果 n 是偶数，那么通常把中值定义为两个中间数据值的中点。）

数据中的最普遍值被称为最频值（mode）。有些分布会有多个最频值，例如，对于某一变量，可能有 10 个对象取值为 3，10 个对象取值为 7，取其他值的对象数都小于 10。这种情况被称为多峰型（multimodal）。

另外的一些位置尺度则侧重于数据值分布的其他方面。第一四分位值（quartile）就是大于四分之一数据点的值。第三四分位值就是大于四分之三数据点的值。（这里为什么我们不提第二四分位值呢？我们把这个问题留给读者。）类似地，有时也会用到十分位值（decile）和百分位值（percentile）。

常见的还有衡量分散性（despersion）或者变化性（variability）的不同尺度。包括标准差（standard deviation）和它的平方，也就是方差（variance）。方差被定义为各个数据值和均值的差异平方的平均值：

$$\hat{\sigma}^2 = \sum_i (x(i) - \mu)^2 / n \tag{3.2}$$

需要注意的是既然均值使这些差的平方和最小化，那么均值和方差之间就存在着紧密的关联。如果μ是未知的，这是实践中常出现的情况，那么我们可以用$\hat{\mu}$（建立在数据基础上的估计）来代替μ。在用$\hat{\mu}$来代替μ后，我们可以用下式得到方差的无偏估计（我们将在第4章中讨论无偏估计的含义）：

$$\sum_i (x(i)-\hat{\mu})^2/(n-1) \tag{3.3}$$

标准差是方差的平方根：

$$\hat{\sigma}=\sqrt{\sum_i (x(i)-\mu)^2/n} \tag{3.4}$$

一些应用经常使用的还有四分位值域（interquartile range），它是指第三和第一四分位值之间的区域。值域（range）就是最大和最小数据点之间的区域。

倾斜度（skewness）用来衡量一个分布是否具有单一而且很长的末端，通常被定义为：

$$\frac{\sum (x(i)-\hat{\mu})^3}{(\sum (x(i)-\hat{\mu})^2)^{3/2}} \tag{3.5}$$

例如，人们收入的分布通常表现为大多数人只赚很少或中等数量的钱，只有少数人有很高的收入——如比尔·盖茨。如果一个分布的漫长末端是伸向数值增长方向的，那么我们称其为*右倾斜*（right-skewed），反之称其为*左倾斜*（left-skewed）。右倾斜的分布更加常见。对称分布的倾斜度为零。 56

3.3 显示单个变量的一些工具

直方图是显示一元数据的最基本工具之一，它显示了位于各个连续区间中的变量值数目。对于很小的数据集合，直方图可能造成误导：值的随机波动或对区间端点的不同选择会得到截然不同的直方图。比如起初看来是有多峰性的，而后却由于区间选择的不同或样本的变化而消失了。不过，随着数据集合的变大，这些影响会逐步减小。对于一个大的数据集合来说，即便是直方图的细微变化也代表了数据分布的真实特征。

图3-1显示了1996年某种个人信用卡持有者使用该信用卡在超市消费的周数（为了回避商业上的敏感细节，纵轴的标签被隐去了）。在直方图的左侧存在一个很大的波峰（最频值）：这说明大多数人在超市购物时不用或很少用信用卡。使用信用卡一定次数的人数随着使用次数的增大迅速下降。然而，图中所反映出的数量比较大的人群让我们发现了另一个事实：在图表的右边快到末端处有一个比刚才那一波峰小的多的波峰。显然，人们往往是很有规律的每周去超市购物一次，尽管这一波峰不是位于52周处，这可能是由于休假等中断原因导致的。

例3.1 图3-2显示了768位具有印第安比马人血统的女性的血压舒张压的直方图。这一变量是为建立预测糖尿病的分类模型而收集的八个变量中的一个。这一数据集合的文档（可以从UCI机器学习在线数据文档中得到）指出该数据集合不存在残缺值。然而，粗略地扫视一下直方图发现有35个被测试者的血压值为零，这显然是不可能的，除非接受检验时她们已经死了。一种可能的解释是这35个人 57

事实上错过了测试，“0”值就是用来表示错过测试的代码。因为很多变量（例如三头肌折叠肌肤厚度）值都是不可能为零的，所以这一解释是有可能的。

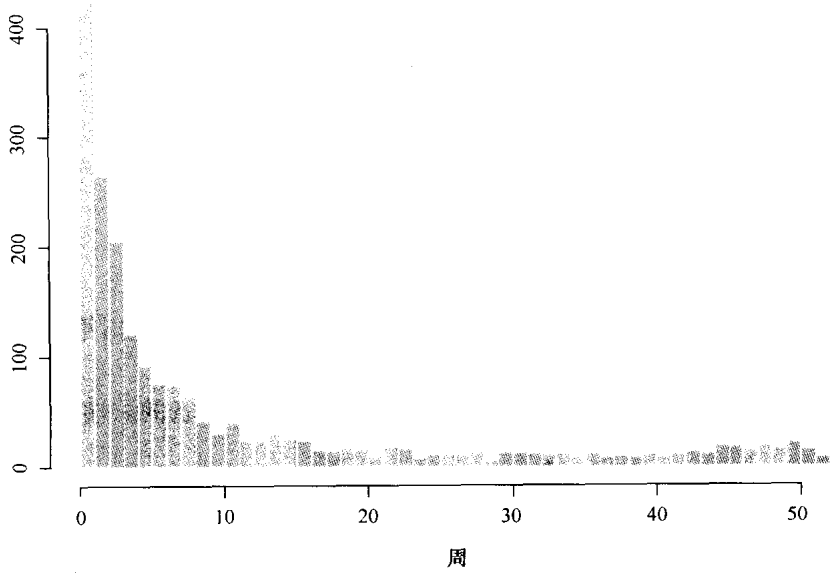


图3-1　1996年使用某种信用卡在超市购物周数的直方图

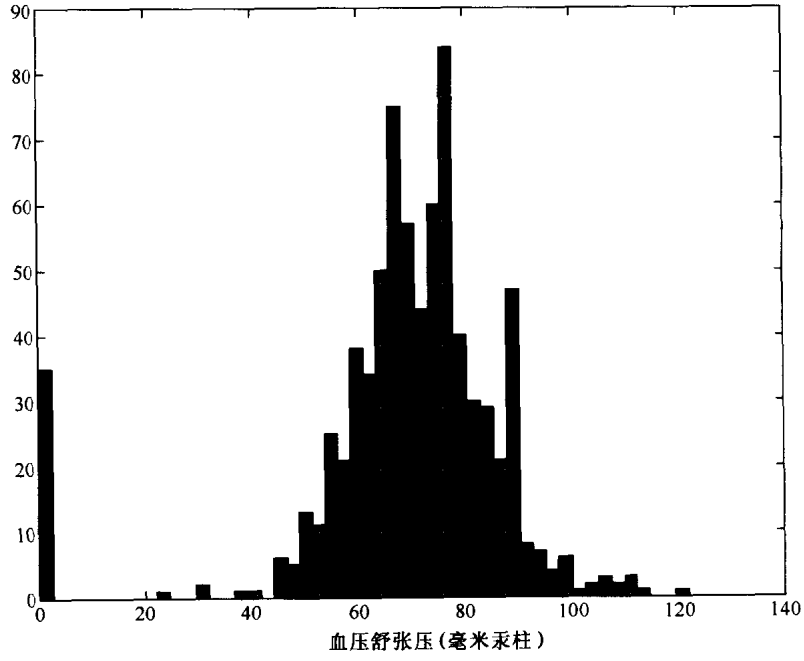


图3-2　具有印第安比马人血统的768位女性的血压舒张压直方图

虽然直方图有不足，但是在更详细建模前使用它来探索数据是非常有价值的。在印第安比马人数据的案例中，直方图清楚地显示出了数据中和被测量变量的物理解释相矛盾的可疑变量值。在进行数据挖掘之前做一下这样的简单检查总是明智

的。因为一旦我们所应用的算法无法发现如上的数据质量问题，那么这些问题很可能会以一种无法预知的方式歪曲我们的分析。

可以通过平滑估计来弥补直方图的不足之处。应用最广泛的方法之一便是核估计（kernel estimate）。

核估计对每个观察数据点的贡献相对其邻域进行平滑处理（在第 9 章中我们还会讨论核估计）。考虑一个单一变量 X，我们对其测出了一系列值$\{x(1), \cdots, x(n)\}$。数据点 $x(i)$对在某一点 x^*的估计的贡献依赖于 $x(i)$和 x^*间的距离有多远。可以做出贡献的范围依赖于所选核函数的形状以及相应的核宽度。如果用 K 代表核函数，用 h 代表它的宽度（或者叫带宽），那么任意一点 x 处的估计密度可以表示为：

$$\hat{f}(x)=\frac{1}{n}\sum_{i=1}^{n}K\left(\frac{x-x(i)}{h}\right) \tag{3.6}$$

其中 $\int K(t)dt=1$，以保证估计 $f(x)$本身的积分为 1（也就是说，它符合密度函数的基本条件），而且通常把核函数 K 选为波峰位于 0 的平滑单峰函数。核估计的质量受 K 形状的影响比受 h 的影响小。

58 ~ 59

K 的一种常见形式就是正态（高斯）曲线，h 就是它的分布参数（标准差），也就是：

$$K(t,h)=Ce^{-\frac{1}{2}(\frac{t}{h})^2} \tag{3.7}$$

其中 C 是一个用于标准化的常量，$t = x - x(i)$是查询点 x 与数据点 $x(i)$之间的距离。带宽 h 等价于高斯核函数的标准差（也就是宽度）σ。

有许多正规的方法可以用来优化这些估计与产生数据的未知分布的拟合情况，但是，本章中我们感兴趣的是图形过程。这种估计的优点在于通过改变 h 的值，我们可以寻找样本分布形状的独有特征。小的 h 值产生非常尖利的估计（根本不平滑），大的 h 值会导致估计过于平滑。h 值的极限就是当 $h \to 0$ 时数据点的经验分布（也就是，关于每个数据点 $x(i)$的“delta 函数”），和 $h \to \infty$时的均匀平滑分布。这两种极端情况分别对应于完全依赖观察数据（除了观察到的数据点外不考虑任何其他量）和彻底忽略观察数据。

图 3-3 显示了参加骨质疏松症研究的 856 位老年女性体重的密度核估计。该分布表现出明显的右向倾斜性，并且存在多峰性的迹象。很明显经典统计中经常使用的正态分布假定是不适用于本例的。（这并不是说基于该假设的统计技术已经失效了。很多情况下该理论是渐进性的——是以中心极限定理所确定的正态性为基础的。在本例中，做出如下假定对实践目的是合理的：856 个接受试验者的样本均值会因样本的变化依照正态分布而变化。）

图 3-4 显示了当为平滑参数 h 取一个更大值时的效果。很难回答这两个核估计哪一个更“好”些。图 3-4 更保守些，因为它对观察到数据的局部（可能是随机性的）波动给予的信任度更小。

尽管这一节的焦点在于显示单个变量的特征，但很多时候我们也希望把单个变量的值分成多个不同的组，然后分别显示出每个组，目的是对各个组进行比较。（当然，我们也可以把这种情况看作是二变量的情况，其中一个变量是分组因子。）我们可以对每个组分别使用直方图、核曲线以及其他一维显示方法。然而，如果组数超过两三个时，那么处理起来便会比较困难。对于这种情形，另一种有用的替代显示方法是框须图（box and whisker plot）。

60

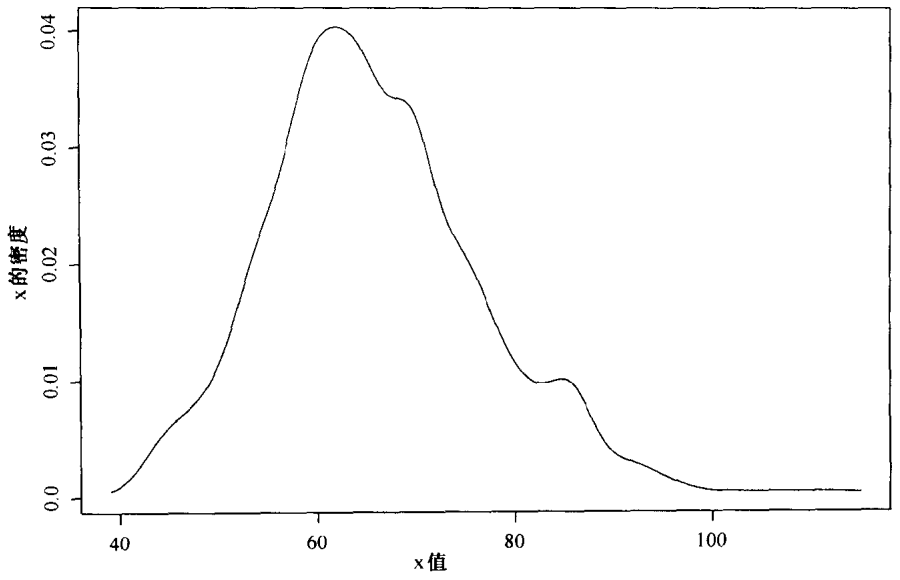


图3-3　856位老年女性体重（kg）的密度核估计

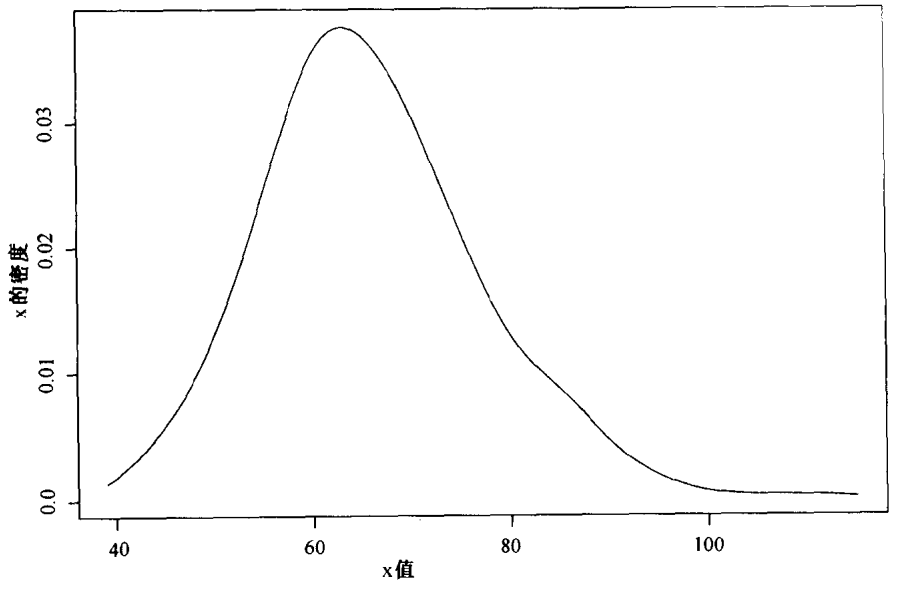


图3-4　与图3-3一样，不过更平滑

尽管框须图有很多个版本，但是其核心思想是相同的。框包含了数据的主要定义域——例如第一四分位值和第三四分位值之间的区间。横跨框的一条直线表示出某个位置尺度——通常是数据的中值。框末端的须状投影表示实验分布的末端散布范围。

下面用图 3-2 中糖尿病数据的子集来举例说明框须图。图 3-5 画出了四幅框须图，每幅都包含了数据中两个类（健康的（1）和患糖尿病的（2））的各自框须图。图中清晰地表示出了均值、分散度和倾斜度是如何随分组变量值的变化而变化的。

61

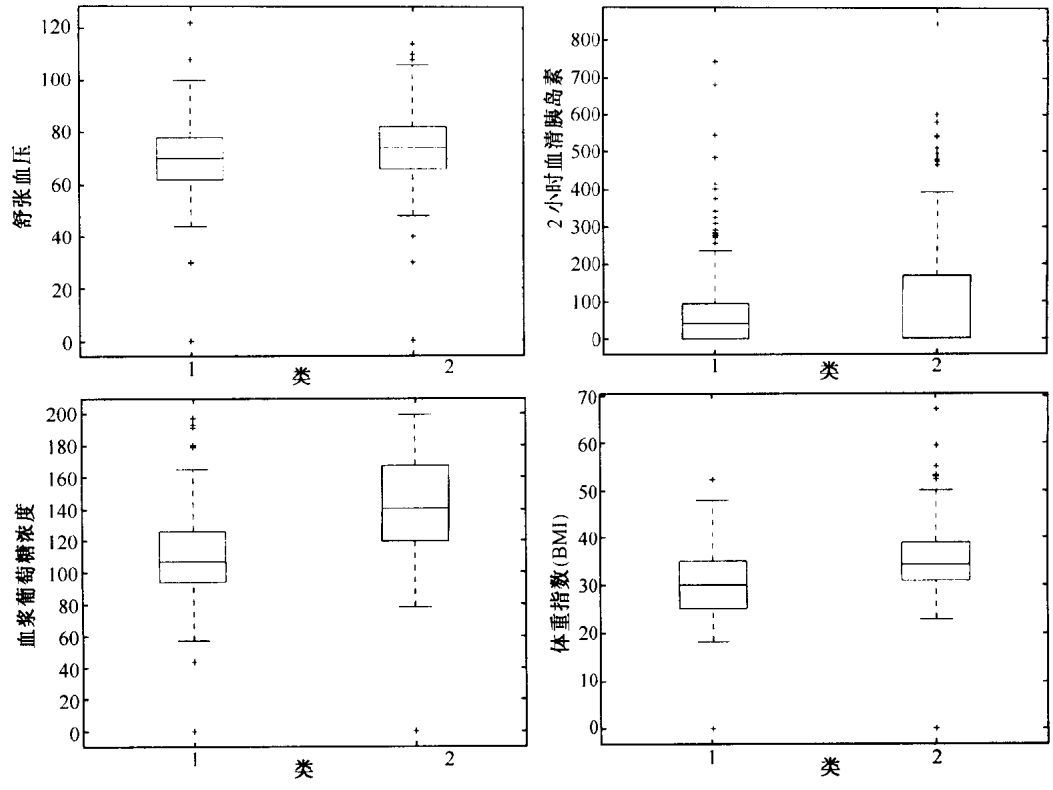


图3-5 印第安比马人糖尿病数据集的四个不同变量的框须图。图中为每个变量分别画出了健康对象（分类标签为1）和患糖尿病对象（分类标签为2）的框须图。每个框的上边界和下边界分别表示数据的较高和较低的四分位值。框中的水平线表示数据的中值。须从每个框的两端延伸到四分位值域的1.5倍处。位于须范围外的所有数据点是分别单独标出的（尽管有些点是重叠的，比如值为0时的点）

3.4 显示两个变量间关系的工具

散点图是同时表示两个变量的标准工具。图 3-6 显示了用于描述信用卡偿还模式（细节是保密的）的两个变量之间的关系。从这张图可以非常清楚地看出两个变量间的关系是很紧密的——当一个变量的值很高（低）时，另一个变量的值也很可能很高（低）。然而，也有相当数量的人不符合这一模式，当一个变量值很高时另一个变量值很低。为什么这些个体表现出异常，这或许正是值得我们去研究的。

62

不幸的是，在数据挖掘中，散点图并不总是这么有用。如果数据点数太多，那么我们发现自己面对的将是一个几乎纯黑的长方形。图 3-7 说明了这一问题。该图画出了来自一项银行信贷研究的 96 000 个数据点的散点图。图中很难辨别出任何明显的结构，尽管它似乎显示出后来申请者的年龄通常比较大。从另一方面来说，如果右边的样本数较大，那么同样也可以导致图中右侧的纵轴较大值明显较多。事实上对这些数据的线性回归拟合表明它们具有很小但非常显著的向下倾斜性。

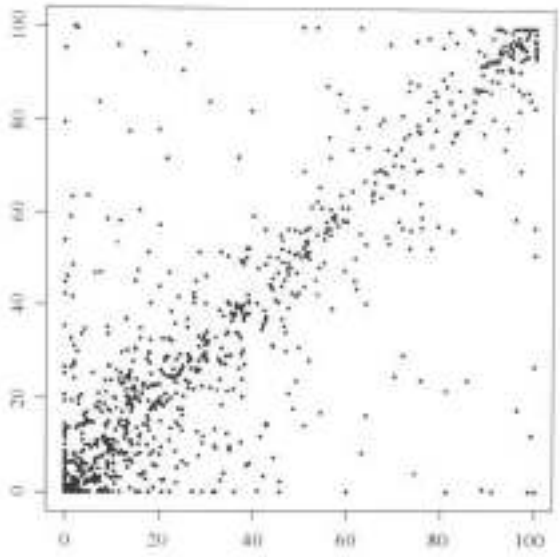


图3-6 两个金融变量的标准散点图

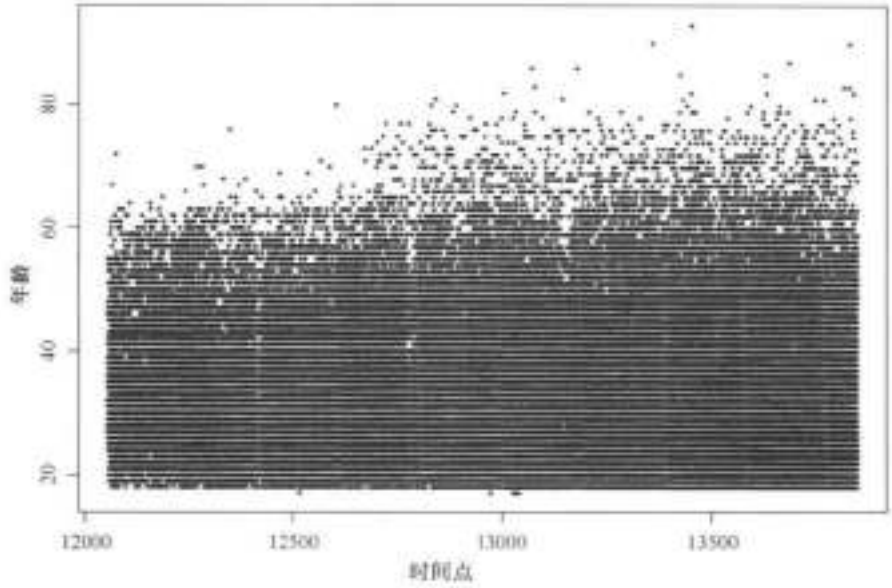


图3-7 包含96 000个数据点的散点图。图中很多点是重叠的。每个数据点表示一个独立的信贷申请人。纵轴表示申请人的年龄，水平轴表示提出申请的日期

63

即使情况并不是如此极端，如果散点图对应的数据点数量很大，那么它隐藏的特征可能仍然比它表现出的特征要多。图 3-8 描绘了过去某一年中用某种信用卡买汽油的周数相对该信用卡被在超市中使用周数的散点图（每个数据点代表一张信用卡）。这两个变量间显然存在关联，但实际的关联系数 0.482 比这里显示出的要高许多。这张图表之所以容易使人误解是因为它在底部左边转角处隐藏了大量的重叠数据点——这里一共表示了 10 000 个用户。图 3-1 所表示出的双峰性在这张图中也可以辨别出，不过没有图 3-1 那么清晰。

图 3-8 中还有一个很有趣的明显现象。在加油站使用该信用卡的周数分布是向超市变量

低值区域倾斜的，但是对于较高值，却是相当均匀的。如何解释这个现象呢？（当然，应该记住前面所说明的一点，这种表面现象需要由数据点的重叠来解释。） 64

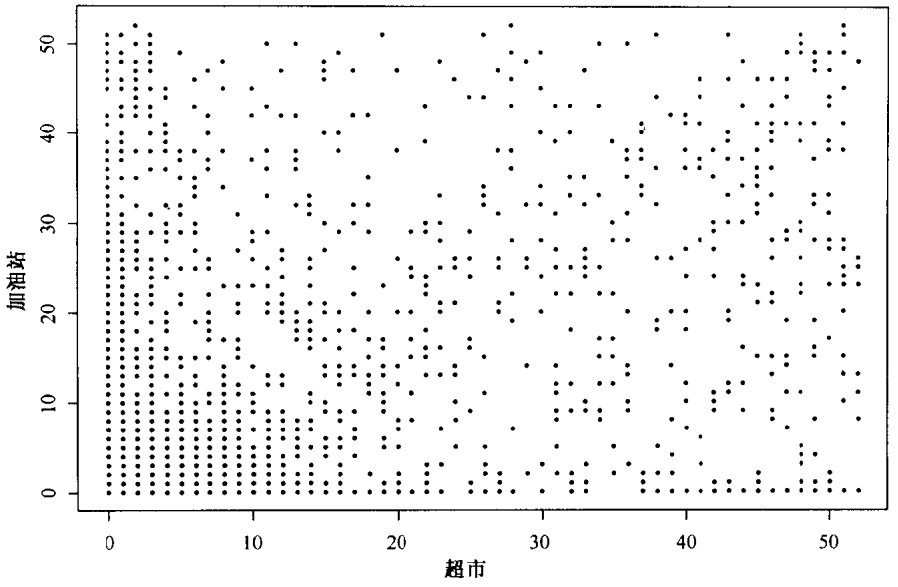


图3-8 数据点重叠隐藏了真实的关联程度

等高线图可以克服前面讨论的一些问题。不过需要注意的是，要建立有效的二维等高线图需要建立二维密度评估，使用的方法类似于公式 3.6 的核方法的二维推广。这再次涉及到选择带宽的问题，但是这次是在二维环境中。图 3-9 用等高线显示出了图 3-7 所显示的 96 000 个数据点。通过该图可以清楚地看出图 3-7 中难以辨别的某些倾向。例如，越靠近图的右侧，数据点的密度越大；纵轴的明显渐增分布是由于这个区域的数据点更加集中所导致的。数据的垂直倾斜性在这幅图中也十分明显。数据的单峰性以及单一波峰的位置在图 3-7 中是无法 65
看出的，但在图 3-9 中却可以清楚地看出来。注意，因为这幅图中水平轴是时间，所以另一种可选的数据显示方法是画出随时间推进的固定条件概率密度等高线。

当两个变量中的一个是时间时，还可以使用其他的标准显示形式，以显示出另一个变量值随时间推进的变化情况。这对于探测数据走势和了解它们与预期或标准行为的偏离情况是非常有效的。图 3-10 画出了代表 1985 年至 1993 年（含这两年）英国发行信用卡数量的各个数据点。图中用一条光滑曲线来拟合这些数据，以强调数据关系的主要特征。显然大约在 1990 年某种原因终止了此前的线性增长势头。事实上，原因是在 1990 年和 1991 年信用卡开始需要缴纳年费，所以许多用户将他们持有的信用卡数量减少到一张。 66

图 3-11 显示了英国航空公司自 1963 年 1 月到 1970 年 12 月间每月飞行英里数的曲线。从该图中可以立即看出几个和我们的预期一致的明显模式，比如逐步增长的总体趋势和周期性（夏季的较大波峰和新年前后的较小波峰）。该图还显示出了夏季高峰的有趣分叉，这表明旅行者有更喜欢在夏季的初期和末期而不是中期出游的趋势。

图 3-12 是说明两个变量之一为时间的图形表示的优点的又一例子。在 1930 年的 2 月至 6 月间，在苏格兰的拉纳克郡进行了一个实验，目的是调查在儿童食谱中添加牛奶是否有“增强体质、利于健康以及提高智力”的作用（Leighton and McKinlay, 1930）。在这一研究中，

67 把 20 000 名儿童分配到三个组中，并让 5 000 名儿童每天饮用四分之三品脱[⊖]的生牛奶，5 000 名儿童每天饮用四分之三品脱的经巴氏灭菌法处理的牛奶，另 10 000 名儿童构成一个控制组，在他们的饮食中没有牛奶。在实验开始时和四个月后结束时对每个儿童各称一次体重。研究者的目标是考察三组儿童的成长情况是否有差异。

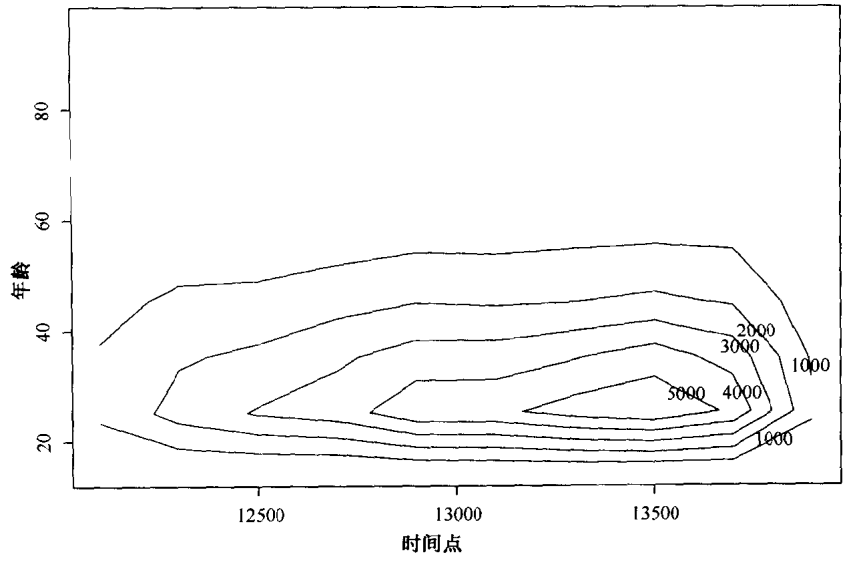


图3-9　用等高线来显示图3-7中的数据

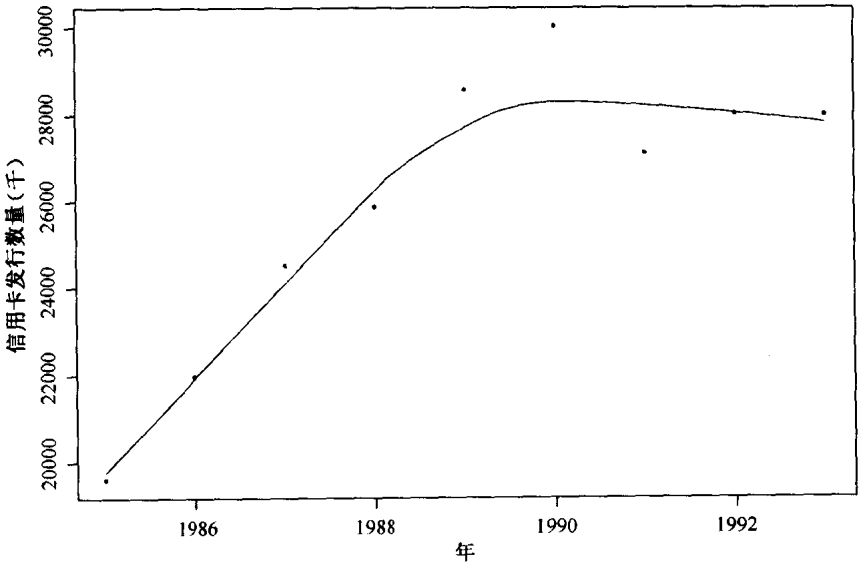


图3-10　英国的信用卡流通数量曲线

图 3-12 画出了控制组中女孩的平均体重相对她们所在组的平均年龄的曲线。第一个点对应最小年龄组（平均年龄 5.5 岁）在实验开始时的体重，第二个点对应该组在四个月后的

⊖ 译注：1 品脱=0.568 升。

体重。第三个点和第四个点对应第二个年龄组，依次类推。所有的点被连成了一条线以便于观察其形状。显然实验中所有组的形状都很相似。

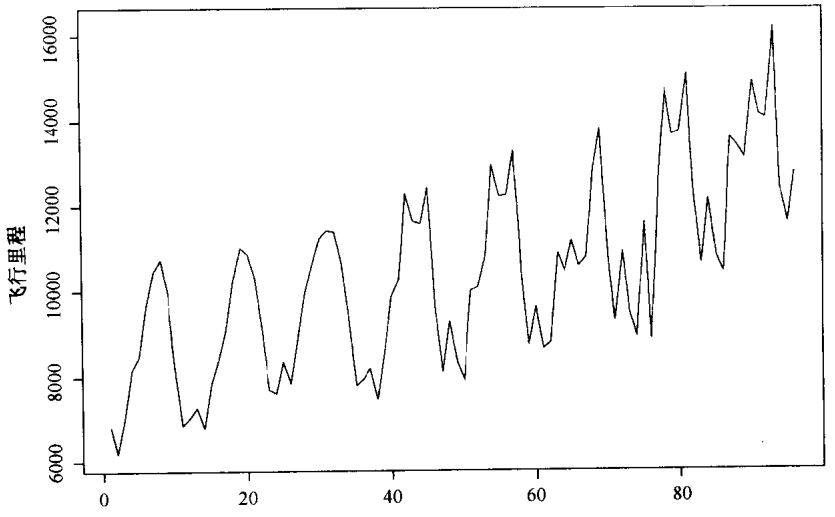


图3-11 19世纪60年代英国航空公司飞行英里数相对时间的变化曲线

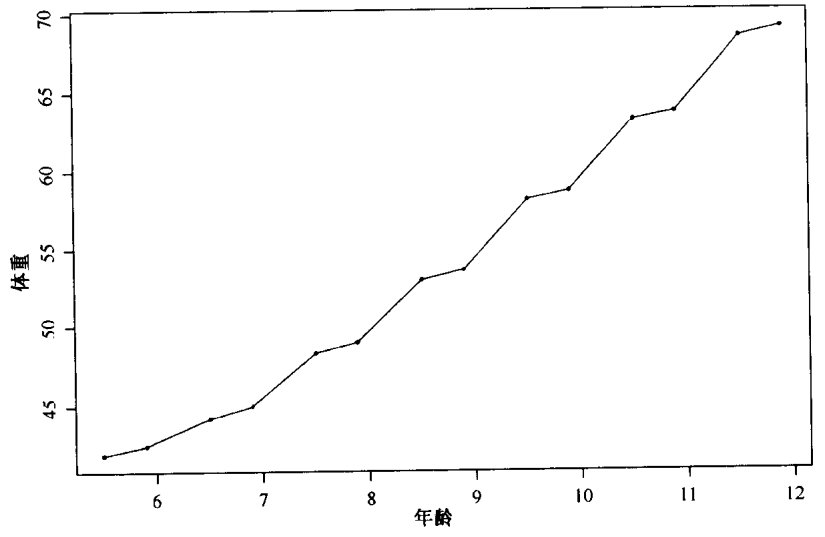


图3-12 10 000名学龄儿童体重随时间变化的曲线。台阶状模式使数据测量过程中的问题明显暴露出来

该图直接显示出了一种令人意想不到的从数据表格中无法看到的模式。我们本以为会得到一条光滑的曲线，但图中显然存在很多台阶。似乎是每个年龄组的孩子都没有达到预期的体重增长。对这种形状有许多可能的解释。也许儿童在上半年长得比下半年慢。然而，关于身高的类似曲线却没有表现出间歇性成长的特征，所以我们需要一种更周密的解释来说明为什么身高的增长是均一的而体重的增长是间歇的。另一种可能的解释是孩子们也许穿着衣服在测体重。但是报告上说："称体重时，所有孩子是脱去靴子或皮鞋只穿着普通户外衣服的。男孩们还被要求拿出他们口袋中的各种小物品，而且外套或围巾等等也是被脱掉的。如果发现哪个孩子穿了三或四件运动衫，那么会脱下其余的只剩一件"。然而，这种解释还是有可

能的，因为无论如何夏季服装比冬季服装要轻。这个例子说明了数据挖掘发现的模式虽然不能完全弄清楚被调查的现象，但是发现数据的反常和缺陷也是同样有价值的。

3.5 显示两个以上变量间关系的工具

68 ~ 70

因为纸张和电脑屏幕都是平面的，所以它们非常适合于显示二维数据，却不适合于显示多维的数据。对于多维数据，我们需要以某种投影方式将其转到二维平面，通过变换来表示其他维的特征。沿这一路线的最明显做法就是分析所有变量对间的关系，将 3.3 节中描述的基本散点图推广到散点图矩阵（scatterplot matrix）。

图 3-13 就是一个散点图矩阵的例子，它比较了近 10 年来 209 个计算机 CPU 的特性、性能尺度和相对性能尺度。其中的变量为时钟周期、最小内存（KB）、最大内存（KB）、高速缓存大小（KB）、最小通道数、最大通道数、相对性能和估计出的相对性能（相对于 IBM 370/158-3）。尽管一些变量对是无关的，但是另一些却表现出很强的相关性。画刷（brushing）法使我们可以通过突出每张散点图中对应同一对象的点的方式来强调散点图中的数据点。这对于交互式探索数据是非常有用的。

当然，散点图矩阵并非真正的多元解决方案：而是多重的两元解决方案，它使多元数据投影到多个二维图中（在每个二维图中忽略了所有其他变量）。这种投影必然会丢失信息。设想有一个由 8 个小立方体构成的大立方体。如果在相错开的子立方体中数据点是均匀分布的，其他子立方体为空[⊖]，那么所有三个一维投影和所有三个二维投影都是均匀分布的。（这种异或（exclusive-or）结构会给感知器带来很大困难，感知器是神经网络的前身，我们将在第 5 章和第 11 章中讨论。）

当涉及两个以上的变量时，交互式绘图很流行，因为这样我们就可以通过旋转投影的方向来搜索合适的结构。一些系统甚至可以让软件进行随机旋转，我们只要观察并等待感兴趣的结构出现。虽然理论上这是一个非常好的想法，但是当看着数据像云朵一样随着视角转换移动时，这种兴奋将很快就会变得平淡无味，因此我们需要更加结构化的方法。第 11 章中介绍的投影跟踪（projection pursuit）就是一种这样的方法。

格架（trellis）图也是以多个二元图为基础的。不过，该方法不再是为每对变量画出一幅散点图，而是固定针对一对要显示的特定变量，然后以其他一个或多个变量为条件画出一系列散点图。

图 3-14 所示为癫痫病发作数据的格架图。每幅图的横轴代表二周内 58 个病人发作的次
71 数，纵轴代表在随后二周内这些人的发作次数。左侧的两幅图对应的是男性病人，右侧的两幅图对应的是女性病人。靠上的两幅图对应的年龄在 29 到 42 岁之间，靠下的两幅图对应的年龄是 18 到 28 岁之间。（原始数据集中还包括一个发作次数高得多的患者。我们将其删除了，以便更清晰地观察其他对象结果间的关系。）从这些图中我们可以看到年轻病人的平均发作次数比年长病人要低。这些图还暗示了联系 y 和 x 轴的最佳拟合直线的斜率间可能存在差异，不过我们必须进行正式的检验才能确信这些差异确实存在。

还可以用其他任何形式的子图来产生格架图。也就是说除了在每个单元中用散点图外，我们还可以用直方图、时序图、等高线图或者其他任何类型的图形。

⊖ 即 4 个立方体为空，4 个立方体均匀充满数据点，空的与非空的相重叠。——译者注

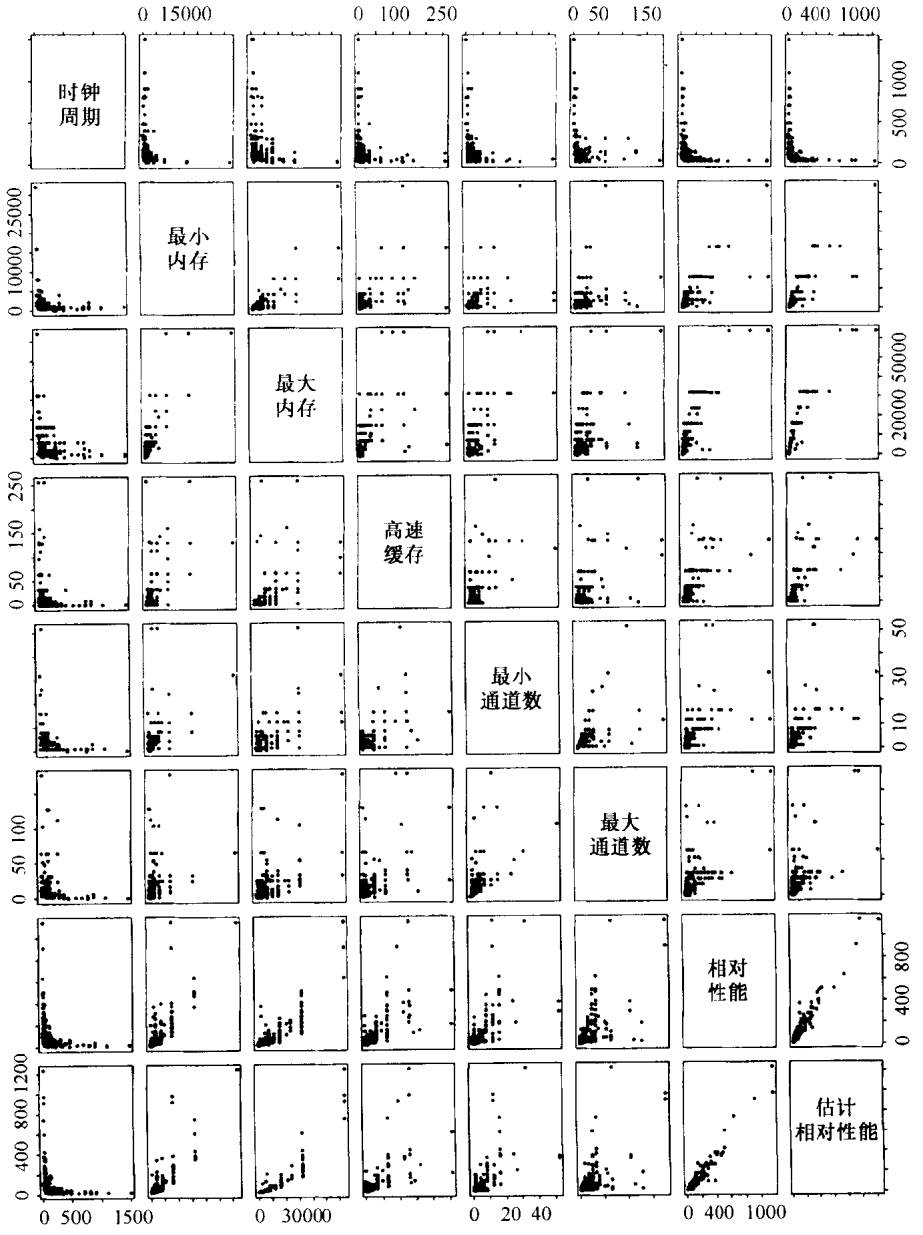


图3-13 计算机CPU数据的散点图矩阵

另一种显示多元数据的完全不同的方法是使用图标（icon），图标是一些很小的图，其不同特征的大小是由特定变量的值决定的。其中星形图标是最为流行的。在星形图标中，相对于原点的不同方向对应不同的变量，投影在这些方向上的半径长度对应于变量的幅度。图3-15显示了一个例子。其中的数据是这样得到的，首先钻入地球表面深层并等间隔的采集53个矿石样本，然后测定出这些样本的12种化学属性。

另一种常见的图标图是Chernoff面容（Chernoff's face），很多这方面的入门级教材经常讨论它。在这些图中，卡通画面部特征的尺寸（鼻子的长度、笑的程度、眼睛的形状等等）代表了各个变量的值。这种方法所依据的原则是，人类的眼睛特别善于识别和区分面容。这

种方法非常有趣，但这种图很少用于严肃的数据分析，因为在实践中当卡通面容超过一定数量后，这种方法工作的就不那么好了。通常，图标显示只适用于少数实例的情况，因为需要用眼睛分别浏览每一个实例。

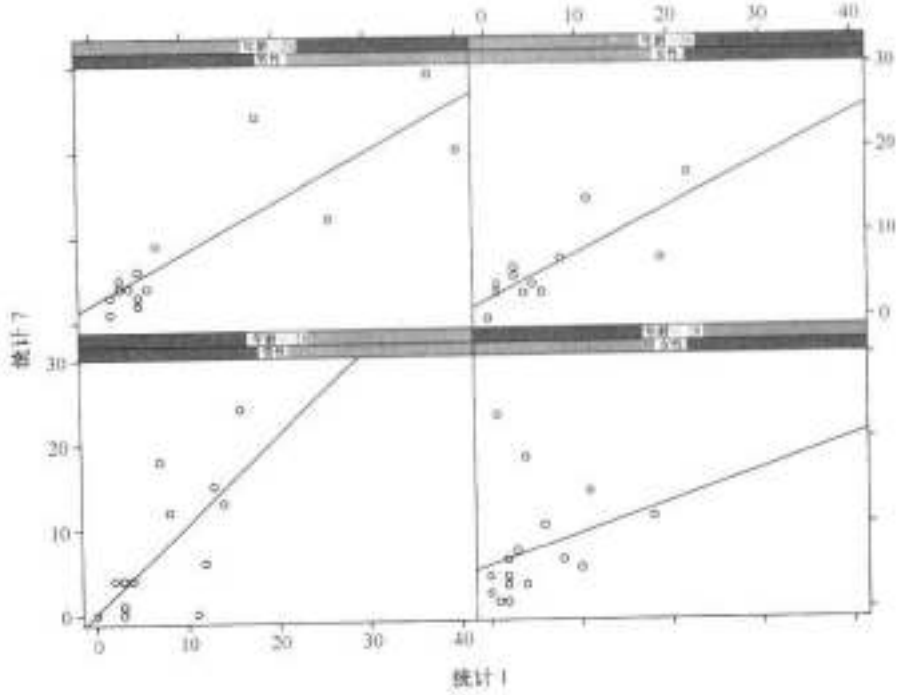


图3-14　癫痫病发作情况数据的格架图

平行坐标图以平行的坐标轴来显示变量，用直线连接起每个实例的值，这样每个实例便被表示为一条折线。图 3-16 显示了这样的一幅图，所画的是 58 位癫痫病人发作次数的四次重复测量结果，每次的测量时段是两周。这些数据明显是倾斜的，可以用泊松分布（参见附录）来对其建模。由于数据集合不是很大，因此我们可以看出每个病人的轨线。

另一种代表维度的方法是使用彩色。线型也可以达到相同的目的，就像上面的平行坐标图那样。

用一种方法完全解决所有的多元数据显示问题是不可能的。哪种方法最适用取决于具体数据和要寻找的结构。

3.6　主分量分析

72 / 74

散点图把多元数据投影到仅由两个变量定义的二维空间。这使我们可以成对的分析变量间关系，不过这样的简单投影可能会隐藏更复杂的关系。要分析这些更复杂的关系就要沿不同方向进行投影，方向是由变量的加权线性组合所定义的（例如沿 $2x_1 + 3x_2 + x_3$ 所定义的方向）。

如果仅有几个变量，那么手工旋转数据分布搜索有趣方向还是可行的。然而对于较多变量的场合，最好还是让计算机进行搜索。为了实现这个目的，我们必须定义“有趣”投影的特征，这样计算机才知道什么时候已经找到了要找的东西。投影跟踪方法（projection pursuit method）

就是建立在让计算机去寻找感兴趣方向这一一般原则基础上的。（然而，这种技术所需的运算量非常大，在第 11 章中讨论回归时我们将回过头来继续介绍投影跟踪法。）

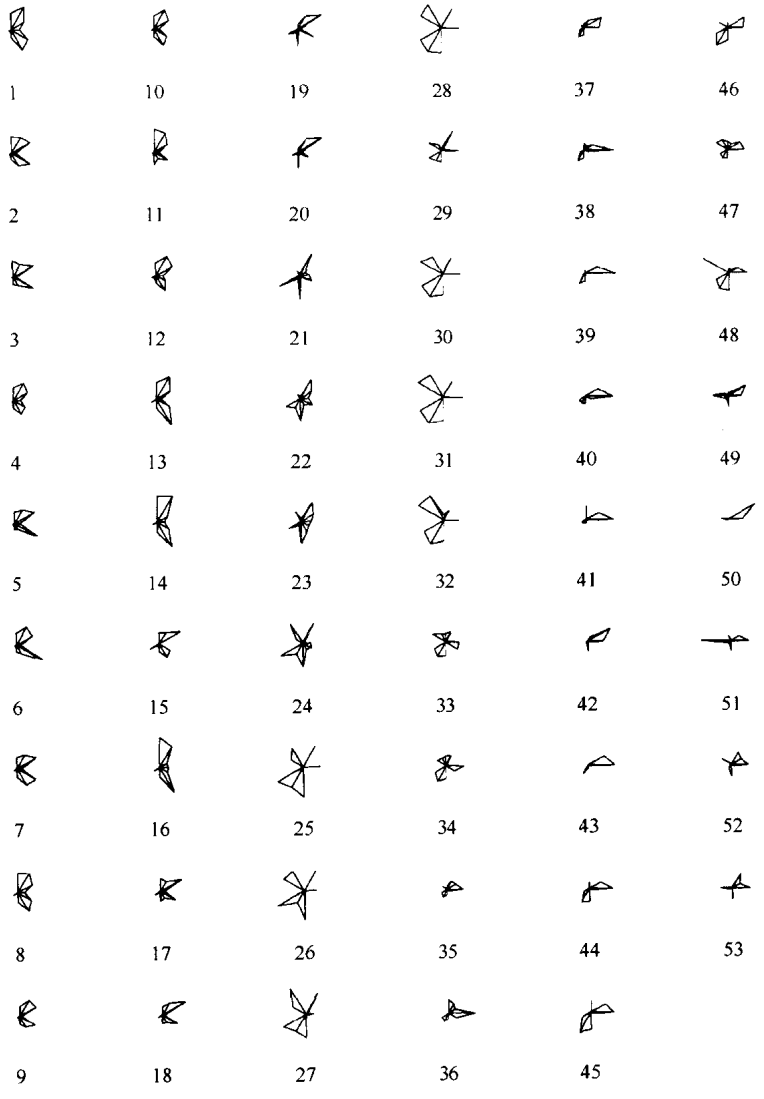


图3-15 星图示例

然而，对于一种特例情况——对“有趣”方向的一种特殊定义——可以得到计算效率很高的明确解。也就是当我们向这个二维平面投影时，数据点和它们在该平面上的投影的差异平方和比向其他平面投影时更小。（为了方便起见，这里我们使用二维投影，但通常我们可以使用 k 维投影，$1 \leqslant k \leqslant p-1$。）可以证明由以下线性组合决定的二维平面就是这样的平面：（1）具有最大样本方差的变量的线性组合；（2）与第一个线性组合无关的具有最大方差的线性组合。因此这里是按数据的最大变化性（maximum variability）来定义“有趣”性的。

当然，我们可以进一步推进这一过程，寻找与所有已选线性组合无关的使方差最大化的其他线性组合。通常，如果幸运的话，我们仅可以发现几个可以精确描绘数据的这种线性组合（“分量”）。下文将介绍这一过程的数学描述。这里我们的目标是捕捉数据的内在变化性。

这是降低数据集合维度的一种有效方式，这样既可以使数据易于解释，又可以避免过度拟合并为后续分析做准备。

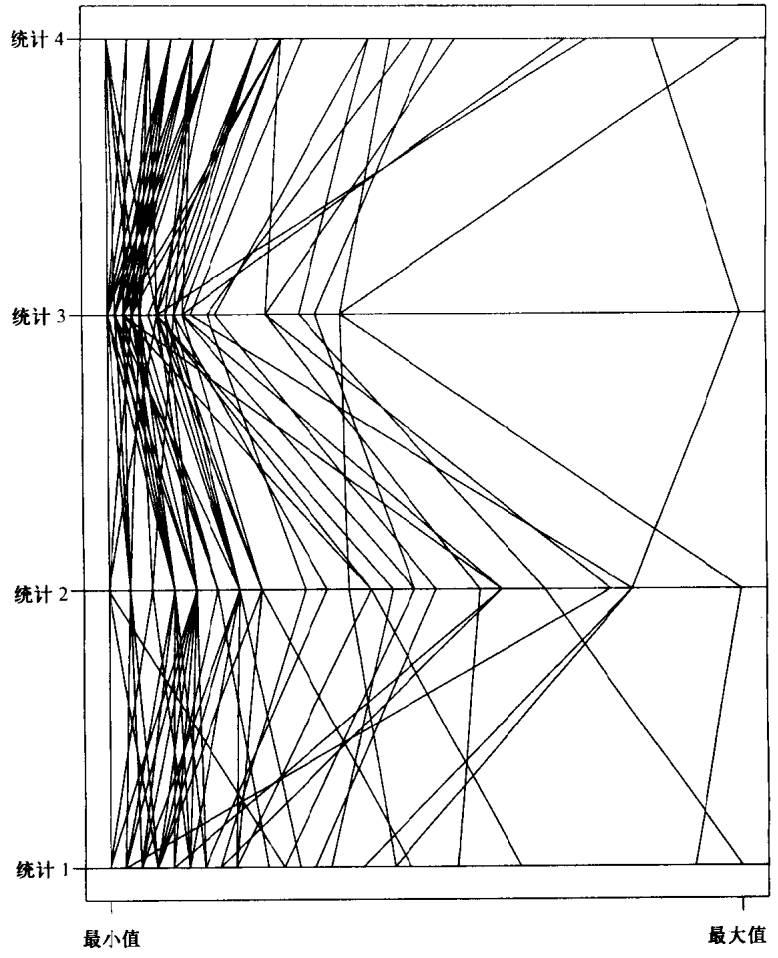


图3-16　癫痫病发作数据的平行坐标图

假定 **X** 是一个 $n\times p$ 的数据矩阵，行代表实例（每行是一个数据向量 **x**(*i*)），列代表变量。严格来说，矩阵的第 i 行是数据向量 **x**(*i*)的转置 $\mathbf{x}^T$，因为习惯上是把数据向量看作是 $p\times 1$ 的列向量而不是 $1\times p$ 的行向量。此外，假定 **X** 是以均值为中心的（mean-centered），这样每个变量的值都是相对于这个变量的样本均值的（也就是说每列都已经减去了这个估计均值）。

75
~
77

设 **a** 为当 **X** 沿其投影时会使方差最大化的 $p\times 1$ 列向量（现在还不知道）。那么任何特定数据向量 **x** 的投影就是线性组合 $\mathbf{a}^T\mathbf{x}=\sum_{j=1}^{p}a_j x_j$。注意我们可以将 **X** 中所有数据向量投影到 **a** 的投影值表示为 **Xa**（$n\times p$ 乘 $p\times 1$，产生一个 $n\times 1$ 的投影值列向量）。此外，我们可以将沿 **a** 投影的*方差*定义为：

$$\begin{aligned}\sigma_{\mathbf{a}}^2 &= (\mathbf{Xa})^T(\mathbf{Xa})\\ &= \mathbf{a}^T\mathbf{X}^T\mathbf{Xa}\\ &= \mathbf{a}^T V\mathbf{a}\end{aligned} \tag{3.8}$$

其中 $V=\mathbf{X}^T\mathbf{X}$ 是数据的 $p\times p$ 协方差矩阵（由于 $\mathbf{X}$ 的均值为 0），和第 2 章所定义的一样。因此，我们可以将 $\sigma_{\mathbf{a}}^2$（我们要最大化的投影数据的方差（标量））表示为 $\mathbf{a}$ 和数据协方差矩阵 $\mathbf{V}$ 的函数。

当然，直接最大化 $\sigma_{\mathbf{a}}^2$ 是没有意义的，因为通过增大 $\mathbf{a}$ 的各元素可以无限制的增大 $\sigma_{\mathbf{a}}^2$。所以必须强加上一些约束，我们对向量施加一个标准化约束使 $\mathbf{a}^T\mathbf{a}=1$。

有了这个标准化约束，我们便可以把这个最优化问题转变为使以下这个量最大化：

$$u=\mathbf{a}^T\mathbf{V}\mathbf{a}-\lambda(\mathbf{a}^T\mathbf{a}-1) \tag{3.9}$$

这里 λ 是拉格朗日乘子（Lagrange multiplier）。然后对 $\mathbf{a}$ 求导得到：

$$\frac{\partial u}{\partial \mathbf{a}}=2\mathbf{V}\mathbf{a}-2\lambda\mathbf{a}=0 \tag{3.10}$$

这样便得到了熟悉的特征值形式：

$$(\mathbf{V}-\lambda\mathbf{I})\mathbf{a}=0 \tag{3.11}$$

因此，第一主分量 $\mathbf{a}$ 就是与协方差矩阵 $\mathbf{V}$ 的最大特征值联系的特征向量。而且，第二主分量（和具有最大投影方差的第一主分量正交的方向）就是 $\mathbf{V}$ 的第二大特征值所对应的特征向量，依次类推（第 k 大特征值的特征向量对应于第 k 个主分量方向）。

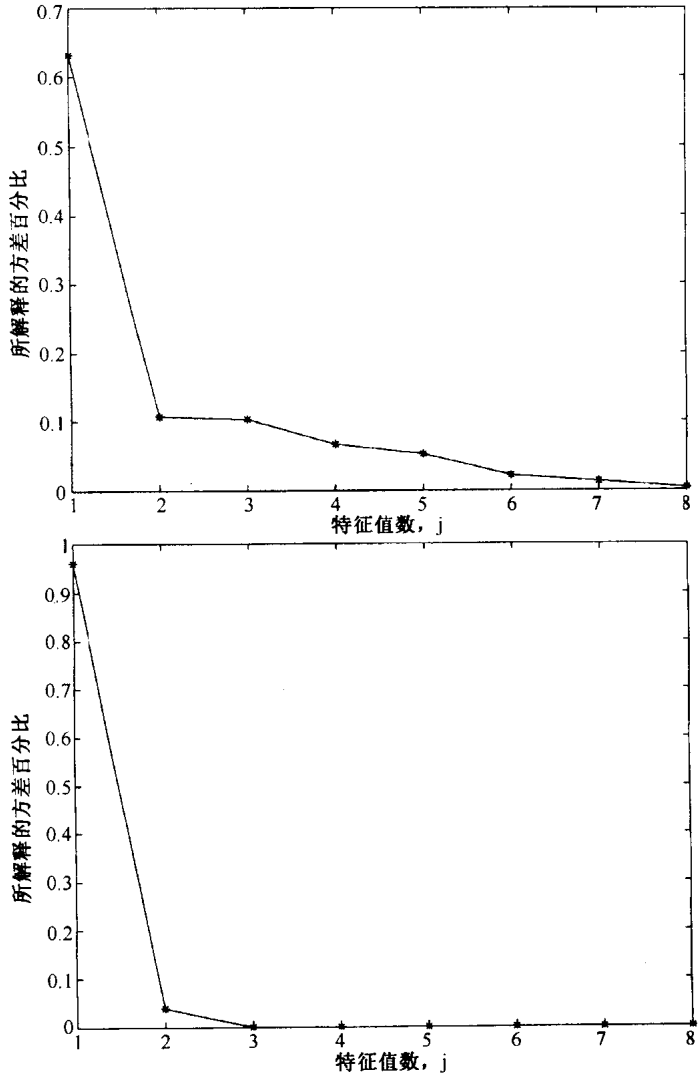
当然，在实践中我们可能对投影到二维以上的情况感兴趣。这种投影模式的一个基本属性 78
是如果数据投影到前 k 个特征向量，那么投影数据的方差可以被表示为 $\sum_{j=1}^{k}\lambda_j$，其中 λ_j 是第 j 个特征值。同样，只使用前 k 个特征值来近似真实数据矩阵 X 的误差平方可以被表示为：

$$\frac{\sum_{j=k+1}^{p}\lambda_j}{\sum_{l=1}^{p}\lambda_l} \tag{3.12}$$

因此，选择主分量的适当个数 k 的一种方法是增大 k 直到这个误差平方小于某个可接受的程度。对于多维数据集，各个变量经常是密切关联着的，因此以相当少的主分量（比如说 5 或 10 个）捕获 90%或更多的数据变化性并不稀奇。

基于这一背景的一种有效可视手段是*碎石堆图*（scree plot）——显示出每个特征值所解释出的方差的量。它必然是随分量数非上升的，而且我们希望它呈现出突然的向零下降。对前面介绍的计算 s 机 CPU 数据的相关矩阵主分量分析得到的特征值正比于 63.26、10.70、10.30、6.68、5.23、2.18、1.31 和 0.34（参见图 3-17）。从第一到第二特征值的变化是很剧烈的，但之后便逐步下降了。（对应于八个变量的第一分量的权是（0.199、-0.365、-0.399、-0.336、-0.331、-0.298、-0.421、-0.423）。注意，给每个变量的权是大体相似的，但给第一个变量（时钟周期）的权的符号与给其他变量的是相反的。）如果，我们用协方差矩阵代替相关矩阵进行分析，那么取值范围较大的变量往往会占据优势。对于本例中的这些数据，给内存的值要远远大于给其他变量的值。（这是因为它是以千个字节为单位的。如果用兆字节来表示就不是这样了——这就是不同量纲变量缩放所造成的任意性（参见第 2 章））。利用协方差矩阵的主分量分析给出的变化性正比于 96.02、3.93、0.04、0.01、0.00、0.00、0.00 和 0.00（参见图 3-17）。这里从第一个分量到第二个分量的下降是非常显著的——事实上，数据中的变化性确实几乎完全可以按内存容量的不同来解释。然而，通常情况下不会出现这样明显

的下降——数据其余的变化性不可以都归为随机变化。因此选取多少个分量是有相当任意性的。占总方差多大比例才足以描述数据取决于具体的应用领域。在某些场合，描述 60%方差的前几个分量就足够了，但在其他场合也许希望 95%或更多。



78
~
80

图3-17　计算机CPU数据集的碎石堆图。上图显示的是相关矩阵的特征值，下图是协方差矩阵的情形

当在进一步分析之前进行主分量分析时，选择不能很好地解释数据变化性的很小分量数是有危险的。这样会丢失信息，并且不能保证丢失的信息与进一步的分析是无关的。（事实上即使保留的分量可以很好地说明变化性，但只要达不到 100%，上述判断就是正确的，即这样做仍有危险。）例如，我们可能在对数据进行分类之前进行主分量分析。由于减少维数的目的和分类是有些不同的，因此被缩减后的几个分量就有可能丢失了各个类间的重要差异信息——在第 9 章的末尾我们将介绍一个这样的例子。类似地，对于许多属于两个（或更多）类的多维数据集来说，预先的主分量分析都可能彻底抹杀类分布间的差异。另一方面，对于具有很多说明性变量的回归问题（第 11 章），除非数据集非常庞大，否则也可能造成使系数估计不稳定的

问题。在进行回归分析之前有时会进行主分量分析，目的是把大量的说明性变量减少为几个变量的线性组合。

尽管有无法提取相关信息的危险，但是主分量分析仍然是一种强大而且有价值的工具。因为它是建立在线性投影和最小化方差（或误差平方和）基础之上的，所以可以显式的进行各种数字操作，不需要做任何迭代搜索。从特征向量公式直接计算主分量解的复杂度大体为 $O(np^2 + p^3)$（np^2 用于计算 $\mathbf{V}$，p^3 用于求解 $p \times p$ 矩阵的特征值方程）。这意味着该方法适用于记录数 n 很大的数据集（但维度 p 的伸缩性就不这么好了）。正如刚才的例子所演示的，当进行主分量分析时，不论是对于协方差矩阵还是相关矩阵，该方法对原始变量的重新调节（rescale）不是恒定的。因此应根据分析的目的采取合适的步骤。典型地，如果不同的变量测量不同的属性（如身高、体重和肺活量），那么将重新调节数据，因为不然的话，直接主分量分析的结果将依赖于每个属性所选择的单位。

为了演示主分量分析的简单图形应用，图 3-18 显示了 17 种药丸在前两个主分量所决定的平面上的投影（以数字表示）。对每种药丸的六种测量是特定比例（分别为 10%、30%、50%、70%、75%和 90%）的该种药丸的溶解时间。从图中可以清楚地看出有一种药丸（位于图的右下角）与其他药丸的区别很大，与其他点相距很远。 81

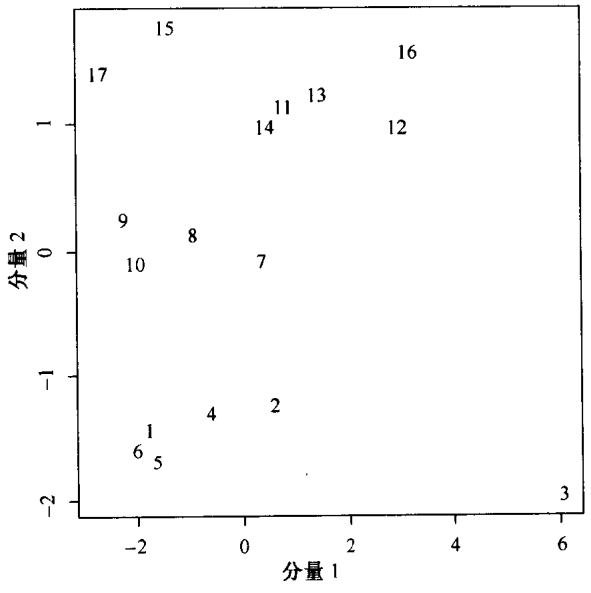


图3-18 在前两个主分量上的投影

有时我们可以从定义主分量的权模式（或者有时叫加载（loading）模式）中获得很多内部细节。Huba et al.（1981）收集了 1684 位洛杉矶学生消费 13 种合法和不合法兴奋性物质的数据，这些物质有：香烟、啤酒、红酒、酒精、可卡因、镇定剂、用于达到高潮的药房药剂、吗啡和其他鸦片制剂、大麻、麻药、吸入性麻醉剂（例如 glue）、迷幻药和安非他明。Huba 等人把使用每种药的情况定为：1（从未尝试）、2（用过一次）、3（用过几次）、4（用过好多次）和 5（经常使用）。按照这些变量的顺序，主分量分析的第一分量权是（0.278，0.286，0.265，0.318，0.208，0.293，0.176，0.202，0.339，0.329，0.276，0.248，0.329）。这一分量赋给每个变量的权是大体相等的，因此可以被认为是衡量学生多么频繁地使用这些物质的 82

一个一般尺度。因此，学生之间的最大区别是通过他们使用这些物质的频率来衡量的，不论他们使用的究竟是哪种物质。

第二分量的权为（0.280，0.396，0.392，0.325，-0.288，-0.259，-0.189，-0.315，0.163，-0.050，-0.169，-0.329，-0.232）。这个权非常有趣，因为它赋给所有合法物质的权都为正，赋给所有非法物质的权都为负：因此，一旦我们控制了总体的药物使用情况，那么学生之间的主要区别就是他们使用的药物是合法还是非法的。这正是在数据挖掘中我们希望发现的关系。

另一种统计技术，*因素分析*（factor analysis），经常与主分量分析混淆，但是这两种技术的目的是不同的。正如前面所介绍的，主分量分析是将数据向新的变量转换。而后我们可以仅选取这些变量作为对数据的一种充分描述。另一方面，因素分析是一种数据模型，其基本思想是我们可以把测量变量 X_1，…，X_p 定义为更少数量数 m（$m<p$）个“潜在”因素（未观察的或不能明确测量的）的线性组合。因素分析的目的就是揭开这些隐藏变量的信息。

我们可以把 $\mathbf{F}=(F_1,\cdots,F_m)^T$ 定义为代表未知潜在变量的 $m\times 1$ 列向量，其值为 $\mathbf{f}=(f_1,\cdots,f_m)$。然后把已经测量的数据向量 $\mathbf{x}=(x_1,\cdots,x_p)^T$（定义为 $p\times 1$ 的列向量）看作是是 $\mathbf{f}$ 的线性函数，定义为：

$$\mathbf{x}=\mathbf{\Lambda f}+\mathbf{e} \tag{3.13}$$

其中$\mathbf{\Lambda}$是一个 $p\times m$ 的*因素加载*（factor loading）矩阵，它给出了每个因素对每个显性变量的贡献的权。$p\times 1$ 的向量 e 的各元素是不相关的随机变量，有时被称为*特殊因素*（specific factor），因为它们只对单个的显性（观察）变量 X_j（$1\leqslant j\leqslant p$）有贡献。因素分析是第 9 章介绍的结构化线性关系模型的特例，所以在此不介绍其详细的估计过程。因素分析是这种形式模型的最早结构，它具有特殊的位置，不仅因为它的历史，还因为它一直是这种模型中被应用最广的形式之一。

人们也曾对因素分析方法有很多争议，部分原因是它的解对于不同变换不是恒定的。从
83 公式 3.13 中容易看出可以通过 $m\times m$ 的正交矩阵 $\mathbf{M}$ 来定义新的因素，只要满足 $\mathbf{x}=(\mathbf{\Lambda M})(\mathbf{Mf})+\mathbf{e}$。这相当于在这些因素所跨越的空间中旋转这些因素。因此，提取的因素本质上是不唯一的，除非强加额外的约束。有很多种约束可用于一般性的应用，包括使提取因素的权尽可能接近 0 或 1，这样便可以根据因素的子集尽可能清楚地定义变量。

3.7　多维缩放

在前一节我们描述了如何使用主分量分析将多元数据投影到数据可以最大限度分布的平面上。这使我们可以在牺牲最少信息的条件下使分析数据可视化。这种方法只对被测量变量所跨越区域的二维线性子空间内的数据是有效的。如果数据集本质上是二维的，但却不是“平坦的”，而是弯曲的或者失真的，那么会怎样呢？（想像一张弄皱的纸，本质上是二维的，但占用了三维。）在这种情况下，主分量分析很有可能无法找出潜在的二维结构。对于这样的场合，多维缩放是很有帮助的。多维缩放在尽可能远的保留数据点与点间的距离的同时，争取在更低维的空间内来表示数据。因为我们最关心的是二维表示，所以我们将讨论主要限制在这种情况中。它可以直接扩展到更多维显示的情况。

多维缩放的方法有很多，区别在于如何定义所保持的距离；如何映射；以及如何进行计

算。可以把主分量分析当作一种基本形式。在这种方法中点之间的距离是欧氏的（或者Pythagorean），而且它们是被映射到也是用欧氏标距测量的压缩空间中。原始数据点与它们投影点间的距离平方和为衡量这种表示的质量提供了一种尺度。其他多维缩放方法也有其对应的表示质量尺度。

因为多维缩放方法力争保持各点间的距离，所以我们可以把这些距离作为分析的起点。 84
也就是说，我们不需要知道被分析对象的任何变量测量值，只要知道以距离衡量的对象相似性就可以了。举例来说，数据可能是通过让回答者比较两个对象间的相似性来采集的。（这样的一个经典例子是用来显示代表不同字母的摩尔斯代码被搞混次数的矩阵。这里不存在“变量”，只是用一个“相似性”矩阵来度量一个字母被搞混成另一个字母的频繁程度。）这个过程的最终目标也是一样的——数据点在二维空间中的布局。从某种意义来说，我们是利用这些对象和回答者来决定在什么样的维上测量“相似性”。多维缩放方法广泛应用在心理测试和市场调查等领域中，用来理解对象间的关系和相似性。

从一个 $n\times p$ 的数据矩阵 $\mathbf{X}$ 我们可以求出一个 $n\times n$ 的数据矩阵 $\mathbf{B}=\mathbf{X}\mathbf{X}^T$。（因为这样做的时间和空间复杂度都是 $O(n^2)$，所以这种方法对于非常大的 n 值显然是不适用的）。由此可以看出第 i 个和第 j 个对象间的欧氏距离为：

$$d_{ij}^2=b_{ii}+b_{jj}-2b_{ij} \tag{3.14}$$

如果我们可以把这种关系反过来，那么对于一个给定的距离矩阵 $\mathbf{D}$（通过计算欧氏距离或以其他途径根据原始数据点导出），我们就能求出 $\mathbf{B}$ 的元素。然后可以把 $\mathbf{B}$ 因子化以得到点的坐标。一种因子化 $\mathbf{B}$ 的方法是按照特征向量进行的。如果我们选择和两个最大特征值相联系的因子，那么我们就可以得到一个最大可能保持数据结构的二维表示。

这个过程的可行性关键在于能否逆转等式 3.14。不幸的是，如果不强加额外的约束是不可能的。因为平移均值和旋转点布局并不影响点间的距离，所以对于任意的给定距离集合，都存在无数个解，只是点布局的位置和方向有所不同。

一种充分的约束是假定所有变量的均值为零。也那就是说，假定对于所有 $k=1,\cdots,p$，都有 $\sum_i x_{ik}=0$。这意味着 $\sum_i b_{ij}=\sum_j b_{ij}=0$。现在，通过汇总等式 3.14，首先对 i，然后对 j，最后对 i 和 j，我们得到：

$$\begin{aligned}\sum_i d_{ij}^2&=tr(\mathbf{B})+nb_{jj}\\ \sum_j d_{ij}^2&=tr(\mathbf{B})+nb_{ii}\\ \sum_{ij} d_{ij}^2&=2ntr(\mathbf{B})\end{aligned} \tag{3.15}$$

85

其中 $tr(\mathbf{B})$是矩阵 $\mathbf{B}$ 的迹。第三个等式以 d_{ij}^2 表示了 $tr(\mathbf{B})$，第一和第二个等式以 d_{ij}^2 和 $tr(\mathbf{B})$（因此也就是以 d_{ij}^2 本身）表示了 b_{jj} 和 b_{ii}。把这些插入到等式 3.14 中便把 b_{ij} 表示成了 d_{ij}^2 的函数，这样便得到了所需的逆转。

这种过程被称为主坐标（principal coordinate）法。可以证明对数据矩阵 $\mathbf{X}$（和矩阵 $\mathbf{X}^T$ 的因子化）的主分量分析所计算出的主分量值与上面缩放分析的坐标是相同的。

当然，如果矩阵 $\mathbf{B}$ 不是按 $\mathbf{X}\mathbf{X}^T$ 的乘积产生的，而是通过其他途径（如变量对间的主观

差异）产生的，那么不能保证所有的特征值都是非负的。如果负特征值的绝对值很小，那么可以忽略它们。

经典的多维到二维缩放寻找使下式最小化的到二维空间的投影：

$$\sum_i \sum_j (\delta_{ij} - d_{ij})^2 \tag{3.16}$$

其中 δ_{ij} 是 p 维空间中数据点 i 和数据点 j 间的观察到的距离，d_{ij} 是二维空间中代表这些对象的点之间的距离。通过这种表示，使这种过程可以被立刻推广。有了距离或相异度，通过这样或那样的推导，我们可以在二维空间中找到一种点分布使差异平方和 $\sum_i \sum_j (\delta_{ij} - d_{ij})^2$ 最小化。这样便缓解了必须用投影来寻找布局的局限。对于这种情况，通常不可能得到精确的代数解，因此必须用数值方法：我们仅有一个要最小化的含 $2n$ 个参数（点在二维空间中的坐标）的函数。

评分函数 $\sum_i \sum_j (\delta_{ij} - d_{ij})^2$ 衡量了导出布局中的点间距离和原始点间距离的匹配程度，它对于旋转和平移是恒定的。然而，它对于数据的再缩放并不是恒定的：如果把 δ_{ij} 乘以一个常数，我们将得到同样的解，但是得到的 $\sum_i \sum_j (\delta_{ij} - d_{ij})^2$ 值是不同的。为了合理

86 的比较不同的情况，我们把 $\sum_i \sum_j (\delta_{ij} - d_{ij})^2$ 除以 $\sum_{i,j} d_{ij}^2$，这样便得到标准的残差平方和。一种常见的评分函数是取该量的平方根，称为*应力*（stress）。应力的一种变体是 s 应力（sstress），定义为：

$$\sqrt{\sum_i \sum_j (\delta_{ij}^2 - d_{ij}^2)^2 / \sum_i \sum_j d_{ij}^4} \tag{3.17}$$

这些尺度实质上假定了二维布局中的距离和原始相异性的区别是由随机偏差和失真所造成的——也就是 $d_{ij} = \delta_{ij} + \varepsilon_{ij}$。也可以建立更加完善的模型。例如，我们假定 $d_{ij} = a + b\delta_{ij} + \varepsilon_{ij}$。现在这个过程必须分为两个阶段。从提出的布局开始，应用给定的相异性对二维空间中的距离 d_{ij} 进行回归，得到对 a 和 b 的估计。然后寻找使以下应力最小化的新 d_{ij} 值：

$$\sqrt{\sum_i \sum_j (d_{ij} - a - b\delta_{ij})^2 / \sum_i \sum_j d_{ij}^2} \tag{3.18}$$

然后重复此过程直到达到了满意的收敛结果。

像上面这样对相异性建模的多维缩放方法被称为标距（metric）法。然而有时需要一种更通用的方法。例如，我们可能并不知道精确的相似度，只有它们的相似程度排序（对象 A 与 B 比 B 与 C 更相似，等等）；或者我们不能假定在 d_{ij} 和 δ_{ij} 之间的关系符合特定的形式，只能确定存在某种单调的关系。这便要求使用一种类似于上一段中描述的两阶段过程，不过使用的是一种被称为单调回归的技术，而不是简单的线性回归，这便是非标距（non-metric）多维缩放。这里术语非标距的含义是指这种方法只保持原来的顺序关系。

多维缩放是显示数据以揭示其结构的一个强大工具。然而，和本章描述的其他图形方法一样，如果数据点太多的话，结构就会变得模糊不清。此外，由于多维缩放对数据应用了非常复杂精密的变换（所以比简单的散点图或主分量分析更加复杂），所以可能会引入假象（artifact）。尤其是在有些情况下，对象很相似时比它们相差迥异时可以更精确地决定对象

间的相似性。不妨以机械制品样式的演变过程为例。在很短时间内生产出的那些产品彼此间很可能会有很多共同点，而那些生产时间间隔很长的产品可能就没什么共同点。结果可能在多维缩放图中出现一个感应性弯曲，而我们希望得到更直一些的直线。这种现象称为*马蹄铁效应*（horseshoe effect）。 87

图 3-19 所示为一幅通过用非标距缩放使等式 3.17 的 s 应力评分函数最小化而生成的图形。这些数据来自于一项对英语方言的研究。该研究对 25 个乡村进行两两比较，依据是这两个乡村用不同单词表达 60 个内容（item）的百分比。表 3-1 列出了这些乡村，和它们所在的郡。从图中可以看出同一郡（因此地理上比较近）的乡村往往使用相同的单词。

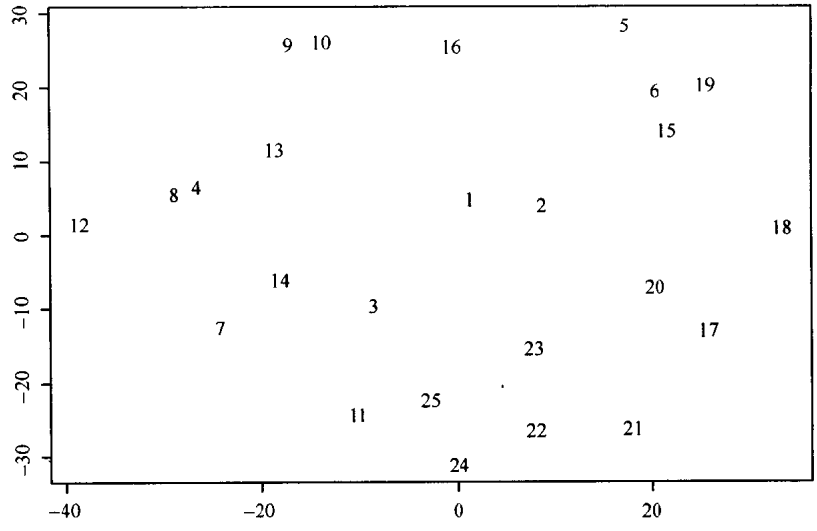


图3-19 反映乡村方言相似点数据的多维缩放图 88

表 3-1 图 3-19 显示的具有相似方言的 25 个乡村的编码、名称和所在郡

1	North Wheatley	诺丁汉郡
2	South Clifton	诺丁汉郡
3	Oxton	诺丁汉郡
4	Eastoft	林肯郡
5	Keelby	林肯郡
6	Wiloughton	林肯郡
7	Wragby	林肯郡
8	Old Bolingbroke	林肯郡
9	Fulbeck	林肯郡
10	Sutterton	林肯郡
11	Swinstead	林肯郡
12	Crowland	林肯郡
13	Harby	莱斯特郡
14	Packington	莱斯特郡

（续）

15	Goadby	莱斯特郡
16	Ullesthorpe	莱斯特郡
17	Empingham	拉特兰郡
18	Warmington	北安普敦郡
19	Little Harrowden	北安普敦郡
20	Kislingbury	北安普敦郡
21	Sulgrave	北安普敦郡
22	Warboys	亨廷登郡
23	Little Downham	剑桥郡
24	Tingewick	白金汉郡
25	Turvey	贝德福德郡

多维缩放方法通常在二维空间中显示数据点。如果变量也是在这一空间描述的（假定数据是向量形式的），那么就可以清楚地看出数据点和变量间的关系。如果原始变量所定义的空间和用于显示数据的空间之间存在复杂的非线性关系，那么表示原始变量就不是一种微不足道的任务了。既显示数据点又显示变量的图被称为双用图（biplots）。这里的“双用（bi）”
89 代表同时显示两种模式——数据点和变量——并不是说它是二维显示。事实上，已经开发出了三维的双用图。涉及非线性变换的多维缩放形式产生的是非线性的双用图。甚至还可以为范畴型数据产生双用图，在这种场合下，变量的级别（level）是用图中的区域（range）表示的。要有效地解释多维双用图显示需要丰富的实践经验。

3.8 补充读物

John Tukey 的《Exploratory Data Analysis》（探索性数据分析）（Tukey,1977）一书的出版为探索性数据分析奠定了基础并赢得了声望。从此以后，随着计算机技术的飞速发展和简单易懂的精确图形显示产品的出现，这种方法不断地发展。现代的数据可视化技术为探索各种结构提供了强有力的工具。关于图形方法的著作包括 Tufte（1983）、Chambers et al.（1983）以及 Jacoby（1997）。Wilkinson（1999）是新加入到可视化文献中的一本特别有趣的著作，该书以新颖的语言分析了很多通用的数据可视化技术。

Asimov（1985）、Becker, Cleveland and Wilks（1987）、Cleveland and McGill （1988）、Buja, Cook and Swayne（1996）重点介绍交互式的动态方法。Silverman（1986）、Scott（1992）、和 Wand and Jones（1995）介绍了显示一元分布的平滑方法，以及对多元情况的扩展。Carr et al.（1987）讨论了针对庞大数据集的散点图技术。Wegman（1990）讨论了平行坐标。范畴型数据在某种程度上比实数值数据更难可视化，因此，范畴型数据的可视化技术并未得到广泛发展和应用。尽管如此，Blasius and Greenacre（1998）对范畴型数据的可视化和探索性数据分析的最新发展作了一个全面和有用的回顾。Cook and Weisberg（1994）描述了图形技术在回归建模方面的应用。

Card, MacKinlay and Shneiderman（1999）汇编了关于“信息可视化”这一主题的大量论文，介绍了很多如何显示各种复杂异质（heterogeneous）数据集的技术。Keim and Kriegel

（1994）介绍了一个特别为探索数据库而设计的系统。 90

多维缩放已经发展成为一个很大的领域。这方面的书籍包括 Davidson（1983）、Cox and Cox（1994）。Gower and Hand（1996）详细地讨论了双用图。

CPU 数据来源于 Ein-Dor and Feldmesser（1987），在 Hand et al.（1994）的 325 号数据集中有其拷贝。英语方言的数据来源于 Morgan（1981），在 Hand et al.（1994）的 145 号数据集中有其拷贝。Thall and Vail（1990）和 Hand et al.（1994）都给出了癫痫病发作数据。Chernoff（1973）介绍了图标图所显示的矿石数据。 91

第 4 章　数据分析和不确定性

4.1　简介

这一章我们将集中讨论不确定性（uncertainty）和处理不确定性的方法。不仅从真实世界映射到数据库的过程极少是十全十美的，而且被映射的领域——真实世界本身——也充满了模糊和不确定性。处理不确定性的基本工具是概率，因此我们首先从有关的概念入手，然后再说明如何使用概率理论来建立统计模型。4.2 节简要的讨论了概率计算和概率解释的差异，集中讨论了两种主要的解释方法：频率论的和主观法（贝叶斯）。4.3 节把讨论范围延伸到随机变量的概念，重点讨论了存在于多个随机变量间的关系。

样本的概念是许多数据挖掘行为的基础。有时数据库中包含的就是来自所有可能情况的一个样本；4.4 节对此进行了探索，解释了为什么很多时候工作在样本上就足够了。4.5 节描述了估计（estimation）——推广到样本数据之外、求解数据模型参数的过程。尤其是，我们比较细致地讲解了最大似然（maximum likelihood）和贝叶斯这两种估计方法的基本原理。4.6 节讨论了与估计方法密切相关的一些话题，即如何根据观测到的数据评价假设的质量。4.7 节重点讨论了从数据中抽取样本的一些系统方法。4.8 节对本章内容进行了总结，4.9 节推荐了一些更加详细的读物。

4.2　处理不确定性

描述不确定性以及其相关概念的词汇异常丰富，这说明了这个概念的普遍性。例如概率（probability）、偶然性（chance）、随机性（randomness）、运气（luck）、意外（hazard）和天数（fate）仅仅是一部分。不确定性是无所不在的，这要求我们必须采取措施来对付它们：对不确定性建模几乎是所有数据分析工作的一个必不可少的部分。甚至，有些情况下我们的主要目的就是对不确定性和数据的随机特征建模。我们已经对不确定性有了非常深入的了解，这是最伟大的科学成就之一。今天人们不再用“上帝的反复无常”来解释这个世界的难以预测性，取而代之的是数学、统计和基于计算机的各种模型，因为这些工具使人们可以理解并处理不确定事件。我们甚至可以尝试那些看起来不可能的不确定事件，并对其进行预测。对于一个数据挖掘者来说，预测可能意味着对未来事件的预测（这种情况下的不确定性概念是非常熟悉的），也可能意味着对某个变量做非时间意义上的预测，这个变量的真实值因某种原因不为我们所知（例如，仅根据描述出的症状诊断一个人是否患了癌症）。

产生不确定性的原因很多。我们的数据可能仅是我们要研究的总体的一个样本，所以我们不能确定不同样本相互之间以及样本与整个总体之间的差异程度。或许我们的目标在于根据今天的数据来对明天的情况做出预测，那么我们的结论是受将来结果的不确定性支配的。或许我们对某些情况并不知晓或不能观察到某个值，因而必须把我们的想法建立在我们的“最好猜测”之上。等等。

目前已经建立了很多用于处理不确定性和未知性的基本概念。这其中，迄今为止应用最

广的是概率理论。模糊逻辑是另一个应用很广的理论，但这个领域——以及与之密切相关的一些领域，比如可能性理论（possibility theory）和粗糙集（rough sets）——还存在相当多的争议：缺少概率理论所具备的完整理论框架，而且并不像概率理论那样被广泛接受和应用。可能有一天这些思想会奠定坚实的基础，并被广泛地使用，但因为它们目前还处于不确定状态，所以本书不对其作进一步的讨论。

把概率论（probability theory）和概率计算（probability calculus）区分开来是有意义的。前者致力于如何解释概率，而后者致力于如何操纵概率的数学表示。（不幸的是，并非所有
94 的教科书都明确地区分了这两个术语——经常看到关于概率计算的书籍中出现“概率论简介”这样的标题。）这个区分之所以重要是因为这样我们可以把那些具有统一共识的领域（概率计算）从那些观点不同的领域（概率理论）中分离出来。概率计算是数学的一个分支，它建立在精确定义并被普遍接受的一些公理（由前苏联数学家 Kolmogorov 在三十年代提出）之上；它的目标是探索那些公理的推论。（有一些领域使用了不同系列的公理，但那是专门针对某个领域的，一般不会关系到数据挖掘问题。）另一方面，概率理论为关于如何把真实世界映射到这种数学表示的各种观点留出了空间——例如什么是概率。

对概率理论历史和哲学的研究表明有多少个思想家就有多少种对概率含义的不同观点。不过，可以把这些观点分为几种不同类型的变体。这里我们把讨论范围限制在两种最重要的类型（根据它们对数据挖掘实践的影响）。喜爱哲学研究的读者可以参考 4.9 节，那里介绍了一些包含更广泛讨论的资料。

频率论观点（frequentist view）认为概率是一个客观概念。特别是把一个事件的概率定义为在绝对一致的条件下重复某一行为时这个事件发生次数的比例极限（limiting proportion）。一个简单的例子是当反复投硬币时正面出现次数的比例。这种解释限制了概率的应用：例如我们不能评估某个运动员在下一次奥运会上获得金牌的概率，因为这是个一次性事件，“比例极限”的思想没有意义。另一方面，我们当然可以评估顾客在超市购买某一种商品的概率，因为我们可以使用大量相似顾客作为比例极限的基础。在这个例子中进行了某种理想化（idealization）：不同顾客与一个顾客的重复行为事实上是不同的。和所有的科学建模一样，我们需要决定哪些方面对保证我们的模型足够准确是重要的。在预测顾客行为时，我们可能判定顾客间的差异是无关紧要的。

在上个世纪的绝大多数时间里，频率论观点主导了人们对概率的看法，而且因此成为大多数流行统计软件的基础。然而，在最近十年左右时间里，一种对立的观点已经受到了越来
95 越多的重视。这种主观概率（subjective probability）观点自从人们最初开始整理概率思想时就有了，然而直到最近它才开始引起人们的重视。导致这种方法复兴的因素是计算机的发展和用来操纵和处理主观概率的强大算法的出现。从主观概率观点派生出的数据分析理论和方法经常被称为贝叶斯统计（Bayesian statistics）。贝叶斯统计的一条核心原则是显式地刻画数据分析问题中所有形式的不确定性，包括从数据中估计的任何参数的不确定性；一系列模型结构中哪一个最好或最接近“真实”的不确定性；我们可能要做的任何预测的不确定性；等等。主观概率是对这些不同形式不确定性建模的一种非常灵活的框架。

根据主观概率观点，概率是一个人对一个特定事件能否发生的确信程度。因此概率不是外部世界的客观属性，而是个人的一种内心状态——因此可能由于个体的不同而不同。幸运的是，已经证明如果我们采用某种合理的行为原则，那么主观概率的公理集与频率论观点的公理集是相同的。因此两种观点的计算（calculus）是相同的，虽然潜在的解释（interpretation）

是完全不同的。

当然，这并不意味着使用这两种方法得到的结论一定是相同的。至少，主观概率可以应用在频率概率不适用的领域。还有，基于主观概率的统计归纳必然包含某种主观的成分——认为一个事件会发生的初始或先验（prior）信心。正像前面所指出的，这个因素可能因人而异。

尽管如此，频率论观点和主观概率观点在很多情况下会得到大体相同的答案，尤其是对于简单的假设和庞大的数据集。很多实践者并不把自己约束在一种或另一种观点上，相反，他们认为两种观点在各自的前提下都是有价值的，分别适用于不同的条件。由频率论观点推导出的数据分析方法往往计算更简单，因此当数据集的大小不适合使用复杂计算方法时，它具有明显的优势（至少到目前为止是这样）。然而，当应用得当时，贝叶斯（主观的）方法可以从数据中发现更加细微的信息。近年来，在应用统计中人们已经大大提高了对贝叶斯方法的重视程度，因此，我们可以预期将来会更多地在数据挖掘中应用贝叶斯思想。在本书的其余部分，我们将在适当的地方再次提到频率论观点和贝叶斯观点。正如在本章的后面将要 96
看到的，可以从某种意义上把这两种观点统一起来：可以把拟合模型和模式（到数据）的频率论方法实现为更通用的贝叶斯方法的一种特例。对于实践者来说这是非常有用的，因为这意味着可以使用一套通用的建模和计算方法。

4.3 随机变量和它们的关系

我们在第 2 章中介绍了变量的概念。这一章我们要介绍随机变量（random variable）的概念。随机变量是一种从对象属性到变量的映射，它可以取一系列值中的一个，一般来讲随机变量的取值过程对于观察者具有某些不可预测的因素。随机变量 X 的所有可能值被称为 X 的定义域。我们使用像 X 这样的大写字母来表示随机变量，并用像 x 这样的小写字母来表示随机变量的值。

随机变量的一个例子是投硬币的结果（定义域是集合{heads，tails}）。随机变量的不太明显的例子包括要抛出硬币的正面所需要的次数（定义域是正整数的集合）；以及纸飞机飞行的秒数（定义域是正实数的集合）。

本书的附录定义了一元（单一）随机变量的基本属性，既包括了当 X 的定义域有限时的概率质量函数 $p(X)$，又包括了当 X 的定义域为实数集或实数集的任意区间时的概率密度函数 $f(x)$。在附录中我们也回顾了 X 的基本属性——期望，对于实数值的 X，$E[X] = \int xf(x)dx$，因为 E 是线性运算所以有 $E[X+Y] = E[X] + E[Y]$。这些基本的属性是非常重要的，因为我们可以根据它们来推导出用于数据分析的一些一般原理，在本章的其余部分我们会经常提到分布、密度、期望等概念。

多元随机变量

因为数据挖掘经常处理多个变量，所以我们必须也介绍一下多元随机变量（multivariate random variable）的概念。一个多元随机变量 $\mathbf{X}$ 是一系列随机变量 X_1，…，X_p 的集合。我们使用 p 维向量 $\mathbf{x} = \{x_1, \cdots, x_p\}$ 来表示 $\mathbf{X}$ 的一套值。多元随机变量 $\mathbf{X}$ 的密度函数（density function）$f(\mathbf{X})$ 被称为 $\mathbf{X}$ 的联合密度函数（joint density function）。我们把它表示为 $f(\mathbf{X})=f(X_1 = x_1, \cdots, X_p = x_p)$，或简写为 $f(x_1, \cdots, x_p)$。类似地，我们可以得到变量在有限集合中取值的 97

联合概率分布。注意 $f(\mathbf{X})$是 p 个变量的标量函数。

$\mathbf{X}$ 中的单个变量（或者，更一般的情况是整个变量集合的任意子集）的密度函数被称为联合密度的边缘密度（marginal density）。从技术角度讲，它是根据联合密度通过对子集中未包含变量进行求和或积分推导出的。例如，对于一个三元随机变量 $\mathbf{X}=\{X_1, X_2, X_3\}$，$f(X_1)$ 的边缘密度为 $f(x_1)=\iint f(x_1, x_2, x_3)dx_2dx_3$。

某一变量（或者整个变量集合的一个子集）在给定其他变量取值（也就是“以这些值为条件”）情况下的密度被称为条件密度（conditional density）。这样我们就可以说，给定 X_2 取值为 6 后 X_1 变量的条件密度，并将其表示为 $f(x_1 \mid x_2=6)$。一般地，给定 X_2 的某个值后，X_1 的条件密度被表示为 $f(x_1 \mid x_2)$，并将其定义为：

$$f(x_1 \mid x_2)=\frac{f(x_1, x_2)}{f(x_2)} \tag{4.1}$$

对于离散值的随机变量，我们也有相应的定义（$p(a_1 \mid a_2)$ 等等）。我们也可以使用二者的混合——例如，以分类变量为条件的连续变量的概率密度函数 $f(x_1 \mid a_1)$，以及相反情况下的概率质量函数 $p(a_1 \mid x_1)$。

例 4.1 假定我们有一批来自超级市场的产品销售数据，数据矩阵中的每个观察（行）表示一个顾客购买的产品。每一列表示一种特定的产品，为每一列定义一个随机变量，这样每一种产品都有一个随机变量。如果顾客购买了某种产品，那么它对应的行和这种产品对应的列交叉处的观察值为 1，否则为 0。

用 A 表示一个特定列的二进制随机变量，对应于事件“购买产品 A”。A 取值为 1 的数据驱动概率估计就是购买产品 A 的顾客比例——也就是 n_A/n，其中 n 是顾客总数，n_A 是购买产品 A 的顾客数。例如，如果 $n=100\,000$，且 $n_A=10\,000$，那么一个随机选取的顾客购买产品 A 的概率估计是 0.1。

98 现在考虑第二种产品（数据矩阵中的第二列），并用和定义 A 相同的办法定义这一列对应的随机变量 B。设 n_B 为购买产品 B 的顾客数量；假定 $n_B=5\,000$，那么 $p(B=1)=0.05$。现在设 n_{AB} 为既购买了产品 A 又购买了产品 B 的顾客数。根据与前面相同的依据，对 $p(A=1, B=1)$ 的估计可以通过 n_{AB}/n 得到。现在我们可以把 $p(B=1 \mid A=1)$ 估计为 n_{AB}/n_A。例如，如果 $n_{AB}=10$，那么 $p(B=1 \mid A=1)=10/10\,000=0.001$。从这里我们看到，如果我们预先知道这个顾客购买了产品 A，那么它购买产品 B 的概率就由前面的 0.05 下降到这里的 0.001。对于数据库中的顾客，在购买了 A 的人中购买了 B 的人远比在这个数据库所有人中购买了 B 的人少（而且在购买了 A 的人中购买了 B 的人比没有购买 A 的人中购买了 B 的人少）。这提出了一个这样的问题，是否购买 A 通常会使购买 B 的可能性变小了呢？还是这个发现完全是仅对于我们数据库中碰巧得到的数据偶然成立的呢？这正是我们在本章的其余部分要针对的问题，特别是在 4.6 节的**假设检验**（hypothesis testing）中。

从上面的例子我们看到，多元变量集 $\mathbf{X}$ 的某些特定变量可能以某种方式密切地相互联系。实际上，数据挖掘的一个一般问题就是发现变量间的关系。购买商品 A 可能与购买商品 B 有关系吗？从一个测量仪器的记录中探测出的模式 A 是一个特定错误所导致的必然结

果吗？如果多个变量的取值相互间不存在任何关系，那么就说这些变量是独立的（independent）；否则它们就是依赖的（dependent）。更严格地讲，变量 X 和 Y 是独立的当且仅当对于 X 和 Y 的所有值有 $p(x, y)= p(x)p(y)$。一个等价的定义是 X 和 Y 是独立的当且仅当对于 X 和 Y 的所有值有 $p(x \mid y) = p(x)$和 $p(y \mid x)= p(y)$。（注意在这些定义中表达式中的所有 p 要么是概率质量函数，要么是概率密度函数——在后一种情况下变量独立的充要条件是 $f(x, y)= f(x)f(y)$）。第二种形式的定义表明当变量 X 和 Y 独立时，不论是否知道 Y 的值 X 的分布都是相同的。因此，Y 的取值不会影响 X 取值的概率，从这个意义上来说 Y 不带有任何关于 X 的信息。在描述超市销售的例子中，按照例子中给出的数据，变量 A 和变量 B 是依赖的。

我们可以把这些思想推广到多于两个变量的情况。例如，如果对于 X、Y 和 Z 的所有值 $p(x, y \mid z)= p(x \mid z)p(y \mid z)$都成立，那么我们就说给定 Z，X 对 Y 是条件独立的（conditionally independent）。下面举个例子来说明，假定一个人购买了**面包**（这使随机变量 Z 取值为 1）。然后又接着购买了**黄油**（随机变量 X 取值 1）和**干酪**（随机变量 Y 取值 1）。那么 X 和 Y 就有可能是条件独立的——一旦我们知道已经购买了面包，那么购买干酪不会受是否购买黄油的影响。 99

注意，条件独立未必意味着边缘（marginal）独立。也就是说，上面的条件独立关系并不意味着 $p(x, y)= p(x)p(y)$。例如在上面的例子中，通常我们会推测购买**黄油**和购买**干酪**是依赖的（既然它们都依赖于购买面包）。刚才的论断反过来也是成立的：X 和 Y 可能是（无条件）独立的，但对于给定的第三个变量 Z 它们是条件依赖的（conditionally dependent）。这些关于独立或依赖关系的细微之处对于数据挖掘者来说是非常重要的。尤其是，即使两个观测变量（例如**黄油**和**干酪**）对于给定的数据可能看起来是依赖的，但是它们的真实关系可能被第三个（潜在但没有观测的）变量（例如例子中的**面包**）掩盖了。

例 4.2 在研究和解释条件独立的结论时必须谨慎。例如考虑下面的假想例子。A 和 B 表示两种不同的治疗，下面表中显示的分数是康复患者的比例（也就是，左上角的 2/10 表示 10 个接受 A 治疗的老年患者中有 2 个康复了）。数据被分割成老年和青年两组，分组的依据是他们是否超过 30 岁。

	A	B
老年	2/10	30/90
青年	48/90	10/10

对于两个年龄层的每一组，B 治疗看起来都优于 A 治疗。然而，现在考虑总的结果——通过把上表中的两行汇总：

	A	B
汇总	50/100	40/100

总体来看，在这个汇总表中 A 治疗似乎比 B 治疗要好。乍一看，这个结果似乎是相当神奇的（事实上，这被称为**辛普森悖论**（Simpson's paradox）（Simpson, 1951））。 100

导致这两种似乎矛盾结果的原因是，第一张表的结果是以特定年龄层为条件的，而第二张表的结果是无条件的。当合并两个有条件的结论时，四组样本的大小差异导致基于较大样本的比例（“老年 B” 和 “青年 A”）支配了另两个比例。

条件独立的假定被广泛用于处理序列化（sequential）数据的场合。对于数据序列，只要给定序列中的当前值，那么序列中的下一个值经常是独立于序列中所有过去的值。在这种情况下，条件独立被称为一阶马尔可夫（first-order Markov）属性。

在后面的章节中我们会看到，独立和条件独立（可以把条件独立看作是独立的一般化）的思想是数据分析中很多关键概念的核心。独立和条件独立的假定使我们可以把多个变量的联合密度表示成更容易处理的较简单密度的连乘，也就是：

$$f(x_1,\cdots,x_n)=f(x_1)\prod_{j=2}^{n}f(x_j\mid x_{j-1}) \tag{4.2}$$

其中每个变量 x_j 是在给定 x_j 值（原著为 x_j，应为 x_{j-1}。——译者注）的情况下与变量 x_1，…，x_{j-2} 条件独立的（这是一阶马尔可夫模型中的一个例子）。这样的简化除了带来计算的方便外还有助于以更少参数建立更好理解的模型。但是，很多实际情况是不符合独立假定的（例如，假定文本中的字母序列符合一阶马尔可夫模型是不现实的）。尽管如此，应该知道我们的模型只是对真实世界的近似，恰当的独立假定所带来的好处经常胜过建立一个更加复杂却不太稳定的模型。在第 6 章中我们会更详细地讨论这样的建模问题。

依赖的一个特例是相关（correlation），或者说线性依赖，在第 2 章中我们介绍了这个概念。（注意统计依赖与相关不同：两个变量可能依赖但并不线性相关）。如果一个变量的较高值与另一个变量的较高值关联那么我们说它们是正相关；相反，如果一个变量的较高值与另一个变量的较低值关联那么我们说它们是负相关。千万注意不要把相关混淆为因果关系
101 （causation）。两个变量可能高度正相关，但它们间不存在任何因果关系。例如，指甲熏黄和肺癌可能相关，但它们仅是通过第三个变量有因果联系 ，也就是一个人是否吸烟。类似地，人的反应速度和他挣钱多少可能是负相关，但是这不意味着一个导致另一个。这种情况下一个更有说服力的解释是：第三个变量，年龄，与这两个变量都有因果联系。

例 4.3　美国医疗协会杂志（Journal of the American Medical Association）1987 年发表的一篇论文（257 卷，785 页）分析了美国 77 家医院所作的 18 986 例冠状动脉旁路移植手术的院内死亡率。回归分析（见第 11 章）表明手术次数越多的医院趋向于越低的院内死亡率（已经根据不同医院的不同病例类型对数据作了调整）。基于这个模式该文得出结论，如果关闭低手术量的手术室，那么这种类型手术导致的院内死亡率就会降低。

然而，要判断手术结果的质量和治疗病例数量间的关系，需要一种纵向的分析。在这一分析中不该过分重视规模的大小。如果手术量大的医院的手术量继续增长，那么可能导致它们的手术质量变差。手术结果和规模的相关可能不是由于较大的规模就导致出众的治疗效果，而是因为出众的治疗效果吸引了更多的患者，也有可能无论是患者的数量还是手术的结果都是与其他因素相联系的。

4.4　样本和统计推理

正如我们在第 2 章中所指出的，一些数据挖掘问题包括了感兴趣的整个总体；而另一些问题仅包括了来自这个总体的一个样本。对于后一种情况，可能本来就只有样本——或许仅

选择纳税者的一个样本来做详细的调查；或许仅是偶尔开展全面的人口普查，在大多数的年份仅是选择样本；或许数据集是由市场调查结果组成的。另一方面，即使可以得到完整的数据集，但数据挖掘操作是在一个样本上进行的。如果目标是建模（参见第 1 章），那么这样做是完全合理的，因为建模是要寻找数据的显著结构，而不是细小的特异和偏离（deviation）。这样的结构会保持在样本中，只要样本不要太小。然而，如果目标是模式识别，那么对大的数据集抽取小的样本就不太适合了，因为这时的目标是探测数据主体的细小偏离，因此如果样本太小，那么偏离就可能被排除在外。此外，如果目标是探测反常行为的记录，那么必须基于整个样本进行分析。 102

正是当使用样本时，才发挥出了统计推理的作用。通过统计推理（statistical inference），我们可以论断总体的结构，估计这些结构的大小，并指出对这些结论的置信度（degree of confidence），而这一切都依赖于样本（参见图 4-1，图中简单画出了概率和统计的作用）。例如，我们可以说总体值的最佳估计是 6.3，也可以说，我们对真实的总体值位于 5.9 到 6.7 之间有 95%的把握。（定义和解释这样的区间是很繁琐的，因为这依赖于我们采用的哲学基础——例如是频率论的还是贝叶斯的。我们将在本章的后面更多地介绍这样的区间。）注意这里我们对总体值使用了估计（estimate）一词。如果我们是基于整个总体进行分析，那么我们将使用计算（calculate）这个词：因为如果已经知道了所有的组成要素，那么我们就可以实际计算出总体的值，也就不存在估计的概念了。

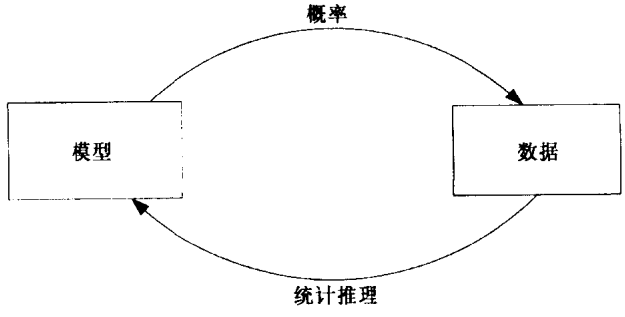


图4-1 概率和统计在数据分析中的作用图示。概率确定了观测数据是如何从模型中产生的。统计推理允许我们从观测数据中推理出模型

为了对总体结构做出推理，我们必须在脑海中有一个模型或模式结构：如果我们从来没有想到某种结构的存在，那么我们也就无法评估数据中潜在某种结构的证据。例如，我们可能假设某一变量 Z 的值依赖于其他两个变量 X 和 Y 的值。我们的模型是 Z 与 X 和 Y 有关。然后我们可以在数据中估计这个关系的支持度。（当然，我们可能得出这样的结论：这两个关系的支持度为 0——也就是根本没有关系。） 103

统计推理是基于这样的前提的：样本是从总体中以随机方式抽取的——这使得总体中的每一个成员都有一定的概率出现在样本中。模型将确定总体的分布函数——随机变量的特定值在样本中出现的概率。例如，如果模型指出数据是从一个正态分布产生的，这个正态分布的均值为 0，标准差为 1。那么这也同时告诉我们观察到 20 这样大的数据的概率是很小的。而且，如果假定模型是正确的，那么我们可以给出观察到大于 20 的值的精确概率。给定了模型，我们一般便可以计算一个观察的结果落入任意区间的概率。对于符合范畴型分布的样本，我们可以估计新的值与已经出现的每一个值相等的概率。一般来说，如

果我们得到了数据的模型 M，那么我们就可以指出一个随机抽样过程得到数据 D 的概率，$D = \{\mathbf{x}(1), \cdots, \mathbf{x}(n)\}$，这里 $\mathbf{x}(i)$是第 i 个 p 维测量向量（在我们的 $n \times p$ 数据矩阵中的第 i 行）。这个概率被表示为 $p(D \mid M)$。很多时候我们并不明确地指出对模型 M 的依赖，而简单地写为 $p(D)$，依赖上下文来做出说明。（正如附录中所指出的，如果一个变量服从连续累积分布函数，那么观察到它取任何特定值的概率是 0——特定值意味着区间的长度为 0，因此概率密度函数跨越这一区间的区域面积是 0。然而，所有的实际数据实际上都是指有限的（即使很小）区间（例如，如果说某个人的身高为 5 英尺 11 英寸，那么别人知道这个人的高度是介于 5 英尺 10.5 英寸和 5 英尺 11.5 英寸之间）。因此在实践中，谈论观察到任何特定值的概率是没有意义的。）

设 $p(\mathbf{x}(i))$为个体 i 取测量向量 $\mathbf{x}(i)$的概率（这里的 p 可能是概率质量函数或概率密度函数，视 $\mathbf{x}$ 的特性而定）。如果我们进一步假定总体中的每一个成员被选择进入推理用样本的概率不会影响其他成员被选择的概率（也就是每个观测是独立的，或者说数据是随机抽取的），那么观察到所有样本值的总概率就是个体概率的乘积：

104

$$p(D \mid \theta, M) = \prod_{i=1}^{n} p(\mathbf{x}(i) \mid \theta, M) \tag{4.3}$$

其中 M 为模型，θ 是模型的参数（假定在这一点是固定的）。（当把这个公式看作模型参数 θ 的函数时，这个公式被称为似然函数（likelihood function）。我们将在后面对此进行详细的讨论。）已经开发出了一些方法来处理观测到一个值会改变观测到另一个值的机会的情况，但是各个观测相互独立是迄今为止最普遍使用的假定，尽管这仅是近似的正确。

根据这个概率，我们可以判断假定模型的真实性。如果我们的计算表明假定模型产生观察数据的可能性非常小，那么我们会觉得拒绝这个模型是合理的；这是假设检验的基本原则（4.6 节）。在假设检验中，如果符合模型的观察数据的概率低于某个预先定义的值（经常是 0.01 或 0.05——检验的显著性水平（significance level）），那么我们就会决定拒绝这个假定模型。

在估计模型参数的总体值时使用了一个类似的原则。假定我们的模型指出数据服从单位方差的正态分布，但均值μ未知。我们可以提出很多用作均值的不同值，对于每一个，计算如果总体的均值为该值时观察数据的发生概率。我们可以对每一个值进行假设检验，拒绝导致观察数据发生概率很低的那些值。或者我们可以缩短这个过程，就使用可以使观察数据的发生概率最高的均值估计。这个值被称为均值的最大似然估计值，这一我们刚刚描述的过程被称为最大似然估计（见 4.5 节）。当把一个特定模型产生观察数据的概率表示为模型参数的函数时，这个函数被称为似然函数。也可以用这个函数来定义一个参数可能值区间[⊖]；例如，我们可以说，假定我们的模型是正确的，那么按这种方式根据数据样本产生的参数可能值区间中有 90%将包含参数的正确值[⊜]。

⊖ 译注：即选择一个阈值 T，导致似然大于 T 的任何参数值位于这个区间中；导致似然小于 T 的任何参数值不在这个区间中。

⊜ 译注：给定一个数据集，我们便可以定义一个可能值区间，因此这个区间会随数据集的变化而变化。对于许多数据集来说，会有 90%的区间包含真实参数值（90%是举例来说的，当然也可以选择其他百分比）。

4.5 估计

在第 3 章中我们描述了几种技术来概括一个给定的数据集。当我们致力于统计推理时，我们希望得出更通用的结论，关于被抽样总体的结论。这些结论是关于概率分布或者概率密度函数的（或者等价地说是关于累积（cumulative） 分布函数的），数据被假定为是从这些 105
分布中产生的。

4.5.1 估计量的理想属性

在接下来的小节中，我们将描述两种最重要的模型参数估计方法：极大似然估计和贝叶斯估计。注意不同方法间的差异是很重要的，因为这样我们才能选出一种适合我们的问题的方法。这里我们先简要地描述估计量（estimator）的一些重要属性。设 $\hat{\theta}$ 是参数 θ 的估计量。因为 $\hat{\theta}$ 是从数据推导出的一个数字，那么如果我们抽取不同的数据样本，我们就会得到一个不同的 $\hat{\theta}$ 值。因此 $\hat{\theta}$ 是一个随机变量。所以，它具有一种分布，随着抽取样本的不同而取不同的值。我们可以得到这个分布的一些描述性概括。例如，这个分布将具有一个均值或期望值——$E[\hat{\theta}]$。这里期望函数 E 是由假定数据从中采样的真实（未知）分布决定的——也就是，对于所有可能发生的容量为 n 的数据集按它们的发生概率加权。

$\hat{\theta}$ 的*偏差*（bias）（在第 2 章中我们非正式的介绍过这个概念）是这样定义的：

$$Bias(\hat{\theta}^{\ominus}) = E[\hat{\theta}] - \theta \tag{4.4}$$

也就是估计量的期望值 $E[\hat{\theta}]$和参数 θ 的真实值的差异。满足 $E[\hat{\theta}] = \theta$ 的估计量的偏差为 0，被称为是*无偏的*（unbiased）。平均来看，这样的估计量与真实参数值间没有*系统的*（systematic）偏离（departure），尽管对于任一特定单一数据集 D，$\hat{\theta}$ 可能远离 θ。注意样本分布和 θ的真实值实际上都是未知的，我们通常不能计算对于给定数据集的实际偏差。尽管如此，偏差（以及下面的方差）的一般概念在估计中是绝对重要的。

就像估计量的偏差可以衡量它的质量一样，估计量的方差也可以做到这一点：

$$Var(\hat{\theta}) = E[\hat{\theta} - E[\hat{\theta}]]^2 \tag{4.5}$$

方差衡量了估计误差中的随机的和由数据导致的那一部分；它反映了估计量对数据集中的特异性的敏感程度。注意方差不依赖于 θ 的真实值——它仅衡量我们的估计对于不同的观测数据集变化程度有多大。因此，尽管真实的采样分布是未知的，原则上我们还是可以得到一个估计量方差的数据驱动估计（对于给定的 n 值），方法是反复对原始的数据做二次抽样并计算从这 106
些模拟样本估计出的 $\hat{\theta}$ 的方差。我们可以在具有相同偏差的估计量中选取最小方差的一个估计量。具有最小方差的无偏估计量被顺理成章地称为*最佳无偏估计量*（best unbiased estimators）。

举一个极端的例子，设想我们完全忽视数据 D 并且武断地说对于任意的数据集都有 $\hat{\theta} = 1$，那么 $var(\hat{\theta})$便为 0，因为 $\hat{\theta}$ 的估计根本不随着 D 的改变而改变。然而在实践中这是一个根本无效的估计量，因为除非我们非常幸运地猜中，否则我们对 θ 的估计几乎一定是错误的，

⊖ 译注：原书此处为 $Bias(\theta) = E[\hat{\theta}] - \theta$，系排版错误。

也就是说存在一个非0（而且可能非常大）的偏差。

$\hat{\theta}$ 的*均方误差*（mean squared error）是 $E[(\hat{\theta} - \theta)^2]$，即估计量的值和参数的真实值间的差异平方的均值。均方误差可以分解为 $\hat{\theta}$ 的偏差的平方以及它的方差的和：

$$\begin{aligned} E\left[(\hat{\theta}-\theta)^2\right] &= E\left[(\hat{\theta}-E[\hat{\theta}]+E[\hat{\theta}]-\theta)^2\right] \\ &= \left(E[\hat{\theta}]-\theta\right)^2 + E\left[(\hat{\theta}-E[\hat{\theta}])^2\right] \\ &= \left(Bias(\hat{\theta})\right)^2 + Var(\hat{\theta}) \end{aligned} \tag{4.6}$$

在从第一行到第二行的转化中，我们利用了平方表达式中交叉项相互抵消；当θ为常数时 $E[\theta] = \theta$；等等。均方误差是一个非常有价值的标准，因为它联合了估计量和真实值间的系统（偏差）差异和随机（方差）差异（当然，它也主要是为理论研究服务的，因为计算它需要知道θ 这个在实践中不知道的量）。不幸的是，偏差和方差经常是按不同方向变化的，修改一个估计量以减小它的偏差会增加它的方差，反之亦然。所以关键是得到一个最佳的折衷。平衡偏差和方差是数据挖掘的一个核心问题，我们将在第6章返回到这个话题作一般性的讨论，在之后的章节将结合具体问题作更有针对性的讨论。

在估计中使用均方误差还该注意一些更细微的问题。例如，误差平方同等对待偏离θ一
107 样远的估计值，无论它在θ之上还是之下。这对衡量位置是合适的，但对衡量离差（dispersion）（根据定义，离差的下边界可以小于0）或估计概率或概率密度就可能不适合了。

假定我们有一个估计量的序列 $\hat{\theta}_{n_1}$，…，$\hat{\theta}_{n_m}$，它们是基于递增的样本大小 n_1，…，n_m。如果随着样本容量的增大 $\hat{\theta}$ 与真实值θ 的差异大于任一给定值的概率趋向于0，那么就说这个序列是*一致的*（consistent）。这显然是一个有吸引力的属性（特别对于数据挖掘场合，样本非常庞大），因为样本越大估计量可能越靠近真实值（假定数据来自于一个特定的分布——根据第1章和第2章的讨论，对于非常庞大的数据库这个假定可能是不合理的）。

4.5.2 最大似然估计

最大似然估计是应用最广的参数估计方法。考虑一个包含 n 个观测的数据集 $D = \{\mathbf{x}, \cdots, \mathbf{x}(n)\}$，它是从同一个分布 $f(\mathbf{x} \mid \theta)$独立采样得到的（用统计学家的话来讲即*独立同分布*（independently and identically distributed），或者叫 iid）。*似然函数* $L(\theta \mid \mathbf{x}(1), \cdots, \mathbf{x}(n))$是对于给定的$\theta$ 值这些已经发生数据的概率，也就是 $p(D \mid \theta)$，它是关于θ 的函数。注意尽管我们在这里隐含的假定了一个特定的模型 M，但是就像定义 $f(\mathbf{x} \mid \theta)$一样，为了方便，我们没有明确地写出 M——后面当我们考虑多个模型时，我们将需要明确的区分我们谈论的是哪一个模型。

既然我们已经假定了观察是独立的，那么我们就可以得到：

$$\begin{aligned} L(\theta \mid D) &= L(\theta \mid \mathbf{x}(1), \cdots, \mathbf{x}(n)) \\ &= p(\mathbf{x}(1), \cdots, \mathbf{x}(n) \mid \theta) \\ &= \prod_{i=1}^{n} f(\mathbf{x}(i) \mid \theta) \end{aligned} \tag{4.7}$$

这是θ 的一个标量函数（其中θ 本身可能是参数向量，而不是一个单一的参数）。一个*数据集的似然*（likelihood of a data set）$L(\theta \mid D)$，即实际观测的数据对于一个特定模型的概率，是数据分析的一个基本概念。为一个给定问题定义似然等同于确定产生数据的概率模型。

已经证明，一旦我们能够找到这样的似然，那么我们便打开了统计推理的大门，可以应用其 108
中很多通用的强大方法。注意既然似然是定义为θ 的函数，那么我们就可以删除或忽略$p(D\mid\theta)$中所有不含θ 的项，也就是说，似然仅定义在任意缩放的常量范围内，所以我们所关心的是θ 的函数的形状，而不是函数的实际值。也该注意上面的 idd 假定对于似然的定义是不必要的：例如，如果 n 个观察符合马尔可夫依赖关系（其中每一个 $\mathbf{x}(i)$依赖于 $\mathbf{x}(i-1)$），那么我们可以把似然定义为像$f(\mathbf{x}(i)\mid\mathbf{x}(i-1),\theta)$这样的项的乘积。

使已经发生数据的概率最大的θ 值就是最大似然估计量（maximum likelihood estimator）（或者叫 MLE）。我们用$\hat{\theta}_{ML}$表示θ 的最大似然估计量。

例 4.4 超市中的顾客要么购买牛奶，要么不购买牛奶。假定我们要对购买牛 109
奶顾客的比例做出估计，根据是从数据库中随机抽取的 1 000 个观测值的样本 $x(1)$，…，$x(1\,000)$。这里如果第 i 个顾客确实购买了牛奶，那么 $x(i)=1$，否则为 0。假设这些独立观察遵循的是二项分布（参见附录），但参数 $0\leqslant\theta\leqslant 1$ 未知；也就是说，θ 是一个随机顾客购买牛奶的概率。对于给定的模型，在通常的条件独立假定下，似然函数可以写为：

$$L(\theta\mid x(1),\cdots,x(1000))=\prod_i\theta^{x(i)}(1-\theta)^{1-x(i)}=\theta^r(1-\theta)^{1000-r}$$

其中 r 是 1000 名顾客中购买牛奶的顾客数。对上式取对数得到：

$$l(\theta)=\log L(\theta)=r\log\theta+(1000-r)\log(1-\theta)$$

对上式求导数并令其为 0，得到

$$\frac{r}{\theta}-\left(\frac{1000-r}{1-\theta}\right)=0$$

从上式我们可以解出$\hat{\theta}_{ML}=r/1\,000$。因此，购买牛奶者的比例事实上也就是在这个二项分布中θ 的最大似然估计。

在图 4-2 中我们画出了在这个二项分布模型下的三组假想数据关于θ 的似然函数曲线。三个数据集分别对应 $n=10$、$n=100$、$n=1000$ 个顾客中有 7 个、70 个和 700 个牛奶购买者。每一种情况中似然函数的峰值都发生在$\theta=0.7$ 时，但随着 n 的增大（也就是当我们具有庞大的顾客数据库时）θ 真实值的不确定范围越来越小（反应在似然函数的伸展范围）。似然函数的绝对值是不重要的，重要的是它的形状。

例 4.5 假定 n 个数据点的样本 $x(1)$，…，$x(n)$是从一个正态分布独立抽取的，正态分布具有单位方差，但均值θ未知。当不确定性的来源是测量误差时就可能发生这样的情况：我们可能知道结果具有确定的方差（这里为 1），但不知道反复测量的对象的均值。那么θ 的似然函数为：

$$L(\theta\mid x(1),\cdots,x(n))=\prod_{i=1}^{n}(2\pi)^{-1/2}\exp\left(-\frac{1}{2}(x(i)-\theta)^2\right)$$

110

$$= (2\pi)^{-n/2} \exp\left(-\frac{1}{2} \sum_{i=1}^{n} (x(i) - \theta)^2 \right)$$

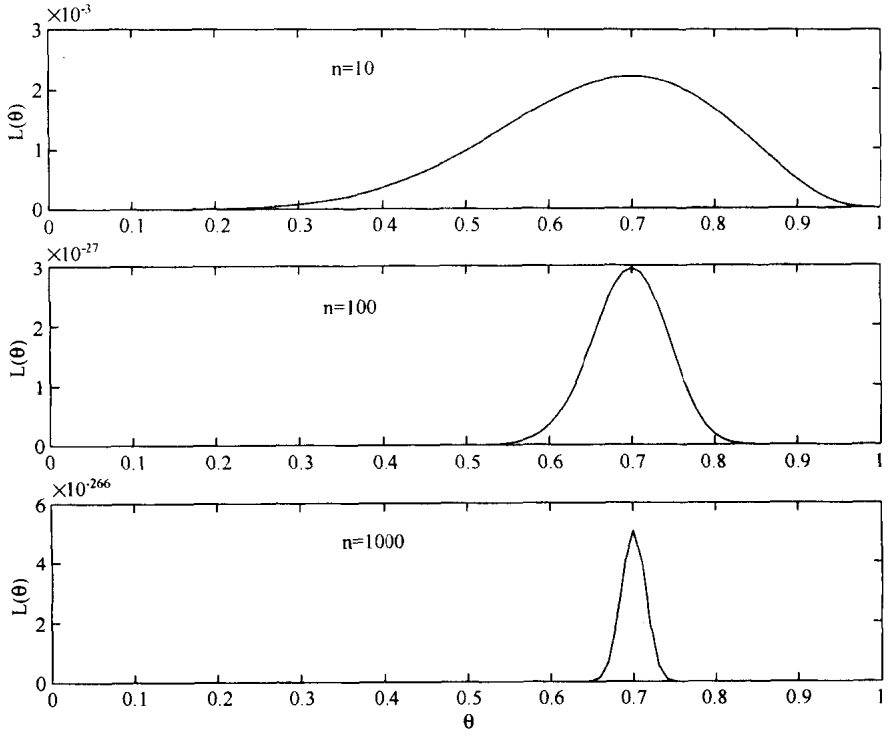


图4-2　二项模型下的三个假想数据集的似然函数。三幅图分别对应$r=7$，$n=10$（上）；$r=70$，$n=100$（中）；$r=700$，$n=1000$（下）

它的对数似然（log-likelihood）被定义为：

$$l(\theta \mid x(1), \cdots, x(n)) = -\frac{n}{2} \log 2\pi - \frac{1}{2} \sum_{i=1}^{n} (x(i) - \theta)^2 \tag{4.8}$$

为了求出 MLE 我们令导数 $\frac{d}{d\theta} l(\theta \mid x(1), \cdots, x(n))$ 为 0，于是得到：

$$\sum_{i=1}^{n} (x(i) - \theta) = 0$$

111　所以，θ的最大似然估计量为 $\hat{\theta}_{ML} = \sum_i x(i)/n$，即样本的均值。

图 4-3 画出了关于θ 的似然函数和对数似然函数 $l(\theta) = \log L(\theta)$，所用的样本是来自正态分布的 20 个数据点，正态分布的真实均值为 0，而且已知标准偏差为 1。图 4-4 画出了同样类型的图形，但是是对于 200 个数据点的。注意似然函数的峰值在真实均值 0 附近的情况。也请注意随着数据的增多似然函数是如何变窄的，这反映了数据对不靠近 0 的θ 值的支持的下降。

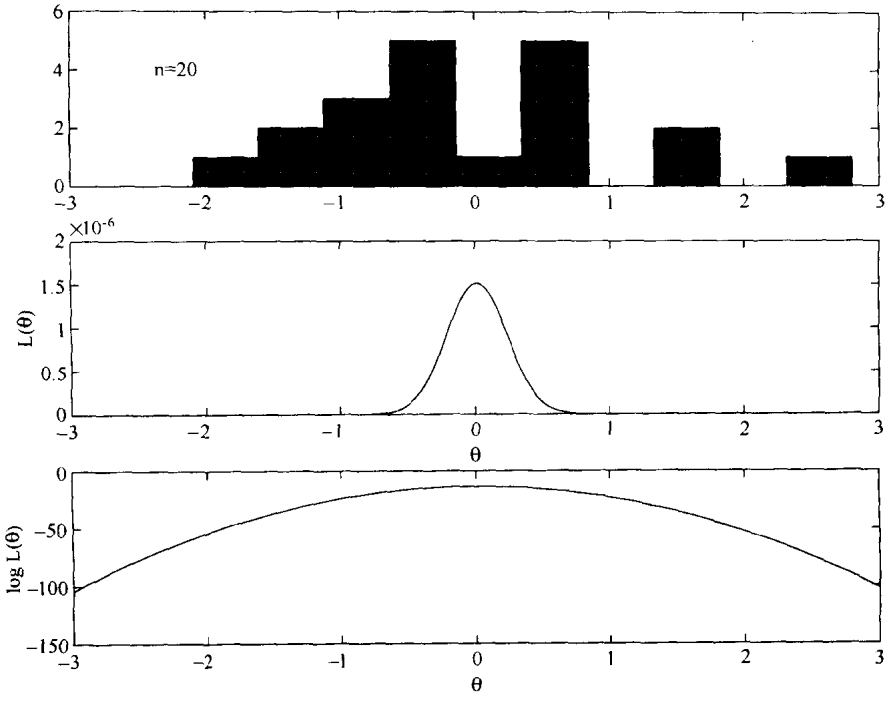


图4-3 数据点来自真实均值为0而且已知标准差为1的正态分布：（a）从真实模型产生的20个数据点的直方图（上）；（b）关于θ 的似然函数（中）；（c）关于θ 的对数似然函数（下）

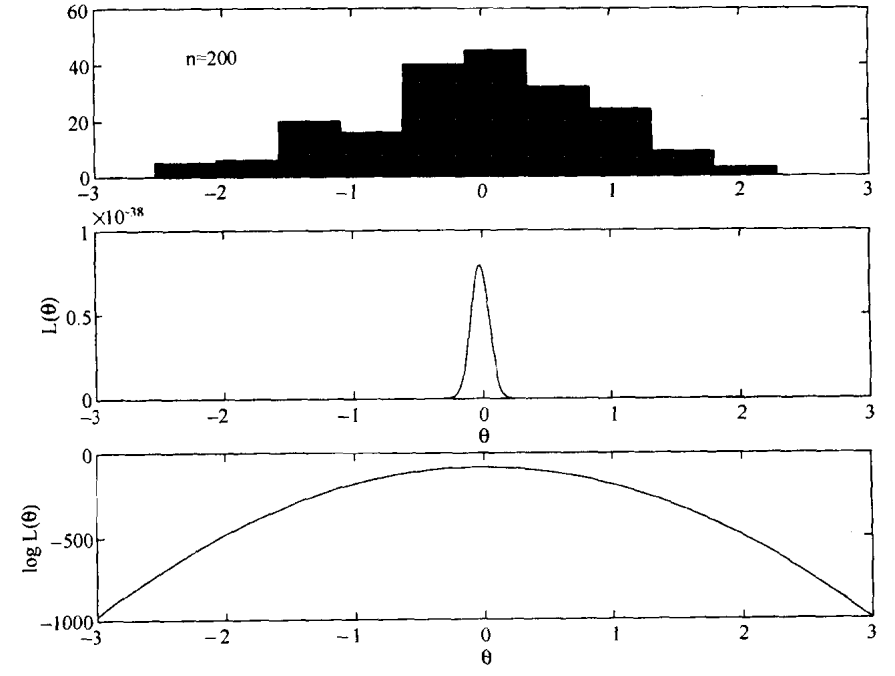


图4-4 与图4-3相同模型的200个数据点的似然函数。（a）从真实模型产生的200个数据点的直方图（上）；（b）关于θ 的似然函数（中）；（c）关于θ 的对数似然函数（下）

例 4.6 在统计估计中，**充分统计量**（sufficient statistic）的概念是很有价值的。

简单地讲，如果似然函数 $L(\theta)$仅通过一个量 $s(D)$依赖数据，那么我们就把这个量 $s(D)$
112 定义为θ 的充分统计量。于是，在上面的二项模型中，“成功”的总数 r（购买牛奶的人数）就是二项参数θ 的充分统计量。充分的含义是似然仅是 r 的函数（假定 n 已经知道）。从我们的二项模型的角度来看，一旦我们已经知道了总的 r，那么知道哪一个特定的顾客购买了牛奶（数据矩阵中哪一个特定行的“牛奶”列为 1）是无关紧要的。类似地，对于上面的需要估计正态分布均值的例子，观察的总和 $\sum_{i=1}^{n} x(i)$ 是均值似然的充分统计量（记住似然仅定义为θ 的函数，因此所有不包含θ 的其他项都可以删除）。

对于大规模的数据集，充分统计量的概念在实践中是非常有价值的——不必工作在整个数据集上，只需简单地计算和存储充分统计量，只要我们知道了对于似然估计这些量是充分的。例如，如果我们在收集大批量的日常数据（例如网络日志），那么原则上我们只要每天晚上更新充分统计量，然后把原始数据扔掉。然而不幸的是，对于很多更复杂的模型经常是不存在充分统计量的，特别是那些我们想在数据挖掘应用中使用的，例如在本书的后面要详细讨论的树，混合模型等等。尽管如此，对于比较简单的模型，充分统计量是一个非常有价值的概念。

最大似然估计既具直观性，又有数学严密性，所以它是一种有吸引力的参数估计方法。例如，根据前面的定义它是一致的估计量。而且，如果$\hat{\theta}_{ML}$是参数θ 的 MLE，那么 $g(\hat{\theta}_{ML})$是函数 $g(\theta)$的 MLE，但是当 g 不是一对一的函数时应该引起注意。另一方面，任何事物都不是十全十美的——最大似然估计量经常是有偏差的（依赖于参数和潜在的模型），尽管对于庞大的数据集这个偏差可能相当小，经常按 $O(1/n)$缩小。

对于简单的问题（这里“简单”是指问题的数学结构，而不是数据点的数量，数据点可以非常多），可以使用求导运算求解 MLE。在实践中，通常是用最大化对数似然 $l(\theta)$的方法（就像上面的二项分布和正态分布的例子），因为这可以用求和取代定义中难以处理的乘积形式；这样的处理与直接最大化 $L(\theta)$的结果是一样的，因为对数是单调的函数。当然，我们经常对一个以上参数的模型感兴趣（像神经网络这样的模型（第 11 章）具有成百上千的
113 参数）。似然的一元定义可以直接推广到多元的情况，但这时似然就是 d 个参数的多元函数（也就是定义在 d 维参数空间中的一个标量值函数）。因为 d 可能很大，所以如果不存在闭合形式的解（closed-form solution），那么要发现这个 d 维函数的最大值可能是有很大难度的。我们将在第 8 章详细讨论这种关于优化的话题，在那里我们介绍了迭代搜索方法。多个最大值也会使问题复杂化（正因为此，很多情况下必须使用随机的最优化方法），最优值出现在参数空间边界的情况也会导致困难。

例 4.7　简单线性回归在数据挖掘中应用非常广泛。我们在第 1 章中曾经简单提到过，而且在第 11 章中会再次详细地讨论。在最简单的线性回归形式中，它联系两个变量：X，**预报**（predictor）变量或者叫**解释**（explanatory）变量；Y，**响应**（response）变量。它们的关系被假定为具有这样的形式：$Y = a + bX + e$，其中 a 和 b 为参数，e 是一个随机变量，假定 e 服从均值为 0 方差为σ^2的正态分布，而且我们可以将其写为 $e = Y - (a + bX)$。数据是由一系列有序偶组成的，即 $D = \{(x(1),$

$y(1))$, …, $(x(n), y(n))\}$, 对于给定的解释数据反映数据的概率密度函数为 $f(y(1), \cdots, y(n) | x(1), \cdots, x(n), a, b)$。我们的兴趣不是为 x 的分布建模，而是要对 $f(y | x)$建模。

s 于是，这个模型的似然（或者更确切地讲是条件似然）函数可被写为:

$$L(a,b|D)=\prod_{i=1}^{n}\frac{1}{\sigma\sqrt{2\pi}}\exp(-0.5(y(i)-(a+bx(i))/\sigma)^2)$$

$$=\frac{1}{(2\pi\sigma^2)^{n/2}}\exp\left(-0.5/\sigma^2\sum_{i=1}^{n}(y(i)-(a+bx(i)))^2\right)$$

为了求解 a 和 b 的最大似然估计量，我们可以取对数并抛弃不包含 a 和 b 中任一个的项。这得到:

$$\sum_{i}^{n}(y(i)-(a+bx(i)))^2$$

于是，我们可以通过最小化预测值 $a + bx(i)$和观察值 $y(i)$的差异平方和来估计出 a 和 b。这样最小化平方和的过程在数据挖掘中是很常见的，而且被称为**最小平方方法**（least squares method）。平方和标准具有重要的历史地位，它的根源可以追溯到高斯和更早以前。起初选取平方和看起来好像是任意的（为什么不选择其他呢？比如绝对值的和），但从上式可以看到选择最小平方是选择正态分布后为了表示模型的误差项而导致的自然结果。 114

直到现在我们一直讨论的是点估计（point estimates），为问题中的参数估计出单一的数字。从某种意义上来说，点估计是“最佳的”估计，但是点估计不能传达与之关联的任何不确定性——或许存在大量的几乎等价的好的估计，或许这个估计只是目前最好的。区间估计提供了这样的信息，它不再使用单一的数字，而是给出一个具有确定置信度的区间，这个区间含有未知的参数值。这样的区间被称为置信区间（confidence interval），这个区间的上下边界被称为置信边界（confidence limits）。置信区间的解释是相当微妙的。这里，既然我们假定θ 是未知的，但我们已确定它的估计值，那么说θ 具有一定的概率位于一个给定的区间是没有意义的，因为θ 要么在这个区间中，要么不在。然而，说通过给定过程计算的区间具有一定的概率包含θ 是有意义的，因为毕竟区间是从样本计算来的，因此是一个随机变量。

例 4.8 为了使解释更加简单，下面的例子是特意编制的。假定数据是由来自正态分布的 100 个独立观测组成的，正态分布的均值μ未知，方差σ^2 已知。现在我们要求出μ的置信度为 95% 的置信区间。也就是说，给定数据 $x(1)$, …, $x(n)$，我们要求出一个上限 $u(x)$和一个下限 $l(x)$，使 $P(\mu \in [l(x), u(x)])= 0.95$。

这种情况下样本均值$\bar{x}$ 的分布服从均值为μ，方差为$\sigma^2/100$ 的正态分布，所以标准差为$\sigma/10$。从正态分布的属性（参见附录）可知，95% 的概率位于距离均值 1.96 个标准差的范围内。所以，

$$P(\mu-1.96\sigma/10 \leqslant \bar{x} \leqslant \mu+1.96\sigma/10) = 0.95$$

上式可以写为:

$$P(\bar{x}-1.96\sigma/10 \leqslant \mu \leqslant \bar{x}+1.96\sigma/10) = 0.95$$

于是，$l(x)=\bar{x}-1.96\sigma/10$ 和 $u(x)=\bar{x}+1.96\sigma/10$ 定义了一个 95% 的置信区间。

大多时候，置信区间是基于这样的假定：样本的统计量大体符合正态分布。这一点是经常会被满足的：中心极限定理（central limit theorem）告诉我们很多统计量可以用一个正态
115 分布来很好的近似，特别是当样本容量很大时。使用这种近似，我们就得到了一个区间，对于给定的未知参数θ 的值，统计量位于这个区间的概率是已知的，然后再反过来求未知参数的区间。为了应用这种方法，我们需要估计估计量$\hat{\theta}$ 的标准差。导出这个估计的一种方法是 bootstrap 方法。

例 4.9 在过去的 20 年中人们已经开发出了很多种 bootstrap 方法，使这种方法得到逐步完善。这种方法的基本思路如下：数据最初来自分布 $F(X)$，我们需要对这个分布做出某种推论。然而，我们仅有数据的一个样本，我们用 $\hat{F}(X)$表示这个样本。现在我们要做的是从 $\hat{F}(X)$中抽取一个子样本 $\check{F}(X)$，抽样时就把 $\hat{F}(X)$当作是真实的分布。我们可以重复这个过程很多次，为每一个这样的子样本计算出统计量。这个过程为我们提供了根据从 $\hat{F}(X)$中抽取的样本计算出的统计量的采样属性信息，我们希望这些信息与从 $F(X)$中抽取的样本计算出的统计量的采样属性信息是相似的。

为了说明这种方法，考虑一种估计预测分类规则性能的早期方法。正如我们前面讨论的，要估计分类规则的性能，简单地通过重新分类用来设计这些法则的数据是不明智的——很可能导致偏向乐观的估计。假定 e_A 是通过简单的重组过程得到的误分类率估计，在这个过程中使用的数据是与我们用来估计分类模型参数相同的数据。我们真正需要的估计量是 e_C，“真实”的误分类率，我们希望它能适用于未来的对象。这两个估计的差为（e_C-e_A）。如果我们能够估计这个差异，那么我们就可以调整 e_A以得到更好的估计。事实上，我们可以通过下面的方法来估计这个差异。假定我们把 $\hat{F}(X)$当作这时的分布并从中抽出一个子样本——$\check{F}(X)$。现在，就把 $\hat{F}(X)$当作真实的分布，那么我们根据子样本 $\check{F}(X)$中的数据建立一个规则，并把它同时应用到 $\hat{F}(X)$和 $\check{F}(X)$。这两种情况规则性能的差异就为我们提供了差异（e_C-e_A）的信息。为了降低由于采样过程的随机性所带来的影响，我们重复这种二次采样的过程很多次并取平均值。最终的结果是对（e_C -e_A）差异的估计，可把这个差异加到 e_A值中，以得到对真实误分类率 e_C的估计。

4.5.3 贝叶斯估计

116 在本节之前所描述的频率论推理方法中，总体的参数是固定但未知的，数据组成了一个来自总体的随机样本（因为样本是以随机方式抽取的）。因此本质的变化性存在于数据 $D=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$中。与此相反，贝叶斯统计把数据当作是已知的——毕竟，数据是已经被观察到并被记录下的，并且把参数θ 看作随机变量。因此，尽管频率论的方法把参数θ 看作是固定但未知的量，但是贝叶斯方法把θ 当作有很多可能值的随机变量，它服从一定的分布，并认为已观察到的数据可以揭示这个分布的信息。$P(\theta)$反映了我们对参数θ 真实（未知的）取值的确信程度。如果对于θ 的某个值 $p(\theta)$的曲线非常尖锐，那么说明我们非常确信我们的结论（当然我们可能是完全错误的!）。如果 $p(\theta)$的曲线是非常宽广平坦（这是更典型的

情况），那么这表示我们对θ的位置不太确定。

虽然贝叶斯这一术语在统计中有相当精确的含义，但它有时也被很随便的用在计算机科学和模式识别等文献中，用来指数据分析所使用的各种形式的概率模型。在本书中，我们采用更标准的广为流传的统计学定义，具体将在下面描述。

在分析数据之前，θ取不同值的概率分布被称为先验（prior）分布$p(\theta)$。这个分布在数据分析时会被修改，以纳入实验数据中的信息，修改后得到后验（posterior）分布，$p(\theta \mid D)$。从先验分布修改为后验分布是通过贝叶斯定理来进行的，这个定理是以托马斯·贝叶斯的名字命名（Thomas Bayes）的：

$$p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{p(D)} = \frac{p(D \mid \theta)p(\theta)}{\int_{\Psi} p(D \mid \Psi)p(\Psi)d\Psi} \tag{4.9}$$

注意这个更新过程产生一个分布，而不是θ的一个单一值。然而，可以用这个分布得到一个单一的估计值。例如，我们可以取后验分布的均值，或它的最频值（mode）（后一种技术被称为最大化后验（maximum a posteriori）法，或简称为MAP）。如果我们以特定的方式选取先验分布$p(\theta)$（例如，在某个范围里$p(\theta)$是均匀的），那么MAP和θ的最大似然估计可能很好吻合（这是因为这个先验分布是“平的”，所以不会优先任何一个θ值）。从这个意义上讲，可以把最大似然估计看作是MAP过程的一个特例，前者是贝叶斯估计的一种特定形式（“点估计”）。

对于一个给定的数据集D和一个特定的模型，公式（4.9）的分母是一个常数，所以我们还可以把表达式写为另一种形式： 117

$$p(\theta \mid D) \propto p(D \mid \theta)p(\theta) \tag{4.10}$$

现在我们看到对于确定的D（也就是这个分布是以已经观察到的数据D为条件的），θ的后验分布与先验分布$p(\theta)$和似然$p(D \mid \theta)$的乘积成正比。如果在收集数据前我们对参数的可能值仅有非常小的把握，那么我们希望选择一个概率散布很广（例如，具有很大方差的正态分布）的先验分布。在任何情况下，观察到的数据集合越大，似然对后验分布的支配性越大，同时先验分布形状的重要性也就越小。

例 4.10 重新考虑讨论购买牛奶客户比例的例4.4，在这个例子中我们考虑了一个二进制的单一变量X，并希望估计$\theta = p(X = 1)$。对于变化范围介于0和1之间的参数θ，一种广为应用的先验分布是Beta分布，具体定义如下：

$$p(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{4.11}$$

其中，$\alpha > 0$，$\beta > 0$是这个模型的两个参数。容易得出$E[\theta] = \frac{\alpha}{\alpha+\beta}$，$\theta$的最频值为$\frac{\alpha-1}{\alpha+\beta-2}$，方差为$var(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$。于是，如果我们假定$\alpha$和$\beta$都取大于1的值，那么我们可以看到$\alpha$和$\beta$的相对大小控制着均值和最频值的位置：如果$\alpha = \beta$，那么均值和最频值都位于0。如果$\alpha < \beta$，那么最频值小于0.5，等等。

类似地，方差是与$\alpha+\beta$成反比的：$\alpha+\beta$的值控制了先验分布$p(\theta)$的“狭窄程度”。如果α和β是相当大的，那么先验分布是最频值附近的相当狭窄的尖峰。以这

种方式，我们可以选取α和β来反映我们关于参数θ的验前信心（prior belief）。

回忆例4.4，在二项分布下，关于θ的似然函数可以被写为：

$$L(\theta \mid D)=\theta^{r}(1-\theta)^{n-r} \tag{4.12}$$

其中r是总共n个观察值中取值为1的数量。我们可以看到，Beta似然和二项似然在形式上是很相似的：Beta似然看起来像具有$\alpha-1$个验前成功值和$\beta-1$个验前失败值的二项似然。因此实际上，我们可以把$\alpha+\beta-2$看作先验分布的等价样本大小，
118 换句话来说，这就好像我们的Beta先验分布是基于这些验前观测值的。

把似然函数和先验分布结合起来，我们得到：

$$\begin{aligned} p(\theta \mid D) &\propto p(D \mid \theta) p(\theta) \\ &=\theta^{r}(1-\theta)^{n-r} \theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &=\theta^{r+\alpha-1}(1-\theta)^{n-r+\beta-1} \end{aligned} \tag{4.13}$$

这正好是另一个Beta分布的形式，也就是说，θ的后验分布$p(\theta \mid D)$本身是另一个参数为$r+\alpha$和$n-r+\beta$的Beta分布。

于是，这个后验分布$p(\theta \mid D)$的均值为$\frac{r+\alpha}{n+\alpha+\beta}$。这是非常直观的。如果$\alpha=\beta=0$，我们得到标准MLE为$r/n$。否则，我们得到一个修改的估计，新的估计不完全依赖于数据本身（r和n）。例如，在数据挖掘实践中，普遍使用启发式估计$\frac{r+1}{n+2}$作为概率估计，而不用MLE，实际上这相当于使用一个基于后验均值和$\alpha=\beta=1$的Beta先验分布的点估计。这具有使估计避免0和1极端值的“平滑”效果。例如，设想我们要估计超市中一个特定商品被购买的比例，但我们仅有$r=0$的样本D（或许有人购买这种商品的情况相当少见，而且恰好我们抽样那天没有人购买）。这种情况下，MLE为0，然而后验均值为$\frac{1}{n+2}$，当n很大时，这个估计接近0但在这种商品平均每天被购买情况的模型中又允许一个小的（但不为0）概率。

对于高维的（也就是p很大）数据集合，我们可能遇到在我们观察到的数据集D中不会发生某些事情。但是通常不使用MLE把这些事件的概率θ估计为0（这相当于指出根据我们的模型这个事件是不可能的）；而是使用这里描述的贝叶斯估计，这样更稳妥。对于超市的例子，先验分布$p(\theta)$可以来自同一超市的历史记录，也可以来自地理上位于同一地区的多家商店。这样便可以使其他有关（时间和空间）的信息发挥作用了，这就是更加通用的贝叶斯层次模型（这超出了本书的范围）。

贝叶斯方法区别于其他方法的一个主要特征是避免了所谓的点估计（例如参数的最大似然估计），喜欢保留问题中涉及的所有不确定性的全部知识（例如计算关于θ的完整后验分布）。

119 下面举一个例子，设想使用贝叶斯方法预测一个新的数据点$\mathbf{x}(n+1)$，它不属于我们的训练数据D。这里$\mathbf{x}$可能是股票市场每天关闭时的道琼斯指数值，$n+1$是将来的某一天。贝叶斯方法不是使用预测模型来给出$\hat{\theta}$的一个点估计（像我们在最大似然或MAP框架中那样），

而是对θ 的所有可能值进行加权平均，权就是每个可能值的后验分布概率 $p(\theta \mid D)$：

$$\begin{aligned} p(x(n+1) \mid D) &= \int p(\mathbf{x}(n+1), \theta \mid D) d\theta \\ &= \int p(\mathbf{x}(n+1) \mid \theta) p(\theta \mid D) d\theta \end{aligned} \tag{4.14}$$

因为根据定义，对于给定的θ，$\mathbf{x}(n+1)$对与训练数据 D 是条件独立的。实际上，我们可以对此作进一步的扩展，使用一种称为贝叶斯模型平均的技术对不同的模型进行平均。无疑，所有这样的平均过程可能需要比最大似然法大得多的计算量。这是为什么贝叶斯方法近年来才被用于实践（至少已应用于小规模的数据集）的主要原因。对于大规模的问题或高维数据，全面的贝叶斯分析方法可能面临相当大的计算负担。

注意公式 4.9 和 4.10 的结构允许我们可以不断的更新分布。例如，在我们使用数据 D_1 建立模型后，我们可以使用另外的数据 D_2 更新这个模型：

$$p(\theta \mid D_1, D_2) \propto p(D_2 \mid \theta) p(D_1 \mid \theta) p(\theta) \tag{4.15}$$

因为结果独立于数据的顺序（当然，条件是对于给定的模型 P，D_1 和 D_2 是条件独立的），所以对于庞大的数据集，这种可以不断更新的特征是非常有价值的。

公式 4.9 中的分母，$p(D)=\int_{\psi} p(D \mid \psi) p(\psi) d\psi$，被称为 D 的预测分布（predictive distribution），代表了我们对 D 值的预测。通过先验分布 $p(\theta)$ 表示我们对θ 的不确定性；通过 $p(D \mid \theta)$表示当θ 已知时我们对 D 的不确定性。这个预测分布会随着观察到的新数据变化，因此对模型检查是有价值的：如果从预测分布来看观察到的数据仅有很小的概率，那么这个分布不太可能是正确的。

例 4.11 假定我们相信一个数据点 x 来自一个已知方差α但未知均值θ 的正态分布，也就是 $x \sim N(\theta, \alpha)$。现在假定θ 的先验分布是$\theta \sim N(\theta_0, \alpha_0)$，$\theta_0$，$\alpha_0$ 已知。 120
那么，

$$\begin{aligned} p(\theta \mid x) &\propto p(x \mid \theta) p(\theta) \\ &= \frac{1}{\sqrt{2\pi\alpha}} \exp\left(\frac{-1}{2\alpha}(x-\theta)^2\right) \frac{1}{\sqrt{2\pi\alpha_0}} \exp\left(\frac{-1}{2\alpha_0}(\theta-\theta_0)^2\right) \\ &\propto \exp\left(-\frac{1}{2}\theta^2(1/\alpha_0 + 1/\alpha) + \theta(\theta_0/\alpha_0 + x/\alpha)\right) \end{aligned}$$

这里的数学推导看起来相当繁琐（对于贝叶斯方法这是司空见惯的），但如果定义另外两个参数，就会简单很多。设

$$\alpha_1 = (\alpha_0^{-1} + \alpha^{-1})^{-1}$$

和

$$\theta_1 = \alpha_1 (\theta_0/\alpha_0 + x/\alpha)$$

再使用一些代数变换，我们得到

$$p(\theta \mid x) \propto \exp\left(-\frac{1}{2}\theta^2/\alpha_1 + \theta\theta_1/\alpha_1\right) \propto \exp\left(-\frac{1}{2}(\theta-\theta_1)^2/\alpha_1\right)$$

既然这是θ 的概率密度函数，所以它的积分一定为 1。从而，θ 的后验分布具有以下的形式：

$$p(\theta \mid x)=\frac{1}{\sqrt{2\pi\alpha_1}}\exp(-\frac{1}{2}(\theta-\theta_1)^2/\alpha_1)$$

这是一个正态分布 $N(\theta_1, \alpha_1)$。于是正态的先验分布已经被更新为一个正态的后验分布，从而避免了复杂的数学计算。给定关于均值的正态先验分布和来自这个正态分布的数据（即上面所给出的），我们可以仅仅通过计算要被更新的参数得到后验分布。而且，更新参数并不像乍看起来那样复杂。

方差的倒数被称为**精度**（precisions）。在这里更新后的分布精度为 $1/\alpha_1$，也就是先验分布和数据分布的精度之和。这正好验证了“增加数据会降低方差，提高精度”的结论。同样地，更新后的均值θ_1就是先验均值θ_0和数据 x 的加权求和，权为这两个值的精度。

对上面所描述的情况，当有 n 个数据点时，后验分布也是正态的，不过被更新的参数值为：

$$\alpha_1=(1/\alpha_0+n/\alpha)^{-1}$$

和

121

$$\theta_1=\alpha_1(\theta_0/\alpha_0+\bar{x}\,n/\alpha)$$

选择先验分布在贝叶斯分析中起着重要的作用（就像前面所提到的，对于小样本比对大样本更是如此）。先验分布代表了我们对参数取值的初始看法。我们对参数取某些值的信心越大，先验分布就与这些值越紧密。我们的信心越小，先验分布的分散程度也就越大。在正态均值的例子中，如果对真实值一无所知，我们可能会使用对每一个可能值都给出相等概率的先验分布，也就是相当平坦的或具有无限大方差的先验分布。这不会得到任何正常的（proper）密度函数（密度函数必须具有某个非 0 值并且必须积分为 1）。尽管如此，采用相对整个参数空间的不正常（improper）均匀先验分布有时是有价值的。我们可以把这种先验分布看作在参数可能发生的所有区域都是基本平坦的。即便如此，仍然存在对特定参数均匀的先验分布对该参数的非线性变换不均匀的困难。

另一个问题是，先验分布体现了个人对不同参数可能值的验前信心——因此会因人而异，这既可以被看作贝叶斯推理的不足也可以被看作是这种分析的强大之处。你的先验分布和我的不同是完全可能的，所以对于同一个分析我们可能会得到不同的结果。在某种情况下这是好的，但在有些情况下并非如此。克服这一问题的一种方法是使用所谓的参考先验，一种与惯例一致的先验。一种普遍的参考先验是 Jeffrey 先验。为了定义这个先验我们需要首先定义费歇尔信息（Fisher information）：

$$I(\theta \mid \mathbf{x})=-E\left[\frac{\partial^2 \log L(\theta \mid x)}{\partial\theta^2}\right] \tag{4.16}$$

以上是标量的参数θ 的费歇尔信息——也就是，对数似然的二次导数的期望的负数。本质上这个尺度度量了似然函数的曲率和平坦程度。似然函数越平坦，它所能提供的参数信息也就越少。Jeffrey 先验是这样定义的：

$$p(\theta)\propto\sqrt{I(\theta \mid \mathbf{x})} \tag{4.17}$$

这是一个很方便的参考先验，因为如果$\phi = \phi(\theta)$是θ的某个函数，那么就得到一个与$\sqrt{I(\theta|\mathbf{x})}$成正比的先验。这意味着一个一致的先验不受参数变换的影响。

前面所举的例子中的分布是以 Beta 或正态先验开始的，也是以 Beta 或正态后验结束的。 122
共轭族（conjugate family）分布通常满足这一特征：先验分布和后验分布属于同一类分布。使用共轭分布的优点是避免了复杂的更新过程，只要简单地更新参数。

我们已经说明了从后验分布可以很容易的直接得到单一的点估计。得到区间估计也是很简单的——对后验分布在一个区域积分就给出了参数位于这一区域的估计概率。当只包含单一的参数而且估计范围是一个区间时，得到的结果是可信区间（credibility interval）。最短的可能可信区间是包含一个给定概率（例如 90%）从而使后验密度在这个区间最高的区间。如果一个人准备接受基本的贝叶斯思想——参数是一个随机变量，那么这种区间的解释比频率论的置信区间的解释更容易理解。

当然仅包含一个参数的模型是少见的。通常模型都包含多个或很多参数。这种情况下我们可以同时计算所有参数的联合后验分布，或者为每个（或一部分）参数单独计算后验分布。我们也可以研究给定其他参数值后某个参数的条件分布。直到最近，贝叶斯统计仅是一个推理和归纳方面令人感兴趣的哲学观点，没有什么实践价值；从复杂的联合分布得到参数个体的边际分布所需的积分运算过于困难（仅在很少的情况下可以发现分解的解决办法，而且经常需要做出不希望的假定）。然而，在最近的 10 年左右中这个领域已经经历了很多变革。随机的估计方法——以从被估计的分布抽出随机样本为基础——使我们可以估计和研究参数的分布特征。这些方法被称为马尔可夫链蒙特卡罗（Markov chain Monte Carlo, MCMC）方法，我们将在第 8 章中对此作简要讨论。

有必要强调贝叶斯统计的主要特色在于它对待不确定性的方式。贝叶斯方法的哲学是揭示出任何数据分析中的所有不确定性，包括被估计的参数以及模型中的任何不确定性。在最大似然方法中，参数的点估计是考虑的首要目标，但贝叶斯分析者会报告一个关于参数的完整后验分布，以及一个关于模型结构的后验。贝叶斯预测是对参数值和模型结构的 123
加权平均得到的（其中权与参数或模型的似然（对于给定数据）乘以先验成正比）。原则上，与其他（广为应用的）以单一模型为条件对参数进行点估计的方法相比，这种加权平均可以提供更精确的预测。然而在实践中，贝叶斯方法需要加权平均做出估计，这对高维问题是很困难的。此外，如果描述是首要目标，那么参数或模型的加权平均也不太可能产生易于解释的结果。

4.6 假设检验

尽管数据挖掘主要致力于寻找数据中的未知特征（这不同于检验我们见到数据前就形成的假设），但是实践中我们确实经常也需要检验特定的假设（例如，数据挖掘算法产生的我们想进一步探索的令人感兴趣的假设）。

很多情况下，我们需要分析数据是否支持关于参数值的某个设想。例如，我们可能要知道一种新的治疗是否比标准的治疗方法有更好的疗效，或者两个变量是否在总体中有关。因为很多时候，我们不能根据总体来衡量这些假设，所以我们必须基于样本得出结论。探索这些假设的统计工具被称为假设检验（hypothesis test）。

4.6.1 古典假设检验

基本的假设检验原理如下。我们从定义两个互补的假设开始：零假设（null hypothesis）和备选假设（alternative hypothesis）。零假设经常是某一点的值（例如，对讨论的问题影响为 0 的那个点），而备选假设就是零假设的补。例如假定我们要得到关于参数θ的结论。零假设，用H_0表示，可能是$\theta=\theta_0$，于是备选假设可能就是$\theta\neq\theta_0$。使用观察到的数据，我们可以计算一个统计量（统计量的形式最好由被检验假设的属性决定；我们将在下面给出示例）。
124 统计量会因样本的不同而不同——是一个随机变量。如果我们假定零假设是正确的，那么我们可以求出选出统计量的期望分布，并且统计量的观察值是来自这个分布的一点。如果观察值位于分布的很远的末端，那么我们将不得不做出结论：要么是发生了一个低可能事件，要么零假设事实上并不正确。观察到的值越是靠近末端，我们对零假设的信心越小。

我们可以量化这个过程。看一下统计量分布（这个分布基于零假设为真的假定）的末端，我们可以找到发生概率加在一起为 0.05 的那些潜在值。这些是统计量的极端（extreme）值——假定零假设是正确的，这些值与大多数值偏离的足够远。如果这个观察到的极端值确实位于这个末端区域，我们就会“在 5%的显著水平上”拒绝这个零假设：要是零假设是正确的，那么就仅有 5%的可能我们看到发生在这个区域的结果。因此，这个区域被称为拒绝区（rejection region）或临界区（critical region）。当然，我们可能不仅仅对零假设在一个方向的偏离感兴趣。也就是说，我们可能对分布的低端末尾以及高端末尾都感兴趣。这种情况下，我们或许把拒绝域定义为概率分布最低端 2.5%概率对应的检验统计量的值和概率分布的最高端 2.5%概率对应的检验统计量的值的联合。这就是双边检验（two-tailed test），与此相对前面描述的叫单边检验（one-tailed test）。拒绝域的大小，被称为检验的显著性水平（significance level），可以任意选取。常见的值为 1%、5%和 10%。

我们可以按照不同检验过程的能力（power）比较它们。检验的能力就是它正确拒绝错误的零假设的概率。为了评估检验的能力，我们需要指定一个备选假设，目的是计算检验的统计量在备选假设正确的情况下落入拒绝域的概率。

一个重要的基本问题是如何找到适合特定问题的好的检验统计量。一种策略是使用似然率（likelihood ratio）。用来检验假设H_0：$\theta=\theta_0$和备选假设H_1：$\theta\neq\theta_0$的似然率被定义为：

$$\lambda=\frac{L(\theta_0\mid D)}{\sup_{\psi} L(\psi\mid D)} \tag{4.18}$$

其中，$D=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$。也就是说，当$\theta=\theta_0$时似然率达到当$\theta$不被约束时似然的最大
125 值。显然，当λ很小时应该拒绝零假设。这个过程可以被简单地推广到零假设不是单点假设而是包括θ的一系列可能值的情况。

例 4.12 假定我们有一 n 个点的样本，独立采样于一个单位方差均值未知的正态分布，我们希望检验均值为 0 的假设。在这个（零假设）假定下似然为：

$$L(\theta\mid x(1),\cdots,x(n))=\prod_i p(x(i)\mid 0)=\prod_i \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(x(i)-0)^2\right)$$

正态分布的最大似然估计量是样本均值，所以无约束的最大似然是：

$$L(\mu \mid x(1),\cdots,x(n)) = \prod_i p(x(i) \mid \mu) = \prod_i \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x(i)-\bar{x})^2\right)$$

以上二式的比例可以被简化为

$$\lambda = \exp(-n(\bar{x}-0)^2/2)$$

所以，对于一个适当选取的 c 值，我们的拒绝域为$\{\lambda \mid \lambda \leqslant c\}$。这个表达式可以写为：

$$\bar{x} \geqslant \sqrt{-\frac{2}{n}\ln c}$$

其中 $\bar{x} = \frac{1}{n}\sum_i x(i)$ 为样本均值。因此，检验的统计量 $\bar{x}$ 必须与一个常数相比较。

某些类型的检验是被频繁使用的。它们包括不同均值的检验，比较方差的检验，和比较一个观察分布和一个假设分布的检验（所谓的拟合程度（goodness-of-fit）检验）。我们将在下面描述常见的比较两个独立总体均值间差异的 t 检验。其他检验的描述可以参阅介绍统计量的书籍。

例 4.13 设 $x(1)$，…，$x(n)$为从一个正态分布 $N(\mu_x, \sigma^2)$随机抽出的 n 个观察值，并设 $y(1)$，…，$y(m)$为从一个正态分布 $N(\mu_y, \sigma^2)$随机抽出的 m 个观察值。假定我们希望检验这两个分布均值相等的假设，H_0：$\mu_x = \mu_y$。这种情况下似然率统计量被简化为：

$$t = \frac{\bar{x}-\bar{y}}{\sqrt{s^2(1/n+1/m)}}$$

126

其中，

$$s = s_x^2\frac{n-1}{n+m-2} + s_y^2\frac{m-1}{n+m-2}$$

其中，

$$s_x^2 = \sum (s-\bar{x})^2/(n-1)$$

是 x 样本的估计方差，s_y^2 是对于 y 的同样系数。于是 s 就是两个样本的样本方差的加权求和，检验的统计量就是两个样本均值间的差异，再除以估计的差异标准差。在零假设下，t 服从自由度为 $n+m-2$ 的 t 分布。

尽管这里被比较的两个总体被假定为服从正态分布，但是这个检验对于非正态的情况也具有相当好的鲁棒性，尤其是当两个样本的大小和方差大体相等时。这个检验的应用非常广泛。

例 4.14 变量间的关系经常是数据挖掘的中心问题。有时，我们可能想要知道两个变量是否根本无关，以便一个变量值的分布根本不受另一个变量取值的影

响。卡方检验适合检验两个范畴型（categorical）变量的独立性。本质上，这是一种拟合程度的检验，即在检验中把数据与一个独立零假设模型相比较。

假定，我们有两个变量 x 和 y，x 的取值为 x_i，相应的概率为 $p(x_i)$，i =1，…，r；y 的取值为 y_i，相应的概率为 $p(y_i)$，i = 1，…，s。假定联合概率为 $p(x_i, y_i)$。那么如果 x 和 y 是独立的，就有 $p(x_i, y_i)= p(x_i)p(y_i)$。通过简单计算每个观察值落入（fall at）变量 x 的各档值的比例，和落入变量 y 的各档值的比例可以估计出 $p(x_i)$和 $p(y_i)$的分布。设变量 x 取值为 x_i 的估计概率为 $n(x_i)/n$；变量 y 取值为 y_i 的估计概率为 $n(y_i)/n$。在独立的假设下，把这些数据相乘便得到了每个单元（cell）的估计概率；于是，在独立的假定下，$p(x_i, y_i)$的估计概率是 $n(x_i)\ n(y_j)/n^2$。既然总共有 n 个观察值，这就意味着在零假设下我们可以在第（i, j）个单元发现 $n(x_i)\ n(y_j)/n$ 个观察值。为了方便，我们按某种顺序为所有单元从 1 到 $t(t = r \cdot s)$编号，并设 E_k 表示期望在第 k 个单元中看到的数量。我们可以把这一值与在第 k 个单元中观察到的实际数量
127 （我们用 O_k 表示）比较。下面，我们需要以某种方式汇总对所有 t 个单元的比较。一种适合的汇总方法是：

$$X^2 = \sum_{k=1,t} \frac{(E_k - O_k)^2}{E_k} \tag{4.19}$$

这里取平方是为了防止正负差异相互抵消，除以 E_k 是为了防止大的单元支配了这一尺度。如果独立的零假设是正确的，那么 X^2 服从自由度为（r−1）（s−1）的卡方分布，显著性水平要么可以通过查表得到，要么可以直接计算得到。

下面我们以医疗数据为例来加以说明。即根据进行手术的医院种类（推荐的（referral）或非推荐的（non-referral））来对外科手术的结果（没有好转，部分好转，全部好转）进行分类。数据被列在下面，我们感兴趣的问题是手术结果是否独立于医院类型（也就是说，对于两种类型的医院结果的分布是否相同）。

	推荐的医院	非推荐的医院
没有好转	43	47
部分好转	29	120
全部好转	10	118

来自推荐医院的患者总数是（43 + 29 + 10）= 82，根本没有好转的患者总数是（43 + 47）= 90。全部患者是 367。于是，在独立的假定下，表格最左上角单元的期望数量是 82 × 90/367 = 20.11。实际观察的结果是 43，因此这个单元对 X^2 的贡献是（20.11−43）^{2}/20.11。对所有六个单元进行类似的计算，并把结果相加得到 X^2 = 49.8。把这个结果和自由度为（3 − 1）（2 − 1）= 2 的卡方分布相比较，显示了非常高的显著水平，这表明外科手术的结果确实依赖于医院的类型。

上面列举的假设检验策略是以从某个分布抽取随机样本这一假定为基础的，而且检验的目标是对分布的参数做出一个概率陈述。最终的目标是根据样本做出对隐含总体潜在值的推理。出于明显的理由，这种策略有时被描述为采样模式（sampling paradigm）。有时需要另
128 一种策略，特别是当我们不确信样本是通过概率采样（probability sampling）（参见第 2 章）

得到的，因此不可能对隐含总体进行推理的时候。这种情况下，我们有时还是可以对零假设下的某种效果（effect）做出概率陈述。例如考虑对一种治疗和一个控制组的比较。我们可以把零假设取为治疗没有效果，也就是接受了治疗的人和没接受过的人的得分（scores）分布是相同的。如果我们取一个人群的样本（可能不是随机抽取的），并随机地分配到治疗和控制组中，如果零假设是正确的，那么两组间的平均得分差异将是很小的。实际上，在相当广泛（general）的假定下，如果没有治疗效果，或者差异仅是由于随机分配的不平衡而导致的结果，那么求出两个组的样本均值间差异的分布是不困难的。然后我们就可以探索差异与实际观察到的一样大或更大是如何的不可能。基于这一原则的检验被称为*随机检验*（randomization tests）或*置换检验*（permutation tests）。注意，以上过程并没有做出任何从样本到整个总体的统计推理，但它确实允许我们对治疗效果做出条件概率结论，条件就是观察到的数据。

很多统计检验都对从中抽取样本的总体的分布做出了假定。例如，上面演示的 t 检验例子中的两个样本被假定为服从正态分布。然而，很多时候做出这样的假定是不方便的。或许我们对假定没有什么理由，或者因为我们知道数据实际上不服从标准检验所需要的形式。这种情况下，我们可以采用*独立于分布*（distribution-free）的检验。基于排名（rank）的检验属于这一类。在这里，基本数据被替换为它们的对应位置的数字标签。例如，为了探索两个样本是否来自同一个分布，我们可以把他们的实际数值替换为它们的排名。如果它们确实来自同一分布，我们就可以期望这两个样本成员的排名是均匀混合的。而且，如果一个分布比另一个具有更大的均值，那么我们可以期望这个样本趋向于有较高的排名，而另一个具有较低的排名。如果两个分布具有同样的均值但一个的方差比另一个的大，那么我们可以期望一个样本倾向于高的和低的排名，而另一个占据了中间的排名。可以根据排名的平均值或关于排名的其他尺度来建立统计量，而且可以用随机检验理论来评估它们的显著性水平。这样的统计量包括 sign 检验统计量、Kolmogorov-Smirnov 检验统计量和 Wilcoxon 检验统计量。有 129
时术语*非参数检验*（nonparametric test）是用来描述这种检验的——这个命名根据的是这些检验不检验任何假设分布的参数的值。

从贝叶斯的观点来看，对假设 H_0 和 H_1 的比较可以通过比较它们的后验概率来实现：

$$p(H_i \mid x) \propto p(x \mid H_i)p(H_i) \tag{4.20}$$

取两个假设的比就得到了以先验赔率（prior odds）和似然率表示的因式，也就是贝叶斯因子（Bayes factor）：

$$\frac{p(H_0 \mid x)}{p(H_1 \mid x)} \propto \frac{p(H_0)}{p(H_1)} \cdot \frac{p(x \mid H_0)}{p(x \mid H_1)} \tag{4.21}$$

然而这里还存在一定的复杂性。似然是通过对假设中未指定参数的积分而得到的*边缘似然*（marginal likelihoods），如果 H_i 是指连续可能值（例如，参数 θ 的值，θ 可以取 0 到 1 间的任意值）中的某一个，那么先验概率将为 0。处理这种问题的一种策略是为 θ 的给定值赋一个离散的非零先验概率。

4.6.2 数据挖掘中的假设检验

目前为止本节所描述的都是假设检验的经典（频率论的）方法。然而在数据挖掘中，分

析可能变得更为复杂。

首先，因为数据挖掘所针对的是庞大的数据集，所以我们应该对统计显著性有所戒备：即便假设模型形式的微小偏差也可能被确认为是非常显著的，即使这些变化根本没有实践意义。（如果他们是有实践意义的，那当然很好。）更糟的是，由于数据污染或失真产生的微小偏差也会很显著地表现出来。而且我们已经指出这种数据质量的问题是不可避免的。

第二，系列化的（sequential）模型拟合过程是很常见的。我们将从第 8 章开始描述各种分步的（stepwise）模型拟合过程，通过增加或删除某些项来逐步提炼模型。对每一步中的模型进行分别的检验（就好像又是一个新的）导致出错的概率增大。已经开发出了正规的系
130 列检验过程，但非常复杂。而且，由于同时进行多个检验，这些过程可能很脆弱。

第三，数据挖掘本质上就是一个具有很多未知因素的探索过程。很重要的一点是数据挖掘中将分析很多个模型。假定我们在 5%的水平检验出 m 个正确的零假设（尽管这是我们无法知道的），每一个基于它自己的数据子集，独立于其他的检验。对于每一个备选假设来说，存在 5%的概率错误拒绝了这个假设。既然检验是独立的，那么至少错误拒绝一个假设的概率是 $p=1-(1-0.05)^m$。当 $m=1$ 时，$p=0.05$，这是可以的。但当 $m=10$ 时，$p=0.4013$；当 $m=100$ 时 $p=0.9941$。因此，如果我们仅检验了 100 个正确的零假设，我们几乎就肯定错误地拒绝了至少一个备选假设。另一种做法是，我们可以控制总体的族（family）错误率，设定错误拒绝 m 个正确的零假设中的一个或多个的概率为 0.05。这时我们使用 $0.05=1-(1-\alpha)^m$，对于给定的 m 求出检验每一个零假设的显著水平α。对于 $m=10$，我们达到$\alpha=0.0051$；对于 $m=100$，我们达到$\alpha=0.0005$。这意味着我们有很小的概率错误拒绝任何一个单独的假设组成部分。

当然，实践中情况会更加复杂：假设不可能是完全独立的（对于极端的情况，如果假设是完全依赖的，接受或拒绝一个假设就意味着接受或拒绝了全部）；要处理本质上独立性不可知的结构；而且通常都是正确的（或大致正确）和错误的零假设混合在一起。

已经开发出了很多同步的检验过程（simultaneous test procedures）来缓解这些困难。一种基本的方法是基于 Bonferroni 不等式。我们可以把没有拒绝任何正确零假设的概率$(1-\alpha)^m$加以扩展得到$(1-\alpha)^m \geqslant 1-m\alpha$。它是根据 $1-(1-\alpha)^m \leqslant m\alpha$——也就是，一个或多个零假设被错误拒绝的概率小于或等于 $m\alpha$——推导而来的。通常，错误拒绝一个或多个正确的零假设的概率小于错误拒绝它们中每一个的概率的和。这就是一阶 Bonferroni 不等式。通过在展开式中包含其他项，我们可以推出更加精确的边界——尽管它们需要假设间依赖关系的信息。

对于某些检验过程，可能发生这样的问题：对假设族的全局检验拒绝了零假设（所以我们相信至少其中之一是错误的），但是却没有任一个单独的假设组成部分是被拒绝的。人们
131 也已经开发出了一些策略，用来克服特定应用中的这种问题。例如，在对方差的多变量分析中，需要比较已经测量了多个变量的几组对象，人们开发出了克服以上问题的检验过程，做法是用一个单一的阈值比较每一个检验统计量。

从上面的讨论可以清楚地看到，尽管通过假设检验为不同类型的结论给出不同的概率在数据挖掘中确实占有一席之地，但是这种方法还没有形成一套完整的解决方案。不过，可以把这种方法看作一种更一般的过程——即把数据或结论映射为一个数字值或者说分数——的一个特例。较高的分数（或较低的，依赖于具体过程）表示一个结论或模型优于其他的，而不需要做任何绝对的概率解释。可以认为第 7 章中描述的惩罚性拟合程度评分函数是属于这一框架的。

4.7 采样方法

前面曾经指出，数据挖掘是一种次级的数据分析，因此数据挖掘者一般不参与直接的数据采集过程。然而，如果我们有关于数据采集过程的信息，那么对我们的分析可能是有价值的，我们应该发挥这些信息的优势。传统的统计数据采集通常是从回答某个或某些特定问题的角度，使用某种高效的方式来进行的。然而，既然数据挖掘是发现意外的（unexpected）或无法预计的（unforeseen）信息，所以数据挖掘不是要回答数据收集前就已确定的问题。由于这个原因，我们不会介绍统计中被称为试验设计（experimental design）的子学科，因为它主要研究采集数据的最佳方法。数据挖掘者通常对数据采集过程没有任何控制的事实有时解释了数据质量低劣的原因：数据可能对被采集的目的很理想，但对于数据挖掘是不够理想的。

我们曾经指出，如果数据库包含了整个总体，那么统计推理的思想就没有用了：如果我们想知道某个总体参数的值（比如说，平均交易额，或最大交易额），那么计算出来就可以了。当然，这里假定数据理想地描述了总体，不存在测量误差，数据残缺（missing data），数据损坏，等等。不过正如我们所看到的，这是不可能的条件，所以我们还是要根据记录的数据来对“真实的”潜在总体值做出推理。 132

此外，有时总体和样本的概念可能会产生误导。例如，即使总体值已经被捕捉到数据库中，但多数情况下我们的目标并不是描述总体，而是要做出关于将来可能值的某个结论。例如，我们可能已经得到某一天一个连锁超市的销售数据的总体。这时我们可能非常希望得出某种推理性的结论——指出下一天或将来某一天的平均销售额。这也涉及很多不确定性，但这与前面讨论的有所不同。实质上，这里我们所关心的是预报（forecasting）。在市场分析中，我们实际上不是要描述上个月的顾客购买模式，而是要预报顾客下个月的可能行为。

我们已经区分了数据挖掘中样本产生的两种方式。第一，有时数据库本身仅是更大总体的一个样本。第 2 章中我们描述了这种情况的含义和与之联系的风险。第二，数据库包含了总体中所有对象的记录，但数据分析仅是以从中抽取的一个样本为基础的。后一种技术仅适合于建模的情况和某些模式识别的情况。当我们要寻找异常的记录个体时它是不合适的。

我们的目标是从数据库中抽取一个样本以建立一个反映数据库中数据结构的模型。仅使用一个样本，而不使用整个数据集的原因是效率。对于极端的情况，使用庞大的整个数据库在时间和所需运算方面可能是不可行的。通过仅对样本进行运算，我们可以使计算变得更简单和更迅速。然而，抽取的样本应该反映完整集合的结构是非常重要的——也就是要保证样本代表了整个数据库。

有很多种策略保证抽取的样本具有代表性。如果我们仅要从两条记录中取一条（*采样率*（sampleing fraction）为 0.5），那么我们可以简单地每隔一条记录取一条。这种直接的方法被称为*系统采样*（systematic sampling）。很多时候这种方法是足够好的。然而，也可能导致意外的问题。例如，如果数据库包含已婚夫妇的记录，丈夫和妻子是对应不同记录的，那么系统采样的结果可能是极差的——得出的结论可能是完全错误的。一般来说，在任何按照某一规律选取案例的采样模式中都存在和数据库中的未知规律相作用的风险。显然我们需要的是一种避免规律性的选择模式——一种随机的选择模式。 133

这里使用随机一词的含义是为了避免规律性。这与本章前面使用这个词的用法略微不

词，前面这个词是指选取样本的机制，它描述了一条记录被选作样本的概率。可以看出，具有第二种随机含义的样本可以用作统计推理的基础；例如，我们可以做出有多大可能样本均值和总体均值间会存在本质差异的结论。

如果我们使用随机过程抽取一个样本，那么这个样本满足第二种含义，而且也可能满足第一种。（实际上，如果我们明确指出我们所指的“规律”是什么，那么我们就可以给出随机抽取的样本不匹配这个规律的精确概率。）为了避免我们的结论存在偏差，我们应该把样本选取机制设计为数据库中的每一条记录具有相同的被抽取机会。总体中每个成员具有相同被抽取概率的样本被称为 epsem（等概率选择每成员）样本。等概率的 epsem 采样形式是简单的随机采样，也就是从数据库中的 N 条记录中抽取 n 条记录的样本，抽取的方式是保证 n 条记录中的每一条被抽取的概率都是相同的。简单随机样本对总体均值的估计就是样本均值。

现在我们应该指出放回（replacement）抽样和不放回抽样的差异。对于前者，一条已经抽取的记录有机会被再次抽取，但对后一种情况，一条记录一旦被抽出就不可能被第二次抽到。在数据挖掘中，因为样本容量相对总体容量经常是很小的，所以这两种过程结果的差异通常是被忽略的。

图 4-5 演示了一个简单随机抽样过程的结果，样本是用来计算某个总体的一个变量的均值的。总体的真实均值为 0.5。随机抽取指定容量的样本，然后计算它的均值；我们重复这个过程 200 次并画出了结果的直方图。

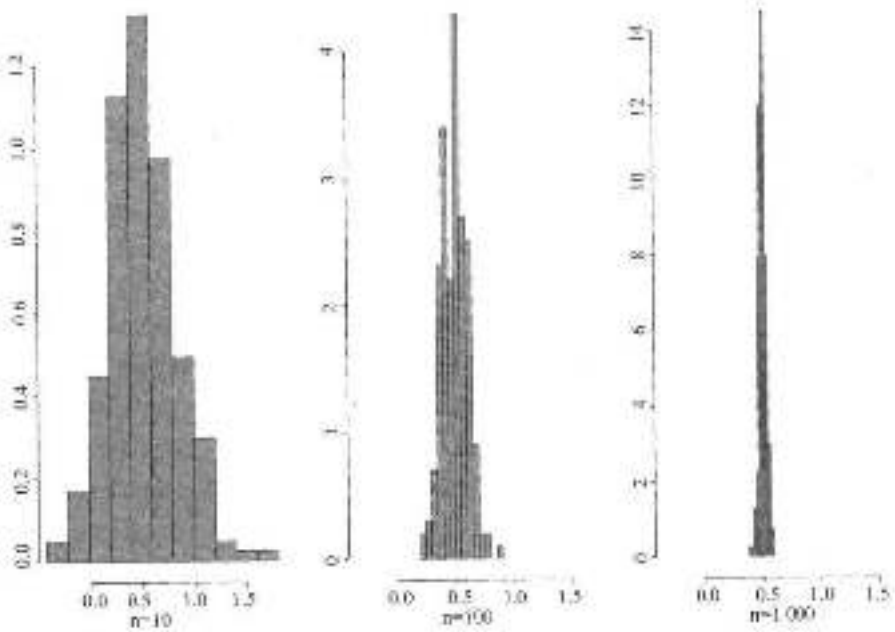


图4-5 显示了样本容量为10（a），100（b）和1 000（c）时计算出的样本均值分布

从图中可以明显看出，样本越大，样本均值分布就越靠近真实的均值。通常，如果大小为 N 的总体的方差是 σ^2，那么从这个总体抽出的大小为 n 的简单随机样本（不放回抽样）的均值的方差为：

134

$$\frac{\sigma^2}{n}(1-\frac{n}{N}) \tag{4.22}$$

因为通常我们要处理的情况是相对 n 来说，N 是很大的（也就是涉及较小采样率的情况），所以我们通常可以忽略第二项，于是这个方差的一个很好近似就是σ^2/n。由此可以得出，样本越大，样本均值显著偏离总体均值的可能性越小——这解释了为什么图 4-5 中的直方图随着样本容量的增大散布得越来越窄。也要注意这个结果是独立于总体容量的。这里起作用的是样本容量，而不是采样率的大小，也不是包含这个样本的总体的大小。我们也可以看到，当样本容量加倍时，标准差不是按因子 2 减小的，而仅是按$\sqrt{2}$——存在对样本容量增加的扣减。我们根据样本使用以下标准估计量来估计σ^2：

$$\Sigma(x(i)-\bar{x})/(n-1) \tag{4.23}$$

其中 $x(i)$是第 i 个样本单元的值，$\bar{x}$ 是样本中 n 个值的均值。

简单随机样本是样本设计的最基本形式，但也已经开发出了具有要求属性的其他方法以适合不同的应用环境。对此的详细论述可以查阅有关调查采样的书籍，比如本章末尾所列出的那些。这里我们简要描述两种重要的采样模式。

在*分层随机采样*（stratified random sampling）中，总体被分成不重叠的子总体或称为层（strata），然后从每一层中分别抽出一个样本（经常是简单随机样本，但不是必须的）。使用这一过程有很多潜在的优势。很明显的一个优势是这样我们可以对每一个子总体分别做出结论，不需要担心无法保证每一个子总体都有一定数量的观察值。一个更微妙但经常更重要的优势是，如果从某个感兴趣变量的角度来看，每个层是同质的（homogeneous）（从而使变量之间的大多数变化都反映在了层间的差异上），那么整体估计出的方差可能比从简单随机样本估计出的小。为了说明这一点考虑以下例子。一家信用卡公司把它的交易分成 26 个类目：超市、旅行代理、加油站，等等。假定我们要估计交易的平均值。我们可以从数据库的 135
记录中取一个简单随机样本，并计算它的均值作为我们的估计。然而，使用这样的过程，某些交易类型可能在我们的样本中没有被充分代表（underrepresented），而代表某些类型的数据可能又过多（overrepresented）。我们可以通过强制为每一种交易类型包含一定数量的数据来对此进行控制。这就是分层采样，交易类型就是层。这个例子说明了为什么层必须是内部相对同质的，异质发生在层间。如果所有的层都有和整个总体相同的散布，那么分层是没有任何优势的。

通常，假定我们要估计对于某一变量的总体均值，而且我们使用一个分层的样本，在每一层中使用简单随机采样。假定第 k 层中有 N_k 个元素，而且这些值中有 n_k 个被抽出作为这一层的样本。用 $\bar{x}_k$ 表示第 k 层的样本均值，总体均值的估计可以通过下式给出：

$$\sum\frac{N_k\bar{x}_k}{N} \tag{4.24}$$

其中 N 是总体的总容量。这个估计量的方差是

$$\frac{1}{N^2}\sum N_k^2 var(\bar{x}_k) \tag{4.25}$$

其中 $var(\bar{x}_k)$ 是第 k 层大小为 n_k 的简单随机样本的方差。

数据经常具有一种层次结构。例如，字母出现在词汇中，词汇在句子中，句子组成段落，

段落出现在章节中，章节形成书籍，很多书籍组成图书馆，等等。有时建立一个完整的采样框架并抽出一个简单随机样本是很困难的。一个组织可能有很多网络站点，每个站点有很多不同的计算机，每台计算机上有很多文件，如果我们要研究这些文件，那么我们可能发现要产生一个完整的文件列表以便从中进行简单随机采样是不可能的。在成簇（cluster）采样中，不是抽取我们感兴趣的元素个体作为样本，而是抽取包含多个元素的单元作为样本。在计算机文件的例子中，我们可能抽出多台计算机作为样本。我们可以分析每一台抽出的计算机上的所有文件，或者也可以进行进一步的采样。

各个簇的大小经常是不等的。在上面的例子中，我们可以把计算机看作提供了文件簇，一个组织的所有计算机内具有同样数量的文件几乎是不可能的。但具有相同大小簇的情况确实存在。工业生产提供了很多这样的例子：例如六瓶一箱的啤酒。如果每一个被选出的簇的所有单元都被选出（如果二次采样率是 1），那么每个单元被选择的概率是 a/K，其中 a 是从 K 个簇的整个集合中选出的簇数。如果并非所有单元都被选出，但每一簇中的采样率是相同的，那么每个单元有相等的概率被样本所包含（这个样本将是一个 epsem 样本）。这是一种
136 ~ 137 常见的设计。基于这一设计的统计量的方差估计不如前面所描述的情况那样直接，因为样本大小也是一个随机变量（它依赖于哪一簇恰好被样本所包含）。此时变量均值的估计是两个随机变量的比率：样本中包含的单元总和和样本中包含的单元总数量。用 n_k 表示从第 k 个簇抽出的简单随机样本的大小，用 s_k 表示从 k 层选取的单元的总和，那么样本均值 r 为：

$$\sum x_k / \sum n_k \tag{4.26}$$

如果我们用 f 表示整个样本采样率（经常是很小的所以可以忽略），那么 r 的方差是：

$$\frac{1-f}{(\sum n_k)^2}\frac{a}{1-a}\left(\sum s_k^2 + r^2\sum n_k^2 - 2r\sum s_k n_k\right) \tag{4.27}$$

4.8 本章归纳

事无定论。在数据挖掘中，我们的目标是从数据中寻找新的发现。我们希望对我们的结论的正确性尽可能地信心十足，但是很多时候我们必须满足于一个可能错误的结论——尽管如果我们可以同时指出我们对结论的置信程度会好一些。当我们分析整个总体时，不确定性会通过不尽人意的数据质量悄悄混进来：某些值可能是被记录错误了；某些值可能是残缺的；总体的某些成员可能被整个数据库所遗漏了；等等。当我们工作在样本上时，我们的目的经常是得出一个结论，这个结论可以应用到从中抽取样本的更广阔总体。对付这些问题的基本工具是概率。这是处理不确定性的统一语言，一种在上个世纪中一直被提炼的语言，而且它已经被应用到无数的领域。概率思想的应用使我们能够得到最佳的估计值，即使是面对数据不够充分的情况，甚至当仅仅测量了一个样本的时候。而且，应用这种思想，我们还可以量化我们对所得结论的把握。

本书的其余部分大量地应用了概率理论。这些理论是很多——甚至是大多数——数据挖
138 掘工具的基础，从全局的建模到局部的模式识别。

4.9 补充读物

讨论不同概率学派以及统计推理的著作包括 DeFinetti（1974，1975），Barnett（1982），Bernardo and Smith（1994）。关于统计量和特定统计模型的其他参考文献在第 6 章、第 9 章、第 10 章和第 11 章的末尾给出。

有很多关于基本概率计算的好书，包括 Grimmett and Stirzaker（1992）和 Feller（1968，1971）。Hamming（1991）是面向工程师和计算机科学工作者的（包含了很多有趣的示例）一本教科书，Applebaum（1996）是面向数学专业研究生的。概率计算是应用数学的活跃领域，而且它也大大得益于它所应用的各个领域。例如，Alon and Spencer（1992）描绘了概率在现代计算机科学中应用的迷人之旅。

关于 Kolmogorov 复杂性的著作（例如 Li and Vitanyi，1993）讨论了避免规律和可预测性的随机思想。

Whittaker（1990）精彩论述了在图形模型中处理条件依赖和独立的一般原则。Pearl（1988）是一本从人工智能的角度探索这一领域的奠基之作。

有大量关于统计推理的入门教材，例如 Daly et al.（1995），还有一些提高性教材深入的讨论了推理概念，比如 Cox and Hinkley（1974），Schervish（1995），Lindsey（1996），Lehmann and Casella（1998），and Knight（2000）。Edwards（1972）对似然及其应用进行了广泛的讨论。目前，贝叶斯方法是几乎所有书籍的一个主题。Gelman et al.（1995）是关于贝叶斯方法的一本很好教材。Bernardo and Smith（1994）是关于贝叶斯方法的一本全面的参考书，Lee（1989）作了比较简要的介绍。讨论非参数方法的著作有 Randles and Wolfe（1979）和 Maritz（1981）。Efron and Tibshirani（1993）介绍了 bootstrap 方法。

Miller（1980）介绍了同步检验过程。我们前面列举的对多参数推理问题的解决方法不是仅有的，Lindsey（1999）描述了其他方法。

关于调查采样的书讨论了抽取样本的高效策略——例如，Cochran（1977）以及 Kish（1965）。

第5章　数据挖掘算法概览

5.1　简介

这一章我们从一般意义上来探讨一下数据挖掘算法及构成这些算法的组件。我们对数据挖掘算法的定义是：

> *数据挖掘算法是一个定义完备的（well-defined）过程，它以数据作为输入并产生模型或模式形式的输出。*

定义完备（well-defined）指的是这个过程可以被精确地编码为有限的规则。作为一个算法，它的过程必须总能在有限步后终止并输出结果。

相对而言，计算方法具备除了不能保证过程在有限步后终止外的所有算法特征。通常在算法的说明中定义了许多实际的实现细节；而计算方法一般只进行比较抽象的描述。例如，最陡峭下降（steepest descent）搜索技术是一种计算方法，它本身并不是一个算法（这种搜索方法是在参数空间里沿着使评分函数（score function）相对当前参数值最陡峭下降的方向不断移动）。要使用最陡峭下降方法来定义一个算法，我们需要给出精确的方法来确定从哪里开始下降，怎样确认最陡峭下降的方向（是要精确计算还是大约估计？），要在选定的方向上移动多远，以及什么时候终止搜索（例如，检测到收敛在一个局部极小值）。 141

正如第一章中所简要讨论的，求解某个特定任务的数据挖掘算法的说明中包含了算法组件的具体定义：

1．该算法所针对的数据挖掘任务（例如，可视化、分类、聚集、回归等等）。通常，不同的任务需要不同类型的算法。

2．用于拟合数据的模型或模式的结构（函数形式），例如线性回归模型，层次聚类模型等等。这个结构定义了我们可以近似或学习的边界。在这个边界范围内，数据引导我们得到特定的模型或模式。在第6章里我们将更加详细的讨论数据挖掘算法中所广泛应用的模型或模式结构。

3．用于根据观察到的数据判断拟合后的模型或模式质量的评分函数（例如误分类率或误差平方等）。正如第7章将要讨论的，评分函数就是当我们把参数和模型及模式拟合起来时要最大化或最小化的函数。因此，评分函数在反映模型或模式的不同参数化过程的实际效果方面是很重要的。此外，评分函数对于学习和泛化也是至关重要的。它可以仅仅基于拟合完满度（也就是模型多好的描述了观察数据），也可以尽可能的捕捉泛化性能（也就是模型多好的描述了我们没有见到过的数据）。在后面的章节中我们会看到，这是一个很有讲究的问题。

4．用于对参数或结构进行搜索的搜索方法或优化方法，也就是使评分函数相对特定的模型或模式最大化（或最小化）的计算过程或算法。这里的问题包括优化评分函数（例如，最陡峭下降）以及与搜索相关的参数（例如，迭代的最大次量以及迭代算法的收敛性）。如果模型（或模式）结构是简单且固定的（例如输入数据的 k 阶多项式函数），那么搜索将在

参数空间里进行，目的是相对这个固定结构形式优化评分函数。如果模型（或模式）的结构包含一组（或一族）不同的结构，那么搜索既要针对这些结构又要针对和这些结构相联系的
142 参数空间。优化和搜索通常是所有数据挖掘算法的核心部分，我们将在第 8 章中非常详细地讨论这个内容。

5．用于存储、索引、检索数据的数据管理技术。许多统计和机器学习算法并不指定任何数据管理技术，实质上是假定数据集足够小可以驻留在主存储器中，以至于相对于总的实际计算开销，随机访问任何数据点的时间都是可以忽略的。然而，大规模数据集可能超过了现有主存储器的存储能力，因而驻留在二级（例如磁盘）或三级（例如磁带）存储器中。访问这样的数据显然要慢于访问主存储器中的数据，因此，对于大规模数据集，数据的物理位置和访问方式对于算法的效率是至关重要的。有关数据管理这一方面的内容将在第 12 章里作更深入的讨论。

表 5-1 演示了如何把三个著名的数据挖掘算法（CART，反向传播（backpropagation），（A Priori）算法）按它们的基本组件来描述。本章后面将详细的讨论这三个算法中的每一个。（从表中可以容易地看出统计学与数据挖掘的一点不同。统计学家会认为 CART 是一个模型，反向传播是一个参数估计算法。 而数据挖掘更倾向于从算法的角度看问题：用算法处理数据以产生结果。这个差异完全是观察角度上的而不是实质上的。）

表 5-1 把三种著名的数据挖掘算法分解成算法组件

	CART	反向传播	A Priori
任务	分类和回归	回归	规则模式发现
结构	决策树	神经网络（非线性函数）	关联规则
评分函数	交叉验证的损失函数	误差平方	支持度/精度
搜索方法	“贪婪”搜索各种结构	对参数进行梯度下降	带修剪的广度优先
数据管理策略	未指定	未指定	线性扫描

确定模型（或模式）结构以及评分函数的过程通常是“脱机的”，属于解决数据挖掘问题过程中以人为中心的那一部分。一旦数据、模型（或模式）结构、以及评分函数都确定下来，那么剩下的问题——优化评分函数——很大程度上是计算上的了。（实践中，由于要根据前一次的结果改进模型和评分函数，所以这个过程要重复很多次。）因此，数据挖掘算法的算法核心是用来实现搜索和数据管理部分的计算方法。

本章给出的对数据挖掘算法的基于组件的描述为数据挖掘算法的分解与合成提供了一个高层次的框架。从分解的角度来看，以分解形式描述现有的数据算法可以阐明每个组件的作用，而且可以更容易的比较多个类似的算法。例如，可以根据每个组件来判断两个算法之间的区别，比较它们在模型结构、评分函数、搜索方法或数据管理策略方面是否不同。从合成的角度来看，以不同的组合方式将不同的组件合成在一起，就能建立起具有不同性质的算法。在第 9 到第 14 章里我们将结合具体的算法更详细地讨论每个组件。在这一章里我们将集中从宏观上讨论怎样把各个部分组合在一起。本章的主题是对数据挖掘算法的基于组件观点为数据挖掘算法的描述、分解以及合成提供了一种简洁而且结构化的语言。

对于大部分内容，我们的讨论仅限于只有单一模型或模式结构（比如树、多项式等等）的情况，不考虑对同一问题使用多种类型模型结构的那些情况。当然组件观点也可被推广到

可以处理这些情况，但通常评分函数，搜索方法，以及数据管理技术都会变得更加复杂。 143~144

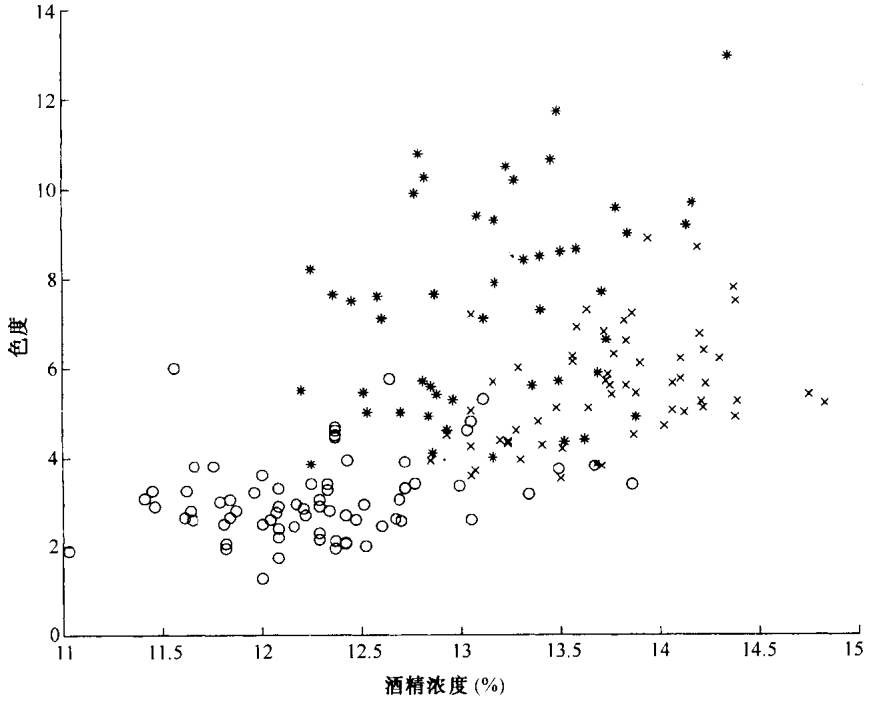


图5-1 **色度**对**酒精浓度**的散点图。数据挖掘的任务是要将酒分为三类（三个不同的品种），它们在图上有各自不同的标记。数据最初来自于一个13维的数据集，其中的每个变量衡量了酒的某种特征

5.2 建立树分类器的 CART 算法

为了阐明按算法组件分析算法的一般思想，我们首先看一个用于分类问题的著名算法。

CART（分类和回归树，Classification and Regression Trees）算法是一种广泛应用的基于树结构产生分类和回归模型的统计过程。为简便起见，我们只考虑 CART 算法的分类功能， 145
也就是将一个输入向量 **x** 映射到一个范畴型的（类）输出标记 y（参见图 5-1）。（关于 CART 算法更多的细节讨论将在第 10 章中给出。）按照上面讨论的组件说法，可以把 CART 看作是由以下组件构成的“算法组合（algorithm-tuple）”：

1. *任务* = 预测（分类）
2. *模型结构* = 树
3. *评分函数* = 交叉验证的损失函数（cross-validated loss function）
4. *搜索方法* = “贪婪”局部搜索（greedy local search）
5. *数据管理方法* = 未指定

CART 算法的突出特征是其应用的模型结构——分类树。CART 树模型为一个分层的一元 146
二叉树结构。图 5-2 给出了这种分类树的一个简单例子它对应于图 5-1 中的数据。树中的每一个内部节点指定了一个对单一变量的二择一测试，对于实数值变量和整数值变量使用的是用阈值；对于范畴型变量使用的是隶属关系子集。（通常，我们在每个节点使用 b 个分支，$b \geq 2$。）

一个数据向量 **x** 由树根沿唯一路径下降到某个叶子节点，具体的路径取决于 **x** 的各个分量在内部节点的二择一测试结果。每个叶子节点指定了那个叶子的最可能分类的类标签，或更准确地说，叶子结点指定了分类值的条件概率分布，条件就是通往这个叶子的分支。

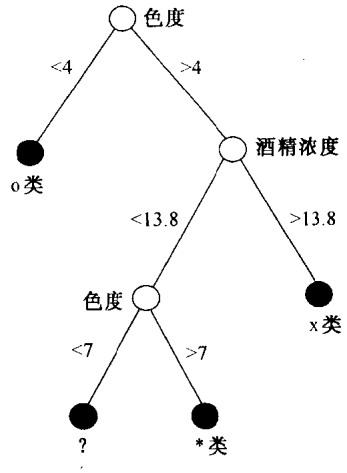


图5-2　图5-1中数据的分类树，其中测试是由内部节点上变量的阈值（显示在分支旁边）组成的，叶子节点包含了分类决定。注意标有问号的叶子，这是表示这个区域数据点的分类标签有相当大的不确定性

树的结构是由数据得来的，而不是预先确定的（这正是数据挖掘起作用之处）。CART是这样工作的，首先方式是用最好的变量将数据在根节点处分为两组。它可以在几种不同的分裂标准中任选一种；所有标准实质上都是将数据在中间节点处划分为两个不相交的子集（分支），同时使每个子集中的分类标签尽可能同质。然后对每个子节点上的数据重复地应用这种分裂方法，等等。最终树的大小是由下面将要讲的复杂的“修剪”过程所决定的。如果树太大可能会导致过度拟合，但太小又不能为精确分类提供足够的预测能力。

树结构的分层形式将 CART 这样的算法从其他基于非树结构（例如，使用所有变量的线性组合在给定空间里定义边界的模型）的算法中清楚的区分出来。用于分类的树结构很容易处理混合类型的输入数据（例如，范畴型数据和实数值数据的组合），因为每一个内部节点都依赖于唯一一个简单的二择一测试。此外，因为 CART 每次只用一个变量建立树，因此它可以容易地处理大量的变量。另一方面，树结构的表示力是比较粗糙的：用于分类的决策区域局限于超矩形，而且矩形的边局限于和输入变量坐标轴平行（参考图 5-3）。

用于衡量不同树结构质量的评分函数通常是误分类损失函数，被定义为：

147

$$\sum_{i=1}^{n} C\big(y(i), \hat{y}(i)\big) \tag{5.1}$$

其中 $C\big(y(i), \hat{y}(i)\big)$ 是因为这个树将第 i 个数据向量的分类标记 $y(i)$ 预测为 $\hat{y}(i)$ 而导致的损失（正的）。一般地，C 是由 $m \times m$ 的矩阵确定的，其中 m 是分类的数量。为简单起见，假设当 $\hat{y}(i) \neq y(i)$ 时，损失为 1，否则为 0。（这被称为“0-1”损失函数，或者如果再进一步将上面的和被 n 除，那么便得到误分类率。）

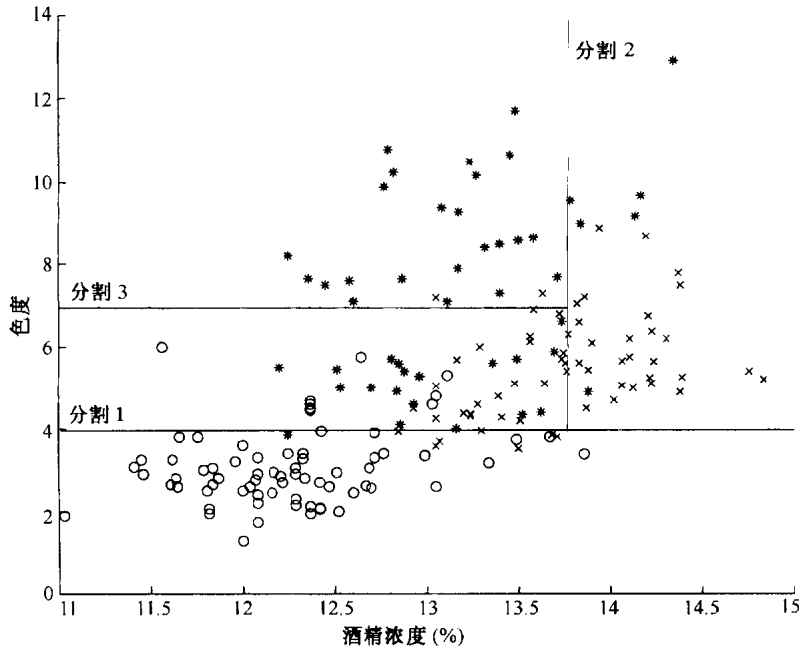


图5-3 把图5-2中分类树的决策边界叠加到原始数据上。注意决策边界与坐标轴的平行特征

CART 使用一种被称为交叉验证的技术来估计误分类损失函数。在第 7 章里我们将更加详细的介绍交叉验证。简单来讲，这种方法先从训练数据中划分出一个子集用于建立树，然后在剩余的验证子集里估计误分类率。然后针对不同的子集多次重复这种划分，再对得到的误分类率进行平均从而得到关于特定大小的树对于新的未见过数据的性能的交叉验证估计。 148
产生最小交叉验证误分类估计的树的大小被确定为最终树模型的合适大小。（上面的描述概括了通过交叉验证选择树的关键环节，在实践中这个过程会更复杂一些。）

交叉验证使 CART 可以估计出树模型对于在构造树时没有使用过的数据的性能——也就是，它提供了一种对泛化性能的估计。这在增长树的过程中是至关重要的，因为对于训练数据（用于构造树的数据）的误分类率，只要通过提高树的复杂度经常就可以减小；因此，一个分类树对于训练数据的误差不一定可以表示出这个树对新数据的性能。

图 5-4 利用典型误差率相对树大小的假想函数曲线说明了这一点。可见，对于训练数据，误差率单调下降。（如果变量为每个单独类的数据产生一个叶子，那么误差率将下降为零）。对于新的数据（通常这才是我们所感兴趣的希望做出预测的数据），测试误差率起初也是下降。这是因为非常小的树（靠左边）没有足够的预测力做出精确的预测。然而，与训练误差不同的是，测试误差到了“最低点”后又开始上升，这是因为算法过度拟合了数据，增加节 149
点仅仅预测了训练数据中的噪声和随机波动，这些噪声和波动和预测任务并不相关。像 CART 这样算法的目的是要找到与最佳树大小（当然事先是未知的）接近的树。它试图找到一个足够复杂的模型以捕捉任何存在的结构，但是不能过度拟合。对于少量到中等数量的数据，最好是不要保留用以估计样本外误差的数据。对于非常大的数据集，可以把数据划分为训练和验证数据集，然后在验证数据集上监控模型的性能。

交叉验证评分函数的使用将 CART 与其他基于树模型的数据挖掘算法区分开来。例如，C4.5 算法（另一个广泛用于构建分类树的可以代替 CART 的算法）通过启发式地调整在训

练数据上估计出的误差率来近似测试误差率（试图纠正训练误差率通常低估样本外误差率的事实）以判断各个树结构。然后在修剪阶段使用调整的误差率以找到使评分函数最大化的树。

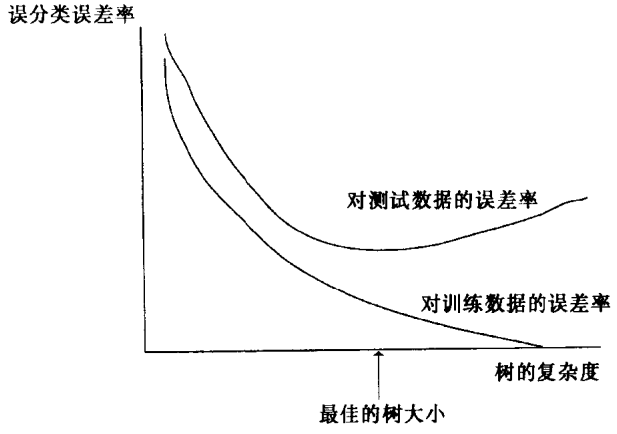


图5-4　误分类误差率相对树复杂度（例如树上树叶的数目）的假想函数曲线

CART 用“贪婪”局部搜索方法来确定好的候选树结构：从根节点递归地扩展树，然后再逐渐地“修剪”掉这个树的特定分枝。这种启发式的搜索方法是以下两个因素共同作用的结果：大规模的搜索空间（例如，所有可能的二叉树结构空间）和没有任何易驾驭的方法可以找到唯一的最优树（相对于给定的评分函数）。在树学习中的一句流行名言就是：“贪婪”的局部搜索和更复杂的启发式方法工作的几乎一样好，而且它也比其他更复杂的搜索方法要易于实现得多。因此，“贪婪”局部搜索是大多数实际树学习算法的首选方案。

在数据管理方面，CART 隐含假定数据都在主存中。对 CART 来说这也是很正常的，因为除了数据库文献之外，很少有发布的算法为大规模数据集的数据管理提供任何明确的指导。对于有些算法，添加一项数据管理技术是很容易的，可以以模块化的方式完成。例如，如果每个数据点只需被访问一次，而且顺序是无关紧要的，那么数据管理就很微不足道了（只需顺序的将数据点子集读入主存）。

然而对于树算法，模型、评分函数和搜索方法的复杂程度足以使数据管理很不简单。为
150 了说明为何如此，回忆树算法是以数据驱动的方式不断地将各个观察结果（数据矩阵的行）分割成小的数据集，这要求我们不断地在数据库中找到观察结果的不同子集并且确定这些子集的性质。在这种算法的朴素实现中，对于超过主存容量的数据集，会导致多次的重复扫描第二存储器中的数据（例如磁盘），从而导致非常差的时间性能。最近已经开发出了树算法的可伸缩版本，这个版本使用特殊的数据结构来高效地处理主存储器外的数据。

下面总结一下我们关于 CART 的介绍，我们注意到这种算法是由以下几部分构成的：（1）树模型结构；（2）交叉验证评分函数；（3）对树结构的两个阶段贪婪搜索（“增长”和“修剪”）。由此看出，一旦掌握了 CART 的核心思想，这个算法还是很容易理解的。显然，我们可以使用相同的树结构、交叉验证评分函数和搜索技术开发出与 CART 本质相似的其他算法，只是在实现细节上是针对具体应用的（例如如何处理训练和预测过程中的残缺数据）。对于给定的数据挖掘应用，按这种方式定制算法是很有必要的。总而言之，像 CART 这样的算法的强大之处在于它所体现的基本思想，而不是实现的具体细节。

5.3 数据挖掘算法的化约主义观点

根据本章的基本口诀，一旦我们有了数据集和明确的数据挖掘任务，那么数据挖掘算法就可以认为是由{模型结构、评分函数、搜索方法、数据管理技术}构成的“组合（tuple）”。尽管这看上去很简单，但是其中有相当丰富的内涵。首先，我们能构造的算法数量是相当大的。只需把不同的模型结构和不同的评分函数、搜索方法、数据管理技术组合起来，我们就能生成相当大数量的不同算法（专业的研发人员也注意到了这一点）。

然而，一旦我们意识到下面的第二层内涵，那么便可以很容易地处理“算法空间”的复杂度了：尽管有大量的可能算法，但是在这个组合中每个组件的基本“值”数量是相对较小的。特别地，我们可以使用定义完备的模型和模式来解决像回归、分类和聚类这样的问题； 151
我们在第 6 章中将详细的介绍这些模型。类似地，正如我们将要在第 7 章看到的，具有广泛吸引力的评分函数（例如，似然、误差平方和、错误分类率）并不多。具有广泛适用性的搜索和优化方法也是较少的，而且数据管理的关键原理可以被简化为数量相当少的几种不同技术（分别在第 8 章和第 12 章讨论）。

许多著名的数据挖掘算法都是由定义完备的组件构成的。换句话说，各种算法趋向于比较紧凑地聚集在“算法空间”（由模型结构、评分函数、搜索方法及数据管理技术这些“维”所组成的空间）中。

在实践中，数据挖掘算法的化约主义（reductionist）（也就是，基于组件的）观点是非常有用的。它通过把算法分解为一些核心组件阐明了特定算法潜在的机制。这也使得比较不同的算法变得很容易，因为我们能够从组件层清楚地看出相似点和不同点（例如，我们可以根据它们所使用的评分函数来比较 CART 算法和 C4.5 算法）。

更重要的是，这个观点强调了算法的基本性质，而不是通常的从算法列表角度来考虑。当面对一个数据挖掘应用时，一个数据挖掘者应当考虑的是哪些组件更适合他的要求，而不应考虑选取哪个现成的算法。在理想的情况下，数据挖掘者应拥有一个软件环境，针对他们的特定应用编选组件（从模型结构、评分函数、搜索方法等的库里选取）以合成算法。不幸的是，这还只是一种理想的情况，而不是实际的规范。目前的数据分析软件包经常仅提供一个算法的列表，而不是一个基于用来合成算法的组件的工具箱。考虑到为没有技术背景和时间从组件层次理解算法潜在细节的数据挖掘者提供可用的工具，这也是可以理解的。然而对于希望定制和合成面向问题的算法的熟练的操作人员来讲这就不够理想了。“菜谱”方法（cookbook method）也是多少有些危险的，因为数据挖掘工具的初级使用者可能不能够完全理解他们所使用的黑盒子算法的限制（及潜在的假定）。与此相反，基于组件的描述使得 152
黑盒子里的东西更为清晰。

为了描述化约主义观点的一般应用，在下面三小节里我们将从组件的角度分析三个著名的算法。在第 9 章到第 14 章里我们将更加详细的阐述这些算法和与之有关的算法，在那里我们针对不同的数据挖掘任务讨论了更完整的解决方案。

5.3.1 用于回归和分类的多层感知器

在一般的人工神经网络模型中，前馈多层感知器（MLP）是应用最广泛的模型。MLP 结构提供了从实数的输入向量 **x** 到实数的输出向量 **y** 的非线性映射。因此，MLP 可以用作

回归问题的非线性模型，也可以通过对输出数据作出恰当的解释来用于分类。 MLP 的基本思想是，一个 p 个输入值的向量被乘以一个 $p\times d_1$ 的权矩阵，得到的 d_1 个值中的每一个分别经过一个非线性变换得到 d_1 个“隐藏节点”的输出。然后把得到的 d_1 个值再乘以一个 $d_1\times d_2$ 的加权矩阵（另一“层”权），然后把得到的 d_2 个值中的每一个再经一层非线性变换。变换后得到的 d_2 个值可以作为模型的输出结果，也可以再做一层加权乘法及非线性变换，
153 等等（所以称这种模型是“多层”的；术语“感知器”指的是 20 世纪 60 年代提出这种形式时的最初模型，它包括一个加权层和一个非线性阈值）。

作为例子，考虑图 5-5 中有一个“隐藏”层的简单网络模型。经第一层加权（α 和 β）计算两个内积 $s_1=\sum_{i=1}^{4}\alpha_i x_i$ 和 $s_2=\sum_{i=1}^{4}\beta_i x_i$，然后对每一个在隐藏节点上经一次非线性变换后产生两个标量值：h_1 和 h_2。在隐藏节点，广泛使用的是非线性逻辑函数，也就是 $h_1=h(s_1)=1/(1+e^{-s_1})$。接下来，$h_1$ 和 h_2 被加权并组合得到结果 $y=\sum_{i=1}^{2}w_i h_i$（原则上我们还可以对 y 做一次非线性变换）。因此，y 是一个关于输入向量 $\mathbf{x}$ 的非线性函数。可以把 h 看作是四维输入的非线性变换，两个“基函数” h_1 和 h_2 的集合。对于这个模型来说，要从数据中估计的参数有输入层上的八个权值 $(\alpha_1,\cdots,\alpha_4,\beta_1,\cdots,\beta_4)$ 和输出层上的两个权值（w_1 和 w_2）。一般地，如果有 p 个输入，一个有 h 个隐藏节点的隐藏层，一个输出，那么总共就有 $(p+1)h$ 个参数需要从数据中估计。通常，我们可以做多层这样的加权乘法和非线性变换，但通常我们只用一个隐藏层，因为多层网络的训练速度比较慢。MLP 的权是模型的参数，必须由数据来确定。

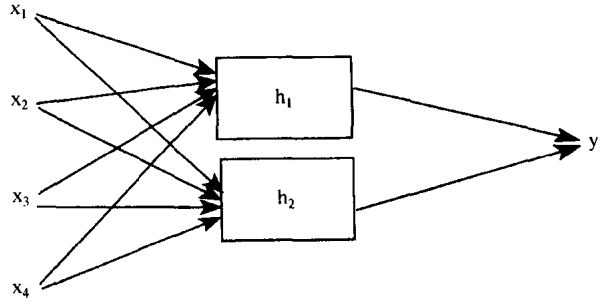


图5-5 一个简单的多层感知器（或者说神经网络）模型。这个网络有两个隐藏节点 $(d_1=2)$ 和唯一的输出节点 $(d_2=1)$

注意到，如果输出结果 $\mathbf{y}$ 是一个标量 y（即 $d_2=1$）并且介于 0 到 1 之间 （我们可以从上一层的加权值里选择一个非线性变换来保证这个条件），对于二分类的问题，可以使用 y 做类隶属关系的指示变量，并用阈值（举例来说）0.5 来决定是属于第 1 类还是第 2 类。因此可以很方便地使用 MLP 来解决分类和回归问题。由于模型的非线性性质，由网络模型生成的不同类之间的决策边界也可能是高度非线性的。图 5-6 给出了这样的一个决策边界的例子。注意到，与图 5-3 中分类树生成的边界相比，这里的边界是高度非线性的，然而，和图 5-2 中的决策树不同，不存在一种简单的总结形式来描述神经网络模型的运作方式。

根据化约主义观点，可以把 MLP 学习算法归纳为以下“算法组件”的组合：

1. 任务 = 预测：分类或回归
154 2. 结构 = 输入数据加权和的多层非线性变换

3. 评分函数 = 误差平方和
4. 搜索方法 = 从随机选取的初始参数值开始的最陡峭下降
5. 数据管理技术 = 在线或分批处理

这个算法最显著的特点是模型结构的多层非线性性（注意到，不但输出结果 y 是关于输入数据的非线性函数，而且参数 θ（权）在评分函数里也是非线性的）。这使神经网络明显不同于传统的线性和多项式形式的回归方法以及基于树模型的分类方法。

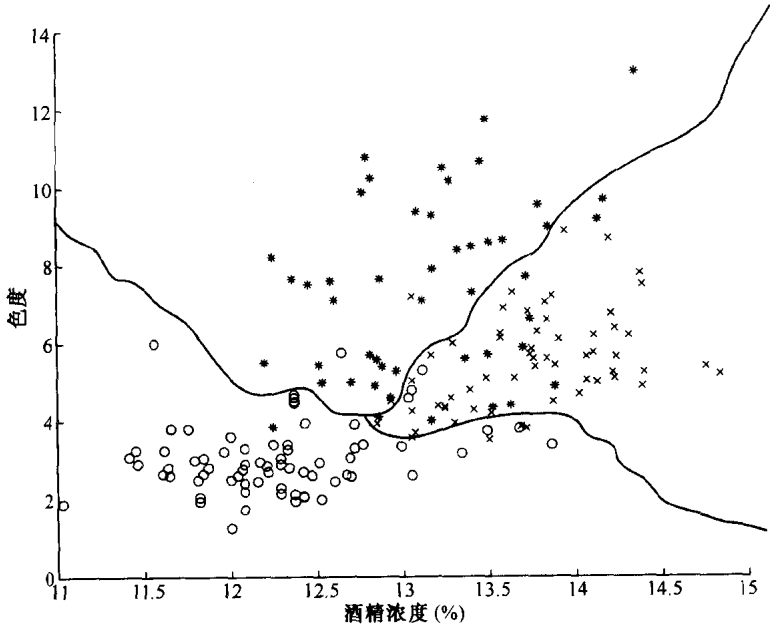


图5-6 神经网络模型产生的决策边界例子。对应的数据为图5-1[⊖]中有关酒的二维数据

MLP 中使用最广泛的评分函数是误差平方和（SSE），其定义如下：

$$S_{SSE} = \sum_{i}^{n} \big(y(i) - \hat{y}(i)\big)^2 \qquad (5.2)$$ 155

其中 $y(i)$ 和 $\hat{y}(i)$ 分别是第 i 个数据点真实的目标值和网络的输出结果，其中 $\hat{y}(i)$ 是关于输入向量 $\mathbf{x}(i)$ 和 MLP 的参数（权）θ 的函数。有人认为误差平方和是唯一可用于神经网络模型的评分函数。事实上，只要是关于模型参数的可微函数（使我们可以确定最陡峭下降的方向），那么任何评分函数都可以用做最陡峭下降搜索方法（例如反向传播）的基础。举例来说，如果我们把误差平方和看作是更一般的对数似然函数（参见第 4 章的讨论）的一种特例，那么我们就可以根据具体应用使用大量的其他基于似然的评分函数来代替误差平方。

训练神经网络就是把 S_{SSE} 看作是未知参数 θ 的函数，使其最小化（也就是通过给定数据来估计参数 θ）。如果每个 $\hat{y}(i)$ 通常是关于参数 θ 的高度非线性函数，那么评分函数 S_{SSE} 也是关于 θ 的高度非线性函数。所以，对于 MLP 来说不存在可以使 S_{SSE} 最小化的参数 θ 的闭合形式解。此外，因为在 S_{SSE} 关于 θ 的函数曲面上存在很多局部最小值，所以很多情况下

⊖ 译注：原书此处为 5-2（a），疑有误。

训练神经网络（即，对特定的数据集和模型结构求出使 S_{SSE} 最小化的参数）都是一种非平凡（non-trivial）的多元优化问题。要找到满意的局部最小值需要迭代的局部搜索技术。

最初为 MLP 提出的训练方法被称为反向传播，这是一种相当简单的优化方法。它实质上是在参数空间里对评分函数（误差平方和）进行最陡峭下降，也就是通过从参数空间中随机选取的起始点下降到一个局部最小值来解决这种非线性优化问题。（实际应用中，我们通常从多个起始点下降，并从找到的所有最小值中选择最好的。）在更一般的框架下，有很大一族优化方法可以求解这样的非线性优化问题。很多时候，最陡峭下降被认为是训练 MLP 的唯一可用优化方法，但事实上像共轭梯度这样更强大的非线性优化技术也可以用来解决这一问题。第 8 章中我们将讨论这些技术中的一部分。

从数据管理角度说，可以使用在线方式（基于一次一个数据点的循环更新权）也可以使用批处理方式（观察所有数据点后再更新权）来训练神经网络。这种算法的在线更新版本是
156 在线估计算法的一种特例（第 8 章中进一步讨论了使用这种算法时涉及的折衷问题）。

MLP 与分类树的一个重要的实际差别是树算法（例如 CART）以相对自动的方式搜索不同复杂度的模型（例如，找到一个大小合适的树是 CART 算法的基本功能）。相反，尚没有一种被广泛接受的方法可以用来确定 MLP 的合适结构（也就是决定模型中应有多少层及多少个隐藏节点）。现在有许多算法可以自动构造网络，包括从一个小网络开始，逐步增加节点和权的方法，以及从一个大网络开始逐步修剪权和不相关的节点。逐步增加能承受局部最小值问题的网络结构（具有 k 个隐藏节点的最好网络可能与具有 $k-1$ 个隐藏节点的最好网络在参数空间里差异非常大）。另一方面，训练一个过大的网络的开销可能是非常惊人的，特别是当模型结构很大时（例如，输入维度 p 很大）。实际应用中，网络结构通常是由反复的“试验-误差”过程确定的，即手工调整隐藏节点的数目直到在验证数据集（一组没有用于训练的数据点）上达到令人满意的性能。

对 MLP 的基于组件分析说明它的一般步骤并没有与更传统的统计估计和优化技术相差太远。许多这样的技术（例如，将贝叶斯先验结合到评分函数使很小的权变为 0（使模型“规格化”）；或者在搜索权期间使用更复杂的像共轭梯度技术这样的多元优化方法）都可以用于训练网络模型。20 世纪 80 年代，当神经网络刚刚面世的时候，还不清楚它与统计学的联系（尽管现在回顾起来那是非常显然的）。毫无疑问，神经模型方法的主要贡献在于模型结构所具有的非线性多层特征。

5.3.2　关联规则学习的 A Priori 算法

关联规则是数据挖掘中用来表示局部模式的最流行方法之一。第 13 章给出了更详细的介绍，这里只勾画一下它的一般思想并从构成它的组件的角度简要描述一种通用的关联规则
157 算法。（这里的介绍大体上是基于著名的 A Priori 算法——寻找关联规则的最早算法之一。）

关联规则是对数据库中的某些特定事件一起发生的概率的简单陈述，它尤其适用于稀疏的数据集。为简单起见，假设所有变量都是二值的。关联规则具有如下的形式：

$$\text{如果 } A=1 \text{ 并且 } B=1 \text{ 那么 } C=1 \text{ 的概率为 } p \tag{5.3}$$

其中 A，B，C 是二值变量，并且 $p=p(C=1 \mid B=1,\ A=1)$，即给定 $A=1$ 及 $B=1$ 时 $C=1$ 的条件概率。条件概率 p 有时被称为规则的“精度”或“置信度”，$p(C=1,\ B=1,\ A=1)$ 被称为“支持度”。这种模式结构或者说规则结构是非常简单而且是可以解释的，这是这种

方法具有广泛吸引力的原因。寻找关联规则的典型目标是寻找满足以下约束的所有规则：置信度 p 大于某个阈值 p_a，支持度大于某个阈值 p_s (例如，找到满足支持度大于 0.05，置信度大于 0.8 的所有规则)。这些规则由一种相对较弱的知识形式构成；它们就是对观察数据中一起出现的模式（co-occurrence patterns）的简单归纳，而不是能刻画出整个总体的强结论。事实上，“规则”一词通常暗示了一种因果关系（从左到右），因此严格地说“关联规则”这个术语用词并不十分合适，因为这些模式是内在的相互关联而不一定是因果关系。

挖掘关联规则的一般思想最早起源于涉及“购物篮数据”的应用。这些数据通常被记录在数据库中，其中的每个观察就是一篮商品（例如各种生活用品），每个变量用以表明是否购买某一种商品。我们可以把这种类型的数据想像为一个 n 行（对应于各个购物篮）p 列（对应于各种商品）的数据矩阵。这样的矩阵可能很大，n 是百万级的，p 是上万的，而且一般是很稀疏的，因为典型的一个购物篮只包含几种商品。关联规则的作用是提供了一种手段，以一种相对有效的计算方式在这样的数据中找到简单的模式。

按照我们的化约框架，寻找关联规则的典型数据挖掘算法包含以下组件：

1. 任务 = 描述变量之间的关联关系
2. 结构 = 用概率表示的“关联规则”（模式） 158
3. 评分函数 = 精确度和支持度的阈值
4. 搜索方法 = 系统搜索（带修剪的广度优先）
5. 数据管理技术 = 多重线性扫描

在关联规则搜索中使用的评分函数是简单的二择一函数。有两个阈值：p_s 是规则支持度的下限（例如，当我们仅想要得到至少覆盖 10%的数据的规则时便令 $p_s = 0.1$）而 p_a 是规则的置信度的下限（例如，当我们仅想要得到精度不低于 90%的规则时便令 $p_a = 0.9$)。如果一个模式满足了两个阈值条件，那么它的得分为 1；否则得分为 0。我们的目标是要找到所有得分为 1 的规则（模式)。

所有可能关联规则的数量是指数级的——对于二进制变量如果我们限制规则的左侧和右侧都为肯定的命题（例如，A=1)，那么也就是 $O(p2^{p-1})$，因此这个搜索问题是比较复杂的。尽管如此，利用评分函数性质的优势，我们还是可以将算法的平均运行时间降低到一个可控的范围。注意到如果只要 $p(A = 1) \leqslant p_s$ 和 $p(B = 1) \leqslant p_s$ 二者中有一个成立，那么显然 $p(A = 1, B = 1) \leqslant p_s$。我们可以将这个规律应用到关联规则搜索中，首先找到概率大于阈值 p_s 的所有单个事件（例如 $A = 1$）（这只需对整个数据库做一次线性扫描)。如果一个事件（或一组事件）的概率大于支持度阈值 p_s，那么就称其为“频繁的”。我们把这些频繁事件的所有可能对（pairs）作为容量为 2 的候选频繁集。

在更一般的情况下，也就是当从容量为 $k - 1$ 的频繁集生成容量为 k 的频繁集时，我们可以剪除任何容量为 k 的集合，只要它包含一个 $k - 1$ 项的子集，而且这个子集本身在 $k - 1$ 级是不频繁的。例如，如果我们只有频繁项集$\{A = 1, B = 1\}$及$\{B = 1, C = 1\}$，那么可以将它们组合为容量为 3 的频繁项集$\{A = 1, B = 1, C = 1\}$。如果$\{A = 1, C = 1\}$这个子集是不频繁的（即这一项不在容量 $k = 2$ 的频繁集列表中)，那么$\{A = 1, B = 1, C = 1\}$也是不频繁的，因此完全可以把它剪除。注意到这种修剪可以在不直接搜索数据的情况下进行，对于大数据集来说，这可以大大提高计算速度。

确定了修剪后的容量为 k 的候选频繁集列表后，算法对数据库再执行一次线性扫描以确

159 定哪些集合确实是频繁的。然后把确认后的容量为 k 的频繁集（如果存在）进行组合，以生成所有可能的含有 $k+1$ 个事件的频繁集，随后再进行修剪，然后再对数据库进行扫描，等等——直到再也无法生成频繁集。（在最坏的情况下，所有可能的事件集合都是频繁的，因而算法执行需要指数级的时间。然而，因为实际中这些算法要处理的交易数据集中的数据都是非常稀疏的，最大的频繁项集的容量通常是非常小的（相对于 n 来说），至少相对于比较大的支持度来说是这样的。）然后算法用已经找到的所有频繁集对数据集再做最后一次线性扫描。这决定了频繁集的哪些子集组合表达为规则后也满足置信度阈值，然后返回对应的关联规则。

关联规则算法中有很多有趣的数据挖掘算法，在这些算法中搜索和数据管理组件是它们的最关键部分。特别地，关联规则使用广度优先、一般到特殊的系统搜索方法，以尽可能使对数据库的线性扫描次数达到最小。尽管在机器学习文献中还有很多其他的规则发现算法（具有类似的基于规则表示），但是关联规则算法是特别为大规模数据集设计的一种相当高效的算法。举例来说，关于关联规则算法的研究报告往往强调的是计算效率，而不是对算法产生规则的解释。

5.3.3 检索文本的向量空间算法

可以把一般的根据内容检索任务大体描述为：对于一个查询对象及一个庞大的对象数据库，我们要在数据库中找到与查询对象最相似的 k 个对象。我们对在线文本搜索中的这种问题都很熟悉。例如，我们的查询可能是一个很小的关键词集合，“数据库”对应于非常庞大的网页集合。这时我们的任务是要找到与关键词最相关的网页。

第 14 章将进一步讨论这个检索任务。这里我们只从组件的角度探讨一下一般的文本检索算法。这个任务的最重要问题之一就是如何定义相似性。文本文档的长度和结构都是不一样的。我们怎样才能比较如此变化各异的文档呢？文本检索的一个关键思想是将所有文档简
160 化为如下所述的统一向量表示。令 t_1，…，t_p 为 p 个项（单词，短语等等），我们可以把它们看作变量，或数据矩阵中的列。并把文档（数据矩阵中的一行）表示为分量数为 p 的向量，其中第 i 个分量包含了 t_i 项出现在文档中的次数。就像购物篮数据一样，实际应用中，我们会得到非常庞大的数据矩阵（n 为百万级，p 为万级），但却非常稀疏（大多数文档向量都有非常多的零）。当然，我们不会真的存储一个这么大的 $n \times p$ 矩阵：一个更为有效的办法是为每个 t_i 项建一个文档链表，列表中是所有包含 t_i 的文档。

确定了这样的“向量空间”表示，接下来就可以很容易地定义相似性了。一种简单的定义就是把相似距离定义为 p 维空间中两个向量之间的夹角。这种角度衡量了“项空间”中给定方向上的相似性，而且排除了大文档中出现一个词的概率要大于小文档所导致的差异。向量空间表示和相似性的角度尺度似乎比较粗糙，但在实践中这样的方法效果非常好，而且在文本检索中基于这种基本模式的变体非常多。

有了如上信息，我们就可以定义简单文本检索算法的各个组件了，假设算法的目标是寻找与一篇文档最相似的 k 篇文档：

1. 任务 = 在数据库中检索 k 篇与给定查询最相似的文档
2. 表示法 = 项出现向量
3. 评分函数 = 两个向量之间的夹角
4. 搜索方法 = 多种技术
5. 数据管理技术 = 多种快速索引策略

对于上面给出的组件定义，还有许多不同的定义方法。例如，在定义评分函数时，我们可以定义比角度函数更具一般性的相似尺度。在指定搜索方法时，可以使用很多不同的启发式搜索技术。注意在这种背景下的搜索是一种实时搜索（real-time search），因为算法不得不为用户实时地检索模式（与前面讨论的数据挖掘算法不同，那些搜索是离线搜索最优的参数和模型结构）。

不同的应用可能需要在检索算法中使用不同的组件。例如，在搜索法律文档时，缺少某些特定项是值得注意的，因此我们可能希望在评分函数定义中反映出这一点。换一种情况，我们可能希望相反的效果，即不太重视两个文档中不含有特定项的事实（更重视两篇文档中都出现的项）。 161

显而易见，模型表示是这里的关键思想。一旦使用的向量表示已经建立起来，那么就可以在向量空间中定义各种各样的相似尺度了，然后使用标准的搜索和索引技术在稀疏的 p 维空间中搜索相邻对象。不同检索算法的评分函数和搜索方法在细节上会有所不同，但大多都是基于同样的数据向量表示。如果要为文档定义一种不同的表示（比如基于某种语法形式为数据定义产生式（generative）模型），那么就不得不使用完全不同的评分函数和搜索方法了。

5.4 讨论

无论对于新手还是经验丰富的研究人员，漫步在数据挖掘算法的丛林里总是多少有些困惑的。我们希望本章中介绍的基于组件观点能为读者提供一个评估算法的有效的工具。过程如下：首先尽可能拿掉只有研究报告和产品说明才需要的行话和行销套话，然后把算法精简为它的基本组件。基于组件的描述为比较算法奠定了定义完备的“标准”框架结构——我们可以把新算法与其他著名的算法进行比较，如果它们是不同的，那么可以从组件的角度清晰的看出它们的差异。

有趣的是不同的研究团体关注的数据挖掘算法的侧重点是不同的。大多数统计学刊物都力图展示出大量的公式用来确定模型、评分函数和计算方法，很少有关于如何将模型更好地应用到实践中的详细算法说明。相反，有关机器学习和模式识别的计算机刊物经常强调计算方法和算法，很少强调模型的结构和与之配套的评分函数是否合适。举例来说，对各种算法 162
所做的试验性比较并不少见，但比较内在模型或评分函数的却很少。对于数据挖掘来说，以上两个研究领域的不同侧重点导致在数据挖掘领域中形成了两种完全不同的方法论（而且经常是相反的）。统计方法经常非常强调推理过程的理论性（例如，参数估计和模型选择），很少突出计算问题。面向数据挖掘的计算机科学方法往往相反，更注重高效的搜索和数据管理，不太关心模型（或模式）结构是否合适或评分函数是否贴切。在阅读本书的整个过程中，很有必要留意这种“思想方法”的不同，这有助于理解在这两个研究团体内推动特定模型、推理方法和算法发展的因素。

无论是统计学还是计算机科学，可以说，具有代表性的一些研究论文对特定算法内在组件的说明都不是非常清晰。文献里充满了各种不同算法的奇特名字和缩写。在许多论文里，有关模型结构、评分函数和搜索方法的描述完全纠缠在一起。

在实践中，一个数据挖掘算法的所有组件都是至关重要的。模型、评分函数和计算实现等方面的相对重要性会随着问题的不同而有所不同。对于小的数据集来说，模型的解释和预测能力可能（相对地说）要比计算因素重要的多。然而，随着数据集的增大（无论是测量数

量还是变量数量），那么计算的作用就变得越来越重要了。例如一个时间复杂度为 $O(n^2)$的聚类算法在 $n = 100$ 时是容易驾驭的，但当 $n = 10^8$ 时就完全驾驭不了了（很可能在我们终生也解决不了这个问题！）。进一步说，时间复杂度通常都是按所有数据都驻留在内存的假定来表述的。如果算法的每一个计算步骤不是从主存中读取数据而是必须从磁盘读取的话（打个比方），那么在时间复杂度表达式中就不得不考虑额外成倍的固定时间开销。

对于非常庞大的数据集，必须在建模的完善度和计算开销（例如所用的时间）之间进行折衷以达到某种拟合质量。对于海量数据集，计算方法直接影响到哪一类型模型结构能拟合
163 数据。计算问题在数据挖掘中所起的作用要比在传统的统计建模中大的多。

当然，在任何数据挖掘问题中，模型结构和评分函数的选择都要慎重并且确认清楚。如果返回的模型没有用的话，那么能高效的处理再大的数据集也是没有用的。因此，数据挖掘者必须仔细平衡建立精良的模型或模式结构与找到并稳定的拟合这样的结构所需的计算资源这两方面，寻求最佳的折衷方案。

5.5 补充读物

很少有论文把对数据挖掘算法的分析提升到基于组件的系统高度。Buntine, Fischer and Pressburger（1999）是个例外，他们就如何从高层算法的角度实现数据挖掘算法的快速自动原型给出了一些有趣的讨论（含例子）。关于一般算法的经典教材有 Cormen, Leiserson and Rivest（1990）及 Knuth（1997）。

最早提出 CART 原理的是 Breiman et al.（1984），Quinlan（1993）详细的描述了 C4.5 算法。Buntine（1992）以及 Chipman, George and McCulloch（1998）探讨了对 CART 的贝叶斯扩展。Crawford（1989）介绍了以增量方式构造分类树的方法，Gehrke et al.（1999）介绍了针对大规模数据集的可伸缩树构造算法的有关概念。Ballard（1997）是一篇非常易懂的入门级教材，书中讨论了很多现代神经网络算法和这些算法与真正脑模型之间的关系。Geman, Bienenstock and Doursat（1992）非常精彩的讨论了统计思想和神经网络学习算法之间的关系。Ripley（1996）从统计学的观点全面地浏览了各种神经网络算法（第 5 章）和树学习算法（第 7 章），而 Bishop（1995）是一本完全从统计角度讨论神经网络学习算法的教材。

Agrawal et al.（1996）评论了各种关联规则算法，并深入的分析了这些算法的搜索方法和效率。Salton and McGill（1983）对信息检索作了很有价值的介绍；Witten, Moffatt and Bell
164 （1999）详细全面地讨论了用于大规模文本和图像数据库的检索算法所涉及的各种问题。

第6章　模型和模式

6.1　概述

在前面的章节里已经介绍了模型和模式之间的区别。本章将更深入地探讨这些概念，并考察数据挖掘中使用的几种主要类型的模型和模式，以便为后续章节中的详细分析做准备。

模型是对一个数据集的高层次、全局性的描述。它通过一个很大的样本透视总体。模型可以是描述性的——以方便简洁的方式归纳数据；也可以是推理性的，允许对数据所在的数据总体或者可能的未来数据作出某些论断。本章将讨论几种形式的模型，例如线性回归模型、混合型模型以及马尔可夫模型等。

而模式则是数据的局部特征，或许只支持几条记录或者几个变量（或二者兼有）。一个 p 维变量空间的局部“结构”特征，比如密度分布函数的最频值（mode）（或区间（gap））或者回归曲线上的拐点就是模式的例子。很多情况下模式是很有趣的，因为它描绘了与数据一般行为的背离，例如相关度特别高的一对变量、某些变量值异常高的一系列项、对于某些变量总是具有相同值的一组记录，等等。同模型的情况一样，寻找模式的目的是为了描述或者推理。诸如要找出数据库中那些具有异常特征的数据，或者要预测可能具有异常特征的未来记录。模式的具体例子如脑电图（EEG）曲线上的瞬时波形、零售客户经常购买的产品的 165
异常组合以及半导体生产数据数据库中的孤立点。

数据压缩（data compression）很好地说明了模式与模型的概念。设想一个数据发送器 T 有一图像 I 要发送到接受器 R（尽管这里以图像为例，但是其中的原理对于不是图像的数据集合也成立）。有两种主要的策略：（a）传送描述图像 I 所有像素的数据；（b）传送图像的压缩版本——图像 I 的某种概括。数据挖掘在很大程度上对应于第二种途径，实现压缩的方法要么是把原始数据表示为一个模型，要么是通过模式标识出数据的异常特征。

在建模中，当概括数据时很可能导致某种数据失真——这意味着数据接受器将无法准确地重建这些数据。下面考虑一个对图像数据建模的例子——用原始图像上每个 16×16 像素方块中所有像素值的平均数来代替这个方块。这种情况下的模型就是一组更小的、分辨率更低（1/16）的图像。一种更复杂的模型是把图像分割成一个个大小、形状不同的局部图像。这些区域内像素的像素值可以用区域内的一个像素强度常量来相当准确的描述。这种情况下的模型（或信息）就是每个图像区域中的常量值和每个区域边缘的描述。很显然，对于每一种模型（平均像素模型和局部常量模型），都可以把图像模型的复杂度（被平均的像素数目，局部不变区域的平均大小）当作被传送的信息量（同样也可视为传送过程中丢失的信息量，即压缩率）。

从模式探测的角度来看，图像的模式是图像中的一些结构，它们纯粹是局部性的：例如，图像左上角的一个部分模糊的圆形物体。很明显相对于上述全局性的压缩模型来说，这是完全不同的压缩方式。接受器不再重建整个图像的数据概括，但它的确要对图像的某些局部进行描述。视问题和目标情况的不同，局部结构有时比全局模型更适合。数据发送器 T 并不传送对大量嘈杂的像素值的概括模型描述，而是把接受器的注意力集中到一些重要的特征。这让我们想起了第 5 章讲述的关联规则：它们努力把注意力集中到变量子集间潜在的有趣关联上。 166

图像编码与数据分析之间的比喻似乎不够完美（比如说，正如我们所描述的，数据压缩并没有考虑泛化到未见过数据的概念），尽管如此，这个比喻让我们领会了高分辨率的局部结构与低分辨率的全局结构之间的关键折衷。

本章是这样组织的：6.2 节讨论模型的一些基本特征和在建模过程中所要做的必要选择。6.3 节集中探讨一类重要模型背后的一般原理，在这类模型中，把一个变量从其他众多变量中独立出来作为“响应”变量。这类模型包括回归和有指导的分类模型。许多数据挖掘问题涉及大量的变量，这会带来一些问题，6.5 节将讨论这些问题。6.4 节探讨描述性模型。许多数据集中包含按照一定图式（如时间序列或图像数据）收集的数据，在建模过程中通常需要对这些数据进行特殊考虑。6.6 节讨论与这些结构化的数据相关的问题。最后，在 6.7 节中论述了针对多元序列化数据的模式。

6.2 建模基础

模型是对现实世界中过程的抽象描述。例如，$Y = 3X + 2$ 是一个非常简单的模型，描述了变量 Y 如何与变量 X 相关联。可以把这个特定模型视为更一般的模型结构 $Y = aX + c$ 的一个实例，其中对于这个特定模型我们设定了 $a = 3$ 和 $c = 2$。更一般的情况是 $Y = aX + c + e$，e 是一个随机变量，用来代表从 X 到 Y 的映射（随后将讨论）的随机部分。正如第 4 章中所描述的，我们通常把 a、c 叫做该模型的参数，而且经常用符号 θ 来表示一般参数或者一系列参数（或者是向量）。在此实例中，$\theta = \{a, c\}$。给定模型的形式或者结构，接下来我们的任务就是通过估计为其选择合适的参数值——也就是说选择一个合适的评分函数来衡量模型与数据之间的拟合情况，然后通过最小化或最大化该函数来选择合适的参数值。第 4 章中已经介绍了这个过程，后面的章节将进一步讨论。

然而，在估计模型的参数之前，我们必须首先为模型本身选择一个合适的函数形式。这
167 一节的目的是在较高的层面上概述数据挖掘中所使用的主要模型类型。

数据挖掘中的建模是由数据驱动的（data-driven）。它通常不是由任何潜在机制或“事实”驱动的，它就是为了捕捉数据中存在的关系。即使在存在一种被公认为正确的数据产生机制的情况下，我们也应该记住这一点，正如 George Box 所说的，“所有的模型都是错误的，不过有些是有用的”。例如，尽管我们可能假设存在一个线性模型来解释数据，但是往往是个幻想，因为即使在最好的情况下，仍然会有很小的非线性作用，这是模型所不能捕捉到的。我们要寻找的是能够概括数据产生过程主要特征的模型。

因为数据挖掘是数据驱动的，所以不应该认为模型的发现存在某种因果关系。例如，顾客记录分析表明，购买高品质白酒的人更有可能购买出于设计师之手的服装。很显然，一种倾向并不与另一倾向（在两个方向上）具有必然的因果联系。而是它们都更可能是购买者具有相对较高收入的结果。然而，白酒和衣服这两个变量彼此都与对方没有因果关系的事实并不意味着它们对预测目的是没有用处的。从市场的角度来说，由观察到的购买白酒的行为预言可能购买服装的行为（如果以前已经在数据中发现了这两者的关系）是完全合理的。然而，既然没有建立任何因果关系，那么下面的结论“操纵（manipulating）一个变量将导致另一个变量的变化”就不正确。也就是说，即使在数据中存在这种关系，但是诱导顾客去购买高品质白酒不见得会使他们同样去购买出于设计师之手的服装。

6.3 用于预测的模型结构

在预测模型中，一个变量被表达成其他变量的函数。这样便可以从给定的其他变量（称为解释变量或预报变量）的值预测响应变量的值。通常用 Y 表示预测模型的响应变量，用 X_1，…，X_p 表示 p 个预报变量。这样我们便可以建立一个预测模型，根据申请表和数据库中包含的客户既往行为预测借贷者拖欠贷款的概率。可以把第 i 个以前客户的记录方便地表示为 $\{(\mathbf{x}(i), y(i))\}$，这里 $y(i)$表示第 i 个客户的结果（好或坏），而 $\mathbf{x}(i)$是第 i 个客户申请表中各 168
个值的向量 $\mathbf{x}=(x_1(i)$，…，$x_p(i))$。这个模型将通过下式进行预测：$\hat{y}=f(x_1$，…，$x_p;\theta)$，其中 $\hat{y}$ 是该模型的预测，θ 代表该模型结构的参数。如果 Y 是数量值变量，那么从 p 维向量 $\mathbf{X}$ 到 Y 的映射叫做回归（regression）。如果 Y 是范畴型变量，那么估计从向量 $\mathbf{X}$ 到 Y 的映射叫做分类学习（classification learning）或有指导的分类（supervised classification）。从都是在学习一种从 p 维变量 X 到 Y 的映射这个意义上来说，这两种任务都可以被看作函数近似（function approximation）问题。为了便于说明，本章主要集中讨论回归任务，因为很多一般原理可以直接推广到分类任务中。第 10 章、第 11 章将分别详细讨论有指导分类和回归。

6.3.1 具有线性结构的回归模型

我们从简单的线性预测模型开始讨论，在这种模型中响应变量是预报变量的线性函数，即：

$$\hat{Y}=a_0+\sum_{j=1}^{p}a_jX_j \tag{6.1}$$

这里 $\theta=\{a_0$，…，$a_p\}$。需要再次重申的是这个模型是纯试验性的（empirical），因此存在高匹配性和高预报性的模型并不意味着就存在某种因果关系。在上述表达式的左边用 $\hat{Y}$ 而不是简单地用 Y，是因为它是个模型，是在数据的基础上构建的。也就是说 $\hat{Y}$ 的值是从 $\mathbf{X}$ 预报而来，而不是实际观测到的值。在第 11 章中将详细讨论这一区别。

从几何意义上讲，这个模型描述了一个嵌在 $p+1$ 维空间的 p 维超平面，a_j 决定它的斜率，a_0 为其截距。参数估计的目的就是选取 a 值来确定这个超平面的位置和角度，以便与数据 $\{\mathbf{x}(i), y(i)\}$，$i=1$，…，n 最佳拟合，拟合的质量是由 y 的实际观察值和模型预测值 $\hat{y}$ 之间的差异来衡量的。

这种具有线性结构的模型在数据分析的历史上占有很特殊的地位，一部分是因为用合适 169
的评分函数来评估它的参数非常简单直接，一部分是因为该模型的结构简单、容易解释。例如，模型的可加性意味着任一个预报变量的改变都不会影响其他变量所对应的参数。当然有些情况下贡献独立（individual contribution）是没有什么意义的。特别是，如果两个变量高度相关，那么探讨改变其中一个变量而另一个变量的贡献保持不变就没有意义了。在后续章节中将更详细地讨论这个问题。

我们可以在保持模型可加性特征的前提下，在模型中包含预报变量的非线性函数。也就是：

$$\hat{Y}=a_0+\sum_{j=1}^{p}a_jf_j(X_j) \tag{6.2}$$

其中，函数 f_j 是 X_j 的平滑（但可能是非线性的）函数。f_j 可以是对数、平方根或者原始变量

X 的有关变换。该模型仍然假定了依赖变量 Y 以可加的形式依赖于模型中的独立变量（X）。在实践中，这仍可能是一个很强的假定，但是据此产生的模型可以很容易的解释每一个变量个体 X 的贡献。模型的这种简洁性也意味着要从数据中估算的参数数量相对较少（p+1），这使得参数估计问题更加简单直接。

我们还可以进一步推广这种线性模型结构，使其可以包含具有交差乘积项的一般多项式，以允许模型中各个变量 X_j 之间的相互作用。一维的情况是很熟悉的——我们可以将其想像为用 2 次、3 次或者 k 次多项式来内插表示观测到的 y 值。多维的情况将此推广到（p+1）维空间里定义在 p 个变量上的光滑曲面。

需要指出的是，尽管这些预测模型关于变量 X 是非线性的，但是它们对于参数却是线性的。我们将在第 11 章中看到，相对于参数以非线性形式介入的情况而言，这使得参数评估变得容易得多。

例 6.1 在图 6-1（a）中，我们显示了从等式 $y = 0.001x^3 - 0.05x^2 + x^3 + e$ 模拟出的 50 个数据点，$x \in [1, 50]$，其中 e 是附加的高斯（均值为 0，标准偏差 $\sigma = 3$）噪声。图 6-1（b）显示了对数据的线性拟合，图 6-1（c）显示了对数据的二次多项式拟合。尽管线性拟合抓住了 Y 作为 X 函数的总的向上趋势（在这个特定区间），但是显然二次拟合效果更好。正如我们可以从每一个模型的误差结构中（每个模型的误差都具有作为 x 函数的系统结构）所看到的，两种拟合都没有完全捕捉到真实结构的内在弯曲。两种拟合都是由最小化误差平方和评分函数来确定的。

170

注意到通过在模型中包含更高次项和 **X** 各分量之间的相互作用项，原则上我们能够估算出比简单线性模型所对应的超平面更复杂的曲面。然而，应该注意到随着 p（输入空间的维度）的增加，模型中可能的相互作用项（如 X_jX_k）的数量会按照关于 p 的组合函数增加。因为可加模型中每一项都具有一个权系数（参数），所以随着 p 的增加整个模型（包括所有 p 个变量间的所有可能 k 阶相互作用项）中要估计的参数数量会迅速上升。对这类模型的解释和理解也随 p 的增加而难度更大。实践中的一种替代办法是选择可能相互作用项整个集合的一个很小子集加入到模型中。然而，如果以数据驱动的方式（数据挖掘应用中最具代表性的方式）实施选择的话，那么所有可能的相互作用项（搜索空间的大小）将达到 2^p，随着空间维度 p 的增加搜索问题的难度将按指数规律增加。本章稍后将回过头来讨论如何处理维度的问题。

将线性模型推广到多项式模型带来了一个重要的问题，即模型的复杂度。较复杂模型包含了较简单模型作为其特例（即所谓的嵌套）。例如，一次模型 $a_1X_1 + a_0$ 可以视为二次多项式模型 $a_2X_1^2 + a_1X_1 + a_0$ 当 a_2 为 0 时的特殊情况。因此，不难得出复杂模型（X 变量的高阶多项式）拟合观察到数据的效果总是至少和较简单模型的一样好（因为它包含了相对简单的模型作为特例）。这样便产生了一个复杂的问题，当不同模型的复杂度（或表达能力）不一样时，应该如何选择这一个模型而不选那一个。这是一个棘手的问题：我们可能需要一个最接近某个猜想的未知“事实”的模型；也可能需要寻找一个能够抓住数据的主要特征又不太复杂的模型；也可能需要寻找一个能够对未见过数据做出最好预测的模型；如此等等。后续章节会返回到这个问题上。现在我们回过头把讨论的焦点集中到模型自身的表达能力上，而不考虑针对已知观察数据如何在这些模型中做出选择的问题。

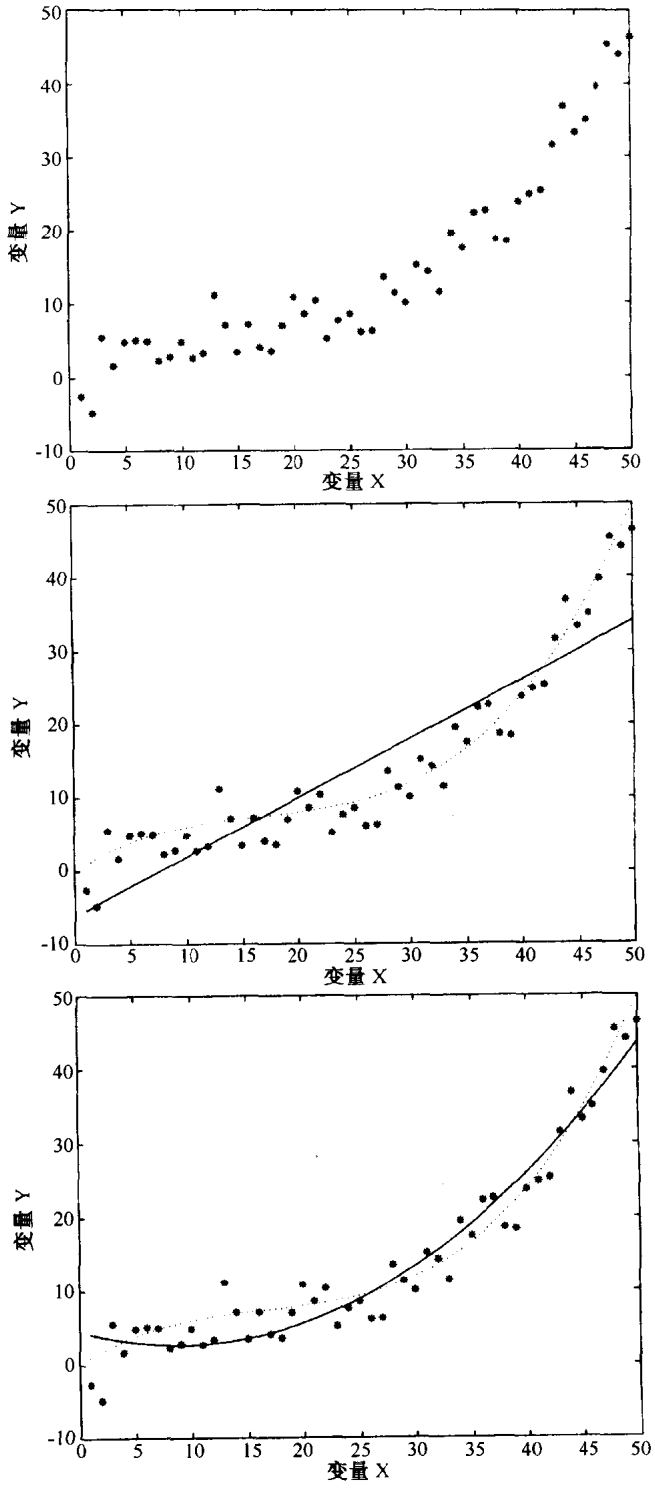


图6-1 模型拟合示例。(a) 用附加了高斯（正态）噪声的三次多项式模拟出的50个数据点；(b) 模型$aX + b$（实线）的拟合情况，(c) 模型$aX^2 + bX + c$（实线）的拟合情况。(b) 和 (c) 中的虚线表示产生数据点的真正模型（参见正文）。每种情况下的模型参数都是用最小化模型预测值和观测值之间的误差平方和来估计的

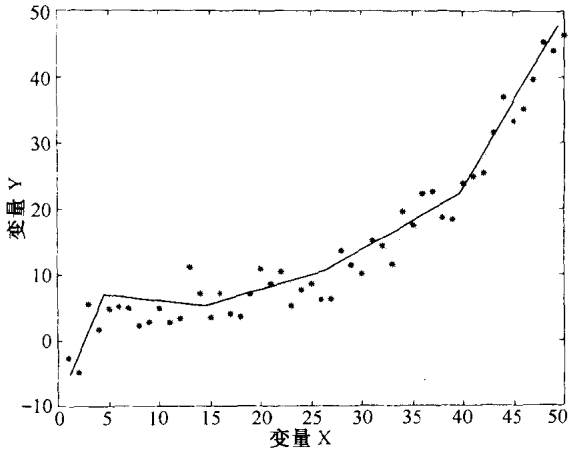


图6-2 对图6-1中数据进行分段线性拟合的例子。图中使用了5个线性片段（$k=5$）

171
~
172

变换预报变量是推广线性结构的一种方式。另一种方式是变换响应变量。*sqrt(Y)*可能很好地与变量 **X** 的线性组合相关，因而与其直接匹配 Y 不如先求它的平方根，然后用变量 **X** 的线性组合去预报 *sqrt(Y)*。当然我们事先并不知到平方根是合适的变换形式。必须通过试验尝试多种变换形式，同时正如第 2 章中所讨论的，要注意问题中所涉及的测量方法特性隐含的约束。正因为此我们才说数据挖掘是一种令人兴奋的发现之旅，而不仅仅是以标准方式应用标准方法的简单运用。

正如第 11 章中所述，我们可以把简单线性回归模型想像为是在预报变量 X 的每个值处预测分布 Y 的期望值，即 $E[Y \mid X]$。也就是说，回归模型预报了 Y 的条件分布的参数，这里参数就是均值。当然，更一般的情况是，从变量 **X** 的线性组合可以预测 Y 的其他条件分布参数。这就是第 11 章中将要探讨的推广的线性模型（generalized linear model）和神经网络。

我们看到，尽管线性模型简单而又易于解释（而且我们也看到，它们的参数很容易估算），但是可以很方便地对它们进行推广，得到强大而又灵活的模型。任何一看到“线性”一词就
173 以为是一种死板模型的想法都是错误的。

6.3.2 用于回归的局部分段模型结构

还有一种进一步推广基本线性模型的方法，那就是假定 Y 是 **X** 的局部线性函数——在 **X** 空间的不同区域具有不同的局部依赖性，这便是分段线性模型。从几何学角度来说，该结构包括一系列 p 维超平面，每一个平面覆盖输入（**X**）空间的一个区域，这个区域与其他区域不重叠。这种模型结构的参数既包括每个超平面的局部参数，又包括各个超平面的位置（边界）。对于 X 为一维的情况，其原理是很容易描绘的：由 k 个不同的直线段逼近一条曲线（图 6-2 中显示了这样的一个例子）。注意在这幅图中线段是彼此相连的，因而线是连续的。也可以定义一种松散的模型结构，线段末端不需要连接。有时这是很有用的一种模型形式，但有时不连续也会导致问题出现，因为这意味着预报变量的极小变化将导致响应变量的值产生突然跳跃。例如，如果两个片段是在收入变量值为$50,000 处断裂的，那么对于两个除了一个收入是$50,001、另一个收入是$49,999 外都一样的申请者，我们得到的对响应变量 $\hat{y}$（借贷方的拖欠概率）的预测值可能差异很大。如果认为不连续性不可取的话，那么可以强制使每条线段在末端具有不同阶的导数连续性（显然不再是直线）。这样的曲线片段被称为样条（spline），对

应的整个模型叫做样条函数（spline function）。通常，每一条线段对应于一个低次（二次或三次）多项式。这样做得到的结果是一条光滑的曲线，但是可能多次改变方向——所以这种模型具有高度的灵活性。

可以把这种思想推广到预报变量数多于一个的情况。同样，各个局部片段（现在不再是线段，而是（超）曲面）可以但不是必须在边界处相连接。第 10 章描述的有指导分类树结构就是这种模型的例子。

分段线性模型是通过把简单的部分（这里是超平面）分段结合在一起构建出相对复杂模型（适用于非线性情况）的很好例子。这是数据挖掘中的经常使用的一种模式——由相对简单的局部组件构建出复杂的全局结构——这种思想同时又是建模和模式探测之间的桥梁。也就是说，局部性也提供了把复杂模型分解成简单局部模式的框架。例如，$\hat{Y}$ 关于 X 函数曲线 174
上的一个“波峰”可以用两条彼此相连的适当斜率的斜线反映出来。

本小节和上一小节介绍了如何由简单模型构建复杂模型。其途径要么是把简单模型合并成复杂模型，要么是通过不同方法泛化简单模型到复杂模型。数据挖掘中使用的模型，没有一个是绝对孤立的，而是通过各种各样的关系相互联系在一起，每个模型要么是其他模型的泛化、要么就是其他模型的特例或者变体。在数据挖掘中，建立一个有效模型的关键之处是选出能够最好地解决所面临问题的一种模型形式。这不是一种简单的操作：选一种模型形式；应用它；然后便给出结论。相反，我们需要拟合模型，根据结果修正或者拓展模型，然后反复重复上述操作。数据挖掘，总的来说像数据分析一样，是一种反反复复的过程。

6.3.3 “基于记忆”的非参数局部模型

在前面一小节中我们给出了一些例子，说明了基于数据的局部特征的模型是如何与广泛的全局模型相联系的，实际是前者包含在后者中。本小节要进一步介绍局部建模的思想。（回想一下模式，它尽管也是局部的，但却是孤立结构，不是数据的全局概括的一部分。因此我们可以说局部建模技术和模式探测是完全不同的。）

粗略地讲，上面简要描述的样条和树模型是用从数据点附近估计出的函数来替代这些数据点。另一种相对的策略是保留这些数据点，推迟对 Y 的预测值的估计过程，直到确实需要估计的时候。即数据不再被函数和它的参数来代替。例如，要估计响应变量 Y 在新情况下的值时，可以取数据集中极相似的对象所对应的 Y 值的平均值，这里的“极相似”是根据预报变量定义的。

可以把这种思想扩展到包含数据集中的所有对象，但必须根据它们与新对象的相似程度对它们进行加权——不相似者权值小，相似者权值大。权值决定了它们的 Y 值对最终估计的贡献。*局部加权回归*（也就是 loess[⊖]*回归*）模型就是这种估计方法的例子。

虽然我们是在预测建模的背景下讨论局部平滑思想的，但该思想也适用于描述建模和密 175
度估计的情况——事实上后者是最早推广这种思想的领域。实际上，在第 3 章我们已经看到了这种方法在显示单变量图形中的应用，当时使用这种思想来估计概率密度函数。在后面的章节中我们将看到更多有关这方面的例子。在这一背景下，第 3 章中介绍的所谓核（kernel）估计量是非常常见的。

对于这种估计量，一个明显的问题是如何确定权函数的形式。随相似性降低缓慢衰减的

⊖ 译注：loess 是 local regression 的缩写。

权函数将产生平滑的估计，而快速衰减的函数将产生锯齿形估计。因此必须找到一种最适合分析目的的折衷。

可把权函数分解成两部分，一部分是它精确的函数形式，另一部分是它的“带宽”。假设 $K\left(\frac{x-z}{h}\right)$ 是一个平滑函数，用来确定数据集中的点 x 对一个新点 z 的估计的贡献。该贡献的大小依赖于 K 的形式，同时也依赖于带宽 h，带宽 h 越大评分函数的平滑性越好，带宽 h 越小评分函数越粗糙（多锯齿）。实践证明带宽比权函数的精确形式更重要。

例 6.2　图 6-3 显示了由三角核函数构建的回归函数例子，图中使用了三个不同的带宽。这里我们的目的是估计发动机尾气中氮氧化物（NOx）比例，相对乙醇（E）的函数。我们使用的数据是在不同条件下对 81 台汽车发动机的测量结果。图中最大的带宽（$h = 0.5$）显然太宽了，产生的评估过于平滑，“遗漏”了中间的波峰和两端的信息；最窄的带宽（$h = 0.02$）给出了一种很“尖刻”的评估，看起来追随了观测数据中的噪声；取值介于二者中间的带宽（$h = 0.1$）比较合理，既保留了 NOx 和 E 之间的关系又不过度拟合。对于简单的一维问题，主观的目视观察方法是选择带宽的一种实用方法，但不适用于多维问题。也可以采用许多自动化方法以数据驱动的方式来选择 h 值，比如“交差验证法”。

核函数法与最近邻法有着非常密切的关系。事实上，这两类方法都在不断地扩展延伸，以至于某些情况下它们已经完全相同了。然而核函数法是按照核函数和带宽来定义平滑度，而最近邻法则按最近邻的数量来定义平滑度，让数据来决定带宽。例如基本的单一最近邻分
176 类器（这里 Y 是分类变量）把数据集中最相似对象所属的分类赋给新对象；而 k 最近邻分类器把数据集中 k 个最相似对象中最普遍的分类赋给新对象。更复杂的最近邻方法根据到被分类点的距离确定对估计的贡献加权，而更复杂的核函数方法让带宽依赖于数据——因此从模型结构上来说这两种方法几乎是相同的。

像核函数模型这样的局模型部结构经常被描述成非参数的，因为这种模型在很大程度上说是数据驱动的，没有传统意义上的参数（带宽 h 除外）。这种数据驱动平滑技术（例如核函数模型）对于解释数据是很有用的，至少在一维和二维的情况下是如此。

很清楚，局部模型有其吸引人的地方。然而，没有任何一个模型可以解决所有问题，局部模型也有不足之处。尤其是随着预测空间中变量数量的增加，要获得准确估计所需的数据点数量在呈指数上升（“维度效应”的结果，见下面的 6.5 节）。这意味着这些“局部近邻”模型对于高维问题的伸缩性往往很差。

从数据挖掘的观点来看，还有一个缺点就是这种模型缺乏可解释性。在低维度（$p \leqslant 3$）的情况时，可以画出曲线进行估计，但在高维度时是不可能的，而且没有直接的方法来概括模型。事实上，把这些表示称为模型是在扩展模型的定义，因为它们从来没有被定义成显式的函数，只是通过数据作了隐含的定义。

6.3.4　模型结构的随机部分

直到这一节，除了有几处简要提及以外，我们一直忽略了这样一个事实：对于真实的数据，通常很难找到存在于预报变量 $\mathbf{X}$ 和响应变量 Y 之间的完美函数关系。换言之，对于预报变量的任何一个给定向量 $\mathbf{x}$，能够观察到的 Y 值不只一个。对于 $\mathbf{X}$ 的每一个值的 y 值分布中体现了一

种偏差（variation），如果仅使用关于变量 **X** 的模型，那么这种偏差是不会随模型的复杂性而降低的。正因为如此，有时把它叫做偏差的不可解释部分、非系统部分或者叫随机部分，而把可以根据变量 **X** 解释的 Y 的偏差叫做可解释偏差或系统偏差。（当然，这仅是因为系统偏差原则上可以被变量**X**解释，并不意味着一定可以构建出能够做到这一点的模型）。

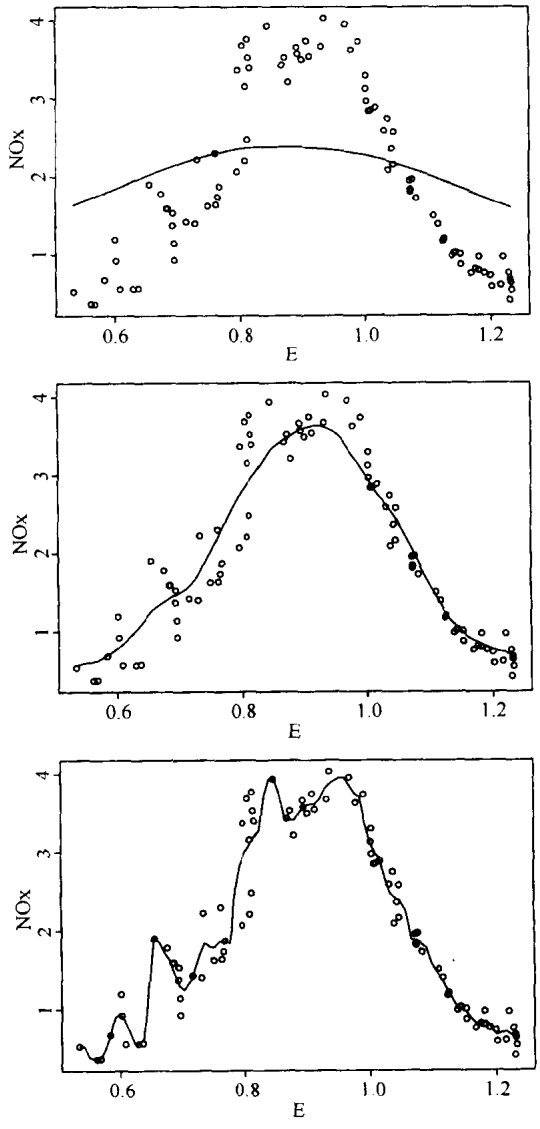


图6-3 利用三角核函数回归估计氮氧化物（NOx）关于乙醇（E）的函数。从上至下对应的带宽分别为：$h=0.5$（上），$h=0.1$（中）和$h=0.02$（下）

在前面的大多数讨论中，我们一直把注意力集中在模型的系统部分，但是我们也需要考虑模型的随机部分。模型的随机部分可能来源于许多方面。可能是由简单的测量误差导致的——正如第 2 章所讨论的，重复测量 Y 将得到不同的结果。也可能是由于变量 **X** 的集合中没有包括完美预测 Y 所需的所有变量所导致的（例如，如果仅仅依赖顾客过去的购买行为来预测他是否要购买特定产品，那么就忽略了可能相关的人口统计学信息，比如年龄、收入

等）。事实上这种情况通常是应该在预料内的：仅用数量有限的变量就可以完美地解释另一个变量的所有变化细节的情况是极少见的。

例 6.3 可以把前面讨论的回归建模框架进行扩展，使之包含一个随机部分。假定对于每一个 **x**，我们将观察到一个特定的 y，但带有一定的附加噪声；也就是说，在 **x** 和 y 关系中存在某种固有的不确定性：

$$y = g(\mathbf{x};\theta) + e \tag{6.3}$$

这里 $g(\mathrm{x};\theta)$是输入 x 的确定性函数，而 e 通常被定义为方差（σ^2）恒定而且均值为零的随机变量（独立于 x）。随机项 e 反映了测量过程的噪声（也就是说，我们并没有观测到 y 的“真实”值，而是得到了一个带有噪声的 y 的观测值）。更一般地讲，随机部分 e 反映了存在着隐藏变量的事实（这些变量没有被测量到，或者对于观测来讲是隐藏的），隐藏变量对 y 的影响方式是无法用 Y 对变量 **X** 的依赖性来表达的。

对 e 的零均值假设并无害处，因为如果噪声是个非零均值，那么就可以把它吸收到 g 里去而不失一般性。例如，作一个常见的假定，假定 e 服从均值为零并且具有恒定方差σ^2的正态分布，那么：

177 ~ 179

$$y|\mathrm{x} \sim N(\mu_{y|\mathrm{x}}, \sigma^2), \qquad \mu_{y|\mathrm{x}} = E[y|\mathrm{x}] = g(\mathrm{x};\theta) \tag{6.4}$$

在实践中，需要慎重考虑恒定方差σ^2这一假定：例如，如果 Y 代表年度信用卡消费，X 代表收入，那么有可能 Y 的变化性会按 **X** 的函数上升。如果是这样的话，那么为了在模型中包含这一特征，上述模型中的σ就必须为 x 的函数。

注意在这些公式中函数 g 的形式是自由的，也就是说，可以选择前面所讨论的任何一种模型结构。我们在第 4 章中已经看到，上面对 e 的正态假设很自然地让我们想起最小平方法回归——也就是通过最小化观测到的 y 值和 $f(\mathrm{x};\theta)$之间的误差平方和来确定 g 的参数θ。

在选择合适的评分函数来估计参数时或在选择模型时，随机部分是很重要的。似然评分函数（第 4 章介绍的，其他地方也讨论过）就是基于对随机部分的分布形式的假定的。扩展的似然函数包括一个平滑性惩罚项以便不拟合过于复杂的模型，该函数也需要对随机部分的分布情况作出假定。基于似然概念的更高级方法（例如，所谓的准似然方法（quasi-likelihood method））放宽了分布假定的细节，但选择参数估计时仍然是以随机部分的分布情况为基础的。

6.3.5 用于分类的预测模型

到目前为止，我们集中讨论的预测模型的被预测变量 Y 都是数量型的。现在，我们简要考虑一下 Y 是范畴型变量的情况，也就是说 Y 的取值范围是几个可能的范畴性值。这就是（有指导的）分类问题，其目的是根据一个新对象的观测到的 **X** 值，将其分配到一个正确的类别中（也就是正确的 Y 范畴）。

在分类建模中，我们实质上感兴趣的是不同类别之间的边界。像回归的情况一样，我们可以对边界的函数形式作一个简单的参数假定。例如，一种分类途径是在 p 维 **X** 空间里采用线性超平面来定义两个类别之间的决策边界。也就是说，模型用线性边界把 **X** 空间分割成不相交的决策区域（每个部分对应一个类别）（参见图 6-4）。更复杂的模型允许包含更高次的多

项式项，这就产生了多项式决策边界。如果允许非常灵活的非线性形式作为边界的话，那么 180
我们就得到了像第 5 章所讨论的神经网络分类器那样的模型。

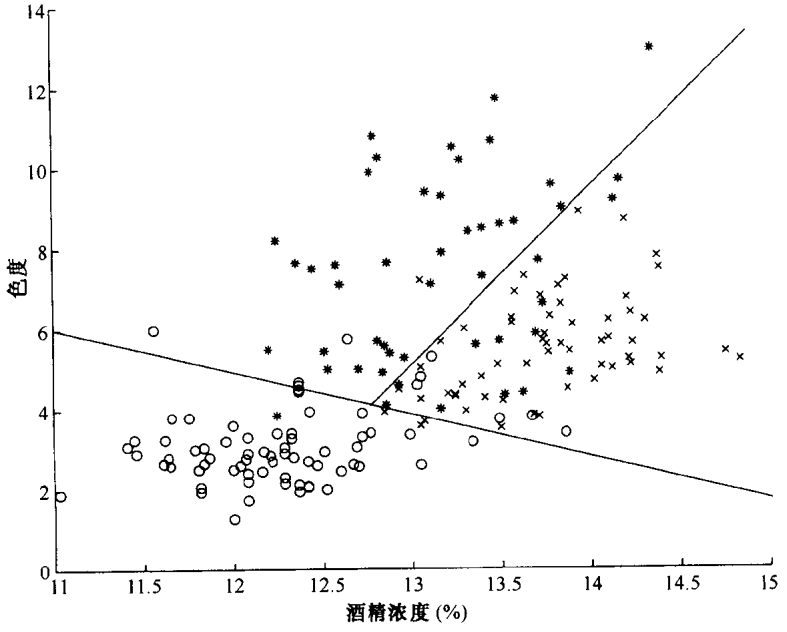


图6-4 线性决策边界的例子，对应的数据集为第5章中的二维酒分类数据集合（参见图5-1）

就像回归建模的情况一样，另一种提高灵活性的方法是组合多种简单局部模型，比如像图 6-5 所示的那样把线性决策边界分段组合起来。举例来说，第 5 章中的分类树模型定义了一类特殊的局部线性决策边界，其边界是分层的并且是与坐标轴平行的。正如前面所提到的，最近邻分类器是用训练数据集中与新的未知类数据点最近点的标签作为预测。虽然这种技术本质上通常被视为一种方法而不是模型，但事实上它确实隐含定义了一种分段线性边界（至少是在使用欧式距离定义近邻时是这样的）。

还有数量相当大的不同分类技术，它们提供了不同的方式来模拟决策边界。像最近邻这 181
样的方法很灵活（对于每一个类别，允许有多个局部的彼此不相连的决策区域，区域具有灵活的边界），而给出单一全局超平面的模型要简单得多。

从实践建模的角度来看，关于分类边界形状的以前知识可能不如回归问题中关于 Y 如何与 $\mathbf{X}$ 相关联的知识那么容易获得。然而在判别模型中成功使用的函数形式与前面在回归建模中讨论的函数形式极为相似，因此二者间有很多同样的问题。我们在第 10 章中将更深入地讨论分类模型。 182

6.3.6 选择适当复杂度的模型

到现在为止我们已经介绍了许多模型结构，从相对简单的到复杂的。例如，在回归问题中“分段-局部”模型结构的复杂度是受局部区域的数目 k 控制的（假定每个区域的局部函数复杂度是固定的）。随着 k 的增大，所得到的曲线可以更紧密地“追随”观测数据。换言之，模型结构的表达能力增加了，因为它能够表示更复杂的函数。

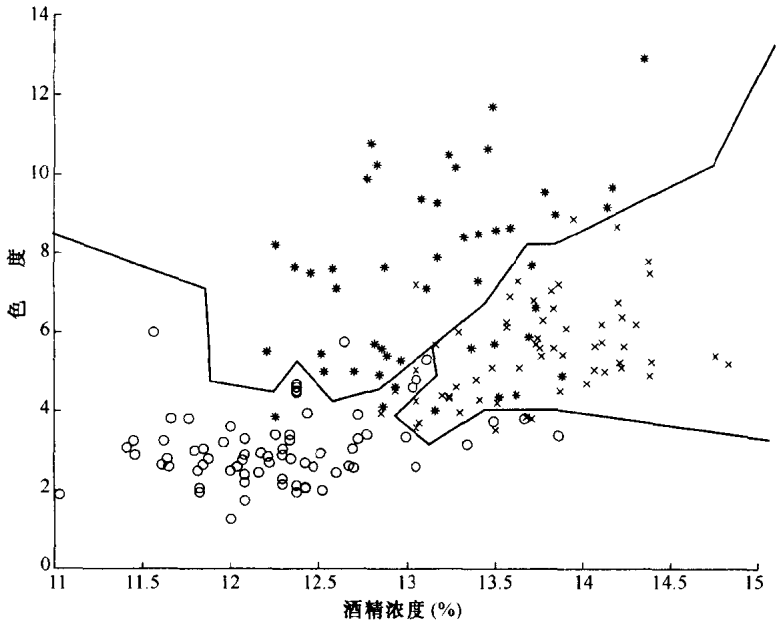


图6-5　分段线性决策边界的例子。对应的数据集为第5章中的二维酒分类数据集合（参见图5-1）

由于增加了模型的表达能力，那么很显然一般情况下会达到对现有数据的更好拟合效果。然而必须要小心。虽然评分函数在训练数据上的效果改善了，但是从泛化到新数据的角度来讲模型的性能实际上可能变得更差了。（回忆一下在第 5 章讨论分类树时介绍的“过度拟合”现象，尤其是图 5-4。）另一方面，如果我们走另一个极端，使模型结构过于简单化，那么就会因其太简单而失败。这种选择适当复杂度模型的问题始终是所有数据分析探索过程的一个关键问题。事实上，在第 7 章中我们将从理论角度对此进行分析，采用的方法是第 4 章中介绍的偏差-方差平衡思想的推广。

那么实践中，如何在复杂和简单之间选择一种合适的折衷呢？从数据驱动的角度(也就是数据挖掘的角度)来看，我们可以定义一种评分函数，它不仅可以考察模型对训练数据的拟合情况，而且可以估计模型对于新数据的性能。一种普遍使用的办法是把普通的拟合度项（对于训练数据）和一个明确惩罚模型复杂度的项组合起来。另一个广泛使用的方法是把训练数据分割成两个或更多的子集（就像第 5 章中描述的用于分类树的交差验证办法那样），然后在一个子集上训练模型，再使用不同的验证子集选择模型。

因为本章的重点是讨论不同模型和模式结构的表示能力，而不是讨论它们相对数据的效果如何，所以我们把对评分函数的详细讨论推迟到第 7 章。然而，对于那些一直想知道如何在已经讨论的不同模型中做出选择的读者来说，答案是确实存在一些定义完备的数据驱动评
183 分函数，这些评分函数允许我们搜索不同的模型结构以找到对给定任务看来最合适的模型（具有一定的限定，我们将在第 7 章中介绍）。

6.4　概率分布和密度函数模型

前面一节纵览了预测问题，在预测问题中特别选出一个变量（标识为 Y），然后使用其他变量对其作出预测。数据挖掘中的很多建模问题都属于这一类。然而还有许多建摸问题是

“描述性”的，目标是给出对数据的描述或总结。如果现有数据是完整的（如某一类化合物的全部），那么就不存在任何推理概念，目标就是简化描述。另一方面，如果现有数据是一个样本或者带有误差的测量值（因而如果再采集一次数据，那么可能会得到略微不同的值），那么建模的目的实质上是一种推理——推理出“真实”或者至少是比较好的模型结构。对后一种情况，可以把数据想像为是由一个潜在的概率函数产生的。

6.4.1 一般概念

在这一节中我们集中讨论几种用于密度估计的通用模型（第 9 章中会给出更详细的讨论）。因为潜在模型的函数形式往往会与我们在前面看到的（如单峰的“凸起”函数和用于回归的线性和多项式函数）多少有些不同，几个重要概念如简单模型的线性组合将再次得到广泛应用。

可以把通用的分布模型和密度模型分成两类：

1. **参数模型**：这种模型采用一种特定的函数形式。对于实数值变量经常使用位置参数（平均值）和范围（scale）参数（刻画变化性）来表征这种函数——例如正态分布和二项式分布函数。参数模型的优点在于简单明了（易于估计和解释），但是可能偏差相对较大，因为真实数据可能不遵循假定的函数形式。附录中简要介绍了一些更著名的参数密度和分布模型。 184

2. **非参数模型**：在这种模型中分布和密度估计是数据驱动的，事先仅对函数形式作很少的假定。例如可以使用第 3 章和 6.3.3 中介绍的核函数估计：可以把点 x 处的局部密度定义为 x 点附近各点的加权平均。

如果把上述两种情况视为极端情况，那么我们还可以定义一些界于参数模型和非参数模型之间的中间模型：混合模型（mixture model）。下面讨论该类模型。

6.4.2 混合模型

$\mathbf{x}$ 的混合密度是这样定义的：

$$p(\mathbf{x}) = \sum_{k=1}^{K} p_k(\mathbf{x}|\theta_k)\pi_k \tag{6.5}$$

该模型把 $\mathbf{x}$ 的整个密度（或者分布）分解为 K 个分量（component）或类（class）的加权线性组合。每个分量密度 $p_k(\mathbf{x}|\theta_k)$ 通常是由一种相对简单的参数模型（参数为 θ_k）（比如一个正态分布函数）组成的。π_k 代表一个随机抽取的数据点由第 k 个分量产生的概率，$\sum_k \pi_k = 1$。

为了说明混合模型的概念，考虑一个用作二维数据集模型的单一正态分布。可以把该分布想像为一种“对称凸形函数”，我们可以在二维空间确定其位置和形状，以尽可能好地模拟数据集（参见图 6-6，图中显示了一个简单的例子）。混合模型的一种直观解释就是它允许在二维空间里使用 k 个这样的凸形函数（或者分量），以逼近真实的密度。k 个凸形函数的位置和形状能够彼此独立地确定。而且我们可以给每一个分量赋一个权值。如果所有的权值是正的而且总和为 1，那么整个函数仍然是个概率密度函数（见 6.5 式）。

随着 k 的增大，混合模型可以具有非常灵活的函数形式，因为局部图形函数可用来捕捉局部密度特征（这使我们联想到回归中的局部建模思想）。很明显，k 值控制着模型的复杂度：因为 k 值越大，得到的模型越灵活，但同时解释也越复杂、拟合也越困难。这又一次验 185

证了通常的偏差-方差折衷规律。当然，我们并不只局限于仅使用正态分量（尽管实践当中这经常是最流行的）。同样也可以使用指数和其他密度形式的混合。至于决定分量位置、形状和 k 值的具体细节将在第 9 章中介绍。这里很重要的一点是混合模型提供了由简单参数密度模型（全局的）到这些密度模型的加权和的自然推广，从而可以局部化的匹配 p 维空间的数据密度。

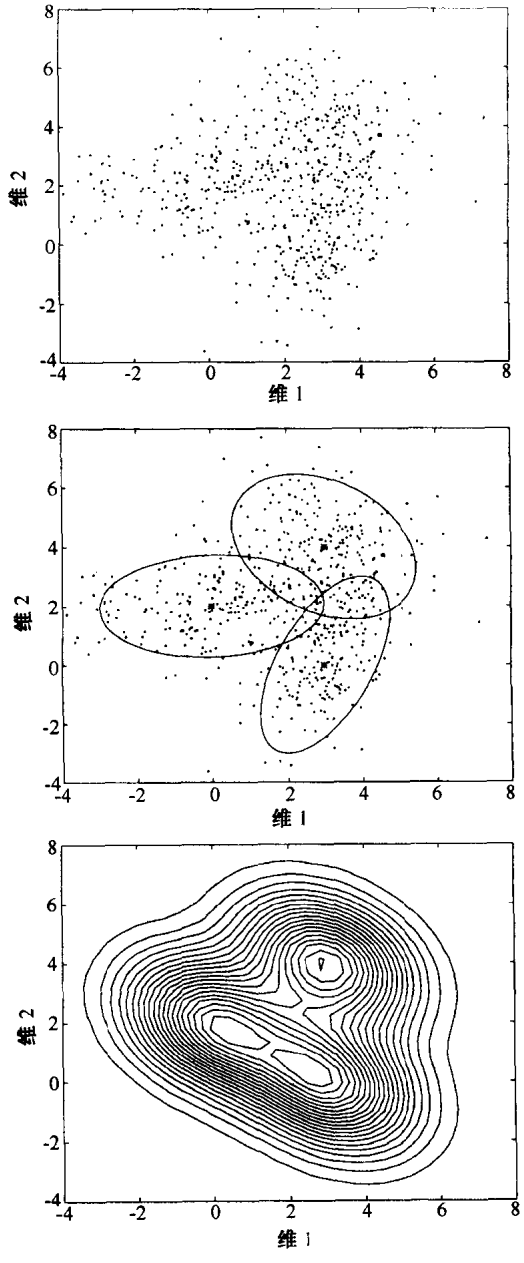


图6-6　混合模型示例。从上至下：（a）三个等权值的二元正态分布组成的混合模型产生的数据点；（b）画为距离平均值3σ处等高线的潜在分量密度；（c）总的混合密度函数产生的等高线

混合模型所蕴含的一般原理具有广泛的用途，这种思想被应用在概率建模的许多领域。

例如，使用混合模型能够很好的捕捉层次结构。第 8 章将讨论如何把混合模型拟合到数据的机理，在第 9 章中我们将看到混合模型是如何被成功地应用于探测数据中的聚类。

在模型解释方面，要么可以把混合模型就当作一个提供了灵活模型形式的“黑盒子”，要么可以给每个混合模型的分量一种明确的解释。例如与顾客数据相拟合的每个混合分量都可以被解释为刻画了不同类型的顾客。混合模型的一种解释（尤其是在聚类背景下）是：各个分量是由取 K 个值的隐含变量产生的，而且预先不知道各个分量在 p 维空间中的位置和形状，但可以由数据来揭示。因此，混合模型和投影追踪及相关的方法都共享了一种思想：即假设一种可能产生了观测到数据的简单的潜在或隐含结构。第 8 和第 10 章中我们将讨论如何利用“期望-最大化”（EM）算法从数据中学习混合模型的参数。 186 ~ 187

6.4.3 无序范畴型数据的联合分布

对于范畴型数据，我们可以得到一个按 p 个变量的所有可能值的交叉相乘定义的联合分布函数。例如，如果变量 A 取$\{a_1, a_2, a_3\}$，B 取$\{b_1, b_2\}$，那么 A，B 的联合分布就有六种可能的取值。这里假定（为了简便）数值确实是范畴型的而且大小和顺序是没有意义的。

当 p 的值和变量值数目很小的情况，以列联表（contingency table）单元格的形式显示出分布的各个值是很方便的，一个单元格显示一个联合值，就像表 6-1 中的例子所示。但随着变量数目的增大而且当值的数量大于 4 或 5 时这样做就不现实了。而且，这种列联表格并不能显示出数据中可能存在的潜在结构。例如表 6-1 中的数据已经经过了特意的构造，变量是彼此独立的，但这个事实并不能马上从表格中看出来。

表 6-1 二维范畴型数据的简单列联表，关于接受痴呆诊断的患者的数据集（经过了人为调整）

		痴 呆		
		无	中	严重
吸烟者	否	426	66	132
	是	284	44	88

与数值型变量的情况相比较，范畴型变量中的类别是无序的，因此不存在平滑概率函数的概念。因此如果所有的变量都具有 m 个可能值的话，那么为了完整地确定一个模型，我们就必须确定 $m^p - 1$ 个相对独立的概率值。很明显，随着 p 和 m 的增大，这将迅速变得难以实现。下一小节将探讨构建分布和密度函数的系统技术，以寻找一种经济的方式来描述高维数据。

6.4.4 因式分解和高维空间中的独立性

在分布和密度估计中空间的维度是一个根本性的难题。随着 x 维度——空间的增长，构建完全确定（fully specified）的模型结构的难度也迅速增大，因为模型结构的复杂度往往按照空间维度的指数增长（本章前面提到的维度效应）。

因式分解（factorization）把密度函数分成更加简单的组成部分，它提供了一种为多元数据构建简单模型的通用技术。这是一种简单有力的方法，它贯穿于整个多元数据建模过程中。例如，如果假定每个变量是相互独立的，那么我们就可以把联合密度函数写成：

188

$$p(\mathbf{x}) = p(x_1, \cdots, x_p) = \prod_{k=1}^{p} p_k(x_k) \tag{6.6}$$

这里 $\mathbf{x} = (x_1, \cdots, x_p)$，$p_k$ 是 X_k 的一维密度函数。显而易见，通常分别为一维密度建模比为它们的联合密度建模更容易。注意 log $p(\mathbf{x})$的独立模型具有可加的（additive）形式，这让我们想起了回归中的线性可加模型结构。

因式分解固然使事情简单了许多，但这是以建模的代价换来的。变量相互独立的假定在许多实际问题中甚至连近似正确都做不到。因此，完全的独立假定本质上是一个极端（最低难度），另外一个极端（最高难度）是完全确定的联合密度模型。当然我们不一定要刻意选择这个难度范围的极端情况；相反，我们可以选择介于二者之间的情况。联合概率函数 $p(\mathbf{x})$ 通常可以写成：

$$p(\mathbf{x}) = p_1(x_1) \prod_{k=2}^{p} p(x_k \mid x_1, \cdots, x_{k-1}) \tag{6.7}$$

公式的右边把联合函数分解成一系列条件分布。现在，我们可以试着给这些条件分布分别建模。很多时候可以进行相当大的简化，因为每一个变量 X_k 只依赖于它的几个前驱。也就是说，在第 k 个变量的条件分布中，经常可以忽略一些变量 X_1，…，X_{k-1}。这种因式分解可以用直观的图形来表示，每一个节点对应于一个变量，每一条边表示变量间的相互依赖关系。因此指向第 k 个变量的节点的边势必来自变量 x_1，…，x_{k-1}。很自然的，这些变量被称为变量 x_k[⊖]的双亲（parents）。

为了寻找这种简化的因式分解形式，有时我们必须通过把不同的模型拟合到数据来试验。在其他情况下，可以从数据结构中就明显地看出这种简化——例如变量代表的是对同一属性的一系列测量（比如在不同的时间）。这种情况下马尔可夫链模型往往是很合适的——在这种模型中，把对第 k 个变量有关的所有前面信息限制在与其紧邻的前一个变量上（从而公式 6.7 中的因式被简化为 $p(x_k \mid x_1, \cdots, x_{k-1}) = p(x_k \mid x_{k-1})$）。图 6-7 显示了一阶马尔可夫链模型的模型结构。

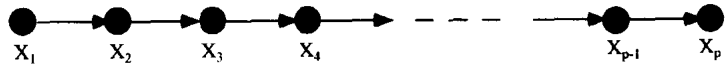


图6-7　对应于一阶马尔可夫假定的模型结构图形

189

用于描述概率模型的图形（比如图 6-7）经常被称为图形模型（graphical model）。在下面的讨论中，我们把焦点特别集中在无环有向图的一个被广泛使用的子集上（在计算机科学中作为概率模型使用时，有时被称为信念网络（belief network））。值得注意的是，这种图表示法强调了模型结构的独立性（例如，可以从图 6-7 中看出这一点），但是却没有指定父子关系的实际函数形式和数值参数。

下面再举一个图形模型例子，考虑以下三个变量：**年龄**、**教育程度**和**秃顶**（一个人是否秃顶）。很显然，**年龄**不可能依赖于其他两个变量中的任一个。相反，不论是**教育程度**还是**秃顶**都直接依赖于**年龄**。此外，在已知年龄的情况下，**教育程度**和**秃顶**情况彼此直接依赖是

⊖ 译注：原著此处为 x_1，当属印刷错误。

不合情理的——换言之，一旦知道了一个人的年龄，那么他是否秃顶并不表示他的受教育程度的高低（反之亦然）。另一方面，如果不知道一个人的年龄，那么**秃顶**往往可以提供一些**教育程度**的信息（例如秃顶的人很可能年龄较大，进而又很可能具有大学文凭）。图 6-8 显示了这样的一个图形模型。

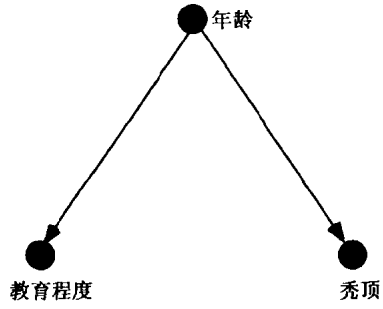


图6-8 一种可能的图形模型结构。变量受教育程度和秃顶情况在已知年龄的情况下是条件独立的

可以进一步拓展这种思想，假定存在观察不到的隐含或潜在变量，这些变量能够解释在数据中观察到的许多相互关系。图 6-9 给出了这样一个例子。在这个模型结构中引入了一个潜在的变量作为中间变量，这样简化了观测数据之间的关系（此处是医疗症状）和潜在的因果因素（此处是两种相互独立的疾病）。以这种方式引入隐含变量可以起到简化模型结构中关系的作用；比如如果给定了这个中间变量的值，那么症状就变成独立的了。然而在实践中，我们必须对应该向模型结构中引入多少个中间变量持慎重的态度，以避免把虚假的结构引入到拟合模型中。此外，正如我们将要在第 8 章、第 9 章中看到的，对于带有隐含变量的情况 190
参数估计和模型选择都是非常繁琐的。

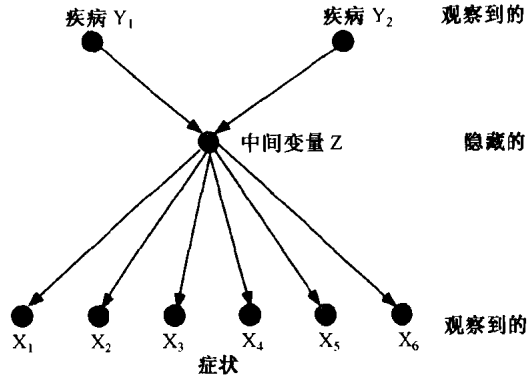


图6-9 关于一个疾病问题的图形模型结构。两种疾病是边缘（绝对）独立的，单一的中间变量Z直接依赖于两种疾病，给定Z的情况下六个症状变量是条件独立的

在分类和聚类中，假定对于分类变量的给定值其他变量互相条件独立会带来很多方便。也就是：

$$p(\mathbf{x} \mid y) = \prod_{j=1}^{p} p_j(x_j \mid y) \tag{6.8}$$

其中 y 是特定的（范畴性的）分类值。这就是 6.3.5 节分类建模中介绍的条件独立（“朴素”）

贝叶斯模型。图 6-10 中画出了这类模型的图形表示。

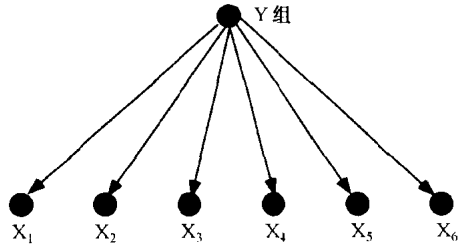


图6-10　一阶贝叶斯图形模型结构。图中画出了一个分类变量Y和六个条件独立的特征变量X_1，…，X_6

公式 6.8 也可以用于如下情况：Y 是一个为了简化模型 $p(x)$而引入的未观测到（隐含的，潜在的）变量，也就是我们可以得到一个如下形式的有限混合模型：

$$p(\mathbf{x})=\sum_{k=1}^{K}\Big(\prod_{j=1}^{p}p_j(x_j\mid y=k)\Big)p(y=k) \tag{6.9}$$

其中 Y 取 K 个值，并且是用公式 6.8 的条件独立假定对每一个分量 $p(\mathbf{x}|\ y\ =\ k)$建模的。举例来说，我们可以以这种方式对顾客如何购买 p 种产品的联合分布建模。按照这一模型如果一个顾客属于特定的分量 k，那么他购买特点产品子集的似然，也就是 $p_j(x_j|\ y\ =\ k)$，会随产品
191 子集 x_j 的增大而增大。这样尽管产品（x_j）是被按照在给定 $y=k$ 时条件独立建模的，但是混合模型单凭在特定分量 k 中某些产品会以高概率同时出现这一事实归纳出了绝对（边缘）独立。从效果上讲，隐含变量 Y 的作用是把变量 x_j 组织到均等的（equivalence）各个类别，在每一个类别中按照条件独立对变量建模。按这种方式使用隐含变量是一种很有力的建模技术，在第 9 章中我们将回到这个话题作更详细的讨论。

6.5　维度效应

在很多地方我们都注意到在一维情况下工作很好的模型并不能很好的被扩展到多维情况。特别是在参数或函数估计时，要保持一定的准确度，需要的数据量随维数的增大呈指数增长。有时这被称为“维度效应”。因为数据挖掘者经常对在高维度问题中寻找模型或模式很感兴趣，所以这个概念是很重要的。注意是否达到了“高维”的程度依赖于有关模型的复杂度和现有数据的数量，最少可能是 $p=10$ 个变量，最多可达 $p=1000$ 个变量或更多。

例 6.4　下面的例子摘自 Silverman（1986），它有力地说明了在高维情况下进行密度估计的难度。考虑由多元正态密度函数（具有单位协方差矩阵，均值为（0，0，…，0））模拟出的数据（参见附录多元正态密度函数的定义）。假定在核函数密度估计中，是通过最小化在均值处的平均误差平方来选取带宽 h。Silverman 计算了为了保证以下要求所需的数据点数：在 0 点的相对平均误差平方小于 0.1，也就是当 x=0 时 $E[(\hat{p}(\mathbf{x})-p(\mathbf{x}))^2]/\,p(\mathbf{x})^2<0.1$，其中 $p(\mathbf{x})$是真实的正态密度，$\hat{p}(\mathbf{x})$是用具有最佳带宽参数的正态核估计估计出的值。因此，我们所分析的是比较“简单”的估计问题：利用正态核估计估计最频值（平均来说此处的点最密集）处的正态密

度（相对精度在 10%以内）——还有比这更简单的吗？Silverman 指出数据点的数量呈指数增加。一维的情况需要 4 个点，二维需要 19 个点，三维需要 67 个点，六维需要 2790 个点，到了十维大约需要 842 000 个点。对于如此简单的问题，这样的数据点数量太惊人了！从这个例子中应该看到，随维度的增大，密度估计问题（也包括其他数据挖掘问题）的难度将急剧增大。

对付高维度问题有两种基本（而且相当明显）的策略。第一种是使用有关变量的子集来构建模型。也就是寻找一个 p'个变量的子集，这里 $p' << p$；第二种办法是把 p 个原始变量变换为 p'个变量，这里同样 $p' << p$。这种途径的例子包括主分量分析方法、投影追踪方法、和神经网络方法。 192 ~ 193

6.5.1 高维数据的变量选择

在处理高维问题的时候，变量选择是一种相当通用（而又敏感）的策略。考虑这样一个例子，用变量 X_1，…，X_p 预测变量 Y。很多时候并不是所有的 p 个变量都是准确预测所必需的。有些变量 X 可能跟被预测变量 Y 没有丝毫联系（比如某人生日的月份不可能与他的信用度有任何关系）。也可能存在两个或更多的变量包含同样的预测信息，从这个意义上讲，有些变量是冗余的。（例如**税前工资**和**税后工资**很可能是高度相关的。）

可以使用独立概念（第 3 章中介绍过）来以定量方式衡量相关性（relevance）。例如，如果 $p(y|x_1) = p(y)$对所有的 y 和 x_1 都成立的话，那么目标变量 Y 就独立于输入变量 X_1。如果 $p(y|x_1, x_2) = p(y|x_2)$，那么如果已经知道 X_2 的值，Y 就独立于 X_1。当然在实践当中不一定能够根据有限的样本确定出哪些变量是独立的，哪些不独立；也就是说，我们要估计这种影响。进一步来说，我们所感兴趣的不仅是严格的独立与不独立，而且还对独立的程度感兴趣。因此，我们可以（比如说）按照估计出的每个 X 变量和 Y 的线性相关系数来评价（rank）这个变量的重要性，线性相关系数可以告诉我们估计出的线性依赖性。如果 Y 是范畴型的（就像分类中那样），那么我们可以衡量 Y 和 X' 之间的平均相互关系信息（如下式）以给出对 X 和 Y 之间依赖性的估计。

$$I(Y;X') = \sum_{i,j} p(y_i, x_j') \log \frac{p(y_i, x_j')}{p(y_i)p(x_j')} \tag{6.10}$$

这里 X'是范畴型变量（例如一个实数值变量 X 的量子化版本）。

然而单个变量 X 与 Y 的相互作用不一定给出变量集合与 Y 之间相互作用的所有信息。一个经典的例子是布尔变量的奇偶校验函数，这个函数是这样定义的：如果变量 X_1，…，X_p 的值（二进制的）的和是偶数，那么 Y 为 1，否则为 0。这里 Y 独立于所有个别 X 变量，但却是整个变量集合的确定性函数。尽管这是一个多少有些偏激的例子，但它表明如果只注重个别 X 变量和 Y 之间的一对一相互作用，那么这种不可加非线性（non-linear non-additive）的相互作用就被掩盖了。因此，在一般情况下，分别评出的（比如用相关法评定的）k 个最佳 X 变量所组成的集合不等于容量为 k 的 X 变量最佳集合。因为 p 个变量的非空子集有 2^p-1 个，除非 p 很小否则穷举搜索是不可行的。同样糟糕的是，对许多预测问题，还不存在其最差搜索时间好于 O（2^p）的优化搜索算法（从确保找到最佳变量集合的意义上来说）。 194

这意味着在实践中子集选择方法往往依赖于启发式搜索来找到好的模型结构。很多算法

基于简单的启发进行“贪婪”的选择，比如每次加或减一个变量。在第 8 章中我们将回过头来探讨这种搜索问题。

6.5.2 高维数据的变换

处理高维数据的第二种通用策略是对预报变量进行变换（transforming）。这里的直观思想就是寻找一个含有 p'个变量（称之为 Z_1，…，$Z_{p'}$）的集合，这里的 p'远小于 p，变量 Z 定义为原始变量 X 的函数，变量 Z 的选择原则是从某个意义上讲使其成为适合具体任务的 p'个变量的最佳集合。

这种通用的模式——用对当前任务更加重要的较少变量取代观测变量——在数据分析的很多不同分支中经常出现。对 Z 的称呼在不同情况下有所不同，有基函数（basis function）、因素（factor）、潜在变量（latent variable）、主分量（principal component）等等，依赖于具体的目标和为了推导它们而使用的方法。在后续的章节中我们将详细的分析这些模型（以及与它们相关的拟合算法），这里我们仅通过两个例子说明这种基本思想：

- **投影追踪回归**（projection pursuit regression）使用如下形式的模型结构：

$$\hat{y}=\sum_{j=1}^{p'} w_j h_j(\alpha_j^T \mathbf{x}) \tag{6.11}$$

其中 $\alpha_j^T \mathbf{x}$ 是向量 $\mathbf{x}$ 在第 j 个权向量 α_j 上的投影（两个向量都是 p 维的，结果得到一个标量内积），h_j 是这个标量投影的非线性函数，w_j 是非线性函数的标量权。确定 w_j、h_j 的形式和“投影方向”α_j的过程是比较复杂而且依赖于具体算法的，但内在的思想非常普通的。

本质上这就是神经网络（第 11 章将详细讨论）的模型结构形式，举例来说，在神经网中通常把 h_j的函数形式选择为 $h_j(t)=1/(1+e^{-t})$。这类模型的一个局限是非常难于解释，除
195 非 $p'=1$。另一个局限是估计这些模型参数的算法在计算方面非常复杂而且对于庞大的数据集可能难以实现。在第 11 章中我们将深入探讨这一族模型。

- **主分量分析**（principal component analysis）：我们在第 3 章中介绍过主分量分析（PCA）。这是一种经典的技术，它把 p 个原始的预报变量替换为含 p 个变量的另一个集合（Z_1，…，Z_p），变量 Z 是由原始变量的线性组合形成的。组成原始数据集的原始向量被映射到 Z 空间中的新向量，而且正如第 3 章中所描述的，定义 $\mathbf{Z}$ 的权集合的选择目标是使按这些新变量表达的原始数据集的方差最大。因而主分量分析是投影追踪一种特例，只不过投影索引是沿投影方向的方差。主分量分析作为一种数据归约技术有两个优点。首先它顺序抽取 X 空间中的数据的绝大部分方差，因此可以期望仅仅前面几个分量（远小于全部原始变量的数目）就包含了数据的大部分信息。其次，根据分量抽取的方式（见第 3 章）可以得出它们是正交的，因而解释很方便。然而，应该注意的是 $\mathbf{X}$ 空间中的主分量对于优化预测不同变量 Y（举例来说）的性能来说不一定是理想的投影方向。比如，当我们要为数据中的组（或簇）间差异建模时（目的是分类或聚类），主分量投影不一定突出组间差异，而且实际上可能隐藏这些差异。（这对于更一般的投影追踪法也是基本适用的）。尽管如此，PCA 是一种广泛使用的维度归约工具。还有很多其他的维度归约技术（各有不同的特性），
196 包括因素分析（第 4 章）、投影追踪（第 11 章和上文）、独立分量分析等等。

6.6 用于结构化数据的模型

许多情况下，事先已经知道个体、变量或二者均具有某种定义完备的关系。例如线性链或序列（sequences）（测量值是有序的，比如蛋白质序列）、时间序列（time series）（测量值按时间排序，或许相同的时间间隔）和空间或图像数据（测量数据是定义在空间栅格上的）。有时甚至有更复杂的数据结构。例如在医学领域，人们可以得到不同时间反复测量的三维栅格脑部图像数据。

这些结构化的数据与本章其他地方讨论的测量值类型有着固有的差异。直到现在我们一直隐含地假定数据集中的 n 个对象个体（蛋白质、顾客等）是从潜在群体中随机抽取的数据样本。特别地，我们一直假定，对于给定的拟合模型，测量向量 $\mathbf{x}(i)(1 \leqslant i \leqslant n)$是彼此条件独立的（也就是说，数据的似然可被表达为单个 $p(\mathbf{x}(i))$的乘积）。例如，如果我们说体重这个变量服从正态密度模型，那么我们就假定了知道某个人的体重并没有得到数据集中其他人体重的任何信息。（当然这里忽略了可能存在的细微的依赖关系，比如数据集中同一个家庭的成员是顺序出现的，他们可能有相同的过重或过轻倾向。）因此，尽管上述假定是一种近似，但是我们一直工作在这个假定上，对于很多实际情况这个假定是很有价值的。

然而在有些问题中依赖关系是很明确的，因此需要对其建模。例如，在 24 小时内每隔 5 分钟测量一次某人的血压，很显然在连续测得的数据中非常可能存在依赖关系，那么怎样为这种依赖性建模呢？

一种途径是利用预期的变量间关系把对每个对象的多个观测减少到一个或少数几个变量（即固定的多元描述 $\mathbf{x}$）。这种方法有时被称为*特征提取法*。例如我们可能预期由于某种药物开始起作用，血压值会在 24 小时内下降，因而可以仅用两个变量（分别表示初始值和线性趋势的下降斜率）来代替对每个人的 60 次观测。或者使用同样的原则使用一条曲线来拟合下降率相对时间降低的情况。然后可以用标准方法分析描述每个对象曲线的数字（经常被称为*衍生变量*）。 197

注意这种途径（把序列化的测量值转化为非序列化的向量表示）可能足以完成有些给定的数据挖掘任务，但是通常这个过程存在信息损失，即丢失了原始测量数据中具有的时间或顺序信息。对于有些应用，这种序列化的信息可能是至关重要的。举例来说，假定有一个网页用户总体，在这个总体中有一组用户总是顺序地从网页 A 浏览到 B，再到 C，并按这个顺序反复重复。如果我们将这种信息转化为被访问网页的直方图（得到一幅具有三个大致相同柱条的直方图），那么就失去了发现潜在于数据中的动态循环模式的能力。

下面考虑一个序列化数据模型的例子，也就是用于 T 个序列化观测数据点（y_1，…，y_T）的一阶马尔可夫模型。注意即使是对于一个中等大小的 T 值，对 $p(y_1,y_2,\cdots,y_T)$的完整联合密度估计也将是非常复杂的（举例来说，如果 Y 取 m 个离散值的话，那么这个估计将需要确定 $O(m^T)$ 个数字）。因此在为具有一定结构的数据建模中，可以直接利用上一小节介绍的因式分解思想；也就是说，数据的结构会为我们要建立的模型暗示出一种自然的结构。因此，我们再回到一阶马尔可夫模型，定义如下：

$$p(y_1,\cdots,y_T) = p_1(y_1)\prod_{t=2}^{T} p_t(y_t \mid y_{t-1}) \tag{6.12}$$

如果做出平稳（stationarity）假定的话，也就是模型中的概率函数不依赖于特定的时间 t，即 $p_t(y_t \mid y_{t-1}) = p(y_t \mid y_{t-1})$，那么我们可以大大简化这个模型。这样，就可以把同一个条件概率函数用在序列的不同部分。这就大大地减少了建模需要的参数数量。例如，如果 Y 是 m 元（m-ary）的，那么非平稳模型需要 $O(m^2T)$ 个参数（序列中每个时间点一个 $m \times m$ 的条件概率矩阵），而平稳模型只需要 $O(m^2)$ 个概率（整个序列使用一个 $m \times m$ 的条件概率矩阵）。

198 平稳的概念还可用于更一般的马尔可夫模型，并不限于上述的一阶模型，而且事实上还可以很自然的拓展到空间数据模型（对于空间的情况我们将假定是相对空间平稳，而不是相对时间）。如果做出平稳假设的话，那么我们就不能把在统计模型中的变化表示为时间或空间的函数了。然而从参数化的角度来看，平稳假定是有很多好处的，因此它是建模中一种非常有用而且可行的假定——我们在以下的所有讨论中都采用此假定，除非有特别说明。

对式 6.12 中的马尔可夫模型有一种简单的产生式解释（见图 6-7，用 y 取代 x）。序列中的一个值 y_1 是根据某个初始分布 $p(y_1)$ 随机抽取的。当 y_1 确定下来以后，可以根据条件密度函数 $p(y_2|y_1)$ 随机选择 $t = 2$ 时的值。按这种方式确定了 y_2 以后，再由 $p(y_3|y_2)$ 产生 y_3，如此下去直到时间 T。

马尔可夫模型假定是很强大的（正如我们在 6.4.4 节中所讨论的）。简单来说，就是假定过去的影响可以完全被 $t-1$ 时的 Y 值所概括。特别的，Y_t 没有“远程”的依赖性，其仅依赖于 Y_{t-1}。很显然在许多情况下这个模型不够准确。例如，考虑为英文文本的语法结构建模，Y 取值为动词、形容词或名词等等。在这里一阶马尔可夫假定显得力不从心，比如说，因为确定动词的单复数要看主语的形式，所以要在句子中进一步向后追溯，而不仅是向后一个单词。

对 Y 取实数值的情况，马尔可夫模型通常被确定为正态条件分布：

$$p(y_t \mid y_{t-1}) = \frac{1}{\sqrt{2\pi}\sigma} exp - \frac{1}{2}\left(\frac{y_t - g(y_{t-1})}{\sigma}\right)^2 \tag{6.13}$$

这里 $g(y_{t-1})$ 充当了正态分布均值的角色（它是联系现在 y_t 和过去 y_{t-1} 的确定性函数），σ 是模型中的噪声（这里做了平稳假定）。通常把函数 g 选为 y_{t-1} 的线性函数，$g(y_{t-1}) = \alpha_0 + \alpha_1 y_{t-1}$，这就产生了著名的一阶自回归模型（first-order autoregressive model）：

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + e \tag{6.14}$$

其中 e 是均值为 0 标准差为 σ 的高斯噪声，α_0 和 α_1 是模型的参数。注意在这些假定下，可

199 以把公式 6.14 表达成公式 6.13 的形式。

从产生式角度来看，对公式 6.14 有一种简单的解释：在序列中时间 t 处的值 y_t 是通过取前驱值 y_{t-1} 乘以常量 α_1，加上偏移量 α_0，再加上随机噪声 e 得到的。要使 y 保持稳定（当 $t \to \infty$ 时有界），那么必须 $-1 < \alpha_1 < 1$。$|\alpha_1|$ 越接近 1，相继 y 值之间的依赖性越强，反之越弱。这一模型结构明显与 6.3 节中讨论的标准回归模型结构有着密切的关系。只不过 Y 不再是在独立的 X 值上回归，而是在自身的“过去”值上回归。于是，根据回归模型结构的知识，我们立即可以想到很多种推广上述一阶模型的方法。例如 y_t 可以依赖于序列中更早的过去值；用 $g(y_{t-1}, y_{t-2}, \cdots, y_{t-k})$ 取代式 6.13 中时间 t 的均值 $g(y_{t-1})$，这就是 k 阶马尔可夫模型。通常仍然是把 $g(y_{t-1}, y_{t-2}, \cdots, y_{t-k})$ 选为一种简单的线性模型 $\alpha_0 + \sum \alpha_i y_i$。原则上除了线性模型以外，还可以采用 6.3 节中讨论的任何一种函数形式，比如可加模型、多项式模型、局部线性模型和数据驱动局部模型等等。

对目前为止我们所讨论的马尔可夫模型的一种进一步的重要推广是对隐藏状态变量的概念显式建模。关于时间序列模型和空间模型隐含状态的一般概念在工程和科学研究中很普遍，而且在许多函数模型中反复出现。这种结构的具体例子包括隐马尔可夫模型（HMM）和 Kalman 滤波器。通过观察 HMM 的对应图形模型结构（见图 6-11）很容易解释它的结构。从产生式的角度看，一阶 HMM 结构是这样工作的（沿着链条从左到右移动产生观测点）。隐含变量是范畴型的（对应于 m 个离散状态），而且是一阶马尔可夫的。因此，x_t 是按一般的一阶马尔可夫链方式从条件分布函数 $p(x_t \mid x_{t-1})$中通过抽样的产生的，其中 $p(x_t \mid x_{t-1})$是 $m \times m$ 的条件概率矩阵。产生了在时间点 t 的状态（值为 x_t）后，就可由概率函数 $p(y_t \mid x_t)$产生观测值 y_t。y_t 可以是一元也可以是多元的；可以是数值型的也可以是范畴型的，或者是它们的组合。因此在 HMM 结构中观测值 y_t 仅仅依赖于时间点 t 处的状态，而且状态序列是一阶马尔可夫链。状态序列是未观察到的或隐藏的，而 y 是直接观测到的：因此关于哪个特定的状态序列产生了数据具有不确定性（对于给定的模型结构和观察到的 y 集合）。

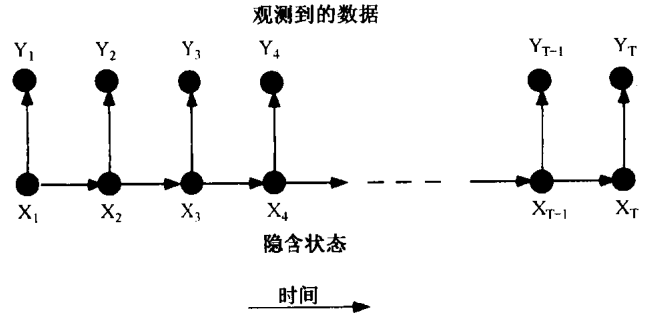


图6-11 对应于一阶隐马尔可夫假定的图形模型结构

可以把 HMM 结构想像为一种混合模型（对于变量 Y 来说有 m 个不同的密度函数），我们已经向这个混合模型中的“相邻”分量 x_t 和 x_{t+1} 间加入了马尔可夫依赖性。更准确地说，对于一阶 HMM 可以把观测到的序列和任意特定的隐含状态序列的联合概率写成：

$$p(y_1,\ldots,y_T,x_1,\ldots,x_T)=p(x_1)p(y_1 \mid x_1)\prod_{t=2}^{T} p(y_t \mid x_t)p(x_t \mid x_{t-1}) \qquad (6.15)$$

等式右侧的因式分解明显来自于图 6-11 的图形模型结构。如果把上式看作分布参数的函数，那么这就是变量（Y_1，…，Y_T，X_1，…，X_T）的似然。观测到的各个 y 的似然可用于把这种模型拟合到数据（即获取参数 $p(y_t \mid x_t)$和 $p(x_t \mid x_{t-1})$）。要计算 $p(y_1$，…，$y_T)$（观测到数据的似然）就必须把左边的项对 m^T 个可能状态序列进行累加，乍一看来，这似乎牵涉到要对指数级数量的项进行累加。幸运的是有一种方便的递归方法可以在 $O(m^2T)$时间内完成这一计算。

很明显，还可以对一阶 HMM 结构进行很多种不同方向的拓展。k 阶马尔可夫模型就是使 x_t 依赖于前面的 k 个状态。也可以推广 y 的依赖性，例如使 y_t 既线性依赖于前面的 k 个 y（就像自回归模型那样），又直接依赖于 x_t。这便把通常的自回归模型结构自然地推广到了*混合自回归模型*（mixture of autoregressive model），我们可以将其想像为是在 m 个不同的自回归模型之间以马尔可夫链的方式进行切换。Kalman 滤波器与 HMM 有着紧密的关系，只不过隐含状态是取实数值的（比如一台机器的未知速度或动力），但该模型的独立结构与我们已经讨论的 HMM 的情况本质是相同的。

对隐马尔可夫模型的产生式描述，计算机方面的学者会很容易地联想起有限状态机（FSM）。事实上正如我们这里所描述的，一阶 HMM 直接等价于带有 m 个状态的随机有限状态机；也就是说，下一个状态是由 $p(x_{t+1} \mid x_t)$控制的。这提示我们可以从不同语法的角度来推广这种模型结构。有限状态机是正则语法（regular grammar）的一种简单形式。再上一层（按照所谓的 Chomsky 语法层次）是上下文无关语法（context-free grammar），可以将其看作是用堆栈丰富了有限状态机器，允许模型结构“记忆”长范围的依赖关系，比如句尾的封闭括弧等。随着语法层次的上升，模型结构的表达能力越来越强，但对数据的拟合也越来越困难了。因此尽管正则语法（或者说 HMM）在结构上比较简单，但是由于那些复杂的结构拟合到真实数据非常困难，所以绝大部分使用马尔可夫模型拟合序列数据的应用还是以正则语法为基础的（相对其他更复杂的语法结构而言）。

最后，尽管这里只描述了简单的数据结构，Y 来自于有序序列，但是很显然我们可以将马尔可夫模型结构推广到对更一般的数据依赖关系（比如二维栅格中的数据）进行建模。例如马尔可夫随机场（Markov random field）实质就是马尔可夫链在多维情况下的推广（比如在二维空间中，我们可以用栅格结构来表示图形模型，而不是链结构）。

事实证明这种模型比链模型更加难以分析和使用。例如对于像汇总似然中的隐含变量（公式 6.15）这样的问题，通常没有可驾驭的求解方法，必须使用近似。因此，处理空间数据比
202 处理序列数据更困难，尽管概念上平稳思想、马尔可夫模型、线性模型等等都适用。一种处理栅格化数据的常见方法是把二维栅格数据（如 $n \times n$ 个栅格点）“调整”为一个长度为 n^2 的单一向量，然后对这些向量进行主分量分析，从而把测得的栅格数据投影到一个小的 PCA 向量集上去，再在这个维度降低了的空间上用标准多元模型对数据进行建模。这种方法忽略了原始栅格数据中的大多数空间信息，尽管如此，该方法在很多情况下还是非常实用的。类似的，对于多元时间系列或序列，在同一时间段里有 p 个不同的时间系列或序列测量值（例如在同一病人身上的不同生物医学监控器），可以采用 PCA 把 p 个原始时间系列减少到数量大大减小的若干“分量”系列，然后再进行进一步的分析。

6.7 模式结构

本书通篇把模型作为对整个（或绝大部分）数据集的全局性描述，而把模式作为是对数据集的某些局部特征的描述。可以把一个模式看作是一个谓词，对于数据集中出现了该模式的那些对象或对象局部它返回真，否则返回假。要定义一类模式，我们需要做两件事情：一是确定模式的语法（说明如何来定义模式的语言）；二是模式的语意（如何解释模式所适用的数据）。在这一节中，我们讨论用于两种不同类型离散值数据的模式：标准矩阵形式的数据和被描述为字符串的数据。

6.7.1 数据矩阵中的模式

建立模式的一种一般方法是从元模式开始，然后再用逻辑连接符把它们组合起来。（另一种方法是为特定的应用建立某种类型的特殊模式）。再回到数据矩阵表示，并假定有 p 个变量 X_1，…，X_p。令 $\mathbf{x} = (x_1, \cdots, x_p)$为这些变量的 p 维测量向量。我们将数据集合中的第 i 个数据个体表示为 $\mathbf{x}(i)$，$1 \leqslant i \leqslant n$。整个数据集为 $D = \{\mathbf{x}(1), \cdots, \mathbf{x}(n)\}$。自然地，$x_k(i)$是
203 第 i 个数据个体的第 k 个测量值。

一般地说，变量 X_1，…，X_p 的一个模式标识了这些变量所有可能观测值的一个子集。可以从元模式（primitive pattern）开始逐步建立一种语言来表示一般的模式。元模式就是变量的取值情况。例如如果 c 是 X_k 的值，那么 $X_k = c$ 就是一个元模式。如果 X_k 的值是有序的（比如是实轴上的数字），那么还可以包括像 $X_k \leqslant c$ 这样的不等式作为元条件。需要的话元模式也可以包括多元条件，比如对于实数值数据的 $X_kX_j > 2$，以及对于离散值数据的 $X_k = X_j$。

有了元模式，就可以使用像 *AND*（∧）、*OR*（∨）这样的逻辑连接符来建立更复杂的模式。例如可以建立一个模式：

$$(\textbf{年龄} \leqslant 40) \wedge (\textbf{收入} \leqslant 10)$$

来描述工资单数据库内输入记录的一个子集。值得一提的是，分类树的每一个分支（第 5 章讨论的）都是一个这种形式的合取模式。另外一个例子如

$$(\textbf{回形针} = 1) \wedge (\textbf{啤酒} = 1 \vee \textbf{软饮料} = 1)$$

描述了购物篮数据库内各记录行的一个子集。

模式类（pattern class）是一组合法模式的集合。一旦确定了一组元模式和组合这些元模式的合法方式，便定义了一个模式类 C。例如，如果变量 X_1，…，X_p 的取值范围都是{0, 1}，那么我们可以定义一个模式类 C，它是由以下形式的所有合取式组成的：

$$(X_{j1}=1) \wedge (X_{j2}=1) \wedge \cdots \wedge (X_{jk}=1)$$

在数据集 D 中频繁出现的模式被称为（变量的）频繁集（frequent set）。因为每一个这样的模式都是由变量的某个子集所唯一确定的：所以可以把这种模式简写为$\{X_{i1}, X_{i2}, \cdots, X_{ik}\}$。从数据集中寻找像频繁集这样的合取模式是比较简单的，在第 13 章中我们将对此做详细的讨论。

给定一个模式类和一个数据集 D，那么模式的一个重要特性是它在数据集中的频率。可以把模式 ρ 的频率 $fr(\rho)$定义为数据集中使这个模式为真的观测值的相对数目。有些情况下，数据挖掘仅对出现相当频繁的模式感兴趣。然而频率接近零的模式也可能包含丰富的信息。（的确，有时就是那些罕见而且不寻常的模式具有特别的意义。）当然模式频率不是模式的 204
唯一重要特性。诸如语意的简洁性、可理解性和模式的新颖性或新奇性显然也都是我们所感兴趣的。举例来说，对于数据集中的任何一套特定的观测值（x_1，…，x_p），我们都可以写出一个与观察值完全匹配的合取模式（$X_1 = x_1$）∧…∧（$X_p = x_p$）。所有这样的合取模式的析取形成的模式在数据集中的频率是 1。然而这样的模式以一种臃肿的方式重写整个数据集，根本没有意义。

对于给定的模式类，模式发现的任务就是从该类中寻找相对数据集来说满足一定条件的所有模式。例如，我们可能对寻找满足以下条件的所有频繁集模式感兴趣：它们的频率至少为 0.1，而且变量 X_7 出现在其中。广义上讲，模式发现任务的定义还包括模式的信息性、新颖性和可理解性等条件。定义模式类和模式发现任务的难点是如何平衡模式的表达能力、综合能力与求解这个任务的计算复杂度之间的矛盾。

如果给定了模式类 C，那么可以很容易地定义规则。一条规则就是一个这样的表达式 $\rho \Rightarrow \varphi$，这里 ρ 和φ是模式类 C 中的模式。一个逻辑规则的语意就是：如果表达式 ρ 对一个对象来说为真，那么φ也为真。我们可以放宽这个定义，以支持从 ρ 到φ映射的不确定性，也就是如果 ρ 为真，那么φ以一定概率为真。这种规则的精度（accuracy）被定义为 $p(\varphi \mid \rho)$，也就是当 ρ 对于一个对象为真时φ也为真的条件概率。正如第 4 章讨论的，用近似频率计数我们可以很容易的估计出这个概率；即

$$\hat{p}(\varphi \mid \rho) = \frac{fr(\rho \wedge \varphi)}{fr(\rho)}$$

规则 $\rho \Rightarrow \varphi$的支持度（support）$fr(\rho \Rightarrow \varphi)$被定义为 $fr(\rho)$（适用规则的对象的比例）或 $fr(\rho \wedge \varphi)$（对其来说规则左右两边都为真的对象的比例）。

例如，如果模式是频率集，那么规则的形式为：

$$\{A_1, \cdots, A_k\} \Rightarrow \{B_1, \cdots, B_h\}$$

这里每一个 A_k 和 B_j 都是二进制变量。这个规则的完整形式为：

205

$$(A_1 = 1) \wedge \cdots \wedge (A_k = 1) \Rightarrow (B_1 = 1) \wedge \cdots \wedge (B_h = 1)$$

这样的规则被称为关联规则，它是数据挖掘中广泛使用的一种模式结构（在第 13 章中我们将详细的讨论寻找这种模式的算法原理）。

藉此，我们已经介绍了定义原始数据子集的模式。也就是，每个模式都是由用仅指向单一观测的变量的规则所定义的。然而在某些情况下我们需要使用指向多个观测的变量的模式。例如我们可能希望标识出地理数据库中可以组成等边三角形的所有顶点。举一个更正式的例子，考虑一个具有离散变量 $A_1, \cdots, A_p$ 的数据集。函数依赖性是如下形式的一个表达式：

$$A_{i_1} A_{i_2} \cdots A_{i_k} \Rightarrow A_{i_{k+1}}$$

其中 $1 \leqslant i_j \leqslant p$，$i = 1, \cdots, k+1$。注意这与关联规则的定义在语法上很相像。然而，由这个表达式所定义的函数依赖性为真的条件是：对于数据集中所有的观测数据对 $\mathbf{x} = (a_1, \cdots, a_p)$和 $\mathbf{y} = (b_1, \cdots, b_p)$，如果对于所有的变量 A_{i_j}，$j = 1, \cdots, k$，$\mathbf{x}$ 和 $\mathbf{y}$ 是一致的，那么对于 A_{ik+1} 它们也是一致的。也就是说，如果对于所有的 $i = 1, \cdots, k$，$a_{i_j} = b_{i_j}$，那么 $a_{i_{k+1}} = b_{i_{k+1}}$。函数依赖性起源于数据库设计，而且对查询优化也很有意义。知道数据集中的函数依赖性有助于理解数据结构。

这里的模式或写在模式中的条件仅限于出现在数据库单个记录中的值。有时我们也可能希望描述引用其他观测值的模式，比如对应于“在其所在部门中收入最低的雇员”的模式。也可以用逻辑形式来描述这样的条件。例如：

$$\{\mathbf{x}_k \mid \text{年龄} \leqslant 40 \wedge \text{收入} \leqslant 10\}$$

6.7.2　字符串模式

在上一节中我们讨论了适用于传统的矩阵形式数据的模式。其他类型的数据需要其他类型的模式。为了说明这一点，在本节中我们讨论一下字符串模式。确切地说，字母表 S 上的一个字符串是 S 元素（也就是字母）的一个序列 $a_1, \cdots, a_n$。字符表 S 可以是二值字母表{0, 1}、ASCII 代码集、DNA 字母表{A, C, G, T}、或者由 ASCII 字母组成的所有单词的集合。由 S

206

中的字母组成的所有字符串的集合被表示为 S^*。

字符串数据和标准矩阵形式的数据的区别在于：对字符串而言不存在固定的变量集。如果要使用概率概念描述字符串数据的话，可以把字符串中的每一个字符看成一个随机变量。

数据可以是一个或几个字符串，大多数情况下，我们的兴趣在于寻找特定模式在字符串中出现的次数。（例如求出某个 DNA 序列在一个大序列集中的出现次数）。最简单的字符串模式是子串（substring）：比如如果对于所有 $j = 1, \cdots, k$ 都有 $a_{i+j-1} = b_j$，那么便说模式 $b_1 \cdots$

b_k 出现在字符串 $a_1 \cdots a_n$ 的 i 位置上。例如对 DNA 序列，我们感兴趣的可能是找出子串模式 ATTATTAA 的出现次数，对于 ASCII 字符串，感兴趣的可能是模式 data mining 是否出现在给定字符串中。

然而我们可能对字符串中的模式类更感兴趣。一个正则表达式（regular expression）E 定义了一个字符串集合 $L(E)$。进一步来说，表达式 E 可以是下列情况之一：

1. 一个字符串 s，那么 $L(s) = \{s\}$；
2. E_1 和 E_2 的串联 E_1E_2；这种情况下集合 $L(E_1E_2)$ 由 $L(E_1)$ 和 $L(E_2)$ 中任何两个字符串的串联组成；
3. E_1 和 E_2 的选择 $E_1|E_2$；那么 $L(E_1|E_2) = L(E_1) \cup L(E_2)$；
4. E 的递归 E^*；那么 $L(E^*)$ 由 $L(E)$ 中的 0 个或多个字符串串联组成的所有字符串。

如此说来，10(00|11)*01 就是一个正则表达式，它描述了以 10 开始，01 结束，中间包含一系列 00 和 11 序偶的所有字符串。

正则表达式是特别适于描述有趣字符串类的一种模式形式。虽然有些简单类型的字符串无法用正则表达式描述（如由所有对称括弧序列组成的字符串集合），但是可以用它来表达许多非常复杂的字符串规律。

虽然正则表达式可以很好的地定义字符串模式，但对于表达事件发生次数的变化来说，它的表达能力还不够。能够做到这点的一种简单模式类是片段（episode）。从顶层来看，一个片段就是一起发生事件的一个部分有序集（partially ordered collection）。事件可以是不同类型的，而且可以指向不同的变量。例如在生物学统计数据中，“先是头痛，然后在一定时间段内伴随一种昏迷感觉”便是一个片段。片段对干扰事件（比如通讯过程中的警鸣、用户接口行为的记录等等）不敏感，这是很有用的。也可以把片段和前面讨论的各类规则一起使用。 207

6.8 参考读物

讨论回归建模的书籍有很多。Draper and Smith（1981）以及 Cook and Weisberg（1999）都很精彩。McCullagh and Nelder（1989）是关于推广的线性模型的权威著作，而 Hastie and Tibshirani（1990）则是关于推广的可加模型的权威著作。Fan and Gijbels（1996）广泛讨论了局部多项式方法，而 Wand and Jones（1995）更偏重于理论的探讨的核估计方法（这两本书都是关于回归和密度估计的）。Hand（1982）详细的讨论了核估计方法在有指导分类问题中的应用。

McLachlan（1992）、Ripley（1996）、Bishop（1996）、Mitchell（1996）、Hand（1997）和 Cherkassky and Muller（1998）都比较深入的讨论了分类建模。McLachlan 和 Ripley 的教材主要是针对统计方面的读者。Ripley 的著作的一个显著特点是使用了大量不同类型的数据集来阐述基本概念。Bishop、Cherkassky 和 Muller 的著作更侧重于神经网络和有关的进展，而且每一本书中都包含了很多不同于主流统计文献的思想。Duda and Hart（1973）仍然是关于分类建模的经典之作，非常清晰全面的讨论了分类建模的核心思想。在《统计和计算》（Statistics and Computing）杂志第 8 卷第一期里有对 Bishop（1996）、Ripley（1996）、Looney（1997）、Nakhaeizadeh and Taylor（1997）的评论。

关于混合模型的综合性教材包括 Titterington, Makov and Smith（1985）、McLachlan and

Basford（1988）以及 McLachlan and Peel（2000）。关于这一领域的一般性讨论还有 Redner and Walker（1984）以及 Everitt and Hand（1981）。Silverman（1992）是一本包含了有关密度估计的大量内部细节的教材，Scott（1992）也是关于这一主题的，书中对“average-shifted
208 histogram”模型的讨论特别值得关注，该模型综合了直方图和核估计的特征，这可能对面向海量数据集的直方图模型很有意义。

Jolliffe（1986）是专门讨论主分量方法的教材。Huber（1985）详细探讨了投影追踪法，Hyvarinen（1999）全面分析了独立分量分析和有关的维度归约技术。

Elliott et al.（1995）和 MacDonald and Zucchini（1997）讨论了隐马尔可夫模型。Chatfield（1996）介绍了大量有关自回归和时间序列模型的文献，非常值得一读。Harvey（1989）、Box, Jenkins and Reinsel（1994）以及 Hamilton（1994）从数学角度深入地讨论了时间序列建模并展望了它的应用。Kim and Nelson（1999）深入的探讨了如何切换各个模型。Cressie（1991）是关于空间数据分析的著名教材，Dryden and Mardia（1998）广泛地讨论了二维形体的建模问题。Grenander（1996）讨论了用于序列数据和空间数据的产生式模型，该书把统计学和计算机科学中的很多思想联系起来，很吸引人。

Ramsey and Silverman（1996）讨论了对时间和（或）空间数据建模的通用方法，例如对来自不同气象站的时间序列数据建模。Crowder and Hand（1990）、Hand and Crowder（1996）、Diggle, Liang and Zeger（1994）以及 Lindsey（1999）讨论了对重复测量建模的方法。

采用逻辑公式描述模式的思想在数据库系统中应用很广。比如 Ramakrishnan and Gehrke（1999）和 Ullman and Widom（1997）都是这方面的入门级教材。Agrawal, Imielinski and Swami（1993）介绍了频率集。许多有关计算理论的书籍都介绍了正则表达式，比如 Lewis and Papadimitriou（1998）。Gusfield（1997）也讨论了文本模式概念。Mannila, Toivonen and Verkamo
209 （1997）探讨了 episode 的概念。

第 7 章　数据挖掘算法的评分函数

7.1　简介

在第 6 章，我们重点讨论了可用于把模型和模式拟合到数据的不同表示方法和结构。现在我们可以讨论如何将这些结构拟合到数据了。回想模型或结构是一种函数形式，它的参数是“浮动的”。例如：$Y = aX + b$ 就是一种这样的模型结构，其中 a 和 b 是参数。如果确定了模型或模式结构，那么我们必须根据数据评价不同的参数值设定，以便我们能够选择出一个好的参数集（或者甚至是“最好”的）。在第 1 章的简单线性回归例子中，我们介绍了如何使用最小平方原理从不同的参数值中选取最优的参数。这包括寻找参数 a 和 b 的值使函数 y 的预测值（通过模型计算而得）与 y 的实际观察值（数据）之间的差异平方和最小化。在这个例子中，评分函数就是模型的预测值与实际观测值之间的差异平方和。本章的目标是介绍更多可用于数据挖掘的评分函数，以扩展读者这方面的视野。我们将看到历史悠久的误差平方评分函数只是众多评分函数中的一种，而且实际上它可以被看作是更一般理论的一个特例。

为什么我们要重视评分函数呢？弄清这个问题是非常重要的。从根本上说使用评分函数的目的是用函数的形式来评价一个模型对于数据挖掘者来说的有用程度。然而不幸的是，在实践中对于构建模型的人来说评价和度量模型在实际应用方面的“有用”程度是非常困难的。例如：在预测股票市场的回报率时，人们可能使用预测数据和实际数据的误差平方作为评分函数来训练它的模型。然而，如果把这个模型应用到实际的经济环境中，那么许多诸如交易成本、风险、多样性等其他因素便开始作用并影响这个模型的实际效果。这解释了为什么我们经常满足于更简单的“通用”评分函数（例如误差平方），它们具有很多期望的被普遍接受的特征，同时又易于使用。当然，我们不应该走极端：所使用的评分函数应该尽可能反映数据挖掘任务的整体目标。应该努力避免为了方便（比如是因为使用软件包的缺省设置）而使用与数据挖掘任务完全不相关的评分函数，不幸的是这种情况在实践中出现的非常多。

不同的评分函数具有不同的属性，并且适用于不同的情况。本章的一个目的就是使读者明白这些不同并且理解使用某个评分函数而不使用另一个的真实内涵。正像模型和模式结构中蕴含着一些基本原理一样，不同的评分函数也有一些基本的原理。这些正是本章的重点。

在一开始从三个角度来区分评分函数是很有用的。一是用于模型的评分函数同用于模式的评分函数之间的差别；二是用于预测性结构的评分函数同用于描述性结构的评分函数之间的差别；三是用于具有固定复杂度的模型的评分函数同用于具有不同复杂度的模型的评分函数之间的差别。下面的章节将说明这些差别。

有必要对下文使用的术语作个小小的说明。在某些地方，我们提到的是显然希望被最小化的评分函数（比如误差），然而在另外一些地方我们提到的是显然希望被最大化的评分函数（例如对数似然）。这两种情况的根本概念是一致的：因为一个“基于误差”的评分函数的负数形式（或相反的）就可以被最大化了，反之亦然。

7.2　对模式进行评价

由于从数据中寻找局部模式的完整概念是近年来才形成的，所以相比于用于评价模型的略显过剩的技术来说可用于评价模式的技术要少得多。实际上，目前确实还没有关于如何评价模式的一致结论。一种模式在实际中的有用程度很大程度上还是取决于观察
212 者的看法。某些人认为是噪声的孤立点可能被其他人认为很有价值。从根本上说，可以根据模式对数据分析者的有趣度和未知度来评估模式。但是仅当我们具有了关于用户目前实际已经知道知识的精确模型后，我们才有可能量化这种有趣度和未知度。我们都有过类似这样的经历：我们第一次得知某些令人惊讶的事情时的感触会比我们第五次或第十次重复听到同样的信息时更深。所以，一个模式对某个人的有趣程度必然依赖于他的以前知识。

然而在实践中，我们不能指望（除非在简单情况下）能够对一个人的以前知识建模。面对一个数据集合，科学家或市场专家也难以精确地表达出关于这个问题他们已经知道了多少知识，即使是主观的贝叶斯理论在选择用于复杂的多参数模型的先验时也会遇到问题——通过选择标准形式的先验来回避这个问题，也就是只对以前知识进行简单化的表示。我们发现：一旦某些模式开始从数据中浮现出来时（利用可视化、描述统计学或者通过数据挖掘算法），数据库的拥有者经常会说“噢!是的，不过我们已经知道了”，一旦他们已经看到了数据，那么他们声称的一直所期待的模式就改变了。

尽管说了这么多，但是目前的事实依然是：在数据挖掘中使用的大多数评价模式实质上是都假定它们是相对于一个完全无信息的先验模型来衡量信息性（informativeness）的；也就是，实际上假定了数据分析者对于当前的问题根本没有任何以前知识，除了一些简单的边缘和描述性统计量。这样做的目的是排除非常显而易见的模式（而把注意力集中在那些不同于已知简单模式的模式上），然后让用户对算法发现的其余模式作“后期修剪”以保留真正感兴趣的模式。当然，这样做的危险是：对于某些数据集合和某些模式搜索形式，数据挖掘算法发现的几乎所有模式对数据分析者都是根本无趣的。

为了举例说明以上观点，我们选择一个简单的（但是被广泛应用的）模式结构——概率规则（参见在第 5 章中关于关联规则的讨论），我们将在后面的第 13 章中对此进行更详细的讨论。概率规则具有以下的形式：

IF a THEN b 的概率是 p

其中 a 和 b 都是定义在我们感兴趣变量的子集上的布尔命题（事件），而且 $p = p(b \mid a)$。
213 对于一个没有任何信息的观察者来说，我们如何来衡量这一规则的有趣度和信息度呢？一个简单的方法就是假设这个观察者已经知道了事件 b 概率的边缘（或者说是绝对）分布——$p(b)$。

例如，假设我们正在研究由数据挖掘者组成的总体。用 b 来表示从这些人中随机选择一个人是数据挖掘研究者的事件，而用 a 来表示这个人已经读过本书的事件。假设我们发现 $p(b) = 0.25$ 同时 $p(b|a) = 0.75$；就是说在这样一个数据挖掘者的总体中有 25%的人是数据挖掘的研究者，而且读过本书的人中有 75%是数据挖掘研究者。这是非常有趣的，因为这告诉我们，在读过本书的人当中正在进行数据挖掘研究的人占有的比例比我们正在讨论的数据挖掘

者总体中在进行数据挖掘研究的人的比例要高。（因此，也就暗示了：在读过本书的人当中正在进行数据挖掘研究的人占有的比例比没有读过本书的人正在进行数据挖掘研究的比例高）。需要说明的是，从另外一个方面讲，这并没蕴含任何因果关系。这有可能是本书鼓励了读者从事研究，也可能是已经做研究的读者期望本书能够给予他们帮助。

用于表征信息度的简单评分函数一般离不开先验概率 $p(b)$同后验概率 $p(b|a)$（在知道事件 a 是真的情况下）之间的“距离”。因此，举例来说，一种可能的尺度就是简单地计算这两个概率之间的绝对距离$|p(b|a)-p(b)|$，或者计算对数赔率比例的差值 $\log\frac{p(b|a)}{p(\bar{b}|a)}$，其中$\bar{b}$代表某个人不是数据挖掘研究者这一事件。

当我们比较不同的模式时，比如说比较 $p(b|a)$和 $p(b|c)$时，考虑模式的*覆盖面*（coverage）——也就是说这个模式所适用数据占的比例——是非常有用的。继续我们上面所举的例子，用 c 来表示随机选择的数据挖掘者是本书的三位作者之一这一事件。第二个模式可能是“如果 c 那么 b”（“如果一个任选的数据挖掘者是本书的三位作者之一，那么它是一个数据挖掘的研究者”），并且 $p(b|c)=1$，因为本书的三位作者都是数据挖掘的研究者。事实上，事件 c 仅适用于三位数据挖掘者，是数据挖掘者这个集合中很小的一个部分。另一方面，（我们希望）事件 a 的覆盖面更大一些；也就是说，事件 a 的概率 $p(a)$显著大于事件 c 的概率 $p(c)$。为了说明这一点，假设 $p(a)=0.2$ 而 $p(c)=0.003$。那么，尽管第二个模式非常精确（$p(b|c=1)$），但是并不是非常的有用，因为它仅仅适用于整个总体中很小的一部分（0.3%），而第一个模式尽管不是很精确($p(b|a)=0.75$)，但却具有较广泛的适用性(达到总体的 20%)。可以很容易的设计出不同的尺度来增大评分函数对覆盖面的重视。例如，我们可以把前面定义的评分函数乘上条件事件的概率：$p(a)|p(b|a)-p(b)| = |p(b,a)-p(b)p(a)|$，可以把这一尺度解释为：衡量 214
了两个概率事件 a 和 b 在假定独立情况下的概率和观察到的联合概率之间的差异。另一种方法是定义一个阈值 p_t，然后仅搜索覆盖面大于 p_t 的模式，在关联规则挖掘中所使用的（第 5 章和第 13 章）就是这种方法。

在数据挖掘文献中，还提出了其他众多用于评价模式的评分函数。不过都没有得到广泛的接受或成为通用的方法，这很大程度上是因为判断一个模式的新颖性和有效性经常是非常主观的和面向具体应用的。因而，目前实践中使用得最多的方法还是邀请该领域的专家进行人工解释（也就是让人来浏览并解释数据挖掘算法所产生的候选模式）。

7.3 预测性评分函数和描述性评分函数

我们现在转向讨论用于模型的评分函数。相对于评价模式的评分函数来说，可供选择的评价模型的评分函数要多得多。

7.3.1 评价预测模型的评分函数

一个很方便的切入点是考虑预测性评分函数和描述性评分函数之间的区别。用于预测问题的评分函数都是非常直截了当的。在预测任务中，训练数据具有“目标”值 Y，对于回归来说 Y 是一个数量型变量，对于分类来说 Y 是一个范畴性变量；而且数据集合 D={($\mathbf{x}(1)$, $y(1)$)，…，($\mathbf{x}(n)$，$y(n)$)}是由输入向量和目标值这样的对偶组成的。令 $\hat{f}(x(i);\theta)$为模型使用

参数值 θ 对个体 i 作出的预测，$1 \leqslant i \leqslant n$。令 $y(i)$为训练数据集合中对应于第 i 个个体的实际观测值（或称为“目标”值）。

很明显，我们的评分函数应该为预测值 $\hat{f}(\mathbf{x}(i);\ \theta)$与目标值 $y(i)$ 间差值的函数。对于 Y 为数量型变量的情况，普遍使用的评分函数包括误差平方和等：

$$S_{SSE}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(\hat{f}(\mathbf{x}(i);\theta) - y(i))^2 \tag{7.1}$$

对于 Y 为范畴型变量的情况，普遍使用的是误分类率（或称误差率，又叫“0-1（zero-one）”评分函数），也就是：

215

$$S_{0/1}(\theta) = \frac{1}{N}\sum_{i=1}^{N} I(\hat{f}(\mathbf{x}(i);\theta), y(i)) \tag{7.2}$$

其中，当 a 不等于 b 时，$I(a,b) = 1$ 否则等于 0。这是分别用于回归和分类的两种应用最广的评分函数。这两种评分函数很简单易懂并且经常可以使优化问题变得非常直接明了（至少对于误差平方和是这样）。

然而需要说明的是，我们已经在这些评分函数的定义中作了一些很强的假定。例如，在对每个个体误差求和中我们假定所有个体的误差都被平等地看待。这是一个非常普遍而且通常很有用的假定。然而，如果（举例来说）我们有一个数据集，其中的测量值是在不同时间测出的，那么，我们或许希望在预测评分函数中给最近几次的测量值分配更大一些的权。类似地，在数据集中我们可能有不同的条目子集，某些条目子集中对应的目标值可能比另外一些子集中的更可靠一些（比如可以根据子集测量误差的某个量化指标来判断）。这样我们可能希望在预测评分函数中给那些测量值可靠性较低的条目分配较小的权。

此外，两种评分函数都仅是关于预测值和目标值之间差异的函数——特别值得一提的是它们并不依赖于目标值 $y(i)$。我们需要对此作一些必要的考虑。例如，如果 Y 是表示某人是否患有癌症这一事件的一个范畴型变量，那么我们可能希望给没有检查出真的癌症患者这一误差较大的权，而给误报癌症这一误差较小的权。对于 Y 给出真实值的情况，*误差平方*可能是不恰当的——或许*误差绝对值*可以更恰当地反映模型的质量（误差平方对 Y 的观察值和 Y 的预测值间的极端差异会比误差绝对值给出更大的权）。再举一个例子（第三个例子），在投资方案中，我们更容易接受（能容忍）Y 的预测值低估实际值的情况，而不愿意接受*高估*的预测（从风险的角度来考虑），这告诉我们有时采用不对称函数可能更恰当。

以上介绍的基本评分函数都相当地简单。因此，我们可能需要在实际应用过程中对这些基本评分函数进行调整，以更加地精确反应我们的数据挖掘项目的具体目标。有时候这种调整并不简单（定义“实际目标值”可能就很困难，尤其是在数据挖掘中很多问题经常都是开放性的（open ended））。在另外一些情况下，即使我们不能精确地描述目标值，也能够改善基本评分函数。例如，对于癌症问题的例子，可以不使用 0-1 损失函数，而定义一种基于*成本矩阵*（cost matrix）的评分函数可能更恰当。于是，令 $\hat{k}$ 为预测出的类，令 k 为实际所属的类，那么定义一个“成本”矩阵 $c(\hat{k},k)$，$1 \leqslant \hat{k}$，$k \leqslant K$ 来反映把一个实际属于 k 类的患者
216 分类到 $\hat{k}$ 类中的严重程度。

在选择一个用于特定预测性数据挖掘任务的评分函数时，通常要在选择简单评分函数（比如说误差平方和）和更复杂评分函数之间进行一些平衡。比较简单评分函数通常更便于计算并且更容易定义。然而，比较复杂的评分函数（例如刚才提到的那些评分函数）可能会更好地反映预测问题的实际情况。非常重要的一点是许多数据挖掘算法（比如树模型、线性回归模型等等）原则上可以使用通用的评分函数——举例来说，基于交叉验证的算法可以使用任何定义完备的评分函数。当然在实践中并不是所有软件包都允许数据挖掘者自己定义面向应用的评分函数，尽管在理论上是可以这样做的。

7.3.2 评价描述模型的评分函数

对于描述模型来说不存在任何要预测的“目标”变量，所以如何定义评分函数不像预测建模中那样明确。一种基本的方法是通过似然函数，在第 4 章中我们曾介绍过似然函数，不过在这里我们是从一个稍微不同的角度再对其进行描述。令 $\hat{p}(\mathbf{x};\theta)$为对观察数据点 $\mathbf{x}$ 的估计概率，和模型 $\hat{p}$ 取参数值 θ 时所定义的相同，其中 X 是范畴型变量（扩展到连续变量的情况是非常容易的，只要把 $\hat{p}$ 换为概率密度函数）。如果一个模型很好，那么它应该对观察到数据点的那些 X 值给出较高的概率。因此可以把函数 $\hat{p}(\mathbf{x})$看作评价模型在观察点 $\mathbf{x}$ 处质量的尺度——也就是评分函数。这正是最大似然（第 4 章）的基本思想，再强调一遍就是：更好的模型应该赋给观测到的数据更高的概率。（实际上，只有当我们所考虑的所有模型都具有相同的函数复杂度时，这种比较才是“公平的”——本章的后面将讨论如何比较具有不同函数复杂度的模型）。

如果假定数据点是独立产生的，那么我们把每个独立数据点的评分函数组合起来得到总的评分函数，组合方法就是把它们乘到一起：

$$L(\theta)=\prod_{i=1}^{n}\hat{p}(\mathbf{x}(i);\theta) \tag{7.3}$$

这就是第 4 章中所介绍的似然函数，对于一个数据点集合，我们使该函数最大化以求出 θ 的估计值。正如在第 4 章中所指出的，对数似然（log-likelihood）通常使用起来更加方便。那么现在每个数据点对总的评分函数的贡献就是 $\log\hat{p}(\mathbf{x}(i);\theta)$，总的评分函数就是这些贡献的和： 217

$$\log L(\theta)=\sum_{i=1}^{n}\log\hat{p}(\mathbf{x}(i);\theta) \tag{7.4}$$

很多时候我们取 $\log\hat{p}(\mathbf{x}(i);\theta)$的负数，那么就只需最小化这个评分函数。因此我们定义：

$$S_L(\theta)=-\log L(\theta)=-\sum_{i=1}^{n}\log\hat{p}(\mathbf{x}(i);\theta) \tag{7.5}$$

对此评分函数的直观解释是：$-\log\hat{p}$ 是误差项（它随 $\hat{p}$ 变小而变大），然后对所有数据点的这一误差进行汇总。$\hat{p}$ 的最大可能取值是 1（对于范畴型数据），对应于 $S_L(\theta)$ 的最小值 0。因此，我们可以把 $S_L(\theta)$ 看成是一种熵，它衡量了参数 θ 压缩（或预测）训练数据的好坏程度。

似然（或者负的对数似然也完全等价）的一个特别有用特性就是它非常通用。它适用于模型或模式被表示为概率函数的所有问题中。例如，假定在某个预测模型中 Y 是某个预测变量 X 以及额外随机分布误差的理想线性函数（和我们上一节所讨论的一样）。如果我们能够找到用于描述这些误差概率分布的参数形式，那么就能够相对模型中的参数来计算数据的似然。事实上，正如我们在第 4 章中看到的，如果误差项被假定为均值为 0 的正态分布（关于 X 的确定性函数），那么似然函数就等价于误差平方和评分函数。

尽管（负的对数）似然是一种强有力的评分函数，但也有局限性。特别是当确定参数时如果赋给某些数据点的概率接近于 0，那么负的对数似然将趋向于负无穷大。因此，总的误差会被部分极端数据点所支配。如果同一个数据点的实际概率也非常小，那么模型将会对密度函数末端的预测（可能性非常小的事件）给予惩罚。这可能对模型的实际效果影响很小。反过来看，这样做可能会产生某些问题（例如要预测稀有事件的发生情况），有可能我们非
218 常感兴趣的预测就位于密度函数的末端。因此，尽管似然函数是基于较强的理论基础并且对于评价概率模型一般都是适用的，但是要认识到它并非一定能反映出模型在特定任务下的实际效果，这一点是非常重要的。其他用于判断概率模型预测质量的评分函数还有很多，各有特色。举例来说，我们可以定义估计概率 $\hat{p}(\mathbf{x};\theta)$ 和实际概率 $p(\mathbf{x})$ 间的误差平方的积分，即 $\int(\hat{p}(\mathbf{x};\theta)-p(\mathbf{x}))^2 d\mathbf{x}$。把平方展开，并忽略不依赖于 θ 的项，便得到了一个形式为 $\int \hat{p}(\mathbf{x};\theta)^2 d\mathbf{x} - 2E[\hat{p}(\mathbf{x};\theta)]$ 的评分函数，可以根据试验来近似其中的每一项以估计出关于 θ 的误差平方函数的真实积分。

对于非概率性描述模型（比如说基于分割聚类）可以相当容易地为其找到各种各样的评分函数，比如基于各个聚类的分割程度、紧缩程度等等。举例来说，对于简单的基于原型聚类（在第 9 章中讨论的 k-均值方法），一种简单而且应用很广的评分函数就是对每个聚类内误差平方进行汇总：

$$S_{KSSE}(\theta)=\sum_{k=1}^{K} e_k, e_k = \sum_{i \in cluster_k} \| \mathbf{x}(i)-\mu_k \|^2 \tag{7.6}$$

其中 θ 是聚类模型的参数向量，$\theta=\{\mu_1, \cdots, \mu_K\}$；$\mu_k$ 是聚类的中心。然而，要使评分函数正式地反映各个聚类与“真实”情况（如果这样比较是有意义的）的接近程度是相当困难的。对一种聚类效果的最终裁判依赖于这种聚类的具体应用环境。看它是否从新的角度揭示了数据的内幕？是否可以产生可解释的数据分类？等等。通常仅能够针对特定问题的上下文来回答这些问题，无法用单一的评价标准来表征。换言之，用于像聚类这种任务的评分函数不一定与用于该问题的真实工作函数密切相关。我们将在第 9 章中回来讨论关于聚类任务的评分函数问题。

概括来说，对于诸如分类、回归以及密度估计等任务都有一些简单的“通用”评分函数，并各有特色适用于不同的情况。然而，每一种评分函数都有其局限性，最好是把这些评分函
219 数作为基础然后根据具体应用设计出更合适的评分函数。

7.4　评价不同复杂度的模型

在前面的章节中我们将评分函数描述为衡量观测到数据与提出模型之间差异的某个尺

度。有人可能认为接近实际数据的模型（从评分函数的角度来看）便是“好的”模型。但是还需要看建模的目的。

7.4.1 模型比较的一般概念

我们可以区分两种情况（像我们在前面章节的讨论那样）。一种情况是我们只希望构建一个对数据集合进行概要描述的模型，用来捕捉数据的主要特征。举例来说，我们可能想从一个特定的化合物系列中概要地提取这一家族中的主要化合物，在我们的数据库中包含了这一家族的所有可能成员记录。在这种情况下，模型的精确度是极为重要的——尽管模型的精确度可以通过综合性的考察来调整。可以准确地再生数据的模型，或者以某种等价形式描述了数据的模型的精确度最高。但是这种情况下，建立模型的全部目的就是要降低数据的复杂度，得到某种更易于理解的形式。在像这样的情况下，模型对数据的拟合程度是整个评价尺度的一个部分，另一部分就是模型的易理解程度（comprehensibility）（而且这一部分是主观的）。这一背景下的一种通用技术是以数据压缩和信息理论为基础的，在这种方法中评分函数通常被分解为：

$S_I(\theta, M)$ = 通过给定模型描述数据所需的二进制位数＋描述模型（和参数）的二进制位数

其中第一项衡量了对数据的拟合度，第二项衡量了模型 M 和它的参数的复杂度。实际上，可以使用 S_L=-log $p(D \mid \theta, M)$（负的对数似然函数，底数是 2）作为第一项（“通过给定模型描述数据所需的二进制位数”）。使用-log $p(\theta,M)$（这实质上相当于对第 4 章中讨论的普通贝叶斯评分函数取负的对数）作为第二项（“描述模型（和参数）的字节数”）。直观地讲，我们可以把-log $p(\theta,M)$（第二项参数）看作是把模型结构从某个假设的发送程序以二进制位为单位传送到另一个假设的接收程序所花费的传输“代价”，而把 S_L（第一项参数）看成是 220
传输模型和参数中没有说明的那部分数据（误差）所花费的“代价”。通常这两部分的变化方向是相反的——复杂的模型可以很好地拟合数据，而简单的模型更易于理解。总的评分函数对这两者进行折衷得到可接受的模型。

另外一种一般的情况是，我们的实际目的是从现有数据泛化到可能出现的新数据。例如，我们可能希望推理出新顾客的可能行为或推断还没有观察到的天体的可能属性。因此要再次重申，尽管对观测到数据的拟合度显然是一个好模型的必要条件，但不是全部。特别是因为数据没有代表整个总体（如果代表了的话，那么就不需要泛化了），所以观测到数据的某些特征（“噪声”）并不是整个总体的属性，反之亦然。一个非常好的拟合观测到数据的模型也会拟合这些特征——因此不会提供最好的预测。因此，我们需要修改简单的拟合度尺度以定义一个全面的评分函数。特别地，我们需要加入一个部分来防止模型变得太复杂，避免拟合观察数据的所有特异性。

无论是对于两种情况中的哪一种，理想的评分函数都是在很好的拟合数据和模型的简洁性间达到某种折衷，只不过是实现折衷的理论根据有所不同。这种不同可能意味着不同的评分函数适合于的不同情况。因为当我们的目标就是概括数据集合的主要特性时，那么这种折衷必然包含一定的主观成分（“数据挖掘者认为什么样的模型是可接受的简单模型？”），在这里，我们将集中关注另一种情况：我们的目的是根据现有的数据决定哪一个模型对于未见过的数据会有最好的性能。

7.4.2 再谈偏差-方差

在分析可以评估模型对未见过数据的拟合程度的评分函数前，我们先讨论一下为什么必须避免与现有数据拟合得太近。我们在第 4 章中介绍参数 θ 的估计时讨论了偏差和方差，在这里我们从评分函数的角度再讨论一下这个问题。

221 正如我们在前面章节中所指出的，选择出完全“正确”的模型结构是绝对不可能的。因为现实世界中的很多特征是无法用模型所精确描述的（而且“正确”的含义是什么，这里面也有很多深层的问题）。这意味着选取的模型形式仅仅提供了一种对“真实情况”的近似。就拿预测模型来说。对于 X 的任一个给定值（假设它是一元的以使描述简单——同样的结论也完全适用于 X 为多元的情况），模型给出的 Y 预测值可能不是精确的。更严格地说，假定我们抽取了许多不同的数据集，然后把一个具有指定结构的模型（比如一个分段局部模型，模型的分量数目是确定的，每个分量的复杂度也是给定的；或者一个给定了次数的关于 X 的多项式函数，等等）拟合到这些数据集中的每一个，然后计算对于任意 X 值 Y 预测值的期望。那么这个期望的预测值不可能与实际值完全一致。也就是说，对于一个给定的 X 值，模型可能提供一个对真实 Y 值的有偏预测。（回忆在第 4 章中我们把估计的偏差定义为估计值（预测值）和真实值间的差异）。因此，完美的预测是一种无法实现的奢望！

不过，我们可以通过提高模型结构的复杂度使预测的期望值与未知真实值之间的差异更小（事实上，对于某些情况和某些种类的模型，我们可以使这个差值任意的小）。在上面的例子中，这意味着在分段线性模型中增加分量的数目，或者提高多项式的次数。

乍一看，这岂不是很好——只要使用一种足够复杂的模型结构，那么就可以得到任意精确的模型（以偏差衡量）。不幸的是，没有这样的免费午餐，精度在偏差方面的提高是以损失其他性能为代价的。

由于模型结构的极度灵活性，对于任意的固定 X 值，模型的预测可能会因数据集的不同而有很大的差异。也就是说，尽管对于给定值 X 所获得 Y 预测值的平均值非常接近 Y 的真实值（这就是较小偏差的含义），但是从不同数据集获得的预测值之间会有很大的差异。因为在实践中，我们总是仅仅观察到这些预测值中的一个（我们实际上仅有一个数据集用来估计模型的参数），所以“平均”效果好提供的帮助很少。尽管我们知道了我们选择的是一个产生的预测值与平均值相差很大的数据集。可是又有何用呢？

还可以用另一种方式来观察这个问题。我们这种非常灵活的模型（例如，大量的分段分
222 量或很高的次数）会非常紧密地跟随数据。因为，对任一给定的 X，Y 的观察值是关于它的均值随机分布的，所以我们的灵活模型也会把 Y 的观察值中的随机分量模型化。也就是说，这种灵活模型过度拟合了数据。

最后（不过这实际上又是以另一种方式来观察同一个问题），增加模型结构的复杂度意味着增加了要估计的参数的数量。通常来说，如果要估计更多的参数，那么每个估计的准确性就会下降（它相对不同数据集的方差会增大）。

以上偏差和方差的互补性被称为*偏差-方差平衡*（bias-variance trade-off）。我们希望选取的模型方差和偏差都不要太大——但是降低二者中的任一个往往会增大另一个。可以把它们组合起来以得到一个总的数据和模型之间差异尺度，这便是*均方误差*（mean squared error，

MSE）。考虑我们前面所讨论的标准回归，在那里这我们假定 y 是 $\mathbf{x}$（现在我们将其推广到向量的情况）的决定性函数，并带有一个附加的噪声，也就是 $y=f(\mathbf{x};\theta)+e$，其中 e 是均值为 0 的正态分布（举例）。因此，$\mu_y=E(y|\mathbf{x})$ 代表任何一个给定数据点 x 对应的真实（未知）期望值（这里期望 E 是相对噪声 e 的），而 $\hat{y}=f(\mathbf{x};\theta)$ 是我们的模型和拟合的参数 θ所给出的预测。于是在 $\mathbf{x}$ 点的 MSE 被定义为：

$$MSE(\mathbf{x})=E[\hat{y}-\mu_y]^2$$

$$=E[\hat{y}-E(\hat{y})]^2+E[E(\hat{y})-\mu_y]^2 \qquad (7.7)$$

也就是说，MSE=方差+偏差 ^{2}。（这里的期望 E 是相对于 $p(D)$，也就是对于某个固定的容量 n 所有可能数据集的概率分布）。这个等式经得住细致地推敲。这里我们将预测 $\hat{y}$ 当作一个随机量，它的随机性是由产生训练数据 D 的随机抽样而导致的。不同数据值 D 可导致不同的模型和参数，以及不同的预期 $\hat{y}$。期望值 E 是相对具有同一数据量 n 的不同数据集的（每个都可随机从问题域中任意抽取）。方差项 $E[\hat{y}-E(\hat{y})]^2$ 告诉我们估计 $\hat{y}$ 会如何随不同的可能数据集变化。换句话说，它衡量了 $\hat{y}$ 对训练模型的特定数据集的敏感度。举一个极端的例子，如果我总是选取常数 y_1 作为预测值，而根本不考虑数据，那么这个方差则为零。而在另一个极端，如果我们有一个有许多参数的极端复杂的模型，我们的预测 $\hat{y}$ 则会随不同训练数据集的变化而剧烈变化。

偏差项 $E[(E\hat{y})-\mu_y]$反映了预测中的系统误差——也就是预测的平均值 $E(\hat{y})$和真实值之 223
间的差距。如果我们忽略所有数据，就使用常数 y_1 作为预测，那么偏差会很大（也就是说，这个差异会很大）。如果我们使用一个更复杂的模型，那么我们的平均预测就会更接近真实值，但是方差可能很大。偏差-方差矛盾量化了简单模型（方差小，但偏差可能很大）和较复杂模型（可能偏差很小，但方差通常很高）之间的这种牵制关系。

当然，在实践中我们所感兴趣的是所评估函数在整个定义域上的平均 MSE，因此我们把 MSE（相对于输入分布 $p(\mathbf{x})$）定义为 $\int MSE(\mathbf{x})\,p(\mathbf{x})d\mathbf{x}$，这种定义同样也具有相加分解性（因为期望是线性的）。

注意虽然原则上我们可以衡量预测 $\hat{y}$ 的方差（例如，使用某些二次抽样技术，比如自展法（bootstrap method）），但偏差总是未知的，因为它包含了本身未知的μ_y（要从数据中学习的）。因此，偏差-方差分解目前主要还处于理论研究阶段，因为我们无法衡量偏差部分，因此这也决定了无法给出一个可以把这两方面的估计错误结合到一起的评分函数。但是实践中的意图通常是很清楚的：我们需要的模型既不该太不灵活（以防止预测值存在内在偏差）；也不该太灵活（以防止预测值存在内在方差）。也就是说，我们需要一个这样的评分函数：它可以处理不同复杂度的模型；并且可以考虑偏差-方差之间的折衷；又可以实现。这就是下一节我们要讨论的焦点。

我们应注意到在某些数据挖掘应用中，方差的问题可能不是非常重要，尤其是当与用来拟合模型的数据量相比模型很简单的时候。这是因为方差是样本大小的函数（正如我们在第 4 章中所讨论的）。增加样本大小就会减小估计量的方差。不幸的是，并没有一条通用的法则来说明方差和过度拟合在哪些情况下很重要，哪些情况下不太重要。它既依赖于训练数据 D 的样本大小，又依赖于被拟合模型的复杂度。

7.4.3 惩罚复杂模型的评分函数

那么，如何在灵活性（以便合理地拟合现有数据）和过度拟合（模型拟合了数据中的随
224 机成分）中选择一种合适的折衷方案呢？一种方法是选择一种封装了这种折衷的评分函数。也就是说，选择一种总的评分函数，它是由两个部分组成的：一部分衡量模型对数据的拟合程度；另一部分用来鼓励简洁性。这便得到了一种如下形式的评分函数：

$$\text{score(model)} = \text{error(model)} + \text{penalty-function(model)}$$

我们的目标是最小化这一评分函数。在前面几节中我们已经讨论了几种不同的方式来定义这个评分函数中的误差部分。那么附加的惩罚部分该如何定义呢？

通常（尽管这其中存在一些简化），模型 M 的复杂性与所考虑的参数个数 d 相关的。在接下来的讨论中我们采用以下符号。设有 K 种不同的模型结构 M_1，…，M_k，我们要从中选择一个（理想状态是其中有一个可以最好地预测将来数据）。模型 M_k 有 d_k 个参数。我们假定对于每一种模型结构 M_k，$1 \leqslant k \leqslant K$，我们已选择出最佳拟合的参数 $\hat{\theta}_k$（这些参数使模型最大程度地拟合数据）；也就是说，我们已经求出了这 K 个模型结构参数的点估计，现在只是要从这些模型中选取一个。

著名的 Akaike 信息标准（简称 AIC）是这样定义的：

$$S_{AIC}(M_k) = 2S_L(\hat{\theta}_k; M_k) + 2d_k, \quad 1 \leqslant k \leqslant K \tag{7.8}$$

其中 S_L 是负对数似然，与公式 7.5 的定义相同，惩罚项为 $2d_k$。可以使用极限理论推导出这个公式。

另一种方案是基于贝叶斯理论的，不过也考虑了样本大小 n。这就是贝叶斯信息标准（简称 BIC），它被定义为：

$$S_{BIC}(M_k) = 2S_L(\hat{\theta}_k; M_k) + d_k \log n \tag{7.9}$$

其中 S_L 也是公式 7.5 中的负对数似然。值得注意的是附加惩罚项 $d_k \log n$ 的作用。对于固定的 n 值，惩罚项会随着参数数量 d_k 线性增长，这是非常直观的。对于固定的参数数量 d_k，惩罚项会与 $\log n$ 成比例增长。注意相对 n 的对数增长可能会被 S_L 中相对 n 的线性增长所掩盖（因为它是 n 项的总和）。所以，随着 n 的增大，对于较小的 d_k 值，误差项 S_L（与 n 呈线性关系）
225 会支配惩罚项（与 n 呈对数关系）。直观地讲，对于数据点数量 n 非常大的情况，我们可以“相信”训练数据中的误差，这时惩罚项就不太重要了。反过来说，对数据点数量 n 很小的情况，惩罚项 $d_k \log n$ 会在模型选择中产生较大的影响。

还有很多其他的惩罚性评分函数，它们的相加项和上面介绍的相似（也就是一个基于误差的项加一个惩罚项）。比如用于回归问题的调整了的 R^2 和 C_p 评分函数、最短描述长度法（MDL）（它与第 4 章介绍的 MAP 评分函数关系非常密切）和 Vapnik 的结构风险最小化方法（structural risk minimization，SRM）。

这些惩罚性函数中的一部分可以用比较基本的理论正式推导出来。但是，实际上这些函数经常是在比理论推导所作假定要宽松的多的条件下使用的。尽管如此，因为它们容易计算，而且对于给定的特定数据集和数据挖掘任务，它们至少会给出一种通用概念来表征模型的合

适复杂度，所以在实践中这些函数经常是非常方便的。

另一种不同的途径就是使用第 4 章介绍的贝叶斯框架。我们可以直接计算每个模型对于给定数据的后验概率，然后选择一个具有最大后验概率的；也就是，

$$\begin{aligned} p(M_k \mid D) &\propto p(D \mid M_k)p(M_k) \\ &= \int p(D,\theta_k \mid M_k)P(M_k)d\theta_k \\ &= \int p(D \mid \theta_k)p(\theta_k \mid M_k)d\theta_k\, p(M_k) \end{aligned} \tag{7.10}$$

其中的积分代表在参数空间中计算数据似然的期望（又被称为边际似然（marginal likelihood）），相对于参数空间中的先验 $p(\theta_k \mid M_k)$；$p(M_k)$ 项是每个模型的先验概率。显然，这与“点估计”方法是大不相同的——贝叶斯哲学就是要充分考虑不确定性，因而要对参数求平均（因为不能确定它们的确切值），而不是“拣”一个像 $\hat{\theta}_k$ 这样的点估计。注意这种贝叶斯方法隐含的惩罚了复杂性，因为参数空间的维度越高（模型越复杂）就意味着 $p(\theta_k \mid M_k)$ 中的概率质量分布的越稀薄（相对于更简单的模型）。

当然，在实践中对于许多参数空间和感兴趣的模型来说，直接积分经常是难以驾驭的，因此经常使用 Monte Carlo 抽样技术。进一步说，对于大的数据集，$p(D \mid \theta_k)$函数实际上在某个单一值 $\hat{\theta}_k$ 附近是非常“尖锐的”（回忆第四章中最大似然估计的例子），因此在这种情况 226
下我们可以用尖峰值再加上它附近部分（例如，$p(D \mid \theta)p(\theta)$的后验最频值附近的泰勒级数展开式——可以证明这样做就是前面 BIC 方法的近似）的值给出对上述贝叶斯表达式的合理近似。

7.4.4 使用外部验证的评分函数

有时使用一种不同的策略来选取模型，该策略并不是以增加惩罚项为基础的，而是建立在对模型的外部验证基础上的。它的基本思想就是将数据（随机地）分为两个互不重叠的部分：“设计”部分 D_d 和“验证”部分 D_v。设计部分用来构建模型和估计参数。然后使用验证部分重新计算评分函数。最后用这些验证分数来选择模型（或模式）。这里很重要的一点是，对特定模型分数的估计（比如表示为 $S(M_k)$）本身就是一个随机变量，它的随机性既来自用来训练（设计）模型的数据集又来自验证模型的数据集。举例来说，如果分数是目标值和模型预测值之间的某一误差函数（比如误差平方和），那么理想上说我们应为每一个所考虑的模型对将来数据的分数值建立一个*无偏估计*（unbiased estimate）。在验证环境中，因为两个数据集是互相独立的并随机选取的，所以对于一个给定模型验证分数提供了对模型在新数据点（“样本之外的”）上的分数值的无偏估计。也就是说，设计中不可避免的估计偏差在独立的验证估计中不会出现。由此（以及期望的线性特征）可以得出，两个模型对于验证数据集的分数差异会有利于更好的模型。因此，我们可以使用验证分数来选择模型。注意在前面我们已经讨论了参数 θ 的无偏估计（第 4 章）、预测 μ_y 的无偏估计（本章前面），现在我们又介绍了评分函数 S 的无偏估计。在这三种情况中都使用了偏差-方差原则，而且实际上这三者是相互联系的（例如参数估计的精度会影响预测的精度）——不过，重要的是理解它们之间的差异。

目前，验证的一般思想已经被扩展为*交叉验证*。也就是把分成两个独立集合的操作随机 227

重复很多次，每次根据数据的设计部分估计出符合给定形式的新模型，并根据验证部分得到对每个模型的样本外性能的无偏估计。然后对这些无偏估计进行平均得到总的估计。我们在第5章中讨论如何选择CART递归划分模型时介绍了这种交叉验证的用法。交叉验证在实践中非常流行，这主要是因为它很简单并且鲁棒性很好（从它仅依赖于相当少的假定这个意义上来说）。但是，如果重复分割 m 次，那么它确实也要付出相当代价的：它的复杂度与只使用单一验证集的方法相比是后者的 m 倍。（在一些特例中有例外。例如，在线性判别分析中使用了一种交叉验证方法的特例，它仅留下一个数据点（leaving-one-out）作为验证数据集，它的计算复杂度和基本的模型构建算法是一样的。）

对于很小的数据集，选择验证子集 D_v 的过程可能导致在不同数据集间有显著的差异，因此在实践中必须对交叉验证评分的*方差*进行监控，检查这种差异是不是不合理的过高。最后，在使用交叉验证方法对可能有不同参数但却具有相同复杂度的模型进行平均时需要特别注意。也就是必须保证我们每次确实是对同一个基本模型进行平均。举例来说，如果对于不同的训练数据子集，我们所使用的拟合过程可能陷入参数空间中的不同局部最大值，那么对这些模型的验证分数进行平均的意义就不明确了。

正如前面所指出的，根据这一个过程得到的对一个给定模型的性能估计是无偏的。这就是这种方法在性能评估中应用如此广泛并不断发展（参见补充读物）的原因。但是，也要考虑一些注意事项。如果接下来又使用这一验证尺度来选择模型（例如，在不同复杂度的模型中作出选择），那么最终选择模型的验证分数就是这个模型性能的有偏估计了。为了说明这一点，想像某一模型仅由于偶然性在验证集上表现得异常好。也就是说，这个模型恰好遇到了适合它的验证集，因此它表现很好。接下来这个模型很可能被选为“最佳”模型。但显然这个模型对样本外数据集不会表现这么好。这告诉我们，在实践中如果需要对一个模型的将来可能性能进行评价，那么必须把这种评价建立在第三个数据集上，也就是*检验集*（test set），
228 我们会在下一节中详细介绍检验集。

7.5 模型和模式的评价

一旦我们基于评分函数选择了一种模型或模式，那么我们经常希望知道（在预测情况下）此模型或模式对于新的未见过数据会表现如何。例如，使用给定训练数据建立的预测分类模型对于新的未见过的数据会有怎样的误差率？在上一节讨论选择模型的验证集方法时我们已经提到了这个问题。

值得注意的是，如果再使用那些用来选择模型或用来估计参数的相同数据来进行性能评价，那么这种评价一定是偏向乐观的。因为这个模型就是根据在这个数据集上的性能被选择出的。也就是说，这种*表面的*（apparent）或者说*重新代入*（resubstitution）的性能评价必然是偏向乐观的（因为这种评价是建立在重复使用训练数据集基础上的）。

如果我们只考虑一种模型结构，而且不使用验证方法选择模型，那么我们可使用二次抽样技术（比如验证或交叉验证）将数据分为训练集和检验集来得到对模型未来性能的无偏估计。也可以重复多次，并对结果进行平均。极端来说，检验集可以仅包括一个点，从而使整个过程重复 N 次，然后对 N 次的单一分数进行平均得到最终估计。这种留下部分数据作为独立检验集的原理已经被不断提炼，并且开发出了很多有很高技术性和复杂度的方法，特别值得注意的除了有leaving-one-out交叉验证法外，还有折叠法（jackknife）和自展（bootstrap）

法（这些方法是不同的，尽管彼此有关而且有时被混淆）。后面的补充读物中介绍了一些出版物，其中包含更多的详细内容。

以上讨论的关键一点就是，如果要得到对模型未来可能性能的无偏估计，那么就必须使用与构建和选择模型所用数据集不相关的独立数据集来评估它的性能。这一规则也适用于使用了验证数据集的情况。举例来说，假定我们通过将数据分为两个子集来从 K 个模型中选择，并且在第一个子集上拟合参数，使用基于第二个子集（验证子集）的分数来选择单一“最佳”模型。那么，因为我们将根据在这个对验证数据集上的表现来选择“最佳”模型，所以拟合了这个验证数据集特异性的模型会被选出。本质上，这相当于确认数据集已经在设计过程中使用过，因此根据这个验证数据集衡量出的性能将是过于乐观的。从中选择最终模型的 229
模型集合越大，这一问题就越严重。

例 7.1 可用通过一个假想的二分类分类问题来说明为什么模型在验证数据上会有过于乐观的性能。设想我们使用 100 个数据点的验证数据集来从 K 个模型中选择最佳的模型。我们已经使这两个类具有同样的先验概率，也就是都为 0.5，同时我们设计了一种极端的情况，特意使模型中的所有“预测”变量根本没有任何预测能力；也就是说，所有的输入变量都独立于类变量 Y。这就意味着实际上每个模型所作出的预测都是完全随机的，这样做的目的是使所有模型对于新的未见过数据的总体精度都是 0.5（尽管我们并不知道这个事实）。图 7-1 显示了按这种设计模拟出的交叉验证精度，在这个过程中我们把模型数目 K 从 1 增加到 100。当我们从数量很少（小于 10）的模型中选择时，被最佳模型正确分类的验证集数据点数所占比例非常接近 0.5，然而当 $K=15$ 时使用验证集选择的“最好”模型正确分类验证集数据点数的比例值为 0.55，当 k=30 时这个比例值为 0.61。

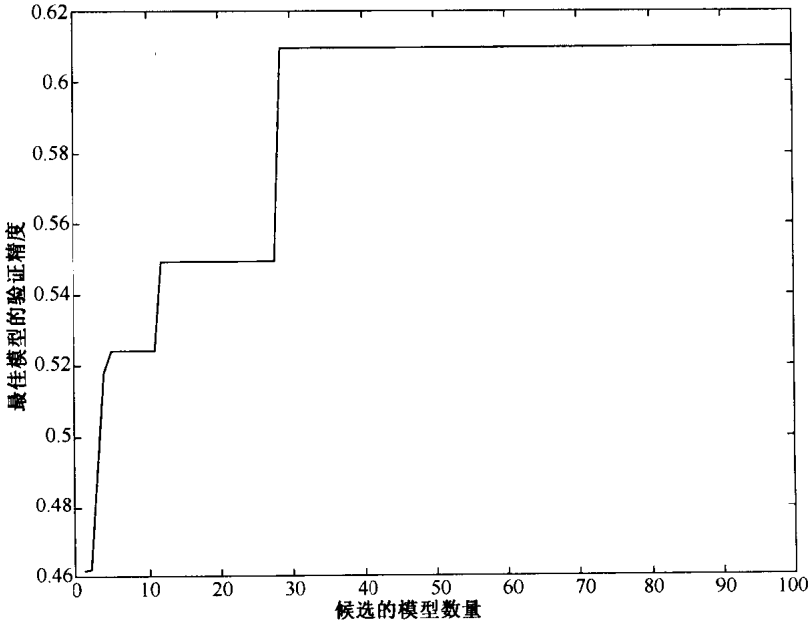


图7-1 根据验证数据集选择出的最佳模型的分类精度，横轴为候选的模型数量K，$1 \leqslant K \leqslant 100$。每个模型所做出的预测都是随机的

从这个例子可以得出一个结论：如果使用验证集来选择模型，那么就不能再使用验证集来估计模型对未来可能数据的性能。原因是：根据验证集作出的对模型在其他未见过数据上的性能估计是有偏的。正如上面所叙述的，既然在选择模型时已经使用了验证集，那么它就变成了设计过程中的一部分。这就是说，要获得对未来性能的无偏估计，我们就必须使用没有被以任何方式用于模型选择或模型估计的其他数据集（一个预先留出的（"hold-out"）数据集）。对于非常庞大的数据集来说，这通常是不成问题的，因为可用的数据是现成的；但是对于较小的数据集合便可能有问题了，因为这样会明显减少用于训练的数据。

7.6 鲁棒方法

我们在其他地方曾经指出"正确"模型的概念是经不起推敲的。相反，所有模型都是对现实情况的一种相似，我们的目的就是找到一个与当前用途足够接近的模型。从这一角度来看，如果模型没有随着它所针对数据的变化而过于剧烈变化，那么这个模型就是稳妥的（reassuring）。因此，如果某个数据点值的轻微变化导致模型的参数估计和预测值发生了根本变化，那么我们使用这样的模型就要谨慎了。换句话说，我们希望我们的模型对数据的微小变化不敏感。同理，模型和评分函数可能是基于某些假定的（比如某种潜在的概率分布）。如果轻微的放宽这些假定，拟合后的模型和它的参数以及模型的预测都没有剧烈的变化，那么这些假定就是稳妥的。

目前已经开发出了很多针对这些目标的评分函数。举例来说，在一种修整（trimmed）均值中，先抛弃很小比例的极端数据点，然后再在剩下的数据点上计算平均值。这样孤立点的值便不会影响估计结果了。随着抛弃数据点的比例越来越高，便产生了一种极端情况（假定是一元分布，并且从每个末端抛弃的数据点数是相等的），这就是中值——被公认为对于孤立点的敏感程度要比算术平均值低。另外一个例子是 Winsorized 平均值，该方法先使具有最极端值的数据点取次极端数据点的值，然后再计算正常均值。

230
~
231

尽管可以把这些修正想像为评分函数的鲁棒形式，但有时从用于计算它们的算法角度来描述它们（以及思考它们）可能更容易。

7.7 补充读物

Piatetsky-Shapiro（1991）、Silberschatz and Tuzhilin（1996）和 Bayardo and Agrawal（1999）论述了用于概率规则和模式的评分函数。

Hand（1997）对用于分类问题的评分函数进行了详细的论述。Bishop（1995）在神经网络的背景下论述了评分函数。Breiman et al.（1984）讨论了如何使用通用的误分类代价（函数）作为树分类器的评分函数。Domingos（1999）介绍了一种灵活的方法把某些作用于 0-1 分类代价假定的分类算法转换为可以使用任何分类代价矩阵的更通用算法。

Devroye（1984）讨论了使用 L1 距离尺度作为密度估计问题的评分函数，而 Silverman（1986）在同一背景下介绍了更传统的误差平方（L2）评分函数。

Geman, Bienenstock and Doursat（1992）的论文在通用的学习背景下论述了偏差和方差的关系。Friedman（1997）提出了可用于分类问题的偏差-方差分解方法，这种方法与传统的误差平方偏差-方差相比有很多根本不同的重要属性。

Linhart and Zucchini（1986）概括了各种统计模型选择技术。Ripley（1996）的第 2 章全面地分析了用于分类和回归的模型选择评分函数。Stone（1974）最先讨论了交叉验证，Hjort（1993）列举了关于交叉验证的许多最新观点和用于模型选择的有关采样技术。统计理论的书籍中（如 Lindsey, 1996）通常都包括对惩罚性模型选择的讨论，包括像 AIC 和 BIC 这样的尺度。Akaike（1973）介绍了 AIC 原理，Schwarz（1978）中包含了对 BIC 的最早论述，Burnham and Anderson（1998）详细论述关于 BIC 的最新情况和相关的方法。Vapnik（1995）详细介绍了模型选择的结构风险最小化（SRM）方法，Rissanen（1987）详细地讨论了随机复杂度、最短描述长度（MDL）和相关概念。Lehmann（1986）在假设检验的框架下介绍了比较两种模型的传统统计方法。

232

Bernardo 和 Smith（1994）详细地描述了贝叶斯理论在评分函数和模型选择方面的应用。（也可参见 Dawid（1984）和 Kass and Raftery（1995））。

Ripley（1996，第 2 章）和 Hand（1997）详细地讨论了对分类和回归模型的性能评价。Salzberg（1997）和 Dietterich（1998）讨论了如何评估多个分类模型和算法性能差异的统计显著性。

Huber（1980）是关于鲁棒方法的一本重要著作。

233

第 8 章　搜索和优化方法

8.1　简介

在第 6 章中我们看到了可以用很多种*模型结构*（model structure）或*表示*（representation）来结构化的表达知识。紧接着，在第 7 章中我们讨论了依据它们拟合观察数据的程度来评价这些结构（以模型和模式的形式）的方法。这一章我们将集中讨论数据挖掘算法中用来拟合模型或模式的*计算方法*（computational method）；也就是说，在现有数据和评分函数的引导下搜索并优化参数和结构的过程。在数据挖掘、统计和机器学习算法的文献中经常低估了高效搜索和优化的重要性，但是在实践中这些方法对一个应用成功与否起着关键的作用。

首先回忆一下第 7 章中的*评分函数*（score function），通过评分函数我们可以用数字表示出我们对一个模型或模式胜过其他的青睐。例如，如果我们使用误差平方和评分函数 S_{SSE}，那么我们将优先选择具有较低 S_{SSE} 的模型，因为这个评分函数衡量了一个模型的误差（至少在训练数据上是这样的）。如果我们的算法是在搜索具有不同表示力（和不同复杂度）的多个模型，那么我们可能优先选用像 S_{BIC} 这样带有惩罚项的评分函数（参见第 7 章的讨论），目的是通过加一个与模型中参数个数相联系的惩罚项来惩罚更复杂的模型。

不论选择的评分函数 S 的具体函数形式如何，一旦确定了，那么我们的目标就是使它最优。（在这一章中，我们假定如果不特别指出，那么就是希望使评分函数最小化，而不是使它最大化）。于是，我们设 $S(\theta|D, M) = S(\theta_1, \cdots, \theta_d|D, M)$ 为评分函数。它是关于 d 维参数向量 θ 和模型结构 M（或模式结构 ρ）的标量函数，而且是以特定的观察数据集 D 为条件的。 235

本章分析一些基本的原理，用来寻找使一个一般的评分函数 S 最小化的参数值。从实践的角度来看区分以下两种情况是有用的：一种情况是讨论的参数仅取离散值（离散参数）；另一种情况是参数可以取连续的数值（连续参数），但这两种情况并没有顶层概念上的差异。

离散参数的例子包括那些索引不同模型类别的参数（比如 1 可能对应树；2 对应神经网络；3 对应多项式函数等等）和仅取整数值的参数（例如模型中包含的变量数）。第二个例子中的变量数指出了问题的规模。例如，如果我们使用基于从可能的 p 个变量中选取一个子集的回归模型。那么就存在 $K=2^p$ 个这样的子集，即使是对于 p 为中等大小的情况，这个值也已经很大。类似地，在寻找概率规则的模式问题中，我们可能从 p 个二进制变量中选取某个子集并用这些变量的合取作为规则的左侧（右侧是固定的），然后再分析这些规则。那么就有 $J=3^p$ 个可能的合取规则（每个变量取值 0、1 或根本不在合取中）。这也可能是个天文数字。显然，这些例子都是*组合优化*（combinatorial optimization）问题，对可能解的集合进行*搜索*，目的是发现一个具有最佳得分的解。

连续参数的例子包括代表分布均值的参数；或者代表聚类集合中心的参数向量。连续的参数空间使强大的微分工具有了用武之地。在某些很特别但却非常重要的情况下，可以得到闭合形式的解。然而通常这是不可能的，因此需要迭代的方法。显然，参数向量 θ 为

一维的情况是非常重要的，所以我们将首先分析这种情况。这可以让我们看到多维情况的内幕，不过我们会发现这种情况中也会有一些难以处理的问题。无论是一维还是多维的情况都会因为局部最小值的存在而变得很复杂，对应于局部最小值的参数向量虽然与其他相似向量相比有较小的目标值，但并不是真正的最小值。后面我们将探索克服这些问题的方法。

236 很多时候，对可能模型结构的集合进行搜索与优化给定模型的参数这两个问题是相互关联的；也就是说，既然任何单一的模型或模式结构通常都具有未知的参数，那么当寻找最佳的模型或模式结构时，我们也必须在搜索中为每一个考虑的结构寻找最佳的参数。例如，考虑这样一系列模型，在这些模型中我们要通过三个预测变量 x_1、x_2 和 x_3 的某个子集的简单线性组合来预测 $\hat{y}$。其中的一个模型可能是 $\hat{y}(i) = ax_1(i) + bx_2(i) + cx_3(i)$，其他的模型可能具有同样的形式，但是仅包含两个或一个预测变量。如上面所指出的，我们的搜索必须遍历变量 x_j 的所有可能子集，但对每一个子集，还必须寻找最小化评分函数的参数（对于包含所有三个变量的情况是 a、b 和 c）。

这个描述提示我们，对于上面的问题，一种可能的算法设计是在使评分函数对于模型结构最小化的循环中嵌套一个使评分函数对于参数估计最小化的循环。这是经常使用的一种做法，因为这样做很简单，不过从计算的观点来看可能不总是最高效的。

有必要尽早指出在一些数据挖掘算法中，算法的焦点是根据选取的评分函数在参数空间中寻找模型、模式或区域的集合，而不仅仅是单一的最佳模型、模式或参数向量。例如，在贝叶斯平均技术中和在搜索模式集合的应用中都是如此。通常（尽管存在例外），搜索和优化方法的一般原理是针对单一模型、模式或参数的情况表达的，因此为了表示和说明的简单我们将主要集中在寻找单一最佳模型、模式和参数向量的问题。

有时，模型空间或被搜索的参数空间中根本不存在连续的概念，第 2 节集中讨论针对这种情况的一般搜索方法。具体内容包括：通常难以穷举分析所有解的组合问题；搜索问题的状态空间表示；特殊搜索策略；以及像分支定界这样的方法，这些方法利用参数空间或评分函数的优势来减少必须明确分析的参数向量个数。第 3 节转向对连续参数空间优化方法的讨论，包括单变量和多变量的情况，以及由于限制参数的允许值范围所导致的复杂性。第 4 节描述了可以克服残缺值问题的各类方法。在很多数据挖掘问题中，数据集非常庞大，所以必
237 须避免多次遍历数据。第 5 节介绍了针对这一目标的算法。因为很多应用都涉及评分函数具有多个最小值（或最大值）的问题，所以已经开发出了随机搜索方法以提高发现全局最优值的机会。在第 6 节中我们将描述一些这样的方法。

8.2 搜索模型或模式

8.2.1 搜索背景

这一小节讨论一些有关搜索的一般问题。在很多实际的数据挖掘应用中，我们事先不知道什么样的模型结构 M 或模式结构 ρ 最适合解决我们的任务，所以我们要对一族（family）模型结构 $M = \{M_1, \cdots, M_K\}$ 或模式结构 $P = \{\rho_1, \cdots, \rho_J\}$ 进行搜索。我们前面曾给出了两个这样的例子：在线性回归问题中寻找最佳的变量子集；寻找合取规则的左侧该包含的最佳条件集合。这两个问题都可以被看作是“最佳子集”问题，都具有这样的一般特征：从 p 个

“分量（component）”（这里是 p 个变量）中可以组合产生出数量非常庞大的解方案。寻找“最佳子集”是数据挖掘中的一个普遍问题。例如，对于一般的预测分类模型（例如最近邻、朴素贝叶斯，或神经网络分类器），我们都需要寻找对于验证数据集产生最低误分类率的变量子集。

一个有关的模型搜索问题是从 p 个变量的“池”中发现最佳的树结构分类器，我们在第 5 章中曾经用到这个例子。这个问题具有更显著的组合特征。下面考虑一下对所有可能的二叉树（也就是树的内部节点有两个子节点）进行搜索的问题。假定考虑的所有树的深度为 p，并且从根节点到任何叶子节点的路径上都有 p 个变量。此外，假定任何变量都可以出现在树的任一节点上，记得分类树的每一节点都包含一个单变量测试，测试的结果定义了从这个节点应该取的分支。对于这一族树存在 p^{2^p} 种不同的树结构——也就是说，有 p^{2^p} 个不同的分类树，它们至少有一个内部节点彼此不同。实践中，可能的树结构数量事实上还会更大， 238
因为还要考虑全深（full-depth）树的不同子树。彻底无遗漏地搜索所有可能树显然是不可行的。

我们注意到从纯数学的观点来看，我们没有必要始终区分不同的模型结构，比如所有这些模型结构可以被看作一个“完全（full）”模型的特例，只要把适当的参数设为 0（或者其他与模型形式对应的常量）那么某些部分就会从模型中消失。例如，线性回归模型 $y=ax_1+b$ 是 $y = ax_1 +cx_2 + dx_3 +b$ 当 $c=d=0$ 时的特例。这样就把模型结构搜索问题简化为本章后面要讨论的参数优化问题。尽管数学上是正确的，这一观点经常不是最有价值的考虑问题方式，因为不利于突出所考虑的模型结构的重要结构信息。

在接下来的讨论中我们将经常使用模型一词来代替模型或模式以使行文简洁，但应该视为是指这两种类型的结构：搜索模型的一般原理对于搜索模式的问题也是适用的。

关于搜索，值得进一步说明的问题还有：

- 在本节的前面我们指出，从一族μ中寻找具有最优分数的模型或结构必然涉及为族内的每一模型结构 M_k 寻找最佳的参数θ_k。这意味着概念上和很多实践中都需要一个嵌套的循环搜索过程，也就是在模型结构的搜索内嵌套了对参数值的优化。
- 正如我们已经指出的，通常根本不存在评分函数在模型空间中是否为平滑函数的概念，因此很多传统的依赖于平滑性的优化技术（例如梯度下降）是不适用的。相反，我们研究的范畴是组合优化，在这一领域中问题的内在结构本来就是离散的（例如对模型结构的索引）而不是连续的函数。对于数据挖掘中的大多数组合优化问题来说，保证找到最佳解的唯一方式就是穷举遍历所有的可能解，从这个意义上来说，这些问题具有固有的难以驾驭性。
- 对于某些问题，当我们从一个模型结构转移到下一个时我们有可能不必对参数空间 239
重新进行一次完全的参数优化。例如，如果评分函数是可分解的，那么新结构的评分函数就是前一结构的评分函数和表征结构变化的项的加函数。例如，增加或删除分类树的一个内部节点仅改变了这一节点关联的子树所对应的数据点的分数。然而在很多情况下，模型结构的改变意味着旧的参数值对于新的模型不再是最优的。例如，假定我们要建立一个从 x 预测 y 的模型，根据是两个数据点（x，y）=（1，1）和（x，y）=（3，3）。我们首先试验一个非常简单的模型 $y=a$，即 y 是一个常函数（以

使我们的所有预测都相同)。那么使误差平方和 $(1-a)^2+(3-a)^2$ 最小化的 a 值为 2。现在我们试一个更复杂的模型 $y=bx+a$。这在模型中又加了一项。现在使误差平方和(这是一个标准的回归问题，尽管相当简单)最小化的 a 和 b 值分别为 0 和 1。我们看到 a 的估计依赖于模型中的其他因素。按照数据的正交性(orthogonality)，总结出模型改变而参数估计不被影响的条件是可能的。通常知道何时可以适用这一规律是有价值的，因为这样就可以开发更快的算法(举例来说，如果回归中的变量是正交的，那么我们就可以一个一个地分析这些变量)。然而，这样的情况更多地出现在事先设计好的实验中，在数据挖掘情况下的“二手”数据中出现的较少。由于这个原因，本书不再讨论这种问题。

对于线性回归，参数估计并不困难，因此为每个考虑的模型结构重新计算最优参数更直观易懂(可能多少消耗一些时间)。然而，对于像神经网络这样的复杂模型，参数优化可能既有较高的运算要求又需要小心地调整优化方法本身(在本章的后面将看到这一点)。因此，模型搜索算法的“内层循环”可能包括相当繁重的运算。一种缓解这一问题的方法是保持模型中已经存在的参数为它们的以前值，仅仅估计增加到模型中的参数的值。尽管这一策略显然不是最优的，但它平衡了仅对很少的模型估计出高度精确的参数和对远多于此的模型近似
240 估计出参数这两者间的矛盾。

- 显然对于最佳子集问题和最佳分类树问题，穷举搜索(对模型族μ中的所有候选模型计算评分函数)对于 p 取任何非平凡值的情况都是难以驾驭的，因为每种情况有 2^p 和 p^{2^p} 个模型要分析。不幸的是，这种可能模型或模式结构数量的组合爆炸在数据挖掘中是很常见的。因此，即使还没有考虑对于每一个模型要进行参数优化过程的复杂运算，仅仅枚举模型对于很大的 p 就可能变得难以驾驭。在涉及高维数据集(p 很大)的数据挖掘问题中这个问题尤其严重。
- 对于存在固有难驾驭性的问题，我们必须借助被称为*启发式搜索*(heuristic search)的技术。实验证明(或者平均来看)这些技术可以提供好的性能，但是不能保证始终得到最佳解。“*贪婪*”*启发*(greedy heuristic)(又被称为*局部改善*(local improvement))是一种更好的方法。对于模型搜索的情况，“贪婪”搜索意味着如果给定了一个当前模型 M_k，那么便寻找“邻近” M_k 的其他模型(需要定义“邻近”的含义)，并且如果确实有优于 M_k 的模型那么就选择这当中最好的(根据评分函数)。

8.2.2 数据挖掘中的状态空间搜索

描述离散空间中搜索算法的一种通用方式是逐一确定下面这些问题：

1. 状态空间表示：我们把搜索问题看作一种在离散的状态集合中的移动。对于模型搜索，每一个模型结构就是状态空间中的一个状态。把每一个状态想像为图(非常庞大)中的一个顶点有助于建立这一概念。对搜索问题的一种抽象定义是从某个特定的节点(也就是状态)(比如 M_1)开始，然后在状态空间中移动，目的是找到对应于具有最高得分的状态的那个节点。

2. 搜索算子：搜索算子对应于在搜索空间中的合法“移动”。例如，在线性回归中选择
241 模型的算子可能被定义为从当前的模型中增加一个变量或删除一个变量。可以把搜索算子看

作是状态空间中的一个有向边。也就是说，如果在一个模型结构 M_i 和另一个模型结构 M_j 之间存在一个算子，那么在图中就有一个从 M_i 到 M_j 的有向边。

下面举一个简单的例子以帮助我们理解这个概念。考虑为一个特定的分类模型（例如，最近邻模型）从 p 个变量中选择最佳子集的一般性问题。设评分函数就是特定子集的交叉验证分类精度。令 M_k 表示我们考虑的模型族（也就是包含 $K=2^p-1$ 个不同子集（每个子集至少包含一个变量）的所有模型）内的一个模型结构个体。因此，这个状态空间有 2^p-1 个状态，从仅包含单一变量的模型子集 $M_1=\{x_1\}$，$M_2=\{x_2\}$，… 到包含所有 p 个变量的完全模型 $M_K=\{x_1, \cdots, x_p\}$。接下来定义搜索算子。对于子集选择问题经常考虑简单的算子，比如一次增加一个变量或一次删除一个变量。因此，在模型族中任何具有 p' 个变量的状态（模型结构）上有两个移动“方向”：加一个变量移动到具有 $p'+1$ 个变量的状态，或删除一个变量移动到具有 $p'-1$ 个变量的状态（图 8-1 显示了四个变量的子集选择问题的状态空间）。我们可以很容易地把这两个算子推广到每次增加或删除 r 个变量。这种“贪婪的局部”启发方法被嵌入到很多数据挖掘算法中。可以根据起始状态有所不同把使用这一思想的搜索算法分为以下两种：前向选择（forward selection）算法从最小容量的模型开始向前工作不断增加变量，而后向选择（backward selection）算法从全模型开始以相反的方式工作。在实践中当 p 很大时前向选择经常是唯一的可驾驭方法，因为反向工作在计算方面可能是不可行的。 242

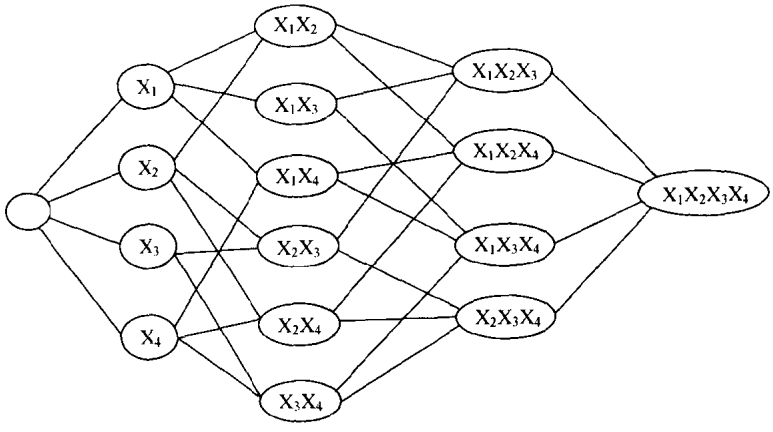


图8-1 状态空间的一个简单例子。这里包含了四个变量X_1, X_2, X_3, X_4。最左边的节点是个空集——也就是说在这个模型或模式中没有任何变量

有必要指出通过有限连接把我们的问题转化为状态空间表示并没有改变一般模型搜索问题的内在难驾驭性。为了发现最佳的状态，仍需要访问总数量达指数级的大量状态。状态空间/算子表示的作用是定义了一种对状态空间进行局部探索（local exploration）的系统方法，其中“局部”一词是从探索的是状态空间的邻近状态这个角度来说的（也就是有算子和它们相连的那些状态）。

8.2.3 简单贪婪搜索算法

可以像下面这样定义一种通用的贪婪搜索算法：

1. **初始化**：选取一个初始状态 $M^{(0)}$，对应于一种特定模型结构 M_k。

2. 迭代：设 $M^{(i)}$ 为第 i 次循环时的当前模型结构，使用评分函数评估所有的可能邻近（按照算子的定义）状态并转移到最好的一个。注意这个评估过程可能为每一个邻近的模型结构进行参数估计（或根据评分函数进行调整）。必须演算评分函数的次数就是可以应用到当前状态的算子的数量。因此，在使用的算子数量和选取状态空间内下一个状态所需的时间之间存在一个折衷问题。

3. 停止判据：重复第 2 步直到再也无法进一步改善局部评分函数的结果（也就是，遇
243 到了状态空间中的局部最小值）。

4. 多次重新启动：（这一步是可选的）使用不同的初始起始点重复步骤 1 到 3 并选取其中的最佳结果。

这个通用的算法与我们在本章后面要讨论的用来优化参数的局部搜索方法实质上是很相似的。注意在第 2 步中，我们必须显式地评估出移动到离散空间中邻近模型结构的效果；与此不同，对于连续空间中的参数优化问题，我们经常能够使用显式的梯度信息来决定移动的方向。第 4 步[○] 有助于避免在局部最小值处结束，而没有得到全局最小值（不过这不保证一定得到全局的最小值，后面我们还会讨论这个问题）。对于很多结构搜索问题，“贪婪”搜索被证明并非是最优的。然而，通常它是一种有用的启发式方法（对于很多问题，它找到的解平均看来是非常好的），并且当从随机选取的初始状态多次重复时，这种方法的简洁性使它对很多实际的数据挖掘应用都很有价值。

8.2.4 系统搜索和搜索启示

上面描述的通用算法经常被称为“爬山”算法，因为（当目标是最大化函数时）它仅沿着状态空间中的单一“路径”寻找评分函数的最大值。一种更通用（但也更加复杂）的方法是同时跟踪多个模型，而不是单一的当前模型。理解这种方法的一种简单方式是想像一个*搜索树*—— 一种动态建立的数据结构，当我们搜索状态空间时，使用这一结构来跟踪我们已经访问和评估过的状态。（当然这与分类树根本无关。）搜索树并不与状态空间等价；相反，它是描述特定搜索算法如何在状态空中移动的一种表示。

举一个例子会有助于阐明搜索树的思想。再次考虑寻找供分类模型使用的最佳变量子集的问题。我们从根本不包含任何变量的“模型”开始，对于这个初始模型，训练数据中最可能类的值就是对所有数据点的预测值。这就是搜索树的*根节点*。假定我们使用
244 仅允许每次加入变量的前向选择算子。在根节点，有 p 个变量可以被加入到没有变量的模型，而且我们可以把这 p 个新的模型表示为原始根节点的 p 个子节点。依此类推，我们可以为这 p 个节点的每一个加入 p 个变量，即为每一个创建 p 个子节点，即总共有 p^2 个（显然，$p^2-\binom{p}{2}$ 个是冗余的，实践中我们需要实现一种重复状态探测方法来从树上删除冗余的节点）。

图 8-2 显示了搜索树的一个简单实例，它针对的是图 8-1 中的状态空间。这里根节点包含空的集合（没有变量）并且在搜索的任何阶段仅考虑两个最佳的状态。这个搜索算法已经发现（截止图中所示状态）的两个最佳状态（由评分函数所决定的）为 $\{X_2\}$ 和 $\{X_1, X_3, X_4\}$。

○ 译注：原书此处为第 3 步，已确认有误。

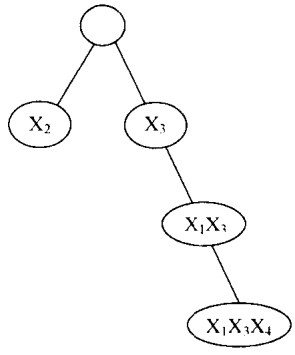


图8-2 针对图8-1中状态空间的简单搜索树示例

随着我们对状态空间的搜索，搜索树也不断地演进，我们可以把跟踪的所有叶子结点（模型结构）想像（假设）为供选择的候选模型。因为在树的第 k 层（根节点的深度为 0，分支因子是 p）有 p^k 个叶子结点需要跟踪，所以这种方法很快就进行不下去了。使用这种蛮力搜索方法（实质是对搜索树的*广度优先*（breadth-first）搜索），我们会很快用光内存。另一种存储效率很高的方法是*深度优先*（depth-first）搜索，正如名字所暗示的，这种方法先探索搜索树到某个最大深度，然后再折回以递归的方式对下一个分支重复这种深度优先搜索。

这两种技术都是*盲目搜索*（blind search）的例子，因为在这些方法中它们仅机械地排序要探索的节点，而不是利用评分函数。通常，先探索更有希望的节点可以提高性能（更快地找到更高质量的模型）。在搜索树中这意味着首先考虑具有最高分数的叶子节点的子节点；在子节点被加入作为叶子后，再分析具有最高分数的新的叶子节点。但是这种策略也会很快就产生过多的模型结构（树上的节点），超出内存的存储能力。因此，必须采取相应的策略。例如，可以实现一种*束状搜索*（beam search）[○]，使用大小为 b 的*束宽*在搜索的任意点仅跟踪 b 个最佳的模型（等价于仅跟踪树上的 b 个最佳叶子）。（在图 8-2 中 $b=2$。）如果发现最佳模型的唯一方式恰恰是先要考虑不最优的模型（因此，可能在束外），那么这种方法自然也不是最优的。然而，一般情况下束状搜索是非常高效的。毫无疑问，很多时候它都比简单的爬山算法更高效，后者类似于以它探索搜索树的方式进行深度优先搜索：在任何迭代步中仅考虑一个模型，并且选取下一个模型作为当前模型的最高分子节点。 245

8.2.5 分支定界法

在实践中一种很有用的方法是*分支定界*（branch-and-bound）。它的基本思想非常简单。在探索搜索树时，我们一直记录着到目前为止被评估出的最佳模型结构，于是对搜索树的一个特定分支（还没有探索过的）我们可以分析计算出一个最佳可能分数的下界。如果这个边界大于当前最佳模型的分数，那么我们就不必搜索这个子树了，可以把它剪除。例如，考虑从 p 个变量的集合中寻找用于分类的 k 个变量的最佳子集的问题，我们使用在训练集合上的分类误差率作为评分函数。定义一个树，它的根节点是所有 p 个变量的集合，根节点的紧邻

○ 译注：即限定范围搜索。

子节点是删除了一个变量的节点（每个有 p-1 个变量），下一层删除两个变量（有$\binom{p}{2}$个这样的唯一节点，每一个有 p-2 个变量），依此类推到$\binom{p}{k}$个叶子，每一个是包含 k 个变量的子集（这些就是候选答案）。注意训练集误差率不会随着我们沿树枝向下而降低，因为越下面的
246 节点基于的变量数越少。

下面我们按深度优先的方式探索这棵树。在深度优先算法已经下降到一或多个叶子节点后，我们会计算出模型（叶子）所对应的 k 个变量集合的分数。显然其中分数最低的是目前最佳的候选 k-变量模型。现在假定，再沿树的其他分支向下探索使我们遇到了一个分数大于目前最佳 k-变量节点的分数。既然分数不会随着我们沿这个分支继续向下探索而降低，那么就没有必要沿这一分支继续寻找了：这个分支的更低节点不可能具有比我们已经发现的最佳 k-变量解更低的分类误差率。因此我们可以不必继续沿这一分支评估下面的节点。相反，我们向上返回到最近的包含未探索过分支的节点，并开始分析这个节点。可以通过对搜索树排序来改进这一基本思想，以便先探索最有希望的节点（“最有希望的”节点是指最可能在训练集上产生最低误差率的节点）。这可以使修剪更高效。这种基本的分支定界策略可以大大地提高模型搜索的运算效率。（尽管它当然不是万无一失的解决方案——对于很多问题来说，这一策略过于庞大以至于无法在合理的时间内得出答案。）

以上给出了一些非常通用的用于搜索模型结构的思想。对于具体的模型结构和评分函数通常可以设计出更高效率的算法。尽管如此，像迭代局部改善、束状搜索和分支定界这样的一般原理是很有实践价值的，这些思想经常以各种形式出现在众多数据挖掘算法之中。

8.3 参数优化方法

8.3.1 参数优化：背景

设 $S(\theta) = S(\theta|D, M)$为我们要优化的评分函数，$\theta$ 是模型的参数。为了简单起见，我们通常不考虑对 D 和 M 的显式依赖。现在我们假定模型 M 是固定的（也就是说，在参数估计的内层循环中暂时这样，外层循环是对多个模型结构的）。我们又再次假定，我们的目标是最小化 S，而不是最大化它。注意如果 g 是 S 的单调函数（比如 log S），那么 S 和 $g(S)$ 会在同一个θ 值处最小化。

247 一般来说θ 是 d 维的参数向量。例如，在回归模型中θ 是系数和截距的集合。在树模型中θ 是分割内部节点的阈值。在人工神经网络模型中，θ 是网络中的权。

在我们要考虑的很多更加灵活的模型（神经网络是一个很好的例子）中，参数向量的维度会非常迅速地增长。例如，一个有 10 个输入、10 个隐藏单元和 1 个输出的神经网络可能有 10×10+10=110 个参数。这给我们的优化问题一个暗示：在这种情况下我们要在 110 维的空间中寻找一个非线性函数的最小值。

而且，这个高维函数的形状可能相当复杂。例如，除了结构特别简单的问题外，S 总是多峰的（multimodal）。还有，既然 S=S（$\theta|D, M$）是对观察数据 D 的函数，那么对于任意给定问题 S 的精确结构是依赖数据的。于是这意味着对于不同的数据集 D 我们要优化一个完全不同的函数 S，以至于要作出一般情况下有多少个局部最小值的结论也是困难的。

正如第 7 章所讨论的，某些情况下（例如，当训练数据点被假定为彼此独立时）可以把很多常用的评分函数写为局部误差函数和的形式：

$$S(\theta)=\sum_{i=1}^{N} e\left(y(i),\hat{y}_{\theta}(i)\right) \tag{8.1}$$

其中 $\hat{y}_{\theta}(i)$ 是我们的模型对训练数据中目标值 $y(i)$ 的估计，e 是一个衡量模型的预测和目标间距离的误差函数（比如误差平方或对数似然）。注意 S 的函数（关于 θ 的函数）形式可能通过以下两个因素的任一个而变得复杂：正在使用的模型结构的复杂性（也就是 $\hat{y}$ 的函数形式）；误差函数 e 的形式。例如，如果 $\hat{y}$ 是关于 θ 的线性函数，而且 e 被定义为误差平方，那么 S 是 θ 的二次函数，因为二次函数仅有唯一的（全局）最小值或最大值，所以这会使优化问题相对直观。然而，如果 $\hat{y}$ 是通过一个更加复杂的模型产生的，或者 e 是是关于 θ 的更复杂的函数，那么 S 就未必是关于 θ 的简单平滑函数，也未必具有唯一的易发现的极值。一 248
般来讲，求解使 $S(\theta)$ 最小化的参数 θ 的问题等价于在高维空间中最小化一个复杂函数的问题。

我们不妨这样定义 S 的梯度函数：

$$g(\theta)=\nabla_{\theta}S(\theta)=\left(\frac{\partial S(\theta)}{\partial\theta_1},\frac{\partial S(\theta)}{\partial\theta_2},\cdots,\frac{\partial S(\theta)}{\partial\theta_d}\right) \tag{8.2}$$

这是 d 维的 S 对 θ 偏导数向量。通常 $\nabla_{\theta}S(\theta)=0$ 是 S 在 θ 处取极值（比如最小值）的必要条件。这是关于 d 个变量的 d 个方程的联立方程组（即每个偏导数对应于一个方程）。因此，我们可以对这 d 个方程的解 θ（对应于 $S(\theta)$ 的极值）进行搜索。

我们可以把参数优化问题分成两种类型：

1. 一种是我们可以以闭合形式（closed form）求解的最小化问题。例如，如果 $S(\theta)$ 是 θ 的二次函数，那么梯度 $g(\theta)$ 是 θ 的线性函数，于是 $\nabla S(\theta)=0$ 的解就包含了 d 个线性方程的解。然而，在实际的数据挖掘问题中这种情况是很少见的。
2. 第二种是一般的情况，$S(\theta)$ 是 θ 的平滑非线性函数，由 $g(\theta)=0$ 得到的 d 个方程没有闭合形式的解。对于这种类型的问题，通常我们要使用迭代提高的搜索技术，利用关于 S 曲率的局部信息来引导在 S 曲面上的局部搜索。这本质上就是“爬山”（hill-climbing）或下降方法（例如最陡峭下降）。用于训练神经网络的反向传播技术就是这种最陡峭下降算法的一个例子。

因为第二种情况依赖于局部信息，所以它有可能以收敛到局部最小值而结束，没有收敛到全局最小值。因此，经常通过一种随机的成分来补充这种方法，例如，从随机选取的不同起始点启动优化过程。

8.3.2 闭合形式解和线性代数方法

考虑当 $S(\theta)$ 是 θ 的二次函数的特例。这是一种非常重要的特例，因为这时梯度 $g(\theta)$ 是 θ 的线性函数，而且 S 的最小值是 $g(\theta)=0$ 时的 d 个方程的唯一解（假定 S 在这些解处的二次导数矩阵满足正定的条件）。在第 11 章的多元回归（通常使用误差平方和函数）中我们对 249
此作了详细的阐述。第 4 章指出了如果采用似然作为评分函数也可以得到与此同样的结果，条件是假定误差服从正态分布。通常，这样的问题可以被看作解一个 $d\times d$ 矩阵的逆的问题，

所以一般可以通过 $O(nd^2+d^3)$ 衡量求解这种线性问题的复杂度，即需要 nd^2 步建立所需的原始矩阵，d^3 步来求逆。

8.3.3 优化平滑函数的基于梯度方法

当然我们通常所面对的情况是 $S(\theta)$ 并非是关于 θ 的具有单一最小值的简单函数。例如，如果我们的模型是隐单元为非线性函数的神经网络，那么 S 将是关于 θ 的相当复杂的非线性函数，具有多个局部最小值。正如我们曾指出的，很多方法是以迭代式地重复某种对模型的局部改善过程为基础的。

典型的局部改善迭代算法可以被分解为四个相当简单的部分：

1. **初始化**：为参数向量 θ 选取初始值 θ^0（经常是随机选取的）。
2. **迭代**：从 $i=0$ 开始，令

$$\theta^{i+1}=\theta^i+\lambda^i\mathbf{v}^i \tag{8.3}$$

其中 $\mathbf{v}$ 是下一步的方向（相对于参数空间中的 θ^i），λ^i 决定了要移动的距离。通常（但不是必须）$\mathbf{v}^i$ 的选择标准是使其指向改善评分函数的方向。

3. **收敛**：重复第 2 步直到 $S(\theta^i)$ 达到一个局部最小值。
4. **多次重新启动**：从不同的初始起点重复第 1 到 3 步，并选取发现的最佳最小值。

基于这一一般结构的具体方法在以下方面有所不同：选取在参数空间中的移动方向 $\mathbf{v}^i$；
250 沿选取方向移动的距离 λ^i。注意这个算法本质上与 8.2 节定义的搜索离散状态集的局部算法具有同样的设计，唯一不同的是这里我们在连续的 d 维空间中移动，而不是取图中的离散步骤。

这种算法的移动方向和距离必须由搜索当前点的局部信息来确定——例如是采集一次导数还是二次导数信息来估计 S 的曲率。然而，必须注意平衡采集信息的质量和计算这些信息所需的资源（时间、内存）这一对矛盾因素。不存在所有方面都比其他方法优秀的单一方法；每一种方法都有优点和不足。

下面要讨论的所有方法都需要确定初始点和收敛（终止）判据。这些要素的具体选择会因应用的不同而不同。此外，所有这些方法总是努力寻找 $S(\theta)$ 的*局部极值*（local extremum）。实践中我们必须检查找到的解确实是最小值（不是最大值或鞍点（saddlepoint））。还有，对于有多个最小值的非线性函数 S，无法判断局部最小值相对于全局最小值的质量，除非对整个空间进行蛮力搜索（或使用复杂的概率理论，这超出了本书的范围）。尽管存在这些限制，基于以上算法的优化技术在数据挖掘实践中是相当有用的，并且成为了很多数据挖掘算法的核心。

8.3.4 一元参数优化

先考虑一种特例：仅有一个未知的参数 θ，并希望最小化评分函数 $S(\theta)$（例如图 8-3）。尽管在数据挖掘实践中我们通常要优化的模型是多于一个参数的，但是一元的情况是相当值得重视的，因为从中可以清晰地看出与更一般的多元参数优化问题密切相关的一些基本原理。此外，一元搜索可以作为多元搜索过程的一个部分，在后一种情况中我们首先利用梯度找到搜索的方向，然后使用一元搜索决定沿这一方向移动的距离以搜索最小值。

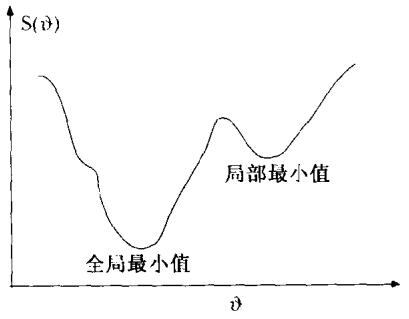


图8-3 评分函数$S(\theta)$的一个例子。$S(\theta)$是唯一的参数θ的一元函数，具有一个全局的最小值和一个局部最小值

令 $g(\theta)=S'(\theta)=\frac{\partial S(\theta)}{\partial \theta}$，那么当 $g(\theta)=0$，并且二次导数 $g'(\theta)>0$ 时，S 取最小值。如果可能有闭合形式的解，那么我们就可以发现这个解并解决这个问题。如果不然，那么我们可以使用以下方法之一。 251

Newton-Raphson 方法

假定解出现在某个未知点θ^S；也就是 $g(\theta^S)=0$。根据泰勒级数展开式，对于距离θ^S不远的点θ^*我们可以得到

$$g(\theta^S)\approx g(\theta^*)+(\theta^S-\theta^*)\,g'(\theta^*) \tag{8.4}$$

以上的线性近似忽略了$(\theta^S-\theta^*)^2$及更高次的项。既然θ^S满足 $g(\theta^S)=0$，那么这个表达式左侧等于 0。所以通过重新整理各项，我们得到

$$\theta^S\approx\theta^*-\frac{g(\theta^*)}{g'(\theta^*)} \tag{8.5}$$

换句话来说，这说明了给定一个初始值θ^*，那么按公式 8.5 调整θ^*可以得到 $g(\theta^S)=0$ 的解。通过反复迭代，在理论上我们可以无限逼近真实解。这个迭代过程就是基于一阶和二阶导数信息的一元优化 NR（Newton-Raphson）迭代更新。迭代的第 i 步可以通过下式计算

$$\theta^{i+1}=\theta^i-\frac{g(\theta^i)}{g'(\theta^i)} \tag{8.6}$$ 252

这种方法的有效性依赖于等式 8.4 的线性近似质量。如果起始值是靠近真实值θ^S的，那么我们可以期待这种近似工作得很好；也就是说，我们可以以抛物线形式局部近似 $S(\theta^*)$周围的曲面（或者等价的，导数 $g(\theta)$ 在θ^S和θ^*附近是线性的）。事实上，第 i 步的迭代误差 $e_i=|\theta^i-\theta^S|$可以被递归的写为

$$e_i\propto e_{i-1}^2 \tag{8.7}$$

从这个意义上来说，当当前的θ靠近解θ^S时，NR 方法的收敛率是二次的（quadratic）。

为了使用 NR 更新，我们必须知道导数函数和二阶导数 $g'(\theta)$ 的闭合形式。实践中，对于复杂的函数我们可能无法得到闭合形式的表达式，只能取 $g'(\theta)$和 $g(\theta)$ 的数值近似，这会给在参数空间中的移动引入更多的误差。然而，一般来讲如果我们可以精确计算闭合形式的梯度和二阶导数，那么这样做还是很有利的，因为可以在迭代优化的过程中应用这些信息决

定在参数空间中的移动。

当然，NR 方法的缺点是初始估计θ^i可能没有足够靠近真实解θ^S而使这个近似很好的工作。这种情况下，NR 步骤可能很容易地越过 S 的真正最小值或根本不收敛。

梯度下降方法

另外一种方法是仅使用梯度信息（至少提供了对于一维问题的正确移动方向）启发式的选取步长λ：

$$\theta^{i+1}=\theta^i-\lambda g(\theta^i) \tag{8.8}$$

这种方法尤其在优化的初期（远离真实解θ^S时）特别有用。这种方法的多变量版本被称为梯度（gradient）或最陡峭（steepest）下降。这里λ通常被选为相当小的值以保证我们不会在选取的方向步进得太远。我们可以把梯度下降看作 NR 方法的一个特例，通过把二阶导
253 数信息$\frac{1}{g'(\theta^i)}$替换为常数λ。

基于冲量的方法

在实践中必须折衷地选取λ。如果选得太小，那么梯度下降可能确实收敛得太慢，每一次迭代仅前进非常小的一步。另一方面，如果λ太大，那么就失去了对收敛的保证，因为可能步进得太远而越过最小值。我们可以通过增加一个冲量（momentum）项来加速梯度下降的收敛过程：

$$\theta^{i+1}=\theta^i+\Delta^i \tag{8.9}$$

其中Δ^i被递归的定义为：

$$\Delta^i=-\lambda g(\theta^i)+\mu\Delta^{i-1} \tag{8.10}$$

其中μ是一个冲量参数，$0\leqslant\mu\leqslant1$。注意当$\mu=0$时就是公式 8.8 中的标准梯度下降方法，$\mu>0$时当前的方向Δ^i还是前一次方向Δ^{i-1}的函数，从这个意义上讲增加了一个“冲量”项。Δ^i的作用是在 S 的低曲率区域加速收敛（于是改善了标准的梯度下降在这种区域非常缓慢的不足），而且幸运的是在高曲率的地方它影响很小。已经证明在实践中这种冲量启发（momentum heuristic）和有关的思想在训练像神经网络这样的模型时非常有价值。

括号法

对于有些特殊的函数（例如，如果 S 的导数是不平滑的）有一类不同的标量优化方法，这些方法根本不依赖于任何导数信息（也就是说，它直接工作在函数 S 上，而不是在它的导数 g 上）。通常这种方法是基于一种加括号（bracketing）的思想——找到一个证实包含函数极值的括号$[\theta_1,\theta_2]$。例如，如果存在一个“中间的”θ值，满足$\theta_1>\theta_m>\theta_2$并且$S(\theta_m)$比$S(\theta_1)$和$S(\theta_2)$都小，那么显然在$\theta_1$和$\theta_2$之间一定存在一个函数 S 的局部最小值（假定 S 是连续的）。我们可以使用这种思想匹配一条经过这三个点θ_1、θ_m和θ_2的抛物线，并求出$S(\theta_p)$的值，其中θ_p对应抛物线的最小值那一点。如果θ_p就是要求的最小值，那当然好，不然的话我们可以通过排除θ_1和θ_2来缩小括号。有很多不同复杂度的方法都使用了这种思想（例
254 如，一种被称为 Brent 方法的技术，它的应用很广）。从以上的说明来看括号方法明显是一种搜索策略。然而，我们在这里介绍，部分是由于这种方法在寻找最优参数值方面的重要性，也部分地由于这种方法依赖于具有连续结构（例如，序列性）的参数空间，即使被最小化的函数是不连续的。

8.3.5 多元参数优化

现在我们来研究更为复杂的问题：寻找 d 维多元参数向量 θ 的标量评分函数 S 的最小值，这也是实践中经常遇到的情况。很多方法对多元情况的处理与标量的情况是相似的。另一方面，对于我们的模型，d 可能非常大，所以多维的优化问题要明显地比相应的一元情况更为复杂。例如，在高维空间中局部最小值现象比低维空间更为普遍。此外，一种类似于（实际上是等价的）组合爆炸（我们在讨论搜索时提到过）的问题也会出现在多维优化中；这就是第 6 章中我们已经讨论过的维度效应（curse of dimensionality）。假定我们希望找到使某个评分函数最小化的 d 维参数向量，而且其中的每一个参数是定义在单位区间[0，1]中的。那么多元参数向量就是定义在 d 维单位超立方体（hypercube）中的。现在假定我们知道最优解 θ 的任何分量都不在区间[0，0.5]内。当 $d=1$ 时，这意味着已经排除了一半参数空间。然而，当 d=10 时，仅有 $\left(\frac{1}{2}\right)^{10} \approx \frac{1}{1000}$ 的参数空间被排除，如果 d=20 时，仅有 $\left(\frac{1}{2}\right)^{20} \approx \frac{1}{1\,000\,000}$ 的参数空间被排除。读者可以想像——或者亲手做一下这个算术运算——当问题中包含相当大数量的参数时会发生什么。这清楚地说明了确实存在错过全局最小值使优化结束于某个局部最小值（并非最优的）上的风险。

遵循前一小节的模式，我们先描述优化连续函数的方法（Newton-Raphson 等方法的扩展），然后描述可以应用到不连续函数的方法（与括号法相似）。

前面小节中列出的迭代方法从某个初始值开始，然后反复迭代改善。因此如果假定在第 i 步参数向量的取值为 θ^i。那么要把前面列出的方法扩展到多维情况，我们必须回答两个问题： 255

1. 我们该从 θ^i 向哪一个方向移动？
2. 我们该在这个方向上移动多远？

可以这样描述局部迭代的一般过程：

$$\theta^{i+1}=\theta^i+\lambda^i \mathbf{v}^i \tag{8.11}$$

其中，θ^i 是在第 i 步迭代时的估计参数，$\mathbf{v}$ 是指定下一步移动方向的 d 维向量（由使用的具体优化技术来确定）。

例如，对于*多元梯度下降*（multivariate gradient descent）方法上式被具体化为：

$$\theta^{i+1}=\theta^i-\lambda g(\theta^i) \tag{8.12}$$

其中 λ 是标量的*学习速率*（learning rate），$g(\theta)$ 是 d 维的梯度函数（就像公式 8.2 中定义的那样）。这种方法也被称为*最陡峭下降*（steepest descent），因为 $-g(\theta^i)$ 会指出从 θ^i 点的最陡峭倾斜方向。如果 λ 被选取的足够小，那么梯度下降保证会收敛到函数 S 的局部最小值。

神经网络中广为应用的*反向传播*（backpropagation）方法实际上就是一种最陡峭下降算法。它是比标准方法更复杂一些的，但这仅是因为网络中的多个层使上面所需的导数必须使用链式法则来推导。

注意最陡峭下降的梯度不一定直接指向最小值。因此，对于图 8-4 所示的情况，如果拘泥于仅沿梯度的方向移动那么可能是效率极差的寻找函数最小值的方式。一类更巧妙的多元优化方法使用 θ 的二阶导数信息决定在参数空间中下一步的移动方向。特别是，Newton *方法*（一元 NR 方法的多元形式）是这样定义的：

$$\theta^{i+1}=\theta^i-H^{-1}(\theta^i)\,g(\theta^i) \tag{8.13}$$

其中 $H^{-1}(\theta^i)$ 是S在θ^i这点的二阶导数的 $d\times d$ 矩阵的逆（被称为 Hessian 矩阵）。Hessian 矩阵的元素是这样定义的：

256

$$h_{lm}=\frac{\partial S(\theta)}{\partial\theta_l\partial\theta_m},\quad 1\leqslant l,\ m\leqslant d \tag{8.14}$$

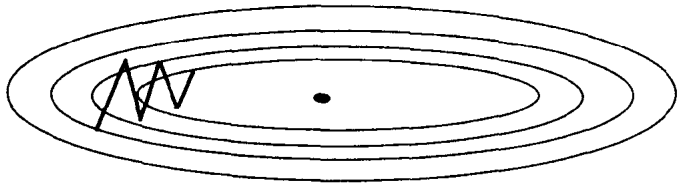


图8-4　最小化评分函数的一个例子。这个具有两个变量的评分函数的形状是一个像抛物面的“碗形”，它的最小值在中央。梯度下降没有直接指出最小值的方向，相反往往指出的是穿越碗的方向（左边的粗实线），在到达最小值之前产生一系列间接的步骤

就像一元的情况一样，如果 S 是二次的，那么根据 Newton 标准选取的步骤直接指向 S 的最小值。我们有理由希望对于很多函数来说，函数在最小值附近的形状可以被近似为θ 的二次函数（可以认为是用抛物面来近似“平滑”山峰的形状），因此，至少在最小值附近，Newton 策略可以作出关于 S 形状的正确假定。事实上，这个假定就是泰勒级数展开式的多元版本。当然，尖峰的形状通常不会恰好为抛物线形状，所以有必要递归的应用 Newton 迭代直到收敛。也与一元的情况相同，Newton 方法可能发散而不是收敛（例如，如果 Hessian 矩阵 $H(\theta^i)$ 是奇异的，那么在θ^i就不存在逆矩阵 H^{-1}）。

使用 Newton 方法是有代价的。因为 H 是 $d\times d$ 矩阵，所以每一步中估计 H 和对它求逆需要复杂度为 $O(nd^2+d^3)$ 的计算。对于参数数量很庞大的模型（比如神经网络）这种方法可能是根本不可行的。不过，我们可以通过 H 的对角线来近似 H（每一步的复杂度为 $O(nd)$）。尽管对角线近似可能是明显不正确的（因为我们可能希望参数体现出相互的依赖性），但无论如何这种近似是有价值的，因为这样只需要一个线性的开销，而不需要计算完整的 Hessian 矩阵。

另一种方法是，当在参数空间中移动时基于梯度信息迭代建立对 H^{-1} 的近似。这种技术被称为准 Newton 方法。起初我们按梯度方向运行一定步数（假定初始估计 $\hat{H}=I$，

257

是一个单位矩阵），接下来的步骤按方向$-\hat{H}_i^{-1}(\theta^i)g(\theta^i)$，其中 $\hat{H}_i^{-1}(\theta^i)$ 是第 i 次迭代时对 H^{-1} 的估计。BFGS（Broyden-Fletcher-Goldfarb-Shanno）方法是基于这一思想的一项广为应用的技术。

当然，有时对于特定的模型和评分函数会使用特定的方法。例如，第 11 章中描述了用于拟合推广线性模型的迭代加权最小二乘法（iteratively weighted least squares method）就是这样的一个例子。

我们刚刚描述的方法在每一次迭代中都要寻找这一步的“最佳”方向。另一种简单的做法是始终沿与各坐标轴平行的方向移动。这样做的不足是这种算法可能陷入粘滞状态（stuck）——例如，如果在这个方向有一条很长的狭窄凹谷。如果在最小值附近函数的形状可以被一个二次函数来近似，那么方向由这个函数的原则（principal）坐标轴定义（可能不与各坐标轴平行）。采用这个补充的坐标系统并沿新的坐标轴搜索会加快搜索的速度。事实上，如果要被最小化的函数确实是二次的，那么这个过程会正好在 d 步内发现最小值。这些新的坐标轴被称为共轭方向（conjugate direction）。

一旦我们已经决定要移动的方向，我们可以使用一个“直线搜索（line search）”过程来决定沿选取的方向要移动的距离；也就是只要应用一种上面讨论的一维方法。大多数情况下，使用一种一元方法中选取步长的快速近似方法对于多元优化问题是足够的，因为选取方向本身就是基于很多近似的。

迄今为止我们描述的方法都是基于，或者至少是从中推导出，寻找最佳步骤的局部方向，然后沿这个方向移动。*单纯形搜索方法*（simplex search method）（不要与线性规划中的*单纯形算法*相混淆）维护一个 d 维参数空间中的单纯形（一个“超四面体”（hypertetrahedron）），计算 $d+1$ 个点的函数值，并依此定义要步进的方向。为了说明这种方法，我们考虑 $d=2$ 时的情况。这时要在三个（$=d+1$，当 $d=2$ 时）点处计算函数值，这三个点被组织为一个等边三角形的顶点，也就是二维的单纯形。然后把三角形以具有最大函数值的顶点相对的一边为轴翻转。这给出一个新的顶点，然后使用这个三角形（由新的顶点和翻转中没有移动的两个顶点组成）重复前面的过程。重复整个过程直到发生了振荡（三角形仅是以同一条边为轴来回摆动）。当发生振荡时，把三角形的边长折半，然后再继续前面的过程。 258

人们已经以各种不同的方式对这种基本的单纯形搜索方法进行了扩展。例如，Nelder 和 Mead 变体不仅允许三角形缩小，而且允许增大，目的是在合适的条件下加速运动。有证据表明尽管这种方法很简单，但在高维空间中它的性能可与前面描述的复杂方法相比。此外，这种方法不需要计算导数（或者甚至不需要存在导数）。

一种有关的被称为*模式搜索*（pattern search）的搜索方法也进行一种局部搜索来决定步进的方向。如果这一步降低了评分函数，那么就增大这一步的步长。如果这一步的效果很差，那么就减小步长（直到到达最小值，搜索终止）。（这里*模式搜索*中的*模式*一词与本书前面讨论的数据挖掘中的模式无关。）

8.3.6 约束优化

很多优化问题中包含对参数的约束（constraint）。常见的例子包括参数是概率（约束参数应为整数且汇总之和为 1）的问题；或者包含方差作为参数（一定要为正数）的模型。约束经常是以不等式的形式出现的，例如 $c_1 \leqslant \theta \leqslant c_2$，其中 c_1 和 c_2 是常数；但也有更复杂的约束是以函数表示的，例如 $g(\theta_1, \cdots, \theta_d) \leqslant 0$。偶尔约束具有等式的形式。通常把满足约束的参数向量区域称为*可行区域*（feasible region）。

具有线性约束和凸评分函数的问题可以用*数学规划*（mathematical programming）的方法来解决。例如，*线性规划*（linear programming）方法已经用于有监督的分类问题；*二次规划*（quadratic programming）被用在支持向量机（support vector machine）中。评分函数和约束是非线性的问题具有更大的难度。

有时有约束的问题可以被转化成无约束的问题。例如，如果参数（$\theta_1, \cdots, \theta_d$）的可行区域被限定在正值范围内，那么我们可以对（$\phi_1, \cdots, \phi_d$）进行优化，其中 $\theta_i = \phi_i^2$，$i=1, \cdots, d$。其他的（更为复杂的）转换可以去除 $c_1 \leqslant \theta \leqslant c_2$ 形式的约束。

一种去除相等约束的基本策略是使用*拉格朗日乘子*（Lagrange multiplier）。假定评分函数 $S=S(\theta)$ 要服从约束 $h_j(\theta)=0$，$j=1, \cdots, m$，那么这个评分函数取局部最小值的一个必要条件是对于某个标量 λ_j，θ 满足 $\nabla S(\theta) + \sum_j \lambda_j \nabla h_j(\theta) = 0$。这些方程和约束得到一个具有 259

（$d+m$）个（非线性）方程的联立方程组，这个方程组可以被标准的方法求解（经常使用最小平方方法使（$d+m$）个方程的左边的平方和最小化）。可以把这些思想扩展到以 Kuhn-Tucker 条件表达的不等约束（参见补充读物）。

可以修改无约束的优化方法使其适用于有约束的情况。例如，可以向评分函数加入惩罚项以便在优化过程中抵制那些靠近可行区域边界的参数估计。

8.4　存在残缺数据时的优化：EM 算法

这一节我们考虑一类特殊但很重要的问题——在某些数据残缺的情况下最大化似然评分函数，也就是说，我们的数据集中缺少一些变量某些情况下的值。已经证明实践中相当数量的问题可以归入数据残缺问题。例如，在关于医疗患者的测量中，对于每一个患者通常仅有一部分化验结果；或者在申请表数据中，对某些问题的反应依赖于对其他问题的回答。

更一般的情况是，任何含有隐含变量（也就是，不能直接观察到的变量）的模型都可以被归纳为数据残缺的问题，在这些问题中，这个变量值对于所有 n 个对象或个体是未知的。聚类便是一个例子；我们假定存在一个离散值的隐藏变量 C，它的取值为$\{c_1, \cdots, c_k\}$，聚类的目的是估计出每一个观察值 $\mathbf{x}(i)$（$1 \leqslant i \leqslant n$）所对应的 C 值。

期望最大化（Expectation-Maximization，EM）算法是解决数据残缺问题的一种出色算法。具体来讲，令 $D = \{\mathbf{x}(1), \cdots, \mathbf{x}(n)\}$为 n 个观察到的数据向量。设 $H = \{z(1), \cdots, z(n)\}$ 表示隐藏变量 Z 的 n 个值，与观察到的数据点 D 一一对应；也就是说 $z(i)$ 与数据点 $\mathbf{x}(i)$ 相联系。我们假定 D 是离散的（这不是必要的，只是为了我们描述算法的方便），这样我们便
260 可以把未知的 $z(i)$值想像为数据的不可见分类（聚类）标签。

我们可以把观察到数据的对数似然写作

$$l(\theta) = \log p(D \mid \theta) = \log \sum_H p(D, H \mid \theta) \tag{8.15}$$

其中右侧的项表明观察到的似然可以表示为观察到数据和隐藏数据的似然对隐藏值的求和，这里假定了一个以未知参数θ 为参量的概率模型 p（D, $H \mid \theta$）。注意，我们这里的优化问题由于参数θ 和隐藏数据 H 二者都是未知的而变得加倍复杂了。

设 Q（H）为残缺数据 H 的任意概率分布。我们可以用以下方式表示似然：

$$\begin{aligned} l(\theta) &= \log \sum_H p(D, H \mid \theta) \\ &= \log \sum_H Q(H) \frac{p(D, H \mid \theta)}{Q(H)} \\ &\geqslant \sum_H Q(H) \log \frac{p(D, H \mid \theta)}{Q(H)} \\ &= \sum_H Q(H) \log p(D, H \mid \theta) + \sum_H Q(H) \log \frac{1}{Q(H)} \\ &= F(Q, \theta) \end{aligned} \tag{8.16}$$

其中的不等式是根据对数函数的凹陷性（被称为 Jensen 不等式）得出的。

函数 $F(Q, \theta)$ 是我们要最大化的函数（似然 $l(\theta)$）的下限。EM 算法在以下二者间交替：固定参数 θ，使 F 相对于分布 Q 最大化；固定分布 $Q=p(H)$，使 F 相对于参数 θ 最大化。具体地说：

$$\text{E 步骤：} Q^{k+1} = \arg\max_{Q} F(Q^k, \theta^k) \tag{8.17}$$

$$\text{M 步骤：} \theta^{k+1} = \arg\max_{\theta} F(Q^{k+1}, \theta^k) \tag{8.18}$$

可以很容易地证明在 E 步骤中当 $Q^{k+1}=p(H \mid D, \theta^k)$ 时似然达到最大值，对于很多模型可以有相当直接的方法明确地计算出 $p(H \mid D, \theta^k)$。此外，对于这个 Q 值不等式变成了一个等式 $l(\theta^k) = F(Q, \theta^k)$。

261

在 M 步骤中，最大化问题简化为最大化 F 中的第一项（因为第二项不依赖于 θ），因此我们可以得到：

$$\theta^{k+1} = \arg\max_{\theta} \sum_{H} p(H \mid D, \theta^k) \log p(D, H \mid \theta^k) \tag{8.19}$$

这个表达式也经常可以幸运的得到闭合形式的解。

显然根据定义 E 和 M 步骤在每一步中不会降低 $l(\theta)$：在 M 步骤的开始根据定义我们有 $l(\theta^k) = F(Q^{k+1}, \theta^k)$，而且以后的 M 步骤调整 θ 来使 F 最大化。

对 EM 步骤有一个简单的直观解释。在 E 步骤中，我们以参数向量 θ^k 的特定设置为条件估计隐藏变量的分布。然后，保持 Q 函数固定，在 M 步骤中我们选取一个新的参数集 θ^{k+1}，来使观察到数据的期望对数似然（相对 $Q = p(H)$ 定义的期望）最大化。反过来，我们可以在给定新的参数 θ^{k+1} 的条件下寻找新的 Q 分布，然后再一次应用 M 步骤得到 θ^{k+2}，并以这种方式迭代下去。正如上面所简要叙述的，每一次应用 E 和 M 步骤都保证不会降低观察到数据的似然，而且这反过来也意味着在相当普通的条件下参数 θ 会至少收敛到对数似然函数的局部最小值。

要确定一个精确的算法我们需要取一个初始起点（例如，从一个初始的随机选取的 Q 或 θ 值开始）和一种探测收敛的方法（例如，Q，θ，或 $l(\theta)$ 中的任一个在一次迭代后和上一次迭代后没有明显变化）。EM 算法本质上与多元参数空间中的局部爬山形式（在本章前面小节中讨论过）很相似，E 和 M 步骤隐含（而且自动的）确定每一步的方向和距离。因此，与爬山算法一样，EM 算法对初始条件是敏感的，所以选取不同的初始条件会得到不同的局部最大值。正因为此，实践中从不同的起始点多次运行 EM 算法是明智的，这样可以降低最终得到一个相当差的局部最大值的可能性。EM 算法可能相当慢的收敛到最终的参数值，所以（例如）可以把它与传统的优化技术（比如 Newton-Raphson）一起使用来加速收敛。

262

虽然如此，标准的 EM 算法因为具有宽广的适用范围和可以相当轻松地移植到各种不同的问题而被广为应用。

EM 算法的计算复杂度是由两个因素共同决定的：收敛所需迭代的次数；每一个 E 和 M 步骤的复杂度。实践中，经常发现当 EM 算法接近解时，它收敛得相当慢，不过实际的收敛速度依赖于很多不同的因素。尽管如此，至少对于简单的模型，该算法经常经过几次（比如 5 或 10）迭代就收敛到解的附近。每次迭代中 E 和 M 步骤的复杂度依赖于被匹配到数据的模型的属性（也就是似然函数 $p(D, H \mid \theta)$ 的特征）。对于很多简单的模型（比如下面讨论

的混合模型），E 和 M 步骤所需的时间关于 n 是线性的，也就是每一次迭代仅需访问每个数据点一次。

例 8.1 和例 8.2 演示了 EM 算法的应用，用来估计正态混合模型和泊松混合模型的参数，测量数据 x 是一维的。每一种情况中，假定数据来自于 K 个潜在的分量分布（分别是正态和泊松分布）。然而，没有观察到分量的标签，因此我们不知道每一个数据点来自哪一个分量分布。我们将在第 9 章中更详细地讨论如何估计这些类型的混合模型。

例 8.1　我们希望拟合一个正态混合模型

$$f(x)=\sum_{k=1}^{K}\pi_k f_k(x;\mu_k,\sigma_k) \tag{8.20}$$

其中，μ_k 是第 k 个分量的均值，σ_k 是第 k 个分量的标准差，π_k 是数据点属于分量 k 的验前概率（$\sum_K\pi_k=1$）。因此对于这个问题，参数向量为 $\theta=\{p_1,\cdots,p_k,\mu_1,\cdots,\mu_k,\sigma_1,\cdots,\sigma_k\}$。假定如果此时我们知道 θ 的值。那么，一个测量向量为 x 的对象来自第 k 个分量的概率为：

$$\hat{P}(k\mid x)=\frac{\pi_k f_k(x;\mu_k,\sigma_k)}{f(x)} \tag{8.21}$$

这就是基本的 E 步骤。

据此，我们可以根据以下各式估计 π_k，μ_k 和 σ_k：

$$\hat{\pi}_k=\frac{1}{n}\sum_{i=1}^{n}\hat{P}(k\mid x(i)) \tag{8.22}$$

263

$$\hat{\mu}_k=\frac{1}{n\hat{\pi}_k}\sum_{i=1}^{n}\hat{P}(k\mid x(i))x(i) \tag{8.23}$$

$$\hat{\sigma}_k=\frac{1}{n\hat{\pi}_k}\sum_{i=1}^{n}\hat{P}(k\mid x(i))(x(i)-\hat{\mu}_k)^2 \tag{8.24}$$

其中的汇总求和是针对数据集中的 n 个数据点的。这三个等式就是 M 步骤。这一组等式形成了一种明显的迭代关系。我们先选取 π_k，μ_k 和 σ_k 的起始值，然后代入等式 8.21 得到估计 $\hat{P}(k\mid x)$，然后在等式 8.22、8.23 和 8.24 中使用这一组估计更新 π_k，μ_k 和 σ_k，然后再返回用更新的参数进行下一轮迭代，直到收敛判据（经常是似然的收敛或模型参数达到某个稳定点）得到满足。

注意等式 8.23 和 8.24 与估计单一正态分布参数时的对应形式非常相似，唯一的差别是每一点的贡献被拆分到各个分量，拆分的比例与分量在这一点的估计大小成正比。从本质上讲，这就是根据每一个数据点属于每个分量的概率进行加权。若是我们真的知道了分类标签，那么对于数据点 $x(i)$ 所属的分量它的权就是 1，对于其他 K-1 个分量的权就是 0。

例 8.2 可以用泊松模型来对个体事件的发生率建模，例如，消费者使用电话呼叫卡的比率。对于某些卡，可能有多个个体（例如一个家庭中的成员）使用同一账号（具有卡的拷贝），因此理论上每个人应有不同的使用比例（例如，家里的孩子用得很频繁，父亲使用得不太频繁，等等）。于是对于 K 个个体，我们可用 K 个泊松分布来概括观察到的事件数据：

$$f(x)=\sum_{k=1}^{K}\pi_k\frac{(\lambda_k)^x e^{-\lambda_k}}{x!} \tag{8.25}$$

用来迭代估计的类似于例 8.1 的等式具有如下形式：

$$\hat{P}(k\mid x(i))=\frac{\pi_k P(x(i)\mid k)}{f(x(i))}=\frac{\pi_k\dfrac{(\lambda_k)^{x(i)}e^{-\lambda_k}}{x(i)!}}{f(x(i))} \tag{8.26}$$

$$\hat{\pi}_k=\frac{1}{n}\sum_{i=1}^{n}\hat{P}(k\mid x(i)) \tag{8.27}$$

264

$$\hat{\lambda}_k=\frac{1}{n\hat{\pi}_k}\sum_{i=1}^{n}\hat{P}(k\mid x(i))x(i) \tag{8.28}$$

8.5 在线和单扫描算法

到目前为止我们已经讨论的所有优化方法都隐含的假定数据全都驻留在主存储器中，因此在搜索的过程中可以很容易地多次访问数据点。对于庞大的数据集我们可能对至多仅观察数据点一次的搜索和优化算法感兴趣。我们称这样的算法为*在线*（online）或*单扫描*（single-scan）算法，毫无疑问当我们面对的是驻留在副存储器中的海量数据集时，这样的算法要优于“多次扫描（multiple-pass）”算法。

一般而言，通常可以直接对上面讨论的搜索算法进行修改，使其一个一个地处理数据点。例如，考虑优化参数的简单梯度下降方法。正如前面所讨论的，在这一算法的“离线”（或批处理）版本中，我们要先找到参数空间中的梯度函数 $g(\theta)$，然后在当前位置θ^k求出梯度的值，然后再沿这个方向移动一步。既然沿梯度 $g(\theta)$ 的方向移动仅是一种启发，因此它未必是最佳的方向。实践中，如果我们沿一个近似梯度的方向移动我们也可能达到同样的效果（至少对于运行很长时间的问题是这样的）。这个想法应用在实践中就是对梯度的*在线近似*（online approximation），即基于当前位置以及当前的和（或许）“最近的”数据点估计出梯度，并在算法中使用这个当前的最佳估计。可以把在线估计看作对批处理算法使用所有数据点产生的完全梯度估计的随机（stochastic）（或者“noisy”）估计。在统计中把有关这类搜索技术的一般理论称作*随机近似*（stochastic approximation），但这超出了本书的范围，我们主要讨论的是在线参数估计。举例来说，在使用梯度下降来寻找神经网络的权参数时，已经发现随机的在线搜索确实在实践中很有效。甚至认为搜索的随机（数据驱动）特征有时会提高解的质量，因为随机性使搜索算法以一种类似于模拟退火（simulated annealing）（后面就要

讨论）的方式逃离局部最小值。

265 更一般地讲，很复杂的搜索方法（例如基于 Hessian 矩阵的多元方法）也可以被实现为在线的方式，只要为所需要的搜索方向和步长恰当地定义在线估计量。

8.6 随机搜索和优化技术

迄今我们已经给出的模型搜索和参数优化方法都主要依赖于在当前状态附近“贪婪”选取下一步的思想。这里的主要不足就是这种方法存在固有的“近视”性。解的质量很大程度上是起始点的函数。这意味着，至少是对于单一起始位置的情况，发现的最小值（或最大值）有不是全局最优值的风险。正因为此，已经开发了很多方法，这些方法允许从当前状态以一种不确定的（随机的）方式步进很远，因此具有更大的全局性。下面每一种方法都既适用于参数优化，又适用于模型搜索问题。但为了描述的简单我们这里仅针对状态空间中的模型搜索问题进行介绍。

- **遗传搜索**：遗传算法是基于进化生物学思想的一类通用启发式搜索技术。它的核心思想是把状态（即我们所说的模型）表示为染色体（经常编码为二进制串），然后“进化”这样的染色体群体（population），方法是选择性的配对染色体以产生新的后代。染色体（状态）配对的根据是它们的“适应度（fitness）”（评分函数的分数），这样做的目的是鼓励更高适应性的染色体在一代代的更新中生存下来（在一代到下一代的更新中仅允许限制数量内的染色体生存下来）。基于这个一般特征已经开发出了很多遗传搜索方法的变体，但核心的思想都是：
  - —— 维护一个候选状态（染色体）的集合，而不是单一的状态，这样便可以使算法同时探索状态空间的不同部分。
  - —— 根据存在状态的组合来产生要探索的新状态，这样做的效果是使算法可以“跳过”状态空间的不同部分（与我们前面讨论的局部改善搜索技术形成对比）。

  266 可以把遗传算法看作启发式搜索的一个特例，所以它可能对某些问题工作得很好，而对其他的差一些。对于特定的问题它能否比更简单的方法（例如从随机点重新启动的局部迭代提高方法）有更好的性能要视情况而定。实践中这种方法的一个不足是通常必须确定很多算法参数（比如染色体的数量，如何组合染色体的说明，等等），而且可能不清楚对于给定的问题这些参数的理想设置是什么。
- **模拟退火**：就像遗传算法的动机来自进化生物学一样，模拟退火（simulated annealing）方法是受物理学中的思想启发的。这种方法的核心思想是不限制搜索算法仅能向使评分函数下降（对于我们要最小化的评分函数）的方向移动，也就是说允许（以某个概率）使评分函数朝上升的方向移动。原则上，这样做可以使搜索算法逃离局部最小值。其中的非下降移动概率在前期被设置得相当高，随着搜索的继续，这个概率逐渐降低。这种概率降低过程类似于在退火金属的物理过程中逐渐降低温度已得到金属内部的低能量状态（所以这种方法叫这个名字）。

  对于这种搜索算法来说，较高的温度对应于一个在参数空间中大幅移动的较大概率，而较低的温度对应于使函数下降的较小移动的较大概率。最终，温度调度表（temperature schedule）使温度降为 0，以便使算法仅向使评分函数下降的方向移动。因此在搜索的这一阶段，算法必然收敛到不可能再进一步下降的一点。我们希望较

早期的（随机性更大）移动就把算法带进评分函数曲面的最深“盆地”。事实上，对这种算法的一种不满是尽管可以数学证明（在相当广泛的条件下）如果使用了适当的温度安排表那么刚才的希望就会实现。但是在实践中，通常没有办法确定对任何特定问题都适用的最优温度安排表（以及如何选择非下降移动的精确细节）。因此，实践应用中的模拟退火方法常“蜕变”成了（已经是另一种）一种特殊的启发式搜索方法：具有自己的以特别方式选取算法参数。

值得注意的是，随机搜索的思想是相当广泛的，在随机搜索中，下一套参数或 267
模型是根据邻近状态质量的条件（当前状态）概率分布而随机选取的。通过以随机方式探索状态空间，原则上，算法可以把更多的时间（平均来看）用在较高质量的状态上，因而建立起关于整个状态空间的质量（或评分）函数分布模型。这种通用的方法在贝叶斯统计中非常流行，像 Monte Carlo 马尔可夫链（Monte Carlo Markov Chain，MCMC）这样的技术应用很广。可以把这些方法看作是对基本的模拟退火思想的推广，其核心思想还是起源于物理学。MCMC 的焦点是找到参数或状态空间中的分数分布，用这些参数或模型对给定数据的概率加权，而不是仅寻找单一全局最小值（或最大值）的位置。

模拟退火和遗传算法这样的方法与更简单的方法（比如带有随机重新启动的局部提高迭代方法）相比实际效果如何呢？要作出关于这一问题的一般结论是困难的，尤其是当我们想把算法所需的时间也考虑在内的时候。在比较不同的搜索算法时，不仅应该看最终解的质量，而且还应该看找到解所花的计算资源，这一点是很重要的。毕竟，如果时间是没有限制的，那么我们始终可以利用穷举方式枚举所有模型来找到全局最优解。像下面这样评价是公平的：随机搜索技术通常要包括值得考虑的额外计算和其他开销（与更简单的其他方法相比），因此，在实践中，它们往往被用在涉及相对较小数据集的特殊问题中；从计算的角度来看对于非常庞大的数据集这种方法经常是不可行的。

8.7 补充读物

Papadamitriou and Steiglitz（1982）是一本关于组合优化的经典教材。Cook et al.（1998）是关于这一主题的一本更新的权威教材。Pearl（1984）是特别针对启发式搜索这一主题的。Clark and Niblett（1989）中的 CN2 规则发现算法是束状搜索的一个例子。

Press et al.（1988）是了解数值优化技术的一个很好起点，书中不仅有一般性的介绍，还包括了一些很好的实践建议，特别是第 9 和第 10 章。Gill，Murray and Wright（1981）以及 Fletcher（1987）也是专门针对优化技术的，书中提供了大量实践建议以及具体方法的很 268
多细节。Luenberger（1984）和 Nering and Tucker（1993）讨论了线性规划和有关约束优化技术的细节。Mangasarian（1997）介绍了约束优化技术在很多数据挖掘问题中的应用，包括特征选取、聚类和鲁棒模型的选择等。Bradley，Fayyad and Mangasarian（1999）沿这一方向作了进一步的讨论。

Thisted（1988）是一本关于优化和搜索方法应用（特别是对统计问题的应用）的综合参考书，非常有价值。Lange（1999）是关于这一主题（统计优化的数值方法）的最近出版的教科书，书中包含了大量有价值的技术和研究成果。Bishop（1995，第 7 章）以神经网络的参数估计为背景广泛的讨论了优化问题，还特别说明了在线技术。

关于 EM 算法的奠基性论文是 Dempster，Laird and Rubin（1977），这一论文最早建立了这一过程的一般理论框架。在这篇论文之前关于 EM 一般概念的研究已经进行了近一个世纪，包括 Newcomb（1886）和 McKendrick（1926）。Baum and Petrie（1966）的研究是 EM 算法在隐马尔可夫模型框架下的早期发展成果。McLachlan and Krishnan（1998）全面归纳了 EM 理论和应用的很多最新成果。Meiljison（1989）介绍了加速 EM 收敛的通用技术，Lange（1995）讨论了在 EM 框架中使用梯度方法的技术。Redner and Walker（1984）讨论了在混合模型中使用 EM 方法涉及的大量计算问题。Neal and Hinton（1998）讨论了 EM 的在线版本，这对海量数据集的问题特别有价值。

理论上可以把回归问题中的在线学习看作 Robbins and Monro（1951）随机近似技术的一个特例——Bishop（1995，第 2 章）在神经网络背景下讨论了这个问题。

Mitchell（1997）对遗传算法的思想作了全面的介绍。Kirkpatrick，Gelatt and Vecchi（1983）介绍了模拟退火方法，但这种方法起源于很早的统计物理著作。Van Laarhoven and Aarts（1987）纵览了这一领域。Brooks and Morgan（1995）对模拟退火和更传统的优化技术（比如基于 Newton 的方法）进行了系统比较，还讨论了这两者混合（hybrid）的方法。他们的
269 结论是混合方法看来比单独的方法都好，不论是传统方法，还是模拟退火方法。Gilks，Richardson and Spiegelhalter（1996）收录了统计学中使用随机搜索（stochastic search，不在
270 本书范围之中）和成果（主要是以贝叶斯思想为背景）。

第9章 描述建模

9.1 简介

在前面的章节中我们解释了模型和模式这两个术语在数据挖掘中的含义。模型是一种顶层的描述，概括并描述了一个庞大数据集合的重要特征。很多情况下模型适用于测量空间中的所有点，从这个意义上来说模型是全局性的。相反，模式是一种局部描述，仅适用于测量空间中的某个子集，可能仅描述了几个点的行为或者刻画了数据中存在的异常结构。例如密度函数的最频值（波峰）或散点图中的少数孤立点。

在前面各章中我们不仅分析了模型和模式的差异，还分析了描述模型和预测模型间的差异。描述模型以方便的形式呈现数据的主要特征。它实质上是对数据的概括，使我们可以看到数据的最重要特征，不会因完整数据集的绝对容量使这些特征变得模糊不清。相对而言，预测模型的目标有所不同，其目的是使我们可以根据观察到的对象特征值来预测它的其他特征值。

本章主要讨论描述模型，介绍在数据挖掘中用来寻找描述模型的几种重要算法。第 10 和 11 章将讨论预测模型，第 13 章将讨论描述模式。

我们曾经指出数据挖掘所关心的通常是如何建立实验模型（empirical model）——这些模形不是根据数据发生机制的某些内在理论推导出的，而是对观察到数据的一种描述。数据挖掘的根本目标是探查和理解数据的内部结构，使我们可以看到它的重要特征。当然除此之外，我们希望发现未知的以及从某种意义上讲有价值的结构。一个好的模型还具有再生性（generative）——根据模型产生的数据与用以产生模型的真实数据具有相同的特征。如果这种生成的数据具有原始数据没有的特征或者不具有原始数据应有的特征（例如变量间的相关性），那么这便不是一个好的模型：它没有能充分地概括数据。

本章将集中讨论拟合描述模型的具体技术和算法。这是以前面各章介绍的很多概念为基础的，比如不确定性理论（第 4 章）；把数据挖掘算法分解成基本的组件（第 5 章）；模型结构、评分函数和搜索参数与模型的一般原理（分别是第 6、7 和 8 章）。

有很多种不同类型的描述模型，每一种以不同的方式和其他的相联系（有些模型是其他模型的特例或者推广，有些模型是以不同的角度观察同一结构，等等）。在一章中分析所有类型的模型是不可能的。因此我们仅讨论一些比较重要的模型类型，特别集中在密度估计和聚类分析上。其他的描述技术（例如 structural equation modeling 和因素分析）请读者参考相关的文献。

有一点需要说明。因为本章所关心的是全局模型——代表大多数对象的结构，所以我们不必担心没有探测到少量对象具有的某种属性；也就是说，在这一章中我们不讨论模式的问题。从可伸缩性的角度来看这是一个好的消息：举个例子来说，根据第 4 章的讨论，我们可以从数据集中抽取一个（随机）样本来进行分析，这样仍然可能得到很好的结果。

9.2 通过概率分布和密度描述数据

9.2.1 简介

对于从很大的总体中抽取出的数据，或者可以被看作是从很大的总体中抽取出的数据
272 （例如，因为测量中已经合入了测量误差），通过潜在分布或密度函数来描述它们是一种基本的描述策略。如果采用第 4 章所用的 p 维数据矩阵表示，那么对于变量 X_1，…，X_p，我们的目标就是模拟联合分布或密度 $f(X_1, \cdots, X_p)$。为了方便，我们在接下来的讨论中统一采用“密度（densities）”这个术语，但这种思想既适用于变量 X 连续的情况，也适用于离散的情况。

从某种意义上讲联合密度提供了关于变量 X_1，…，X_p 的全部信息。有了联合密度，我们就可以回答有关变量子集间关系的任意问题，例如，X_3 和 X_7 是否独立？也可以回答给定其他变量后某一变量的条件密度问题，例如，给定 X_7 的值时 X_3 的概率分布 $f(x_3 \mid x_7)$ 是什么？

在很多实际情况下知道联合密度是很有用的也是我们所希望的。例如，我们可能对密度（对于取实数值的 X）的最频值（modes）感兴趣。假设我们在分析某一银行 n 个客户的数据集中的两个变量：**income**（**收入**）和 **spending**（**信用卡支出**）。对于很大的 n，在散点图中，我们可能仅看到一大群点，而且有很多重叠在其他的上面。相反如果我们估计出联合密度 f（**income**，**spending**）（我们还没有描述如何做到这一点），那么我们便可以把这个密度函数画成一幅二维的等高线图，或者把密度函数作为第三维画成三维的图形。估计出的联合密度可以揭示很多有用的信息，包括数据中潜在的结构和体现的模式。例如，密度函数波峰（最频值）出现的位置可能表明那里存在着子客户群。相反，间隙、空穴或波谷可能说明在对应这些区域这个银行根本没有客户。从密度函数的总体形状可以看出这个客户群的**收入**和**支出**是如何关联的。

与上面讲的大不相同的另一类问题是生成查询庞大数据库的近似结果（又被称为查询的选择能力估计（query selectivity estimation））。进一步说也就是：对于给定的查询（也就是观察记录必须满足的条件），估计满足这一条件的记录行的比例（即查询的选择能力）。在数据库系统的查询优化中需要这样的估计，而且项查询优化任务可能需要上百次这样的估计。如果我们有了对数据库中数据的联合分布的较好近似，那么我们就可以使用它得到近似的选择
273 性，大大地提高评估的计算效率。

所以联合密度是很多分析的重要基础，我们必须找到很好的方式对其（或者它的主要特征）进行估计和概括。

9.2.2 用来估计概率分布和密度的评分函数

正如我们在前面章节中所指出的，用来估计概率函数参数的最常见评分函数是似然（或者是经过单调的对数转换后的对数似然，二者效果是等价的）。我们再回忆一下，如果随机变量 $\mathbf{X}$ 的概率函数是 $f(x;\theta)$，其中 θ 是需要估计的参数，那么对数似然是 $\log f(D \mid \theta)$，其中 $D = \{\mathbf{x}(1), \cdots, \mathbf{x}(n)\}$，是观察到的数据。假定数据矩阵的各行是独立的，于是

$$S_L(\theta) = -\sum_{i=1}^{n} \log f(\mathbf{x}(i);\theta) \tag{9.1}$$

如果 f 具有一种简单的函数形式（例如，它具有附录中列出的一元分布形式），那么通常可以直接最小化这个评分函数，得到参数θ的闭合形式估计量。然而，如果 f 比较复杂，那么就需要使用递归的优化方法了。

尽管似然是一种很重要的尺度，但它并不足以胜任所有的模型评价任务。尤其是当比较不同复杂度（例如，参数个数不同的正态密度）的模型时，就可能产生问题。例如，对于一系列相互包含的模型（较高层次的模型包含较低层次的模型作为特例）来说，更灵活的较高层模型总会有较大的似然。这不足为奇，因为似然评分函数是衡量模型匹配数据好坏的尺度，所以灵活性更大的模型拟合数据的能力必然不会比它所包含的灵活性低的模型差（通常是更好）。这意味着，如果我们的目的是对一个完全的数据总体进行概括时，那么使用似然作为评分函数是很合适的（因为我们的目标就是判断简化的描述和原始数据间拟合的紧密度）；但是如果我们是用它来选择一个适用于来自更大总体的样本的单一模型（隐含的目标是泛化到未观察到的数据）时，那么似然是不合适的。在后一种情况中，我们可以通过修改似然使 274
其考虑模型的复杂性来解决这个问题。我们在第 7 章中对此作了详细的讨论，当时我们列出了几种评分函数，它们都通过向似然中加入额外项来惩罚复杂的模型。例如 BIC（贝叶斯信息判据，Bayesian Information Criterion）评分函数是这样定义的：

$$S_{BIC}(M_k) = 2S_L(\hat{\theta}_k; M_k) + d_k \log n,\quad 1 \leqslant k \leqslant K \tag{9.2}$$

其中，d_k 是模型 M_k 中的参数个数；$S_L(\hat{\theta}_k; M_k)$ 是负对数似然的最小化值（当参数等于$\hat{\theta}_k$时得到此最小值）。

还有另一种方法，正如第 7 章所讨论的，我们可以使用一个独立的数据样本来计算分数，这样便得到了一种“样本外（out-of-sample）”评估。这就是验证对数似然（validation log-likelihood）（又称“holdout log-likelihood”），它是这样定义的：

$$S_{vl}(M_k) = \sum_{x \in D_v} \log f_{M_k}(\mathbf{x}|\hat{\theta}),\quad 1 \leqslant k \leqslant K \tag{9.3}$$

其中点 $\mathbf{x}$ 来自确认数据集 D_v，$\hat{\theta}$ 是根据不相交的训练数据集 $D_t = D \backslash D_v$ 估计出的参数（比如使用最大似然估计），被评估的模型总数为 K。

9.2.3 参数密度模型

我们在第 6 章里指出，可以把密度函数的模型结构分为两大类：参数的和非参数的。参数模型（parametric model）为密度函数假定一个特定的函数形式（通常比较简单），例如均匀分布、正态分布、指数分布、泊松分布等等（参见附录 A，那里介绍了这些常用密度分布的更多细节）。这些分布函数大多是受数据产生机制的潜在因果模型所启发的。那么如何选取密度函数呢？这应根据被观测变量的知识来定（例如，如果要为像收入这样的变量选取对其建模的分布，那么就应该考虑它只可以取正值的知识）。很多情况下，可以用较少数量的参数来刻划参数模型。例如，p 维正态分布是这样定义的：

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)} \tag{9.4}$$ 275

其中Σ是 X 个变量的 $p \times p$ 协方差矩阵，$|\Sigma|$是这个矩阵的行列式，μ是 X 的 p 维均值向量。模形的参数是均值向量和协方差矩阵（因此共有 $p+p(p+1)/2$ 个参数）。

在数据分析中多元正态（即高斯）分布是特别重要的。举例来说，根据中心极限定理，在相当宽广的假定下，N 个独立的随机变量（每一个可以服从任意的分布）的均值趋向于服从正态分布。尽管表面看来这个结论是逐步逼近的（asymptotic），但即使对于相当小的 N 值（比如说 N=10）样本均值通常也非常接近正态分布。所以，如果可以把一个测量看作是多个相对独立的过程得到的汇总结果，那么正态模型经常是可以采用的合理模型。

公式 9.4 中的多元正态模型的函数形式并非像看起来那么复杂。它的指数 $(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)$ 是一个标量（二次形式），被称为数据点 x 和均值μ间的马氏距离（Mahalanobis distance），表
276 示为 $r_\Sigma^2(\mathbf{x},\mu)$。这是对标准欧氏距离的推广，当计算距离时考虑了 p 维空间的相关性（依靠协方差矩阵Σ）。公式 9.4 的分母就是一个进行标准化的常数（称为 C），用来保证这个函数的积分为 1（也就是保证它是一个真正的概率密度函数）。这样，我们就可以用以下更加简单的形式来表示前面的正态模型：

$$f(x)=\frac{1}{C}e^{-r_\Sigma^2(\mathbf{x},\mu)/2} \tag{9.5}$$

要是我们能够画出（比如 p=2 时的）具有同一固定 $r_\Sigma^2(\mathbf{x},\mu)$ 值的所有点，（或等价的，对于某个参数 c，画出在 $f(\mathbf{x})=c$ 的密度等高线上的所有点），那么我们会发现这些点描绘出一个 2 维空间中的椭圆（更一般地讲是 p 维空间中的超椭圆），椭圆的中心是μ。也就是说，描绘多元正态分布的等高线是椭圆形的，到中心的高度按 $r_\Sigma^2(\mathbf{x},\mu)$ 的指数函数下降。图 9-1 画出了二维空间中的简单示意图。椭圆等高线的离心率和方向由Σ的形式决定。如果Σ是单位矩阵（identity matrix）（所有变量具有同样的方差并且不相关），那么等高线是圆。如果Σ是对角线矩阵（diagonal matrix），但对角线上具有不同的方差项，那么椭圆等高线的轴线与变量轴平行，并且椭圆等高线是顺着有较大方差的变量轴的方向伸展的。最后，如果某些变量是高度相关的，那么椭圆（或超椭圆）等高线将沿着这些变量的线性组合所定义的向量方向伸展。在图 9-1 中，变量 X_1 和 X_2 是高度相关的，因此数据是沿 X_1+X_2 的线性组合所定义的方向散布的。

对于高维数据（p 很大），正态模型中参数的数量是由协方差矩阵中的 $O(p^2)$个协方差项支配的。实践中，我们可能不想要对所有这么多协方差项建模，因为对于很大的 p 和有限的 n（现有数据点的数量）我们可能无法可靠地估计出很多协方差项。例如，我们可以假定变量是独立的，在正态的情况下这等价于假定协方差矩阵具有对角线结构（所以仅有 p 个参数）。（注意如果我们假定Σ是对角线矩阵，那么容易得出 p 维多元正态密度可以表示为 p 个
277 一元正态分布的乘积，这也是 p 个变量独立的充要条件。）一个更极端的假定是假定 $\Sigma=\sigma^2 I$，其中 I 是单位矩阵——也就是说，所有 p 个变量不仅独立，而且具有一致的方差。

独立是非常严格的假定。一个宽松一些的假定是，协方差矩阵是分块的对角线结构：假定存在独立的变量组（块），但跨组的变量是不独立的。通常可以作各种可能的假定，因此在实践中对假定进行检验是非常重要的。多元正态分布有一个吸引人的属性：对于给定的两个变量，它们条件独立的充要条件是它们在协方差矩阵的逆矩阵中的对应元素为 0。这意味着协方差矩阵的逆Σ^{-1}反映了变量间的关系模式。（或者，至少原则上可以这样做：有必要决定协方差矩阵的逆阵中的一个很小值是否足够小以至于可以被看作 0。）也可以用一个假想

的图形模型来表示这种情况，在这个模型中根本不存在连接对应于这个逆协方差矩阵中的很小值的节点的边（我们在第6章中讨论了图形模型）。

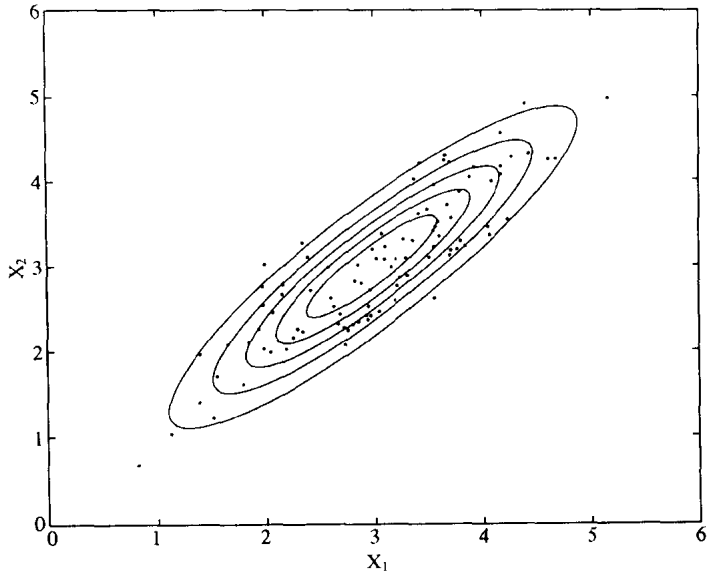


图9-1 二维正态密度函数的密度等高线示意图。密度函数的均值为[3，3]；协方差矩阵为$\Sigma=\begin{pmatrix}1.0 & 0.9\\ 0.9 & 1.0\end{pmatrix}$。图中还画出了根据这一分布模拟出的100个数据点

对模型中所作的假定进行检验是很重要的。经常可以使用某些统计拟合度（goodness-of-fit）检验，但即便是简单的目测也可能看出问题。通过简单的直方图（或第3章列出的其他更复杂版本）可以立即看出容许范围的不当（例如，上面指出的 income 应为非负值），或缺乏对称等问题。如果假定是不合理的，那么可以分析对原始评分函数的转换是否合适。但不幸的是，不存在一种一成不变的简单（hard-and-fast）法则可以用来判断一个假定是否合理。微小的背离很可能是不要紧的——但这要看具体的问题。这就是数据挖掘的艺术性。在很多分布假定被推翻的情况下，我们仍可以很好的得到参数的合理估计，但统计检验是无效的。例如，我们可以使用最小平方评分函数硬性拟合回归模型，不管误差是否服从正态分布，但对估计出参数的假设检验很可能是不准确的。这或许在模型建立过程中是有关系的——有助于决定是否该包含某个变量——但这可能并不影响最终的模型。如果最终的模型能满足它的目标（例如，回归中的预测精度），那么就可以采用这个模型了。

拟合 p 维正态模型是非常简单的。每一个均值和方差项的最大似然（或贝叶斯）估计可 278
以被定义为闭合的形式（如第4章所讨论的），对于每一个参数仅需要 $O(n)$ 步，因此共需要 $O(np^2)$ 步。其他著名的参数模型（例如附录中所定义的）通常也具有闭合形式的解，扫描所有数据一次就可以得到这些解。

正态模型结构是一种比较简单并有局限的模型。它是单峰的，而且关于椭圆轴对称。完全可以使用它的均值向量和协方差矩阵来定义它。然而，这也决定了它无法表达非线性的关系，也不能表示任何形式的多峰性和分组。下一节要讨论的混合模型提供了一种对多峰型和分组建模的灵活框架。读者也该注意到尽管正态模型是实践中应用最广泛的参数模型，但是

还有很多其他不同“形状”的密度函数对于特定的应用是很有价值的（例如，指数模型、对数正态、泊松分布、Γ分布（the Gamma）等，有兴趣的读者可以参见附录）。多元 t 分布在形式上和多元正态分布是很相似的，但它允许有更长的末端，因此对于较多数据出现在末端的问题它比正态模型预测性能更好。

9.2.4 混合分布和密度

在第 6 章中我们看到了可以把简单的模型进行推广以实现多个分量的混合（mixture）——也就是多个简单分布的线性组合。这就是我们在密度建模中要讨论的下一个内容：也就是把参数分布推广到这些函数的加权线性组合，通过组合简单的模型来建立更复杂的密度和分布模型。当在实践中我们不确信哪一种参数形式合适时，混合模型是特别有用的（在本章的后面我们会看到应用在聚类任务中的混合模型）。

在实践中异质的（heterogeneous）数据集是非常普遍的，也就是说数据代表了多个不同的子群体或小组，而不是同质的单一一组。对于特别庞大的数据集异质性更加普遍，其中不同的分组数据可能代表不同的内在现象，而这些又被收集起来形成一个大的数据集。
279 为了说明这一点，考虑第 3 章的图 3-1。这是 1996 年持某一种信用卡的客户使用该信用卡在超市购物周数的直方图。正像我们前面所指出的，这个直方图看起来是双峰的，一个较大的明显的波峰在左边，另一个较小的不过可能很重要的波峰在右边。初看起来，这样的数据可能服从泊松分布（尽管上边界被限定为 52），但是从末端来看又不像泊松分布而且无法解释右侧的波峰。因此我们必须使用更复杂更灵活的模型。一种办法是用具有两个分量的理论分布来概括这个实验分布。可能存在两种类型的客户：一种是不太在超市中使用信用卡的；另一种是大多时候（周）都在使用信用卡的。可以用一个小概率的泊松分步概括第一种人。而第二种人可以用一个反向的泊松分布来概括，它的波峰在 45 或 46 周（波峰的位置是拟合模型到数据时的一个要估计的参数）。这样得到了一个如下形式的全局分布：

$$f(x)=\pi\frac{(\lambda_1)^x e^{-\lambda_1}}{x!}+(1-\pi)\frac{(\lambda_2)^{52-x}e^{-\lambda_2}}{(52-x)!} \tag{9.6}$$

其中 x 是随机变量 X 的值，取值为 0 到 52（指出一个人一年中有多少周在超市中使用他们的信用卡），$\lambda_1>0$，$\lambda_2>0$ 是两个泊松模型分量中的参数。这里π是一个人属于第一组的概率，而且据此，表达式 $\lambda_1^x e^{-\lambda_1}/x!$就是这个人在这一年中使用信用卡 x 次的概率。类似的，$1-\pi$是这个人属于第二组的概率，表达式 $\lambda_2^{52-x}e^{-\lambda_2}/(52-x)!$就是这个人在这一年中使用信用卡 x 次的条件概率。

思考这种模型的一种方法是分两步考虑一个特定个体的行为过程。在第一步中，个体有 π（或 $1-\pi$）的概率属于一组或另一组。在第二步中再考虑观察值 x 是根据他或她在第一步中所属的分量分布产生的。

公式 9-6 是有限混合分布（finite mixture distribution）的一个例子，在这个分布中总体模型 $f(x)$ 是有限数量（这里是两个）的分量分布的加权线性组合。显然，这个混合模型比单一的泊松分布具有更高的灵活性——至少它包含了三个参数，而不是只有一个。然而，这
280 个模型是基于一定猜测的，所以得到的可能也只是对潜在数据的一种更接近的描述。这两个

方面——较多数量参数所带来的额外灵活性和建立在潜在总体异质性猜测之上的论据——意味着混合模型广泛地应用于单一标准形式难以胜任的复杂分布建模中。

混合分布（对多元的 $\mathbf{x}$）的一般形式为：

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x};\theta_k) \tag{9.7}$$

其中π_k是一个观察值来自第 k 个分量的概率（即所谓的第 k 项混合比例（mixing proportion）又被称为权），K 是分量数，f_k（$x;\theta_k$）是第 k 个分量分布，θ_k是描述第 k 个分量的参数向量（在泊松分布混合模型的例子中，θ_k就是单一的参数λ_k）。在大多数应用中分量分布 f_k 具有统一的形式，不过也有例外。应用最广的混合分布形式是使用正态分量。注意混合比例π_k必须在 0 和 1 之间并且相加之和为 1。

根据理论研究，可以认为符合混合分布的实际例子包括鱼的长度分布（因为它们在每年的确定时间孵化），失败数据（存在不同的失败原因，每一个原因导致一种失败次数分布），死亡时间，和不同人群的特征分布（比如男人和女人的身高）。

9.2.5 混合模型的 EM 算法

和本章前面讨论的简单参数模型的情况不同，对于给定一个数据集 $D=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$，当潜在模型是混合模型时通常不存在可以直接最大化似然评分函数的闭合形式的技术。通过列出混合模型的对数似然就可以很容易的看出这一点：我们得到的是类似 log（$\sum_k \pi_k f_k$（$\mathbf{x};\theta_k$））形式的多项和，这是一种非线性的优化问题（并不像多元正态模型那样存在闭合形式解）。

近年来，已经应用了很多方法来估计一个给定混合形式的混合分布参数。这其中一种应用最广的现代方法就是 EM 方法。正如第 8 章所讨论的，可以把这种方法看作是一种通用的迭代优化算法，对于给定的概率模型和有残缺值的数据，可以使用该算法最大化似然 281
评分函数。对于现在的情况，可以把混合模型看作是一种分类标签残缺的分布。如果我们知道这些标签，那么就可以通过把数据点划分到它们各自的小组来得到每一个分量的闭合形式估计。然而，既然我们不知道每个数据点的由来，我们必须想办法同时分析一个数据点来源于哪一个分量和这些分量的参数。EM 算法可以干净利落地解决这个“先有鸡还是先有蛋”的问题：它先猜想每个分量的参数值，然后计算每一个数据点来自 K 个分量中的一个的概率（这一步被称为“E 步骤”），再根据得出的这些隶属关系概率计算每一个分量的参数（这一步被称为“M 步骤”，而且通常是以闭合形式求出的），而后再重新计算隶属关系概率，并以这种方式继续下去，直到这个似然收敛。正如第 8 章所讨论的，尽管这一算法看起来是试探性的，但已经证明在每一步的 EM 过程中似然都只会增大，因此可以保证（在相当宽广的条件下）这种方法会收敛，至少会收敛在似然（关于参数空间的函数）的局部最大值。

EM 算法的复杂度依赖于每一步迭代中 E 步骤和 M 步骤的复杂度。对于含有 K 个分量的多元正态混合模型，主要的计算是在每一次迭代的 M 步骤中的 K 个协方差矩阵的运算。在 p 维空间中，对于 K 个聚类，要估计 $O(Kp^2)$个协方差参数，而且对每一个参数需要对 n 个数据点和隶属关系权进行汇总，所以每一步的时间复杂度为 $O(Kp^2\mathrm{n})$。对于一元混合模型（就像上面所介绍的泊松分布的例子），可以算出其时间复杂度为 $O(Kn)$。空间复杂度

一般为 $O(Kn)$，用来存储 n 个数据点 $\mathbf{x}(i)$ 中每一个的隶属关系概率向量（每个向量含 K 个分量）。然而，对于很大的 p，我们经常不必特别存储 $n\times K$ 个隶属关系概率的矩阵，因为我们可以在每一次的 M 步骤中通过一次扫描 n 个数据点来增量（incrementally）计算参数估计。

EM 算法经常在起始的几次迭代中以较大的幅度增长似然，然后慢慢地收敛到最终值。然而似然函数相对迭代次数的函数未必是凹形的。例如，图 9-2 画出了对数似然相对于 EM 算法迭代次数的函数曲线（从中可以看出 EM 算法的收敛性），对应的问题是用高斯混合模型来拟合二维医疗数据集（我们会在 9.6 节中更详细的讨论这个数据集）。对于很多数据集
282 和模型我们经常可以仅用 5 到 20 步迭代就得到可接受的解。当然 EM 算法所给出的每个解都是关于搜索起始点的函数（因为 EM 是一个局部搜索算法），因此，从随机选取的起点多次重新启动算法是一个好的主意，这样可以避免最终得到一个很差的局部最大值。注意无论是 K 或 p（还是全部）增大，似然局部最大值的数值也会随着参数空间的维度变化而相应明显增大。

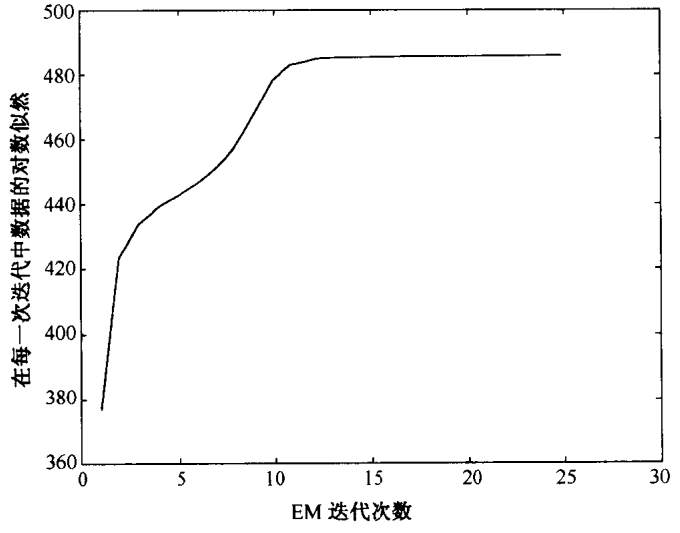


图9-2　血红细胞数据的对数似然相对于迭代次数的函数曲线。
使用的模型是两分量正态混合模型（参见图9-11）

在使用最大似然方法估计混合分布时，需要注意一些特殊情况。例如，在正态混合模型中，如果一个分量的均值与一个样本点相等，并且它的标准差趋向于零，那么似然会无限地增长。而在这种情况下最大似然解很可能是有限值的。有很多方法可以用来解决这个问题。可以选取似然的最大有限值来给出被估计的参数值。另外，如果限制标准差相等，那么这个问题也不会发生。一种更通用的方案是使用贝叶斯方法来处理这个问题，取关于参数的先验，并且不再最大化似然，而是最大化MAP评分函数。这样先验提供了一个框架使评分函数（MAP
283 评分函数）远离参数空间中有问题的区域。注意可以很容易地把 EM 算法从最大化似然推广到最大化 MAP（例如，把 M 步骤替换为 MAP 步骤，依次类推）。

可能发生的另一个问题是由于混合分布缺乏可识别性（identifiability）造成的。如果一族混合分布满足以下的充要条件，那么我们就说它是可识别的。即当且仅当这一族中的两个

成员相等：

$$\sum_{k=1}^{c}\pi_k f(x;\theta_k)=\sum_{j=1}^{c'}\pi'_j f(x;\theta'_j) \tag{9.8}$$

这意味着 $c=c'$，并且对于所有的 k 存在某个 j 使 $\pi_k=\pi'_j$ 和 $\theta_k=\theta'_j$。如果一族分布是不可识别的，那么就无法区分它的两个不同成员，这会导致估计中出现问题。

对于离散分布不可识别（nonidentifiability）的问题可能比连续的情况更严重，因为对于 m 个类目，仅可以建立 $m-1$ 个独立的等式。例如，对于几个伯努利分量混合的情况，在数据中仅存在一条有用的信息，也就是数据中 1 发生的比例。因此，没有办法估计分别属于每一个伯努利分量的比例，因此也就无法估计这些分量的参数。

9.2.6 非参数的密度估计

第 3 章中我们简要地介绍了通过取感兴趣点周围的 x 测量值的局部加权平均来估计密度函数的思想（即所谓的“核密度（kernel density）”方法）。例如，直方图是这种思想的一个主要版本，我们只要数出落入特定“柱位（bin）”的点数。我们对密度的估计就是给定柱位里的点数，经适当的缩放。用直方图作密度的模型结构是有问题的，这有几个原因。首先即使是对那些确实平滑的函数，它给出的估计也是不平滑的；另外，没有一种明显的方法来选取柱位的数量、位置和宽度。而且当我们从一维的直方图转到 p 维的直方图时这些问题会变得进一步恶化。但是，对于很大的数据集和很小的 p（尤其是 p=1 时），柱位的宽度会变得很小，结果得到的密度估计可能还是比较平滑的，而且对柱位的宽度和精确位置是不敏感的。对于庞大的数据集，观察每一个变量的柱状图总是一个好的注意，因为直方图可以提供丰富的信 284
息，比如孤立点、多峰型、对称性、末端特征等等（回忆第 3 章中的 PIMA，印第安人血压数据的例子，那里的直方图清晰地指出了一些相当值得怀疑的血压值为 0 的数据）。

一种更通用的局部密度模型结构是把任意点 x 的密度定义为与训练数据集中所有点的加权求和成正比，其中的权是由适当选取的核函数（kernel function）所定义的。对于一维的情况我们有（如第 3 章所定义的）

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}w_i,\quad w_i=K\left(\frac{x-x(i)}{h}\right) \tag{9.9}$$

其中 $f(x)$ 是对查询点 x 的核密度估计，$K(t)$ 是核函数（例如可以这样定义：如果 $t\leqslant 1$，$K(t)=1-|t|$；否则 $K(t)=0$），h 是核的带宽（bandwidth）。直观地讲，x 点的密度与在 x 点计算出的权成正比，而这个权又依赖于训练数据中的 n 个点和 x 的接近度。与使用非参数回归（在第 6 章中讨论的）一样，这里并没有明确的定义模型，而是由数据和核函数隐含决定。因为所有的数据点是保留在模型中的，所以从这个意义上来讲这种方法是一种基于记忆的方法（memory-based），也就是说，没有进行任何概括。当然对于庞大的数据集从计算和存储的角度来看这种方法可能是不适用的。

在一维中，核函数 K 通常被选为一个平滑的积分为 1 的单峰函数（比如正态或三角形的分布），精确的形状通常不是很关键的。和在回归中一样，带宽 h 起到了决定模型平滑程度的作用。如果 h 比较大，那么核比较宽，这样使很多点会对求和贡献出显著的权，从而使密度估计很平滑。如果 h 比较小，那么核估计是由靠近 x 的少数点所决定的，这样使密度

估计对局部数据更加敏感（看起来更加长而尖）。实践中为 h 估计一个很好的值是有一定难度的。不存在一种固定方法来找到普遍适用的带宽 h。基于交叉验证的技术有时是有效的，但是通常需要复杂的计算而且不保证总是可靠的。通常推荐对具体情况使用简单的“目测”来检查选取的 h 值是否合理。

285 在适当的假定下，这些核模型具有足够的灵活性来逼近任何平滑的密度函数，如果选取的 h 合适那会更好。然而，要达到这种近似我们可能要取无限数量的数据点，这与实践中我们所看到的有限数据集的现实多少有些不相符。尽管如此，核模型作为一种决定数据结构的可视化方法（比如局部的尖峰或空隙）对于低维问题是非常有价值的，因为其他的方法可能无法做到这一点。

例 9.1　图 9-3 中画出了对乙醇（ethanol（E））测量结果的几种不同密度估计，这个变量是取自一个包含不同地理位置空气污染情况测量结果的数据集。左上角的直方图是很“粗糙的”而且有噪声（至少对于当前选择的柱位宽度和位置来说是这样的）。带宽 h=0.5 的正态核可能是最平滑的（右上）。相反，带宽 h = 0.1 的正态核估计可能是噪声最多的（右下），它在密度中引入了很可能虚假的波形。H=0.25 的估计（左下）看起来是最好的，而且在过于平滑和不平滑间比其他估计作了更好的折衷。从这幅图可以看到乙醇的测量结果具有双峰特征。虽然可视化观察方法对于交互式的决定带宽是有价值的，但是这种方法主要适用于一维和二维问题。

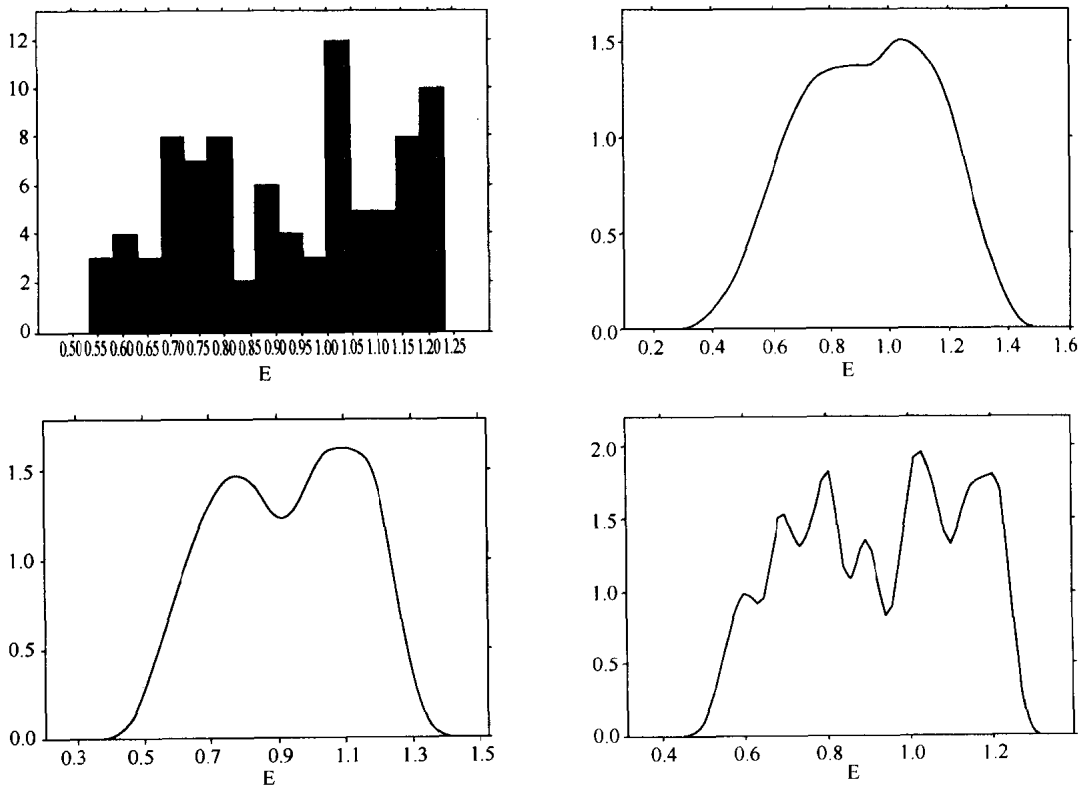


图9-3　对变量ethanol（E）的密度估计。图中分别使用了柱形图（左上），和三个不同带宽的高斯核估计：h = 0.5（右上），h = 0.25（左下），h = 0.1（右下）

随着 p 的增长核模型密度估计会而面临更多困难。首先，我们需要定义一个 p 维的核函数。一种流行的做法是把 p 维核函数定义为一维核的乘积，每个一维核有自己的带宽（每一维的带宽分别是 h_1，…，h_p），这样使参数数量与维数呈线性关系。一个不太明显的问题是在高维空间中点相距很远是很正常的（又是“维度效应”）。事实上，如果要保持估计的误差不随维数的增长而变化，那么所需的数据点数是随 p 指数增长的。（回忆第 6 章中的例子，要得到 10 维正态分布中均值的可靠估计需要 842 000 个数据点。）这意味着实践中核模型确实仅对低维问题才是可行的。

对于庞大的数据集，要实现核模型经常是很复杂的。除非核函数 $K(t)$的紧凑性（compact）非常好（也就是，在 t 的某个有限范围之外函数值都为 0），否则要计算某一点 x 的核估计 $f(x)$就要对数据库中所有 n 个数据点的贡献进行汇总。当然，在实践中这些贡献中的大多数是可以忽略的（也就是，大多数是位于核的末端），因此有很多方法来加速这种计算。尽管如此，这种基于记忆的表示在存储和计算方面都是比较复杂的（仅计算一个查询数据点的密度所需的计算量就可能是 $O(n)$）。

9.2.7 范畴型数据的联合分布

我们在第 6 章中讨论了为多元范畴型（categorical）数据建立联合分布的问题。比如说如果我们有 p 个变量，每一个可以取 m 个值，那么这个联合分布需要确定 $O(m^p)$个不同的概率。这种指数的增长会导致很多方面的问题。

第一个问题是如何估计这么大数量的概率。举例来说，设$\{p_1, \cdots, p_{m^p}\}$为未知分布中的所有联合概率项，我们要从 n 个 p 维观察值的数据集中估计出这些项。因此，我们可以想象有 m^p 个不同的“单元格（cell）”$\{c_1, \cdots, c_{m^p}\}$，每一个含有 n_i 个观察值，$1 \leqslant i \leqslant m^p$。如果样本是来自 $p(\mathbf{x})$容量为 n 的随机样本，那么可以将落入单元格 i 中的期望数据点数写为 $E_{p(\mathbf{x})}[n_i] = np_i$。假定（举例来说）$p(\mathbf{x})$是近似的均匀分布（也就是，$p_i \approx 1/m^p$），那么

$$E_{p(\mathbf{x})}[n_i] \approx \frac{n}{m^p} \tag{9.10}$$

286
|
287

于是，如果 $n < 0.5m^p$，那么落入任何给定单元格的期望数据点数都更接近于 0，而不是接近 1。而且，如果我们使用直接的频数计数（最大似然估计——见第 4 章）作为估计概率的方法，那么我们会把每一个空的单元格估计为 $\hat{p}_i = 0$，不管实际上是否真的 $p_i = 0$。注意如果 $p(\mathbf{x})$是不均匀的分布，那么这个问题会更严重，因为会有更多单元格有更小的 p_i（也就是落入任何数据的机会更小）。这里的根本问题是单元格数量以指数 m^p 增长。对于 p=20 的二进制变量（m=2）$m^p \approx 10^6$。如果变量数加倍到 p = 40，那么 $m^p \approx 10^{12}$。比如说，对于 p=20 的情况我们有 n 个数据点，而且我们要加入一些新的变量但要保持每一个单元格中的期望数据点数量不变。如果我们要再加入 20 个变量，那么数据数就要从 n 增长到 $n' = 10^6 n$，成百万倍的增长。

第二个实践问题是即使我们可以从数据中估计出完全的联合分布，那么要直接操作这个分布时间和空间上都是指数级的。一个完全的联合分布的内存需求为 $O(m^p)$，例如，对于一个 40 个二进制变量的完整分布需要存储 $O(10^{12})$个实数值概率。而且，很多使用这一分布的计算的计算量也呈指数增长。设变量为$\{X_1, \cdots, X_p\}$，每一个取 m 个值。如果我们想要计算任一个单一变量 X_j的边际分布，我们就要这样计算：

$$p(x_j)=\sum_{X_1,\ldots X_{j-1},X_{j+1},\ldots,X_p} p(X_1,\ldots,X_{j-1},x_j,X_{j+1},\ldots,X_p) \tag{9.11}$$

也就是说，要对分布中的所有其他变量汇总。右侧的求和包括 $O(m^{p-1})$ 次求和——当 p=40，$m=2$ 时是 $O(10^{39})$。显然这种操作是难以处理的，除非 m 和 p 的值很小。

288 可行的做法是我们仅能对低维问题使用完全的联合分布。而且尽管我们的例子是对范畴型数据的，但实质上对于排序型数据（ordered）和实数值数据也是适用的。

正像我们在第 6 章中所见到的，克服这种维度效应的一种关键思想是从分布的结构着手——例如，假定各个变量是独立的，那么

$$p(\mathbf{x})=p(x_1,\cdots,x_p)=\prod_{j=1}^{p} p_j(x_j) \tag{9.12}$$

这样就不再需要 $O(m^p)$ 个分别的概率。而仅需要 p 个“边际”分布 $p_1(x_1)$，…，$p_p(x_p)$，其中的每一个可以用 m 个数字确定，所以共有 mp 个概率。当然，就像刚才所强调的，独立只是一种假定，通常这一假定对于大多数现实的数据挖掘问题是过强的（too strong）。

第 6 章中介绍了一种弱一些的假定，即假定存在一个隐藏的（“潜在的”）变量 C，取 K 个值，而且测量 $\mathbf{x}$ 是相对给定的 C 条件独立的。这等价于前面讨论的混合分布，只不过附带了一个额外的假定：每一个分量内部条件独立，也就是：

$$p(\mathbf{x})=\sum_{k=1}^{K}\pi_k p_k(\mathbf{x})=\sum_{k=1}^{K}\pi_k\left(\prod_{j=1}^{p} p_{k,j}(x_j)\right) \tag{9.13}$$

这个模型的每一个分量需要 mp 个概率，再乘上 K 个分量，再加上 K 个分量的权π_1，…，π_K。因此它是随 K、m、和 p 线性变化的，而不是指数。可以再次使用 EM 算法来估计每一个分量 $p_k(\mathbf{x})$的参数（和权π_k），在估计中利用了条件独立的假定。理解这种“混合独立模型”的一种方式是：我们要在数据中发现 K 个不同的分组，对于每一分组各个变量至少是近似条件独立的。事实上，给定一个固定的 K 值，EM 算法会尽可能找到 K 个分量分布（每一个都具有条件独立的形式）使数据的总体似然最大化。这种模型在对庞大又很稀疏的事务性数据集或表示为二进制向量的文本文档集合建模中是非常有价值的。如何找到适当的 K 值呢？这依赖于我们的目标：从描述的角度看，我们可以根据我们要匹配的模型的复杂度来

289 调整 K。注意，这种形式的模型与第 6 章讨论的一阶“朴素”贝叶斯模型是等价的（在第 10 章的分类中还会再次讨论），只不过这里的分类变量 C 是未观察到的，而且必须要从数据中学习。在这一章的后面我们也将看到这也为聚类数据提供了一个有价值的基础，在那里我们把每一个分量 $p_k(\mathbf{x})$解释为一个聚类。

一种多少有些不同的建立概率分布结构的方法是以通用形式对条件独立建模。我们在第 6 章中介绍了这种方法的一种通用框架（称为信念网络（belief network），或者叫，无循环有向图形模型（acyclic directed graphical models））。回忆这种模型的基本公式可以写为：

$$p(\mathbf{x})=\prod_{j=1}^{p} p_j(x_j \mid pa(x_j)) \tag{9.14}$$

也就是把总的联合分布因式分解为多个条件分布的乘积。实际上，总可以使用链式法则来定义这样的因式分解式，但是这个模型的威力在于它发挥了当依赖性很弱时独立假定的作用。回忆一下，在图型表示中每一个变量是与图中的一个节点相联系的。一条从 X_i 到 X_j 的直接连线表示 X_j 直接依赖于 X_i。pa（x_j）表示从变量 X_j 的父变量集 pa（X_j）取的值。图中的连接结构蕴含着关于 p (**x**)的一系列条件独立关系。这些独立关系可以归纳为给定 X_j 的父结点的值，pa（X_j），那么节点 X_j 与图中非 X_j 后代的所有变量是独立的。

如果图中父集合的大小与 p 相比很小，那么我们可以把联合分布表示成一种比完整模型简单得多的形式。在这一背景下，独立模型对应于根本没有连线的图，整个图对应于没有任何独立假定的完全分布。另一种著名的图结构是马尔可夫链模型，在这种模型中变量是按某种方式（例如，时间）排序的，而且变量 X_j 仅依赖于 X_{j-1}。因此每一个变量仅与其他两个变量连接，使整个图是各个节点相连成的一条直线（参见第 6 章图 6-7）。

图形形式的一个主要吸引人之处是，它可以用系统的数学化的精确语言来描述和交流概率分布中的独立性。可能更重要的一点是它还为处理联合分布概率的计算方法提供了一种系 290
统框架。举例来说，如果图是单连接的（singly-connected）（也就是，当忽略边的方向性时，图根本没有循环），那么就可以指出计算任何感兴趣的边缘或条件概率的时间复杂度上限为 pm^{d+1}，其中 p 是变量数，m 是每一个变量的取值个数（简单起见，假定所有变量的取值数都是一样的），d 是图中最大父集合的变量数。例如，对于马尔可夫链模型，d=1，于是得到这种模型的复杂度为 O (pm^2)。对于有循环的图，等价的复杂度上限为 $pm^{d'+1}$，其中 d'是等价的单连接图（以一种系统方式从原始图得到的）中的最大父集合大小。

从数据挖掘的角度来看，从数据中学习图形模型的问题可以分为两类：给定图形模型的结构，学习它的参数；第二种是同时学习参数和结构，这显然更为复杂。注意对于范畴型数据的情况，模型的参数就是每一个变量的条件概率表，p（$x_j \mid pa(X_j)$），$1 \leqslant j \leqslant p$。

如果给定了固定的图形模型结构，那么就不需要进行结构搜索，因此简单的最大似然和 MAP 评分函数就可以工作的很好了。如果没有任何隐藏变量，那么学习问题就简化为给定每一个变量的 pa（X_j）来估计它的条件概率表：不论是使用最大似然还是 MAP 这都简化为简单的计数问题（见第 4 章）。如果有隐藏变量，并假定这些变量在图中的连接是已知的，那么也可以在相当宽广的条件下直接使用 EM 算法（第 8 章）。剩下的问题就是迭代估计条件概率表，并像以往一样注意初始条件的选取和收敛的检测。可以把前面讨论的混合模型看作具有单一隐藏变量的图形模型。可以把隐马尔可夫模型（比如在语音应用中所使用的）看作具有一个隐藏变量（该变量是离散的并且依赖于时间）的图形模型。

有必要强调，如果我们预先有很大的把握认为某一图形模型结构会适合我们的数据挖掘问题，那么通常是值得利用这一知识的（假定它是可靠的）：可以把它作为一个固定的结构，也可以作为我们下面要介绍的学习方法的一个起始结构。

从数据中学习有向图形模型结构已经成为最近研究的一个热点课题，而且也已经开发出 291
了针对这一目标的很多算法。首先考虑学习无隐藏变量的结构的问题。通常评分函数是某种形式的惩罚似然：例如 BIC 评分函数（见 9.2.2 小节）应用得非常广泛，因为它易于计算。给定了评分函数，问题便简化为在图空间中搜索产生最大分数的图结构（带有估计出的参数）。已经证明寻找最大分数的一般性问题是 NP-困难（NP-hard）的（似乎数据挖掘中的大多数结构寻找问题都如此）。因此，要使用递归的局部搜索方法：从某个“先验”结构开始

（比如空的图），然后增加或删除边直到不可能再对评分函数作出任何局部改善。从计算的角度来看一个有价值的特征是：因为可以把分布表示为因式形式（factored form）（见公式9.14），所以似然和惩罚项也可以被加入到图形结构的局部表达式中——例如，仅包含 X_j 和它的父的项。于是我们可以通过局部计算（因为修改仅影响评分函数中的一个因子）来观察对模型的局部修改的效果。

学习含有隐藏变量的结构还是一个在研究的问题，显然它比学习不含隐藏变量的结构（就已经是NP-困难的）要复杂得多。EM算法也是适用的，但这种搜索问题通常是非常复杂的，因为有太多的不同方式把隐藏变量引入多元模型。

对数线性模型（log-linear models）族是对无回路有向图形模型的进一步推广，这种模型用更一般的形式来刻画依赖关系。对这类模型的讨论超出了本书的范围（补充读物中提供了一些参考资料）。马尔可夫随机场（Markov random fields）是另一类图形模型，它使用一种无向图来表示依赖性，也就是表示图象像素间的关联效应。这些随机场模型广泛的应用于图
292 像分析和空间统计学中，用来定义栅格或图像测量值的联合分布。

9.3 聚类分析背景

现在我们放下对概率密度和分布模型的讨论，转向另一种与描述有关的数据挖掘任务——聚类分析，也就是把一个数据集（通常是多元的）分解（decomposing）或划分（partitioning）成组，使同一组中的点彼此相似，但与其他组中的点尽可能不同。尽管使用的技术经常是相同的，但是我们还应该把两种不同的目标区分开来。我们可以把其中之一称为区隔（segmentation）或细分（dissection），它的目标就是以一种方便的方式划分数据。这里的“方便”可能代表管理上的方便、实际操作上的方便或任何其他方面的便利。例如，一个衬衫生产者可能希望仅用几个尺寸和体型就最大可能地覆盖男性群体。他可能要选取领口大小、胸围、臂长等尺寸，以防止有人找不到适合他的尺寸。要实现这一目的，他要按 **collar**、**chest**、**arm length** 这些变量把男人分成几组。然后为每一组制造一种尺寸的衬衫。

与此相反，我们可能希望了解样本数据是否存在自然的子类。例如，可以用这些变量来刻画威士忌酒：**color**，**nose**，**body**，**palate**，**finish**；并且我们想看一看它们是否属于按这些变量所确定的各个类中。这里我们不是为了实践的方便来划分数据，而是希望发现样本和产生样本的总体的某些属性——事实上是要看总体是否是异质的。

严格来讲，第二种行为是聚类分析的目标——分析数据是否属于各个独立的分组，使每一组中的成员彼此相似，但与其他组中的成员不同。然而，“聚类分析”这一术语也经常泛指区隔和聚类分析问题二者中的任一种（这样我们也会方便一些）。每一种情况的目标都是把数据分裂成多个类，所以这也不是很严重的误用。可以这样解决这个问题，正如我们将要看到的，划分数据的不同算法非常多，因此我们可以使用算法来称呼问题。重要的是把方法和目标相匹配。这样不论我们怎么称呼这种行为都无所谓。

293 **例 9.2** 可以根据信用卡持有者如何使用信用卡来把他们分成多个子类，也就是他们用信用卡购买了什么，花了多少钱，用卡的频繁程度如何，在哪里用卡，等等。标识出卡主所属的群体对于市场营销是非常有价值的，这样就可以向卡主发送他们感兴趣的促销资料（这显然既有利于卡主，又有利于公司）。事实上，本节所

讨论技术的一个主要应用就是市场区隔（market segmentation）。区隔的方式有生活方式、以前的购买行为、人口统计中的属性或其他特征。

连锁店或许希望知道销路相似的各家分店（根据社区环境、规模、员工数量、与其他点的邻近度等）是否有相似的营业额和取得相似的利润。那么首先要做的就是根据这些变量划分市场，然后再分析每一个小组内的营业额分布。

聚类分析已经用于精神病学等很多医疗领域，用来标识混在同一诊断下的不同子疾病类型。

生物学中使用聚类分析方法来研究表面上一致的植物和动物是否实际上属于不同的种类。类似的，可以根据生存在那里的动植物种类来把各个地理区域划分成组。

为了举例说明在什么情况下区隔和聚类分析的不同是有关系的，考虑一个对城镇中的房屋进行划分的例子。如果我们在组织一项分发服务，我们可能希望按位置来划分，并使每一组内的房屋尽可能地靠近，那么就可以把属于同一组的包裹放到一辆分发车上。另一方面，一个生产家居改善产品的公司可能希望按照房屋的自然状况来划分。一组可能是由小的刚组建的家庭组成的，另一组可能是拥有三到四个房间的家庭，还有一组是（可能很少）高级公寓。

从这个例子可以清楚看出这两种方法的不同关键是在距离的含义上。因为如果要把一系列点分成子群体，使组中的成员距离这一组中的其他成员比距离其他组里的成员更近，那么我们必须先确定“更近”的含义。在第 2 章中我们已经讨论了距离的概念和度量它的不同方法。那里描述的任一种尺度，或者事实上是任何其他的尺度，都可以用作聚类或区隔分析的 294
基础。对这两种技术来讲，距离比点的坐标更加重要。原则上，要进行聚类分析我们所需要知道的就是点之间的距离，而不是任一个变量的值。但是，某些方法要使用聚类的“中心点”，所以需要原始坐标。

聚类分析已经吸引了无数的研究者为之努力，这可以追溯到几个时代以前，所以这方面的文献很多，也很分散，相当一部分被归入统计学和机器学习文献中，但也可以在其他地方找到关于聚类分析的著作。造成这种状况的一个原因是不断地有新的方法被开发出来，有时根本没有注意到这种方法已经开发出来了。更严重的是，很多方法并没有正确理解聚类分析的特征和处理不同类型数据的方式。造成这一问题的原因之一是要说明一个聚类分析是否成功是有难度的。和预测模型不同，在预测模型中我们可以用一个验证数据集来看目标变量预测值的精确度。但不幸的是，对于聚类问题来说不存在泛化到验证数据集的直接概念，虽然后面我们会看到在某些条件下的概率聚类（本章后面会讨论）中可以提出这样的问题：从训练数据中发现的聚类结构是否真正体现了隐含总体的特征。一般来讲，聚类的验证经常是通过目测来完成的，例如，如果一个聚类揭示出了一个有趣而且科学的内幕（insight），那么我们可以判定它是有价值的。对此进行精确的定量分析即便可能，也是很困难的，因为表示一个聚类的有趣程度难免要依赖于具体应用，而且有一定程度的主观性。

正如在后面的几个小节中我们将看到的，不同的聚类分析方法适用于探测不同类型的聚类，当我们选择算法时应该考虑这一点。也就是说，我们应该考虑我们赋予或想赋予“聚类”的含义。实际上，不同的聚类算法在寻找数据中不同类型的聚类结构（或“形状”）时会有不同的偏向，而且有时不能根据聚类算法的描述精确定位偏向的具体细节。

为了说明这一点，我们考虑聚类一个点集。一种方法是使聚类内任何两点间的距离尽可

能的小，那么在一个聚类内每个点与其他任意点是相似的。于是我们会选取一个算法来划分
295 数据使聚类内的点间最大距离最小化（后面有更多介绍）。显然我们期望这种方法产生一个紧凑的大体为球形的聚类。另一种方法是使聚类中的每一点与该聚类中的某一其他成员尽可能地近——不必与所有其他成员。这种方法发现的聚类不一定是紧凑的和大体球形的，而可能是长的（不一定是直的）香肠形状。第一种方法无法捡选出这样的聚类，因此第一种方法适合于区隔的场合。如果每个假想小组中的各个对象是在某种演进过程的不同阶段测量出的，那么第二种方法更合适。例如，在对患某一疾病的人进行聚类分析以寻找这种疾病的不同子类型时，我们应该考虑到患者可能是在患这种疾病的不同阶段接受检验的，因此即使它们属于同一子类型它们也可能有不同的症状。

从这个例子要吸取的最重要经验是我们必须保证方法和目标匹配。尤其是，我们必须明确当前问题中“聚类”定义的含义，然后采用适合于探测与这一定义一致的聚类的聚类分析工具。可能还值得补充一点，那就是在这一问题上我们不应该太自信。因为毕竟数据挖掘是要发现未知的信息，所以我们一定不能武断的把我们以前的概念强加到这种分析中。或许搜索不同类型的聚类结构就会推翻我们以前的看法。

概括地讲，我们可以把聚类分析算法分成三种不同的类型：试图找到一个最优划分以把数据分成指定数量聚类的方法；试图发现聚类结构的层次方法；对潜在聚类建模的基于概率模型方法。我们在接下来的三个小节中依次讨论这些方法。

9.4　基于划分的聚类算法

在第5章中我们介绍了很多情况下可以按五个部分来考虑数据挖掘算法，也就是任务、模型、评分函数、搜索方法和数据管理技术。在基于划分的聚类中，任务就是把数据集划分成 k 个不相交的点集，使每一个子集中的点尽可能同质，也就是，给定 n 个数据点的集合 $D=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$，我们的任务是找到 K 个聚类 $C=\{C_1,\cdots,C_K\}$，使每一个点 $\mathbf{x}(i)$被分配到一个唯一的聚类 C_k。

296 同质性（homogeneity）是这样实现的：选取适当的评分函数（下面将要讨论）并使每一点到它所属聚类矩心（centroid）的距离最小化。基于划分的聚类方法的重点就是评分函数，算法的其他部分与一般算法没有太大的不同。大多数情况下，把属于一个聚类的各个点的矩心或平均值作为这个聚类的代表，而且对被寻找聚类的形状没有明确的要求。然而，对于每个聚类一个“中心”的聚类表示来说，聚类间的边界是隐含定义的。例如，如果根据点 $\mathbf{x}$ 与聚类中心的欧氏距离来分配它，那么在 $\mathbf{x}$ 空间中聚类间的边界是线性的。

我们会看到在聚类分析中，最大化（或最小化）评分函数通常是计算复杂度很高的搜索问题，因此经常使用递归的启发式搜索方法（比如第8章中讨论的那些方法）来优化评分函数。

9.4.1　基于划分聚类的评分函数

人们使用了大量的不同方法来衡量聚类的质量，也开发出了各种算法来搜索最优的（或至少是好的）划分。

为了定义聚类的评分函数，我们需要先建立输入点间距离的概念。我们用 $d(\mathbf{x},\mathbf{y})$表示

点 $\mathbf{x}$，$\mathbf{y} \in D$ 的距离，我们简便地假定函数 d 是定义在 D 上的一种标距。大多数为聚类目的所设计的评分函数都着重于两个方面：每个聚类应该是紧凑的；各个聚类间的距离应该尽可能地远。实现这种直观概念的一种直接方法就是观察聚类 C 的聚类内差异（within cluster variation）$wc(C)$ 和聚类间差异（between cluster variation）$bc(C)$。聚类内差异衡量了聚类的紧凑性或密集度，而聚类间差异衡量了不同聚类间的距离。

假定我们已经为每个聚类选取了聚类中心（cluster center）$\mathbf{r}_k$。它可能是指定的有代表性的数据点 $\mathbf{x}(i) \in C_k$，按某种方式来说它定义了聚类的中心。如果输入点所属空间中取均值是有意义的，那么我们可以把聚类 C_k 中各点的矩心作为聚类的中心，也就是可以这样定义 $\mathbf{r}_k$：

$$\mathbf{r}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x} \tag{9.15}$$

297

其中 n_k 是第 k 个聚类中的点数。聚类内差异的一个简单尺度是看聚类内每一点到它所属聚类中心距离的平方和：

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{\mathbf{x}(i) \in C_k} d(\mathbf{x}, \mathbf{r}_k)^2 \tag{9.16}$$

当 $d(\mathbf{x},\mathbf{r}_k)$ 被定义为欧氏距离时，$wc(C)$ 被称为聚类内平方和（within-cluster sum-of-squares）。

可以把聚类间差异定义为聚类中心间的距离：

$$bc(C) = \sum_{1 \leqslant j < k \leqslant K} d(\mathbf{r}_j, \mathbf{r}_k)^2 \tag{9.17}$$

聚类 C 的总体质量可被定义为 $wc(C)$ 和 $bc(C)$ 的单调组合，比如二者的比 $bc(C)/wc(C)$。

上面的聚类内尺度从某种意义上说是全局性的：对于聚类 C_k，为了不会对这个尺度产生大的贡献，C_k 的所有点必须靠近聚类中心。因此使用这种衡量聚类紧密度的方法得到的聚类是球形的。下一小节要讨论的著名的 K-均值算法就是用每一组内的均值作为聚类中心并用欧氏距离定义 d，通过使公式 9.16 中的聚类内差异最小化来搜索测量值 $\mathbf{x}$ 在欧氏空间 R^p 中的聚类 C。

如果给定了一种候选聚类方案，那么求 $wc(C)$ 和 $bc(C)$ 值的复杂度如何呢？计算 $wc(C)$ 需要 $O(\sum_i |C_i|) = O(n)$ 次操作，而计算 $bc(C)$ 需要 $O(k^2)$ 次操作。因此，为一个聚类计算评分函数需要（至少原则上是）遍历整个数据一次。

对聚类内差异的另一种定义是考虑聚类内每个点与同一聚类内最近点的距离，并取这些距离中的最大值：

$$wc(C_k) = \max_i \min_{\mathbf{y}(j) \in C_k} \{d(\mathbf{x}(i), \mathbf{y}(j)) \mid x(i) \in C_k, x \neq y\} \tag{9.18}$$

这种最小距离（minimum distance）或单链接（single-link）标准得到的聚类是像腊肠的形状。我们在 9.5 节的层次凝聚聚类算法中还会讨论这一评分函数。

298

对于欧氏空间中的聚类 C，我们可以使用协方差的概念定义更通用的评分函数。我们可以为一个特定聚类 C_k 中的点定义一个 $p \times p$ 矩阵

$$\mathbf{W}_k = \sum_{\mathbf{x}\in C_k} (\mathbf{x}-\mathbf{r}_k)(\mathbf{x}-\mathbf{r}_k)^T \tag{9.19}$$

这是聚类 C_k 中点的（未正规化的）协方差矩阵。一个特定聚类的聚类间平方和就是这个矩阵的迹（trace）（对角线元素的和），$tr(\mathbf{W}_k)$，因此公式 9.16 中的总聚类内平方和可以表示为：

$$wc(C) = \sum_k tr(\mathbf{W}_k) \tag{9.20}$$

在这一框架下，如果设 $\mathbf{W}=\sum_k \mathbf{W}_k$，那么使 **W** “更小”（例如最小化 **W** 的迹或 **W** 的行列式）的评分函数会使数据的聚类更加紧凑。

我们可以定义一个矩阵 **B**，用来对与聚类中心的差异平方求和：

$$\mathbf{B} = \sum_{k=1}^{c} n_k (\mathbf{r}_k - \hat{\mu})(\mathbf{r}_k - \hat{\mu})^T \tag{9.21}$$

其中 $\hat{\mu}$ 是对 D 中所有数据点全局均值的估计。这个 $p\times p$ 的矩阵刻画了聚类均值彼此间的协方差（通过 n_k 加权）。例如，$tr(\mathbf{B})$是聚类均值相对于数据全局均值估计的加权距离平方和。因此，强调使 **B** 更大的评分函数使聚类均值更加分散。

我们再次强调很重要的但却经常被忽视的一点，评分函数的属性对从数据中发现的聚类的类型有非常大的影响。不同的评分函数（例如 **W** 和 **B** 的不同组合）会优先（prefer）选择不同的聚类结构。

基于 **W** 和 **B** 的传统评分函数是 **W** 的迹 $tr(\mathbf{W})$、**W** 的行列式$|\mathbf{W}|$、和 $tr(\mathbf{B}\mathbf{W}^{-1})$。$tr(\mathbf{W})$的不足是它依赖于个别变量的标度（scaling）。改变一个变量的单位就会得到一个不同的聚类结构。当然，可以通过在分析前把所有变量标准化来克服这个不足，但很多时候变量单位还
299 是和其他选项一样是任意的。使用 $tr(\mathbf{W})$标准得到的往往是紧凑的球形聚类，而且产生的组倾向于大体相等。这两个特征都使这种评分函数在区隔时很有价值，但对于发现自然的聚类（例如，在天文方面发现一个独立的非常小的聚类可能预示了一个重要的发现）就不太有吸引力了。

$|\mathbf{W}|$评分函数没有 tr(W)那种标度依赖性，因此它探测到的聚类也是椭圆形的，而且它也倾向于得到相同大小的聚类。已经有人提出把聚类大小考虑进来（例如，除以 $\prod n_k^{2n_k}$），以抵消聚类大小相等的倾向，但与其调整一个有缺点的方法还不如重新订立一个新的不同标准。还该注意到，如果认为数据是来自一个多元正态混合分布，那么本来的$|\mathbf{W}|$评分函数具有最佳性（optimality）的特征，而修改后的版本失去了这个特征。（当然，如果可以认为数据是这样产生的，那么我们可以考虑拟合一个正式的混合模型，就像 9.2.4 小节讲的那样）。

评分函数 $tr(\mathbf{B}\mathbf{W}^{-1})$也倾向于得到等大小的聚类，而且是大体相同的形状。注意因为这个评分函数等价于对 $\mathbf{B}\mathbf{W}^{-1}$ 的特征值进行汇总，它主要受最大特征值的影响，所以这个评分函数倾向于得到共线的（collinear）聚类。

从这些评分函数得到的聚类具有相似形状的特征并不是在所有情况下都是有吸引力的（实际上是很少情况下会喜欢这一特征）。基于对单独的聚类内矩阵 $\mathbf{W}_k$ 的其他组合方式的评分函数会好一些——例如 $\prod |\mathbf{W}_k|^{n_k}$ 和 $\prod |\mathbf{W}_k|^{1/p}$，其中 p 是变量数。 然而即使是这些评分函数，也有优先大小相似聚类的倾向。（与$|\mathbf{W}|$评分函数的情况类似，一种有助于克服这

一不足的方法是修改$\prod|\mathbf{W}_k|^{n_k}$，用$\prod n_k^{2n_k}$除以每一个$|\mathbf{W}_k|$。这相当于使不同聚类间的距离不同。）

这些方法的一个变体是不使用到聚类均值的距离平方和，而是使用到聚类中某个特定成员的距离平方和。那么搜索过程（见下文）便包含了一种对聚类中成员的搜索，搜索的目标是找到一个使评分函数最小化的成员。当然，通常可以不使用到聚类中心距离平方和这一尺度，而使用其他尺度。尤其是，把距离平方和替换为对距离的鲁棒估计可以减小孤立点的影响。也有人提出使用L_1标准作为距离尺度。典型的做法是用中值（median）向量作聚类“中心”。

300

可以把基于最小化聚类内距离平方和矩阵的方法当作是最小化到组矩心的偏差（deviation）。一种被称为最可能预测分类（maximal predictive classification）（是为在分类中使用二值变量分类学而开发的，但也适用于更广的范围）的技术也可以被看作是最小化到组“中心”的偏差，尽管对中心的定义不同。假定测量向量的每一个分量都是二值的——也就是，每一个对象都可以用一个向量来描述。并且现在我们要对这些对象进行聚类。对于每一组，我们可以定义一个向量，它是由组内每个变量的最常见值组成的。这个向量的模（而不是均值）将被作为组的“中心”。组成员到这个中心的距离是按它与中心向量取不同值的变量数来度量的。要优化的评分函数就是对象和它所属组的中心有不同值的总数。最佳的分组就是最小化这种差异总数的那一个分组。

下一节要描述的聚类分析的层次方法所建立的并不是对数据的单一划分，而是建立一种各个聚类（通常）相互嵌套的层次。那么我们就可以决定在哪里切割层次就可以做到按这种方式划分数据就能得到最合理的划分。然而，对于基于划分的方法，必须在开始时就决定要划分成多少个聚类。当然，我们可以多次重复划分过程，每次使用不同的聚类数量，但这还是需要我们有好的方法来在竞争的数目间做出选择。对于这个问题没有最佳解。当然，我们可以分析聚类评分函数如何根据聚类数量的增长而变化，但对于不同的聚类数量这种比较可能是不准确的。举例来说，或许随着数量的增长不论是否真的存在更好的聚类结构分数都显示出了明显的改善（比如说，聚类内距离平方和保证不会随着K的增长而增长）。对于被最优分割成K个聚类的多元均匀分布，评分函数$K^2|\mathbf{W}|$对于所有K逐步趋向于取同一个值，像这样的结果可以用作比较不同K值所产生划分的基础。

显然聚类分析很大程度上是一种数据驱动的工具，在这种分析中很少有一成不变的建模方法。然而一些学者已经尝试把它置于一种更可靠的基于模型的基础之上。例如，我们可以对这一过程进行补充：我们假定除了存在某种机制产生了聚类内的点之外，还存在一个随机过程产生了稀疏分布的点，而且对整个空间是均匀的。这使这种方法更不容易受孤立点的影响。一个更进一步的做法是使用特定的分布假定来对每个聚类内的数据分布建模——我们将在9.6节中再回过头来讨论这种基于模型的概率聚类。

301

9.4.2 基于划分聚类的基本算法

前面我们分析了很多种可以判断聚类质量的评分函数。那么用于优化这些评分函数的算法是什么样的呢？至少在理论上这个问题的答案是直截了当的。我们只要对把各个点分配到聚类C的可能分配方案所组成的空间进行搜索就可以了，搜索的目标是使评分函数最小化

（或最大化，视选取的评分函数定）。

可以认为这种搜索问题本质上是组合优化的一种形式，因为我们是对把 n 个对象放入 K 个类的分配方案进行搜索以最大化（或最小化）选取的评分函数。可能分配方案（聚类数据的不同方法）的数量可以近似为 K^n。例如，把 100 个对象分成两类有大约 $2^{100} \approx 10^{10}$ 种可能分配。因此，和我们已经看到的其他数据挖掘问题一样，直接的穷举搜索肯定是不可行的，除非要处理的数据集微乎其微。尽管如此，对于某些聚类评分函数，已经开发出这样的方法：它们穷举式的覆盖所有可能的聚类方法但不真的进行穷举搜索。这样的方法包括分支定界方法，该方法剪除比已经发现的备选方案更差的可能聚类，并不实际计算可能聚类的分数值。这样的方法尽管扩大了穷举方法的适用范围，但是即使对于中等大小的数据集仍是不适用的。因此，我们不再讨论这种方法。

不幸的是，并不是任何感兴趣的评分函数都存在闭合形式的解，也就是说，通常不存在直接的方法找到最小化评分函数的特定聚类 C。既然闭合形式的解不总存在，而且穷举搜索又不可行，那么我们就必须依赖于某种形式的系统搜索方法（在第 8 章中讨论了这样的搜索方法）。有必要强调，如果给定了特定评分函数，那么聚类问题就已经被简化为一种优化问
302 题，因此可以在优化的文献中找到大量的可能适用的方案。

基于局部搜索的递归改善算法在聚类分析中特别流行。其一般思想是：从随机选取的聚类开始；然后重新分配点使评分函数最大程度的增长（或降低）；然后再重新计算更新后的聚类的中心；再次重新分配点，如此继续直到评分函数没有变化或聚类成员没有变化。这种贪婪方法的优点是简单而且保证至少得到评分函数的局部最大值（最小值）。当然这种方法也有贪婪搜索算法的普遍缺陷，即无法知道收敛到的聚类 C 与最佳的可能聚类（所用评分函数的全局最优值）相比的好坏程度。

下面我们介绍一种运用这一原理的著名范例，也就是 K-均值算法（与第 8 章中介绍的 EM 算法有密切关系，而且在 9.2.4 节中我们提到过这种方法）。聚类的数量 K 是在算法运行前确定的（这是很多聚类算法的典型情况）。K-均值算法有很多种变体，基本的版本是从随机捡取 K 个聚类中心开始的，再根据欧氏距离把每个点分配到最接近其均值的聚类中，然后计算被分配到每个聚类的点的均值向量，并作为新的中心进行递归。具体算法是这样的：假定数据点 $D=\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，任务是找到 K 个聚类 $\{C_1, \cdots, C_K\}$：

```
for k = 1, …, K 令 r(k)为从 D 中随机选取的一个点;
while 在聚类 C_k 中有变化发生 do
    形成聚类:
    for k = 1, …, K do
        C_k = {x ∈ D | d(r_k, x) ≤ d(r_j, x) 对所有 j=1, …, K, j≠k};
    end;
    计算新的聚类中心:
    for k =1, …, K do
        r_k = C_k 内点的均值向量
    end;
end;
```

例 9.3 在美国国家航空和航行局的深层宇宙网络（NASA's Deep Space
303 Network）中，用来跟踪和与深层宇宙探测器通信的两个天线接收器是 34m 和 70m

的庞然大物，它的电机控制系统是这个网络中的一个重要部分。这个电机控制系统的发动机电流可以非常敏感的感受到天线运行情况的微小变化，因此可以作为在线监控和故障探测的根据。图 9-4 显示了来自 34m 深空网络天线的样本数据。每个二变量的数据点对应于发动机电流测量值的一个两秒时间窗，测量值已经用一个简单的自回归（autoregressive）（线性）时序模型模型化了，并且数据点的两维分别对应于自回归模型对特定窗格估计出的前两个系数。这个模型是每隔两秒与数据实时拟合一次，因此系数的变化反映了发动机电流测量值频谱特征的变化。

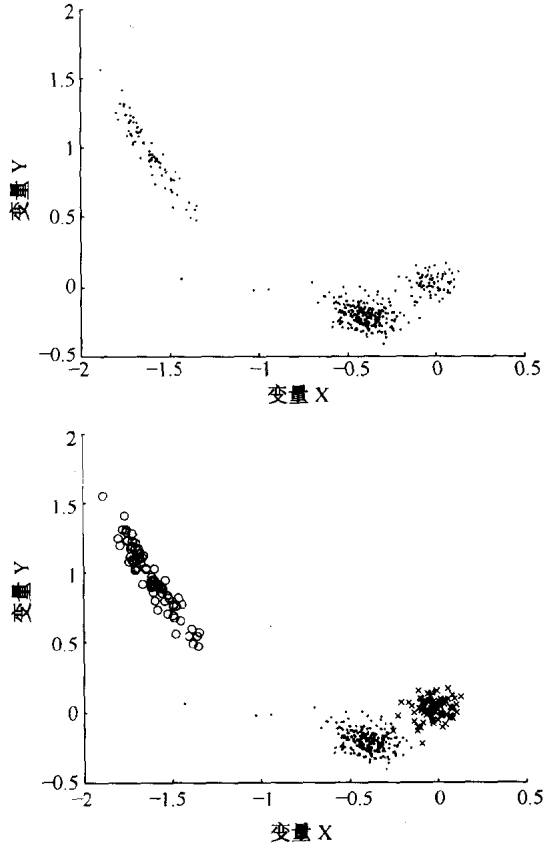


图9-4 天线数据。上图显示的是没有分类标签的数据点，下图使用了不同符号代表三个已知分类（点是正常情况，圆是转速计噪声。×是短路的情况）

图 9-4 下方散点图中的数据显示出了各个数据点所属的不同情况（三组，一组正常情况，两组故障情况）。图 9-5 显示了应用 *K*-均值算法对这些数据进行聚类的结果，使用的 *K* 值为 3，而且在聚类时删除了分类标签（也就是使用图 9-4 上图中的数据点作为 *K*-均值算法的输入）。算法的三个初始起点都位于中央（正常的）点群内，但仅经过 4 次迭代，算法就迅速收敛到一种聚类（聚类均值的变化轨迹画在图 9-6 中）。四次迭代后的最终聚类（图 9-5 中的下图）产生的三个分组与图 9-4 显示的已知分组非常接近。当然，这个数据集的分组是比较明显的，可以看出不同的故障情况与正常情况是分离的（特别是左面的转速计噪声）。尽管如此，我们看到了在该例中 *K*-均值算法迅速并准确地收敛到非常接近真实分组的聚类。

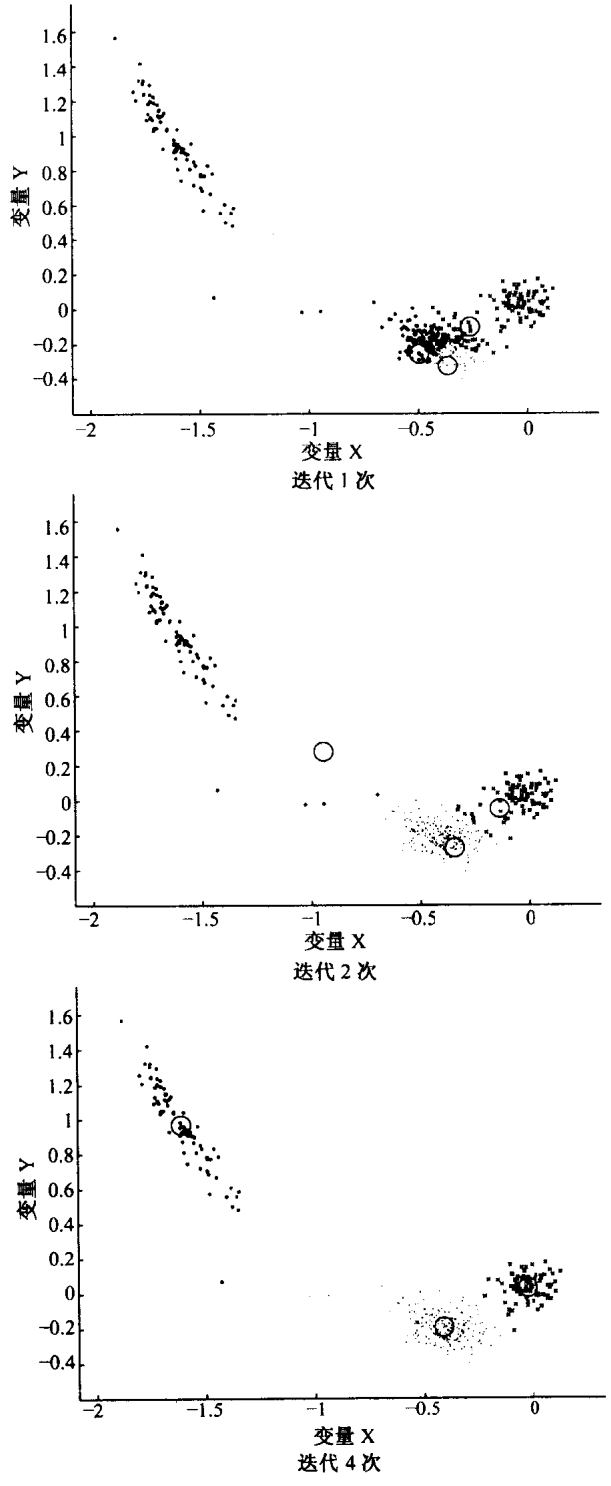


图9-5　在二维的天线数据上运行*K*-均值算法。这些散点图显示了*K*-均值算法经过不同次迭代后聚类均值的位置（大的圆圈），以及每一次迭代后数据点的分类（根据它最靠近的均值）（点、圆和×分别对应三个聚类）

K-均值算法的复杂度是 $O(KnI)$，其中 I 是迭代次数。也就是说，给定了当前的聚类中心 $\mathbf{r}_k$，我们可以只要遍历数据一次就能计算出所有的 Kn 个距离 $d(\mathbf{r}_k, \mathbf{x})$，并为每个 $\mathbf{x}$ 选择最短的一个；而后也可以在 $O(n)$ 次内完成对新聚类中心的计算。

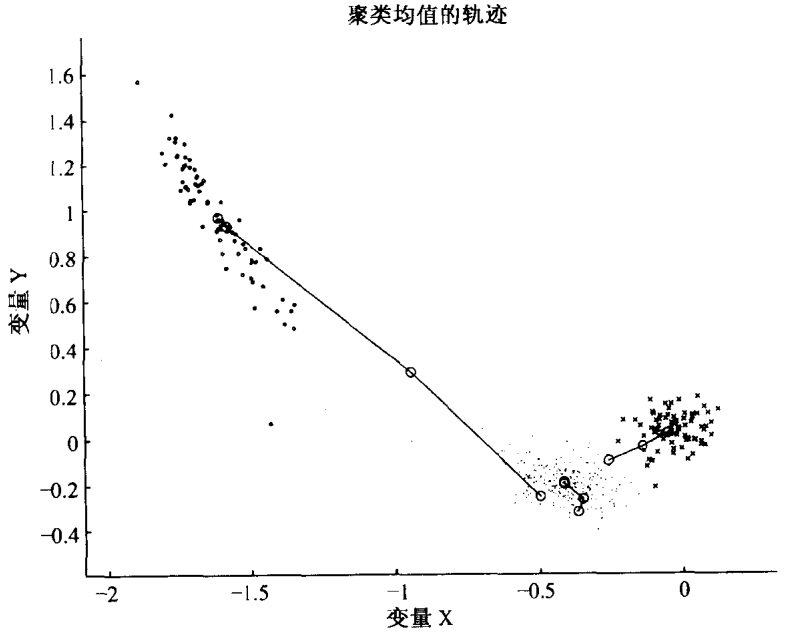


图9-6 在图9-5所示的K-均值迭代过程中三个聚类均值的移动轨迹

这种算法的一个变体是依次分析每个数据点，而且一旦有数据点被重新分配就更新聚类中心，反复地在数据点中循环直到解不再变化。如果数据集是很庞大的，那么还可以仅加入每个数据点，不要循环。进一步的扩展（例如 ISODATA 算法）包括分裂（splitting）和（或）融合（merging）聚类。注意在大量的基于划分的聚类算法中，很多都是围绕每次从聚类中增加或删除一个数据点这一思想的。已经开发出了一些高效的更新公式用来计算数据点进出聚类所引起的评分函数变化——尤其是对于包含了上一节讨论的 W 的所有评分函数。

K-均值算法的搜索过程局限于全部可能划分空间的一个很小部分。因此有可能因为算法收敛到评分函数的局部而非全局最小值而错过一个更好的解。一种缓解（如果没有解决的话）这一问题的方法是从不同的随机选取的起始点进行多次搜索。甚至可以更进一步的采用模拟退火策略（如第 8 章讨论的）来尽可能避免陷入评分函数的局部最小值。

聚类分析实质上是搜索庞大的解空间以优化特定评分函数的搜索问题。因此，很多数学规划方法已经应用到这个领域。这些方法包括线性规划、动态规划、以及线性和非线性整数规划。

聚类方法经常应用在庞大的数据集上。如果观察值的数量过于庞大以至于标准算法难以处理时，我们可以通过用紧缩表示替换对象组来压缩数据集。例如，如果有 100 个观察值在度量空间中很接近，那么我们可以把它们替换为它们的矩心所在的观察值并附带一个特征（所表示的这组点的半径）。而且只要对一些算法进行简单修改就可以使它们操作这种“紧缩的”表示。

304
1
307

9.5 层次聚类

基于划分的聚类方法是从指定数量的聚类开始搜索可能的点分配方案，来寻找使某个聚类评分函数最优的分配方法；与此不同，层次方法逐步地融合点或切分超聚类（supercluster）。事实上，根据这一基本思想我们可以把层次方法划分成两类：凝聚（agglomerative）（对应于融合）和分裂（divisive）（对应于切分）。在这两种方法中，凝聚方法更重要而且应用更广。注意可以把层次方法看作降低搜索规模的一种特定方式（而且特别直观）。它与本书其他部分介绍的用于建模的分步方法很类似。

层次聚类的一个明显特征是难以把模型从评分函数和用来决定最佳聚类的搜索方法分离出来。因此，在这一节中我们直接就把讨论的焦点集中到聚类算法上。我们可以把最终的层次看作一个从数据点到聚类的层次映射模型，但是这个模型（也就是聚类的“形状”）的属性是隐含在算法中的，不能明确地单独表示。类似的，这里的评分函数也没有全局评分函数的明确概念。而是使用不同的局部方法计算树上叶对（也就是，数据的特定层次聚类的聚类对）的分数来决定哪一对聚类是凝聚（融合）和分裂（切分）的最佳候选者。注意就像在基于划分的聚类中使用不同全局评分函数的情况一样，不同的局部评分函数会得到迥异的最终聚类。

可以很方便地用图形来显示聚类分析的层次方法，在图形中可以显示出融合（或切分）的整个过程。因为它的特征与树相似，所以这种图被称为树状图（dendrogram）。我们将在下面的例子中进一步说明。

聚类分析对于存在两个以上的变量情况特别有价值：如果仅有两个变量，那么我们就可
308 以目测一个散点图来寻找结构。然而，为了在一个我们可以看出真实情况的数据集上说明这种方法的基本思想，我们依然在一个二维数据集上介绍层次方法。这个数据集节选自 Azzalini and Bowman（1990）中给出的一个更大的数据集。图 9-7 显示了这个二维数据的散点图。纵轴是喷发持续时间，横轴是喷发的间隔时间，都是以分钟为单位的。图中的点是用数字给出的，这仅是为了在说明中把它们与树状图联系起来，并没有其他实质性的用途。

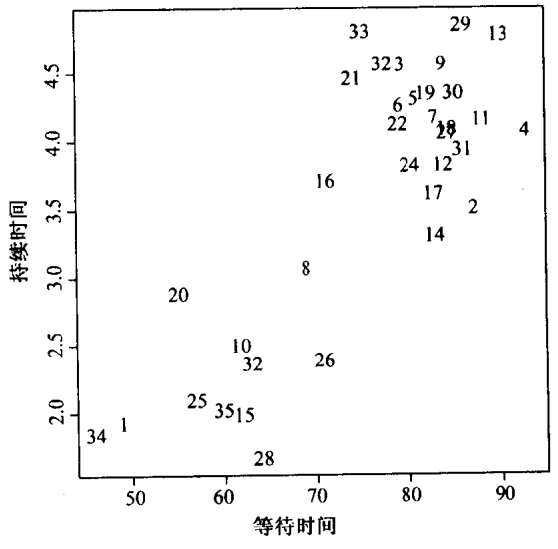


图9-7　美国黄石国家公园旧费斯富尔间歇喷泉的喷发持续时间与间歇时间（分钟）

图 9-8 显示了使用凝聚方法（把对聚类内平方和产生最小增长的两个聚类融合起来）所得到的一个树状图实例。树状图中交叉点（分支融合的地方）的高度显示了评分函数的值。因此，最初点 18 和点 27 的融合产生的增长最小。从图 9-7 我们可以看到这两点确实离得非常近（实际上是最近的）。注意视觉观察到的邻近程度是失真的，因为在页面上横向的刻度与纵向刻度相比被压缩了。接下来是对点 6 和点 22 的融合。又经过了一些对相邻点对的个体融合后，点 12 是与由点 18 和点 27 组成的聚类融合的，因为根据聚类标准这是产生最小增长的融合。继续这个过程直到最后一个融合：两个大的点聚类的融合。从树状图中可以清晰地看到这一过程。（没有必要总是聚类到这种程度。有时最终的融合是把大的聚类和一个单一的孤立点融合——就像我们后面将看到的那样。）显示在树状图中的层次结构也使我们可以清楚地看到我们可以在其他点停止这个过程，这相当于在某个高度水平切断树状图，这将得到多个聚类。

309
~
310

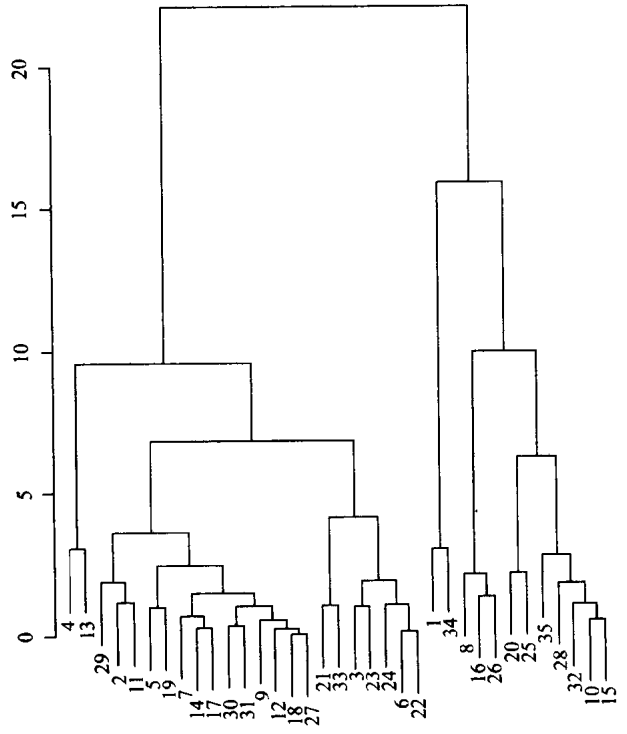


图9-8 聚类图9-7中的数据产生的树状图。这里使用的聚类标准是把对聚类内平方总和产生最小增长的聚类融合起来

9.5.1 凝聚方法

凝聚方法是以聚类间的距离尺度为基础的。实质上，对于给定的初始聚类，凝聚方法是把最邻近的聚类融合起来以降低聚类的数量。重复这个过程，每次都把两个最邻近的聚类融合，直到仅有一个包括所有数据点的聚类。通常这个过程的起始点是每个聚类仅含一个数据点的初始聚类，也就是从要被聚类的 n 个点开始。

假定给定 n 个数据点 $D=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$和一个用来衡量两个聚类 C_i和 C_j间距离的函数 $\mathcal{D}$

(C_i, C_j)。那么可以把聚类这个数据集的凝聚方法描述为：

```
for i=1, …,n 令 Ci={x(i)};
while 存在一个以上的聚类 do
    令 Ci和 Cj为使系统中任意两个聚类间的距离 D(Ck ,Ch) 最小化的两个聚类;
    Ci = Ci ∪ Cj;
    删除聚类 Cj;
end;
```

这种方法的时间复杂度如何呢？在开始时有 n 个聚类，结束时有 1 个聚类，因此在主循
311 环中有 n 次迭代。在第 i 次迭代中我们必须在 n–i+1 个聚类中寻找最靠近的两个聚类。我们马上会介绍很多种定义聚类间距离 $D(C_i,C_j)$的不同方法。但所有这些方法都需要在第一次迭代时找到最近的一对对象。除非我们知道对象间距离的特别知识，否则这个过程需要的时间是 $O(n^2)$，因此在大多数情况下，这个算法需要的时间是 $O(n^2)$，而且经常超过。还请注意这个算法的空间复杂度也是 $O(n^2)$，因为必须在算法开始时就计算出所有对象两两间的距离。因此这个算法对于 n 值很大的情况是不适用的。而且，解释一个庞大的树状图也是非常困难的（这与解释一个庞大的分类树是很困难的一样）。

注意到在凝聚聚类中，我们需要知道数据对象个体间的距离以开始聚类，而且在聚类过程中我们必须能够计算数据点分组间的距离（也就是聚类之间的距离）。因此这种方法的一个优势（例如，胜过基于划分的聚类）是不需要把每个对象表示为向量，只要我们能够计算对象间或对象集间的距离。因此，凝聚聚类为聚类那些不易表示为向量的对象提供了一种自然框架。一个很好的例子是聚类蛋白质序列，在这个问题中有几种不同的距离定义，比如两个序列间的编辑距离（edit-distance）（一种衡量从一个序列转换到另一个序列所需基本编辑操作次数的尺度）。

根据对象集（也就是聚类）间距离的一般情况，人们已经提出了很多种距离尺度。如果对象是向量，那么可以应用 9.4 节中描述的任一种全局评分函数，只要使用融合前分数和融合两个后分数的差异就可以了。

然而，局部对间（也就是各对聚类间）的距离尺度特别适合于层次方法，因为可以直接根据聚类内成员的对间距离计算这些尺度。*最近邻或单链接*方法是此类方法中最早和最重要的一种。这种方法把两个聚类间的距离定义为两个最近点（每个聚类中取一点）间的距离：

$$D_{sl}(C_i,C_j)=\min_{\mathbf{x},\mathbf{y}}\{\mathrm{d}(\mathbf{x},\mathbf{y})\mid \mathbf{x}\in C_i,\mathbf{y}\in C_j\} \tag{9.22}$$

312 其中 $d(\mathbf{x}, \mathbf{y})$是对象 $\mathbf{x}$ 和 $\mathbf{y}$ 间的距离。单链接方法是易受“链条”现象影响的（这可能是有利的也可能是不利的，依赖于我们的目的），以至于很长的点串被分配到同一个聚类中（这与紧凑的球形聚类形成对比）。这意味着单链接方法对于区隔问题的价值很有限，这也意味着这种方法对于数据的微小扰动和孤立点很敏感（这也既可能有利又可能不利，要看我们要实现什么目标）。这种单链接方法还有一个特征（这是该方法独有的——其他距离尺度不具有这一特征），就是如果有两对聚类是等距离的，那么先融合哪一对都无所谓。无论融合的顺序如何，最终的结果都相同。

对图 9-7 中的数据应用单链接方法得到的树状图显示在图 9-9 中。尽管对于这个特定的数据集单链接聚类和图 9-8 中的聚类非常相似，但是通常这两种方法产生的结果是有很大差异的。

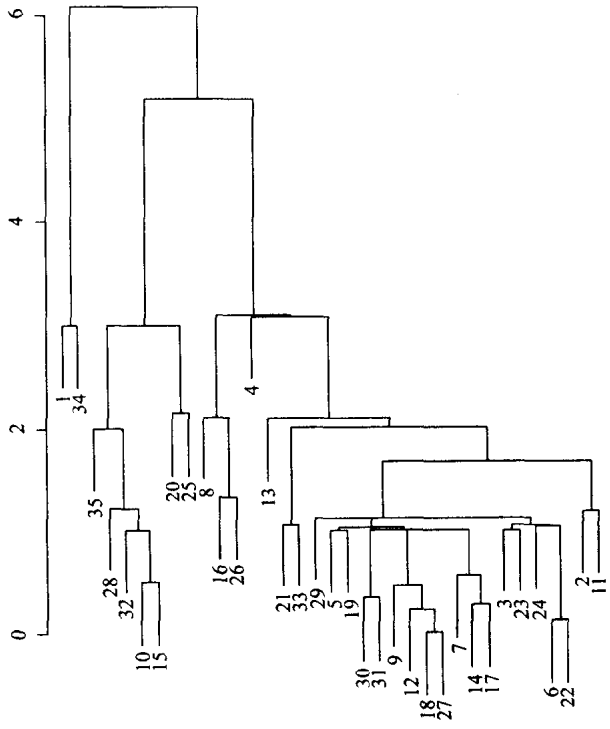


图9-9 对图9-7中的数据应用单链接方法得到的树状图

与单链接相反的另一种方法是最远邻（furthest neighbor），又叫完全链接（complete link），这种方法把两个聚类间的距离定义为两个最远离点（两个点分别来自两个聚类）间的距离：

$$D_{fl}(C_i,C_j)=\max_{\mathbf{x},\mathbf{y}}\{d(\mathbf{x},\mathbf{y})\mid \mathbf{x}\in C_i,\mathbf{y}\in C_j\} \tag{9.23}$$

其中 d(x, y)是对象 x 和 y 间的距离。对于向量对象，这个尺度迫使产生的分组所占的空间体积倾向于具有相等的大小（数据点数不一定相等），这使这一尺度特别适合于区隔问题。

其他介于单链接和完全连接之间的重要尺度包括（对于向量对象）矩心尺度（两个聚类间的距离是它们的矩心间的距离），组平均尺度（两个聚类间的距离是两个聚类中各点间所有距离（每个聚类一点）的平均值），和对于向量数据的 Ward 尺度（两个聚类间的距离是两个聚类中分别计算出的聚类内平方和的差异，聚类内平方和是从上面讨论的对两个聚类的融合中得到的）。这些尺度每个都有一些略微不同的特征，还存在一些变体，例如，用于向量数据的中值尺度忽视聚类的大小，把两个聚类组合的“中心”定义为连接两个部分的中心的直线的中点。因为数据挖掘就是要寻求新奇的发现，所以很有必要试验不同的尺度，以便突然发现一些异常而又有趣的信息。

313

9.5.2 分裂方法

选择变量的分步方法可以从没有变量开始逐步地加入变量（根据能否最大的改善模型），也可以从所有变量开始逐步地删除变量（删除的依据是使其对模型的损伤最小）。聚类分析也与此类似。聚类分析的凝聚方法相当于前一种情况，分裂方法相当与后一种方法。分裂方

法从一个由所有数据点组成的聚类起步，然后想办法把这个聚类分割成多个部分。而后再对这些分出的部分进行进一步分割，并重复这个过程直到满足需要为止。当然，当每个聚类仅包含一个数据点时这个过程会结束。

单分裂（monothetic divisive）方法每次使用一个变量拆分聚类（所以它们类似于第 5 章讨论的树分类方法）。这是限制必须要分析的可能划分数量的一种有效方式（不过是有局限的）。它还有一个优点是易于用树状图来描述得到的结果——每个节点处的分割都是仅以一个变量定义的。有时用关联分析（association analysis）这一术语来描述应用到多变量二值数据的单分裂过程。（这里关联一次的含义与第 5 章中介绍的“关联规则”中的关联用法不同。）

多分裂（polythetic divisive）方法基于对全部变量的综合分析进行拆分。在拆分中可以使用任何聚类间距离尺度。这种方法的难点在于如何选取向聚类中分配对象的各种可能方案——也就是，如何限制在可能划分的空间中的搜索范围。一种途径是一次一个的分析对象，并选取把它从主聚类放入子聚类就可以最大改善聚类分数的那一个。

一般来说，分裂方法的运算量比凝聚方法更大，而且应用不如后者广泛。

9.6　基于混合模型的概率聚类

我们还可以使用 9.2.4 节的混合模型以概率理论为背景建立起用于聚类的一般框架。这就是通常所说的基于模型的概率聚类（probabilistic model-based clustering），因为在这些方法中每一个聚类（分量）都对应于一个假定的概率模型。在这一框架中，我们假定数据来自于一个多元有限混合模型，模型的一般形式是：

$$f(\mathbf{x})=\sum_{k=1}^{K}\pi_k f_k(\mathbf{x};\theta_k) \tag{9.24}$$

314 ~ 315

其中 f_k 是分量分布。粗略地讲，建模的一般过程如下：对于给定的数据集 $D=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$，首先决定想要用多少个聚类拟合数据（即确定 K）；然后再为这 K 个聚类中的每一个选取参数模型（比如多元正态分布，这是最常用的方案）；最后再用 9.2.4 节的 EM 算法（第 8 章中描述的更加详细）来根据数据确定分量的参数 θ_k 和分量的概率 π_k。（当然，我们也可以根据数据来确定 K 的最佳值，在这一节的后面我们将回过头来讨论这个问题。）通常使用数据的似然（对于给定的混合模型）作为评分函数，尽管也可以使用其他的标准（比如所谓的分类似然）。一旦找到了混合分解模型，便可以把数据分配到各个聚类了——比如按它最可能来自的聚类分配每个点。

为了说明这一思想，我们把这种方法应用到一个事实上已经知道分类标签的数据集上，但是我们先把这些标签去掉，然后让算法来“发现”它们。

316

例 9.4　对于慢性铁缺乏贫血的人来说，他们的血红细胞量往往比正常的低，而且血红蛋白的浓度也较低。可以抽取血液样本以得出一个人的血红细胞平均量和血红蛋白浓度。图 9-10 所示为 182 个人的血红细胞平均量相对于血红蛋白浓度的散点图，图中的点带有通过诊断试验测得的结果标签。图 9-11 所示的是使用 K=2 的正态混合模型来拟合去掉标签的这些数据而得到的结果。从图中可以看出，两个

分量的混合模型捕捉到了数据的主要特征，如果不知道分类标签（也就是不进行化验分析）也将会给出较好的聚类。图 9-2 验证了似然（或者说对数似然，二者是等价的）不会随迭代次数下降。然而注意，收敛速度是变化的（nonmonotonic），也就是说，第 5 次到第 8 次迭代间的对数似然上升率放慢了，但第 8 次到 12 次间又上升了。

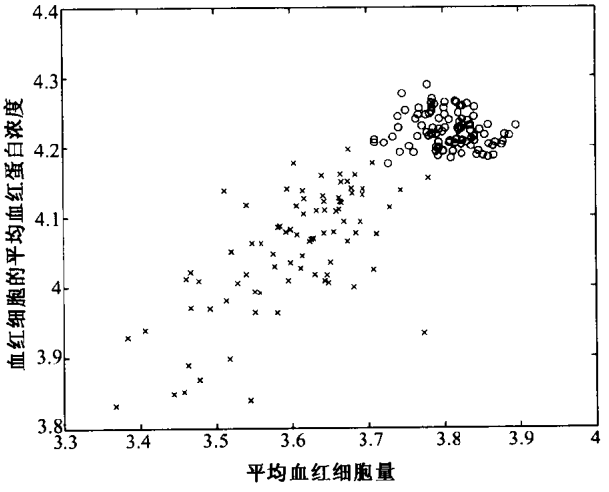


图9-10　182个人的血红细胞测量结果。从图中可以看出所有个体形成两组：健康的（圆圈）和患铁缺乏贫血的（叉号）

图 9-11 的血红细胞例子说明了概率方法具有以下几个特征：

- 概率模型为每个分量提供了完全的分布描述。例如，注意观察这个例子中两个分量的差异。正常的分量相对紧缩，这表明在正常人之间数据的变化性是相当低的。另一方面，患有铁缺乏贫血的聚类分散的很广，表明变化性较大。这正好与我们的直觉一致，而且这种信息对于科学的研究数据产生过程的基本机制是很有价值的。
- 对于给定的模型，每个个体（每个数据点）都有一个 K-分量的向量与其相关，这 K 个分量对应于它来自每个组的概率，而且通过贝叶斯法则可以很简单的计算出这个向量。对于血红细胞的例子，位于一个组或另一个组的大多数个体属于该组的概率都接近 1。但也有一定数量的个体（靠近两个云团交叉的地方）的概率接近 0.5——也就是说，它们属于哪一组具有不确定性。从探索数据的角度来看，这些数据点可能很有价值并值得进一步探测和更仔细的研究（因为这些个体可能已经开始患有铁缺乏贫血）。
- 在概率框架下，选择似然和 EM 算法分别作为评分函数和优化过程是很自然的。因为这样除了可以发挥大量现有算法库的作用外，还有很多完善的理论可以用来拟合模型参数。扩展 MAP 和贝叶斯估计（允许把以前的知识结合进来）也相当简单。
- 基本的有限混合模型为各种不同的扩展提供了很好的理论框架。举例来说，一种很有价值的想法是加入第（K+1）个噪声分量（比如用均匀密度），目的是“拾起”看起来不属于任何其他 K 个分量的孤立点和背景点；这个背景分量所对应的权π_{K+1}可以用 EM 算法直接从数据中学习得到。

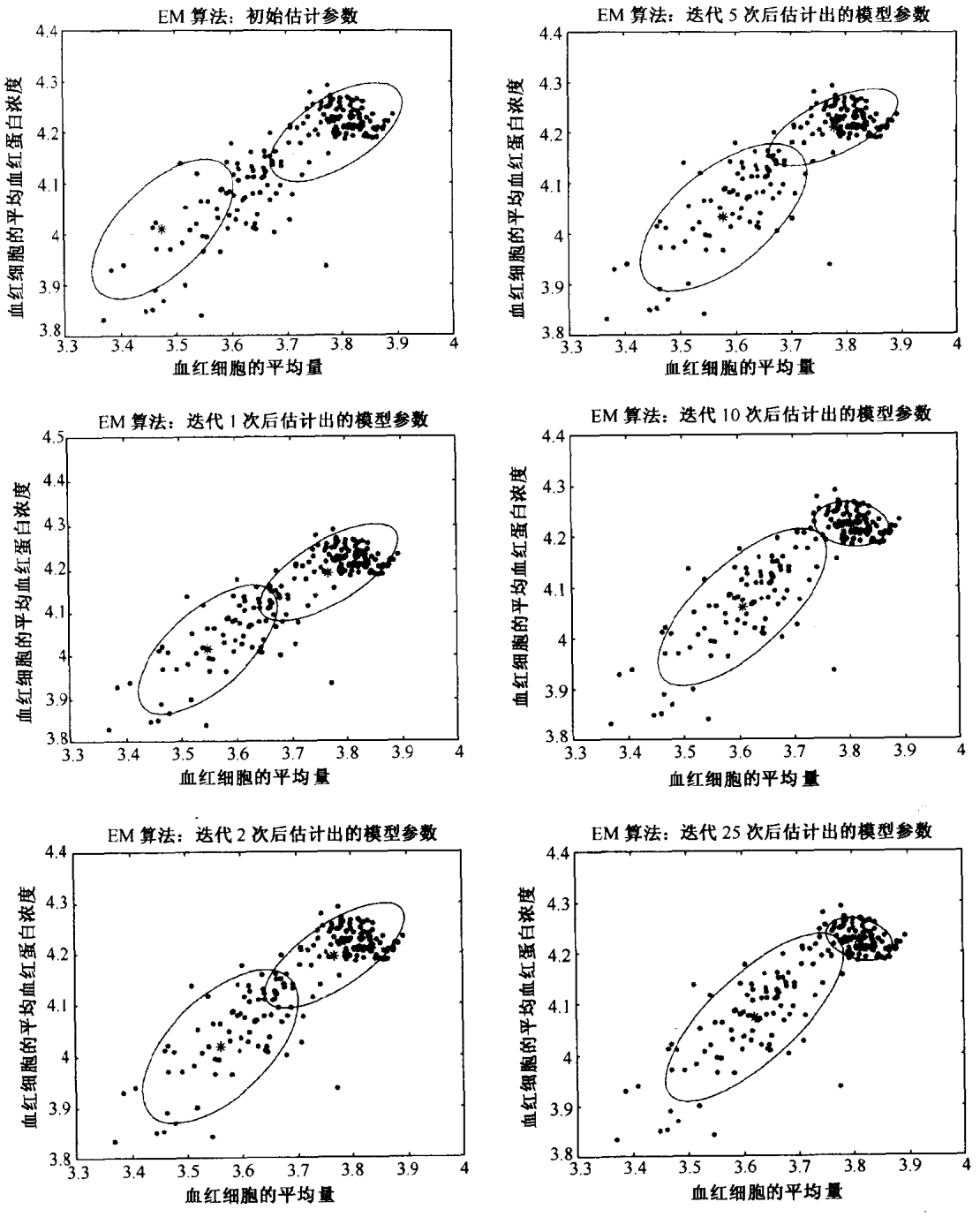


图9-11 对图9-10中的血红细胞测量数据运行EM算法的例子。上面的散点图中画出了在EM算法的各个不同阶段时两个拟合分量的3σ协方差椭圆和均值（顺序为从上到下，从左至右）

- 可以把这种方法扩展到数据并非是 p 维向量的形式。例如，我们可以在同样的 EM 通用框架下使用混合的概率序列模型（比如说混合马尔可夫模型）来聚类序列，使用混合的回归模型来聚类曲线，等等。

这些优势是以一定的代价得来的。主要的代价是要为每个分量假定参数模型。对于很多

问题来说，事先很难知道分布的形式是什么。因此，实际上基于模型的概率聚类仅适用于我们有理由相信假定的分布形式很合适的情况。对于前面的血红细胞数据，我们可以通过可视化的分析认为正态的假定是非常合理的。此外，既然两个测量值都是由来自很大的红细胞样本估计出的均值组成的，所以基本的统计理论也可以提示我们正态分布的假定很可能是非常合适的。

概率方法的另一个不足是相关估计算法的复杂性。不妨把 EM 算法和 K-均值算法加以比较。我们可以把 K-均值算法看作是对 EM 算法（当其应用于正态混合分量的混合模型（其中每个聚类的协方差矩阵都被假定为单位矩阵）时）的分步近似。然而，K-均值算法并不一直等到收敛完成才把点分配到各个聚类，而是在每一步进行分配。

例 9.5 假定我们有一个数据集，其中的每个变量 X_j 都取 0/1 值——比如说来自于一个大的交易数据集，x_j=1（或 0）表示一个顾客购买了商品 j（或者没有）。我们可以这样应用混合模型框架，假定对于给定的聚类 k，各个变量是条件独立的（根据 9.2.7 节的讨论），也就是说，我们有：

317 ~ 319

$$p_k(\mathbf{x};\theta_k)=\prod_{j=1}^{p}p_k(x_j;\theta_{kj})$$

要确定适合数据的模型，只要确定在第 k 个分量中观察到第 j 个变量取值为 1 的概率。我们把这个概率表示为 θ_{kj}，于是便可以把第 k 个分量的分量密度写为：

$$p_k(x_j;\theta_{kj})=\theta_{kj}^{x_j}(1-\theta_{kj})^{1-x_j},\quad 1\leqslant k\leqslant K$$

这是表示在混合模型的第 k 个分量中观察到 x_j 的概率的一种很方便的方式。$\mathbf{x}(i)$ 的完全混合公式就是这些分量分布的加权和：

$$p(\mathbf{x}(i))=\sum_{k=1}^{K}\pi_k p_k(\mathbf{x}(i);\theta_k) \tag{9.25}$$

$$=\sum_{k=1}^{K}\pi_k\prod_j\theta_{kj}^{x_j(i)}(1-\theta_{kj})^{1-x_j(i)} \tag{9.26}$$

320

其中 $x_j(i)$表示顾客 i 是否购买了商品 j。

这个模型的 EM 公式是非常简单的。令 $p(k\mid i)$为顾客 i 属于聚类 k 的概率。根据贝叶斯法则，对于给定的参数θ 集合，这个概率可以写作：

$$p(k\mid i)=\frac{\pi_k\prod_j\theta_{kj}^{x_j(i)}(1-\theta_{kj})^{1-x_j(i)}}{p(\mathbf{x}(i))} \tag{9.27}$$

其中，$p(\mathbf{x}(i))$ 的定义和公式 9.26 一样。计算 $p(k\mid i)$ 需要 $O(nK)$ 步，因为要对每个个体 i 和每个聚类 k 进行计算。计算这些隶属概率相当于求解这个问题的 E 步骤。

M 步骤就是在已知一个顾客属于聚类 k 的前提下，加权估计出这个顾客购买商品 j 的概率：

$$\theta_{kj}^{new}=\frac{\sum_{i=1}^{n}p(k\mid i)x_j(i)}{\sum_{i=1}^{n}p(k\mid i)} \tag{9.28}$$

在这个问题中，$x_j(i)$是按照 $p(k\mid i)$ 加权的，也就是个体 i 产生于聚类 k 的概率（根据模型）。个体 i 购买特定商品 j 就相当于把商品按比例（即权 $p(k|i)$，$1\leqslant k\leqslant K$）分配到 K 聚类模型。M 步骤需要 $O(nKp)$ 次操作，因为分子上的加权求和是针对所有 n 个个体的，并对于每个聚类 k，和 p 个参数中的每一个（在独立的模型中每个变量需要一轮操作）。如果我们在 EM 算法中进行 I 次迭代，那么基本的复杂度就是 $O(IKnp)$，可以看作是数据矩阵大小的 KI 倍。

然而，对于现实的位于磁盘上的庞大数据集合来说，对整个次数据集做 I 次扫描是不可行的。因此人们已经开发出了概括聚类表示的各种技术，在聚类过程中这些技术实际上起到了压缩数据集的作用。例如，在混合建模中，很多数据点很早就“趋向于”某个分量；也就是说，它们隶属于这个分量的概率接近于 1。因此可以更新这些点的隶属关系并在以后的迭代中忽略这些点。类似的，如果一群点的隶属关系始终是一样的，那么用一个简短的描述来表示这些点。

为了对概率聚类加以总结，我们考虑如何根据数据来选取最佳 K 值的问题。注意随着 K
321 （聚类的数量）的增长，似然的最大值对 K 的函数是不会下降的。因此，似然本身不能直接告诉我们从 K 的角度来说哪个模型最接近真实的数据产生过程。而且，由于和混合似然有关的技术原因，通常的假设检验方法（例如检验一个分量相对两个分量的效果，两个相对于三个，等等）不适用。然而，人们已经开发出了很多其他的巧妙方法，这些方法很大程度上是以对理论分析的近似为基础的。我们这里介绍其中三种应用较广而且比较通用的技术：

惩罚似然：它的基本思想是从似然的最大化值中减掉一项。其中贝叶斯信息判据（BIC）应用得很广。这里

$$S_{BIC}(M_K)=2S_L(\hat{\theta}_K;M_K)+d_K\log n \tag{9.29}$$

其中，$S_L(\theta_K;\ M_K)$是负对数似然的最小化值，d_K 是参数个数，二者都是相对于具有 K 个分量的混合模型来说的。具体做法是先求出从 K=1 到某个 $K_{\max}$ 时上式的值，然后把对应于最小值的 K 值作为最可能的值。BIC 的原始推导是基于回归框架中的极限理论，这个论据和混合建模并不严格一致。然而，已经发现这个技术在实践中的效果很好，而且具有比下面将介绍的其他方法计算代价小的优点。图 9-12 所示为相对血红细胞数据的 BIC 评分函数曲线，图中指出了 K=2 时模型最佳（回忆前面曾指出根据独立的医学知识（化验）我们知道这些数据属于两组，所以这个结果是令人满意的）。有人还提出了很多其他的惩罚项（参见第 7 章），但是在聚类中 BIC 似乎是使用最广泛的。

二次采样技术：我们还可以利用二次采样的思想使用自展方法或者是交叉验证方法来估计哪个 K 值是最佳的。这些技术的不足是明显比 BIC 需要更大的运算量——举例来说，10-折交叉验证所需的时间是 BIC 方法的 10 倍。然而，这些方法切实的提供了对模型质量的直接评估，不需要像 BIC 方法那样作出假定。

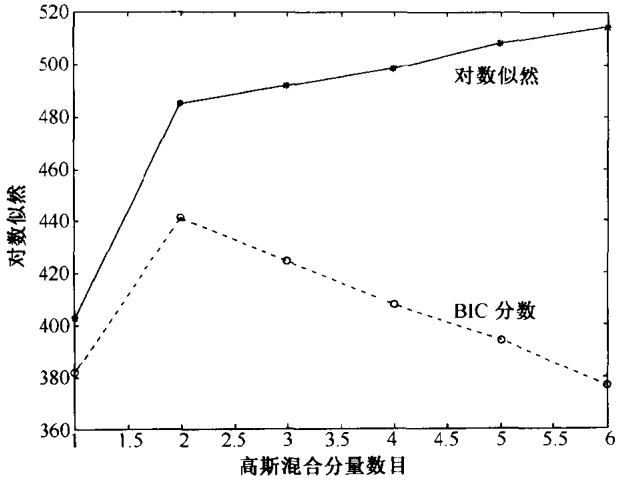


图9-12 对数似然分数和BIC分数相对于正态分量数目的函数曲线。所针对的问题是把混合模型拟合到图9-11所示的血红细胞数据

贝叶斯近似：完全贝叶斯解需要估计分布 $p(K|D)$ ——也就是每个 K 值对于给定数据的概率，这需要以普通方式对参数的所有不确定性进行积分。当然这在实践中是不可行的（不 322
要忘记我们是在 d_K 维空间中求积分），因此人们寻找了很多种不同的近似。既使用了解析的近似方法（比如对后验分布的最频值进行拉普拉斯近似）也使用了采样技术（比如马尔可夫链 Monte Carlo）。对于模型中有很多参数的庞大数据集，采样技术在运算方面来讲可能不可行，因此解析方法应用得更广泛。例如 Cheeseman and Stutz（1996）的 AUTOCLASS 算法，使用对后验分布的解析近似来选取模型。基于惩罚的 BIC 函数也可以被看作是对完全贝叶斯方法的近似。

从某种意义上来讲，建立在混合分解模型上的概率建模比聚类分析更具一般性。聚类分析的目的就是产生对现有数据的一个划分，而混合分解产生了对数据潜在分布的完全描述（这个分布是由很多分量构成的）。一旦确定了这些分量的概率分布，那么就可以把数据集中的点分配到最可能产生它的聚类。我们也可以按另一种方式来看这个问题：聚类分析的目的是把数据自然的划分到产生它的区域，在这些区域中，各个点很接近或者说聚集在一起，从而使聚类之间出现相对稀疏的区域。从概率密度的角度来看，这相当于低密度的谷底分隔出很多高密度的区域，所以从根本上讲这个概率密度函数是多峰的（multimodal）。然而，混合分布很可能是单峰的，尽管它是由几个分量组成的。

下面考虑两个分量的一元正态混合模型的情况。显然，如果这两个均值是相等的，那么这个模型是单峰的。事实上，当均值不等时这个混合模型为单峰的充分条件（对于所有的混合比例）是：$|\mu_1-\mu_2| \leqslant 2\min(\sigma_1, \sigma_2)$。而且，对于均值和偏差值的所有情况在二分量的正态混合模型中都存在某些混合比例值使这个混合模型是单峰的。这意味着如果均值足够靠近，那么就只存在一个聚类，虽然存在两个分量。这时我们还是可以使用混合分解来进行聚类，也就是把每个点分配到它最可能来自的聚类，但是这不可能是有价值的聚类。 323

9.7 补充读物

Ross（1997）介绍了参数概率建模，Everitt and Dunn（1991）介绍了多元数据分析中的一般概念。介绍混合分布的普通教材包括 Everitt and Hand（1981）、Titterington, Smith and Makov（1985）、McLachlan and Basford（1988）、Böhning（1998）和 McLachlan and Peel（2000）。Diebolt and Robert（1994）给出了用通用贝叶斯方法进行混合建模的一个例子。从统计角度讨论图形模型的文献包括 Whittaker（1990）、Edwards（1995）、Cox and Wermuth（1996）和 Lauritzen（1996）。Pearl（1988）和 Jensen（1996）特别从表示和计算角度对这些模型进行了讨论，Jordan（1999）收编了有关从数据中学习图形模型的最新研究论文。Friedman and Goldszmidt（1996）和 Chickering, Heckerman and Meek（1997）给出了从数据中学习图形模型的详细具体算法。Della Pietra and Lafferty（1997）介绍了马尔可夫随机场在文本建模中的应用，Heckerman et al.（2000）描述了使用马尔可夫随机场来实现基于模型的协同过滤（collaborative filtering）。Bishop, Fienberg and Holland（1975）是对数线性模型方面的标准参考书。

目前，已经有很多关于聚类分析的著作。推荐读者参考 Anderberg（1973）、Späth（1985）、Jain and Dubes（1988）和 Kaufman and Rousseeuw（1990）。分切（dissection）和寻找自然划分的差异不总是很容易察觉到的，然而这个差异是非常重要的，因此不该忽视。区分这两种情况的著作包括 Kendall（1980）、Gordon（1981）和 Späth（1985）。Marriott（1971）证明了对于多元均匀分布的最佳划分来说 $K^2\mathrm{tr}(\mathbf{W})$标准趋向于常量。Krzanowski and Marriott（1995）中的表 10-6 列出了基于 **W** 的各种聚类标准的更新（updating）公式。Gower（1974）开发了极大预测分类方法。Koontz, Narendra and Fukunaga（1975）和 Hand（1981）描述了如何使用分枝定界思想来扩大穷举方法在评估可能聚类时的适用范围。MacQueen（1967）介绍了 *K*-均值算法，Hall and Ball（1965）介绍了 ISODATA 算法。Kaufman and Rousseeuw（1990）描述了一种变体，每个聚类的中心点是聚类的一个元素，而不是元素的矩心。Rao（1971）回顾了应用在聚类分析中的数学规划方法的早期著作，Mangasarian（1996）浏览
324 了这方面的一些更新的成果。

关于聚类分析的单连接方法的最早著作是 Florek et al.（1951），Sibson（1973）对推动这一思想起到了重要作用。Lance and Williams（1967）给出了通用的公式，对于计算目的来说是很有价值的，同时作为特例介绍了单连接和完全连接方法。聚类分析的中值方法应该归功于 Gower（1967）。Lambert and Williams（1966）描述了单分裂划分的“关联分析”方法。聚类分析的多分裂方法应该归功于 MacNaughton-Smith et al.（1964）。重叠聚类方法应归功于 Shepard and Arabie（1979）。

当然还有其他的聚类形式。比如，Karypis and Kumar（1998）讨论了基于图的聚类算法。Zhang, Ramakrishnan and Livny（1997）描述了适用于非常庞大数据集的聚类框架。聚类的应用更是数不尽的。Lapointe and Legendre（1994）使用聚类分析对威士忌酒做了研究。Eisen et al.（1998）阐述了层次凝聚聚类在基因数据上的应用。Zamir and Etzioni（1998）介绍了专门用于聚类网络文档的聚类算法。

Titterington, Smith and Makov（1985）和 McLachlan and Basford（1987）以混合模型为背景讨论了概率聚类。Banfield and Raftery（1993）提出了一种新的想法：通过向整个空间

迭加一个产生底层随机点的泊松过程来增加一个遍及整个空间的“聚类”，目的是缓解由于孤立点所造成的聚类失真。以下著作介绍了基于模型概率聚类的一些最新成果：Celeux and Govaert（1995）、Fraley and Raftery（1998）和 McLachlan and Peel（1998）。Poulsen（1990）、Smyth（1997）、Ridgeway（1997）和 Smyth（1999）介绍了混合模型在聚类序列方面的应用。针对参数模型的基于混合曲线（curves）聚类最早出现在 Quandt and Ramsey（1978）和 Späth（1979）中，后来 Gaffney and Smyth（1999）将其推广到非参数模型的情况。Jordan and Jacobs（1994）介绍了一种对标准混合方法的推广，即一种被称为“mixtures of experts”的基于混合结构，该结构为函数近似提供了一种通用的基于混合方法的框架。

以下文献介绍了混合模型分量数目试验和有关研究：Everitt（1981）、McLachlan（1987）和 Mendell, Finch and Thode（1993）。Shibata（1978）给出了对 BIC 判据的早期推导。Kass and Raftery（1995）概括了这一领域的一些最新成果，包括 BIC 在很多模型选取任务中的应用。决定混合模型分量数量的自展法是由 McLachlan（1987）引入的，McCulloch（1996）和 325
McLachlan and Peel（1997）又作了进一步的完善。Smyth（2000）介绍了针对这一问题的交叉验证方法。Cheeseman and Stutz（1996）描述了一种针对基于模型聚类问题的通用贝叶斯框架，Chickering and Heckerman（1998）通过试验比较了不同贝叶斯近似方法对于求解分量数 K 的效果。

Neal and Hinton（1998）、Bradley, Fayyad and Reina（1998）和 Moore（1999）介绍了用来提高基本 EM 算法在处理大数据集时速度的不同技术。

Cheng and Wallace（1993）介绍了层次凝聚聚类的一个有趣应用：对地球大气层的空间测量数据进行聚类。Smyth, Ide and Ghil（1999）利用正态混合模型给出了分析这些数据的另一种方法，并且使用交叉验证似然给出了对 Cheng 和 Wallace 得到聚类的定量验证。McLaren（1996）描述了血液学中的混合模型。Wedel and Kamakura（1998）以非常广的视角浏览了混合模型在客户建模和市场方面的大量应用。Cadez et al.（2000）描述了马尔可夫混合模型在聚类网络文档方面的应用（以来自大量网络日志中的页面请求序列为基础）。

Smyth（1994）更详细地描述了图 9-4 的天线数据，Cadez et al.（1999）介绍了图 9-10 326
中的血红细胞数据。

第 10 章　用于分类的预测建模

10.1　预测建模概览

第 9 章中讨论的描述建模就是对数据进行概括，从而可以更方便地使用数据，或者可以使我们更好的理解事物的运转机制。相对而言，预测建模的目标更加明确：其目的就是在给定其他变量值的条件下对我们感兴趣的未知变量值做出预测。这样的例子包括：根据患者的一系列化验结果给出对他的诊断；在已知顾客购买了其他商品的前提下，估计出他们购买产品 A 的概率；或者给定目前和过去的道·琼斯指数值，预测出从现在开始在将来 6 个月中该指数的值。

在第 6 章中我们讨论了很多可以用作预测模型的基本函数形式。在这一章和下一章中，我们将更详细的分析这些模型，并讨论把这些模型拟合到数据上的具体算法和判据。

可以把预测建模看作是学习一种映射，这种映射把输入测量向量 $\mathbf{x}$ 的集合映射到标量的输出 y（也可以把输出映射为向量，但是在实践中标量的情况更普遍）。在预测建模中，训练数据 D_{train} 是由测量对（pairs）构成的，每个对由一个向量 $\mathbf{x}(i)$和一个对应的“目标”值 $y\ (i)$（$1 \leqslant i \leqslant n$）组成。因此，预测建模所要做的就是根据训练数据估计出一种映射或者函数 $y = f(\mathbf{x};\theta)$，可以在给定测量值输入向量 $\mathbf{x}$ 和模型 f 的参数 θ 的情况下预测出 y 值。回忆前面讨论过的内容，f 是模型结构的函数形式（第 6 章），θ 是 f 中的未知参数，θ 值是通过在数据上最小化一个合适的评分函数（第 7 章）来确定的，而搜索最佳 θ 值的过程实际上就是数据挖掘算法的基本部分（第 8 章）。因此我们需要作出三项选择：一种特定的模型结构（或一族模型结构），一个评分函数和一种用来在一族模型中发现最佳参数和模型的优化策略。

在数据挖掘问题中，由于我们事先对函数 $f(\mathbf{x};\theta)$ 的形式知之甚少，所以为 f 选取比较灵活的函数形式或模型是有吸引力的。另一方面，正如第 6 章中所讨论的，较简单的模型具有更加稳定和更易于解释的优势，而且还经常可以为更复杂的模型结构提供函数分量。对于预测建模来说，评分函数的定义通常是相当直接的，它的典型定义就是模型 $\hat{y}(i) = f(\mathbf{x}(i);\theta)$的预测值与 $y(i)$的真实值之间差异的函数——即：

$$\begin{aligned} S(\theta) &= \sum_{D_{train}} d\big(y(i), \hat{y}(i)\big) \\ &= \sum_{D_{train}} d\big(y(i), f(\mathbf{x}(i);\theta)\big) \end{aligned} \tag{10.1}$$

其中，累加是针对训练数据集 D_{train} 中各个元组（tuples）$(x\ (i);y\ (i))$ 的，函数 d 则定义一种标量性的距离，比如对 y 取实数值的情况可以使用误差平方；对 y 为范畴型变量的情况下可以使用一种指示函数（关于这部分内容的详细讨论请参见第 7 章）。接下来数据挖掘算法的核心问题实际就是使函数 S 相对 θ 最小化，这个最小化过程的细节是由距离函数的特征和 $f\ (\mathbf{x};\ \theta)$ 的函数形式共同决定的，因为二者共同决定了 S 如何依赖于 θ（参见第 8 章的

讨论）。

为了比较各个预测模型，我们需要估计它们对于“样本外数据”的性能，所谓“样本外数据”就是没有被用来构建模型的数据（否则，就像前面所讨论的，估计出的性能很可能是有偏的）。这种情况下，我们可以重新定义评分函数 $S(\theta)$，并在验证数据集上估计模型的性能，也可以使用交叉验证（cross-validation）或者惩罚性的评分函数（penalized score function），总之不能直接在训练数据上估计模型的样本外性能（如第7章中所讨论的）。

我们在第6章中曾经指出，根据 Y 是范畴型的还是实数型的，可以把预测建模分成两种不同的主要任务。对于范畴型的 Y，称其为分类（classification）（或者叫有指导的分类
328 （supervised classification），目的是区别于那些按第一个实例定义类的问题，比如聚类分析）；对于 Y 取实数值的情况，称其为回归（regression）。本章集中讨论分类问题，下一章将集中讨论回归问题。尽管我们可以在同一个通用的框架下同时讨论这两种形式的建模（它们都建立在很多相同的数学和统计基础之上），但是为了内容组织的方便我们把分类和回归各立一章。有必要提醒读者，本章中讨论的很多模型结构都有一种适用于下一章的回归问题的对应形式。例如，我们在这一章中讨论的树结构也可以用于回归。同样，我们在回归中讨论神经网络，但它也可用于分类。

在这两章中，我们覆盖了许多用于分类和回归问题的流行方法——模型结构-评分函数-优化技术这三者的常用组合。这些算法的分类特征往往是和用于预测的模型结构（比如树结构、线性模型、多项式等等）密切联系的，从而使本章大体上是根据不同的模型结构来划分的。尽管模型、评分函数和优化策略的某些特定组合已经很流行（“标准的”数据挖掘算法），但要记住，第5章介绍的数据挖掘算法中的通用化哲学是很重要的；因为对于一个特定的数据挖掘问题我们总是应该根据具体的应用来谨慎地选择裁剪模型、评分函数和优化策略，而不是把现成的技术照搬照抄。

10.2 分类建模简介

第6章中我们介绍了用于分类的预测模型，这里我们简要回顾一些基本概念。在分类问题中，我们希望学习到一种从测量值向量 $\mathbf{x}$ 到分类变量 Y 的映射。这个被预测的变量通常被称为分类变量（class variable）（理由显而易见），而且为了表示的方便在本章的其余部分我们将使用变量 C（而不是用 Y）来表示这个分类变量，C 的取值为 $\{c_1, \cdots, c_m\}$。观察或测量到的变量 $X_1,\cdots,X_p$ 有多种称呼，比如特征、属性、解释（explanatory）变量、输入变量等等
329 ——在本章中我们则使用输入（input）变量这一通用术语。我们用 $\mathbf{x}$ 表示 p 维向量（就是说，我们用它代表 p 个变量），它的每一个分量可以是实数型、序数型、范畴型等等。$x_j(i)$ 是第 i 个输入向量的第 j 个分量，其中 $1 \leqslant i \leqslant n$，$1 \leqslant j \leqslant p$。在我们介绍性的讨论中，我们隐含假定使用“0-1”损失函数（参见第7章）作为评分函数，也就是不管正确的分类和预测出的分类是什么，我们认为：正确预测的损失是0；错误分类预测的损失是1。

下面将从两种不同但又相关的分类观点开始：决策边界（或者判别）观点和概率观点。

10.2.1 判别分类和决策边界

在判别框架下，分类模型的输入为以向量 $\mathbf{x}$ 表示的测量值，产生的输出是集合 $\{c_1, \cdots, c_m\}$ 中的一个符号。下面以一个仅有两个实数值输入变量 X_1 和 X_2 的简单问题为例考虑映射函数

的特征。映射实际上是在（X_1,X_2）平面上产生一个分段的固定曲面；也就是说，仅在一定的区域内曲面的取值为 c_1。取值为 c_1 的所有区域的联合称为 c_1 类的决策区域（decision region）；就是说，如果输入的 **x** (*i*) 落入这个区域，那么它的分类就被预测为 c_1（并且这个区域的补（complement）是所有其他类的决策区域）。

知道了决策区域在（X_1, X_2）平面中的位置等价于知道了各区域间的决策边界（decision boundary）或者决策曲面（decision surface）。因此我们可以把学习分类函数 f 的问题看作学习各分类之间决策边界的问题。在这个前提下，我们可以从考虑可用来描述决策边界的数学形式入手，比如直线或平面（线性边界）、低次多项式这样的弯曲边界或者其他特殊的函数。

大多数现实的分类问题中，各个类在空间 **X** 中是不可能被完全分割的。也就是说，在 **X** 的某些（或许所有）值处出现的成员可能属于多个类——尽管各类的成员在任何给定的 **x** 值处发生的概率是不同的。（正是因为概率不同，才使我们可以作出分类。概括地讲，我们就是把点 **x** 分配到它最可能属于的类别。）各个类相互“重叠”的事实导致了另一种观察分类问题的方式，不再把注意力集中于决策曲面，而是寻找一个使类别间的某个分割尺度最大 330
化的函数 f (**x**; θ)。这样的函数被称为*判别函数*（discriminant functions）。事实上，最早的正式分类方法——*费歇尔线性判别分析方法*（Fisher's linear discriminant analysis method）（Fisher,1936）——就是完全基于这种思想的：它寻找变量 **x** 的线性组合，以使各个（两个）类别间的差异最大化。

10.2.2 分类的概率模型

设 $p(c_k)$为随机选取的对象或个体 i 来自 c_k 类的概率。如果假定各个分类互不包含并且没有遗漏，那么$\sum_k p(c_k) = 1$。但是事实并非总是如此——例如，如果一个人患有一种以上的疾病（各个类是互相包含的），那么我们可以把这个问题模型化为多个二分类问题（“患有或没有患有疾病 1”，“患有或没有患有疾病 2”等等）。可能还有一种疾病没有在我们的分类模型中（即类别集合是不完全的），在这种情况下我们可以向模型中加入一个额外类别 c_{k+1}，对应于“所有其他的疾病”。尽管这些潜在的实践复杂性是客观存在的，但是除非特别指出，我们在这一章中都使用“互不包含和没有遗漏”这一假定，因为这是被实践所广泛接受的，而且是概率分类的核心基础。

设想有两种类别：男性和女性，并用 $p(c_k)$（k=1，2）来表示在受精时一个人接受到适当的染色体而成为男性或女性的概率。因此如果我们根本没有任何其他关于个体 i 的信息（没有测量值 **x**(*i*)），那么 p (c_k) 就是个体 i 属于分类 c_k 的概率。有时把这个 p (c_k) 称为 c_k 类的“先验概率”，因为它代表了在观察到向量 **x** 之前的类隶属关系概率。在很多情况下从数据中估计 p (c_k) 都是相当简单的：如果已经抽取了总体的随机样本，那么 p (c_k) 的最大似然估计就是 c_k 在训练数据集中发生的频率。当然，如果已经采用其他的样本模式，事情可能更复杂一些。例如，在一些医疗问题中故意的从每一个类别中抽取等数量的样本是很常见的，这样就必须使用某种其他的手段来估计这些先验概率了。

我们假定属于类别 k 的对象或个体的测量向量 **x** 符合某种分布或密度函数 p (**x**|c_k, θ_k)，其中θ_k 是未知的参数，它控制了 c_k 类的特征。例如，对于多变量的实数值数据，可以假定 331
每个类别的模型结构都是多元正态分布，而且参数θ_k 代表每个类的均值（位置）和方差（范围）特征。如果各个均值离的足够远，而且方差足够小，那么我们可以希望各个类在输入空

间中是被充分分隔的（well separated），这使分类的误分类率（或错误率）很低。通常的问题是预先既不知道 x 分布的函数形式又不知道分布的参数。

一旦已经估计出了 $p(\mathbf{x}|c_k, \theta_k)$ 分布，那么我们就可以应用贝叶斯定理得到后验概率：

$$p(c_k|\mathbf{x}) = \frac{p(\mathbf{x}|c_k,\theta)p(c_k)}{\sum_{l=1}^{m} p(\mathbf{x}|c_l,\theta_l)p(c_l)} \qquad 1 \leqslant k \leqslant m \tag{10.2}$$

后验概率 $p(c_k|\mathbf{x}, \theta_k)$ 隐含的把输入空间 **x** 分割成 m 个决策区域，每个决策区域具有相应的决策边界。例如，对于二分类的情况（m=2），决策边界就是 $p(c_1|\mathbf{x},\theta_1) = p(c_2|\mathbf{x},\theta_2)$ 的轮廓线。注意如果我们能知道真实的后验分类概率（而不是不得已估计出它们），我们就可以对给定的测量 **x** 做出最优的预测。例如，对于所有错误都导致相等损失的情况，我们应该把后验概率 $p(c_k|\mathbf{x})$ 最高的类别 c_k 作为对任意给定的 **x** 值类别预测（因为这个类别最可能产生这个数据）。我们说这种方案最优，是从没有其他的预测方法可以做的更好这个意义上来讲的——所以这并不意味这种方法不会产生预测错误。事实上，在大多数实际的问题中，最优分类方案的错误率都不为 0，这是由分布 $p(\mathbf{x}|c_k, \theta_k)$ 的重叠所导致的。这种重叠意味着属于某一类的最大概率 $p(c_k|\mathbf{x}) < 1$，因此尽管 **x** 点的最优分类决策是选取 c_k，但是 **x** 点的数据来自其他类的概率 $1-p(c_k|\mathbf{x})$ 是不为 0 的（可能性较小）。把这一讨论扩展到整个分类空间，并相对 **x** 平均（或者对离散值的变量求和），便得到了贝叶斯误差率（Bayes Error Rate）：

$$p_B^* = \int (1 - \max_k p(c_k|\mathbf{x}))p(\mathbf{x})d\mathbf{x} \tag{10.3}$$

这是可能的最小误差率。没有其他分类器可以对新的未见过数据达到更低的预期误差率。在实践中，贝叶斯误差是求解问题的可能最佳分类器的误差下限。

332

例 10.1　图 10-1 显示了一个简单的假想例子，在这个例子只有一个预报变量 X（水平轴）和两种分类。上部的两幅图分别显示了数据是如何在类别 1 和类别 2 中分布的。这两幅图显示了分类的联合概率和变量 X 的关系 $p(x, c_k)$，k=1,2。两者都相对于 X 的一定范围具有均匀的分布；类别 c_1 比类别 c_2 趋向于具有较小的 x 值。沿 x 轴有一个区域（值 x_1 和 x_2 之间）是被两个类都覆盖的。

下图显示了类别 c_1 的后验分类概率 $p(c_1|x)$，它是根据上面两幅图中给定的类分布通过贝叶斯法则计算的。对于 $x \leqslant x_1$ 的 **x** 值，后验概率为 1（因为在这个区域仅有类别 2 可以产生数据），对于 $x \geqslant x_2$ 的 **x** 值，后验概率是 0（因为仅有分类 2 可以产生这一区域的数据）。重叠区域（x_1 和 x_2 之间）的后验概率大约是 1/3（根据贝叶斯法则），因为类别 2 在这个区域的可能性大约是类别 1 的两倍。因此，对于任何 $x \geqslant x_2$ 的 **x** 值，贝叶斯最优决策是类别 c_2（那些 $p(x, c_1)$和 $p(x, c_2)$都为 0 的区域是没有意义的，这些区域的后验概率是未定义的）。然而，注意在 x_1 和 x_2

333

间，关于这一区域内的一个给定 x 值应该属于哪个类别存在一种根本的模棱两可性；也就是说，尽管 c_2 是更有可能的分类，但是仍有 1/3 的机会属于 c_1。事实上，既然在这个区域存在 1/3 的可能作出错误的决策，而且我们从图中可以推测出 **x** 值落入这一区域的机会是大约是 20%，那么这个问题的贝叶斯误差率的大约为 $20/3 \approx 6.67\%$。

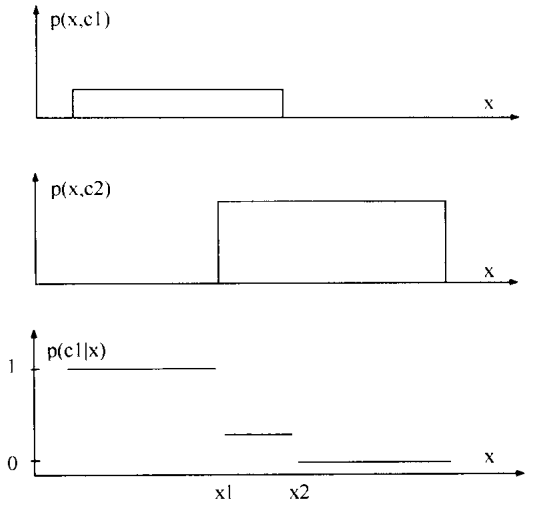


图10-1 演示后验分类概率的一个简单例子。这个问题只涉及两种分类，而且输入是一维的

现在考虑 **x** 为二元并且一个类的成员完全被其他类的成员包围的情况。在这种情况下，两个 X 变量中没有一个可以单独产生具有 0 误差率的分类规则，但是基于两个变量的联合规则可能达到 0 误差率（如果使用了合适的模型）。在实践中经常会发生类似的情况（不过很少有达到 0 误差率这么极端的情况）：新的变量加入了信息，所以我们可以通过加入额外的变量来降低贝叶斯误差率。这样便提出一个问题：我们为什么不干脆在分类问题中增加测量，直到误差率足够低？这个问题的答案就是第 4 章和第 7 章中讨论的偏差-方差（bias-variance）原则。尽管如果我们向模型中加入更多的变量，那么贝叶斯误差率会停留在同一个值处或者降低，但是事实上我们不知道最优的贝叶斯分类器或贝叶斯误差率。我们必须根据有限的训练数据集评估分类规则。如果变量数增多了而训练数据的点数不变，那么训练数据表示潜在分布的精度就更差了。增加变量是可能使贝叶斯误差率下降，但是我们对它的近似更差了。当变量数上升到某一点后，我们对潜在分布近似的不足胜过了贝叶斯误差率的降低，因此分类规则开始退化。

正确的做法是谨慎的选取变量，我们需要的变量是把它们放在一起时可以很好地分割各个类别。寻找合适的变量（或者少数的特征——变量的组合）是实现有效分类的关键。对于复杂的和潜在维数很高的数据（比如图像）这一点更加突出，在这些问题中人们公认发现合适的特征对分类精度的作用远远超过了选取不同的分类模型可能造成的影响。在这一背景下，一种数据驱动的方法是使用像交叉验证误差率这样的评分函数来引导搜索寻找特征组合——当然，对于某些分类器这可能需要非常高的运算开销，因为可能需要对每个要分析的子集重新训练分类器，而且这些子集的数量是 p（变量数）的组合。

334

10.2.3 建立实际的分类器

尽管这个框架从理论角度给出了分类的内部细节，但是它没有提供分类建模的规范框架。也就是说，它没有告诉我们如何构建分类器，除非我们恰好完全知道 $p(\mathbf{x}|c_k)$ 的函数形式（这在实践中是很少见的）。我们可以列出三种基本的方法：

1. 判别法：在这种方法中直接对决策边界建模——也就是说，直接把输入 $\mathbf{x}$ 映射到 m 个类标签 $c_1, \cdots, c_m$ 之一。根本不直接尝试对分类条件或后验分类概率建模。这种方法的例子包括感知器（参见 10.3 节）和更具一般性的支持向量机（参见 10.9 节）。

2. 回归法：在这种方法中显式地建立分类的后验概率模型，并且选取这些概率中的最大值（可能通过一个代价函数加权）用作预测。在这个范畴里最广泛应用的技术被称为 logistic 回归，我们将在第 10.7 节中讨论。注意决策树（例如第 5 章中的 CART）既可以被看作判别法（如果树仅在每个叶子节点给出预测分类），又可被看作回归法（如果树还在每个叶子节点处提供后验分类概率的分布）。

3. 分类-条件法：这种方法显式的建立分类的条件分布 $p(\mathbf{x}|c_k, \theta_k)$，并且和对 $p(c_k)$的估计一起利用贝叶斯法则（公式 10.2）推导出每种分类 c_k 的 $p(c_k|\mathbf{x})$，然后选取其中的最大值（可能被代价加权），就像回归法那样。我们可以把这种模型叫做“产生”模型，因为我们精确地指出了（通过 $p(\mathbf{x}|c_k,\theta_k)$）每个类的数据是如何产生的。使用这种方法的分类器有时也被称为贝叶斯分类器，因为它使用了贝叶斯定理，但是从第 4 章讨论的贝叶斯参数估计的正规含义来看它们不一定是贝叶斯的。在实践中，用在公式 10.2 中的参数估计 $\hat{\theta}_k$ 经常是通过每个分类 c_k 的最大似然估计出的，然后再“插入”到 $p(\mathbf{x}|c_k,\theta_k)$ 中。还有一种可选的贝叶斯方法是对 θ_k 求平均。此外，$p(\mathbf{x}|c_k, \theta_k)$ 的函数形式非常广泛——任
335 何参数的（例如正态），准参数的（例如有限混合），或非参数的（比如，核函数）模型都可以用来估计 $p(\mathbf{x}|c_k, \theta_k)$。而且从理论上讲，可以为每个类 c_k 使用不同的模型结构（例如可以用正态密度来对 c_1 类建模，用指数混合来对 c_2 类建模，用核密度估计来对 c_3 类建模）。

例 10.2　选取最可能的类别通常等价于选取 k 的值使判别函数 $g_k(\mathbf{x}) = p(c_k|\mathbf{x})$最大化，$1 \leqslant m$。很多时候把判别式重新定义（通过贝叶斯法则）成 $g_k(\mathbf{x})=\log p(\mathbf{x}|c_k)p(c_k)$更为方便。对于多元实数值数据 $\mathbf{x}$，一种普遍使用的分类条件模型是第 9 章中讨论的多元正态模型。如果我们对正态多元密度函数取对数（以 e 为底），并忽略不包含 k 的项，便得到了以下形式的判别函数：

$$g_k(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\mu_k)^T \sum_k^{-1}(x-\mu_k) - \frac{p}{2}\log|\Sigma_k| - \log p(c_k) \quad 1 \leqslant k \leqslant m \quad (10.4)$$

一般的情况下，每一个 $g_k(\mathbf{x})$包含各个 $\mathbf{x}$ 变量的二次项和成对的乘积。任何两个分类 k 和 l 的决策边界是由关于 $\mathbf{x}$ 的方程式 $g_k(\mathbf{x}) - g_l(\mathbf{x}) = 0$ 所定义的，而且通常这也是 $\mathbf{x}$ 的二次方程。因此，多元正态分类条件模型通常产生二次的决策边界。实际上，如果限制每个分类 k 的协方差矩阵是一样的（$\Sigma_k=\Sigma$），那么很容易证明此时函数 $g_k(\mathbf{x})$被简化为 $\mathbf{x}$ 的线性函数，并且产生的决策边界是线性的（也就是，它们定义了 p 维空间的超平面）。

图 10-2 显示了用多元正态分类模型拟合第 9 章的血红细胞数据的结果。μ_k、Σ_k 和 $p(c_k)$的最大似然估计（参见第 4 章）是使用来自两个分类 k=1，2 的数据得到的，然后再把这些估计插入贝叶斯法则来确定后验概率函数 $p(c_k|\mathbf{x})$。我们可以看到得到的决策边界形式上确实是二次的，这和理论分析是一致的（画出的另两条后验概率等高线也是如此）。注意等高线沿着从健康类（图中的叉号）均值向外的方向下降

的相当快。因为健康类（c_1 类）通常比贫血类（c_2 类，图中的圆圈）具有更小的方差，所以最优的分类器（假定正态模型）产生的决策边界完全包围了健康类。

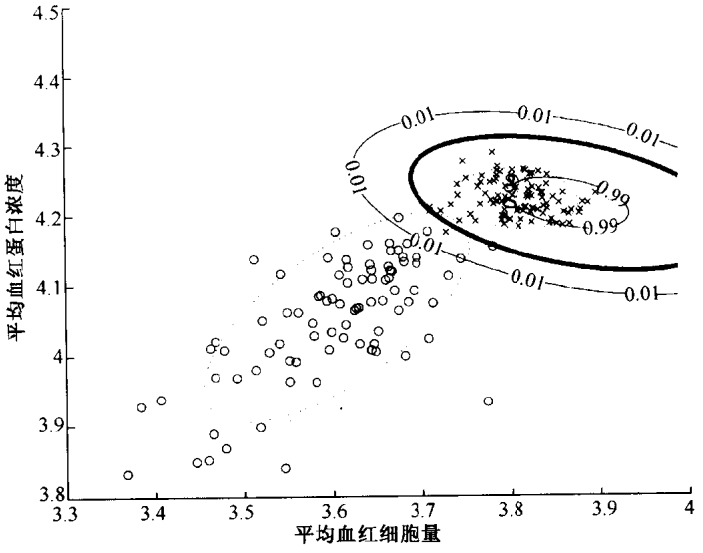


图10-2 后验概率$p(c_1|\mathbf{x})$的等高线。图中画出的是$p(c_1|\mathbf{x})$的后验概率等高线，其中c_1是代表第9章讨论的血红细胞数据的健康类的分类标签。粗线是（$p(c_1|\mathbf{x}) = p(c_2|\mathbf{x}) = 0.5$）的决策边界，另两条等高线对应于$p(c_1|\mathbf{x}) = 0.01$和$p(c_1|\mathbf{x}) = 0.99$。图中还标出了原始的数据点供参考，以及和每个类拟合的协方差椭圆（虚线所示）

图 10-3 显示了应用到不同数据集的同一分类过程（多元正态，最大似然估计）的结果。在这个例子中，来自印第安比马人数据集（最初在第 3 章中讨论的）的两个变量被用作分类变量，其中取值为 0 的有问题的测量值（被认为是孤立点，参见第 3 章）被预先删除了。与图 10-2 中的血红细胞数据不同，这两种分类（健康的和患糖尿病的）在这两维中是严重重叠的。被估计的协方差矩阵Σ_1和Σ_2未加约束，使得产生的决策边界和后验概率等高线也是二次的。重叠的程度被反映在后验概率的等高线上，现在这些等高线比图 10-2 中的更加分散（它们下降得很慢）。

可以看到，不论是判别法还是回归法，它们的焦点都在于各类之间的差异（或者更正式的说，它们的焦点都集中在以 **x** 值为条件的类隶属关系概率上），然而分类条件法（或者说产生法）的焦点是 **x** 相对各个类的分布。有时把焦点直接集中在类隶属关系概率的方法称为诊断（diagnostic）方法，而把焦点集中在 **x** 值分布的方法称为采样（sampling）方法。当然，所有这些方法都是相互联系的。分类条件（即产生法）方法与回归方法的相同之处在于前者最终产生的也是后验分类概率，不同之处在于前者以一种非常特别的方式（借助贝叶斯法则）计算这些概率，而回归法在如何对后验概率建模方面是没有限制的。类似地，不论是回归法还是分类条件或产生法都隐含地包括了决策边界，也就是说，在决策模式中它们都把输入 **x** 映射到 m 个类别中的一个，并且这都是在概率框架下完成的，但是“真正的”判别分类器并不一定要这样做。

336 ~ 338

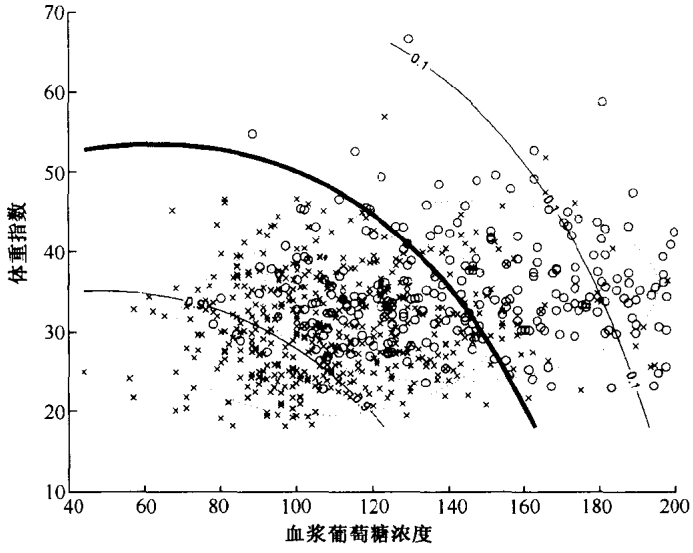


图10-3　后验概率$p(c_1|\mathbf{x})$的等高线[⊖]。其中c_1是第3章的印第安比马人数据的糖尿病类的标签。粗线是（$p(c_1|\mathbf{x}) = p(c_2|\mathbf{x}) = 0.5$）的决策边界，另两条等高线对应于$p(c_1|\mathbf{x}) = 0.1$和$p(c_1|\mathbf{x}) = 0.9$。和每个类拟合的协方差椭圆是用虚线画出的

我们将在接下来的几节中讨论每种方法的实例。至于哪种类型的分类器在实践中工作得更好，这要看问题的特征。对于某些应用（比如医疗诊断），产生后验分类概率的分类器比仅仅给出分类标签的分类器更有价值。基于分类条件分布方法的优点是提供了对每个类别的完整描述（例如，它提供了一种检测孤立点——看起来不属于任何已知类别的输入 $\mathbf{x}$——的方式）。然而正如第 9 章中所讨论的，在高维情况下要精确的估计函数 $p(\mathbf{x}|c_k,\theta_k)$是非常困难的（即便可能）。在这种情况下，判别式方法可能工作得更好。通常，基于分类条件分布的方法需要拟合的参数最多（因此会产生最复杂的模型），回归方法需要拟合的参数少一些，而判别式模型在这三种方法中最少。可以这样直观地解释这一点，最优的判别式模型仅包含了最优的回归模型中的信息的一个子集（决策边界，而不是完全的分类概率曲面），而最优的回归模型所包含的信息又比最优的分类条件分布模型要少。

10.3　感知器

感知器（perceptron）是最早的以计算机为基础的自动分类规则之一。感知器的目标直接指向学习决策边界曲面，从这个意义上来说它是判别式规则的一个实例。感知器模型最初是受简单的神经网络模型的启发而产生的，是用来模拟人类大脑中真实神经元的“积聚（accumulate）——引发（fire）”这一阈值行为，在第 11 章的回归模型中我们将讨论更通用的和最新的神经网络模型。

339　形式最简单的感知器模型（用于二分类）就是关于测量 $\mathbf{x}$ 的线性组合。我们定义 $h(\mathbf{x}) = \sum w_j x_j$，其中 w_j（$1 \leqslant j \leqslant p$）是模型的权（参数）。人们通常向该式中加入一个值固定为 1 的附加输入，目的是向模型中加入一个可训练的偏移项。分类是通过把 $h(\mathbf{x})$和一个阈值进行比较

⊖　译注:体重指数(又称 BMI 指数)=体重(KG)/身高(M)^2。

而实现的，为了简便，我们在这里把阈值取为 0。如果对于所有第一类点都有 $h(\mathbf{x})>0$ 并且对于所有第二类点都有 $h(\mathbf{x})<0$，我们便可以完全分隔这两个类。我们可以通过寻找一系列权使训练集合中的所有点都满足上面的条件来实现这种分隔。这意味着评分函数就是使用给定的一组权 $w_1,\cdots,w_{p+1}$ 来分类训练数据时的错误分类数。我们把第二类数据点的测量值 x_j 转换为 $-x_j$，那么问题会变的更加简单。因为这样一来我们就仅需要一组权，这组权满足对于所有的训练集合中的点都有 $h(\mathbf{x})>0$。

可以通过依次分析训练数据点来估计权 w_j。我们从一个初始的权集合开始，用它分类第一个数据点。如果分类是正确的，那么权保持不变。如果分类是不正确的，那么一定是 $h(\mathbf{x})<0$，于是对权进行更新使 $h(\mathbf{x})$ 上升。通过向权中加一个误分类向量可以很容易地做到这一点，也就是这样定义权更新法则：$\mathbf{w}=\mathbf{w}+\lambda\mathbf{x}_j$。这里λ是一个小的常数。对所有的数据点重复这个过程，如果需要还可以重复对训练数据集操作几次。可以证明，如果两个类别是被线性决策曲面完全分割的，那么只要选取的λ值足够小，这个算法最终便可以找到分隔曲面。这种更新算法使人联想起第 8 章中讨论的梯度下降技术，不过这种方法没有实际计算梯度，而是逐步降低误差率评分函数。

当然也可以使用其他的算法，而且当两个类别不是线性可分时，其他方法确实更有吸引力。在这种情况下，要分析错误分类误差率是相当困难的（因为它不是关于权的平滑函数），因此经常使用误差平方评分函数来代替：

$$S(\mathbf{w})=\sum_{i=1}^{n}\left(\sum_{j=1}^{p+1}w_j x_j(i)-y(i)\right)^2 \tag{10.5}$$

因为这是一个二次的误差函数，所以关于权向量 $\mathbf{w}$ 的函数具有唯一的全局最小值，而且最小化的方法也是相当简单的（要么使用第 8 章中的局部梯度下降方法，要么使用线性代数直接求闭合形式的解）。

340

这种基本的感知器思想有很多变体，例如处理两种以上分类情况的扩展。感知器模型的吸引力在于它易于理解和分析。然而，在实践中它的适用性受到决策边界是线性（也就是输入空间 $\mathbf{X}$ 中的超平面）的这一事实所限制，因为现实的分类问题可能需要更复杂的决策曲面以实现更低的分类误差率。

10.4 线性判别式

可以把线性判别式分类方法看作是感知器模型的“孪兄弟”，因为它们都属于线性分类器这一家族。判别式方法基于一种简单但很有用的概念：搜索可以最佳分隔各个类别的变量线性组合。可以把线性判别式看作是判别法的一种，因为它既不显式地估计分类隶属关系的后验概率，也不估计分类的条件分布。Fisher（1936）是最早讨论线性判别式分析的著作之一（对于二分类的情况）。设 $\hat{\mathbf{C}}$ 为按如下方式定义的组合样本协方差矩阵：

$$\hat{\mathbf{C}}=\frac{1}{n_1+n_2}\left(n_1\hat{\mathbf{C}}_1+n_2\hat{\mathbf{C}}_2\right) \tag{10.6}$$

其中 n_i（$1\leqslant i\leqslant 2$）是每个类的数据点数，$\hat{\mathbf{C}}_i$（$1\leqslant i\leqslant 2$）是每个类的 $p\times p$ 样本协方差矩阵（估计的）（和第 2 章中的定义相同）。为了表征任意 p 维向量 $\mathbf{w}$ 的分隔能力，费歇尔这样定义了一个标量的评分函数：

$$S(\mathbf{w})=\frac{\mathbf{w}^T\hat{\mu}_1-\mathbf{w}^T\hat{\mu}_2}{\mathbf{w}^T\hat{\mathbf{C}}\mathbf{w}} \tag{10.7}$$

其中 $\hat{\mu}_1$ 和 $\hat{\mu}_2$ 分别是第 1 类和第 2 类数据中 $\mathbf{x}$ 的 $p\times 1$ 均值向量。分子项是每个类的均值投影差异，我们希望这一项最大化。分母是数据在 $\mathbf{w}$ 方向投影的估计方差，并考虑了不同变量 x_j 可能既有各自的不同方差，又有相互间的不同协方差。

给定了评分函数 $S(\mathbf{w})$，接下来的问题就是确定使这个表达式最大化的方向 $\mathbf{w}$。实际上，存在一个闭合形式的解，从而可以得到最大化以上表达式的 $\mathbf{w}$，它是由下式给出的：

341

$$\hat{\mathbf{w}}_{lda}=\hat{\mathbf{C}}^{-1}(\hat{\mu}_1-\hat{\mu}_2) \tag{10.8}$$

分类新数据点的方法就是把它投影到最大化分隔的方向，如果 $\mathbf{x}$ 满足下式便把它分类到第一类中：

$$\mathbf{w}_{lda}^T\left(\mathbf{x}-\frac{1}{2}(\hat{\mu}_1-\hat{\mu}_2)\right)>\log\frac{p(c_1)}{p(c_2)} \tag{10.9}$$

其中 $p(c_1)$和 $p(c_2)$分别是两种类别的概率。

图 10-4 中显示了把费歇尔线性判别式应用到前面讨论的有关贫血的二分类问题的结果。可以看出这种线性决策边界对训练数据的分隔效果不如图 10-2 中的二次边界好。

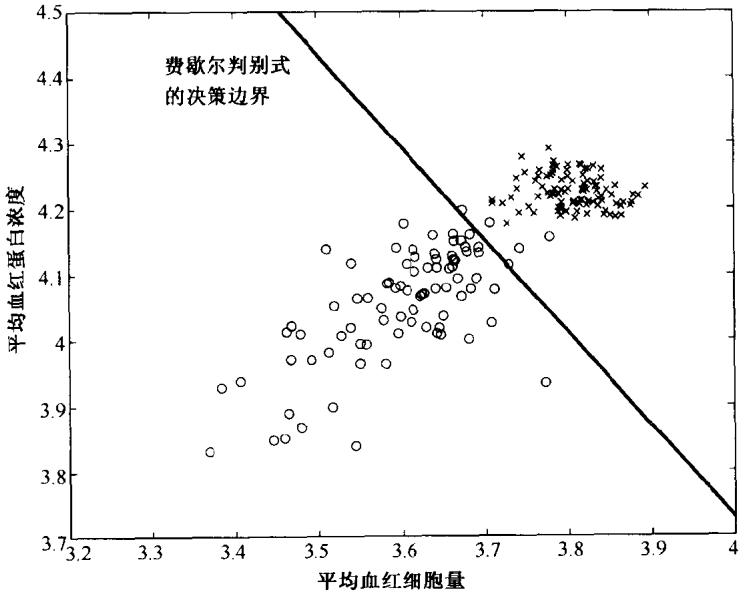


图10-4　费歇尔线性判别式产生的决策边界。这里的数据就是第9章中讨论贫血问题的血红细胞数据，其中叉号表示健康的一类，圆圈表示铁缺乏贫血的一类

342

一种特例是每种类别内部的数据都符合具有共同的协方差矩阵的多元正态分布，这时得到了和公式 10.2 相同的最优分类规则（并且实际上只要两种类别都具有相同二次形式的椭圆分布，那么分类规则就是最优的）。不过应该注意，既然 $\mathbf{w}_{lda}$ 是在没有假定正态的情况下求出的，所以即使在不服从正态分布的情况下线性判别式方法也经常可以提供有价值的分类器。还应该注意，如果我们从潜在分布的假定形式角度来分析线性判别式方法，那么与其把它看作判别法，还不如把它看作基于分类条件分布的方法。

已经开发出了很多基于原始费歇尔线性判别式的扩展形式。正则判别式函数（canonical discriminant functions）产生 m–1 个不同决策边界（假定 m–1<p）来处理类别数 m>2 的情况。当放宽了协方差矩阵相等的条件时，二次判别式函数（quadratic discriminant functions）在输入空间中产生二次的决策边界，就像例 10.2 中所讨论的那样。正规化判别式分析（regularized discriminant analysis）则将二次方法的形式更加简化。

线性判别式模型的计算复杂度是 $O(mp^2n)$。这里我们假定 $n>>\{p, m\}$，所以主要的开销是估计分类的协方差矩阵 $\hat{\mathbf{C}}_i$，$1\leqslant i\leqslant m$。至多对数据库进行两次线性扫描便可以发现所有这些矩阵（一次是取得均值，一次是产生 O (p^2)个协方差矩阵项）。因此，这个模型对观察值数量的变化有很好的伸缩性，但是对于变量数目的增大特别敏感，因为它对变量数 p 的依赖性（需要估计的参数数量）是二次的。

10.5 树模型

树模型的基本原理是以一种递归的方式来划分输入变量所跨越的空间，目的是最大化关于类纯度的评分函数，也就是使（大致如此，依赖于所选取的特定评分方法）划分出的每个单元（cell）的大多数点都属于同一类。例如，对于三个输入变量 x，y 和 z 的情况，可以切割 x，把输入空间分裂成两个单元。然后再把这两个单元中的每一个一分为二，或许是再次按 x 的某个阈值，也可以按 y 或 z 的某个阈值。一直重复这个过程直到没有必要继续下去（参见下文），并用每个分支点定义树的节点。如果要预测一个已知输入变量值的新案例的分类 343
值，那么便沿着树模型向下追溯，在每个节点把新案例和该节点变量的阈值进行比较，然后选取合适的分支。

树模型在很久以前便出现了，但建立树的正式方法出现的时间并不长。在这种方法出现之前，人们基于以前对数据的内在产生过程和现象的理解来构建树。树有很多有吸引力的特征。它易于理解和解释；它可以轻松的处理混合类型的变量（比如连续的和离散的），因为树使用二元测试来划分空间（对于实数值变量使用阈值；对于分类型变量可使用子集成员测试）；它可以非常迅速地预测新案例；它还非常灵活，因此可以用来建立强大的预测工具。然而，它所固有的顺序性（构建树的方式决定了这一点）有时可能导致所得化分对输入变量空间来讲不是最优的。

建立树的基本策略极其简单：就是递归地分裂输入变量空间的各个单元。分裂给定单元（或者说，如何选取用来分裂节点的变量和阈值）的方法是搜索每个变量的每个可能阈值，目的是找到可以最大改善制定评分函数的阈值分裂。分数是以训练数据集合中的数据为基础进行评估的。如果目标是要预测一个对象属于两种类别中的哪一种，那么就选取对局部分数产生最大平均改进（对两个子女结点求平均）的变量和阈值。节点的分裂不会导致评分函数对训练数据的恶化。已经证明，对于分类的情况，直接使用分类误差并不是选取分裂变量的有效评分函数。人们发现像熵这样不太直接的其他尺度效果更好。注意，对于有序变量，二元分裂对应于关于变量值的单一阈值；对于标称型变量，分裂对应于把变量值划分成两个子集。

例 10.3 用于特定实数值阈值测试 T（T 代表了对一个变量的阈值测试 $X_j>T$）的熵标准被定义为执行这个测试后的平均熵：

$$H(C|T)=p(T=0)\,H(C|T=0)+p(T=1)\,H(C\mid T=1) \qquad (10.10)$$

其中，条件熵 $H(C|T=1)$被定义为：

344

$$-\sum_{c_k} p(c_k \mid T=1)\log_2 p(c_k \mid T=1)$$

熵平均就是来自每个分支（T=1 或 T=0）的不确定性对沿每个分支下降的概率的平均。因为我们的目标就是把数据分裂成各个子集从而使尽可能多的数据点属于一个类或者另一个类，这完全等价于使每一分支的熵最小化。在实践中，我们对所有的变量进行搜索，目的是找到一个测试 T，使经过这个二元分裂后的平均熵最小。

原则上，这个分裂过程该一直继续到每个叶子节点仅包含唯一的训练数据点——或者当多个训练数据点具有相同的输入变量向量时（如果输入变量是范畴性的，那么这有可能发生），这个过程该继续到每个叶子节点仅包含具有相同输入变量值的训练数据点。然而，这样做可能导致严重的过度拟合。通常不必分裂到这种极端的情况（也就是说，构建更小单位的，更简约的树）就可以得到更好的树（从对来自同一分布的新数据产生更好的预测这个意义上来说）。

早期的研究通过在达到这种极端情况之前停止分裂来实现这个目的（这类似于我们下一章中将要讨论的，通过终止收敛过程来避免神经网络中的过度拟合）。然而，这种方法因受分裂过程的序列性影响而存在天生不足。有可能出现因为下一步可以取得的改善非常微小便停止了增长的树，只要再向下一步就可以产生非常显著的改善。这个效果“很差”的步骤可能是得到再下一步显著改善的必要基础。当然关于这个问题不存在特别有效的专门办法。这是顺序性方法的通病：这也完全适用于下一章将讨论的分步回归搜索算法——它也是为什么更周密的算法不仅包含向前进入而且包含反向回溯的原因。类似的算法已经渗透到了树方法中。

目前，一种普遍的策略是先建立起一棵庞大的树——持续分裂直到每个叶子结点都满足了某个终止条件（例如一个节点的所有数据点都属于同一类或者都具有相同的 x 向量）——然后再对这棵树进行剪枝。也就是，每一步融合两个叶子节点，选择融合对象的标准是使树对训练集合的预测性能降低最小。可供选择的方法还有，使用像最短描述长度这样的尺度或者交叉验证（例如第 5 章中描述的 CART 算法）来防止过度拟合训练数据。

345 还有两种广泛使用的避免过度拟合训练数据问题的策略。第一种是对叶子作出的预测和通向叶子的节点的预测进行平均；第二种方法是根据对几棵树的平均来做预测，建立每一棵树时都以某种方式对数据进行轻微的打乱（perturbing）。最近第二种方法吸引了更多的注意。事实上，这种*模型平均方法*（model averaging methods）对于所有预测建模都是通用的。模型平均方法对于树模型特别有效，这是因为从以下角度来看树模型具有相当高的变化性：树对训练数据中的微小变化特别敏感，因为数据中的微小波动可能导致选取不同的根结点并产生一个完全不同的树结构。取基于多种扰动数据集的树的平均（也就是根据来自训练数据的多个自展（bootstrap）样本建立多棵树，然后再对它们的预测取平均）大多时候可以通过降低方差来抵消这种影响。

通常把给定叶子节点上的训练数据点的最普遍分类值（大多数分类）作为对到达这个叶子的任意数据点的预测分类。这相当于把通向这个叶子节点的分支所定义的输入空间区域的最可能分类标签赋给了这个区域。有时，在给定叶子节点的训练数据总体分类概率分布中包含了有价值的信息。注意对于任何一个特定的类别，树模型所产生的概率实际上相当于输入空间中的一个固定分段，所以输入变量值的一个很小的变化都可能导致以完全不同的分类概率，从而把数据点发送到一个不同的分支（进入不同的叶子或者说区域）。

剪枝之前为了建立树而寻找最佳的分裂时，算法搜索所有的变量和这些变量的所有可能分裂。对于实数值的变量，分裂的可能位置数通常被取为 $n'-1$（也就是比每个节点的数据点

数 n'小 1），每个分裂的位置取在两个数据点的中间（取中间位置未必是最优的，但具有简单的优点）。如果直接寻找p个实数值变量中的最佳分裂，那么计算的复杂度通常在$O(pn' \log n')$这个规模。$n' \log n'$项是用来对节点的变量值进行排序以便计算评分函数：对于任意的阈值我们需要知道多少个点在这个阈值之上，多少个点在它之下。对于很多评分函数我们可以证明有序变量的最优阈值一定位于属于不同类的两个变量值之间。可以利用这个事实来提高搜索的速度，尤其是当数据点数量很大的时候。此外，可以利用各种记录方法来避免从一个节 346
点到另一个节点时的重新排序。对于范畴型的变量，必须进行某种形式的组合搜索来发现定义分裂的最佳变量值子集。

从数据库的观点来看，树增长是一种开销很大的过程。如果节点上的数据点数超过了主存储器的存储能力，那么函数就必须操作主存储器中的数据缓存和放在副存储器中的其余数据。“蛮力”实现方法为树上的每个节点线性扫描数据库，从而使算法非常慢。因此，当要对超过了主存储器存储能力的数据应用树算法时，要么使用聪明的算法（具有专门的数据管理策略，可以使对副存储器的访问最小化）；要么对数据随机采样，以在一个主存储器可以容纳的样本上工作。

基本树模型的一个不足在于它是*单描述的*（monothetic）：每个节点仅根据一个变量做出分裂。在有些现实的问题中，分类变量随输入变量组合的不同变化很快。例如，在一个包含两个变量的分类问题中，可能的情况是，一种分类对应于两个输入变量的值都很低的数据；而另一种分类对应于两个变量的值都很高的数据。这个问题的决策曲面是输入变量空间的对角线。标准的方法使用多个分裂来实现这个曲面，结果得到一个对对角线决策曲面的楼梯状近似。图 10-5 简单地说明了这种情况。当然，最佳的方法是为输入变量的线性组合定义一个阈值——对树方法的一些扩展就是这样做的，这些方法允许在要被分裂的可能变量集合中包含原始输入变量的线性组合。当然，这加大了建立树所需搜索过程的复杂度。

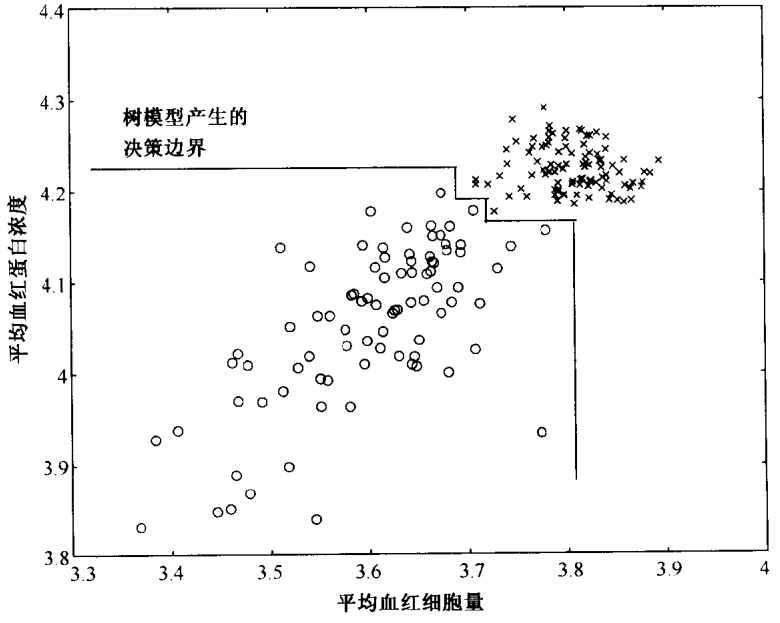


图10-5 决策树模型对血红细胞数据产生的决策边界。这里使用的是第9章中的数据。决策边界是由多个和坐标轴平行的线性分段组成的（相对而言，图10-4中的决策边界更加简单）

10.6　最近邻方法

最近邻方法的基本形式是非常易于理解的：要分类一个输入向量为 **y** 的新对象，就在训练数据集合中找出与 **y** 最靠近的 k 个点，然后把这 k 个点中的大多数点的分类赋给这个新对象。这里的“靠近”标准是定义在 p 维输入空间中的。因此我们就是根据输入变量寻找和新
347 对象最靠近的训练数据，然后把新对象归类到这些最相近对象中最有代表性的类中。

从理论上讲，我们是取变量空间中以 **x** 为中心到第 k 近的近邻的距离为半径的一个小胞体（volume）。然后，用这个小胞体中训练数据点属于每个类的比例作为这个胞体中的点属于每一类的概率的极大似然估计量。k 近邻方法把新的点分配到具有最大估计概率的类中。
348 最近邻方法实质上属于我们所说的“回归”法——它直接估计类隶属关系的后验概率。

当然，前面的简单勾勒遗漏了很多问题。首先是我们必须选取 k 的值和一种定义靠近的尺度。最基本的形式是取 k=1，但这样得到的分类器相当不稳定（变化性太大，对数据过于敏感），因此很多时候可以通过提高 k 值（降低了方差，但由于取平均的范围扩大了，所以可能增大偏差）来使预测更加一致。然而，增大 k 意味着包含进来的训练数据点未必是和要分类的对象非接近的。也就是“小的胞体”可能根本不小。因为是对属于这个胞体内每一类的概率做出平均估计，所以这可能完全偏离这个胞体内任意点的值——而且这种偏离很可能随着胞体的增大而增大。当然这里面维数 p 起了重要的影响：对于数量点数 n 固定的数据集合来说增加 p（加入变量）会使数据变得越来越稀疏。这会导致预测出的概率和这一点在问题中的真实概率相偏离。

我们又回到了偏差（bias）/方差（variance）平衡这一无所不在的问题，因为增大 k 会降低方差但会增大偏差。有关于选取 k 的理论指导，但是因其不仅依赖于一些普遍因素，还依赖于数据集的特定结构，所以选取 k 的最佳策略应该是一种对数据具有适应性（data-adaptive）的策略：试验不同的值，画出性能判据（比如误分类率）对 k 的曲线，然后选择一个最佳的 k 值。在基于这种策略的方法中，必须使用独立与训练数据的数据集来进行这种评估（不然的话会导致常见的过度拟合问题）。然而对于较小的数据集，为了分离出测试集合而减小训练集合是不明智的，因为最佳的 k 值无疑依赖于训练数据集中的数据点数。一种有效的策略（尤其是对于小数据集）是采用“留出一个”（leaving-one-out）交叉验证评分函数。

许多最近邻方法都采用欧氏距离尺度，如果 **y** 是要分类点的输入向量，**x** 是训练集合中的点的输入向量，那么它们间的欧氏距离为$\sum_j (x_j-y_j)^2$。正如第 2 章中所讨论的，这里存在的一个问题是没有对不同输入变量的相对重要性提供一种明确的衡量方法。我们可以通过加权来解决这个问题，即使用$\sum_j w_j(x_j-y_j)^2$，其中 w_j 是权。这看起来比欧氏距离更复杂一些，但是
349 不需要选取权的欧氏尺度是不可靠的。只要在计算欧氏距离前改变一个变量的测量单位就可以看出这一点。（这种情况的一个例外是所有的变量都是以相同的单位测量的——比如，在多种不同的场合测量同一个变量的情况——也就是所谓的重复测量（repeated measures）数据。）

对于二分类的情况，最佳尺度是以属于类别 c_1 的概率（也就是 $P(c_1|\mathbf{x})$）等高线来定义的。与 **y** 在同一等高线上的训练数据点和在 **y** 点的数据点属于 c_1 类的概率相同，因此把它们包含进 k 个最近邻没有引入任何偏差。与此相反，和 **y** 靠近但不在 $P(c_1|\mathbf{x})$等高线上的点属于

c_1 类的概率是不同的，因此把它们包含进 k 个近邻中往往会引入偏差。当然，我们不知道等高线的位置。要是知道了，我们就根本不需要进行这个过程了。这意味着在实践中我们是估计出近似的等高线并把尺度建立在它们之上。不论是全局的方法（例如用多元正态分布来估计各个类别）还是局部方法（例如迭代应用最近邻方法）都已经被用来发现近似的等高线。

最近邻方法与第 6 章中讨论的用于密度估计的核方法的关系非常密切。基本的核方法定义一个确定带宽的单元（cell），然后计算这个单元中的点属于每个类的比例。这意味着这个比例的分母是一个随机变量。基本的最近邻方法把这个比例（k/n）固定，让带宽成为随机变量。在实践中，基于这两种方法扩展出的方法（例如平滑衰减核函数，给最近邻点根据它们与 x 的距离赋予不同的权，或者选取根据 x 变化的带宽）几乎是无法分辨的。

最近邻方法有几种有吸引力的特征。它易于编程并且不需要优化和训练；对于某些问题它的分类精度很高，可以和像神经网络这样的更专业方法相比；它允许方便地应用否决选项（reject option）（如果我们对预测出的分类没有足够的信心那么可以推迟这个决策）；可以直接扩展到多分类的情况（尽管这时如何选择最佳尺度不太明确）；本身就可以处理被分类对象向量中的残缺值：只要工作在那些提供了值的变量子空间中就可以了。

从理论角度讲，最近邻方法是一种很有价值的工具：随着设计样本容量的增大，估计出的概率的偏差会降低（对于固定的 k）。如果我们可以把 k 增大到一个适合的程度（使估计的 350
方差也下降），那么最近邻规则的误分类率将收敛到一个和贝叶斯误差率相关的等值。例如，当数据点数 n 趋向于∞ 时，最近邻方法的误分类率的上限为贝叶斯误差率的二倍。

对于所有的方法，过高的维数都会产生问题。从本质上讲，要克服这一问题就必须不使用那些太灵活以至于过度拟合数据的分类规则，因为变量数太多增大了过度拟合的可能性。在这种情况下，简单形式的参数模型（比如线性模型）经常表现得很好。而最近邻方法大多时候表现得不这么好。当变量数目非常大时（并且对应的训练数据数量并没有那么大时），最近邻的 k 个点经常是实际上距离非常远。这意味着要引入非常粗略的平滑处理，而这种平滑处理和分类的目标是不相关的，就导致对于变量数很多的问题最近邻方法的性能非常差。

此外，理论的分析还提出了最近邻方法在高维情况下可能存在问题。在某种分布条件下，任一特定点 **x** 距其最近点的距离与距其最远点的距离的比例随着维数的增长接近 1，于是最近邻的概念变得没什么意义了。然而，得到这个结果所需的分布假定是相当强的，在其他更现实的假定下，最近邻概念是完全经得起推敲的。

最近邻方法的另一个可能不足是它并不建立模型，而是依赖于把所有训练数据集合中的点都保留下来（由于这个原因，有时把这种方法称为“消极”方法）。如果训练数据集很庞大，那么要搜索到 k 个最近邻点是很费时间的一个过程。特别是当使用蛮力方式搜索时，要访问 n 个训练数据点中的每一个，并进行 p 次操作来计算到每个点的距离，因此每次查询所需的时间复杂度是 $O(np)$。因此对于 n 值很大的应用和/或者要求实时的分类来说（例如使用最近邻算法从数百万条记录的客户数据库中搜索出与网站的当前访问者相似的客户，然后向访问者实时的推荐产品），直接应用最近邻方法在空间和时间方面都是不可行的。

人们已经开发出了很多基本方法的变体来加速搜索并降低内存需求。例如，可以应用分枝定界法：如果已经知道在距离要分类的点距离为 d 的范围内至少有 k 个点，那么如果一个 351
点位于已知与要分类点距离超过 $2d$ 的点的 d 半径范围内，那么就没必要再考虑这个点了。这需要对训练数据集进行预处理，还有一些抛弃某些训练数据点的预处理方法。例如，压缩（condensed）最近邻方法和简化（reduced）最近邻方法选择性地抛弃一些设计集合数据点，

选择的标准是使那些剩下的数据点仍然可以正确分类所有其他训练数据点。改进（edited）最近邻方法抛弃那些位于另一个类的稠密区域中的这个类的孤立点，以这种方式来平滑决策曲面。这些方法在速度和内存方面的改善通常依赖于很多因素：包括 n 和 p 的值；当前数据集的具体特征；使用的具体技术；和时间与内存二者间的折衷等。

还有一种提高最近邻方法对大数据集和高维情况的伸缩性的方法，它使用聚类来对数据分组，然后根据数据点在聚类中的隶属关系把它们存储到磁盘上。当要寻找和输入点 **y** 最近的点时，先找到最靠近 **y** 的聚类，然后在这些聚类的范围内进行搜索。在相当宽松的假定下这种方法都能以很高的概率找到真正的最近邻。

10.7　logistic 判别式分析

对于二分类的情况，从回归角度出发的应用最广的基本分类方法之一就是 logistic 判别式分析（logistic discriminant analysis）。给定一个数据点 **x**，它属于 c_1 类的估计概率是：

$$p(c_1 \mid \mathbf{x}) = \frac{1}{1+\exp(\beta'\mathbf{x})} \tag{10.11}$$

既然属于两种分类的概率的和为 1，那么只要做减法就可以得到属于第二类的概率：

$$p(c_2 \mid \mathbf{x}) = \frac{\exp(\beta'\mathbf{x})}{1+\exp(\beta'\mathbf{x})} \tag{10.12}$$

对上面的关系进行变换，容易看出对数赔率（odds ratio）是 x_j 的线性函数。也就是：

352

$$\log \frac{p(c_2 \mid \mathbf{x})}{p(c_1 \mid \mathbf{x})} = \beta'\mathbf{x} \tag{10.13}$$

这种对后验概率建模的方法具有很多有吸引力的特征。例如，如果分布是具有相等协方差矩阵的多元正态分布，那么它就是最优的解。此外，对于 **x** 为离散变量的情况，如果可以用具有同一个交叉项的对数线性模型（在第 9 章中曾经提到）来对分布建模，那么它也是最优的。还可以把这两个最优的特征组合在一起，那么就得到了一个可用于混合变量（也就是既有离散变量和又有连续变量）的有吸引力的模型。

费歇尔线性判别式分析方法对于具有相等协方差矩阵的边缘正态分类情况也是最优的。如果已经知道数据是从这样的分布采样的，那么费歇尔的方法更高效。这是因为它通过对协方差矩阵建模显式的使用这种信息，而 logistic 方法避开这一点。另一方面，logistic 方法具有更一般的适用性（实践中根本不存在严格的多元正态分布），这使其如今比线性判别式分析方法更受青睐。这里使用如今（nowadays）一词是因为这种算法需要计算两个模型的参数。线性判别式分析模型的数学简洁性意味着它可以找到显式的解。而 logistic 判别式分析并非如此，它必须采用迭代的估计过程。这种算法的最常见形式就是极大似然方法，使用似然作为评分函数。我们将在第 11 章中描述这种方法，那时我们将在推广的线性模型这一更广泛的框架下来进行讨论。

10.8　朴素贝叶斯模型

理论上讲，建立在分类条件分布（所有变量都是范畴性变量）基础上的方法是很直接明了的：我们只要先估计出来自每个类的对象落入离散变量每个单元（变量向量 **X** 的每种可

能离散值）中的概率，然后使用贝叶斯定理来产生分类。然而在实践中，这经常是难以实现的，因为对于 p 个 k 值变量，必须估计的概率数量是 $O(k^p)$。例如，如果在一个应用中变量数 p=30，并且每个变量都是二值的（k=2），那么我们就必须估计出 $2^{30}\approx10^9$ 个概率。假定（根据经验）我们应该为模型中每个要估计的参数至少准备 10 个数据点（这里模型参数就是指用来说明联合分布的概率），那么我们将需要 10^{10} 数量级的数据点来准确地估计所需的联合分布。对于 m（$m>2$）种分类的情况，需要的数据点数是这个数字的 m 倍。显然对于 p 较 353
大的情况，这种方法是不可行的。

我们曾在第 6 章和第 9 章中指出，总是可以通过适当的独立假定来简化联合分布，本质上这相当于用小得多的表的乘积来近似 k^p 个概率的完整表格。对于极端的情况，我们可以假定所有的变量对于给定的分类是条件独立的，也就是说：

$$p(\mathbf{x}\mid c_k)=p(x_1,\cdots,x_p\mid c_k)=\prod_{j=1}^{p}p(x_j\mid c_k),\quad 1\leqslant k\leqslant m \tag{10.14}$$

有时这被称为朴素贝叶斯或一阶贝叶斯假定。这种近似允许我们用一元分布的乘积来近似需要 $O(k^p)$个概率的完整条件分布，近似后每个类所需的概率总数是 $O(kp)$。因此条件独立模型对变量数 p 是线性的而不是指数的。如果使用这个模型进行分类，那么我们只要使用这种乘积形式的分类条件分布，它便是朴素贝叶斯分类器。

上面使用朴素贝叶斯模型大大减少了参数量，但这是有代价的：我们做出了一种非常强的独立性假定。在一些问题中，条件独立的假定可能是非常合理的。例如，如果 x_j 是医疗症状，c_k 是不同的疾病，那么对于一个患有疾病 c_k 的病人，假定具有任何一种症状的概率仅依赖于疾病 c_k，而不依赖于出现的其他任何症状，那么这种假定可能（或许）是合理的。换句话来说，我们是在各种症状对于给定的每种疾病没有相互作用的条件下（注意这不同于假定各种症状边缘（无条件）独立）对症状是如何出现的进行建模。在很多现实情况中，这种条件独立假定是很不现实的。例如，设 x_1 和 x_2 分别是一群人的年收入和存款总额，c_k 代表他们的信誉度，信誉度被分为两种：好和坏。即使在同一类范围内我们也可以看到 x_1 和 x_2 的依赖性，因为收入越多的人可能存款也越多。如果假定两个变量是独立的，那么这相当于把它们当作两种独立的信息，这显然是与问题中的实际情况不符的。

尽管独立假定模型对所涉及的概率可能不是非常切合实际的，但是它仍有可能作出相当精确的分类。这有很多原因，包括：要估计的参数相对较少，这使估计的变化性很小；尽管 354
产生的概率估计是有偏的，但是因为我们感兴趣的并不是它的绝对值，而仅是它的排列次序，所以这可能并不要紧；很多时候已经对变量进行了筛选，在筛选中抛弃了那些高度相关变量对中的多余变量；朴素贝叶斯分类器的决策曲面可能与最优分类器的决策曲面一致。

除了因为朴素贝叶斯分类器的性能经常好的惊人之外，它流行的另一个原因就是这种分类器的形式特别简单。利用贝叶斯定理和条件独立，可以得出一个测量向量为 $\mathbf{x}$ 的点属于第 k 个类的概率估计是：

$$\begin{aligned}p(c_k\mid\mathbf{x})&\propto p(\mathbf{x}\mid c_k)p(c_k)\\&=p(c_k)\prod_{j=1}^{p}p(x_j\mid c_k)\qquad 1\leqslant k\leqslant m\end{aligned} \tag{10.15}$$

下面假定只有两种类别 c_1 和 c_2，然后计算对数赔率。经过一些简单变换便得到：

$$\log\frac{p(c_1\mid\mathbf{x})}{p(c_2\mid\mathbf{x})}=\log\frac{p(c_1)}{p(c_2)}+\sum\log\frac{p(x_j\mid c_1)}{p(x_j\mid c_2)} \tag{10.16}$$

因此，一个实例属于 c_1 类的对数赔率可以通过把先验的贡献和每个变量分别的贡献简单相加来给出。这种相加的形式对于解释特别有价值，因为可以把每一项 $\log\frac{p(x_j\mid c_1)}{p(x_j\mid c_2)}$ 看作更可能属于 c_1 还是 c_2 的正向或负向贡献。

可以很容易的从许多不同的角度来推广朴素贝叶斯模型。如果测量值是实数的，那么我们仍然可以作出条件独立的假定，然后计算一元密度估计（而不是分布）的乘积。对于任何实数值 x_j，我们可以使用我们喜欢的密度估计技术来估计 $f(x_j|c_k)$，例如像正态密度这样的参数模型，或者像核密度函数这样的非参数模型。对于实数变量和离散变量组合的情况，只要在公式 10.15 中使用密度和分布的乘积就可以了。

尽管上面的公式形式很简单，但是它所对应的决策曲面可能非常复杂，而且不一定局限
355 于线性范围（例如，多元正态的朴素贝叶斯模型通常产生二次的决策边界），这与对原始变量的简单加权求和（比如感知器和费歇尔线性判别式）所产生的线性曲面形成了对比。朴素贝叶斯模型的这种简洁性、俭省性（parsimony）和可解释性使它的应用非常广泛，特别是在机器学习中。

我们可以通过包含超出一阶范围的一些（但非全部）依赖性来推广朴素贝叶斯模型。可以把这种推广想像为对更高阶的依赖性进行搜索，然后选择出一些“重要的”依赖性加入模型（比如 $p(x_j,x_k|c_k)$，以及三元组等等）。通过这样，我们实际上是在建立一种通用的图模型（或者说是信念网络——参见第 6 章）。然而，实践表明对于很多数据集，这种改进的模型对分类性能的改善经常是很有限的，这再次证明了建立精确的密度估计量和建立好的分类器是不同的。

最后我们讨论一下朴素贝叶斯模型的计算复杂度。因为我们（本质上）仅使用了建立在一元密度的简单函数基础之上的加法模型，所以这种模型的计算复杂度大体是估计每个单独一元分类依赖密度和分布的复杂度的 pm 倍。对于离散值的变量，充分统计量就是在每个柱位（bin）中的点数，所以只要扫描数据一次就可以建立起贝叶斯分类器了。对于实数值变量的一元密度参数模型来说扫描一次也是足够的（我们仅需要搜集充分统计量，比如正态分布的均值和方差）。对于更加复杂的密度模型，比如混合模型，由于匹配这些密度函数的迭代性（如第 9 章中所讨论的），我们则可能需要多次扫描数据才能建立模型。

10.9 其他方法

近年来开发出了大量的预测分类方法。在现代计算设备所提供的惊人性能的推动下，其中很多方法已经非常强大和灵活。前面我们已经介绍了这些方法中的一部分，说明了它们是如何相互联系的。其他的方法还有很多，但是在一本书的一章中讨论所有这些方法是不现实
356 的，而且，还不断有新的方法被发明和开发出来。令人激动的研究在我们写作本书的同时仍在继续，没能在本章中提到的方法还有：

- 混合模型和径向基函数（radial basis function）方法使用简单分布（例如多元正态分

布）的混合来近似每个分类条件分布。即使仅使用几个分量分布来模拟分类条件分布也可以产生令人吃惊的效果。

- 前馈神经网络（在第 5 章的反向传播方法中曾经讨论过，在下一章的回归方法中也会提到）是对感知器的推广。有时被称为*多层感知器*（multilayer perceptrons）。第一层产生 h_1 个线性项，每一项是 p 个输入的加权组合（在效果上相当于 h_1 个感知器）。然后对这 h_1 个项进行非线性变换（logistic 函数是一种流行的供选变换方案），然后在多个层中重复这个过程。变换的非线性使决策曲面的形状具有高度的灵活性，因此这种模型对于某些分类问题特别有效。然而，这种根本上的非线性意味着估计过程不是显而易见的，必须使用迭代技术（比如爬山方法）。估计过程的计算复杂度导致这种方法尤其不适用于庞大的数据集。
- 投影追踪（projection pursuit）方法可以被看作神经网络方法的“姊妹篇”（在第 11 章中讨论回归时我们将再回到这个话题）。可以数学证明它们具有同等的功能，但是它的优势是估计更加简洁。它也是先对原始变量进行线性组合，然后作非线性变换，进而再对变换的结果进行线性组合。不过，在神经网络中变换是固定的，而在投影追踪中变换是数据驱动的。
- 就像神经网络是从对感知器的早期研究中衍生的一样，支持向量机也是如此。早期的感知器研究假定各个类别是完全可分的，然后寻找一个适当的分裂超平面。最佳的泛化性能是当超平面距离所有数据点都尽可能远时得到的。支持向量机通过延伸测量空间把这种思想推广到更复杂的曲面，以便包括原始变量的变换（组合）。在这种增强的空间中完全分裂数据的线性决策曲面等价于在本来的原始测量空间中完全分裂数据的非线性决策曲面。这种方法的独有特征是使用了一种被称为“边际（margin）”的特殊评分函数，这个评分函数试图使两个类间的线性决策边界位置最优，最优的标准就是最佳的可能泛化性能。使用这种方法的实践经验是可以很快提高的，但是估计过程可能很慢，因为它要求解存储复杂度为 $O(n^2)$、时间复杂度为 $O(n^3)$ 的复杂优化问题。 357

很多分类应用经常是匹配一个非常灵活的模型，然后再以某种方式对其进行平滑处理以防止过度拟合（或者同时进行这两个过程），以求做到对偏差和方差的适当折衷。这表现在对树的剪枝；拟合神经网络的权衰减技术；判别式分析中的正规化处理；支持向量机中的“平滑”等等。一种很不一样的策略是先估计出几个（或者很多）模型，然后对它们的预测进行平均，这和对多个树分类器取平均一样，已经证实这种策略对于预测建模特别有效。显然这种方法和第 4 章中的贝叶斯模型平均方法在概念上是相似的，后者明确地把模型的参数（和模型本身）当作不确定性，然后在做预测时对这种不确定性进行平均。模型平均起源于统计，而从多个分类器的预测中选取多数结果的类似方法起源于机器学习。不过还可以用其他方式来组合分类器，例如我们可以把分类器的输出当作更高层分类器的输入。原则上讲，可以在每一阶段使用任何类型的预测分类模型。当然参数估计通常都不是简单的事。

模型平均策略面临的一个明显问题是：如何加权对平均的不同贡献——也就是应该给每个分类器多大的权？最简单的策略是使用相等的权，但是很显然允许使用不同的权可能更有优势（至少来说相等的权是不等的权这种更一般模型的特例）。人们已经提出了寻找权的各种策略，包括让权依赖于每个模型个体的预测精度和依赖于模型的相对复杂度。boosting 方

法也可以被看作是一种模型平均方法。它建立一系列连续的模型，并在数据集合上训练每个模型，在这一过程中，被前一个模型分错类的数据点被给予更大的权。这与在早期感知器算
358 法中使用的误差纠正策略具有明显的相似性。最近的研究已经提供了实验和理论证据表明boosting方法是建立平滑预测模型的一种非常有效的数据驱动策略。

10.10　分类器的评估和比较

本章讨论了预测分类模型——使用对象的一系列测量来对一个新对象可能隶属的类别做出预测的模型。有很多不同的模型可以实现这一目的，所以一个很自然的问题就是“对于一个给定的问题到底应该使用哪种方法？”。不幸的是对于这个问题根本没有通用的答案，方法的选择必须依赖于问题、数据和目标的特征。当然了解这些方法的特征会有助于对它们做出选择，但是理论上的属性不总能有效的指导实践（贝叶斯模型中的独立假定的有效性说明了这一点）。当然，预期的和观察到的性能的差异起到了刺激进一步理论研究的作用，使理论不断深入。

如果目前的理论理解无法对实践结果做出解释，那么很多时候必须通过对性能的实验性比较来指导我们选择不同的方法。评估分类规则的研究成果非常多。其中大多都是以其他领域的建模为背景提供了一种初步的实验方法。这一节简要地介绍如何评估分类模型的性能。

到目前为止我们使用的评估标准都是分类模型的误差率或者叫误分类率——也就是这个规则可能错误分类将来对象的比例。我们把贝叶斯误差率定义为最优的误差率——假设我们的模型是建立在数据的潜在真实分布函数基础上时可以达到的误差率。当然在实践中必须事先选择这样的函数形式（或者使用替代的判别法或回归法，然后估计它们的参数），所以模型很可能偏离这个最优的情况。这样模型便具有一个真实的或者说实际的误差率（它不会小于贝叶斯误差率）。真正的误差率有时被称为条件误差率，因为它是以给定的训练数据集为条件的。

我们需要有一种方式来估计这种真实误差率。一种显而易见的方法是重新分类训练数据，然后计算被错误分类的比例。这就是表观（apparent）误差率，或者叫重新代入
359 （resubstitution）误差率。不幸的是，这很可能低估了将来的错误分类率。这是因为预测模型就是在这个训练数据集上建立的，所以对于这些数据它表现的可能更好。（退一步来说，故意的选取在训练数据上也表现很差的模型是不正常的！）既然数据不过是从问题中的分布抽取出的样本，所以它不可能完全反映这个分布。这意味着我们的模型可能仅反映了针对训练数据的这部分特征。因此如果重新分类训练数据，那么正确分类率会比分类将来数据的情况好。

我们已经在很多话题中讨论了这种现象，人们也已经提出了很多克服这一问题的方法。一种显而易见的做法是在一个新的样本中计算错误分类率，以估计将来的误差率，这个新的样本被称为检验集合（test set）。这种方法非常好——但它忽视了一个事实，要是有检验集合可以使用的话，那么我们用它来组成一个更大的训练集合可能收获更大。因为这样可以建立一个更加精确的模型。在构建模型时故意地忽略一部分数据似乎是不经济的，当然除非 n 非常大，并且我们有把握认为（举例来说）在一百万个数据点上（保留另外一百万用作测试集合）训练和在完整的二百个数据点上训练效果大体是一样的。

当数据集合的容量为中等大小时，人们提出了很多不同的交叉验证方法（参见第 7 章和其他有关部分），也就是在组建规则时留出一小部分数据（比如说十分之一），然后在留出的这部分数据上对规则进行测试。并且可以留出数据的不同部分来重复这个过程。基于这一原理的重要方法有：

- 留一（leaving-one-out）法，在每一阶段仅留出一个数据点，但是每个数据点是依次被留出的，所以最终测试集合的大小等于整个训练集合的大小，但是在这个过程中每个唯一一点的测试集合是独立于它所测试的模型的。其他的方法是用较大比例的数据作为测试集合（例如整个数据集的十分之一），但是由于对将来模型的性能估计是基于整个数据集的，所以这些方法比“留一”法具有更大的偏差。
- 自展（bootstrap）法，这种方法有很多变体。该方法用样本和从样本中轮番抽出的同样容量的子样本间的关系来对未知的真实分布和样本的关系建模。在一种方法中，使用这种关系来纠正重新代入所引起的偏差。已经开发出了一些非常周密的自 360
展方法的变体，它们是目前为止的最有效方法。Jackknife 方法也是以每次留出训练集合中的一部分数据（就像交叉验证中那样）为基础的，但是它等价于自展方法的一种近似。

还有很多其他的误差率估计方法。这个领域一直是很多论文所探讨的一个课题——参见补充读物，那里对此做了详细介绍。

误差率同等对待错误分类每一对象的严重性，然而，这经常是不切实际的。很多时候，某种误分类比其他更加严重。例如，当某个人患了某种小毛病时，把它错误诊断为一种可以医治的疾病显然没有把他诊断为不治之症严重。在这种情况下，我们希望可以为不同的分类附加代价（costs）。这样建模的标准就不再是简单的误差率，而是使总体代价最小。

这些思想很快被推广到多分类的情况。很多时候画混淆（confusion）矩阵（行和列分别预测分类和真实分类）是很有价值的。可以把这个矩阵的每个单元和做出相应误分类（或者正确分类——对于矩阵对角线的情况）的代价联系起来，这样便可以估计出总的代价。

不幸的是，很多时候代价是难以确定的。这时，另一种可以使用的策略是对一种代价相对另一种（对于二分类的情况，也可以推广到两种以上分类的情况）代价的所有可能比值进行积分。这种便产生了所谓的性能吉尼系数（Gini coefficient）。这个尺度与用来比较两个独立样本的 Mann-Whitney-Wilcoxon 统计验证中的验证统计量是等价的，也等价于接受者操作特性（Receiver Operating Characteristic）曲线或者 ROC 曲线（类别 1 中的对象被正确分类到类别 1 的估计比例相对类别 2 的对象被错误分类到类别 1 的估计比例的曲线）下的面积。ROC 曲线和该曲线下的面积广泛应用于很多研究领域。不过，它们也存在解释上的问题。

分类模型的性能仅仅是选择方法时要分析的一个方面，另外一个要素是方法适合数据的程度。例如，一些方法更加适合于离散型变量，一些方法更适于连续型变量，而还有一些方法以同等的性能工作在这两种类型的数据上。当然数据残缺对于任何方法来说都是一个潜在的（事实上是普遍存在的）问题，某些方法可能比其他方法能更容易地处理不完整数据。例 361
如，独立假定下的贝叶斯方法处理这样的数据非常简单，而费歇尔判别式分析方法就不是这样。数据残缺的原因很多，并且这些原因可能影响建立在不完整数据上的模型的有效性，这时事情就变得更加复杂。补充读物中列出了讨论这一问题的参考资料。

总而言之，分类模型的评估是一个重要的领域，而且已经成为一个备受关注的课题。

10.11　高维分类的特征选取

数据挖掘者在实践中经常面对的一个问题是变量数太多。简而言之，并不是测量出的所有变量都是实际判别所必需的；而且要是把它们包含到分类模型中会使模型（比删除它们）更差。考虑一个简单的例子，假定要建立一个系统来判别男性和女性面容图像（这个任务对于人类来讲不费吹灰之力，但是对于图像分类算法来说却是非常富有挑战性的）。在这个问题中，人的眼睛、头发或者皮肤的颜色对于判别几乎没有什么价值。这些变量易于测量（而且确实是人外貌的一般特征），但是携带的识别分类的信息却很少。

在大多数数据挖掘问题中，哪些变量是（或者不是）有关的并不像上面的例子那样明显。例如，把一个人的人口统计特征和在线购买行为联系起来就不是十分明显，而且这也未必符合传统的模式（试想一个受过博士教育的高收入群体会花很多钱来买连环漫画书籍吗？——如果存在这样的群体，那么连环漫画零售商会很想知道！）在数据挖掘中，我们特别希望让数据来说话，也就是使用适应数据的方法来选取变量（由于通常没有预先的知识告诉我们哪个变量对任务是明显无关的，所以无论如何我们还该使用这个信息）。

在第6章中讨论一般的建模问题是我们已经提到了这一问题，在那里我们列出了一些通用的策略，这里再简要的回顾一下：

- **变量筛选**：该方法的思想是从原始的 p 个变量中筛选出一个包含 p'个变量的子集。
362 当然我们事先并不知道 p'的值应该为多少以及到底应该包含哪些变量，所以可以考虑的变量子集的搜索空间是组合性的，非常庞大。因此大多数方法都依赖于对变量子集空间的启发式搜索，很多时候是使用贪婪的搜索方法来每次加入或删除一个变量。这里存在两种一般性的方法，第一种是使用自动进行变量筛选的分类算法，这些算法把变量筛选作为基本模型定义的一部分，分类树模型就是这种模型的最著名代表。第二种方法是把分类器看作一个“黑盒”，在外层设计一个循环（或者叫“包裹”（wrapper））有条理地向变量子集中增加或从中减少变量，并基于分类模型的性能评估每个子集的效果。
- **变量转化**：该方法的思想是通过一个预处理步骤对原始测量进行某种线性的或非线性的函数变换，这样通常便可以得到一个小得多的导出变量子集，然后基于这个转化后的变量集合建立分类器。这种方法的例子包括主分量分析（在这种方法中我们尽可能找到在输入空间中变化最大的方向，这本质上是一种数据压缩技术——参见第3章和第6章），投影追踪（在这种方法中，算法对有趣的线性投影进行搜索——参见第6章和第11章），以及像因素分析和独立分量分析这样的有关技术。虽然这些技术本身可能是非常强大的，但是它们未必和提高分类性能的总体目标很好吻合，这是一个不足。这方面的一个例子是主分量分析，图10-6显示的例子说明了这一点。图中第一主分量的方向（数据要投影的方向，并且这个方向可能被用作分类器的输入）与问题中的最佳线性判别投影方向是完全垂直的——也就是说，对于这个分类任务这个方向是完全错误的。这并不是主分量方法本身有问题，而是因为在分类任务中应用了不合适的技术。当然这个例子多少有些人为性和不合理性；在实践中，主分量投影对于分类经常是非常有价值的。尽管如此，时刻牢记问题的目标还是很重要的。

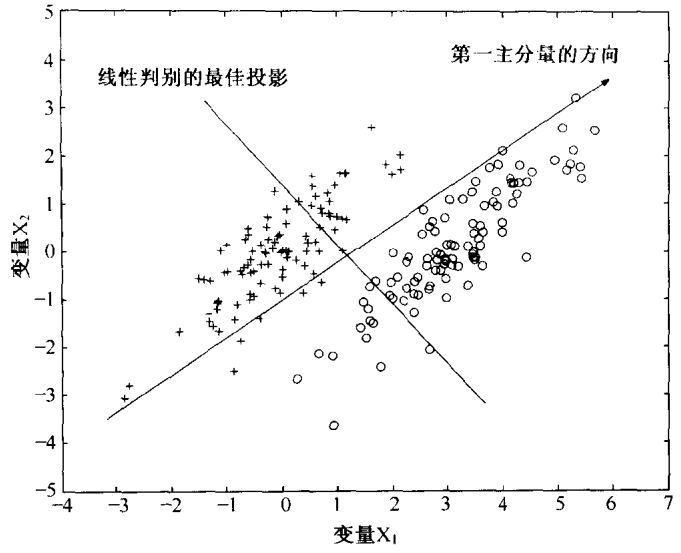


图10-6 利用主分量分析进行分类预处理的可能缺陷。这是一个人为编制的二维分类问题，图中用不同的符号标出了来自不同类的数据。第一主分量方向（对于高维的问题，这就是投影数据的首选方向）实际上和使用费歇尔线性判别式技术确定的最佳投影方向几乎是完全垂直的

10.12 补充读物

费歇尔关于线性判别式分析的最初论文可以追溯到 1936 年。Duda, Hart and Stork（2001） 363
（Duda and Hart（1973）所著的经典模式识别教材的第二版）中包含了各种分类方法的丰富细节，尤其是特别详细的讨论了正态多元分类器（第 3 章）和线性判别式及感知器学习算法（第 5 章）。以下文献从统计角度讨论了分类：Hand（1981，1997），Devijver and Kittler（1982），Fukunaga（1990），McLachlan（1992），Ripley（1996），Devroye，Gyorfi and Lugosi（1996）以及 Webb（1999）。Bishop（1995）从神经网络的角度，Mitchell（1997）从人工智能的角度，Witten and Frank（2000）从数据挖掘的角度探讨了分类。

Dasarathy（1991）的文章中包含很多关于最近邻分类的经典论文，这些论文有的来自统计模式识别的文献。对最近邻方法的一般性讨论（包括降低保留集合大小的方法）可以在 Hand（1981）和 McLachlan（1992）的论文中找到。Short and Fukunaga（1981）、Fukunaga and Flick（1984）及 Myles and Hand（1990）都讨论了为最近邻方法选取距离尺度的策略。Hastie and Tibshirani（1996）描述了一种估计标距的局部适应性技术。Devroye and Wagner（1982）讨论了最近邻规则的渐进性。Hand（1982）讨论了有关的核函数方法。以下文献考虑了高维中“最近”的含义：Beyer et al.（1999）以及 Bennett, Fayad and Geiger（1999）的论文，他们也讨论了使用聚类来近似搜索。

对建立在树基础之上的模型最早讨论的有 Morgan and Sonquist（1963）。Quinlan（1986，1993）把决策树应用于分类，使这种方法在机器学习中流行起来。在统计中，Breiman et al.（1984）所著的书讨论了 CART（分类和回归树）算法，它对树模型的广泛应用产生深远的影响。Ripley（1996）的第 7 章广泛地浏览了统计学、计算机科学和工程实践对树学习方法的不同贡献。最新的一篇调查文章是 Murthy（1998）所写。以下著作讨论了构建决策树的

可伸缩算法：Shafer, Agrawal and Mehta（1996），Gehrke, Ramakrishnan and Ganti（1998），以及 Rastogi and Shim（1998）。Shafer, Agrawal and Mehta（1996）提出的 Sprint 方法仅需要非常小的内存空间来运行，但是它仅适用于 CART 分裂判据。Gehrke, Ramakrishnan and Ganti（1998）的雨林（RainForest）框架可用于多种分裂判据，但是它的内存使用量依赖于变量定义域的大小。Rastogi and Shim（1998）的方法交替进行树的构建和剪枝，因此避免了不必要的数据访问。关于伸缩性问题的一个非常好的调查是 Ganti, Gehrke and Ramakrishnan（1999）做的。

以下文献讨论了独立假定下的贝叶斯方法：Russek，Kronmal and Fisher（1983），Hilden（1984），Kohavi（1996），Domingos and Pazzani（1997）以及 Hand and Yu（1999）。

Vapnik（1995），Burges（1998）和 Vapnik（1998）讨论了支持向量机。Scholkopf, Burges and Smola（1999）搜集了有关这一主题的最新文献，Platt（1999）描述了加速这种分类器的训练过程的一种有用技术。

以下文献讨论了像模型平均这样的分类器组合技术：Xu, Krzyzak and Suen（1992），Wolpert（1992），Buntine（1992），Ho, Hull and Srihari（1994），Schaffer（1994）以及 Oliver and Hand（1996）。Freund and Schapire（1996）讨论了 boosting 技术，关于这方面的更新进行理论探讨的还有 Schapire 等人（1998）以及 Friedman, Hastie and Tibshirani（2000）。

Hand（1997）详细的评论了评估分类算法的方法。特别针对误差率评估方法进行评论的有 Toussaint（1974），Hand（1986），McLachlan（1987）以及 Schiavo and Hand（1999）。Devijver and Kittler（1982）详细讨论了否决选项（reject option）。MacMillan and Creelman（1991）对 ROC 和相关方法进行了综述。

364
~
366

Little and Rubin（1987）对数据残缺、残缺的不同类型以及如何处理做了有创见性的讨论。

第 11 章　用于回归的预测建模

11.1　简介

在第 6 章中我们讨论了预测模型和描述模型的区别。在第 10 章中我们详细地描述了被预测变量（*响应变量*（response variable））是标称型变量的预测模型——也就是它仅可以从有限（通常很少）数量的值中取一个值，并且这些值根本没有数值意义，它们就是*类标识符*（class identifier）。这一章我们转向响应变量具有真正数值意义的预测模型。比如某个零售商店十年内会从一个给定客户那里挣多少钱；正常条件下某种类型汽车的耗油率是多少；某个网站在给定的某个月中用户访问量有多大等等。在预测中用作输入的变量被称为*预报变量*（predictor variable），被预测的变量被称为*响应变量*（response variable）。有些作者有时把响应变量称为*依赖*（dependent）变量或者*目标*（target）变量，而把预报变量称为*独立*（independent）变量、*解释*（explanatory）变量或者*回归*（regressor）变量。第 10 章中还提到了用在分类中的其他一些术语。注意，预报变量可以是数字型的，但不是必须这样。我们的目标就是使用对象的样本来构建一个模型，通过这个模型预测出一个新案例的响应变量值，对于样本来说，响应变量和预报变量都是已知的；而对新案例来说仅有预报变量是已知的。这实质上和第 10 章中的问题是相同的，只不过是响应变量的类型由标称型变成了数值型。事实上，在本章的后面我们将看到我们也可以在回归的通用框架内预测标称型变量（也就是分类）。

预测的精度是模型的最重要特征之一，因此人们设计了很多用来衡量精度的不同尺度。 367
我们也可以使用这些尺度来选取各种候选模型，以及选取模型中的参数值。按照前面的说法，这些尺度就是用来比较不同模型的*评分函数*。

预测精度是模型的一个关键指标，但它不是唯一的指标。例如，我们可能希望模型能够显示出哪一个预报变量最为重要这样的内部细节。我们还可能坚持应该在模型中包含某些变量，因为我们有充分的理由包含它们，即使这仅对预测有很小的改进。与此相反，我们有时要删除一些可以增强模型性能的变量。（这种情况的一个例子发生在信用评估问题中，在很多国家中把性别和种族包含进预报变量是不合法的。）我们可能对预报变量是否存在相互作用感兴趣，也就是一个变量对响应变量的影响是否依赖于其他变量的取值。出于很明显的原因，我们可能对用简单的模型是否可以实现好的预测感兴趣。有时我们甚至愿意牺牲一些预测精度来换取模型复杂度的根本性降低。因此尽管预测精度可能是预测模型性能的最重要部分，但是我们必须根据模型所应用的环境来综合考虑这一问题。

11.2　线性模型和最小二乘法拟合

第 6 章中介绍了线性模型的概念，之所以这样称呼是因为它们相对参数是线性的。这种模型的最简单形式得到的响应变量 y 的预测值 $\hat{y}$ 也是预报变量 x_j 的线性组合：

$$\hat{y} = a_0 + \sum_{i=1}^{p} a_j x_j \tag{11.1}$$

当然实际上我们通常不能完美的预测出响应变量，因此普遍的目标是预测出 y 在预报变量的每个向量位置处所取的均值——所以 $\hat{y}$ 是我们对 y 在 $\mathbf{x} = (x_1, \cdots, x_p)$ 点的均值的预测性估计。这种形式的模型被称为*线性回归模型*（linear regression model）。在最简单的情况中仅有一个预报变量（*单一回归*），这时在响应变量和预报变量所跨越的空间中可以得到一条回
368 归直线（regression line）。更一般的情况是*多重回归*（multiple regression），这时是一个*回归平面*（regression plane），这种模型是最古老、最重要而且应用最广泛的预测模型形式。之所以如此的一个原因是这种模型具有明显的简洁性：简单的加权求和既易于计算又易于理解。另一个非常有说服力的原因是它们在很多情况下都可以达到非常好的性能——即使是对于我们有足够把握认为预报变量和响应变量不是线性关系的情况。不过这并不是空穴来风，而是有道理的：如果我们把连续的数学函数用泰勒级数展开，那么我们经常会发现次数最低的项——线性的项——是最重要的，因此可以使用线性模型得到最好的简单近似。

选取的模型恰好完全正确的情况是非常少的。对于数据挖掘来说更是如此，因为在数据挖掘中模型通常都是试验性的，而不是建立在理论基础之上的（参见第 9 章）。此外，我们的模型可能没有包含理想预测所必需的所有预报变量（很多变量可能还没有被测量出来或者甚至是不可测量的）；或者可能没有包含预报变量的某个函数（或许不仅需要 x_1 还需要 x_1^2，或者需要预报变量相乘，因为它们对 y 的影响是相互作用的）；而且任何情况下测量值都不是十全十美的。所以对变量 y 的预测会存在关联误差，从而使每个向量（$x_1, \cdots, x_p$）是和可能 y 值的分布相联系的，就像我们上面所指出的。

所有这些问题意味着样本中实际 y 值会和预测出的值不同。观察到的和预测出的值之间的差异被称为*残差*（residual），我们把它表示为 e：

$$y(i) = \hat{y}(i) + e(i) = a_0 + \sum_{j=1}^{p} a_j x_{j(i)} + e(i)，\ 1 \leqslant i \leqslant n \tag{11.2}$$

按照矩阵表示，如果我们用向量 **y** 表示训练样本中的 n 个对象的观察到 y 测量值，用 $n\times(p+1)$ 的矩阵 **X** 表示测量 n 个对象得到的 p 种预报变量值（加入额外的一列 1 是为了和模型中的截距项 a_0 对应），那么我们可以根据前面的模型把观察到响应值和预报测量值间的关系表示成：

$$\mathbf{y} = \mathbf{X}\mathbf{a} + \mathbf{e} \tag{11.3}$$

其中 **y** 是 $n\times1$ 的响应值矩阵，$\mathbf{a} = (a_0, \cdots, a_p)$表示$(p+1)\times1$ 的参数值向量，$\mathbf{e} = (e(1),\cdots,e(n))$
369 是包含残差的 $n\times1$ 向量。显然我们需要选取模型中的参数（向量 **a** 中的$(p+1)$个值）使得到的预测尽可能地准确。换个角度来说，我们必须以某种方式找到对 a_j 的估计使分歧 **e** 最小化。为了实现这个目的，我们合并 **e** 中的元素以得到一种可以最小化的单一数学尺度。人们已经提出了很多种合并 $e(i)$的方法，但是到目前为止最流行的方法是对它们的平方求和——也就是误差平方求和评分函数。这样，我们只要求出使下式最小化的参数向量 **a**：

$$\sum_{i=1}^{n} e(i)^2 = \sum_{i=1}^{n} \left(y(i) - \sum_{j=0}^{p} a_j x_j(i) \right)^2 \qquad (11.4)$$

在这个表达式中，$y(i)$是在第 i 个训练样本点观察到的 y 值，并且$(x_0(i), x_1(i), \cdots, x_p(i)) = (1, x_1(i), \cdots, x_p(i))$ 是这个点的预报变量向量。出于很明显的理由，这种方法被称为最小二乘法（least squares method）。简单起见，我们把使上式最小化的参数向量表示为$(a_0, \cdots, a_p)$。（当然如果我们使用某种符号指出它是一种估计，比如$(\hat{a}_0, \cdots, \hat{a}_p)$，那么会更准确，但是我们的表示更加简洁。）如果使用矩阵形式，那么可以证明使公式 11.4 最小化的参数值为：

$$\mathbf{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \qquad (11.5)$$

在线性回归中通常把参数 **a** 称为*回归系数*。一旦估计出了这些参数，就可以把它们代到公式 11.1 中进行预测了。对于预报变量的向量 $\mathbf{x}_k$，可以用 $\hat{y}_k = \mathbf{x}_k^T\mathbf{a} = \mathbf{a}^T\mathbf{x}_k$ 预测出它的 y 值 $\hat{y}_k$。

11.2.1 拟合模型的计算问题

直接求解公式 11.5 需要矩阵 $\mathbf{X}^T\mathbf{X}$ 是可逆的。如果样本容量很小（这在数据挖掘情况下很少见）或者预报变量的测量值间存在依赖性（这并不少见），那么就会产生问题。对于后一种情况，各种现代软件包通常会发出警告，这是可以采取一些合适措施，比如去掉一些预报变量。

有时多个预报变量的测量值不是严格线性依赖的，但依赖程度又很高，这会导致更加棘手的问题。这时矩阵是可逆的，但解是不稳定的。这意味着观察到 **X** 值的轻微变化会导致 **a** 估计值的重大变化。不同的测量误差或训练样本的轻微变化导致不同的参数估计，这个问题被称为*多重共线性*（multicollinearity）。如果估计出的参数值是我们所感兴趣的焦点——例如我们要知道哪个变量在模型中最重要，那么估计出的参数不稳定就是一个问题。然而，如果我们所关心的就只是预测的精度，那么这通常不是问题：尽管数据的轻微变化会产生根本不同的 **a** 向量，但是所有这些向量都会对大多数 $\mathbf{x}_k$ 向量产生相似的预测。 370

通常是用线性代数中的等式求解技术（比如 LU 分解或奇异值分解（SVD））来解公式 11.5，这往往比直接求 $\mathbf{X}^T\mathbf{X}$ 的逆矩阵具有更好的数值稳定性。不论使用哪种特定的技术，潜在的计算复杂度通常都是一样的，也就是 $O(p^2n + p^3)$。p^2n 项对应于计算 $p \times p$ 矩阵 $\mathbf{C} = \mathbf{X}^T\mathbf{X}$ 中的每个元素都需要做 n 次乘法。p^3 项对应于从 $\mathbf{Ca} = \mathbf{X}^T\mathbf{y}$ 中解出 **a**。

在第 6 章中我们曾经指出，当在回归模型中加入更平滑的模型形式时（不仅包含原始变量 x_j，而且包含原始变量的变换），它仍会保持可加性。图 11-1 所示的散点图显示了当逐步增加接受实验者所执行的体力运动的难度时而采集到的数据。水平轴显示的是吸氧量，垂直轴显示的是一种衡量从肺中呼出气体的尺度。从散点图中容易看出这两个变量间的关系是非线性的。从图中可以看出直线 $y = a_0 + a_1x$ 对数据拟合得很差。根据这个模型所作出的预测仅对于 x（吸氧量）大于 1000 并小于 4000 的情况是比较精确的。（尽管如此，这个模型也不是非常的粗劣——正如前面所指出的，关于 x 的线性模型可以给出比较好的近似，这一点显然是正确的。）然而，模型 $y = a_0 + a_1x + a_2x^2$ 所对应的拟合曲线如图 11-2 所示。这个模型的参量仍然是线性的，所以使用公式 11.5 中的标准矩阵可以很容易地估计出这些参数。显然从这个模型得到的预测已经接近完美了，余下的不精确性是不可避免的

371 测量误差所带来的。

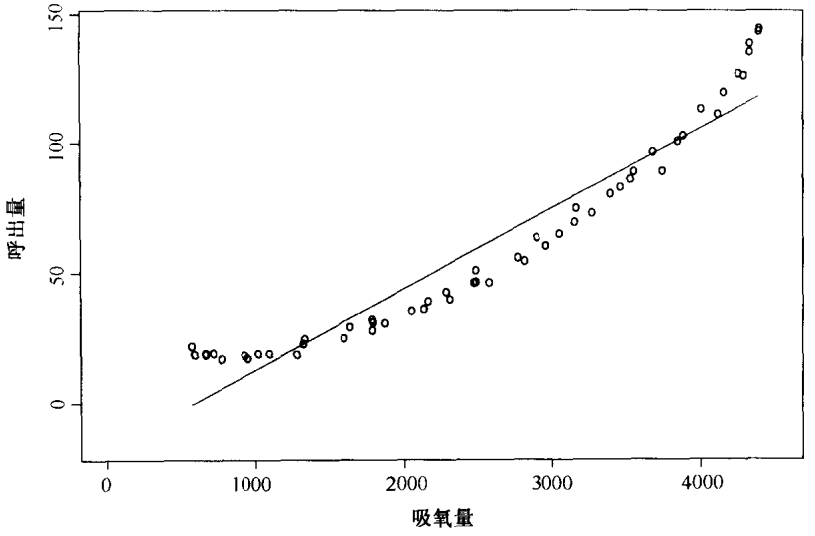


图11-1　呼出量对吸氧量的散点图。图中还画出了一条用来拟合这些点的直线

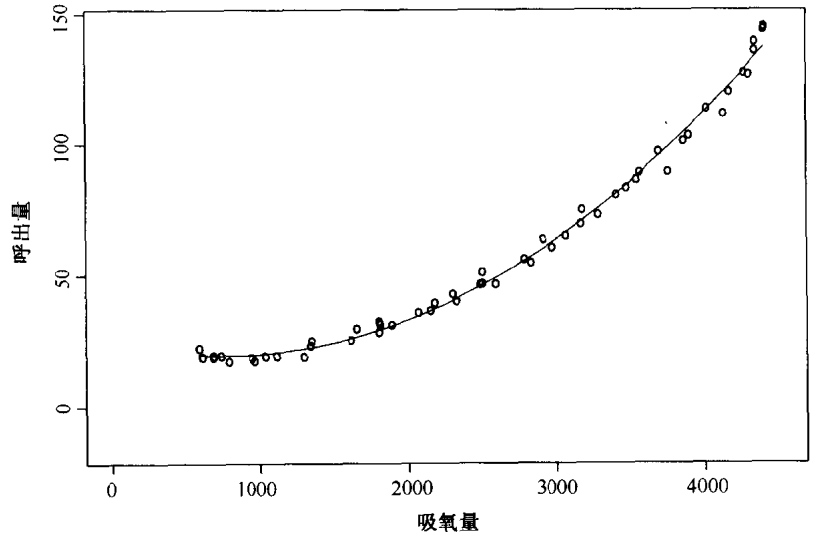


图11-2　用包含x^2项的模型拟合图11-1中的数据

11.2.2　线性回归的概率解释

我们可以使用这种非正式的数据分析方法把回归模型拟合到任何包含一个响应变量和一系列预报变量的数据集上，并得到一个估计出的回归系数向量。如果我们的目标仅仅是产生一种对训练数据的方便的概括（非常少的时候是这样），那么我们便可以就此停止了。然而这一章所关心的问题是预测建模，我们的目标超出了训练数据的范围，是要为样本外的其他对象预测出 y 值。拟合给定数据固然很好，但是我们真正感兴趣的是对从同一过程产生的未来数据的拟合情况，以使未来的预测尽可能地精确。为了实现这个目标，我们必须把这种建

模过程置于更正式的推理框架内。为此，我们假定每个观察值 $y(i)$是由预报变量$\alpha^T\mathbf{x}(i)$的加权求和和一个随机项 $\epsilon(i)$产生的，并且 $\epsilon(i)$服从独立于其他值的 $N(0,\sigma^2)$ 分布。（注意这里隐含假定了对于预报向量的所有可能值来说，随机项的方差都是相同的——σ^2。在下面我们 372
将进一步讨论这个假定。）于是 $n\times1$ 的随机向量 $\mathbf{Y}$ 的形式为 $\mathbf{Y}=\mathbf{X}\alpha+\epsilon$。公式 11.3 中的 $n\times1$ 的向量 $\mathbf{y}$ 是来自这个分布的一个实现。$n\times1$ 的向量 ε 的分量经常被称为误差。注意它们与残差 e 不同。误差是来自一个给定分布的随机实现，而残差是拟合后模型和观察到 y 值间的差异。也该注意α不同于 $\mathbf{a}$。α 代表了潜在的未知真实值，而 $\mathbf{a}$ 代表了实际模型中所取的值。

可以证明在这个框架内对 $\mathbf{a}$ 的最小二乘法估计也就是对α的极大似然估计。此外，前面得到的 $\mathbf{a}$ 估计的协方差矩阵为$(\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$，这个协方差矩阵表示了在对参数 $\mathbf{a}$ 的估计中的不确定性。在仅有一个预报变量的情况中，下式给出了截距项的方差：

$$\left(1+\frac{n\bar{x}^2}{\sum_i(x(i)-\bar{x})^2}\right)\frac{\sigma^2}{n} \qquad (11.6)$$ 373

并且下式给出了斜率的方差：

$$\frac{\sigma^2}{\sum_i(x(i)-\bar{x})^2} \qquad (11.7)$$

其中 $\bar{x}$ 是这个唯一预报变量的均值。前面的 $\mathbf{a}$ 的协方差矩阵的对角线元素给出了回归系数的方差——可以用这些数字来测试某个回归系数是否和零有显著的差异。 如果 v_j是$(\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$的第$j$个对角线元素，那么可以把 $a_j/\sqrt{v_j}$ 的比率和分布 $t(n-p-1)$ 比较来看回归系数是否为零。然而，正如下面将要讨论的，这种测试仅当模型中包含了另一个变量时才有意义，对更加精密的建模过程来说，可以使用其他的方法，这也将在下面讨论。如果 $\mathbf{x}$ 是新对象的预报向量，预测出的 y 值是 $\hat{y}$，那么 $\hat{y}$ 的方差是 $\mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}\sigma^2$。对于仅有一个预报变量的情况，这简化为$\sigma^2(\frac{1}{n}+(x-\bar{x})^2/\Sigma(x(i)-\bar{x})^2)$。注意 $\mathbf{x}$ 距离训练样本的均值越远，这个方差越大。也就是说，从方差角度看，对那些处于预报变量分布末端的对象作出的预测是精度最低的。也请注意基于这个方差的置信区间（参见第 4 章）就是对 y 预测值（predicted value）的信赖度（confidence value）。

我们还可能对预测区间（prediction interval）（这样称呼多少有些使人困惑）感兴趣，因为它给出了对于给定的 x 值，y 观察值的可能范围，而不是预测值的可能范围。预测区间必须包含来自预测的不确定性，以及 y 相对预测值的变化所导致的不确定性。这意味着还该在上面的方差中加入一项σ^2，于是得到$\sigma^2(1+\frac{1}{n}+(x-\bar{x})^2/\Sigma(x(i)-\bar{x})^2)$。

例 11.1 线性回归的一种最重要特例是当仅有一个预报变量的时候。图 11-3 显示了在 1984 年的苏格兰登山赛中时间（单位:分钟）对里程（单位:英里）的散点图。对这些数据的简单线性回归估计出截距值为-4.83，回归系数为 8.33。大多数的现代数据分析软件包都会给出相关的估计标准误差，以及对零假设（这里的零假设是：产生数据的真实参数等于零）的显著性检验。在本例中，标准误差分别是 5.76 和 0.62，显著性概率为 0.41 和 <0.01。据此我们可以得出结论：有充足的证据说明 374

确实存在正的线性关系，但是没有证据证明截距不是零。从图 11-3 的散点图可以看出数据点对两个变量都显示出了明显的**不均匀性**（skewness）（越向图的右上角点越稀疏）。显然回归线的位置对图中右侧点的精确位置比对左侧点更加敏感。对结果可能影响很大的点被称为高**优势**（leverage）点——它们是估计性能的极端值所对应的点。确实产生很大影响的点被称为**影响点**（influential point）。例如，如果图 11-3 中最右边的点的时间为 100（里程仍然是 28），那么显然它会对回归直线产生很大的影响。高优势点的不对称性是我们所不希望的。我们可以通过降低稀疏性来弥补这个不足——比如在拟合回归直线前对这两个变量都进行对数转换。

11.2.3　拟合后模型的解释

可以这样解释多重回归模型中的系数：如果第 j 个预报变量 x_j 在所有其他变量保持固定
375 时增大一个单位，那么响应变量会增大 α_j。因此回归系数说明了每个预报变量的**条件效应**（conditional effect），也就是在所有其他预报变量保持恒定的条件下它对响应变量的影响。这一点是解释回归模型的关键。特别值得注意的是，与第 j 个变量相关的回归系数的大小将依赖于模型中其他的变量。如果我们是以序列化的（sequential）方式构建模型，那么这显然更加重要，因为当加入另一个变量时，已经在模型中的那些变量的系数将会变化。（这里存在一种例外的情况。如果各个预报变量是互不相关的，那么被估计的回归系数不会受模型中其他变量的存在与否影响。然而，虽然在人为设计的实验中这种情况是很普遍的，但是在数据挖掘所面对的次级数据分析中这是很少见的。）我们可以通过回归系数来比较预报变量的单位变化对响应变量所产生的影响，从这个意义上来说回归系数的大小说明了变量的相对重要性。还应该注意影响的大小依赖于测量预报变量所选取的单位。如果我们用公里来代替毫米来测量 x_1，那么与它相对应的系数要乘以一百万。这可能导致变量比较的困难，所以人们经常工作在标准化后的变量上——相对每个预报变量的标准偏差来衡量这个变量。

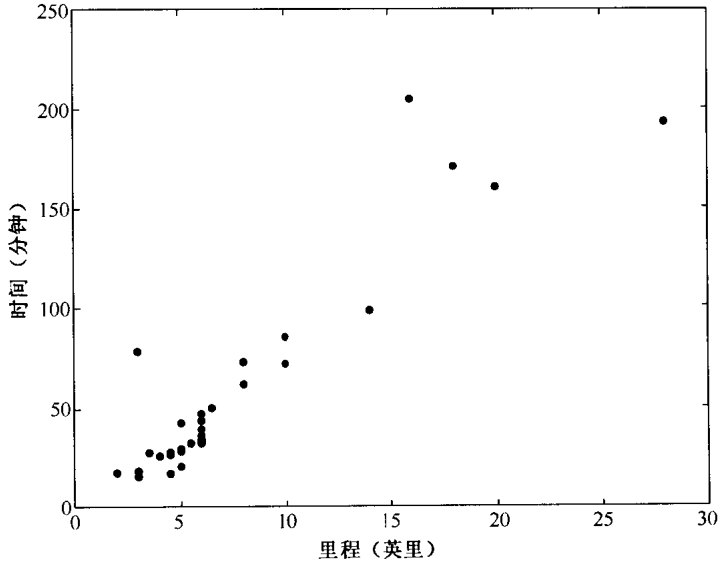


图11-3　记录时间（分钟）对里程（英里）的散点图。图中数据来自1984年的苏格兰登山赛

前面我们使用预测的和观察到的 y 值之间的误差平方和作为选取模型中参数值的标准。这就是*残差平方和*（residual sum of squares 或者 sum of squared residuals）：$\sum e(i)^2=\sum (y(i)-\hat{y}(i))^2$。在某种意义上来说，如果我们就用 $\bar{y}$（样本的 y 值均值）来预测所有的 y 值（也就是忽略模型的输入，不论 x 是什么，总是把输出猜测为 y 的均值），那么将得到最差的模型。我们把这个最差模型的误差平方和 $\sum (y(i)-\bar{y})^2$ 定义为*总平方和*（total sum of squares）。模型的残差平方和与总平方和的差别是可以说明模型的回归属性的平方和——因此称其为*回归平方和*（regression sum of squares）。下面是预测值 $\hat{y}(i)$ 和总体均值 $\bar{y}$ 间的差异平方和：$\sum (\hat{y}(i)-\bar{y})^2$。回归平方和与总平方和的比率被称为“多重相关系数（multiple correlation coefficient）”，经常使用 R^2 来表示它：

$$R^2=\frac{\sum(\hat{y}(i)-\bar{y})^2}{\sum(y(i)-\bar{y})^2} \tag{11.8}$$

如果 R^2 值接近 1，那么说明这个模型概括了数据中 y 的绝大部分变化信息。对每个平方和作出贡献的独立分量的数目被称为这个平方和的*自由度*（degrees of freedom）。总平方和的自由度是 $n-1$（比样本容量小 1，因为所有分量都是相对均值计算的）。残差平方和的自由度是 $n-1-p$（虽然在汇总中有 n 项，但它是 $p+1$ 个回归系数计算出的）。回归平方和的自由度是 p——残差平方和的自由度和总平方和自由度的差。 把这些平方和以及相关的自由度放在一起（如表 11-1 所示）可以很容易看出它们的差异。最后一列的含义将在下文介绍。 376

表 11-1 关于回归的方差分解分析表格

变化的来源	平 方 和	自 由 度	均 方
回归	$\sum(\hat{y}(i)-\bar{y})^2$	p	$\sum(\hat{y}(i)-\bar{y})^2/p$
残差	$\sum(y(i)-\hat{y}(i))^2$	$n-p-1$	$\sum(y(i)-\hat{y}(i))^2/(n-p-1)$
汇总	$\sum(y(i)-\bar{y})^2$	n-1	

11.2.4 推理和泛化

前面我们已经指出建立预测模型的目的就是推理：也就是对不知道 y 值的对象作出论断（预测）。这意味着拟合训练数据并非我们的真正目的。举例来说，不能仅仅因为估计出的回归系数不为零就推出这些变量是有关的：有可能完全是由于我们的模型恰巧捕捉了训练样本的特异性。对于数据挖掘的情况更是如此，因为在数据挖掘中要探索很多模型，而且这些模型是以一种比较自动的方式和数据拟合的。正如前面所讨论的，我们需要一种方式来检验（test）模型，看一看观察到的数据有多大可能性是随机产生的，即使并不知道产生样本数据的总体的结构。这种情况下我们需要检验总体的回归系数是否真的为 0，（当然，这不是我们感兴趣的唯一检验方法，但这是最经常需要的方法之一。）可以证明如果 α_j 的值确实都是 0（并且假定 $\epsilon(i)$ 服从 $N(0,\sigma^2)$ 的独立分布），那么 377

$$\frac{\sum(\hat{y}(i)-\bar{y})^2/p}{\sum(y(i)-\bar{y})^2/(n-p-1)} \tag{11.9}$$

服从 $F(p, n-p-1)$分布。这就是表 11-1 中的两个均值平方的比。检验是通过比较这个比例值和 $F(p,n-p-1)$分布的上限（upper critical level）来进行的。如果这个比例超过了这个分布的上限值那么这个检验就是显著的（也就是说发生了一个小概率事件）——我们可以得出结论：在 y 和变量 x_j 间存在线性关系。如果这个比例小于这个临界值，那么我们便没有证据来拒绝零假设（总体的回归系数都是零）。

11.2.5　模型搜索和建模

前面我们介绍了一种总的检验方法来观察一个给定模型中的回归系数是否都是零。然而，更常见的情况是要对模型空间进行搜索——分析各个模型以寻找一个从某种意义上来说“最好”的模型。尤其是我们经常需要向已经包含在模型内的变量集合中加入新的预报变量。注意这包括仅加入一个额外变量的特例，而且也要处理相反的情况，也就是从模型中删除变量。

为了比较模型我们需要一个评分函数。和以前一样，最明显的方案就是使用预测到的和观察到的 y 值间的误差平方和。假定我们在比较两个模型：一个模型有 p 个预报变量（模型 M），另一个是我们准备考虑的最大模型，它有 q 个变量（包含了我们认为有关的所有未经转化的变量，以及所有我们认为有关的这些变量的转化形式），我们称它为 M^*。因为每个模型都将有一个与之关联的残差平方和，所以这些残差平方和之间的差异可以告诉我们较大的模型比较小的模型更好拟合数据的程度。（或者等价的，我们可以计算回归平方和间的差异。因为回归平方和与残差平方和加起来是总平方和，而总平方和对这两个模型来说是相同的，所以这两种计算会得到相同的结果。）这两个模型的残差平方和之差的自
378 由度是 $q-p$，也就是拟合较大模型 M^* 要计算的额外回归系数。残差平方和之差和自由度之差的比例又给出了一种均方——两个模型间差异的均方。把这个均值与模型 M^*的均方加以比较便得到了对这两个模型间差异的 F 检验，如表 11-2 所示。从表中可以看出，是把这个比例

$$\left[\frac{(SS(M^*)-SS(M))}{(q-p)}\right]\Big/\left[\frac{(SS(T)-SS(M^*))}{(n-q-1)}\right]$$

和 $F(q-p\,,n-q-1)$分布的临界值进行比较。

表 11-2　用于建模的方差分解分析表

变化的来源	平　方　和	自　由　度	均　　方
回归模型 1	$SS(M)$	P	SS(M)/p
完全回归模型	$SS(M^*)$	q	SS(M*)/q
差	$SS(M^*)-SS(M)$	$q-p$	$\frac{(SS(M^*)-SS(M))}{(q-p)}$
残差	$SS(T)-SS(M^*)$	$n-q-1$	$\frac{(SS(T)-SS(M^*))}{(n-q-1)}$
汇总	$SS(T)$	$n-1$	

如果我们仅仅要比较几个模型这样做是很好的，但是数据挖掘问题经常需要依赖于自动

的建模过程。大多数的现代数据挖掘软件包都提供了这样的自动过程，有很多种不同的策略可以采用。一种基本的形式是*正向选取法*（forward selection），也就是每次向当前的模型中加入一个变量，在第 8 章中曾经提到过这种方法。具体来说就是每一步从潜在的变量集合中选取一个变量，选择的标准是可以使预测能力得到最大的提高（以残差平方和的降低来衡量），并且只要提高的幅度超过了一个预先指定的阈值就一直重复这个过程。理想的情况是，只要对预测能力的提高从统计角度来看是显著的，那么就该继续这种加入变量的过程，但是在实践中这是难以保证的：变量的选取过程必然包括很多并非都独立的检验，这使显著性值 379
的正确计算并不是一个简单过程。基于表 11-2 中的简单显著性水平不适用于进行多重依赖检验的情况。（这里隐含的另一个问题是，如果使用显著性水平来选取变量，那么它就是被用作了评分函数，因此就不该被赋予概率解释。）

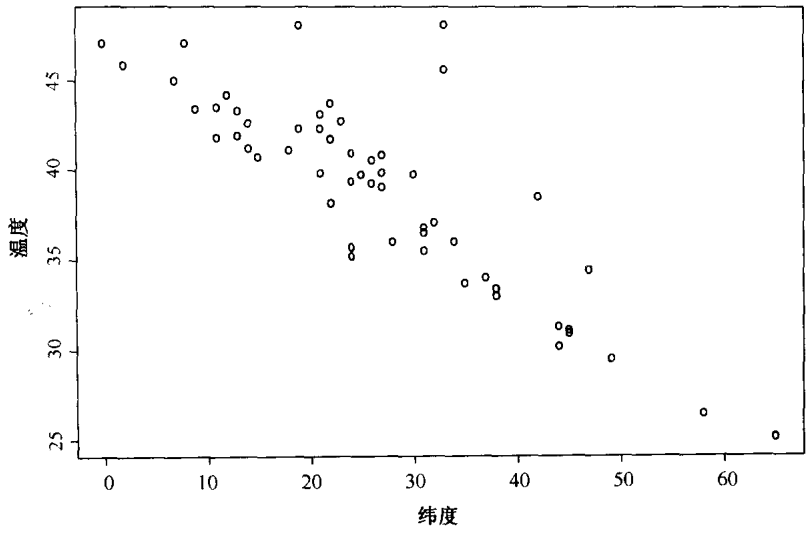
当然，在实践中我们可以使用第 7 章讨论的用于选取回归模型的任一种评分函数，比如 BIC、最短描述长度、交叉验证或者其他的贝叶斯方法。这些评分函数为我们提供了替代假设-检验框架（比较向模型中增加或删除项的统计显著性）的其他方案。像 BIC 这样带有惩罚的评分函数，以及交叉验证方法特别针对回归模型的变体是实践中选取回归模型的最常用评分函数。

与正向选取法相反的策略是*反向消除法*（backward elimination）。从我们可以考虑的最复杂模型（上面的最大模型 M^*）开始，逐步地消除变量，选取被消除变量的标准是使残差平方和的增长最小（也是由某个阈值来控制）。另一种变体是把正向选取法和反向选取法结合起来。举例来说，我们可以加入两个变量，删除一个，再加入两个，再删除一个，依此类推。对于变量数 p 特别大的数据集来说，从计算的角度来看正向选取法比反向选取法更可行。分步（stepwise）方法试图限制要搜索的预报变量可能集合空间，目的是使搜索易于驾驭。但是如果搜索的范围是受约束的，那么就有可能错过最有效的变量组合。很少的情况中（如果潜在的预报变量集合很小），我们可以分析变量的所有可能集合（尽管对于 p 个变量，存在（2^p-1）个可能子集）。通过使用像分枝定界这样的策略（依赖残差平方和的单调性），可以进一步扩大可分析问题的规模（参见第 8 章）。

有两点注意事项需要指出。第一，正像我们已经指出的，随着向模型中加入新的变量，模型中已经存在的变量的系数会逐渐的变化。所以当扩展模型时，对模型很重要的变量的系数可能变小。第二，正如我们在前面的章节中所讨论的，如果进行的搜索过于精细，那么过度拟合训练数据的可能性会很大——也就是说，得到的模型很好地拟合了训练数据（残差平方和很小），但是对新数据的预测效果却很差。 380

11.2.6 模型诊断和审查

尽管多重回归是一种强大而且应用广泛的技术，但是它的一些假定是有局限性的。例如，y 分布的方差在每个向量 $\mathbf{x}$ 处都一样这一假定经常是不合适的。（这个相同方差的假定被称为*同方差性*（homoscedasticity），相反的情况被称为*异方差性*（heteroscedasticity）。例如，图 11-4 所示为美国 56 个州的一月份正常平均最低温度（单位 F 度）相对于纬度（单位纬度）的散点图。有证据表明至少对于较低的温度，温度的方差随着纬度增大而增大（尽管温度的均值看起来是下降的）。在这种新的条件下，我们仍然可以应用前面的标准最小二乘算法来估计参数（而且如果模型的形式是对的，那么得到估计仍然是无偏的），但是因为可能找到具有更小方差的估计量，所以从这个意义上来说这么做并不是最好的。



381

图11-4　美国56个州的温度相对纬度的散点图

要达到更好的效果，我们需要对基本的方法进行修改。最关键的是，我们需要调整各个部分使以较大方差和 y 联系的 **x** 值在模型拟合过程中起到的作用较小。更直接地讲——也就是让更精确的值更大程度的影响估计量。具体来说，需要修改求解公式 11.5。假定 $n\times1$ 的随机向量 ϵ 的协方差矩阵为 $n\times n$ 的矩阵 $\sigma^2\mathbf{V}$（前面我们取 $\mathbf{V}=\mathbf{I}$）。方差不等的情况意味着 **V** 是一个对角阵，它的项不是全部相等的。那么我们可以（参见线性代数标准教科书）找到一个非奇异的矩阵 **P**，使 $\mathbf{P}^T\mathbf{P}=\mathbf{V}$。我们可以使用 **P** 来定义一个新的随机向量 $\mathbf{f}=\mathbf{P}^{-1}\epsilon$，并且容易证明 **f** 的协方差矩阵是 $\sigma^2\mathbf{I}$。利用这个想法，我们通过对老的模型乘以 $\mathbf{P}^{-1}$ 来得到一个新的模型：

$$\mathbf{P}^{-1}\mathbf{Y}=\mathbf{P}^{-1}\mathbf{X}\alpha+\mathbf{P}^{-1}\epsilon \tag{11.10}$$

或者

$$\mathbf{Z}=\mathbf{W}\beta+\mathbf{f} \tag{11.11}$$

这样便具备了应用标准最小二乘算法所需的形式。如果我们这么做，并把得到的解转变成包含原始变量 **Y** 的形式，那么便得到：

$$\mathbf{a}=(\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})\mathbf{X}\mathbf{V}^{-1}\mathbf{y} \tag{11.12}$$

这便是加权的最小二乘解。这个估计出的参数向量 **a** 的方差是 $(\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})^{-1}\sigma^2$。

y 分布的方差对于不同的 **x** 向量不等只是导致基本多重回归的假定不成立的一种情况。还存在其他的情况。因此我们真正需要的是找到一种可以探索模型质量的方式，以及可以使我们探测到模型在哪里和为什么背离了假定的工具。换句话来说，我们需要诊断模型的工具。在最简单的回归形式中，仅存在一个预报变量，我们可以根据 y 对 x 的曲线（参见图 11-1、图 11-2 和图 11-4）来观察模型的质量。但是更一般的情况下，预报变量并不只一个，这种简单的曲线方法是不可行的，因此必须使用更复杂的方法。通常，分析模型的最关键指标是残差，也就是向量 $\mathbf{e}=\mathbf{y}-\hat{\mathbf{y}}$ 的各个分量，如果这些分量中存在某种模式，那么说明这个模型
382 对数据分布的解释是失败的。可以使用各种包含残差的图形，包括残差相对拟合值的图形、

标准残差（通过把残差除以标准误差得到）相对拟合值的图形，以及标准残差相对标准正态分位点（quantile）的图形。（后者就是“正态概率图形”，如果残差近似地服从正态分布，那么这个图中的点应该大体位于一条直线上。）当然，解释这些诊断图需要实践经验。

适用于所有预测模型的一个一般性注意事项是：这样的模型仅在它所对应的数据范围内是有效的。把它推广到它所针对的数据范围外是很危险的。图 11-5 显示的简单例子说明了这一点。这幅图所画的是纸张的抗拉伸强度相对于纸浆中硬木含量的散点图。如果假定仅测量了纸浆中硬木含量值在 1 到 9 之间的这些样本，那么一条直线可以很好地拟合这个数据子集。对于硬木含量在 1 到 9 之间的新纸张样本，使用这条直线可以作出很好的预测。但是从图中非常清晰地看出如果使用这条直线来预测硬木含量值大于 9 的纸张的抗拉伸强度，那么得到的结果肯定是错误的，也就是说这个模型仅在它所对应的数据范围内是可信赖的。在第 3 章中我们介绍了另一个这样的例子，其中显示了每年流通的信用卡数量。一条直线可以很好地拟合 1985 年到 1990 年的情况——但是如果基于这个模型对这些年之外的情况作出预测，那么得到的结果肯定会有问题。 383

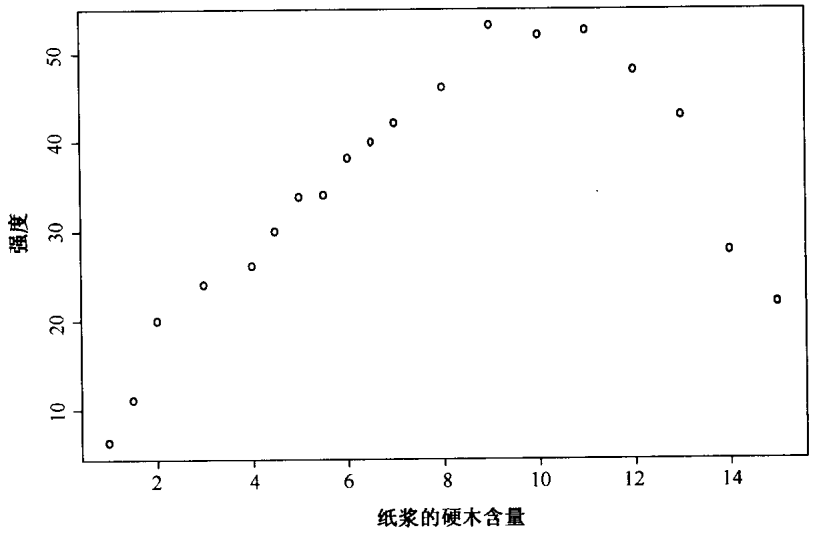


图11-5 纸张的抗拉强度相对纸浆中硬木含量的散点图

这些例子的情况非常清晰——它们仅包含几个数据点和单一的预报变量。但在涉及大量数据和很多变量的数据挖掘应用中，情况可能就没有这么清晰了，所以在作出预测时需要谨慎。

11.3 推广的线性模型

11.2 节中讨论了线性模型，在线性模型中响应变量被分解成两个部分：预报变量的加权求和以及随机分量：$Y(i) = \sum_j \alpha_j x_j(i) + \epsilon(i)$。出于推理的目的我们还假定 $\epsilon(i)$独立并服从 $N(0, \sigma^2)$ 分布。我们可以用另一种方式对此进行描述——把这个模型分成以下三部分来描述，以便对其进行推广。

(i) $Y(i)$是服从 $N(\mu(i), \sigma^2)$的独立随机变量。

(ii) 参数是通过求和 $v(i)=\sum\alpha_j x_j(i)$ 以线性方式进入模型的。

(iii) $v(i)$和$\mu(i)$是通过 $v(i)=\mu(i)$联系起来的。

这样便立刻出现了两种推广形式，而且保留了线性组合参数的优势。第一，我们可以放宽第(i)条中随机变量服从正态分布的要求。第二，我们可以推广第(iii)条中的联系表达式，以使用其他的连接函数 $g(\mu(i))=v(i)$ 把分布的参数和线性项$\sum\alpha_j x_j(i)$ 联系起来。这些扩展后的模型被称为推广的线性模型。它是二十年来数据分析方面最重要的成果之一，后面将看到，还可以把这个模型看作前馈神经网络的基本部分。

数据挖掘中使用的最重要的推广线性模型之一是 logistic 回归（logistic regression）。在第
384 10 章的 logistic 判别式中我们已经遇到了这种模型，在这里我们对其进行更加详细的讨论，并用它来阐述推广线性模型的基本思想。在很多情况下响应变量并不是像我们前面假定的那样是连续的，而是一个比例：一个给定样本中的昆虫遇到杀虫剂后死亡的比例，测验中答对题目的比例，箱子中腐烂橘子的比例。当这个比例仅来自 1 个对象时便产生了一种特例，即观察到的响应是二值的：某个昆虫死了还是没有，某个人答对了某一道题目还是没有，某个橘子是腐烂了还是没有。这正是我们第 10 章中讨论的情况，不过这里我们是把它放在了一个更通用的框架下。现在我们要处理的就是一个二值的响应变量，也就是取值为 0 或 1（对应于两种可能结果）的随机变量。我们假定第 i 个个体取值为 1 的概率是 $p(i)$，而且不同个体的响应是独立的，这意味着对第 i 个个体的响应服从伯努里分布：

$$p(Y(i)=y(i))=p(i)^{y(i)}(1-p(i))^{1-y(i)} \tag{11.13}$$

其中，$y(i)\in\{0,1\}$。这便是 logistic 回归对上面第（i）条的推广：伯努里分布代替了正态分布。

我们的目标是归纳出一个模型，它可以预测变量为 $\mathbf{x}$ 的对象取值为 1 的概率。也就是说，我们需要为响应的均值建立一个模型，即概率 $p(y=1|\mathbf{x})$。我们可以使用线性模型——对预报变量加权求和。然而这不是最理想的，最明显的原因是，线性模型可以取小于 0 或大于 1 的值（如果 $\mathbf{x}$ 值足够极端）。这暗示我们需要修改模型以引入非线性的特征。我们通对这个概率进行非线性转换，以便可以用线性组合对其建模来实现这个目的。也就是我们在第(iii)条中使用非线性的连接函数。一种合适的函数（并不是唯一可用的方案）是 logistic 连接函数（或者称为分对数（logit）连接函数）：

$$g\big(p(y=1|x)\big)=\log\frac{p(y=1|\mathbf{x})}{1-p(y=1|\mathbf{x})} \tag{11.14}$$

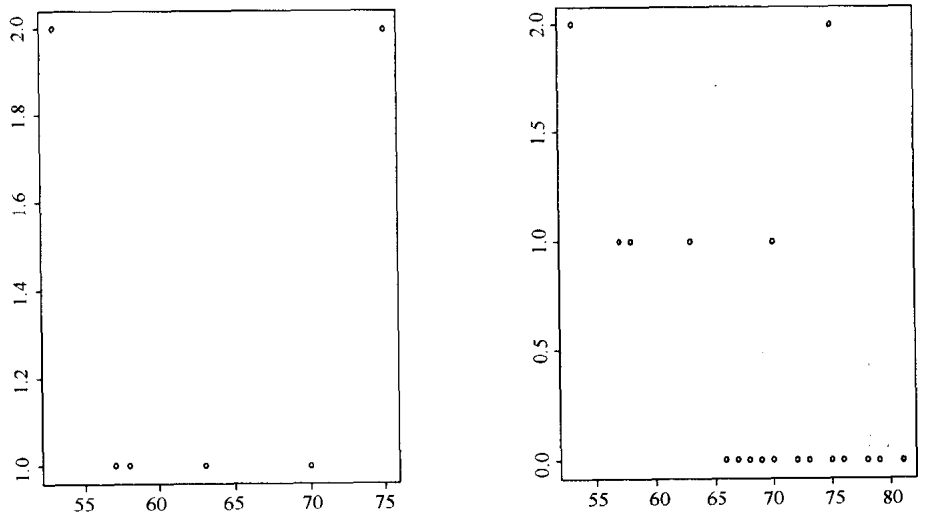
其中 $g(p(y=1|\mathbf{x}))$被模拟为$\sum\alpha_j x_j$。由于 p 是在 0 到 1 间变化的，所以 $\log(p/1-p)$显然是在$-\infty$和$+\infty$间变化的，刚好符合 $g(p)=\sum\alpha_j x_j(i)$的潜在范围。logistic 连接函数相对于其他候选方案的优点是易于解释。举例来说：

- 变换中的比例$\dfrac{p(y=1|\mathbf{x})}{1-p(y=1|\mathbf{x})}$是很熟悉的观察到 1 时的赔率（odds），而$\log\dfrac{p(y=1|\mathbf{x})}{1-p(y=1|\mathbf{x})}$是
385 对数赔率（log odds）。
- 如果给定一个新的预报变量的向量 $\mathbf{x}=(x_1,\cdots,x_p)$，可以根据$\log\dfrac{p(y=1|\mathbf{x})}{1-p(y=1|\mathbf{x})}$推出观察到 1 的概率。把第 j 个预报变量改变一个单位对这一概率的影响就是α_j。因此这个系数反应了对数赔率的差异——也就等价于从这两个值计算出的对数赔率。根据这一点容易看出e^{α_j}是当第 j 个变量改变一个单位时赔率变化的因子（参见 11.2 节关

于多重回归情况下变量单位变化的影响的讨论）。

例 11.2 1986 年 1 月 29 日挑战者号航天飞机在升空两分钟后爆炸，舱内人员全部死亡。这个航天飞机的两个火箭推进器是由多片拼接构成的，每三片的衔接处被一个橡胶 O 形环密封，一共有六个环。人们知道这些 O 形环对温度是敏感的。在以前的飞行中曾经有过 O 形环损坏的记录，而且有当天的温度数据。以往的最低温度是华氏 53 度。在挑战者号航行的这一天温度是华氏 31 度，所以对这一天是否该继续飞行有很多争论。一种观点是建立在对至少导致一个 O 形环损坏的以前七次飞行的分析基础之上的。预测温度导致 O 形环损坏概率的 logistic 回归分析得到的结果是斜率为 0.0014，标准误差是 0.0498。由此预测出在华氏 31 度 O 形环损坏概率的分对数是 1.3466，得到的预测概率是 0.206。这个模型中的斜率是正的，这表明 O 形环在低温下损坏的情况即便有概率也是很低的。而且，这个斜率和 0 的差异并不很显著，所以没什么证据可以说明损坏的概率和温度有关系。

这一分析是很不完善的。首先，华氏 31 度远远低于华氏 53 度，所以是在模型所对应的数据范围外使用模型——我们在前面曾警告的情况。第二，在没有导致 O 形环损坏的 16 次飞行中存在很多有价值的信息。对图 11-6a 和图 11-6b 进行比较，立刻可以明显的看到这一点，图 11-6a 显示了上面提到的 7 次飞行中 O 形环损坏的数量（纵轴）相对温度（横轴）的散点图，图 11-6b 显示了所有 23 次飞行的情况。这 16 次飞行时的温度都比较高。拟合图 11-6b 中数据的 logistic 模型估计出的斜率是-0.1156，标准误差是-2.46（估计的截距是 5.08，标准误差是 3.05）。据此得出华氏 31 度时的预测概率是 0.817。这是一个完全不同的结论，如果在飞行前已经研究了所有的数据那么便可以预先推出这个结论。 386



a）飞行前分析的数据　　b）完整的数据

图11-6 O形环损坏数量对飞行当天温度的散点图

387

因此推广的线性模型具有三个主要的特征：

(i) $Y(i)$（$i=1, \cdots, n$）是独立的随机变量，服从相同的指数族分布（参见下文）。

(ii) 预报变量是以 $v(i)=\sum a_j x_j(i)$的形式组合的，称为线性预报量（linear predictor），其中 a_js 是对α_js 的估计。

(iii) 给定的预报向量的均值$\mu(i)$和(ii)中的线性组合是通过连接函数 $g(\mu(i))=v(i)=\sum a_j x_j(i)$联系起来的。

指数族分布是很重要的一族分布，包括正态、泊松、伯努里和二项分布。可以用一个通用的形式来表示这一族分布：

$$f(y;\theta,\phi)=e^{\frac{y(\theta)-b(\theta)}{\alpha(\phi)+c(y,\phi)}} \tag{11.15}$$

如果ϕ是已知的，那么θ 被称为自然（natural）参数或正规（canonical）参数。当$\alpha(\phi)=\phi$ 时（很多情况下是这样），ϕ 被称为分散（dispersion）参数或范围（scale）参数。简单的代数变换便可以得出该分布的均值为 $b'(\theta)$，方差为$\alpha(\phi)b''(\theta)$。注意方差和均值是通过 $b''(\theta)$相联系的，而且有时把这（表示为 $V(\theta)$的形式）称为方差函数。在上面的第（i）和第（iii）条模型描述中对连接函数并没有任何限制。然而（这也是选择指数族分布的原因），如果连接函数把所选分布的正规参数表示为线性和的形式，那么会更加简单。对于多重回归，连接函数便是恒等式；对于 logistic 回归，它便是前面给出的 logistic 变换。对于泊松回归（第（i）条中的分布是泊松分布），连接函数就是对数连接函数 $g(u)=log(u)$。

根据推广的线性模型进行预测需要把关系 $g(\mu(i))=\sum a_j x_j(i)$反过来。在最小二乘估计算法中，这是非常简单直接的，实质上只要对矩阵求逆便可以了。然而，对于推广的线型模型，事情要更复杂一些：非线性意味着必须采用迭代方案。在这里我们不准备介绍其详细数学过程，不过不难证明通过解以下方程可以得到最大似然解：

388

$$\sum_{i=1}^{n}\frac{x_j(i)(y(i)-\mu(i))}{a_i(\phi)V(\mu(i))g'(\mu(i))}=0,\quad j=1,\cdots,p \tag{11.16}$$

其中 $a_i(\phi)$和$\mu(i)$中的下标 i 是为了说明对于不同的数据点这些量是不同的。应用标准的 Newton-Raphson 方法（参见第 8 章）可以得出迭代方程：

$$\mathbf{a}^{(s)}=\mathbf{a}^{(s-1)}-\mathbf{M}_{s-1}^{-1}\mathbf{u}_{s-1} \tag{11.17}$$

其中 $\mathbf{a}^{(s)}$代表第 s 次迭代的向量值$(a_1,\cdots,a_p)$，$\mathbf{u}_{s-1}$ 是对数似然的一阶导数向量，是在 $\mathbf{a}^{(s-1)}$点计算的，$\mathbf{M}_{s-1}$是对数似然的二阶导数矩阵，也是在 $\mathbf{a}^{(s-1)}$点计算的。

另一种可选的方法是“评分”法（这是一个传统的名字，请不要与我们在“评分函数”中所使用的评分一词相混淆，尽管它们的意思是相似的），它用二阶导数矩阵代替 $\mathbf{M}_{s-1}$。可以把这种方法的迭代步骤表示为一种类似于标准最小二乘矩阵解（11.5）的加权扩展版本（11.12）的形式：

$$(\mathbf{X}'\mathbf{W}_{(s-1)}\mathbf{X})\,\mathbf{a}^{(s)}=\mathbf{X}'\mathbf{W}_{(s-1)}\mathbf{z}_{(s-1)} \tag{11.18}$$

其中 $\mathbf{W}_{s-1}$是对角矩阵，第 ii 个元素为在 $\mathbf{a}^{(s-1)}$求出的$\partial\mu(i)/\partial v(i)^2/var(Y(i))$；$\mathbf{z}_{(s-1)}$是一个向量，第 i 个元素为$\sum_j x_j(i)a_j+(y(i)-\mu(i))\partial v(i)/\partial\mu(i)$，也是在 $\mathbf{a}^{(s-1)}$计算的。因为该式与 11.12 很相似，所以这种方法被称为迭代加权最小二乘法（iteratively weighted least squares）。我们还需要一种尺度来衡量推广的线性模型的拟合度，这便是模型的偏离度（deviance）。实际上平方和是应用于线性模型的偏离度的特例。偏离度被定义为 $D(M)=-2(\log L(M;Y)-\log L(M^*;Y))$，实

质上就是模型 M 的对数似然和我们准备考虑的最大模型 M*的对数似然的差。可以把偏离度分解为平方和的形式以探索各类模型。

例 11.3 在一项关于游泳者耳部感染的研究中，287 位游泳者回答了以下这些问题：是否经常在海中游泳；更喜欢有沙滩还是没有沙滩的游泳环境；年龄；性别；给定时期内已经发生耳部感染的次数。其中最后一个变量是响应变量，我们的目标是寻找一个模型，它可以根据其他变量预测出耳部感染的次数。显然，标准的线性回归是不适用的：响应变量是离散的，而且作为一种计数，不太可能服从正态分布。 389
类似的，它也不是一种比例，不在 0 和 1 之间，所以利用 logistic 回归来建模也是不合适的。但是，把这个响应变量假定为服从泊松分布（参数由预报变量决定）却是合理的。因此可以用响应变量服从泊松分布、对数函数作连接函数的推广线性模型来根据其他变量预测耳部感染的次数，这便得到了表 11-3 所示的**偏离度分析表**。

为了检验零假设（响应变量和预报变量之间没有预测关系），我们把回归偏离度值（1.67，表格中第二列第一行）与自由度为 4（表格的第一行第一列）的卡方分布进行比较。这样得到的 p 值是 0.7962。这个值绝不算小了，表明没什么证据说明响应变量和预报变量是有关的。可见，并不是所有的数据都一定能产生可以作出精确预测的模型。

表 11-3 偏离度分析表

	自 由 度	偏 离 度	平均偏离度	偏 离 率
回归	4	1.67	0.4166	0.42
残差	282	47.11	0.1671	
汇总	286	48.78	0.1706	
变化	-4	-1.67	0.4166	0.42

在结束这一小节前，有必要说明一下公式 11.16 的属性。尽管它们是在假定随机变量服从指数族分布的前提下推导出的，但是分析表明这些估计公式仅使用了均值$\mu(i)$、方差$a_i(\phi)V(\mu(i))$以及连接函数和数据。和分布的其他特征并没有关系。这意味着即使我们不准备做严格的分布假定，我们也可以估计线性预报量 $v(i) = \sum a_i x_j(i)$中的参数。因为在这种方法中没有明确表达出完全的似然，所以被称为准似然估计（quasilikelihood estimation）。当然这种方法也需要迭代。 390

11.4 人工神经网络

人工神经网络（ANN）属于高度参数化的统计模型这一大类（在后面的几节中将简略描述这一类中的其他模型）中的一种，近年来它受到了相当大的重视。在这里我们仅讨论前馈神经网络（feed-forward neural network），也就是多层感知器（multilayer perceptron）（参见第 5 章）。在一节的篇幅内，我们或许仅仅能揭开这一主题的表层，为此我们在后面给出了一些合适的补充读物。ANN 的高度参数化特征使它特别灵活，以至于它可以精确的模拟出函数中非常小的不规则性。另一方面，正如我们前面所指出的，这样好的灵活性意味着非常

严重的过度拟合风险。事实上，早期的（上个世纪 80 年代）研究就是因为神经网络对训练数据的过度拟合而受阻。近年来，人们开发出了克服这一问题的策略，使 ANN 成为一种非常强大的预测模型。

为了理解 ANN，不妨先回忆一下上一节介绍的推广的线性模型，推广的线性模型先对预报变量进行线性组合，然后对组合的结果作非线性变换。前馈 ANN 也是以此作为基本元素的。不过，ANN 不是仅仅使用一个这样的元素，而是使用许多这种要素构成的多个层。一个层的输出——每个基本元素的线性组合的转换结果——又作为下一层的输入。在下一层中，又以同样的方式来组合输入——对每个元素进行加权汇总，然后再作非线性转换。从数学角度来看，对于在输入变量 x 和输出变量 y 之间仅有一个转换层（一个隐藏层）的网络来说，我们有

$$y=\sum_k w_k^{(2)} f_k\left(\sum_j w_j^{(1)} x_j\right) \tag{11.19}$$

其中 w 是线性组合的权，f_k 是非线性变换。这个变换的非线性性是至关重要的，因为不然的话，这个模型就变为线性组合的线性组合——最终还是一种线性组合。之所以叫网络是由这种模型结构的图形表示得来的，在图形表示中，预报变量和每个加权和是节点，用边把和式中的各个项连接到节点。

391 ANN 可以使用的层数是没有限制的，不过可以证明一个隐藏层（层中具有足够的节点）足以模拟任何连续的函数。当然，这一结论的实用性依赖于现有的数据，出于其他目的（比如可解释性）使用多个隐藏层可能更加方便。也有很多推广的形式，在一种推广形式中可以跨越层，一个节点的输入不仅来自于它紧邻的前一层，而且也可以来自前面的其他层。

ANN 的最初形式使用阈值 logistic 单元作为非线性变换：如果输入的加权和小于某个阈值那么输出为 0，不然为 1。然而，为这些函数采用可微的形式具有数学上的优势。在应用中，两种最常见的形式是对加权和进行 logistic 变换 $f(x)=e^x/(1+e^x)$和正切双曲线 $f(x)=\tanh(x)$ 变换。

在上一节中看到，当从简单的线性模型转到推广的线性模型时，参数估计变得更加复杂。当从推广的线性模型转到 ANN 时复杂度又进一步增加了。对于模型中的参数数量（线性组合中的权）和变换的非线性性来说，这是很正常的。不过，这种复杂性限制了 ANN 在涉及庞大数据集的数据挖掘问题中的应用。（但是缓慢的估计和收敛速度并非总是坏事。有很多业内传闻讲 ANN 中的严重过度拟合问题神奇地消失了，这就是因为估计过程被提早终止了。）人们已经提出了很多种不同的估计算法。一种流行的方法是最小化由输出值（目标值）和预测值之间的偏差平方和构成的评分函数，做法是使该评分函数相对权参数最陡峭下降。可以把这一过程表示为一系列步骤，逐步更新各层的权，从输出节点反向考虑输入节点。由于这个原因，这种方法被称*反向传播*（back-propagation）。也可以使用其他的标准，当 Y 仅取两个值时（这时的问题实际上就是有指导的分类，和第 10 章中所讨论的相同），一种更自然的评分函数是以用于伯努里数据的对数似然为基础的：

392
$$\sum_i\left[y(i)\log\frac{\hat{y}(i)}{y(i)}-(1-y(i))\log\frac{(1-\hat{y}(i))}{(1-y(i))}\right] \tag{11.20}$$

事实上，对于数据集合大小正常的实践应用来说，不同评分标准的效果似乎没什么差异。近年来对人工神经网络进行了大量的研究，研究者来自不同的领域，这导致了很多已经非常著名的概念和现象又在其他领域中被重复“发现”，同时也引入了很多不必要的新术语。

不过，对 ANN 的研究也开发出了一些新的通用模型形式，它们在本节中并未讨论。例如，*径向基函数*（radial basis function）网络用径向基函数代替了前馈网络中典型的 logistic 非线性变换函数。一种做法是在 **x** 空间中使用一系列具有指定宽度的 p 维高斯胞（bump）。输出被近似为这些胞函数的线性加权组合。模型的训练过程包括估计核的位置、宽度和核的权，使用的方式类似于第 9 章中所描述的训练混合模型。

11.5 其他高度参数化的模型

神经网络的显著特征是它提供了一种非常灵活的模型来近似各种函数。它备受媒体的关注，部分是因为这种强大性和灵活性，但或许还因为其名字所蕴含的吸引力。然而，它并非是唯一一种具于高度灵活性的模型。目前已经开发出了一些在某些情况下近似能力和神经网络等价的其他模型，其中一些还具有更易于解释和估计的优点。在这一节我们简要的讨论这当中比较重要的两种模型。其他的会在 11.5.2 节中提到。

11.5.1 推广的相加模型

我们已经讨论了如何把线性模型的思想扩展到推广的线性模型中。然而，*推广的相加模型*（generalized additive model）对线性模型又作了进一步的扩展。它们用预报变量的*转换*版本的加权和代替直接对预报变量加权求和。为了实现更大的灵活性，使用了非参数方法来估计预报变量和响应变量间的关系，例如使用核函数或样条平滑方法（参见第 6 章），这样推广的线性模型形式 $g(\mu(i))=\sum\alpha_j x_j(i)$就变成了 $g(\mu(i))=\sum\alpha_j f_j(x_j(i))$。这里等号右侧的项有时被称 393
为*相加预报量*（additive predictor）。这种形式的推广相加模型保留了线性模型和推广的线性模型的优点。尤其是 g 如何随某个预报变量变化不受任何其他变量变化的影响；而且解释起来更加容易。通过在每个 f 分量中包含多个预报变量可以很容易的进一步推广这个模型，但是这是以牺牲简单的相加解释为代价的。相加的形式还意味着我们可以分别分析每个平滑后的预报变量，来观察它拟合数据的好坏。

当 g 是恒等函数时，可以使用*反向拟合*（backfitting）算法来寻找近似平滑函数。如果相加模型 $y(i)=\sum\alpha_j f_j(x_j(i))+\epsilon(i)$是正确的，那么

$$f_k(X_k)=E\left(Y-\sum_{j\neq k}\alpha_j f_j(X_j(i))\mid X_k\right)$$

这样便得到了一种迭代算法，在该算法中，每一步平滑一个预报变量的“部分残差” $y=\sum_{j\neq k}\alpha_j f_j(x_j(i))$，直到这个平滑函数不再变化。当然，具体的细节还依赖于选取平滑函数的方法：使用核、样条，还是其他。

为了把这种相加的形式扩展到推广的相加模型，我们使用和前面把线性模型扩展到推广的线性模型相同的方法。我们已经简要描述了拟合推广的线性模型的迭代加权最小二乘算法。我们曾经说明这实质上是对调整过的响应变量的加权最小二乘解（由$\sum_j x_j(i)a_j+(y(i)-$

$\mu(i)\frac{\partial\eta(i)}{\partial\mu(i)}$ 所定义）的迭代。对于推广的相加模型，不再使用加权的线性回归，而是采用一种拟合加权的相加模型的算法。

394

例 11.4 在某些外科手术中，要使用药物把血压降得非常低。一旦手术结束了，停止用药，并希望血压尽快地恢复到正常水平。本例中的数据描述了停止用药后心脏收缩压恢复到 100 毫米汞柱的速度（分钟）。这里有两个预报变量：特定药物的使用剂量的对数和患者用药期间的平均血压。我们使用推广的相加模型来拟合这些数据，并利用样条（事实上是三次 B 样条（cubic B-splines））来实现平滑。图 11-7 显示了转换后的剂量对数（Log(dose)）相对于观察到剂量对数 Log(dose)的曲线；图 11-8 显示了用药期间转换后的血压相对与观察值的曲线。（在两条曲线中都存在某种明显的非线性性——尽管剂量对数曲线的非线性看起来仅是由于一个点造成的（译注：即曲线最左边的一点）。对新数据点的预测就是把从这两个分量分别作出的预测结果相加得出的。

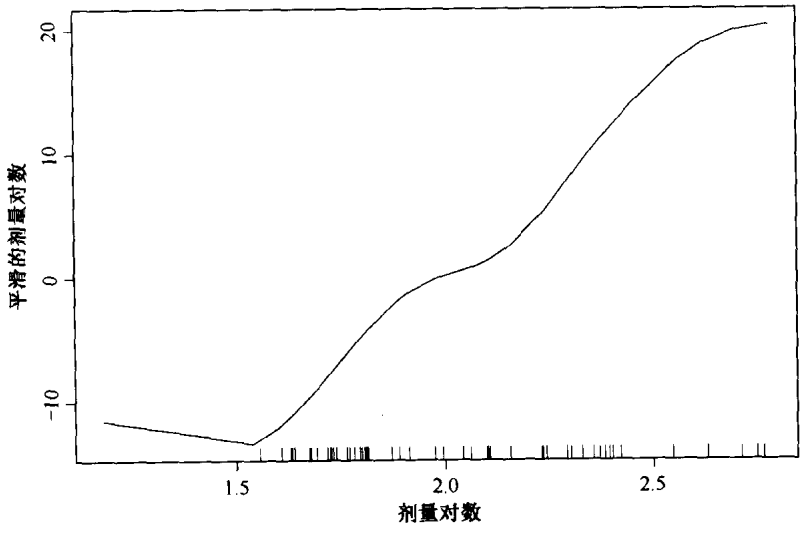


图11-7　剂量对数的转换函数。这是用于预测血压返回正常值的时间的模型中的一部分

11.5.2 投影追踪回归

可以证明投影追踪回归模型具有和神经网络模型相同的能力来估计任意函数，但是它的使用没有后者那样广泛。这或许是令人遗憾的，因为和神经网络相比，它在参数估计方面更有优势。上一节讨论的相加模型本质上是把焦点集中在单个变量上（虽然使用了这些变量的

395

转化版本）。可以把这种模型进行扩展，从而使每个相加的分量包含多个变量，但是没有明确的方法来选取最佳的变量子集。如果现有的变量数目非常庞大，那么我们也会面临组合爆炸的风险。基本的投影追踪回归模型的形式是：

$$\mathbf{Y}=\alpha_0+\sum f_k(\alpha_k^T\mathbf{X})+\varepsilon \tag{11.21}$$

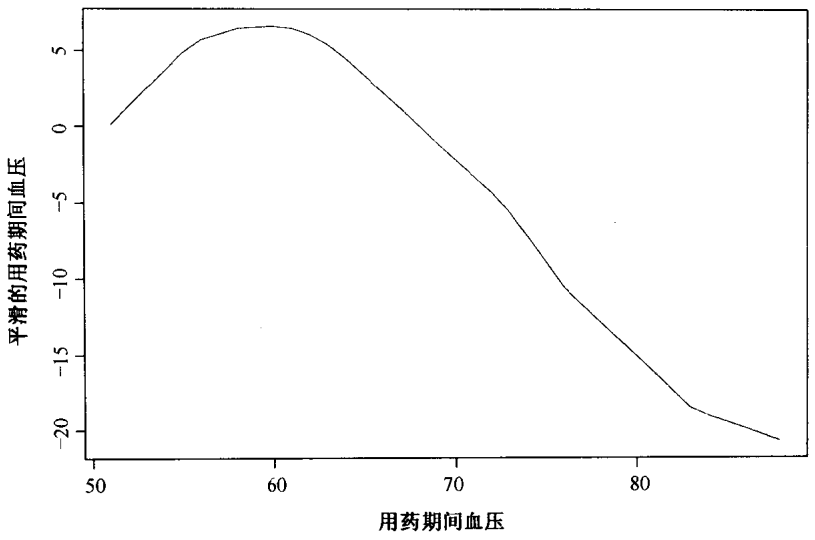


图11-8 用药期间血压的转换函数。这是用于预测血压返回正常值时间的模型中的一部分

这个形式和神经网络的形式非常接近——也是对原始变量的线性组合的变换（可以是非线性的）作线性组合。然而，这里并没有限制 f 函数的具体形式（神经网络中限制了函数的形式），通常通过平滑来寻找合适的函数形式，就像推广的相加模型中那样。因此这个模型是神经网络的推广。已经使用了各种平滑形式，包括样条法、弗雷德曼“超级平滑器”（在需要平滑的点作一个局部的线性拟合）以及多项式函数。之所以把这种模型称为投影追踪（projection pursuit）是因为可以把这种模型看作把 **X** 投影到α_k方向，并使这个投影方向对于某个目的来说最优。（在这种情况下，就是使预测模型的每个分量最优。）人们已经开发出 396
了很多算法来估计这种模型中的参数。一种方法是这样的，顺序的加入和的各个分量到某个最大值，然后再顺序的删除，每一次都根据模型对数据的最小二乘法拟合来选择加入和删除项。对于给定数量的项，使用标准的迭代过程来估计α_k向量中的参数，拟合模型。从计算的角度来看这个过程相当复杂，所以投影追踪回归法往往不适合于很庞大（n 很大）而且维数很高的（p 很大）的数据集。

11.6 补充读物

Draper and Smith（1981）的书是一本深入讨论传统线性回归的经典教材，讨论这个内容的其他教材还有很多（不计其数）。Furnival and Wilson（1974）介绍了经典的“跳跃定界（leaps and bounds）”算法，该算法可以高效的搜索回归模型中应该包含的最佳变量子集。关于推广线性模型的奠基性教材是 McCullagh and Nelder（1989）写出的，Hastie and Tibshirani（1990）则更全面的讨论了推广的相加模型。Friedman and Stuetzle（1981）引入了投影追踪回归（PPR），Diaconis and Shashahani（1984）给出了其理论近似结果。Friedman（1991）引入了一种非常灵活的用于多元回归的数据驱动模型，被称为 MARS（多元可适应回归样条（Multivariate Adaptive Regression Splines）。Breiman et al.（1984）介绍了树结构模型在回归中的应用，Weiss and Indurkhya（1995）介绍了用于基于规则的回归模型的有关技术。在分类背景下（第 10 章）介绍的自展技术也可以应用到回归中。当然也可以把回归置于贝叶斯

框架之下，比如 Gelman, Carlin, Stern, and Rubin（1995）有这方面的论述。

局部回归（local regression）技术依靠可适应的局部拟合来实现非参数的回归函数（参见 Cleveland and Devlin（1988）以及 Atkeson, Schall and Moore（1997）），它类似于用于密度估计的核模型（第 9 章）和用于分类的最近邻方法（第 10 章）。不过这种技术的计算量非常大，而且易于发生估计问题，就像高维空间中的局部核方法一样。

397 介绍神经网络的优秀著作包括 Bishop（1995）、Ripley（1996）、Golden（1996）、Ballard（1997）和 Fine（1999）。Ripley 的教材尤其值得关注，因为它完整而且广泛的讨论了来自神经网络、统计学、机器学习和模式识别领域的许多技术（而其他大多数教材往往只集中于这些领域中的一两个）。MacKay（1992）和 Neal（1996）介绍了训练神经网络的贝叶斯方法。

Hand et al.（1994）中给出了计算机 CPU 数据集、吸氧量数据集、游泳者耳部感染数据集和外科手术后的血压数据。温度和纬度数据来源于 Peixoto（1990）。Chatterjee, Hancock and
398 Simonoff（1995）文章中含有航天飞机数据的拷贝，Lavine（1991）讨论了该数据集。

第 12 章　数据组织和数据库

12.1　简介

数据挖掘区别于其他数据分析任务的特征之一是数据量。在很多数据挖掘任务中（比如网络日志分析），数据矩阵包含了上百万行和上千列，这使数据分析算法的效率问题非常重要。运行时间与行数 n 成指数函数的算法可能只适用于很小的数据集。有些操作的时间复杂度为 $O(n)$ 或 $O(n\log n)$，例如，统计数据频率，寻找离散变量或属性的波峰，以及对数据排序。通常这样的操作对于庞大的数据集也是可行的。然而，如果需要多次扫描数据集，那么即使是线性时间复杂度的算法，其开销也是高得惊人的。

除了数据集的行数 n 会影响算法的复杂度外，变量数 p 也是如此。对于某些应用 p 值非常小（比如说小于 10），但是在其他一些应用中，比如市场指数分析和文本文档分析，我们可能遇到具有 10^5 甚至 10^6 个变量的数据集。在这种情况下，我们就不能再使用包含 $O(p^2)$ 次操作的方法，比如逐对衡量所有属性间的关系。

不论是什么数据分析项目，都可以将其分成两个阶段。第一个阶段是准备分析算法所需的数据，第二阶段是运行分析算法。有人可能认为第一阶段不太重要，但是它却经常成为整个项目的瓶颈。例如，要分析一个数据集，往往有必要把算法应用到这个数据集的不同子集上。这意味着我们必须能够迅速地搜索和标识出每个子集，并且把这个子集装入内存。树算法有力地证明了这一点，在树算法中，数据集被逐步地分割成较小的子集，在扩展树之前必须标识出每个子集。*组织数据*（data organization）的目的就是找到一种方法来存储数据，以使对数据子群的访问尽可能快。即使是所有数据都可以放入内存，组织数据也是很重要的。

除了为数据挖掘算法提供高效的数据访问支持外，组织数据还在整个数据挖掘过程的重复和交互中起着重要作用。本章首先简要介绍现代计算机的存储器层次，然后介绍索引结构——数据库系统用其加速查询的过程。最后讨论了关系数据库和结构化查询语言，以及一些用于特殊目的的数据库系统。

12.2　存储器层次

计算机的存储器被划分成几个层次，访问不同层需要不同的时间（这里访问时间是指检索存储器中一个随机选取字节所需的平均时间）。事实上，如果磁盘存储也像高速缓存那样快，那么就不需要开发任何复杂的组织数据方法了。

以下是对不同存储器的一种通用分类：

1．处理器的寄存器。通常有不到 100 个寄存器，处理器可以直接访问寄存器中的数据；也就是说，访问寄存器不存在延迟。

2．处理器或主板上的高速缓存。这是实现在与处理器相同的一块芯片上或者位于主板上的高速半导体存储器。典型的容量是 16~1000K 字节，访问时间大约是 20 纳秒。

3．主存储器。标准的半导体存储器，容量从 16 兆字节到几个 G，访问时间大约是 70 纳秒。

4. 磁盘高速缓存。介于主存储器和磁盘之间的半导体存储器。

400 5. 磁盘存储器。容量从1G到上百G或上千G的庞大磁盘阵列。典型的访问时间是10毫秒。

6. 磁带。磁带可以存放几个G的数据。访问时间有所差异，可能是分钟级。

这些存储器之间的访问时间差异确实很大：在访问磁盘所需的10毫秒内可以访问高速缓存上百万次了。理解这一点的另一种方式是把访问时间假想成和实际距离成正比。那么，如果我们把到主存储器的距离想像为1米远（伸手可及的范围内），那么访问磁盘存储器的距离比这要远10^5倍，也就是100公里。

主存储器和磁盘的另一个主要差异是可以逐个访问主存储器的每个字节，然而，对于磁盘来说，只要我们访问一个字节，实际上是把包含这个字节的整个磁盘页（大约是4千字节）都调入主存储器。所以如果那一页恰好包含后面要使用的信息，那么它已经在高速的存储器中了。举例来说，如果我们要检索1000个整数，每个整数被存储为4个字节，那么需要访问磁盘的次数在1次到1000次之间，取决于这些整数被存储在同一个磁盘页上还是每个整数位于一个磁盘页。

针对存储器层次的物理特征，我们总结出了如下经验法则：

- 如果可能，数据应该存储在主存储器中。
- 在主存储器中，一起使用的数据项应该在逻辑上相互靠近（也就是说，我们可以快速地找到这个子集的下一个元素）。
- 在磁盘上，应该使一起使用的数据在物理上相互靠近（也就是尽可能在同一个磁盘页上）。

在实践中，系统使用者通常很难控制数据在高速缓存中的存放细节，以及数据在磁盘上的物理布局。正常情况下，系统尽可能把更多的数据载入主存储器，并自己决定如何把数据对象放到磁盘页上，用户可以影响为访问数据子集而创建的各种辅助结构。下一节将简要描

401 述用以访问海量数据的一些数据结构。

12.3 索引结构

组织数据的首要目标是找到一种方式以迅速定位到符合某个给定选取条件的数据点。通常，选取条件是一些针对单个属性的条件的合取（并），比如“**年龄** ≤40”并且“**收入**≤20 000”。我们首先考虑特别适用于仅有一个合取项的数据结构。

对属性 A 的索引就是这样一种数据结构：使用它来定位具有给定 A 值的数据点比直接扫描整个数据集更有效。通常使用B*-树或哈希函数来建立索引。

12.3.1 B-树

搜索树（search tree）可能是最简单的索引结构了。假定我们有一个由数据向量组成的集合 $S=\{\mathbf{x}(1),\cdots,\mathbf{x}(n)\}$，我们的目标是尽可能快地找到序数型（ordinal）属性（变量）为某个特定值的所有数据点。搜索树是一种二叉树结构，每个节点存储 A 的一个特定值，并且每个叶子有一个指针指向 S 的一个元素。此外树的结构是满足以下要求的：包含 a 值的树节点 u 的左子树的叶子所指向的所有 S 元素的 A 值都小于或等于 a。类似地，u 的右子树的叶子所指向的所有 S 元素的 A 值都大于 a。

有了属性 A 的二叉搜索树便很容易从 S 中找到属性 A 等于给定值 b 的数据点。我们只要从树的根节点开始，通过把 b 和节点上的值相比较来选择左子树或右子树。当到达叶子节点时，要么找到了指向 $A=b$ 的记录的指针，要么发现没有这样的“指针存在”。

也很容易寻找到满足条件 $b \leqslant A \leqslant c$ 的所有指针，即所谓的“区间查询”。只要定位到等于 b 的叶子（像上面那样），以及等于 c 的叶子，那么这两个位置之间的叶子指针所指向的记录就是要查询的记录。

找到属性 A 等于给定值的记录所需的时间与树的深度加上符合要求的记录条数成比例。在最坏的情况中，树的深度为 n（数据集 S 中的点数），但是有办法可以保证树的深度为 $O(\log n)$（不过这超出了本书的范围）。在实践中，二叉搜索树的应用不太多，因为下面要讨论的 B*-树在访问磁盘数据方面显然更有优势。 402

B*-树的基本思想和搜索树的思想是一样的：指向数据对象的指针在树的叶子节点上，内部节点包含了属性 A 的值表明如何找到某个指针。然而，在 B*-树中，对应于某个 A 值的每个内部节点不再仅有两个子女，它通常有上百个子女和值。

具体来说，一个数据集合的 M 度 B*-树具有如下特征：

- 所有的叶子在同一深度；
- 每个叶子包含 $M/2$ 到 M 个关键字（key）（可能是目标值）；
- 每个内部节点（可能要把根节点除外）有 K 个子女 $C_1, \cdots, C_K$（其中 $M/2 \leqslant K \leqslant M$）和 $K-1$ 个值 $a_1, \cdots, a_{K-1}$；对于所有的 i，所有存储在 C_i 子树叶子的关键值都大于 a_{i-1} 但不超过 a_i。

搜索 B*-树的方式和搜索二叉搜索树的方式一样：对于树的每个内部节点，用 a_i 值来选取正确的子树。

B*-树不同于二叉搜索树的一个特征是它的高度保证为 $O(\log n)$，因为所有的叶子是在同一深度的。实际上，树的深度是以 $\log_{M/2} n$ 为上限的。通常，M 值的选择标准是使树的每个节点适合于一个磁盘页。如果 M 是 100，那么 $(M/2)^5$ 超过 3 亿，而且我们发现对于大多数现实的 n 值（数据集的元素数），B*-树至多只有 5 层。这意味着用三次磁盘访问就可以完成从 3 亿个数据点中寻找一个数据点的单一属性值搜索，因为根节点和树的第二层可以放在主存储器中。大多数数据库管理系统使用 B*-树作为其索引结构之一。

12.3.2 哈希索引

还是假定我们有一个数据集 S，并且要寻找属性 A 等于 a 值的所有点。如果 A 的可能值集合很小，那么我们可以这样做：对于每个可能值，构建一个列表，使其包含指向属性 A 等于该值的所有数据点的指针。然后，对于给定的查询“寻找 $A = a$ 的点”，我们只要访问这个列表来寻找 a。

如果属性 A 存在大量的可能值，那么这种方法是不可行的：举例来说，我们不可能为 2^{32} 个整数中的每一个值维护一张列表。我们只能对原始值进行变换以缩小可能值的范围。 403

更详细些，令 $Dom(A)$ 为 A 可能值的集合。哈希函数（hash function）就是从 $Dom(A)$ 到 $\{1, \cdots, M\}$ 的函数 h，其中 M 是哈希表 r 的容量。对于每个 $j \in \{1, \cdots, M\}$，我们把 S 中 A 的值 a_i 满足 $h(a_i) = j$ 的记录 x_i 存入 $r[j]$。当要寻找满足 $A = a$ 的所有数据点时，我们只要计算 $h(a)$，并到 $r[h(a)]$ 中遍历数据指针的列表，对于列表中的每一个检查它的 A 属性值是否真的

为 a，或者是否为满足 $h(b) = h(a)$的另一个值 b（这被称为冲突（collision））。

典型的哈希函数是 a 余 M，M 为大于 n（数据点数）的一个适当质数。如果哈希函数选择得当并且哈希表足够大，那么冲突是很少的，并且搜索具有给定 A 值的点所需时间本质上和这些点的数量成比例。不过哈希索引不能直接支持区间查询。

12.4 多维索引

像哈希表和 B*-树这样的传统索引结构提高了访问表中各行的速度，它们是以给定属性或合成属性的值为基础的。然而在某些应用中，必须基于几个属性来表达选取条件，那么前面讲的索引结构就无能为力了。举例来说，考虑地理信息的情况。假定我们要寻找纬度介于北纬 30 度和 40 度，经度介于西经 60 度和 70 度，人口不低于 1 000 的所有城市。这样的查询被称为矩形范围查询（rectangular range query）。假定城市数据表很大，包含有上百万个城市名。应该如何求出这个查询呢？关于纬度属性的 B*-树索引可以找到满足纬度属性的城市，但是要在这些记录中找到满足经度条件的记录我们就不得不使用顺序扫描了。类似地，关于经度的索引也起不到多大的作用。我们所需要的是可以直接用于两个属性的索引结构。

多维索引（multidimensional indexing）是指根据多个属性的条件来搜索数据表记录的技
404 术。一种广为应用的方法是 R*-树。它的每个节点对应于潜在空间的一个区域，即这个节点代表了该区域中的各点。对于维数一直到 10 左右的情况，多维索引结构都会提高对庞大数据库的搜索速度。更高维数（比如说 100）数据集的范围查询还是一个在研究的课题。

12.5 关系数据库

在数据挖掘中我们经常需要访问数据的一个特定子集并根据这个子集的某些属性值来计算函数。我们已经讨论了一些数据结构，借助这些结构我们可以快速地找到有关数据点。但关系数据库提供了一种统一机制来快速访问数据的某一部分。

在数据库术语中，数据模型是指可以用来描述数据结构（structure）的结构（construct）和操纵数据的各种运算。（注意这里所使用的模型（model）一词和本书前面章节所讲的模型完全不同。在这里，它是设计出的一种强加到数据上的结构，而不是发现的已经存在于数据中的结构。模型一词出现两种不同的用法是令人遗憾的，这是由于统计学和数据库理论两个不同学科都对数据挖掘作出了贡献。幸运的是，混淆的时候很少，大多数时候都可以从上下文判断出使用的是两种含义中的哪一种。）关系数据模型是建立在以表格的形式来表示数据这一思想之上的。表格的头（图式（schema））由表名和列名集合构成；列名又被称为属性。实际的表（图式的实例）又被称为关系，是一个指定的行集。属性 A 所对应的列中的每一个表项是来自 A 定义域 $Dom(A)$中的一个值。注意在定义属性时，还必须确定每个属性的定义域。属性可以是任何的数据类型：范畴型的，数字型的，等等。表中行和列的顺序不是很重要。

我们可以用更正式的语言来描述上面的概念。一个关系图式 R 是一个属性集合$\{A_1, \cdots, A_p\}$，其中的每个属性 A_j具有与之相关的定义域 $Dom(A_j)$。模式 R 上的一行是一个映射 t: $R \rightarrow \cup_i Dom(A_j)$，其中 $t(A_j) \in Dom(A_j)$。模式 R 上的表或关系是 R 上的一个行集。关系数据库模式 $\mathbf{R}$ 是关系模式的集合$\{R_1, \cdots, R_k\}$（可能带有对关系实例的某些限制），模式 $\mathbf{R}$ 上的关系数据库 r由 R_i（对每一个 $i = 1, \cdots, k$）上的关系构成。

405

例 12.1 考虑带有条码机的零售店出口，或者一个记录每一笔交易的网络站点。对于每一笔交易，又称为一篮（basket），我们可以采集到这个顾客购买了哪些商品以及每件商品的单价信息。原则上讲，这些数据可以表示为一张表，每一种商品对应表中的一个属性，每一笔交易对应一行。矩阵中 t 行 A 属性的表项 $t(A)$表示了这个顾客购买了多少件 A。也就是说，每个属性 A 的定义域 $Dom(A)$ 是非负整数的集合。这种表的一个实例如图 12-1 所示，我们把这个表叫做 transactions。

transactions

basket-id	chips	mustard	sausage	Pepsi	Coca-Cola	Miller	Bud
t_1	1	0	0	0	0	1	0
t_2	2	1	3	5	0	1	0
t_3	1	0	1	0	1	0	0
t_4	0	0	2	0	0	6	0
t_5	0	1	1	1	0	0	2
t_6	1	1	1	0	0	1	0
t_7	4	0	2	4	0	1	0
t_8	0	1	1	0	4	0	1
t_9	1	0	0	1	0	0	1
t_{10}	0	1	2	0	4	1	1

图12-1 把购物篮数据表示为每个属性对应一种商品的表格

由于商品可能经常变化，所以把商品名编入属性不是一种好的做法。另一种表示方法是使用图 12-2 所示的 baskets 表，在这个表格中每种商品被表示为一个表项。这个表格有三个属性，**basket-id**（篮标识）、**product**（商品）和 **quantity**（数量），商品的定义域是所有字符串集合，数量的定义域是非负整数集合。由此可以看出把给定数据集表示为关系数据库的方式是不唯一的：transactions 表和 baskets 表都表示了相同的数据。

baskets

basket-id	product	quantity
t_1	chips	1
t_1	Miller	1
t_2	chips	2
t_2	mustard	1
t_2	sausage	3
t_2	Pepsi	5
t_2	Miller	1
	...	

图12-2 对购物篮数据的更理想表示

除了关于每笔交易的数据，零售商还要维护每种商品价格的信息。这可以表示成图 12-3 所示的 products 表。

products

product	price	supplier	category
chips	1.00	ABC	food
Miller	0.55	ABC	drink
mustard	1.25	DEF	spices
sausage	2.00	DEF	food
Pepsi	0.75	ABC	drink
Coke	0.75	DEF	drink
...			

图12-3 表示商品价格的表格

商品数据对于了解概要情况来说可能过于详细了。因此，零售商会把不同的商品分成一些较大的商品类目。图 12-4 显示的就是这样的一个例子。

product-hierarchy

Product	category
Pepsi	soft drink
Coke	soft drink
Budweiser	beer
Miller	beer
soft drink	drink
beer	drink
...	

图12-4 把商品层次表示成表格

这个表格描述了一种层次关系，表示了百事可乐和可口可乐是软饮料，软饮料和啤酒是饮料。

可以通过仅列出表名和它们的属性来简要地描述本例中的表格模式：

```
baskets(basket-id, product, quantity)
products(product, price)
product-hierarchy(product, category)
```

406
~
407

可见，关系数据模型是建立在表格表示这一思想之上的。单元格中的值可以是任何原子值，比如数字、整数或者字符串；但不允许用值的集合或列表。这就是说，如果我们要表示人的信息，那么我们可以表示他的年龄和电话号码，但不可以把多个电话号码存储在一个属性中。如果模型是按这一约束建立的，那么就说这个模型符合第一范式（first normal form）。

关系模型广泛的应用在数据管理中，几乎所有主要的数据库系统都是以这一模型为基础的。某些系统还提供了其他功能，比如可以使用面向对象的数据建模方法。

即使是在相当小的组织中，关系数据库也可能有上百个表格和上千个属性。所以管理这样的数据库模式可能是一项很复杂的任务。有时有人提出对于数据分析来说，把所有的表格合并成一个大的观察值矩阵（或者称为“大全表（universal table）”）就足够了，这样在数据挖掘中就不必关心数据是在数据库中这一事实。然而，对简单实例的分析说明这是不可行的：统一表太大以致于操作它的代价高得惊人。

例 12.2 考虑超市中商品的例子。在现实环境中。仅有一个包含商品（product）

和**价格**（price）两个属性的表格显然是不能满足应用要求的，至少还要有一个关于供应商属性的表格，包含**供应商**（supplier）、**地址**（address）、**电话号码**（phone number）等。如果我们要把这两张表格组合成一张表，那么这张表将必须包括以下属性：**交易标识**、**商品**、**数量**、**供应商地址**、**电话号码**、**商品价格**等。此外，如果每种商品平均属于 K 个不同的产品组，那么如果要包含**商品层次**（product-hierarchy）信息，表的大小将增大到原来的 K 倍。即使是对于中等大小的数据库，这种组合过程也会导致表格远远超过存储能力。 408

12.6 操纵表格

能够描述数据结构并存储数据对于数据管理来说还不够，我们还必须能够从表格中检索数据。本书简要地描述两种操纵表格集（也就是关系数据库）的语言：本节讨论关系代数，下一节讨论结构化查询语言（SQL）。关系代数是以集合理论表示为基础的，对于理论研究特别方便；而 SQL 在实践中应用的非常广泛。

在这个例子中，我们使用 r、s 等来表示表，用 R、S 等表示这些表的属性集。

关系代数包含了一系列基本运算来操纵以表格形式表示的数据，而且还可以使用一些导出运算（可以表达为一系列基本运算的运算）。这些基本运算包括三种集合运算：并、交和差，和用来删除列的投影运算，选择行的选择运算，以及组合两个表的联接和笛卡尔积运算。

例 12.3 关系代数的各种运算是这样定义的：

假定 r 和 s 是属性集合 R 上的表格。

并 $r \cup s = \{t \mid t \in r \text{ 或 } t \in s\}$

交 $r \cap s = \{t \mid t \in r \text{ 并且 } t \in s\}$

差 $r \setminus s = \{t \mid t \in r \text{ 并且 } t \notin s\}$

投影 如果给定 $X \subseteq R$，那么 $r[X] = \{t[X] \mid t \in r\}$，其中 $t[X]$ 是仅保留 t 行中 X 列的值而得到的行。

选择 如果给定对表 r 中行的条件 F，那么

$$\sigma_F(r) = \{t \in r \mid t \text{ 满足 } F\}$$

联接 $r \bowtie s = \{tu \mid t \in r, u \in s, \text{对于所有的 } A \in R \cap S,\ t[A] = u[A]\}$，其中 tu 是把 t 和 u 拼在一起而得到的行。 409

集合运算

表是行的集合，而且关系代数中的所有运算都是面向集合的：它们以集合作为输入并输出集合作为结果。因此我们可以用关系来编写查询：查询的结果和参数都是关系。

传统的集合运算对于操纵表也是有价值的。我们把并、交和差（分别表示为 $r \cup s$，$r \cap s$，$r \setminus s$）作为关系代数中的基本运算。并运算把相同属性集的两张表合并起来：$r \cup s$ 的结果包含 r 和 s 中出现的所有行。交运算 $r \cap s$ 所产生的表包含既出现在 r 中又出现在 s 中的行。差运算 $r \setminus s$ 得到的是出现在 r 中但没有出现在 s 中的行。这些运算都假定 r 和 s 是相同属性集上的表。

举个例子来说，假定 r 是表示所有软饮料价格的一张表，s 是表示最高价格为 2 美元的所有商品的表。那么 $r \cup s$ 就是包括所有软饮料和最高为 2 美元的商品的表，$r \cap s$ 就是不超过 2 美元的所有软饮料的表，$r \setminus s$ 包含了高于 2 美元的所有软饮料。当然可以用并和差运算来定义交运算：$r \cap s = (r \cup s) \setminus ((r \setminus s) \cup (s \setminus r))$。

运算时必须保证得到的集合是一张表，它具有一定的图式（schema）。所以 $r \cup s$，$r \cap s$ 和 $r \setminus s$ 定义的前提都是 r 和 s 是同一模式上的表——也就是在同一属性集上。

例如，可以使用交查询来建立规则集。（第 13 章将讨论用来学习规则的算法。）假定我们已经求出了满足条件 F 的观察值的表 r，而且类似地，另一张表 s 对应于满足条件 G 的观察值。交运算 $r \cap s$ 对应于满足这两个条件的那些观察值；交集的势（cardinality）反映了条件间的重叠程度。如果 r 和 s 是从观察值的同一张基本表中求出的，那么我们可以在这个查询中使用条件 $F \wedge G$ 作为选择条件。交查询的最自然应用是当我们需要检查同一个值是否出
410 现在两张表中的时候。

投影

投影运算的目的是剪裁一张表使其仅包含我们感兴趣的特定列。给定一个具有属性集 R 的表 r，并且 $X \subseteq R$，那么 r 在 X 上的投影是通过从表格中删除 X 之外的所有列而得到的。对表格投影的副作用是表的行数和列数会降低。如果 R 上的表被投影到属性集合 X 上，并且 R 上的表 r 包含和 X 属性值相同的两行，但是这两行关于 $R \setminus X$ 中的某些属性是不同的，那么投影后的行是完全一样的。这样的相同行经常被称为重复（duplicate）。既然表是一个集合，那么表中就不能包含重复，应该仅保留每种重复的唯一代表。因为这种特征是蕴含在集合概念中的，所以没有在投影运算的定义中再说明这一点。

商业化的数据库系统在这一点上经常和纯粹的关系模型不同。在实际的实现中，表被存储为文件。当然文件可以包含多条相同的记录。检查记录的唯一性需要大量时间，所以通常商业数据库中的表可以包含重复的记录。

关系数据库中的投影运算和向量空间中的投影有关但是并不相同。两种运算都是取一些点（在数据库中称为行）并产生低维空间中的一些点（属性减少的行）。在关系数据库中，我们仅可以投影到由属性直接定义的子空间上；对于向量空间，投影可以定义在任何子空间上（也就是说，基本向量（这里是属性）的线性组合）。

选择

选择运算用于从表中选择行。如果给定一个对表 r 中各行的布尔条件 F，那么对表 r 应用选择运算 σ_F 得到的表 $\sigma_F(r)$ 由 r 中满足这个条件的行构成。

选择是关系代数中使用最频繁的运算：每当我们要把焦点集中在表的一个特定行或行的子集时，我们都需要使用选择运算。在数据挖掘算法的实现中也经常出现选择运算。例如，在建立决策树时，我们需要选择出属于树的特定节点的记录列表。这个记录集合就是选择查
411 询的答案，查询的选择条件是在从树根到问题中这一节点这些节点中出现的条件的与。类似地，如果我们要使用关系代数来实现关联规则算法，那么就必须执行几个选择查询，每一个查询对应于满足一定条件（候选频繁集中的每个变量取值为 1）的记录子集。

在纯粹的关系代数中，选择是基于精确相等和不等的。对于数据挖掘来说，我们经常需要不精确的或者说近似的匹配。如果可以使用谓词 match 来近似匹配属性值（至少在某些数据库系统中可以这样），那么我们便可以直接使用这些数据库运算来选取满足近似匹配条件

的行。（第 14 章将更详细的讨论近似匹配。）

笛卡尔积和联接

投影和选择都是用来从表中删除数据的。联接（join）和笛卡尔积（Cartesian product）运算是用来把存储在两个不同表中的数据连接到一起。给定属性集分别为 R 和 S 的表 r 和表 s，并假定 R 和 S 是不相交的（也就是说，不存在同时出现在两个集合中的属性），那么 r 和 s 的笛卡尔积 $r \times s$ 是属性集 $R \cup S$ 上的表格，而且它包含了 r 中的任一行和 s 的任一行粘贴到一起可以得到的所有行。因此 $r \times s$ 有$|r|\,|s|$行，其中$|r|$是 r 中的行数。

在合并不同表中的行时需要用到笛卡尔积。但很少单独使用它；更多的时候是使用联接运算。给定一个选择条件 F，r 和 s 的联接 $r \bowtie s$ 是通过从 $r \times s$ 中选择满足条件 F 的行得到的。举例来说，我们可以使用等式 baskets.product = products.product 求出表 baskets 和表 products 的联接。这个运算的结果是一个具有如下列的表格：篮标识、产品名、数量和价格。（更精确地讲，这个结果中有两列产品名，分别来自两张原始表；我们可能需要投影掉其中的一列。）

在数据挖掘算法中联接的一个典型应用是组合不同来源的信息。举例来说，如果我们具有顾客的人口统计学信息和顾客的购买行为，这些数据通常是存储在不同表中的。要合并这里的相关数据条目，我们就需要进行联接运算。 412

12.7 结构化查询语言

关系代数是一种简洁而且实用的表示法。在数据库管理系统中，SQL（结构化查询语言）是被大多数数据库管理系统厂商所采用的标准。SQL 实现了关系代数的超集。这里我们仅介绍 SQL 程序中的一些基本结构。

基本的 SQL 语句是“select-from-where”形式的表达式或者查询，它的具体形式如下：

select $A_1, A_2, \cdots, A_p$
from $r_1, r_2, \cdots, r_k$
where 条件列表

这里，每个 r_i 是一张表，每个 A_j 是一个属性。直观的含义就是测试表 $r_1, r_2, \cdots, r_k$ 中的每个可能的行组合，看其是否满足条件。如果满足了条件，那么就输出由属性 A_j 的值组成的行。

查询的第二行，from 子句，指定了要应用 SQL 语句的表。第三行，where 子句确定了那些表中的行要被语句的结果所接受所必须满足的条件。第一行，select 子句，确定了参与表中的哪些属性会出现在结果中。它相当于关系代数中的投影运算（并非是选择运算）。“where”子句用来表示出现在选择和连接运算中的选择条件。对于一个选择运算来说，选择条件就是 where 子句的条件列表，是使用关键字 **and**、**or** 和 **not** 分隔开的。

例 12.4 可以使用下面的查询找到价格高于 2 美元的所有商品：

select product
from products
where price > 2.00 413

下面的语句可以找到至少包括一件价格超过 2 美元的商品的所有交易：

select basket-id, product, price
from baskets, products

where　baskets.product = products.product and price > 2.00

如果“from”子句中的一些表具有相同的属性，那么当这些属性名出现在“select”子句或“where”子句中时，必须在这些属性名前加上表名和点。如果希望参与表的所有属性都出现在结果中，那么可以用“*”代替“select”子句的属性列表。

数据库查询中的聚合（aggregation）是指把多个值合并成一个，比如通过求和或最大值这样的运算符。关系代数中没有聚合运算，但是 SQL 中有。一个聚合通常是从数据库中计算出一个量，它的值依赖于数据库的多行。

例 12.5　下面的查询说明了如何使用 SQL 来描述和超市销售有关的聚合查询。首先，我们寻找每种商品已经销售了多少份。要实现这个目的，我们使用 SQL 的 **group by** 运算。这个运算按照特定属性的值把输入关系的行分成组；SQL 语句中的其他运算是对每个组[⊖]分别执行的。

```
select item, sum(quantity)
from baskets
group by item
```

这个语句是这样执行的，先把 **baskets** 关系中的行按照属性 item 分成组，然后输出每组商品的名字和这组商品的数量之和。

下面的查询可以求出每种商品的总销售额。

```
select item, sum (quantity)* price
from baskets, products
where item = product
group by item
```

下面的查询可以求出属于软饮料的每种商品的总销售额。

```
select item, sum(quantity)*price
from baskets, products, product-hierarchy
where item = product and products.product = product-hierarchy.product and class=
      "soft drink"
group by item
```

414

SQL 语句是为传统数据库应用而开发的，比如生成报表、并发访问、实时更新很多用户的事务数据等等。因此，像这样的一种语言没有为实现数据挖掘算法提供很好的平台也就不足为奇，这样讲有两个原因：缺乏合适的原语（primitive）和效率的需求。

关于原语，在 SQL 中计数和聚合是非常简单的。所以举例来说，关联规则算法所需的运算使用 SQL 来访问数据是非常直截了当的。在建立决策树时，我们需要能够数出满足出现在从根到问题中节点这些树节点条件的记录数目，可以使用选择和计数查询来完成。然而 SQL 中的原语无法完成常见的统计运算，比如矩阵的转置，奇异值分解（SVD）等等。要使用 SQL 来做这些运算是非常麻烦的，这意味着拟合复杂模型通常是在数据库系统之外来进行的。

即使 SQL 原语足以表达数据挖掘算法中的运算，也还有很多理由要使用松耦合方式来

⊖ 译注：原文此处为“每个子句”，当属笔误。

实现算法，也就是把有关数据下载到算法中。原因是数据库管理系统和应用程序间的连接通常要给每个查询附加很大的额外开销。因此，虽然利用 SQL（比如说）来表示关联规则算法中的基本运算是很经典的，但是这样的算法通常非常缓慢。另外一个导致性能问题的原因是在关联规则算法（举例来说）中我们必须计算大量候选频繁集合的频率。在专门的实现中很容易通过遍历数据一次完成很多个这样的计数，然而在基于数据库管理系统和 SQL 的实现中，要为每个候选频繁集都提交一个独立的查询。

12.8 查询的执行和优化

可以用很多不同的方式来求解查询。举例来说，考虑下面这个查询：

```
select t.product
from baskets t, baskets u
where t.transaction = u.transaction and u.product = "beer"
```

415

这里“**baskets t, baskets u**”的含义是在查询中 t 和 u 是指表 baskets 的行。因为我们希望引用同一张表中的两组行，所以这种表示是必须的。这个查询是寻找那些含有啤酒的交易中购买的所有商品。

求解这一种查询的原始方法是试验 baskets 表中的所有可能行对，检查它们的 basket-id 属性是否一致，并检验第二行的产品属性中是否有“啤酒”。这将需要进行对行的 n^2 次运算，其中 n 是 baskets 表的大小。

一种更有效的方法是首先定位到 baskets 表中产品属性为“啤酒”的行，并把这些行的 basket-id 排列到一个列表 L 中。然后我们可以使用 basket-id 属性作为排序关键字对 baskets 表进行排序，并提取出 basket-id 出现在列表 L 中的行的 products 属性值。假定 L 是比较短的列表，那么这种方法需要 $O(n)$次运算来找到含有“啤酒”的行，$O(n\log n)$次运算来对行进行排序，以及 $O(n)$次运算来扫描排序的列表并选出正确的值；也就是一共需要 $O(n\log n)$次运算。这显然比前面原始方法所需的 $O(n^2)$次运算有很大改进。

查询优化的任务就是要为给定的查询找到最佳的求解方法。通常，查询优化程序把 SQL 查询翻译成一个表达式树，树的叶子表示表，内部节点表示对节点子女的运算。然后，可以使用运算间的代数等式把这个树转化成可以更快求解的等价形式。在前面的例子中，我们使用了等式$\sigma_F(r\bowtie s) = \sigma_F(r)\bowtie s$，其中 F 是选择条件，它仅关心 r 的属性。在找到了适当的表达式树后，便开始选择用于每个运算的求解方法。例如，可以使用几种不同的方式来求解联接运算：通过嵌套循环（就像上面的原始方法那样），通过排序，或者使用索引。每种方法的效率依赖于表的大小以及表中的值。因此，查询优化程序记录这些量变化的信息以发现好的求解方法。从理论上讲，寻找给定查询的最佳求解策略是一种 NP-困难（NP-hard）问题，因此寻找最佳方法是不可行的。不过，好的查询优化程序还是可以达到惊人的效果。

416

数据库管理系统力争为很大范围内的不同查询提供好的求解性能。因此，对于某个单一查询来说，有可能写一个程序求出结果比用数据库管理系统来求解它更高效，数据库管理系统的威力在于快速地执行大多数查询。在数据挖掘应用中，这是很有价值的，因为通常事先都是不知道查询的（比如在构建决策树的应用中）。

12.9 数据仓库和在线分析处理

带有顾客、交易、产品、价格等信息的零售数据库是业务数据库（operational database）的典型例子，业务数据库就是用于处理机构内日常业务操作的数据库，而且这些操作对数据库的依赖性非常强。运转数据库的其他实例包括机票预订系统、银行账户数据库等等。策略数据库（strategic database）是机构内用于决策的数据库。数据挖掘和决策支持是密不可分的，实际上可以说数据挖掘的主要目标就是决策支持。

通常，一个组织有几个不同的业务数据库。比如，零售部门会建立购物篮数据库、仓储数据库、客户数据库（一个或多个）、工资数据库、供应商数据库等等。事实上，一个有多种业务的服务性公司甚至有几个客户数据库。加起来，一个大的机构可能有成百上千的业务数据库。决策支持的目的就是把这些位于不同业务数据库中的信息组合起来并发现公司内部以及公司与客户的整个行为模式。建立直接访问业务数据库的决策支持系统是非常困难的。

像零售数据库、客户数据库或者机票预订系统这样的数据库大多时候是用来回答定义好的重复性查询的，比如“这个篮子中的所有商品的总价格是多少”，“客户史密斯的地址是什么”或者“账户 123456 的余额是多少？”这样的数据库必须支持大量的事务处理任务——对数据内容的查询和更新。数据库的这类用法被称为在线事务处理（OLTP）。

417 决策支持任务需要另一种类型的查询：最重要的是聚合。典型的决策支持查询可能是“找出所有产品按区域和按月份的销售额，并比较该结果与上一年的差异。”术语在线分析处理（OLAP）就是指使用数据库来总结数据，聚合是其主要机制。

例 12.6 假定零售数据库的各个表具有如下形式：

```
baskets(basket-id, item, quantity)
products(product, price, supplier, category)
product-hierarchy(product, category)
basket-stores(basket-id, store, day)
stores(store's name, city, country)
```

这里我们已经加入了一张表 basket-stores 用来说明某一笔销售是在哪个店在哪一天产生的。从决策支持的角度出发，使用下面的表可以更好的表示某一天在某个店销售某一产品的数量：

```
sales(product, store, date, amount)
```

我们可以用 SQL 语句向这个表格中加入行：

```
insert into sales(product, store, date, amount)
select item, store, date, sum(quantity)*price
from baskets, basket-stores, products
where baskets.basket-id = basket-stores.basket-id and item = product
group by item, store, date
```

然后，我们可以利用下面的查询找到所有产品大类在各国家的总销售额：

```
select products.product, store.country, sum(amount)
from sales, stores, dates, products
where dates.year ≥ 1997
        and sales.product=products.product
        and sales.store=stores.store
        and sales.date=dates.date
group by products.category, store.country
```

418

OLTP 和 OLAP 对数据库系统有不同的要求。OLTP 要求数据是最新的，允许查询修改数据，允许多个事务同时执行而互不妨碍，对反应速度要求很高等等。不过，OLTP 的查询和更新本身是比较简单的。与此相反，OLAP 的查询非常复杂，但通常在给定的时间里仅有一个查询在执行。OLAP 的查询不修改数据，而且如果是在探索上一年的销售情况，那么目前的销售信息并不是很关键的。可见二者的差异很大，以至于应该考虑使用不同的存储策略来处理这两种应用。

数据仓库（data warehouse）是以支持决策为目的用来存储不同业务数据库信息的数据库系统。零售商所使用的数据仓库可能包含来自以下这些数据库的信息：购物篮数据库、供应商数据库、顾客数据库等等。如果工资数据库对决策支持不是至关紧要的，那么在这个数据仓库中就不必包含这个数据库的数据。并不是仅仅把来自不同数据库的数据堆积到一个磁盘上就建立起数据仓库了。而是必须进行一些集成工作，例如解决一些属性名和用法可能存在的不一致，查明属性和值的语义等等。很多情况下，建立数据仓库都是要付出很高代价的，因为很多地方需要作手工的调整，而且要理解业务数据库的详细情况。

OLTP、OLAP 和数据挖掘间并不是界限分明的。就拿下面这些查询来说：寻找一个顾客的地址，寻找这个产品的上月销量，按地区和月份寻找所有产品的销量，寻找销量的走势，寻找哪些产品具有相似的销售模式，寻找可以预测某个产品区隔（聚类）销量的规则。通常，第一个查询是典型的 OLTP 查询，第二个是典型的 OLAP 查询，最后两个是数据挖掘查询，但是很难定义数据挖掘和 OLAP 间的界限。

12.10 OLAP 的数据结构

OLAP 需要对很大的数据库表进行不同的聚合计算。因为很多聚合要反复使用很多次，所以把其中的一部分存储起来是很有意义的。*数据立方体*（data cube）是一种以表格的方式观察不同聚合结果的巧妙技术。 419

在前面的例子中我们是用下面的模式来表示销售表：

```
sales(product, store, date, amount).
```

这个表中可能有这样的一行：

```
sales(red wine, store 1, August 25, 17.25),
```

这一行表示 1 号店在 8 月 25 日销售了红酒。如果虚构一个新的值 **all** 来表示所有的产品，那么我们可以把这样的行：

```
sales(all, store 1, August 25, 14214.70),
```

看作1号店在8月25日的所有产品销售额是14 214.70美元。从统计的角度来看，这给出了这个表的一个边际值，汇总了第一个属性的值。

这个销售表的*数据立方体*包含所有的行

```
sales( a, b, c, d),
```

其中，a、b 和 c 要么是对应属性定义域中的值，要么是特定值 all；d 是对应的和。也就是说，*数据立方体*是由原始表和所有边际表（一维的，二维的，直到对每一个属性分别汇总所得到的表）组成的。

12.11　字符串数据库

近年来，对文本和字符串数据库的兴趣迅猛地增长。分子生物学是导致这种增长的一个原因：现代生物科技产生了大量蛋白质和 DNA 数据集合，而这些数据集经常是以字符串的形式记录的。更重要的一个原因是网络的发展：搜索引擎需要高效的方法来寻找包含给定条件的文档。关系数据库擅于以表格方式存储数据，但是并不擅于表示和访问大量文本。最近，一些商业数据库系统已经加入了对高效查询庞大文本数据字段的支持。

对于一个给定的庞大文本汇集，一个典型的查询可能是“找出文本中出现‘挖掘’这个
420 词的所有地方”。更广泛地讲，这个问题是在文本 T 中寻找模式 P 出现的地方。模式 P 可能是一个简单的字符串，一个含有通配符的字符串，或者甚至是一个常规的表达式。P 在 T 中的出现可以被定义为精确的匹配或者是允许误差的近似匹配。

很明显，可以通过顺序的扫描文本并在每个位置测试是否和 P 匹配来寻找模式 T 出现的地方。但还有效率更高的方法，例如利用*后缀树*（suffix tree）数据结构我们可以在和模式 p 的长度成比例（而且不依赖于文本的大小）的时间内找到出现 p 的列表，并在时间 $O(|p| + L)$ 内输出出现 p 的地方，其中 L 是 p 在文本中出现的次数。构建后缀树的时间和原始文本的大小是线性关系，因此在实践中也是很快速的。

简单地讲，网络搜索引擎有两种数据结构：页面关系表 **pages（page-address，page-text）**和后缀树，后缀树包含了调入系统的所有文档的所有文本。当用户提交了一个查询，比如说“找出包含‘*数据*’和‘*挖掘*’这两个词的所有文档”，搜索引擎便使用后缀树来找出包含‘*数据*’一词的和包含‘*挖掘*’一词的两个页列表。假定列表是排序的，那么便可以直接找出两个词都出现的文档。然而应该注意包含两个词的文档数可能远远小于包含其中之一的文档数。

12.12　海量数据集、数据管理和数据挖掘

到目前为止我们讨论的焦点一直都集中在一般意义的数据库技术上，还没有讨论一个重要的问题：数据挖掘和数据库技术是如何相互配合的。我们对这种交互性的讨论将是比较简要的，因为到目前为止还没有一种被研究者和实践者们都公认为很好的方法来处理数据挖掘算法和数据库技术间的配合问题。主要问题是：很多海量数据集要么是已经被存储在关系数据库中，要么是如果把它们转换为关系数据库形式，那么在数据挖掘项目中就可以更高效地管理和访问它们。另一方面，大多数数据挖掘算法的焦点是建模和优化，并且实际上假定数据是驻留在主存
421 储器上的平面文件。如果被挖掘的数据主要是在磁盘上的，而且/或者是以关系形式（也许具有

SQL 接口）存储的，那么我们应该如何解决数据挖掘算法和数据的接口问题呢？

这就是数据管理（data management）所要解决的问题，正如第 5 章中所简要讨论的，大多数数据挖掘算法一般并不明确地指定数据管理方法。或许这确实是最灵活的做法，因为在实践中我们所采用的方法是由很多实际因素所决定的，比如说数据的数量、可供使用的主存储器的大小、需要重新运行算法的频繁程度等等。尽管如此，我们还是可以归纳解决这个问题的几种一般做法，具体讨论如下。

12.12.1 把数据都放入主存储器

最明显的一种方法就是看是否可以把数据存储在主存储器中让数据挖掘算法高效的存取，这是实践中被使用了多年的方法。由于主存储器技术的发展，随机访问存储器（RAM）的容量已经上升到 G 字节的范围，所以对于很多中等大小的数据分析应用，这种方法很实用。当然还有很多应用有数十亿条复杂事务，这种情况下我们不能指望很短时间内可以把这样的数据都调入主存储器。这时我们可以考虑选择数据的一个部分，或许可以产生一个随机样本，从而使要处理的记录数为 n'而不是 n（n'远远小于 n）。

我们也可以用某种方式来选择特征子集。比如说，本书作者之一所研究的某个应用涉及 1 000 个变量和 200 000 个顾客。决策树是根据 5000 个顾客的随机样本建立的，然后在 200 000 条记录的整个集合上使用最终决策树中变量的联合来建立模型（利用树、非线性回归和其他技术）。当然这完全是一种启发性的过程，在建模时某个重要的变量可能已经在根据随机采样得到的决策树中被遗漏了。尽管如此，它是“数据工程”中一个相当典型的例子，因为实践中经常需要在合理的时间内得到有意义的结果。还应注意从关系数据库中产生随机样本本身可能就是个重要而且复杂的过程。当然有很多方法改进了基本的随机采样思想，比如先取一个较小的初始样本来了解数据的总体“地形”（landscape），然后再以某种自动方式进一步提炼这个样本，等等。 422

当然即使主存储器可以容纳全部数据，我们也必须谨慎从事，很可能我们必须对数据进一步抽样以使数据挖掘算法的运行时间更短。此外，简单实现的算法在运行时可能产生大量的中间结构（例如数据矩阵的不必要拷贝），这也可能导致超出可供使用的主存储器的存储能力。因此不言而喻，即使是对于数据都已驻留在主存储器中的情况，保证算法实现在内存和时间方面的高效性仍然是很重要的。

12.12.2 数据挖掘算法的可伸缩版本

“可伸缩（scalable）”这个术语在数据挖掘文献中所表达的含义多少有些不严谨，但是我们可以认为它是指数据挖掘算法可以很好地适应记录数 n 和变量数 p 的增长。举例来说，当 n 增大到足够大时，决策树算法的朴素实现运行时间性能会急剧下降，原因是这个算法需要频繁地访问磁盘上的数据。在实践中，目前关于可伸缩性的研究更多地集中在 n 很大的问题上：p 很大的情况比 n 很大的情况更加复杂。

对可伸缩数据挖掘算法的一条研究路线是开发已有著名算法的可伸缩版本，可伸缩版本保证返回和原来（朴素）实现相同的结果，但是对于很大的数据集通常运行的更快。这种通用策略的一个例子是 Gehrke 等人（1999）的做法，他们提出了一族被称为 BOAT（用

于构建树的自展优化算法（Bootstrapped Optimistic Algorithm for Tree Construction））的算法。BOAT 方法对整个数据集扫描两次。在第一次扫描中，利用一个来自完整数据的较小随机样本（主存储器可以容纳）来构建“优化树”。第二次扫描分析初始树和假定利用所有数据建立起来的树之间的差异。这种方法得到的树和朴素算法（效率很低）建立的树相同。这种方法中使用了很多巧妙的数据结构用以记录树节点的统计量。Gehrke 等人（1999）报告，对于具有 1 千万个数据向量的 9 维合成数据，利用这种方法把分类树拟合到这些数据的时间大约是 200 秒。

一种相关的策略是导出新的近似算法，这些算法凭借各种启发（对数据的线性扫描次数很少）天生具有期望的可伸缩性能。这些算法通常有很好的伸缩性但是不一定和算法原来的“非伸缩”版本保持一致。例如，Bradley，Fayyad and Reina（1998）以及 Zhang，Ramakrishnan and Livny（1997）讨论了具有这种特征的可伸缩聚类算法。

423

12.12.3 考虑磁盘访问的有针对性算法

解决磁盘数据问题的另一种方法是开发与关系数据库和事务数据紧密耦合的新算法。这方面的一个最佳例子是关联规则算法，在第 5 章中我们曾简单提起过这种算法，在下一章中我们将更详细的讨论该算法。关联规则算法的搜索部分利用了事务数据通常都很稀疏的特性（例如，每一笔交易中，大多数顾客仅购买了很少的几种商品）。从顶层来看，这种算法通常采用广度优先搜索策略，对树的每一层扫描一次数据，执行起来比较容易。Agrawal 等人（1996）报告了对包含 1 000 种商品和 1 千万条记录的综合数据进行处理的结果。他们的实验证明了自己的算法在这个数据集上的运行时间是交易数的线性函数。在其他稀疏事务数据集上的结果与此类似，而且已经开发出了基本算法的很多变体（参见第 13 章）。

424

12.12.4 伪数据集和充分统计量

图 12-5 显示了另一种通用的思想，可以把这种思想看作是对随机抽样的推广。产生一个近似的数据集（通常很小），然后让数据挖掘算法访问这个数据集（比如说在主存储器上）而不是处理整个数据（在磁盘上）。当然这种一般方法所得到的仅是对在整个数据上运行算法时所得到结果的近似。然而，如果构建近似数据集的方式足够巧妙，那么很多时候便可能得到几乎相同的结果。实践中的大多数情况是，我们要使用不同的模型、不同的变量等等，要运行数据挖掘算法很多次，最后才停留在最终模型上。对于这样的探索性建模过程，近似数据集特别有价值（探索过程中不必使用整个数据集）。

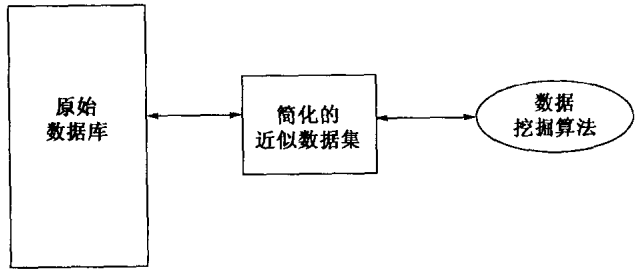


图12-5 让数据挖掘算法操作完整数据集的思想框架

在这一通用框架下，Du Mouchel 等人（1999）提出了一种受统计思想启发的“数据挤压”方法，该方法产生 n' 个“伪”数据点，其中 n' 远远小于原来的数据点数 n，而且这些伪数据点是由算法自动选取的，用来模仿原始大数据集的统计结构。其一般思想就是要尽可能地逼近似然函数的结构，即使在数据挖掘算法中并没有指定所用模型的函数形式。实验表明在逻辑回归问题中，这种方法与对数据集的简单抽样相比明显地降低了预测误差（Du Mouchel et al.（1999））。

对于某些相关主题的数据集，使用比平面文件和关系数据库中的多重表格更高效的数据结构来存储原始数据可能就足够了。Moore and Lee（1998）提出的 AD-树结构提供了一种存储多元范畴型数据（例如计数）的高效机制。数据挖掘算法可以从 AD-树中访问计数和相关的统计量，这比直接访问原始数据要快得多。据报告，这样可以把各种分类算法的计算速度提高 50 到 5 000 倍（和算法的朴素实现相比）（Moore（1999））。

总而言之，可以使用很多不同的技术来实现数据挖掘算法，使其处理大数据集的时间和空间效率都很高。事实上，还有一些算法我们这里没有提到，比如仅观察数据点一次的在线算法（对于数据是高速连续流的应用很有价值）；以及面向硬件的解决方案，比如算法的并 425
行处理实现（对于数据和算法都允许高效并行处理的情况适用）。对特定技术的选择大多数时候都依赖于数据挖掘应用中的实际情况——比如说，数据挖掘算法产生答案的速度必须有多快？模型是否必须不断地更新？等等情况。对可伸缩数据挖掘算法的研究很可能还要继续一段时间，我们期待着这个领域有更大的进展。有必要提醒读者，就像其他所有事情一样，天下没有免费的午餐！换句话来说，模型精度、算法速度和内存等因素通常都是相互制约的。要选择出最适合当前问题的算法和数据结构不仅要仔细地考虑算法方面的问题，还应该考虑算法和模型在实践中的应用细节。

12.13 补充读物

每年都有很多高质量的数据库会议，比如美国计算机学会（ACM）主办的数据管理会议（SIGMOD），以及关于数据库原理和基于知识系统的 SIGACTSIGMOD-SIGART 专题讨论会、超大型数据库会议（VLDB）以及数据工程国际会议（ICDE）。

有几本数据库方面的教材是非常优秀的，其中包括 Ullman（1988），Abiteboul，Hull and Vianu（1995），以及 Ramakrishnan and Gehrke（1999）。Chaudhuri（1998）调查了查询优化的最新成果。Gray et al.（1996）和 Gray et al.（1997）讨论了数据立方体。Chaudhuri and Dayal（1997）很好地介绍了 OLAP。Garcia-Molina et al.（1999）介绍了数据库管理系统的实现。Shoshani（1997）很好地讨论了 OLAP 和统计数据库。Sarawagi et al.（2000）和 Holsheimer et al.（1995）中讨论了使用数据库管理系统来实现数据挖掘算法的问题。

Madigan et al.（出版过程中）讨论了对原始数据挤压方法的各种扩展。Provost and Kolluri（1999）概括了实现可伸缩数据挖掘算法的不同技术，以处理非常庞大的数据集。Provost，Jensen and Oates（1999）以及 Domingos and Hulten（2000）介绍了在数据挖掘中对超大数据集进行采样的一些实例。

第 13 章　寻找模式和规则

13.1　简介

本章将讨论从庞大数据集中寻找有用模式和规则的问题。我们在前面的章节中曾经指出：模式是一种局部概念，它反映了数据某一方面的信息，而模型则是对数据的全面描述。

对于一个描述超市顾客的数据集合来说，模式可能是“十分之一的顾客购买了酒和干酪”；对于一个通信警报数据集来说，模式可能是“如果在 30 秒内相继发生了警报 A 和警报 B，那么有 50%的概率在 60 秒内会发生警报 C”；对于第 1 章中讨论的网络日志数据集，模式的一个例子是“如果一个人访问了美国有线新闻网（CNN），那么他有 60%的可能性会在同一个月内访问美国广播公司（ABC）新闻网站”。在这些例子中，每个模式都是关于部分数据的有趣信息片段。

如何从数据中发现这样的模式呢？如果给定了表示模式的某种方式以及这种表示方式下的所有可能模式，那么最原始的方法就是依次的试验每一种模式，并观察它是否在数据中发生，以及/或者从某个意义上来说它是否显著。如果可能模式的数量很小，那么这种方法或许还是可以接受的，但是通常这种方法是根本不可行的。举例来说，在超市的例子中，如果我们为所有商品的每个子集定义一种模式，那么对于 1 000 件商品来说就有 2^{1000} 个模式。对于图像或报警序列的情况，潜在的模式数量是无限的。

如果各个模式彼此间是毫无关系的，那么我们别无选择，只好使用原始方法。但是通常模式集合中都存在大量的结构，我们应该使用这些模式结构来引导搜索。通常，在各个模式之间都存在泛化/特化关系：如果只要模式β出现在数据中，模式α也一定出现在数据中，那么模式α就是模式β的泛化（更一般○）。例如，模式“至少 10%（译注：此处该为 5%，或者将下面的 5%改为 10%）的顾客购买了酒”是模式“至少 5%的顾客购买了酒和干酪”的泛化。使用这种模式间的泛化关系可以得到一种简单的算法来寻找出现在数据中的所有特定类型的模式。

在本章中，我们给出了很多从庞大的各类数据集中寻找局部模式的方法。我们从非常简单的模式类型和相当直截了当的算法开始，然后讨论一些推广的方法。本章方法的基本思路是通过对更一般模式的提炼来发现使人感兴趣的模式。

模式和规则算法的可伸缩性显然是一个很重要的问题。本章中介绍的方法通常只对数据集进行有限次的扫描，所以这些方法可以很好地适应庞大的数据集。此外，如果我们所感兴趣的仅是适用于数据集中绝大多数数据的模式或规则，那么我们可以利用采样来提高效率。模式在样本中的频率和在整个数据集中的频率会大致相同。所以从理论上来说，从样本中寻找模式同样可以产生很好的效果。如果我们所感兴趣的是仅在数据集中很少出现的模式，例如要在夜空上亿个天体中寻找非常稀少的星体或者星系，那么使

○ 译注：本章中我们将 generalization 译为“泛化”，more general 译为“更一般”，二者的含义是一致的。

用样本是不够的。

13.2　规则表示

规则（rule）是由左侧的命题（前提或者条件）和右侧的结论组成的，例如，“如果下雨，那么地上会湿”。左侧和右侧都是由对世界的一种布尔描述（真或假）组成的。规则的含义是如果左侧为真，那么右侧也为真。概率规则（probabilistic rule）把这个定义修改为：如果左侧为真，那么右侧为真的概率是 p——概率 p 就是给定左侧为真后右侧为真的条件概率。

规则作为认知建模和人工智能中的一种知识表示方式有着悠久的历史。它具有易于解释
428 的优点（至少对于较小的规则集来说是这样的），而且关于机器学习的研究已经发现，规则是从数据中学习可解释知识的一种有用模式。事实上，可以把学习分类树（在第6章和第10章中讨论过）看作是学习规则集的一个特例：可以把从根到每个叶子节点的条件看作是命题的合取，组成规则的左侧；并把叶子节点的类标签看作是规则的右侧。

应该注意到，规则具有固有的离散性，也就是说，规则左侧和右侧都是布尔陈述，因此规则特别适于对离散型和范畴性变量建模。因为可以直接用布尔项作出关于这些变量的陈述。当然我们可以把这个框架扩展到取实数值的变量，方法是把这些变量量子化成取离散值的量子（quanta），例如，“如果 $X > 10.2$，那么 $Y < 1$”（这就是分类树处理实数值变量的方法）。

通常，规则的左侧被表示为简单的布尔函数（例如合取），函数的参数是对各个变量取值情况的陈述（例如：$A = a_1$ 或者 $Y > 0$）。合取的简洁性（相对其他任意布尔函数而言）使合取规则成为迄今为止在数据挖掘中应用最广泛的规则表示形式。对于实数值变量，像 $X > 1 \wedge Y > 2$ 这样的规则左侧定义了一个左侧区域，区域的边界平行于变量空间（X, Y）的坐标轴，也就是一个多维“箱”或者超矩形。当然我们可以进行推广，使语句中可以包含变量的任意函数（导致左侧区域更加复杂），但这会失去简单形式所具有的可解释性。因此，为了处理规则学习中的实数值变量，实践中流行的是使用简单的一元阈值，因为这样既简单又易于解释。

13.3　频繁项集和关联规则

13.3.1　简介

关联规则（在第5章和第12章中曾简要介绍过）为数据挖掘中的规则模式提供了一种非常简单而又有价值的描述形式。再次考虑图13-1中的0/1示例数据（一个“指示（indicator）矩阵”）。图中的行代表关于某个客户的交易（即被一起购买的一“购物篮”商品），列代表商店中的商品。（i, j）位置上的“1”表示客户 i 购买了商品 j，“0”表示这个客户没有购买这种商品。

[htb]

basket-id	A	B	C	D	E		
t_1	1	0	0	0	0		
t_2	1	1	1	1	0		
t_3	1	0	1	0	1		
t_4	0	0	1	0	0		
t_5	0	1	1	1	0		
t_6	1	1	1	0	0		
t_7	1	0	1	1	0		
t_8	0	1	1	0	1		
t_9	1	0	0	1	0		
t_{10}	0	1	1	0	1		

图13-1 人为编制的购物篮数据例子

我们所感兴趣的是从这个数据集中发现规则。对于从变量 $A_1,\cdots,A_p$ 观察到的 0，1 集合， 429
关联规则具有如下形式：

$$\left((A_{i_1}=1)\wedge\cdots\wedge(A_{i_k}=1)\right)\Rightarrow A_{i_{k+1}}=1$$

其中对于所有的 j，$1\leqslant i_j\leqslant p$。可以把这样的关联规则进一步简化为 $\left(A_{i_1}\wedge\cdots\wedge A_{i_k}\right)\Rightarrow A_{i_{k+1}}$。像 $(A_{i_1}=1)\wedge\cdots\wedge(A_{i_k}=1)$ 这样的模式被称为项集（itemset）。于是可以把关联规则看作形式为 $\theta\Rightarrow\varphi$ 的规则。其中 θ 是一个项集模式，φ 是仅包含一个合取项的项集。我们也可以在规则的右侧包含合取式，但是为了简洁性我们不这么做。

关联规则框架最初是为很大的稀疏事务数据集开发的。这个概念可以直接被推广到取有限个数量值的非二值变量的情况，不过我们在这里不这样做（为了表示的简洁）。

如果给定了项集模式 θ，那么它的频率 $fr(\theta)$ 就是数据中满足 θ 的实例比例[⊖] 注意有时把频率 $fr(\theta\wedge\varphi)$ 称为支持度。如果给定关联规则 $\theta\Rightarrow\varphi$，那么它的精度 $c(\theta\Rightarrow\varphi)$（有时被称为可信度） 430
就是满足 θ 的行中又满足 φ 的行的比例，也就是：

$$c(\theta\Rightarrow\varphi)=\frac{fr(\theta\wedge\varphi)}{fr(\theta)} \tag{13.1}$$

按照条件概率的表示，可以把关联规则的试验精度看作是给定 θ 为真的条件下，φ 为真的条件概率的极大似然（基于频率的）估计。注意，对于很小的样本，我们可以使用后验估计（参见第 4 章）的最大值来得到这一条件概率的更佳估计，而不用这种简单的基于频率估计。然而，因为关联规则应用通常都具有非常庞大的数据集，而且项集大小的阈值很大，在这样的情况下，简单的极大似然估计就足够了。

频繁项集是非常简单的模式，它们可以告诉我们数据集中经常一起发生的变量。仅知道频繁项集，并没有得到数据的大量信息：它仅提供了一个很窄的窗口让我们观察数据的某一方面。类似地，一个关联规则则仅告诉我们一个条件概率，并没有告诉我们控制变量的联合概率分布的其余信息。

寻找频繁项集模式（或者说频繁集）并不难：如果给定了一个频率阈值 s，那么就可以找到所有频繁的项集模式，并得到它们的频率。在图 13-1 的例子中，如果把频率阈值设为 0.4，那么频繁集就是 $\{A\}$、$\{B\}$、$\{C\}$、$\{D\}$、$\{AC\}$ 和 $\{BC\}$。由此可以发现规则 $A\Rightarrow C$ 和 $B\Rightarrow C$，

⊖（译注：原书中“比例”为“数量”，疑为误）。

它们的精度分别为 4/6 = 2/3 、5/5 = 1。

寻找关联规则的算法寻找满足频率和精度阈值的所有规则。如果频率阈值太低，那么可能会有很多频繁项集，从而有很多规则。因此，寻找关联规则仅是数据挖掘工作的开始：这些规则中的某些对用户不足一提，而有些是非常有趣的。利用关联规则进行数据挖掘的一个主要难题就是如何从发现的大量规则中选择出特别有趣的规则。

规则的频率告诉我们规则适用的频繁程度。在很多情况下，很低频率的规则是没什么意义的，而且这一假定事实上已经融入了关联规则发现问题的定义之中。关联规则的精度
431 不一定总能指示出它的有趣度。举例来说，在医疗应用中，由怀孕推出这个患者是女性这条规则的精度为 1，但它并没有意义。精度接近 1 的规则可能是有趣的，但是精度接近 0 的规则也可能是有趣的。稍后我们会回到这个话题，即如何衡量规则是对用户有趣的。（在第 2 章中我们讨论了数据质量的问题。对于非常庞大的数据集合，我们很可能发现由怀孕推出这个患者是女性这一规则的精度小于 1。但这并不意味着存在怀孕的男士，而是由于数据不正确所导致的。）

可以利用标准的统计显著性检验技术来评估关联规则 $A \Rightarrow B$ 的统计显著性。也就是分析估计概率 $p(B = 1 \mid A = 1)$是否有别于估计概率 $p(B = 1)$，以及这种差异是否是偶尔发生的。这等价于检验 $p(B = 1 \mid A = 1)$和 $p(B = 1 \mid A = 0)$的差异（参见例 4.14）。

尽管这种检验是可能的，但是利用显著性检验来评估关联规则的质量是有问题的，原因是第 4 章中所讨论的多重检验问题。如果我们从数据中提取出很多个规则，而且对每个进行显著性检验，那么很有可能会（仅是由于偶然性）发现表现出统计显著性的规则（即使数据纯粹是随机的）。

关联规则集合不能给出可以用来系统推理的单一整体模型。例如，规则没有提供预测未知表项的直接方式。对于一个变量，不同的规则可能预测出不同的值，而且根本不存在任何核心结构（就像决策树那样）来决定哪个规则是有效的。

为了说明这一点，假定我们又得到了图 13-1 的一行：$A = 1$，$B = 1$，$D = 1$，$E = 1$；那么可以使用从这些数据得到的规则集推论出（a）$C = 1$ 的精度是 2/3（根据规则 $A \Rightarrow C$）；（b）$C = 1$ 的精度为 1（根据规则 $B \Rightarrow C$）。因此，这个规则集并没有形成一种对数据集合的全局一致描述。（不过，可以认为关联规则或者频繁集的汇集为原始数据集提供了一种有价值的压缩表示，因为可以从这个集合中检索出非常多的有关数据的边际信息。）

根据第 6 章的讨论，寻找关联规则的模型结构是所有可能的合取概率规则。而且可以认
432 为评分函数是二值的：具有足够精度和频率的规则的分数为 1，所有其他规则的分数为 0（仅探索分数为 1 的规则）。在下一小节中，我们将讨论寻找所有频繁集和关联规则的搜索方法（对于预先定义的频率和精度阈值）。

13.3.2 寻找频繁集和关联规则

在这一小节中我们讨论从很大的 0/1 矩阵中寻找关联规则的方法。对于购物篮和文本文档这样的应用，典型的输入数据可能具有 10^5 到 10^8 个数据行，10^2 到 10^6 个变量。这样的矩阵经常是非常稀疏的：任意给定行中 1 的数量是非常少的，比如说，矩阵中任一给定元素为 1 的机会是 0.1%或更小。

发现关联规则的任务就是要找出满足预先指定的频率和精度标准的所有规则。这个任务看起来似乎是令人望而生畏的，因为潜在的频繁集数量是与变量数和数据数呈指数关系的。

举例来说，对于购物篮这样的应用，这个数字是相当庞大的。幸运的是，在实际数据集中，通常情况下频繁集数量是比较小的（比如说，大多数顾客仅购买了全部商品的一个很小子集）。

如果数据集很大，那么主存贮器将无法容纳这些数据。因此理想方法读取数据的次数应该尽可能地少。寻找关联规则的算法通常把这个问题分成两部分：首先寻找频繁集，然后再用这些频繁集来组成规则。

如果知道了频繁集，那么寻找关联规则是很简单的。如果规则 $X \Rightarrow B$ 的频率至少为 s，那么根据定义集合 X 的频率至少为 s。因此，如果知道了所有的频繁集，那么我们就可以产生所有 $X \Rightarrow B$ 形式的规则，并通过对数据的一次扫描计算出每一个规则的精度。

寻找频繁集的原始方法是计算所有子集的频率，但是显然这太慢了。关键的一点是变量集合 X 频繁的必要条件是 X 的所有子集是频繁的。这意味着我们不必计算具有非频繁子集的集合 X 的频率。所以，我们可以这样寻找所有频繁集：首先找出所有由一个变量组成的频繁集；在知道了这样的频繁集后，我们再建立包含两个变量的候选集合：即{A, B}，其中{A}和{B}都是频繁的；在建立了容量为 2 的候选集合后，我们就可以通过观察数据来找出真正的频繁集。这样便得到了容量为 2 的频繁集。依此类推，我们可以得到容量为 3 的候选集合，然后根据数据计算的它的频率，等等。可以把这种方法归纳为： 433

```
i = 0;
C_i = {{A} | A是一个变量};
while C_i不为空 do
    扫描数据库:
        对于 C_i中的每一个集合，验证它是否是频繁的;
        令 L_i为 C_i中频繁集的汇集;
    组成候选集:
        令 C_{i+1}为容量为 i + 1 的那些集合，它们的所有子集都是频繁的;
end
```

这种方法被称为 APriori 算法。还有两个问题需要解决：如何组成候选集合？以及如何计算每个候选集的频率？第一个问题是很容易用一种令人满意的方式解决的。假定我们有一个频繁集的汇集 L_i，并且想要找出所有容量为 $i+1$ 的可能频繁集 Y；也就是子集都为频繁集的所有 Y 集合。可以这样来实现这个目标：先从 L_i 中找出所有{U, V}对，使 U 和 V 的联合容量为 $i+1$，然后再验证这个联合是否真的为潜在候选集。在 L_i 中有少于$|L_i|^2$个的集合对，对于其中的每一对我们必须检查$|L_i|$个其他集合是否在其中。在最坏的情况下，复杂度为 L_i 容量的立方。在实践中，这种方法的实际运行时间相对于 L_i 容量是线性的，因为在 L_i 中经常仅存在很少的重叠元素。注意，候选集形成是独立于实际数据记录数 n 的。

对于给定的候选集合 C_i，可以通过对数据库的一次扫描计算出它们的频率。只要保存每个候选项的计数，当遇到包含这个候选项的记录时便增加其计数。如果检验是以普通方法实现的，那么所需的时间是 $O(|C_i|np)$，可以使用其他数据结构技术来提高这种方法的速度。

寻找频繁集所需的总时间是 $O(\sum_i |C_i|np)$——也就是和数据容量（np）与所有层次候选集合数的乘积成正比。这个算法需要扫描数据库 k 或 $k+1$ 次，其中 k 是最大频繁集的元

434 素个数。

以上的基本关联规则算法有很多种变体。这些变体通常针对如下三个目标中的一或多个：最小化扫描数据的次数；最小化必须分析的候选集数量；最小化计算每个候选集频率所需的时间。

加速候选集频率计算的一种重要方法是使用数据结构以便更容易地发现数据集中的每一行发生的是 C_i 中的哪个候选集。一种可能的方法是用分支因子为 p（变量数）的树结构来组织候选集合。对于每个变量 A，标号为 A 的树的树根的子女包含了第一个变量为 A（按照变量的某种顺序）的那些候选集。标号为 A 的子女节点是以递归方式建立的。

另一种加速频繁集计算的重要方式是使用采样。因为我们的兴趣所在是要发现描述大多数子群的模式，也就是频率高于给定阈值的模式，所以使用样本来代替整个数据集显然可以很好的近似频繁集以及频繁集的频率。也可以使用样本得到一种大多数情况下仅需要扫描数据两次的方法。第一次是根据样本求出频繁集 F 的汇集，使用的阈值略低于用户给定的阈值。然后根据整个数据集计算 F 中每个集合的频率，这样便可以得到整个数据集上的准确频繁集，但条件是不存在这样的变量集合 Y：在样本中它是不频繁的，但是它的所有子集在整个数据集中却是频繁的；这种情况下，我们必须再额外扫描一次数据库。

13.4 推广

也可以把寻找频繁出现变量集合的方法应用到其他类型的模式和数据，因为上面描述的算法没有使用频繁集模式的任何特殊属性。我们所使用的就是（1）频繁集的联合结构以及单调特征，以便可以快速地组成候选模式；（2）快速验证一个模式是否出现在一行中的能力，以便可以通过对数据的快速扫描计算出模式的频率。

435 下面我们以更抽象的方式来表示这个算法。假定我们有一类原子模式 A，我们的目标是要找出这些原子模式中经常出现的模式的合取。也就是说，模式类 P 是以下所有模式合取得到的集合：

$$\alpha_1 \wedge \cdots \wedge \alpha_k$$

其中对于所有的 i，$\alpha_i \in A$。

令 D 为 n 个对象 $d_1, \cdots, d_n$ 所构成的数据集合，并假定我们可以验证模式α相对对象 d 是否为真。合取$\theta = \alpha_1 \wedge \cdots \wedge \alpha_k \in P$ 相对 d 为真的条件是所有合取项α_i相对 d 为真。令σ为一个阈值。我们的目标就是寻找那些频繁发生模式的合取：

$$\{\theta \in P \mid \text{对于至少}\sigma\text{个对象 } d \in D,\ \theta\text{为真}\}$$

对于频繁集的情况，原子模式就是 $A = 1$ 形式的条件，其中 A 是变量，像 ABC 这样的频繁集就是 $A = 1 \wedge B = 1 \wedge C = 1$ 形式合取的简短表示。

假定我们可以决定每个原子模式在数据中出现的次数。那么我们就可以应用上面的算法来从 P 中找出所有出现足够频繁的模式。我们只要先找出出现足够频繁的所有原子模式，然后建立可能频繁发生的两个原子模式的合取，再验证这些合取中哪些出现的足够频繁，而后再建立容量为 3 的合取，等等，这种方法的工作方式和前面的完全相同。如果模式很复杂，那么我们就必须做某些灵巧的处理，以建立新的候选集并检验模式的出现情况。

13.5 寻找序列中的片段

在这一节中我们讨论寻找关联规则这一通用思想的另一种应用：从序列中寻找片段（episodes）。

如果给定一个事件类型（event types）的集合 E，那么一个事件序列（event sequence）s 就是一系列序偶(e, t)，其中 $e \in E$，t 是一个整数，代表时间 e 发生的时间。一个片段α是由事件类型组成的一段局部状态（partial order），就像图 13-2 中所示的那样。可以把片段表示为图。

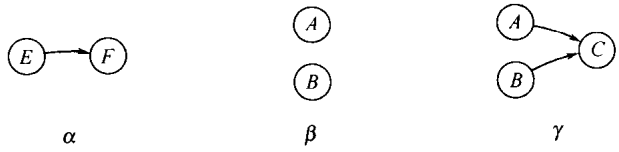


图13-2 片段α、β和γ

如果给定窗口宽度 W，那么片段α在序列 S 中的频率就是包含α中的事件类型并以α所描述的顺序发生的片段占宽度 W 的比例。下面我们集中讨论一个任务：给定事件序列 s，片段集合$\mathcal{E}$，窗口宽度 win，频率阈值 min_fr，目标是寻找出序列 s 上发生比例至少为 min_fr 的所有片段的汇集 $FE(s, win, min_fr)$。下文给出了计算频繁片段汇集的算法。 436

这种方法也是建立在前面的关联规则算法思想上的：从可能的最简单模式开始计算模式的频率。利用上一次扫描数据的信息建立新的候选模式，而且如果一个模式的任一个子模式不够频繁，那么便不再考虑这个模式。和前面列出的算法相比最主要的差异是片段的合取没有那么明显。

如果片段β的所有节点也出现在片段α中，而且β中各节点间的关系也体现在α中，那么就说β是α的子片段。使用图论的术语，我们可以说β是α的导出子图。如果β是α的子片段，那么我们将其写为$\beta \preceq \alpha$；如果$\beta \preceq \alpha$ 并且$\beta \neq \alpha$，那么写为$\beta \prec \alpha$。

例 13.1 如果给定一个事件类型的集合 E，一个 E 上的时间序列 s，一个片段集合$\mathcal{E}$。窗口宽度 win，频率阈值 min_fr，那么可以用下面的算法来寻找频繁片段的汇集 $FE(s, win, min_fr)$。

```
C1 := {α ∈ε| |α| =1};
l :=1;
while C1不为空 do
    /*扫描数据库: */
    F1 := {α ∈ C1 | fr(α, s, win) ≥min_fr};
    l := l + 1;
    /*产生候选项: */
    Cl = {α ∈ ε | |α| = l 而且对于所有满足β≺α 而且|β| < l 的β ∈ ε都有β ∈ F|β|};
End;
For 所有的 l do 输出 Fl;
```

437

这种算法根据子片段的关系对片段进行分层{广度优先}搜索。这一搜索从最一般的片段——也就是仅包含一个事件的片段——开始。在每一层，这个算法首先求出候选片段的汇

集，而后根据事件序列检查它们的频率。

这个算法至多扫描数据 $k+1$ 次，其中 k 是最大频繁片段的边和顶点数。每扫描一次计算$|C_l|$个片段的频率。要计算一个片段的频率需要在序列中找到这个片段出现的窗口。这个操作可以在和序列长度以及片段的容量呈线性关系的时间内完成。因此片段发现算法的运行时间是 $O(n\sum_{l=1}^{k}|C_l|l)$，其中 n 是序列的长度。

可以使用类似方法来寻找任何种类模式的合取，只要存在的频繁模式不是太多。

13.6 选择发现的模式和规则

13.6.1 简介

上一节讨论了用来寻找满足简单频率和精度标准的所有规则的方法。尽管这种方法在很多应用中是很有用的，但是对于一些简单而且重要的模式类来说，我们根本不希望看到其所有的模式。比如说，考虑一个具有连续值变量的数据集，那么正如第 1 章所提到的，我们可能会看到这样的模式：

θ：如果 $X>x_1$，那么 $Y>y_1$ 的概率为 p，这个规则的精度为 q

这个规则很好地描述了数据的一个局部特征。现在的问题是在数据中 X 有 k 个不同的值，Y 有 h 个不同的值，那么存在 kh 个潜在规则，而且其中很多的频率都足以令人感兴趣。例如，从一个包含变量年龄和收入的数据集中我们可能会发现这些规则：

α：如果 年龄 >40，那么收入>62755（概率为 0.34）

438 β：如果 年龄 >41，那么收入>62855（概率为 0.33）

首先，用户不会愿意看到表达的模式大体相同的两条规则。因此，即使我们找到了这样的两条规则，我们也应该避免把它们都显示给用户。第二个问题是在这个例子中，模式α比模式β更具一般性[⊖]，而且满足α_i比α_{i+1}更具一般性的序列α_1，α_2，…很长。因此，上一节中的基本算法思想（从最一般的模式起步，观察数据，以所有可能的方式扩展够条件的模式）在这里不再适用了，因为任一个单一模式都有很多说明而且模式空间很大。

所有这些意味着必须使用频率标准外的其他准则来修剪搜索到的模式。通常使用下面的两个标准来进行修剪：

1. 有趣度（interestingness）：所发现的模式是否有足够的意义而值得输出；
2. 可信度（promise）：所发现的模式是否具有潜在的有趣特例。

注意，模式可能是可信的，即使它不是有趣的。一个简单的例子是对所有数据对象都成立的任何规则：它不是有趣的，但是它的一些特例可能是有趣的。可以按不同的方式利用模式的频率和精度以及背景知识来量化有趣度。

13.6.2 寻找模式的启发式搜索

假定我们已经有了定义模式有趣度和可信度的方法，以及修剪模式的方法。那么可以把寻找有趣模式的通用启发式算法归纳为如下的形式：

⊖ 译注：根据前面的 13.1 节中的定义，说‘α比模式β更具一般性’不甚准确，但这里是从规则β为规则α的一个子集这个意义上来说的。

```
C = {最一般的模式};
while C ≠ Φ do
    E = 所有适当选择的 C 元素特例;
    for q ∈ E do
        如果 q 满足有趣度标准，那么输出 q;
        如果 q 不可信，那么抛弃 q，否则保留 q
    End;
    对 E 的额外修剪;                                    439
End;
C = E;
End;
```

当把这种算法实例化时，我们便得到几种熟悉的方法：

1．假定模式是变量集合，并且把项集的有趣度和可信度都定义为谓词 $fr(X) > \sigma$。不作任何额外的修剪，那么原则上讲这个算法和寻找关联规则的算法是一样的。

2．假定模式是如下形式的规则：

$$\alpha_1 \wedge \cdots \wedge \alpha_k \Rightarrow \beta$$

其中 α_i 和 β 是 $X = c$、$X < c$ 或 $X > c$ 形式的条件，其中 X 为变量，c 为常数。令有趣度标准为某种意义上规则的统计显著性；并令可信度标准永远为真。额外的修剪步骤仅保留 E 中的一条规则——统计显著性最高这条规则。这样我们便得到了一种寻找具有最高统计显著性规则的爬山搜索算法。（当然，不能用一般意义来解释这里的显著性，因为其中包含了很多相互关联的检验。）

3．假定有趣度标准为规则具有统计显著性；而且可信度标准永远为真；额外的修剪步骤保留 E 中显著性最高的 K 条规则。当 $K = 1$ 时是上面的情况；当 K 为任意值时便得到束状搜索（beam search）算法。

13.6.3 有趣度标准

在前一小节中我们提到了衡量规则有趣度的尺度。如果给定一个规则 $\theta \Rightarrow \varphi$，那么有很多种方式来定义它的有趣度。通常，模式 θ 和 φ 中所引用变量的背景知识对规则有趣度有很大的影响。例如，在信用评估数据集中，我们可能预先确定把出生月份和信用等级联系起来的规则不是有趣的（有意义的）。再如，在购物篮数据库中，我们可能说规则的有趣度和规则频率与产品价格的乘积直接成比例；也就是说，我们或许对和昂贵商品相联系的高频率规则更感兴趣。一般来讲，没有一种单一的方法来把背景知识自动地考虑进来，所以规则发现系统必须让用户可以很容易地使用这些针对应用（application-dependent）的有趣度标准。 440

有趣度的纯统计标准更容易使用，因为它是独立于具体应用的。或许这样的最简单标准就是通过建立一个 2×2 的列联表格，θ 和 φ 的出现与否作为变量，然后统计四种不同组合的频率。

	φ	$\neg\varphi$
θ	$fr(\theta \wedge \varphi)$	$fr(\theta \wedge \neg\varphi)$
$\neg\theta$	$fr(\neg\theta \wedge \varphi)$	$fr(\neg\theta \wedge \neg\varphi)$

根据这张表中的数据我们可以计算 θ 和 φ 间不同类型的关联尺度，比如卡方分数。评价规则 $\theta \Rightarrow \varphi$ 有趣度的一种特别有价值的尺度是 J-尺度（J-measure），其定义为：

$$J(\theta \Rightarrow \varphi) = p(\theta)\left(p(\varphi \mid \theta)\log\frac{p(\varphi \mid \theta)}{p(\varphi)} + (1 - p(\varphi \mid \theta))\log\frac{1 - p(\varphi \mid \theta)}{1 - p(\varphi)} \right) \tag{13.2}$$

这里 $p(\varphi|\theta)$是试验观察到的规则可信度（精度），$p(\theta)$和 $p(\varphi)$分别是试验观察到的θ和φ的边际概率。可以把这个尺度看作是有和没有事件θ 这一条件时φ 所定义的二值变量间的交熵，因数 $p(\theta)$指出了这个规则所适用的广度。其他因数衡量了以下两种情况下我们关于φ的知识有多大差异：仅知道边际概率 $p(\varphi)$；和知道条件概率 $p(\varphi|\theta)$。J-尺度具有相对特化表现很好的优点，也就是说，有可能证明给定规则的特化的 J-尺度值的边界。

实践中已经发现不同有趣度尺度所得到的模式大致是相同的，只要评估函数符合一些基本特征（比如精度保持不变时，分数随模式频率的上升单调上升）。第 7 章也讨论了与模式
441 有趣度相关的一般问题。

13.7　从局部模式到全局模型

如果给定了出现在数据中的一系列模式，那么有没有办法利用这些模式来组成全局模型呢？这一节我们简要讨论两种针对这一目标的方法。第一种方法组成一个决策列表或者规则集合来完成分类任务；第二种方法使用模式的频率构建一个近似的概率分布。

简单起见，令 B 为一个二值变量，并假定我们已经发现了一系列形式为$\theta_i \Rightarrow B = 1$ 和$\eta_i \Rightarrow B = 0$ 的规则。那么我们该如何组成一个决策列表以找出或者说预测出 B 的值呢？（变量 B 的决策列表是形式为$\theta_i \Rightarrow B = b_i$的规则的有序列表，其中$\theta_i$是一个模式，$b_i$是 B 的一个可能值。）可以把这样的决策列表的精度定义为这个列表正确预测出的行的比例。可以考虑规则的所有可能排列，并检查每种情况所产生的最优解，这样至少在理论上可以构建出最优的决策列表。然而这样做所需的时间是规则数的指数级，一种比较好的近似方法是把这个问题看作是一个针对任务的加权集合，然后使用“贪婪”搜索算法。

上面是使用局部模式来获取整个数据集信息的一种方法，下面介绍另一种。如果我们知道了对于 $i = 1, \cdots, k$，模式θ_i的频率为 $fr(\theta_i)$，那么我们对所有变量 $A_1, \cdots, A_p$ 的联合分布了解多少了呢？原则上讲，观察结果 $fr(\theta_i)$可能是任何满足模式频率的分布 I 所产生的。然而，要采用的合理模型不应该对分布的一般特征作出任何更进一步的假定（因为不知道任何进一步的信息）。这就要求这个分布使熵最大化，并符合已经观察到的频率模式。利用迭代比例（proportional）算法可以高效地构建这样的分布，简单来说，算法是按如下方式运转的。从针对变量 A_j 的一个随机分布 $p(x)$开始，然后施加每个模式θ_i 的频率约束。这个过程是这样实现的：先对θ_i 为真状态时的 p 求和，然后对这些概率进行缩放使得到的 p 的更新版本使θ_i 能满足 $fr(\theta_i)$尺度集合。依次对每个模式进行这种更新，直到观察到的模式频率和由 p 给出的一致。这种方法可以在相当广泛的条件下收敛，因此它的应用很广泛，
442 例如统计文本建模（statistical text modeling）。这种方法的不足是（至少在直接应用的情况下）它需要构建联合分布的每个状态，使得使用该方法的空间和时间复杂度都随变量数的指数变化。

13.8 预测规则归纳

到目前为止本章主要集中在关联规则和相似规则形式上。但本章最前面是以规则的一般定义开始的，现在我们返回到这个框架。回忆一下，我们可以把分类树的每个分支解释为一条规则：从根到叶子的路径上的内部节点定义了规则左侧的合取项；赋给每个叶子的分类标签定义了规则右侧。对于分类问题来说，规则右侧是 $C = c_k$ 的形式，也就是预测分类变量 C 等于某一特定值 c_k。

从而，我们可以认为分类树是由一个规则集合构成的。这个集合具有一些非常特别的属性——也就是说，它形成了一种对输入变量空间的互斥（不重叠）而且完全的划分。根据这种方式，任何一个观察 **x** 会被且仅被一个规则（也就是定义这个点所在区域的那个分支）分类。我们说这个规则集以这种方式“覆盖”了输入空间。

我们发现有必要考虑比树结构更一般的规则集，因为（举例来说）树结构在表示析取布尔函数时的效率特别低。比如说，考虑以下析取映射：$(A = 1 \wedge B = 1) \vee (D = 1 \wedge E = 1) \Rightarrow C = 1$（并且否则 $C = 0$）。我们可以使用两个规则：$(A = 1 \wedge B = 1) \Rightarrow C = 1$ 和 $(D = 1 \wedge E = 1) \Rightarrow C = 1$ 非常高效地表示出这个映射。但是如果用树表示同一个映射就必须为每个分支（比如 A）引入一个专门的根节点变量，即使这个变量仅和该映射的某部分有关。

一种产生规则集的技术是先建立分类树（使用第 10 章介绍的任一种技术），然后把每个分支看作是一条单独的候选规则。规则归纳算法依次访问每一条这样的规则，判断每个规则左侧的条件是否影响该规则对于它所“覆盖”数据的精度。举例来说，我们可以从规则的左侧删除一个条件，然后评定规则的精度（等价于估计出的条件概率）是否提高了（或者实际上没有显著的变化）。如果提高了或没有显示出任何变化，那么可以认为这个条件是不必要的，可以将其删除以得到一个更简单而且可能更准确的规则。重复这个过程，直到所有规则 443
中的所有条件都被分析过了。实践中经常发现这种方法可以消除很大一部分初始的规则条件，这些条件是在增长树的过程中由于它们对改善模型的平均贡献而被引入的，但是对某个特定分支来说是不必要的。

然后便可以使用按这种方式产生的最终规则集合来完成分类任务。因为原始的规则集是以一种不重叠方式“镶嵌”在输入空间上的，而且我们已经删除了定义这些不重叠区域的一些条件，所以区域的边界已经被扩大了（规则已经被泛化了），因此这些区域现在有可能重叠了。这样就可能有两条如下形式的规则：$A = 1 \Rightarrow C = 1$ 和 $B = 1 \Rightarrow C = 1$。那么一个很自然的问题是：我们该如何使用这两条规则来分类一个新的观察向量 **x**，**x** 中 A 和 B 都等于 1？一种方法是把这两个规则看作是对整个联合分布 $p(A, B, C)$的约束，然后利用本章 13.7 节中的熵最大化方法推出对 $p(C = 1 \mid A = 1, B = 1)$的估计。然而由于熵最大化方法在计算方面多少有些复杂，所以实践中往往是使用一些更简单的技术。比如，我们可以找出给定观察向量 **x** 所引发的所有规则（也就是条件被 **x** 满足的规则）。如果找到的规则多于 1，那么只要挑出条件概率最大的一个就可以了。如果根本没有引发任何规则，那么就选择最可能的验前分类值。也可以使用其他更复杂的模式，比如把规则组织为有序的决策列表，或者在多个规则间进行“投票”或平均。

读者可能会问：为什么要从分类树开始然后再产生规则，而不是直接搜索规则呢？分类

树的一个优点是它可以在建立树的阶段，自动的以一种相当简单的而且计算效率很高的方式把任何实数值变量量子化（尽管这些量子对于最终规则集的所有环境来说不一定都是最优的）。另一个优点是实现技术简单；有很多高效的技术可以用来产生树（正如第 10 章中所讨论的，不论是对于数据位于主存储器的情况，还是数据位于次存储器的情况），而且，加入规则选择组件作为“后处理”步骤是相当简单直接的。

不过，根据树来产生规则存在偏向（bias），因而，在机器学习和数据挖掘中也已经有很
444 多算法直接搜索规则，特别是对于离散值数据的情况。当然，应该再一次指出，可能的合取规则数量是相当庞大的，对于每个变量取 m 个值的 p 个变量来说是 $O(m^p)$。因此，在搜索这样的最优规则集合时（或者甚至是仅搜索最佳的唯一规则），我们通常要求助于某种形式的启发式搜索方法（就像在 13.6 节寻找有趣规则集合中所指出的那样做）。

这里要指出，在分类的情况下，应该把“最优”定义为规则集对于新数据的平均精度最高（或者，当涉及分类成本时使平均损失最低）。就像分类树的情况一样，相对于训练数据的分类精度不必是最优的。举例来说，我们可以为每个训练实例定义一个包含这个实例中出现的所有变量的特殊规则。这样的特殊规则对于训练数据的精度很高（如果包含相同变量值的所有实例都属于相同的分类，那么精度甚至可以达到 1），但是泛化精度会很低，因为它过于特殊。因此，实践中经常使用的评分函数并非单单考察精度，特别是用于选择向现有规则集中加入的下一个规则的评分函数，而是（比如）在规则的覆盖面（左侧表达式的概率）和规则的精度间做某种折中，就像前面介绍的 J-尺度那样。

已经定义了合适的评分函数，那么下一个问题就是如何搜索规则集并在训练数据上优化这个评分函数。许多规则归纳算法使用的是种“一般到特殊”形式的启发，和前面描述的搜索有趣规则的一般形式形同，所不同的就是现在我们把有趣度函数替换为一种和分类有关的函数。这些算法从包含尽可能最一般的规则（比如规则的左侧为空）集合开始，然后通过不断探索目前集合中规则的更特殊版本，以一种贪婪的方式向这个集合中加入新的规则。可以把这个过程看作是对所有子集空间的一种系统搜索，从空集合开始，并使用一种每次仅加入一个条件的算子。大量的搜索技术都可以在这里使用，包括第 8 章中讨论的所有系统启发搜索技术（比如束状搜索）。也可以使用相反的启发策略，即从最特殊的集合开始然后进行泛化，不过从计算的角度来看这样做往往会更复杂一些，因为从什么样的规则集开始不如前面那样明显。对于实数值数据，我们可以把每个实数值变量预先量子化为多个柱位（bins）（比如对每个变量使用聚类算法），也可以在搜索规则的同时进行量子化。后一种方法所需的运
445 算特别高而且不太容易实现，按这种方式运转的一种有趣算法是 PRIM 算法（Friedman and Fisher，（1999）），该算法从每个变量的完整数据域开始逐步“收缩”规则区域。

当然，在这两种技术（一种搜索的规则空间更多，因而对计算和内存的需求更高；另一种更简单，仅搜索空间的一个较小部分）间存在着折中。实践中，就像分类树中那样的使用简单算子的贪婪搜索技术很多时候都表现的几乎和复杂方法一样好，因而非常流行。和分类树的情况一样，也存在何时停止向规则集中加入规则的问题（决定模型应该复杂到什么程度的常见问题——这里可以把规则集解释为数据“模型”）。在估计规则集的真实预测精度时交叉验证的技术非常有用，但是运算量也可能非常大，尤其是当需要在规则的不同搜索阶段进行重复验证时。

下面介绍几种对基本分类模式的著名扩展，以总结我们对预测规则的讨论。第一种扩展是这样的，就像把分类树的思想扩展到回归树一样，我们也可以进行一种基于规则的回归。

规则的左侧条件定义了输入空间中一个特定区域，对于这个给定区域，我们可以估计出该区域数据的局部回归模型（可能非常简单，乃至是（比如说）一个最佳拟合的常量）。如果规则是不重叠的，那么我们便得到一个分片的局部回归曲面；如果规则是重叠的，那么我们必须决定如何组合不同规则来对这个重叠区域作出预测。基于规则的回归框架有一个特别的优点——易于解释，尤其是对于高维的问题，因为大多数情况下仅有一小部分变量包含在这个规则中。

对基本规则归纳模式的第二种值得注意的扩展是使用关系逻辑作为规则的基础。对这个话题的深入讨论超出了本书的范围，其根本思想就是把命题逻辑陈述（“变量=值”）的概念推广到所谓的一阶关系逻辑陈述，比如“Parent(X, Y) ∧ Male(X) ⇒ Father(X, Y)”。原则上讲，这种类型规则的学习是相当强大的，因为它允许使用丰富得多的表示语言来描述数据。关系陈述的命题版本通常非常笨拙（而且可能非常大），因为在（更简单的）命题框架下不存在对象间关系的概念。当然关系逻辑表示的特殊表示能力是有代价的，这一点既体现在使用这 446
种规则进行推理的时候，也体现在从数据中学习这些规则的时候。目前已经开发出了学习关系规则的算法（对应的主题叫“归纳逻辑编程（inductive logic programming）”），得到的一些规则是很有效的，不过主要是基于逻辑而不是基于概率来表示数据。

13.9 补充读物

Agrawal et al.（1993）介绍了关联规则问题。Apriori 算法应主要归功于 Agrawal and Srikant（1994），Mannila et al.（1994）以及 Agrawal et al.（1996）。关于寻找关联规则的不同算法的文献非常多，比如 Agrawal, Aggarwal and Prasad（即将出版），Brin et al.（1998），Fukuda et al.（1996），Han and Fu（1995），Holsheimer et al.（1995），Savasere et al.（1995），Srikant and Agrawal（1995，1996），Toivonen（1996）以及 Webb（2000）。Klemettinen et al.（1994）以及 Silberschatz and Tuzhilin（1996）中谈到了关联规则的后期处理。Mannila（1996），Mannila（1997），Meo et al.（1996），Imielinski et al.（1999），Imielinski and Virmani（1999）以及 Sarawagi et al.（1998）中讨论了如何把关联规则发现集成到数据库系统中的问题。Mannila et al.（1997）介绍了在序列中发现片段的算法。

可以说关于关联规则发现算法的论文要远远多于关于关联规则应用的论文，从这一点来看，目前我们还不是很清楚关联规则除了探索性数据分析外还有哪些主要应用。不过关联规则在零售业中有一个有趣的应用——交叉销售，参见 Brijs et al.（2000）以及 Lawrence et al.（2001）。

Smyth and Goodman（1992）中讨论了规则的有趣度，其中也介绍了 J 尺度。Silberschatz and Tuzhilin（1996）也讨论了 J 尺度。

在机器学习的文献中已经提出了大量不同的归纳规则算法，这些算法间的差异主要是如何进行搜索的细节。C4.5 规则算法是根据分类树推导规则的最著名方法（Quinlan （1987，1993））。CN2 算法（Clark and Niblett（1989））使用一种基于熵的尺度通过束状搜索方式来选择规则。其他更新的规则归纳算法（具有精确分类庞大数据集的能力）包括 RL（Clearwater 447
and Stern（1991））算法，强制（Brute）算法（Segal and Etzioni（1994））以及撕裂器（Ripper）算法（Cohen（1995））——设计者们似乎有喜欢使用这些晦涩名称的嗜好！RISE 算法（Domingos（1996））是利用特殊到一般启发的规则归纳算法的一个有趣例子。Holte（1993）介绍了一

项有趣的研究，非常简单的分类规则模型提供了和更复杂的著名分类器大体一样的性能。Aronis and Provost（1997）介绍了如何实现用于海量数据集的高效规则归纳算法的一些实践技巧。

Friedman and Fisher（1999）介绍了高维数据中的“bump-hunting”算法框架，该框架在很多方面是与众不同的：它使用了一种“耐心的（patient）”搜索策略而不是普遍使用的纯粹贪婪搜索策略；它使用的是通用的函数近似框架，既允许实数值的又允许分类值的目标变量；它是从统计角度出发的。Weiss and Indurkhya（1993）中讨论了基于规则回归，RuleQuest（2000）的商业软件包（被称为 Cubist）也包含这个内容。

Quinlan 的 FOIL 算法（1990）是最早的关系规则归纳算法之一。一些教材归纳了有关关系规则学习（又被称为归纳逻辑编程）的最新成果，例如 Lavrac and Dzeroski（1994）以及 Muggleton（1995）。

448

第 14 章　根据内容检索

14.1　简介

在数据库框架下，传统的查询概念被定义为：查询是一种返回精确匹配指定要求的记录集合（或表项集合）的操作。举例来说，在一个人员信息数据库中查询“[level = MANAGER] AND [age < 30]”返回的结果是具有重要职务的年轻雇员列表。正如第 12 章所讨论的，传统数据库管理系统的设计目标之一就是高效地回答这种精确查询。

然而，在很多情况下，尤其是数据分析中，我们所感兴趣的是更一般的但不很精确的查询。考虑一个医疗方面的例子，假定我们知道了一个患者的人口统计学信息（比如年龄性别等等）、血液和其他常规检查的结果，以及生物医学方面的时间序列和 X-光图像。为了辅助对这个患者进行诊断，医生可能希望知道在这个医院的数据库中是否包含类似的患者，如果有类似的患者，那么他们的诊断、治疗方法和最终结果如何？这个问题的难点在于如何根据不同的数据类型（在这个例子中有多元变量、时间序列和图像数据）来判断各个患者间的相似性。在这里，直接使用精确匹配的概念是行不通的，因为几乎不可能找到和这个患者的各项指标均完全匹配的其他患者。

本章将讨论具有这种特征的问题，特别是要在数据集中执行如下形式的查询而必须解决的各种技术问题：

在数据库中找出和指定查询或指定对象最相似的 k 个对象。

下面是这种查询的一些例子：

- 对道琼斯指数的历史记录进行搜索寻找一个特定时间序列模式的出现情况。
- 对地球卫星图像进行搜索，找出可以证明中美洲最近发生了火山喷发的所有图像。
- 搜索互联网，找出评论赫尔辛基市内饭店的在线文档。

可以把这种形式的检索看作是*交互式*的数据挖掘，因为用户直接参与了探索数据集的过程——指定查询并解释匹配过程得到的结果。这与前面各章中讨论的预测和描述形式的数据挖掘形成了对比，在预测和描述建模中人的判断作用往往没有这么重要。

如果数据集是根据内容批注的（比如说，图像数据库已经经过了人工浏览并根据可视的内容作了索引），那么检索问题就简化为标准的数据库索引问题，就像第 12 章中所讨论的那样。然而在本章中，我们要考虑的是实践中更一般的情况——数据库没有被预先索引。我们仅有要寻找目标——也就是*查询模式*（query pattern）Q——的一个实例。根据这个*查询模式* Q，我们要推论出数据集中哪些其他对象和它最相近。这种检索方法被称为*根据内容检索*（retrieval by content），它的最著名应用是在文本中检索。在文本检索中，查询模式 Q 通常是很短的（查询词汇列表），然后在很大的文档集合中匹配这个模式。

本章中我们将主要讨论文本文档检索，因为它应用最广而且是这种思想的最成熟应用。不过我们也将讨论如何把这些方法推广到图像和时间序列检索应用中。可以把这类问题归纳为三个基本组成部分，也就是：

- 如何定义对象间的相似尺度；

- 如何实现高计算效率的搜索算法（对于给定的相似尺度）；
- 如何在检索过程中融入用户的反馈并进行交互。

本章将主要讨论第一和第三个问题。第二个问题通常可以简化为一种索引问题（也就是，
450 在数据库中找出和指定查询最接近的记录），这在第12章中已经讨论过了。

根据内容检索在很大程度上依赖于相似性的概念。在下文的讨论中，我们既使用了“相似”这个词，又使用了“距离”这个词。从检索的角度来看使用其中哪一个没什么大的影响，因为我们既可以使相似尺度最大化，又可以使距离尺度最小化。因此我们隐含假定，大体来说这两个术语是相反的，在实践中使用哪一个都可以。

我们将看到，在各种应用中（文本、图像等等），把测量结果简化为固定长度的标准向量格式是很常见的，因为这样便可以使用标准的几何概念来定义向量间的距离尺度。可以回忆第2章定义的几种距离尺度，比如说，欧氏距离、加权的欧氏距离、曼哈顿距离等等。有必要指出，尽管这些标准距离函数可能很有价值，但是它们主要是一种数学结构，因此和人类对相似性的直观感觉未必一致。在讨论文本和图像这样的数据类型时这一点尤其如此，因为在这些应用中使用建立在独立于具体领域的通用距离函数基础上的算法来模拟人类基于语义内容的检索能力是很困难的。

在14.2节中我们讨论了一个棘手的问题：如何客观地评估特定检索算法的性能。这种评估是非常复杂的，因为对检索性能的最终裁判取决于提出查询的用户的主观想法，用户决定了检索出的数据是否相关（relevant）。

对于结构化的数据（比如序列、图像和文本），要解决根据内容检索还有另一个问题，也就是如何决定用以计算相似尺度的表示（representation）。举例来说，通常用颜色、纹理和相似特征来表示图像；用单词的出现次数来表示文本。这样的抽象表示通常丢失了很多类似局部上下文这样的信息。然而，很多时候这些表示是必须的，因为要在像素级或ASCII字符级（分别对应于图像和文本）定义有一定含义的尺度是很困难的。14.3节中讨论了针对文本数据的根据内容检索问题，集中讨论了向量空间表示。这一节还讨论了在文档中匹配查询的算法、隐含（latent）语义索引以及文档分类。14.4节讨论了相关性反馈这一话题，介绍了用于对个人偏好（preference）（青睐某一对象而不是另一对象）建模的自动推荐系统。14.5
451 节讨论了图像检索算法中的表示和检索问题。建立通用的图像检索算法是一个很困难的问题，因此我们不仅分析了当前方法的长处而且还指出了其中的不足，尤其是恒定性（invariance）问题。14.6节浏览了匹配时间序列（time series）和序列（sequence）的基本概念。可以把检索序列数据看作是图像检索的一维情况，所以也有和图像数据类似的表示和恒定性问题。14.7节对本章内容进行了概括，14.8节给出了一些补充读物。

14.2 检索系统的评价

14.2.1 评价检索性能的困难之处

在分类和回归中，我们总是能以一种客观的方式来评判模型的性能，也就是通过试验来估计模型在未见过的检验数据上的精度（或者更一般的情况是评价模型的失败率）。这使得比较不同的模型和算法很容易。

然而，对于根据内容检索来说，评价一个特定算法或技术的性能要复杂和棘手的多。主

要的难点是检索系统的最终性能尺度是由检索出的信息对用户的实用性来决定的。因此，在现实环境下，对检索性能的评价存在固有的主观性（与分类和回归的情况形成对比）。检索是一种以人为中心的交互过程，这给评价检索性能带来了很大困难，牢记这一点对理解下文的内容是很重要的。

尽管直接评价特定检索系统对大多数用户的实用性是非常困难的，但是如果我们愿意作出某些简化，那么还是存在一些相对客观的方法可以使用。首先我们假定（为了检验目的），相对一个特定的查询，可以把对象标记为相关或不相关。换句话来说，对于任一个查询 Q，我们假定存在一个二值分类标签的集合，该集合对应于数据中的所有对象，指出哪个对象是相关的，哪个是不相关的。当然，对于实践来说，这是一种简化，因为相关性不一定是一个二值的概念，比如说，杂志文章集合中的某些文章对于某个学生的研究课题可能特别相关或不太相关。此外，这种方法还隐含的假定了相关性是绝对的（不是以用户为中心的（user-centric）），因为相对于给定的查询 Q，每个对象的相关性对于所有用户都是一样的。最后假定已经以某种方式为每个对象附加了标签（假定是以一种比较客观并与人类判断相一致的方式）。在实践中，对于很大的数据集，作出这样的相关性判断是很困难的一项任务。 452

有了这些假定，我们就可以把检索问题看作一种特殊形式的分类问题——类标签依赖于查询 Q，也就是，“对于查询 Q 相关还是不相关”；然后相对 Q 来估计数据库中对象的类标签。然而检索问题具有一些特点，使得对它的处理不同于一般分类。首先，分类变量的定义是由用户掌握的（因为用户定义查询 Q），因此在每次运行系统时都可能变化。第二，主要目标不是分类出数据库中的所有对象，而是返回与用户查询最相关的对象。

14.2.2 查准率对查全率

尽管前面作出了告诫，但是标出大数据集中对象是否相关（相对于给定的预定义查询集合）的通用技术对于客观地评价各种检索算法的性能还是非常重要的。我们将在 14.2.3 节中更详细讨论这种标签问题，一种可行的方法是通过人类专家委员会来判断、区分对象是否相关。

假定我们在一个独立的检验数据集上评价一个指定检索系统相对特定查询 Q 的性能。检验数据中的对象已经被预先分类为相对于查询 Q 是相关还是不相关。假定这个检验数据集没有被这个检索算法使用过（否则的话，这个算法可能记住了给定查询 Q 到分类标签的映射）。我们可以把检索算法想像为就是要对这个数据集中的对象作出分类（按照相对于查询 Q 的相关性）——真实的分类标签对于算法是不可见的，但对于检验来说是已知的。

如果这个算法是使用距离尺度（数据集中的每个对象相对于 Q 的距离）来排列（rank）对象集合的，那么这个算法通常具有一个阈值参数 T。也就是算法将返回 K_T 个对象——和查询对象 Q 的距离小于 T 的 K_T 个对象的有序列表。我们可以通过改变这个阈值来改变检索系统的性能。如果这个阈值非常小，那么我们在决定把哪些对象分类为相关时便很保守。不过，这样我们便会漏掉一些可能相关的对象。如果阈值很大，那么效果相反：返回的对象更多，但是对象实际上不相关的可能性也更大。 453

假定对于有 N 个对象的检验数据集合，检索系统返回了 K_T 个可能相关的对象。那么可以用表 14-1 来归纳这个算法的性能。其中 $N = TP + FP + FN + TN$ 是被标签对象的总数，$TP + FP = K_T$ 是算法返回对象的数量，$TP + FN$ 是相关对象的总数。查准率（precision）被定义为检索出的对象中包含相关对象的比例，也就是 $TP/(TP + FP)$。查全率（recall）被定义为检

索出的相关对象相对于数据集中的相关对象总数的比例，也就是 $TP/(TP + FN)$。这里存在一种天然的折衷：当返回对象数 K_T 增大时（也就是提高阈值使算法将更多的对象分类为相关的），我们可以期望查全率会上升（对于极限的情况，我们可以返回所有对象，这时查全率是 1），然而查准率会下降（随着 K_T 的上升，通常仅仅返回相关对象会更加困难）。如果我们使用不同的阈值 T 来运行检索算法，那么我们会得到一系列（查全率，查准率）的点对。反过来可以使用这些点对描出这个特定检索算法（相对于查询 Q、特定的数据集、以及数据标签）的查全率-查准率曲线。在实践中，我们不是相对于唯一的查询来评价性能，而是相对于一个查询集合来估计平均的查全率-查准率性能（参见图 14-1 中的例子）。注意查全率性能曲线和用来刻划带有可变阈值的二值分类器性能的著名 ROC（receiver-operating characteristic）曲线实质上是等价的。

表 14-1　检索试验的四种可能结果示意

	真实：相关	真实：不相关
算法：相关	*TP*	*FP*
算法：不相关	*FN*	*TN*

表 14-1 中，实验中已经标记出了各文档相关还是不相关（相对于查询 Q）。列对应于真实情况，行对应于算法对文档的判断。*TP*、*FP*、*FN*、*TN* 分别对应于真的为正、假的为正、假的为负和真的为负，其中正负是指算法所给出的分类是否相关。理想的检索算法将产生 $FP = FN = 0$ 的对角矩阵。有时把这种报告分类结果的形式称为混淆矩阵（confusion matrix）。

454 下面考虑如果我们把一系列不同检索算法相对于同一个数据集和查询集合的查全率-查准率曲线画在一起结果会怎样。在大多数情况下，没有哪条曲线会比其他曲线有绝对的优势；也就是说，对于不同的查全率值，根据查准率来看最佳算法是不固定的（参见图 14-1）。因此，我们不能完全根据查全率-查准率曲线来裁判一个算法就比另一个更好。尽管如此，这些曲线对于在一定操作条件范围内评价检索算法的相对、绝对性能还是有价值的。我们可以使用很多模式来通过一个数字概括出查全率-查准率性能，比如检索某
455 些固定数量文档时的查准率、查全率和查准率相等那一点的查准率、或者是多个查全率水平的平均查准率。

14.2.3　查准率和查全率的实践应用

查准率-查全率评价在文本检索中一直特别流行，尽管原则上这种方法对所有类型的数据检索都是适用的。文本检索会议（TREC）就是查准率-查全率评价试验的一个大型例子，这个会议是由美国国家标准技术研究所（NIST）举办的，一般一年一次。在这项试验中使用了很多 G 字节大小的文本文档数据集合，这些数据大约是由一百万个独立的文档（对象）组成的，平均每个文档有 500 个术语索引。这里的一个主要问题是如何评价相关性，特别是如何决定相关文档总数以计算查全率。如果使用 50 个不同查询，那么就需要每个人工裁判员给出 5 千万个分类标签！由于 TREC 会议的参展系统很多（通常为 30 个或更多），所以 TREC 裁判员把他们的裁判范围限制在所有检索系统所返回文档的前 100 篇文档的联合，并假定这个集合通常已经包含了几乎所有的相关文档。因此，每个裁判者仅需作出几千个相关性判断，而不是几千万个。

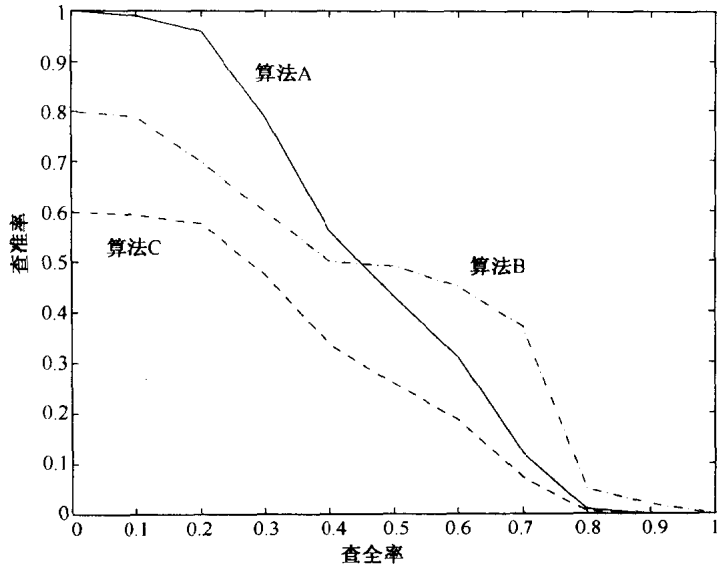


图14-1 三种假想查询算法的查全率-查准率曲线。对于较低的查全率，算法A的查准率最高；对于较高的查全率，算法B具有最高的查准率。在整个范围里，算法C都比算法A和B差，但是我们无法分辨出A和B的性能，除非（举例来说）对于固定的查全率值

更一般地讲，决定查全率是一个重要的实践问题。例如，在检索互联网上的相关文档时，要估计相关文档总数是相当困难的。理论上讲可以使用采样技术，但是，如果考虑了决定相关性时将引入人的主观判断这一事实，就不难理解进行大规模的查准率-查全率试验是相当困难的。

14.3 文本检索

传统上，一直把对文本信息的检索称为信息检索（IR），互联网搜索引擎的出现使其成为一个备受关注的课题。可以把文本看作是由两个基本单位组成的，也就是文档（document）和词条（term）。文档可以是传统的文档，比如说书或杂志的文章，但更一般的情况是指任何结构化的文本片段，比如章节、小节、段落或者甚至是电子邮件、网页、计算机源代码等等。词条可以是单词、词对或文档中的短语，比如单词“数据”和词组“数据挖掘”。

按照 IR 惯例，文本查询是由词条集合指定的。尽管文档通常都比查询长很多，但是使用一种单一的表示语言同时表示文档和查询是很方便的。通过以一种统一的方式来表示这二者，我们可以直接计算查询和文档间的距离，从而为直接实现简单的文本检索算法提供框架。 456

14.3.1 文本的表示

和我们将看到的本章后面要介绍的图像检索一样，大多数关于文本检索的研究集中在寻找支持如下两个特征的通用表示（representation）：

- 尽可能保留数据语义内容的能力；
- 可以高效的计算查询和文档间的距离。

使用检索系统（比如说网络搜索引擎）的用户希望检索出的文档和他所需要的信息在语义内容方面是相关的。从根本上讲，这需要解决一个由来已久的人工智能问题——自然语言

理解（NLP），即通过编程使计算机具有“理解”文本数据的能力，也就是使计算机可以把文本中的ASCII字符映射到某种定义完善的语义表示。已经发现要彻底解决这个问题是非常困难的。*多义词*（同一个词具有几种不同的含义）和*同义词*（使用几种不同的方式来描述同一事物）仅仅是阻碍自动文本理解的两个因素。因此目前使用的大多数IR系统的核心并非是NLP技术（也就是说目前实际的检索系统通常并不包含明确的文档语义模型）。

相反，目前的IR系统通常依赖于简单的词条匹配和计数技术，即通过词条出现次数向量隐含并近似地捕捉了（至少是在理论上）文档语义。假定已经预先定义了要检索的一系列
457 词条 t_j，$1 \leqslant j \leqslant T$，这个集合的规模可以非常大（比如 T = 50 000或更多）。然后把每一篇文档 D_i，$1 \leqslant i \leqslant N$ 表示为词条向量：

$$D_i = (d_{i1}, d_{i2}, \cdots, d_{iT}) \tag{14.1}$$

其中 d_{ij} 表示第 j 个词条在第 i 篇文档中出现的某种信息，各个 d_{ij} 被称为*词条权*（term weight）（不过更确切地说，它们仅是词条向量的分量值）。

在布尔表示中，词条权就是指出某个词条是否在相应的文档中出现，比如说如果文档 i 包含词条 j 那么 $d_{ij} = 1$，否则 $d_{ij} = 0$。*在向量空间表示中*，每个词条权可以是某个实数值的数字，比如说这个词条在文档中出现频繁程度的函数，或者是（可能）这个词条在整个文档集合中的相对频率。在14.3.2节中我们将更详细的讨论词条权。

注意，当一篇文档被表示为 T 维的词条向量时，不仅原始文档中的次序信息丢失了，而且类似语句结构这样的语义信息也失去了。尽管存在这样的信息丢失，词条向量在很多检索应用中仍然是非常有效的。

下面考虑一个涉及10篇文档和6个词条的简单例子。六个词条是

458
- t1 = 数据库㊀
- t2 = SQL
- t3 = 索引
- t4 = 回归
- t5 = 似然
- t6 = 线性的

而且我们可以得到一个 10 × 6 的文档-词条频率矩阵 M，如表14-2所示。元素 ij（i 行，j 列）表示文档 i 包含词条 j 的次数。我们可以清楚地看到前5篇文档 $d1$ 到 $d5$ 主要包含数据库方面的各个词条（查询、SQL和索引的组合），而后5篇文档 $d6$ 到 $d10$ 主要包含回归方面的词条（*回归*、*似然*和*线性*方面的词条）。在本章的后面我们还会讨论这个例子。

表14-2　10篇文档6个词条的文档-词条示例矩阵

	t1	t2	t3	t4	t5	t6
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2

㊀ 译注：根据下文，此处应该为“查询”（query），而不是数据库。

（续）

	t1	t2	t3	t4	t5	t6
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

表 14-2 中，每个元素 ij（i 行，j 列）表示词条 j 在文档 i 中出现的次数。

如果给定了某种向量的空间表示，那么定义文档间距离就很简单了，只要使用一些定义好的距离函数就可以了。第 2 章中所介绍的大多数距离尺度都可以用来（而且已经用来）比较文档。一种广泛应用的距离尺度是**余弦距离**（cosine distance），它是这样定义的：

$$d_c(D_i, D_j) = \frac{\sum_{k=1}^{T} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{T} d_{ik}^2 \sum_{k=1}^{T} d_{jk}^2}} \tag{14.2}$$

这是两个向量夹角的余弦（等价于把它们标准化为单位长度后的内积），因此它反映了两个向量的词条分量的相对分布相似性。尽管这个距离尺度没有什么非常特殊之处，但是已经证明它在实际的 IR 试验中特别有效。

图 14-2 显示的是表 14-2 中的文档-词条频率矩阵的像素形式距离矩阵。图中既显示了欧氏距离矩阵，又显示了余弦距离矩阵。在两种距离矩阵中都可以清楚的看出存在两个文档簇，一类是关于数据库的文档，另一类是关于回归的文档，在图中表现为两个颜色较淡的方形区域。另一方面，不同组的两篇文档间的距离是比较大的（深色的像素）。可以看出，余弦距离更好的区分了两个组。举例来说，在欧氏距离中（上图），文档 3 和文档 4（在数据库簇 459
中）到文档 5（另一篇数据库文档）的距离比到文档 6、8 和 9（关于回归的文档）的距离还要远。导致这一现象的原因就是文档 3 和 4（以及 6、8 和 9）与文档 5 相比更靠近原点。余弦距离发挥了基于角度距离的优点，更强调各个词条的相对分布，因此产生的区分更加明显（见图 14-2 中的下图）。

可以把每个向量 D_i 看作是原始文档的代理（surrogate）文档。并把整个向量集合表示为一个 $N \times T$ 的矩阵。通常这个矩阵是非常稀疏的，比如前面提到的 TREC 文档群大约仅有 0.03%的单元是非零的。对这种矩阵的一种自然解释是这个矩阵的每一行 D_i（一篇文档）是 T 维“词条空间”中的一个向量。因此，如果使用前面章节中用来描述数据集合的数据矩阵来考虑，那么文档的角色就是各个对象，词条就是变量，向量的元素就是对文档的“测量结果”。

在实际实现文本检索系统时，出于对词条-文档矩阵稀疏性的考虑，原始的文档-词条矩阵被表示为一种**倒排文件**（inverted file）结构（而不是直接表示为矩阵形式），也就是按照 T 个词条来索引文件，每个词条 t_j 指向一个 N 个数字的列表，这些数字描述了每篇文档中出现该词条的情况（d_{ij}，j 固定）。

产生词条-文档矩阵本身就不是一件容易的任务，要解决的问题有如何定义词条，比如说是否把名词的单数和复数算作同一个词条？是否该把非常常见的词用作词条？等等。本书中没有详细地论述这个问题，不过我们指出这一部分的“工作量”相当大。

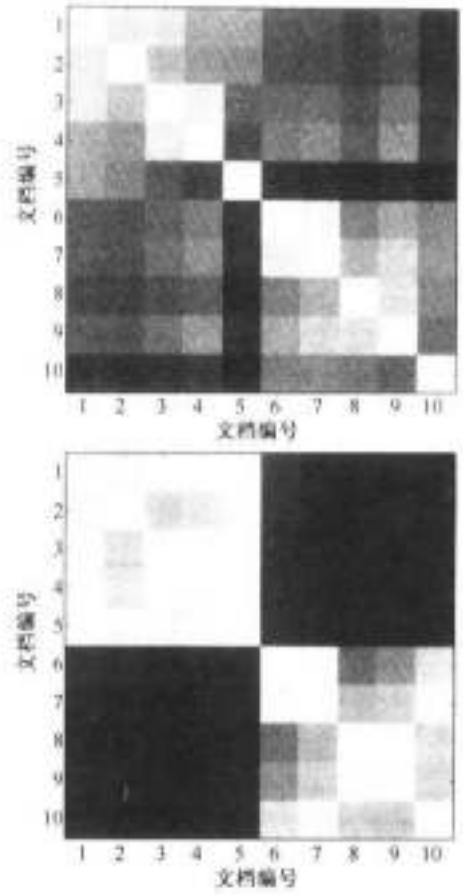


图14-2 文档-词条示例矩阵的两两文档间距离。上图使用的是欧氏距离，下图使用的是余弦距离。较亮的方块表示两篇文档（根据距离尺度）比较相近，较暗的方块表示不太相近。对于欧氏距离，白色对应两篇文档间的距离为0（比如对角线上的方格）；黑色对应于最大的距离。对于余弦距离，较亮的像素对应于较大的余弦值（较小的角度）；较暗的像素对应于较小的余弦值（较大的夹角）

14.3.2 匹配查询和文档

也可以使用与表示文档相同的基于词条的表征来表达查询。实际上，可以把查询本身当作一篇文档来表示，只不过是通常查询仅包含很少的词条（尽管我们当然可以使用一篇真正的文档作为查询，也就是“找出和这篇文档相似的文档”）。

对于布尔表示来说，可以把查询表示为一个逻辑布尔函数，函数的参数是可供使用词条的子集。举例来说，下面是一个典型的查询：“data AND mining AND NOT (coal)”。在这种情况下，检索的基本机制就是扫描倒排文件，判断哪些文档精确地匹配了查询要求。可以对这种基本布尔查询语言进行扩展，比如说增加权用以指出某个词条比其他的更加重要。然而，布尔表示的一个主要不足是不存在一种自然的语义来解释查询和文档间的距离概念，因此没有一种自然的方式来根据相关性对文档进行排序。此外，多少有些令人奇怪，人们经常难以用布尔查询来精确地表达他们的意图。不过，尽管有这些不足，由于布尔查询方法的高效性和简捷性，这种方法在实际 IR 系统中还是很流行的。

460
|
461

在向量空间表示中，可以把查询表示为一个权向量。没有在查询中出现的词条所对应的权根据约定被赋为零。更一般地讲，用户可以指定各个权，以表示每个词条的相对重要性（通常权的取值范围被限定在 0 到 1 之间）。实践中，用户可能对如何用权来概括它们的想法有困难。后面我们会介绍一种被称为相关性反馈（relevance feedback）的模式，利用这种模式可以在多次查询过程中交互式地提炼权，但是在这一节中我们假定用户已给出了查询以及查询中的权。

令 $Q=(q_1, \ldots, q_T)$为查询权向量。在最简单的形式中，查询权要么为 1（这个词条在查询中）要么为 0（这个词条不在查询中），或者使用和表示文档相同的模式来表示查询（参见下文）。下面举个例子来说明简单的简单的二值模式，考虑三个查询，每个都是由一个词条组成的，分别是“数据库”、“SQL”和“回归”。根据前面的例子，可以把这三个查询表达为三个向量：(1, 0, 0, 0, 0, 0)、(0, 1, 0, 0, 0, 0)和(0, 0, 0, 1, 0, 0)。利用余弦距离在表 14-2 所示的数据集中匹配这三个查询，这样便得到了最相近文档，它们分别是 $d2$、$d3$ 和 $d9$。

为了讨论把查询匹配到文档的更一般概念，我们必须首先简要地回顾一下向量空间模型中的权概念。令 d_{ik} 为第 k 个词条在文档 D_i 中的权（也就是分量值），$1 \leqslant k \leqslant T$。IR 文献中已经提出了很多（特别多）关于如何设置这些权的建议，以提高检索的性能。选取这些权的理想目标是使更相关的文档比不太相关的文档有更高的权。已经发现布尔方法（只要某个词条在文档的任何地方出现就把对应的权设为 1）偏向于很大的文档（未必是相关的），这是因为更大的文档更可能在文档的某个地方报给定查询中的词条。 462

已经证明一种被称为 TF-IDF 加权的特殊加权模式在实践中特别有效。TF 代表词条频率（term frequency），就是指词条向量中的每个词条分量被乘以这个词条在文档中出现的频率。这样做的作用是提高了在给定文档中频繁出现的词条的权。表 14-2 中的文档-词条示例矩阵就是以 TF 形式表示的。

然而，如果一个词条在文档集合中的很多文档中都频繁地出现，那么利用 TF 权进行检索的判别力就很小了，也就是它会提高查全率但是查准率可能很差。文档频率倒数（inverse-document-frequency）（IDF）权可以提高判别力。它被定义为 $\log(N/n_j)$，也就是包含词条 j 的文档占整个文档集合的比例的倒数的对数，N 为文档总数。IDF 权偏向于仅在很少文档中出现的词条，也就是说它是有判别力的。使用 IDF 的对数而不是直接使用 IDF 的原因是使这个权对文档总数 N 不特别敏感。

TF-IDF 权就是特定词条在特定文档中的 TF 权和 IDF 权的乘积。和余弦距离尺度（二者经常被一起使用）的情况一样，这种定义权的方式也没有任何特别令人瞩目的动机，但已经发现它的查全率-查准率性能都优于其他的加权模式。有很多不同方法来加强基本的 TF-IDF 方法，但是上面介绍的 TF-IDF 加权仍然是很多评价试验的缺省基准方法。

从文档集合中推导出的 TF-IDF 权可以保持不变的用来对查询词条加权。另一种可选的查询加权方法是仅用 IDF 权来强调比较少见的查询词条。比如说，如果要提交查询“理查德·尼克松”，那么当我们得到包含“尼克松”不包含“理查德”的文档会比得到相反情况的文档会更高兴。

表 14-2 中的文档-词条矩阵所产生的 IDF 权是这样的（使用自然对数）：(0.105, 0.693, 0.511, 0.693, 0.357, 0.693)。注意，第一个词条“数据库”现在的权比其他词条的小了，这是因为包含这一词条的文档更多（也就是说它的判别力较差）。这样便可以得到 TF-IDF 文档-

463 词条矩阵（即把表 14-2 中的 TF 权乘以对应的 IDF 权），如表 14-3 所示。

表 14-3　从表 14-2 得到的 TF-IDF 文档-词条矩阵 ○

2.53	14.56	4.60	0	0	2.07
3.37	6.93	2.55	0	1.07	0
1.26	11.09	2.55	0	0	0
0.63	4.85	1.02	0	0	0
4.53	21.48	10.21	0	1.07	0
0.63	0	0	11.78	1.42	15.94
0.21	0	0	22.18	4.28	0
0.31	0	0	15.24	1.42	1.38
0.10	0	0	23.56	9.63	17.33

在文档中匹配查询的经典方法是这样的：

- 把查询表示为词条向量，1 表示词条出现在查询中，0 表示不出现；
- 利用向量分量的 TF-IDF 权把文档表示为词条向量；
- 使用余弦距离尺度按照文档到查询的距离来排列文档。

表 14-4 显示了一个简单查询实例，比较了 TF 和 TF-IDF 方法。注意，并不像布尔方法那样精确的匹配检索结果（返回所有匹配的文档），距离尺度对至少包含一个相关词条的所有文档进行排序。

表 14-4　在文档中匹配查询

文　档	TF 距　离	TF-IDF 距离
d1	0.70	0.32
d2	0.77	0.51
d3	0.58	0.24
d4	0.60	0.23
d5	0.79	0.43
d6	0.14	0.02
d7	0.06	0.01
d8	0.02	0.02
d9	0.09	0.01
d10	0.01	0.00

表 14-4 中，查询包含的词条是“*数据库*”和“*索引*”，也就是 $Q=(1,0,1,0,0,0)$；对应
464 的文档-词条矩阵来自表 14-2；使用的距离尺度是余弦距离。如果使用 TF 矩阵，文档 $d5$ 是最相近的；使用 TF-IDF 尺度，$d2$ 是最相近的。

14.3.3　隐含语义索引

在前面讨论的文本检索模式中，我们把所有希望都寄托在将文档表示为 T 维词条权向量这一思想上。但是基于词条方法的一个不足是用户可能使用不同的术语来提出查询，这些术语不在用来索引文档的词条当中。举例来说，从词条相似性的角度来看，词条“*数据挖掘*”

○ 译注：经与原作者确认，此表应为十行，且有些行数字不对，请读者自行修正。

和“知识发现”没有什么直接的共同点。然而，从语义角度来看，这两个词条有很大的相同点而且如果我们提出一个包含其中之一的查询，那么我们应该考虑包含另一个的文档。解决这一问题的一种方法是使用预先创建的旨在把语义相关词条连接到一起的知识库（同义词典或本体集）。然而，这样的知识库存在固有的主观性，因为它取决于从何种角度来把词条和语义内容联系起来。

另一种可选的有趣又有价值的方法被称为隐含语义索引（latent semantic indexing）（LSI）。这个名字暗示出 LSI 不是仅使用词条出现信息，而是从文本中提取出隐藏的语义结构信息。实际上，LSI 就是用 T 维词条空间中前 k 个主分量方向来近似原始的 T 维词条空间，使用 $N \times T$ 的文档-词条矩阵来估计这个方向。正如第 3 章中所讨论的，前 k 个主分量方向解释了数据矩阵中的大多数变化，从这个意义上说它提供了 k 个正交基向量（orthogonal basis vectors）的最佳集合。主分量方法可以消除词条中的冗余（如果存在的话）。实践中这样的冗余是经常存在的。举例来说，像“数据库、SQL、索引、查询优化”这样的查询就存在一定的冗余，因为很多数据库方面的文档可能会同时包含这四个词条。主分量方法的直观解释是，由原始词条的加权组合所构成的单个向量可以非常好的近似由大得多的向量集合所起的效果。于是可以把原来的 $N \times T$ 大小的文档-词条矩阵简化为 $N \times k$ 的矩阵，其中 k 可以远远小于 T，这种简化所损失的信息是很少的。从文本检索的角度来看，对于固定的查全率，和前面讨论的向量空间方法相比，LSI 可以提高查准率。

用主分量表示文档-词条矩阵的一个有趣特征是，它通过创建可以更贴切反映文档语义 465
内容的新词条从而捕捉了词条间的关系。例如，如果把词条“数据库、SQL、索引、查询优化”有效地合并成一个单一的主分量词条，那么我们可以认为这个新的词条定义了一篇文档的内容是否是关于数据库概念的。因此，如果有人使用词条 SQL 提出了一个查询，但是文档集合中的有关数据库文档仅包含了“索引”这个词条，那么 LSI 方法将返回这些数据库文档（而严格的基于词条方法不会返回这些文档）。

我们可以对表 14-2 中的矩阵 M 计算奇异值分解式（singular-value decomposition）（SVD）。也就是，找到一个分解式 $M = USV^T$。这里 U 是一个 10×6 的矩阵，它的每一行是相对特定文档的权向量，S 是每个主分量方向特征值的 6×6 对角阵，6×6 的矩阵 V^T 的各列提供了数据的新共轭基，经常被称为主分量方向。

S 矩阵的对角线元素是

$$\lambda_1, \cdots, \lambda_6 = \{77.4, 69.5, 22.9, 13.5, 12.1, 4.8\}$$

可见，前两个主分量捕捉了数据中的主要变化，这和我们的直觉一致。事实上，要是我们仅保留这两个主分量（使用两个代理词条而不是六个），那么这种二维表征所保留的变化比例是 $(\lambda_1^2 + \lambda_2^2)/\sum_{i=1}^{6} \lambda_i^2 = 0.925$，也就是仅丢失了 7.5%的信息（从均方的意义上来说）。如果我们在新的二维主分量空间来表示文档，那么每篇文档的系数对应于 U 矩阵中的前两列：

```
d1   30.8998 -11.4912
d2   30.3131 -10.7801
d3   18.0007  -7.7138
d4    8.3765  -3.5611
d5   52.7057 -20.6051
d6   14.2118  21.8263
```

```
d7  10.8052  21.9140
d8  11.5080  28.0101
d9  9.5259   17.7666
d10 19.9219  45.0751
```

466

而且我们可以把这两列看作新的伪词条，其作用相当于原来6个词条的线性组合。

看一下前两个主分量方向可以得到的信息：

$$\mathbf{v}_1 = (0.74, 0.49, 0.27, 0.28, 0.18, 0.19) \quad (14.3)$$

$$\mathbf{v}_2 = (-0.28, -0.24, -0.12, 0.74, 0.37, 0.31) \quad (14.4)$$

这两个方向（一个平面）是原来六维词条空间中数据最分散（具有最大方差）的方向。第一个方向更突出前两个词条（查询，SQL）：实际上这是描述和数据库有关文档的方向。第二个方向突出了后三个词条——回归、似然和线性，可以认为这是刻画和回归有关文档的

467

方向。图 14-3 以图形方式说明了这一点。我们可以看到，当把文档投影到由前两个主分量方向所决定的平面时，两个不同组的文档分布在两个不同的方向上。注意文档 2 几乎落到文档 1 上，使其有点模糊。（下文讨论了符号 D1 和 D2 的含义。）各点到原点间的距离反映了每篇文档的词条向量（也就是词条数）的幅值。例如，文档 5 和 10 的词条向量最大，因此离原点最远。从图中可以看出，文档间的角度差异显然是相似性的一个有用指标，因为回归和数据库文档在平面上是围绕两个不同的角度聚成簇的。

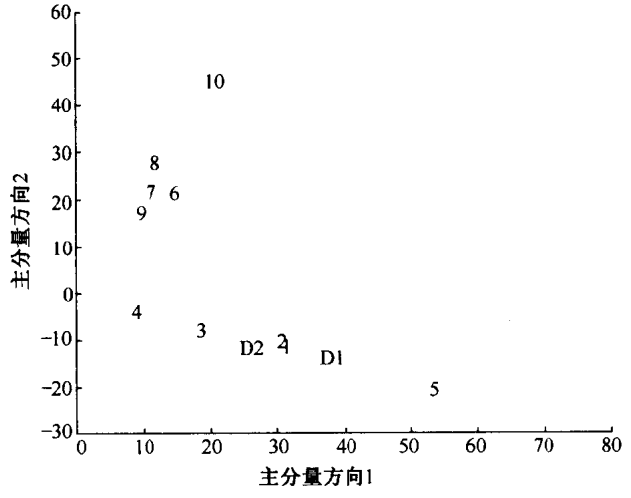


图14-3　主分量方法。图中画出了表14-2中的10篇文档在二维平面上的投影位置，这个平面是由文档-词条矩阵M的前两个主分量决定的

下面举一个例子来说明主分量方法的优点。考虑一个新的文档 D1，词条“*数据库*”[⊖]在该文档中出现 50 次，另一个文档 D2，包含词条“SQL”50 次，而且两篇文档都不包含其他的词条。如果直接使用关键字表示，那么这两文档不会被认为是相似的，因为它们没有包含相同的词条（我们的示例中所使用的特定词条）。然而，如果我们使用两个主分量词条来表示这两篇文档，并把它们投影到这个空间中，那么正如图 14-3 所示的，二者都被投影到“数据库”方向，尽管它们都仅包含和数据库有关的三个词条中的一个。主分量方法隐含地模拟

⊖ 译注：应该为“查询”。

了词条间的相互关系。这一特征对查询很有好处。设想如果我们仅使用词条 SQL 提出一个查询。如果我们使用主分量（例如，前两个伪词条）来表示文档，那么我们可以把查询也转化为伪词条表示，这样查询和数据库文档间的距离（按角度）会比和回归文档间的距离更小，这样便可以检索出根本不包含 SQL 词条但是内容相关的文档。

从计算的角度来看，直接计算主分量向量（例如求解相关矩阵或协方差矩阵的特征值）通常要么是计算上不可行，要么是数值上不稳定。实践中，可以使用特别适合高维稀疏矩阵的 SVD 技术来估计 PCA 向量。

这种基本框架有很多其他的变体。检索文本的主分量方法经常被称为隐含语义索引（LSI）。系统的测试已经证明这种通用技术在很多情况下可以提高检索的性能，这主要是因为它可以匹配不包含相同词条的查询和文档。 468

我们也可以使用概率方式对文档-词条矩阵建模，把这个矩阵看作是由更简单的分量模型所组成的混合模型产生的，每个分量表示以特定主题为条件的单词分布。每个分量模型可以是（举例来说）条件独立的（朴素贝叶斯）也可以是多维正态的，而且可以像第 9 章中所介绍的那样，直接使用 EM 算法来拟合混合模型。

14.3.4 文档和文本分类

从我们的讨论中可以清楚的看出使用词条向量来表示文档为文档分类提供了一种自然框架，有了这一框架对于预先有标签的文档我们可以应用第 10 章的有指导分类，对于没有标签的文档我们可以应用第 9 章的无指导学习（聚类）框架。例如，这种概念的一个实际应用就是把网络文档自动而又准确地聚类成组或类目，以更新并维护网络搜索引擎所使用的庞大数据库。

典型词条向量的维数都是非常高的（例如，10 000 数量级或更多都是很普遍的），由于这一事实，高维空间中的准确性和高效性通常是选择分类器的首要标准。举例来说，尽管分类树通常对高维问题是很有价值的，但是对于文档分类来说单个特征（单个词条）的信息可能还不够丰富。对于体育类文档来说，更可能的情况是这样的文档包含“得分”、“场地”、“体育馆”、“胜利”等词中的某个子集，但不总是包含这个集合中的某个特定单词。因此，对于文档表示来说，像一阶贝叶斯分类器（朴素贝叶斯）这样的分类模型或者是加权线性组合（比如线性支持向量机）往往工作得很好，因为它们以一种比较简单的方式（比如说线性方式）把很多不同的特征组合成分类的依据。前馈神经网络对于大多数文档建模问题都是不可行的，主要原因是不论从模型的参数数量来看，还是从定义训练模型所需的时间来看都过于复杂。

在文档分类这一领域还有很多有趣的问题无法在本书中一一介绍。例如认为每篇文档属于多个主题（类）而不是仅属于某个类是有意义的。这样，便不再限于各个类是互斥的这一通常框架——对于文档来说各个类不一定是互斥的。有很多不同的方法来处理这种“多重隶属”问题。一种简单的方法是为每个类分别训练一个二值分类器，这种方法仅当类别总数较少时是可行的。 469

14.4 对个人偏好建模

14.4.1 相关性反馈

正如前面所指出的，检索系统比本章前面所讨论的其他数据挖掘算法更具交互性。特别是，提出特定查询 Q 的用户可能愿意反复使用算法进行一系列不同的检索尝试，并通过

为返回的文档标记出相关与否来给算法提供用户反馈。可以把这种思想用在各种检索系统中——不仅仅是文本检索，但是本章仅针对文本检索进行讨论。

在这方面，Rocchio 算法应用的特别广泛。该算法的一般思想是：从根本上讲相关性是以用户为中心的，也就是，如果用户可以（理论上）看到所有的文档，那么原则上他可以把所有文档分成两个集合，相关的 R 和不相关的 NR。如果给定了这两个集合，那么可以证明最佳的查询（利用向量模型）为：

$$Q_{optimal} = \frac{1}{|R|}\sum_{D\in R} D - \frac{1}{|NR|}\sum_{D\in NR} D \tag{14.5}$$

其中 D 代表文档的词条向量表示，它的标签（用户作出的）是已知的。

当然实践中，某个特定用户不会一个人把数据库中的所有文档都标上分类标签。相反，用户是从一个特定查询 $Q_{current}$ 开始的，可以把这个查询看作是相对 $Q_{optimal}$ 次优的。算法使用这个初始查询返回文档的一个较小子集，然后用户把这个子集中的文档标记为相关 R'和不相关 NR'。Rocchio 算法按下面的方式来提炼查询：

470

$$Q_{new} = \alpha Q_{current} + \frac{\beta}{|R'|}\sum_{D\in R'} D - \frac{\gamma}{|NR'|}\sum_{D\in NR'} D \tag{14.6}$$

这样便使当前查询朝着被判定为相关文档的均值向量靠近，并远离被判定为不相关文档的均值向量。参数α、β和γ是正的常数（启发式的选取），它们控制着新查询对最近标记文档的敏感性（相对于当前查询向量 $Q_{current}$）。不断重复这个过程，也就是，把新的查询 Q_{new} 与文档集合进行匹配，然后让用户再一次标记文档。注意即使初始查询 Q_0 被用户陈述错了，理论上讲这个算法也可以根据相关性适应并学习到用户的隐含偏好。原则上讲，如果每一次迭代所作的标签是一致的，那么 Q_{new} 会逐步逼近 $Q_{optimal}$。

实验证据表明，这样的用户反馈确实提高了查准率-查全率性能。换句话来说，已经证明融合用户反馈是改善信息检索性能的一种系统有效的方法。当然，要在实践中实现这种方法还有很多细节的问题需要确定，比如说应该显示给读者的文档数量；使用的相关文档和非相关文档的相对数量；选取非相关文档的方法等等，这也就产生了基于这种基本模式的大量变体。

14.4.2 自动推荐系统

我们可以把仅对单一用户偏好建模的方法推广到一种更复杂的情况：使用数据库存储多个用户的信息以及它们对大量对象的偏好。协同过滤（collaborative filtering）技术就是一种发挥这些信息作用的著名方法。举例来说，设想你对某一组音乐感兴趣并在一个网络站点购买了这组音乐的 CD。其他几百个人可能也已经购买了这种 CD，因此很可能至少在音乐品位方面他们的一些偏好和你的是相匹配的。这种情况下，协同过滤就是运行在网络站点上的一种算法，它可以向你提供和你购买了同一张 CD 的人所购买的其他 CD 的列表。显然我们可以从很多个角度来对这种基本思想加以推广。例如，如果我们具有每个用户的采购历史，而且/或者用户愿意提供他们特定兴趣的更多详细信息（以用户简介的形式），那么我们便可以为每个用户建立向量表示，这样本章前面关于定义相似尺度的讨论就可以适用于此了。

从某种意义上来说，协同过滤就是试图捕捉一个很大团体的专业见解和他们的推荐意见，而且这个组是以匹配特定用户兴趣为目标自动选取出的。这种算法通常这样工作：首先找出和目标简介最相似的用户简介，然后根据相匹配简介集合的属性求解推荐意见（把推荐意见作为相匹配简介的属性的函数）。推荐意见的质量依赖于了解的每个用户信息的数量和质量以及用户数据库的大小。这种技术往往在用户数量非常庞大的情况下工作的特别好。在实践中要得到大量用户简介是很困难的，因为用户对花时间提供详细个人信息存在固有的抵触心理。 471

通过用户行为（比如他们买什么，或者他们访问哪些网页）来捕捉用户的偏好是一种不干扰用户又可以暗中估计用户偏好的方式，基于互联网的推荐系统普遍采用这种技术。一种常见的实践问题（例如在电子商务应用中）是算法必须实时地产生推荐意见，比如必须在 1 秒钟之内。如果我们的用户数据库非常庞大（比如记录数为百万数量级），那么就会给计算和数据加工带来严重的挑战。

14.5 图像检索

图像和视频数据集合在不断地增加，从业余爱好者存储的家庭生日宴会数字图像到各种组织（比如(美国)国家航空和宇宙航行局（NASA）以及各种军事机构）远程采集并存储的地球传感图片。随着图像数量的不断增大，人们对图像检索的兴趣也日益浓厚。手工对图像进行注释具有浪费时间、主观性强等缺点，而且可能因为注释者的看法不同而丢失图像的某些特征。一幅图像可能要使用一千个词来描述，但是到底使用哪一千个单词却不是简单的问题！

因此，开发高效而又准确的算法来根据内容对图像数据库进行查询是很有必要的。比如开发交互式的系统，允许用户提交这样的查询“找出和这幅图像最相近的 K 幅图像”或者“找出和这组图像属性最匹配的 K 幅图像”。这种算法的潜在应用非常多：在放射学中搜索相似的诊断图像；寻找有关的影片片段用于广告和杂志；以及在地质、艺术和时尚等领域对图像进行分类编目。 472

14.5.1 图像理解

有必要指出，图像数据查询是非常困难的任务。从某种意义上来说寻找彼此相似的图像等价于求解图像理解问题，也就是从图像数据中抽取语义信息。在这方面人类非常出色，然而，关于模式识别和计算机视觉的几十年研究已经表明，要用计算机算法来“复制”人类在视觉理解和识别方面的能力是极端困难的。（这和前面在文本理解中提到的 NLP 问题非常类似。）虽然目前可以成功解决某些特定的问题，比如面容识别或者起飞跑道探测，但是通用图像理解系统的研究还远未成熟。举例来说，婴儿可以很快地学会在任何背景下辨别各种动物，比如各种大小、颜色、体型（包括卡通图片）的狗，而这种完全无约束的识别问题超出了目前任何视觉算法的能力。这种从原始图像数据中提取语义信息的能力目前还仅为大脑所掌握。因此，目前的大多数图像检索算法还仅依赖于相当低级的可视提示。

14.5.2 图像表示

为了便于检索，可以把原始的像素数据抽象为特征表示，通常是以类似色彩和纹理这样

的原语来表示图像特征。和文本文档的情况一样，也是把原始的图像转换为更标准的数据矩阵格式，每一行（对象）代表一幅特定的图像；每一列（变量）代表一个图像特征。这样的特征表示通常比直接的象素测量值对缩放和平移变化更有效，但是尽管如此，它可能仅对亮度、阴影和视角等的很小变化是保持恒定的。

典型情况下，图像数据库中的图像特征是预先计算并存储好的，以供检索使用。因此只要在多维特征空间中进行距离计算和检索。和文本的情况一样，原始的像素数据被简化为标准的 $N \times p$ 数据矩阵，在这个矩阵中每一幅图像被表示为特征空间中的一个 p 维向量。

通过计算图像局部化子区域的特征可以粗略的引入空间信息。举例来说，我们可以计算
473 一幅 1 024 × 1 024 像素图像的每个 32 × 32 子区域的颜色信息。这样便可以在图像查询中使用粗略的空间约束，比如“寻找中央主要为红色，四周为蓝色的图像”。

除了常规的 $m \times n$ 像素的景物（scenes）图像，图像数据库也可以包含特定的对象图像，也就是单一背景上的对象（比如白色背景上的一张黑色椅子的图像）。因此我们也可以提取针对对象的属性原语，比如对象的颜色、大小和形状（几何信息）特征。视频图像是对图像数据的进一步推广，它把多幅图像（帧）相对时间顺序的连接起来。

应用于图像的根据内容检索系统的一个著名商业实例是 IBM 研究者们在 20 世纪 90 年代早期开发的根据图像内容查询（QBIC）系统。这个系统是建立在 14.5 节所描述的一般思想之上的，它允许用户交互式的查询图像和视频数据，查询的依据可以是图像实例、用户输入的草图、颜色和纹理模式、对象属性等等。该系统允许对景物、对象（景物的一部分）以及视频帧序列或者是这些的任意组合进行查询。QBIC 系统使用了多种特征以及多种和距离有关的尺度用于检索：

- 相对整幅图像进行空间平均的三维颜色特征向量：距离尺度就是简单的欧氏距离。
- K-维颜色直方图，直方图的柱位可以使用像 K-平均这样的基于划分聚类算法来选取，K 值依赖于具体的应用。QBIC 使用颜色直方图向量间的马氏（Mahalanobis）距离尺度来表征颜色相关性。
- 衡量粒度/比例、方向性和对比度特征的三维纹理向量。按照加权的欧氏距离尺度来计算距离，权的缺省值为各个特征方差的倒数。
- 20-维的对象形状特征，比如区域、圆度、离心率、轴方向、各种矩（moments）等等。利用欧氏距离来计算相似性。

14.5.3 图像查询

474 和文本数据的情况相同，用于抽象表示图像的方法（也就是计算特征）决定了支持何种类型的查询和检索操作。特征表示提供了一种表示查询的语言。我们可以用两种基本形式来表示查询。一种方法是通过样例查询，在这种方法中，我们既可以为要寻找的目标提供一个图像样例，也可以勾画出感兴趣图像的形状。接下来便计算样例图像的特征向量，然后再把计算出的查询特征向量和数据库中预先计算出的特征向量进行匹配。另一种方法是直接以特征表征表达查询，比如：“寻找这样的图像：50%的区域为红色，并且包含具有特定方向和粒度特征的纹理”。如果查询是以全部特征的一个子集来表达的（例如在查询中仅指定颜色特征），那么在计算距离时便仅使用这个特征子集。

显然，我们可以根据不同应用对查询形式进行推广（对于给定的特征表示），比如允许

对查询项进行不同的布尔组合。对于图像数据，还可以对查询语言进行特殊处理，使其发挥空间关系（比如，“寻找对象 1 在对象 2 之上的图像”）和序列关系的优势（比如“寻找这样的视频序列：先是足球球员射门，然后是队员们在庆祝”）。

表示图像和查询的特征向量形式与前面讨论的用于文本检索的向量空间表示非常相似。一个主要的差异是图像特征通常是一个实数，例如指出了图像某一区域的特定颜色强度；而词条向量中的词条分量通常是某种形式的加权计数，代表了这个词条在文档中出现的频繁程度。不过，这两种问题都是根据内容检索的问题，这一共同特征决定了用于文本检索的很多技术也适用于图像检索应用，例如使用主分量分析降低特征空间的维度，以及通过 Rocchio 算法进行相关性反馈以改善图像检索过程的性能。

14.5.4 图像恒定性

对于针对图像的根据内容检索问题，我们必须记住（至少对于目前的技术能力来说是这样的）实际上我们仅能使用很有限的语义概念，而且是建立在相当简单的“低级”测量结果上的，比如颜色、纹理、以及对象的简单几何特征。在可视数据中经常存在很多歧变，比如平移、旋转、非线性失真、比例变化和亮度变化（阴影、遮挡（occlusion）、照明等）。人类 475
的视觉系统能够轻松地处理这些歧变，举例来说，对于从完全不同角度、在不同光线下、从不同距离拍摄出的同一个对象的两张照片，人类可以很容易地提取出相同的语义内容（比如“这是在 1995 年以后拍摄的我家房屋”）。

然而，我们前面讨论的根据内容检索方法通常无法在发生了这些歧变的情况下保持恒定性。比例、亮度或观察角度的不同通常都会改变特征的测量结果，从而使景物的歧变版本出现在特征空间中完全不同的位置（与景物的原始版本相比）。换句话来说，检索的结果会随着这些歧变而变化，除非把这种对歧变的恒定性（distortion-invariant）设计到特征表示之中。不过，目前仅知道适用于有限可视环境的抗歧变特征表示，比如刚性对象的线性变换；而对于一般的非刚性对象非线性变换的情况还不清楚。

14.5.5 图像检索的推广

为了对图像检索问题加以总结，我们注意到可以把图像这个术语的解释作进一步推广，使其不仅限于我们前面所描述的现实世界中景物的图像（通常是由照相机产生的）这种隐含解释。更一般地讲，图像数据可以嵌入到文本文档中（比如书和网页）。其他的图像形式包括手工素描（或者手写文本、公式）、油画、线路图（比如建筑和工程上使用的）、图表、曲线、地图等等。显然对于这些情况中的每一种，都必须针对具体的应用来设计检索的方法，不过前面讨论的很多一般原理还是适用的。视频数据的自动索引和交互查询为我们提供了更大的挑战和机遇。比如说，对于像美国有线新闻网络这样的电视新闻组织来说，如果能搜索视频档案并挑选出某种类型的图像，那么会是非常有价值的。

14.6 时间序列和序列检索

在时间序列（time series）和序列数据集合中高效而又准确的定位有意义模式的问题对于很多应用都有重要意义，比如复杂系统的诊断和监控、生物医学数据分析以及对科研和商 476

业时间序列的探索性数据分析。这样的例子包括：

- 找出这样的顾客：他们相对时间的消费模式和给定的消费特征相似；
- 在复杂的实时监控和故障诊断系统（比如航空）中，搜索出与当前异常传感器信号相似的以前实例；
- 在蛋白质序列中进行有噪声子串的匹配。

和二维图像数据相比，可以把序列数据看作是一维的。时间序列（time series）数据或许是最著名的例子，它的一系列观察结果是相对时间测量出的，因此可以用时间变量 t 来索引每个观察值。这种测量经常是按固定时间间隔进行的，这样便可以不失一般性地把 t 看作取值为 1 到 T 的整数。在每个时间 t 的测量结果可能是多元的（并非仅限于单一的测量值），比如每天的股市收盘价格是各个股票价格的集合。时间序列数据的应用非常广泛，对应的领域也五花八门，比如经济、生物医学、生态学、大气和海洋科学、控制工程以及信号处理等。

序列数据（sequential data）的概念比时间序列数据的概念更广，因为序列数据不一定是时间的函数。例如，在计算生物学中，蛋白质是以其在蛋白质序列中的顺序位置来索引的。（当然也可以把文本看作是另一种形式的序列数据，但是通常把它看作单独的一种数据类型。）

与图像和文本数据一样，很多场合都要存储庞大的序列数据集合。例如，(美国)国家航空和宇宙航行局（NASA）的航天飞机在每次执行任务时每秒钟要存储几千个传感器的数据。对于持续几天的任务来说，存储的数据量是很大的（每次任务的数据都在 10G 字节数量级，至今已执行了 100 多次任务）。

可以把这种情况下的检索描述为：寻找和给定查询序列 Q 最佳匹配的子序列。例如，对于航天飞机数据，工程师或许观察到一个可能异常的传感器行为（表示为一个很短的查询
477 序列 Q），并希望断定在以前的飞行中是否存在类似的行为。

14.6.1 时间序列数据的全局模型

传统的时间序列建模技术（比如统计方法）主要是建立在全局线性模型基础上的，就像第 6 章中所讨论的。典型的例子便是 Box-Jenkins 自回归模型族，该方法把当前值 $y(t)$模拟成过去值 $y(t-k)$的加权线性组合，再加上一个额外的噪声项：

$$y(t)=\sum_{i=1}^{k}\alpha_i y(t-i)+e(t) \tag{14.7}$$

其中α_i是加权系数，$e(t)$是时间 t 的噪声（通常被假定为均值为零的高斯函数）。之所以叫“*自动回归*”是因为使用了回归模型的思想（在同一变量的过去值上进行回归）。可以使用第 11 章中非常熟悉的线性回归技术来根据数据估计α_i。并用通常的惩罚似然和交叉验证技术来决定模型结构（也就是阶数 k）。

这种类型的模型和 y 的光谱表示有着密切的关系，因为对于平稳的时间序列过程 y 来说，确定各个α也就确定了它的频率特征。因此很明显，自回归模型仅对于可以完全使用平稳光谱表示刻画的时间序列是有意义的，比如频率特征不随时间变化的线性系统。

Box-Jenkins 方法的一个重要贡献已经被证明，如果在时间序列中存在可识别的系统性非平稳分量（比如某种趋势），那么很多情况下可以把这个不平稳分量删除使这个时间序列变成平稳的形式。举例来说，像国内生产总值和道琼斯指数这样的经济指标中包含着固有的

上升趋势（总体来看），通常要在建模前将这种趋势删除。对于非平稳性比较复杂的情况，另一种有用方法是假定这个信号是相对时间局部平稳（locally stationary）的。举例来说，语音识别系统是这样工作的：首先用来自不同线性系统的序列模拟人类声道和口腔产生的语音序列。然后便用这些线性系统的混合来定义模型，并且假定数据是从这些不同分量线性系统通过某种形式的切换过程（通常是使用马尔可夫过程）产生的。

非线性的全局模型对公式 14.7 进行了推广，比如可以允许 $y(t)$非线性地依赖过去值： 478

$$y(t) = g\left(\sum_{i=1}^{k} \alpha_i y(t-i)\right) + e(t) \qquad (14.8)$$

其中 $g(.)$是非线性的。

这些通用形式（不论是线性还是非线性）的模型有很多扩展版本。它们共有的一个关键特征是，如果给定一个初始条件 $y(0)$以及模型的参数，那么这个过程（y 相对时间的函数分布）的统计量便完全确定了，也就是说，这些模型为时间序列的预期行为提供了一种全局性的简单描述。

从数据挖掘的角度来看，如果我们假定这样的全局模型充分地描述了潜在的时间序列，那么我们就可以使用模型参数（比如上面的各个权）作为表示数据的基础，而不使用原始数据本身。举例来说，如果给定一个不同时间序列的集合（比如不同股票每天的相对时间收益），那么我们可以为每个时间序列拟合一个全局模型（也就是估计出模型的 p 个参数），然后在 p 维参数空间中进行相似性计算。如果同一种模型结构不能完全模拟所有的不同时间序列，那么这种方法便会产生问题。对这一问题的一种解决方案是使用一种嵌套的模型结构（也就是，采用一系列嵌套的复杂度递增的模型结构），并且用最高阶的模型来拟合所有时间序列。

通过把时间序列表示为参数向量，我们实质上是又一次使用了本章前面表示文档和图像的方法。接下来，我们便可以在参数向量空间中定义相似性尺度、在这个空间中定义根据内容检索的查询、等等。

在这个领域中有一种有趣的数据挖掘应用——被称为关键字命中（keyword spotting）的语音识别技术。举例来说，假设有一个监控和记录国际电话对话的国家安全组织（忽略道德和法律问题！）。从安全角度来说，监控的目标是侦探可疑行为。很显然不可能让人来天天监听记录下的大量电话对话。对这一问题的一种自动方法是建立感兴趣的关键字的统计模型。举例来说，我们可以为每个感兴趣的特定关键字（根据训练数据）构建不同的马尔可夫线性切换模型（就像前面讨论的那样）。然后让各个输入语音流平行地穿过每个模型，如果观察 479
到语音数据的似然超过了任一个模型的特定阈值，那么便认为检测到了相应单词，并标记出这个语音流和鉴别出的单词及时间位置。对于这样的系统，自然要考虑大量实际工作量问题，但是基本的概念就是使用一系列训练好的模型可适应的监控实时数据流以探测感兴趣的模式。

14.6.2 时间序列的结构和形状

考虑一个实数值时间序列的子序列 $Q = [q(t), \cdots, q(t+m)]$，和一个长得多的归档时间序列 $X = [x(1), \cdots, x(T)]$，并将前者称为查询序列。我们的目标是在 X 中找到和 Q 最相似的一个子序列。实际情况下，X 可能是由许多单个的时间序列组成的，但是为了简单，我们假定它们已经被合成一条长的序列。此外，为了简单我们还假定 X 和 Q 都是使用相

同采样时间间隔测量的，也就是说，t 递增 1 所对应的时间对二者是相同的。举例来说，Q 可以是一个患者的实时脑电图快照，而 X 可以是已经有诊断结果的其他患者的脑电图档案。

显然，在这种情况下，关于如何定义相似性（similarity）有相当的自由度。注意，上一节所讲的一般方法仅描述了一个时间序列的全局特征，根本没有提供对局部形状（shape）的描述，比如峰值等。通常，全局模型平均了这些局部的结构特征，也就是说，在全局模型表示中没有保留它们。然而，对于很多时间序列来说，用结构特征来描述它们会更自然。一个很好的例子是心脏监控中的 S-T 波形，它有非常独特的可视特征。

一种方法是在整个 X 数据中序列化地扫描查询 Q，顺着 X 每次把查询 Q 移动一个时间点，同时计算出每个时间点的距离尺度（比如欧氏距离）。通常，这样做不仅开销非常惊人（对于蛮力方法来说复杂度是 $O(mT)$），而且其焦点依然集中在低层次（low-level）的数据采样点，而不是高层次的结构特征，比如峰值、高原、走势和波谷等。直接计算出的欧氏距离也对查询 Q 和数据 X 中的微小歧变异常敏感，比如，只要把“理想”的查询 Q 沿时间轴轻微“拉长”，就会导致计算出的距离剧烈增大，即使从视觉观察的角度来看查询 Q 和数据
480 X 仍可以很好地匹配。

这种情况下的一种流行方法是，先局部化地估计出查询 Q 和归档信号 X 的基于形状特征，然后在较高的层次上进行匹配。这样可以使匹配过程有很大的计算优势，因为抽象实质上是一种压缩数据，可以把信号的很多无关细节都忽略掉。更重要的是，它可以以一种适合于人类解释的形式提取结构化的信息。这种技术的一个典型实例是用分段线性化（或者多项式）的片段来逼近信号。然后把分成段的序列表示为局部参数化的曲线列表，而后便可以直接根据参数描述计算结构特征（比如峰和谷）。可以使用概率模型把期望的形状和变化性按这些特征进行参数化，这样便得到一族灵活的可变形的模型模板。可以把在数据档案 X 中匹配 Q 的问题表达为这样的一个搜索问题：给定 Q 的概率模型，在 X 中搜索局部区域使这个区域中数据的似然最大化。对于用全局统计模型不易处理的信号类型这种表示特别有用，比如包含暂态（transient）、阶跃函数（step function）、趋势和其他各种类似某一形状（shapelike）模式的不稳定信号。

对于离散值序列，我们也可以寻找较长序列中的子模式，例如，寻找生物序列数据中出现的图案（motif）。对于这类问题有许多不同技术，从匹配两个串的编辑距离（edit-distance）的非参数方法到利用产生式（generative）马尔可夫模型（或隐马尔可夫模型）的参数模型方法。

14.7　本章归纳

根据内容检索是交互式探索大型数据库的一种重要方法。尤其是对于图像、文本和序列这样的数据类型，根据内容检索算法在很多领域都有重要的应用。然而，要实现普遍适用的算法需要解决几个长期困扰人工智能和模式识别领域的根本问题，比如 NLP 问题的通用解法（对于文本）以及一般性的图像理解问题（对于图像）。简而言之，要开发出可以和人类的大脑相媲美的能从文本和图像这类数据中自动检索语义信息的通用方法，我们还有很长的
481 路要走。

尽管如此，在很多实际应用中由于数据的绝对数量太大以致于手工无法分析，研究人员

还是开发出了很多根据内容检索的技术，这些技术主要是依赖于所谓的“低级语义内容”。比如说我们可以根据像颜色和纹理这样的低级特征来检索图像；根据单词的伴同出现来检索文本。

经常使用的可以跨越不同数据类型的一种常见检索策略大体遵从以下的步骤：

1. 决定一个鲁棒的特征集合用以描述感兴趣的对象。
2. 利用这些特征将原始对象（文本、图像、序列）转换为固定长度的向量表征。
3. 在这个空间中，利用现有丰富的多元数据分析理论计算距离、进行主成分分析等等，匹配查询。

我们可以把这样的系统称为*第一代根据内容检索系统*。当然在有些领域中他们是非常有用的，网络搜索引擎和 QBIC 系统证明了这一点。然而很显然，根据内容检索问题还远未彻底解决，还有相当大的空间有待探索。

14.8 补充读物

Sparck Jones and Willett（1997）文集中包含了很多关于信息（文本）检索的经典论文，其中的一些评论非常深入广泛地探讨了检索问题和研究中的很多核心主题。Van Rijsbergen（1979）、Salton and McGill（1983）以及 Frakes and Baeza-Yates（1992）提供了覆盖这一领域的更多介绍。Salton（1971）包含了关于向量空间表示的许多早期奠基思想，Raghavan and Wong（1986）透视了一些后来的思想。Salton and Buckley（1988）讨论了不同的词条加权方法，虽然很简要但是覆盖面很广，尤其是突出介绍了 TF-IDF 方法。Harman（1993-1999）记录了 TREC 会议，Harman（1995）对 TREC 试验做了一个很有价值的综述。在《Journal of the American Society for Information Science》（1996）的特刊中包含了有关评价文本检索问题 482
的更新讨论。Witten, Moffat, and Bell（1999）精彩地讨论了存储和访问庞大的文本文档所涉及的数据工程方面的很多实践问题。

Deerwester et al.（1990）首次清晰地论证 LSI 在信息检索中的应用。Landauer and Dumais（1997）对使用 LSI 构建语言和知识获取认知模型给出了发人深省的讨论。Berry（1992）以及 Berry, Drmvac, and Jessup（1999）讨论了在像词条-文档表示这样的庞大系数矩阵上进行 SVD 计算的一般技术。Hofmann（1999）介绍了降低文档-词条矩阵维度的基于混合模型的概率方法，为文当建模提供了一个通用的概率框架，而且展示出了很好的实验效果。

“文本挖掘”这一短语是用来描述从文本文档中半自动的发现新的知识的数据挖掘应用。Swanson（1987）以及 Swanson and Smalheiser（1994，1997）介绍了这一领域的一系列有趣研究，他们使用自动的搜索算法发现了在医学文献中看起来无关的子领域间的有趣关系。

Rocchio（1971）介绍了相关性反馈的最初算法。Salton and Buckley（1990）提供了相关性反馈对提高查全率-查准率性能的实验证据，Buckley and Salton（1995）讨论了 Rocchio 算法的最佳方式。Resnick et al.（1994）以及 Shardanand and Maes（1995）介绍了关于协同过滤的最初研究。Breese, Heckerman, and Cadie（1998）讨论了如何对基于模型的协同过滤进行试验评价。Konstan and Riedl（出版过程中）概括了自动推荐系统在电子商务应用中的许多实践问题。Dumais et al.（1998）介绍用于分类文本的支持向量机。

Faloutsos et al.（1994）和 Flickner et al.（1995）较为详细地介绍了 QBIC 系统。第一篇

论文讨论了特征、距离尺度、和使用的索引方案；第二篇论文多少更集中于用户接口问题。其他讨论针对图像和视频的根据内容查询系统的文献包括：Kato，Kurita and Shimogaki（1991），Smoliar and Zhang（1994），Pentland, Picard, and Sclaroff（1996），以及 Smith and Chang
483 （1997）。Rui et al.（1998）讨论了在图像检索中相关性反馈的用法。Maybury（1997）编辑的文集概括了检索多媒体对象（比如图像和视频）这一领域的最新成果。

Box and Jenkins（1976）是讨论时序数据线性全局模型基础的综合性经典教材。Chatfield（1989）讨论的范围更为广泛一些，而且循序渐进的介绍了时间序列的概念，特别适合于不太熟悉这一领域的读者。MacDonald and Zucchini（1997）全面地描述了对离散值时间序列建模的统计方法，Durbin et al.（1998）列举了序列建模和模式识别技术在蛋白质序列和计算生物学有关问题上的应用。

有很多不同的技术可以高效地逼近匹配时间序列的子序列。Faloutsos, Ranganathan and Manolopolous（1994）运用的方法是很典型的。首先把序列分解成各个窗，再从每个窗中提取特征，然后便可以利用一种 R*树结构在特征空间中进行高效的匹配。Agrawal et al.（1995）提出了另一种方法，可以处理振幅的变化、偏移、和数据中的“无所谓”区域，距离是由原始序列的包络（envelope）决定的。Berndt and Clifford（1994）使用动态的时间弯曲（time-warping）方法允许在把查询 Q 匹配到参考序列 R 时时间轴具有“弹性”。另一种流行的方法是把形状的概念抽象化。使用关系树来捕捉序列中波峰（或波谷）的层次，然后使用数匹配算法来比较两个时间序列（Shaw and DeFigueiredo（1990）；Wang et al.,（1994））。Keogh and Smyth（1997）以及 Ge and Smyth（2000）阐述了如何使用可变形的概率模板进行柔韧建模并探测时间序列形状，以及这些方法的实际应用：交互式分析航天飞机传感器数据和联机监
484 控半导体生产数据。

附录 随机变量

A.1 一元随机变量回顾

一元随机变量就是单一的随机变量，设其为 X。如果 X 的定义域是有限的（或者说是可数的），那么我们可以通过列出 X 取每个可能值 x（也就是 $x \in \{x_1, \cdots, x_m\}$）的概率来描述 X 的不确定性。我们把 X 的概率分布写作 $p(X = x)$，或者通常用 $p(x)$来表示单个值的概率分布。当定义域有限时，即当前的情况，概率集合$\{p(x_1), \cdots, p(x_m)\}$经常被称为概率质量函数（probability mass function）。请注意，表达式 $p(X)$是指 m 个数字的集合$\{p(x_1), \ldots, p(x_m)\}$，$p(x)$是指这个集合中的某个（任意）成员。随机变量 X 的累积分布函数（cumulative distribution function）$P(x)$是它取小于等于 x 的值的概率（当 x 值可以排序时）。

也可以为连续的随机变量（可以取一个区间上或实数轴上任意值的变量）定义累积分布函数。这种情况下我们通常用 $F(x)$或 $P(x)$来表示累积分布，并用 $f(x)$或 $p(x)$表示 $F(x)$的导数——x 的概率密度函数（probability density function）（很多时候就简称为“密度函数”）。这个函数给出了观察值位于围绕 x 的无穷小区间内的概率。为了简便，本书经常仅给出密度函数形式的描述，但是类似的结论也适用于概率质量函数的情况。数学统计方面的入门教材更正规的描述了这些概念，不过这些非正式的定义足以满足本书的需要了。

由于很多时候既使用符号 $p(x)$又使用符号 $f(x)$来表示连续变量 x 的概率密度函数。因此 应该从上下文分清它是 x 的概率质量函数还是概率密度函数。

随机变量的随机性是由很多不同原因造成的——实质上也就是不确定性的来源：或许我们所观察的是从总体中随机选取的一个成员；或许测量值存在系统误差；或许 X 是不可以直接观测的，等等。我们经常对这种随机性进行近似：假定实际观察值是由可能值的某个著名分布产生的。某些类型的分布对于数据挖掘特别有价值，附录 A.2 中介绍了其中的一部分。包括正态（也就是高斯）分布和泊松分布。

我们经常使用第 2 章介绍的均值（也就是期望值或期望）概念。对于一个样本（或者是有限总体）来说，均值就是平均值，可以通过把样本（或有限总体）中所有值的和除以值的总个数来得到。更一般地讲，假定 x 值在总体中出现的概率是 $p(x)$。那么变量 X 对于总体的均值就是 $\sum_x xp(x)$。然而，如果 X 可以取连续的值，那么说特定确切值 x 发生的概率是没有意义的，因为确切值发生的概率为 0。这时我们考虑 X 位于宽度为 δx 的很小区间内的概率，并求当这个宽度趋近于 0 时和 $\sum_x xf(x)\delta x$ 的极限值，这样便用积分代替了求和。如果连续变量 X 的概率密度函数为 $f(x)$，那么它的期望值是 $\int xf(x)dx$。

符号 E 经常被用来表示期望，所以随机变量 X 的期望值就是 $E[X]$。希腊字母 μ 经常用来表示均值，如果我们必须明确被讨论的随机变量是 X，那么可以使用 μ_X。更准确地讲，X 相对于密度函数 $f(x)$的期望值被表示为 $E_{f(x)}[X]$。注意，我们可以把 X 的函数 $g(x)$相对于 $f(x)$的期望值定义为 $E_{f(x)}[g(x)] = \int g(x)f(x)dx$，如果我们令 $g(x) = (x - E[x])^2$，那么便得到了方差 σ_x^2 的通常定义。

期望是一种线性算子，这是一个非常有用的一般特征。例如，这意味着多个随机变量加权和的期望值等价于它们的期望值的加权和，不管这些变量是以何种方式相互依赖的（第 4 章中更精确地定义了随机变量的依赖性）。

486 概率公理将不可能发生事件的概率赋值为 0，将一定事件的概率赋值为 1。如果两个事件不可能一起发生，那么一个或另一个发生的概率是它们各自概率的和。因此，在抛硬币问题中（得到正面的概率是 1/2），得到正面或背面的概率是 1/2 + 1/2 = 1。当多个事件可以同时发生但是又不一定如此时，情况就变得更为复杂，也更有趣了。这时便产生了多元随机变量的概念，详细的讨论参见第 4 章。

A.2 一些常见的概率分布

上面讨论了概率分布的一般概念。下面我们介绍一下数据挖掘中常用的概率分布。

伯努里分布

伯努里分布仅有两种可能的结果。可以用这种分布描述的情况包括抛硬币的结果（正面或反面），或者某个顾客是否购买了某种商品。经常用 0 和 1 表示观察结果，如果令 p 为观察到 1 的概率，（$1-p$）为观察到 0 的概率。那么便可以把概率质量函数写作 $p^x(1-p)^{1-x}$，其中 x 取值为 0 或 1。这一分布的均值是 p，方差是 $p(1-p)$。注意这种分布仅有一个参数，也就是 p。

二项分布

它是对伯努里分布的推广，描述了在 n 个独立的伯努里试验（每个的参数为 p）中出现“第一类结果”的次数 x。其概率质量函数的形式为 $\binom{n}{x}p^x(1-p)^{n-x}$，其中 x 可以取 0 到 n 的整型值。其均值是 np，方差是 $np(1-p)$。

多项分布

多项分布是把二项分布推广到存在两种以上结果的情况。例如，存在 k 种可能的结果，第 i 种发生的概率为 p_i，$1 \leqslant i \leqslant k$。各个概率相加的和为 1，这个模型有 $k-1$ 个参数 p_1，…，

487 p_{k-1}（因为 $p_k = 1-\sum_i p_i$）。

假定从一个多项分布中独立地抽取出 n 个观察结果。那么得到第 i 种观察结果的平均数量为 np_i，它的均值是 $np_i(1-p_i)$。注意，因为一种结果的发生便意味着其他结果的不发生，所以各个结果是负相关的。事实上，第 i 种和第 j 种（$i \neq j$）结果间的协方差是 $-np_ip_j$。

泊松分布

如果随机事件是被独立观察的，且其潜在发生率为 λ，那么在长度为 t 的时间间隔内我们可以期望观察到 λt 个事件。当然，有时我们可能在时间 t 内没有观察到事件，而在其他时间里观察到 1 个事件，等等。如果发生率很低，那么很难看到很多个事件（除非 t 非常大）。描述这种事件状态的分布叫泊松分布（Poisson distribution）。它的概率质量函数是 $(\lambda t)^x e^{-\lambda t}/x!$。泊松分布的均值和方差是一样的，都是 λ。

如果一个二项分布的 n 很大，但 p 很小，从而使 np 为一个常量，那么泊松分布可以很好的近似这个二项分布。

正态（也就是高斯）分布

正态分布的概率密度函数具有如下的形式：

$$\frac{1}{\sigma\sqrt{2\pi}}exp\left(\frac{-1}{2\sigma^2}(x-\mu)^2\right)$$

其中μ是分布的均值，σ^2是方差。标准正态分布是均值为零方差为 1 时的特例。正态分布是非常重要的，这部分是因为中心极限定理的作用。粗略地讲，中心极限定理的内容是：n 个观察结果的样本均值随着 n 的增大越来越接近正态分布，不论从中抽取数据的总体分布形式如何。这就是为什么很多统计过程都建立在正态分布假定上的一个原因。

正态分布是关于它的均值对称的，而且其概率的95%都位于距离均值± 1.96 个标准差范围内。 488

学生氏分布（t 分布）

考虑一个来自于正态分布的样本，已知该分布的标准差为σ。可以用以下比例作为推理其均值的检验统计量

$$\frac{\bar{x}-\mu}{\frac{\sigma}{\sqrt{n}}}$$

其中 $\bar{x}$ 是样本均值。利用这一比例，我们可以看出这个样本的均值偏离未知均值的假设值多远。根据中心极限定理（参见上面关于正态分布的讨论），这个比例服从正态分布。注意这里的分母是一个常量。当然在实践中，更可能的情况是在未知标准差的条件下来推测均值。这意味着上面的比例通常会被替换为

$$\frac{\bar{x}-\mu}{\frac{s}{\sqrt{n}}}$$

其中 s 是标准差的样本估计。只要做了这种替换，那么这个比例便不再服从正态分布了，因为分母会随着样本的变化而变化，这样便引入了新的变量。新比例的分布比对应的正态分布跨度更大——末端衰减更平缓（fatter），这个分布被称为 t 分布。注意 t 分布存在很多分布曲线——因样本的大小不同而不同，因为样本大小影响了 s 的变化。可以用（n-1）来索引它们，称为分布的自由度。

我们也可以这样描述上面的情况：如果分子服从正态分布，分母的平方服从卡方分布（参见下文），那么这样的两个随机变量的比例服从 t 分布。

t 分布的概率密度函数是非常复杂的，没有必要在此列出（可以从数理统计方面的教材上得到）。它的均值是 $n-1$，方差是$(n-1)/(n-3)$。

卡方分布

如果 n 个值都服从标准正态分布，那么它们的平方和服从自由度为 n 的卡方分布。该分布的均值为 n，方差为 $2n$。这里也没有必要列出这种分布的概率密度函数了——如果需要的话在数理统计教材上很容易找到。卡方分布在检验拟合度中应用特别广泛。 489

***F* 分布**

如果 u 和 v 是相互独立的服从卡方分布的随机变量，自由度分别为 n_1 和 n_2，那么我们

说下面的比例

$$F = \frac{u}{n_1} / \frac{v}{n_2}$$

服从自由度为 n_1 和 n_2 的 F 分布。这个分布被广泛应用于比较方差的检验中，比如方差应用分析。

多元正态分布

下面把一元正态分布扩展到多个随机变量的情况。令 $\mathbf{x} = (x_1, \cdots, x_p)$表示一个含有 p 个分量的随机向量。那么多元正态分布的概率密度函数具有如下的形式：

$$\frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}$$

其中μ是这个分布的 p 维均值向量，Σ是 $p \times p$ 的协方差矩阵。

就像一元正态分布在概率建模中所起的作用不可替代一样，多元正态分布也是如此。多元正态分布的边际分布是正态的，它的条件分布（也就是在给定一部分变量值的情况下，其余变量子集的联合分布）也是正态的。然而注意，反过来是不成立的：p 个边际分布是正态的并不意味着总的分布就是多元正态的。

490

参考文献

Abiteboul, S., Hull, R., and Vianu, V. (1995) *Foundations of Databases.* Reading, MA: Addison-Wesley.

Adriaans, P., and Zantige, D. (1996) *Data Mining.* Harlow, UK: Addison-Wesley.

Agrawal, R., Aggarwal, C., and Prasad, V. (in press) A tree projection algorithm for finding frequent itemsets. *Journal of Parallel and Distributed Computing.*

Agrawal, R., Imielenski, T., and Swami, A. (1993) Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD'98)*, New York: ACM Press, pp. 207–216.

Agrawal, R., Lin, K.I., Sawhney, H.S., and Shim, K. (1995) Fast similarity search in the presence of noise, scaling, and translation in time-series databases. *Proceedings of VLDB-95*, pp. 490–501.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I. (1996) Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, U.M., Fayyad, G., Piatetsky-Shapiro, P., Smyth, and Uthurasamy, R. (eds.). Menlo Park, CA: AAAI Press, pp. 307–328.

Agrawal, R., and Srikant, R. (1994) Fast algorithms for mining association rules in large databases. *Proceedings of the Twentieth International Conference on Very Large Data Bases (VLDB'94)*, pp. 487–499.

Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, B.N. Petrov and F. Csaki (eds.), Academiai Kiado, Budapest, pp. 267–281.

Alon, N., and Spencer, J.H. (1992) *The Probabilistic Method.* New York: Wiley.

Anderberg, M.R. (1973) *Cluster Analysis for Applications.* New York: Academic Press.

Applebaum, D. (1996) *Probability and Information: An Integrated Approach*, Cambridge, U.K.: Cambridge University Press.

Aronis, J.M., and Provost, F.J. (1997) Increasing the efficiency of data mining algorithms with breadth-first marker propagation. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Heckerman, D., Mannila, H., and Pregibon, D. (eds.). Menlo Park, CA: AAAI Press, pp. 119–122.

Asimov, D. (1985) The grand tour: a tool for viewing multidimensional data. *SIAM Journal of Scientific and Statistical Computing*, 6, pp. 128–143.

Atkeson, C.W., Schaal, S.A., and Moore, A.W. (1997) Locally weighted learning. *Artificial Intelligence Review*, 11, pp. 75–133.

Azzalini, A., and Bowman, A.W. (1990) A look at some data on the Old Faithful geyser. *Applied Statistics*, 39, pp. 357–365.

Babcock, C. (1994) Parallel processing mines retail data. *Computer World*, 6.

Ballard, D.H. (1997) *An Introduction to Natural Computation*. Cambridge, MA: MIT Press.

Banfield, J.D., and Raftery, A.E. (1993) Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49, pp. 803–821.

Barnett, V. (1982) *Comparative Statistical Inference*. Chichester, U.K.: Wiley.

Baum, L.E., and Petrie, T. (1966) Statistical inference for probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 37, pp. 1554–1563.

Bayardo, R.J., and Agrawal, R. (1999) Mining the most interesting rules. *Proc. 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99)*. New York: ACM Press, pp. 145–154.

Becker, R.A., Cleveland, W.S., and Wilks, A.R. (1987) Dynamic graphics for data analysis. *Statistical Science*, 2, pp. 355–395.

Becker, R.A., Eick, S.G., and Wilks, A.R. (1995) Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics*, 1(1), pp. 16–28.

Bennett, K., Fayyad, U., and Geiger, D. (1999) Density-based indexing for approximate nearest-neighbor queries. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, pp. 233–243.

Bernardo, J.M., and Smith, A.F.M. (1994) *Bayesian Theory*. New York, NY: Wiley.

Berndt, D.J., and Clifford, J. (1996) Finding patterns in time-series, a dynamic programming approach. *Advances in Knowledge Discovery and Data Mining*, Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurasamy, R. (eds). Menlo Park, CA: AAAI/MIT Press, pp. 229–248.

Berry, M.J.A., and Linoff, G. (1997) *Data Mining Techniques for Marketing, Sales, and Customer Support*. New York: Wiley.

Berry, M.J.A., and Linoff, G. (2000) *Mastering Data Mining*. New York: Wiley.

Berry, M.W. (1992) Large scale singular value computations. *International Journal of Supercomputer Applications* 6(1), pp. 13–49.

Berry, M.W., Drmvac, Z., and Jessup, E.R. (1999) Matrices, vector-spaces, and information retrieval, *SIAM Review*, 41(2), pp. 335–362.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999) When is "nearest neighbor" meaningful? *Proceedings of the 7th International Conference on Data Theory, ICDT'99*, Lecture Notes in Computer Science, LNCS, Number 1540. New York: Springer-Verlag, pp. 217–235.

Bhandari, I., Colet, E., Parker, J., Pines, Z., Pratap, R., and Ramanujam, K. (1997) Advanced Scout: data mining and knowledge discovery in NBA data. *Data Mining and Knowledge Discovery*, 1(1), pp. 121–125.

Bishop, C.M. (1995) *Neural Networks for Pattern Recognition.* Oxford, U.K.: Clarendon Press, 1995.

Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. (1975) *Discrete Multivariate Analysis*, Cambridge, MA: MIT Press.

Blasius, J., and Greenacre, M. (1998) *Visualization of Categorical Data.* San Diego, CA: Academic Press.

Blum, T., Keislaer, D., Wheaton, J., and Wold, E. (1997) Audio databases with content-based retrieval. *Intelligent Multimedia Information Retrieval*, Maybury, M. T. (ed.). Menlo Park, CA: AAAI Press, pp. 113–135.

Box, G.E.P., and Jenkins, G.M. (1976) *Time Series Analysis: Forecasting and Control.* Oakland, CA: Holden Day.

Box, G.E.P., Jenkins, G.M., and Reinsel, G.C. (1994) *Time Series Analysis: Forecasting and Control*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall.

Bradley, P.S., Fayyad, U.M., and Mangasarian, O.L. (1999) Mathematical programming for data mining: formulation and challenges. *INFORMS Journal on Computing*, 11, pp. 217–238.

Bradley, P.S., Fayyad, U.M., Reina, C. (1998) Scaling clustering algorithms to large databases. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (eds.), Menlo Park, CA: AAAI Press, pp. 9–15.

Breese, J.S., Heckerman, D., and Kadie, C. (1998) Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings 14th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp. 43–52..

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984) *Classification and Regression Trees.* Belmont, CA: Wadsworth Statistical Press.

Brijs, T., Goethals, B., Swinnen, G., Vanhoof, K., and Wets, G. (2000) A data mining framework for optimal product selection in retail supermarket data: the generalized PROFSET model. *Proceedings of the ACM Seventh International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 300–304.

Brin, S., and Page, L. (1998) The anatomy of a large-scale hypertextual Web search engine. *Proceedings of the Seventh International World-Wide Web Conference*, Brisbane, Australia, pp. 107–117.

Brin, S., Motwani, R., and Silverstein, C. (1997) Beyond market baskets: generalizing association rules to correlations. *Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD'97)*, New York: ACM Press, pp. 265–276.

Brooks, S.P., and Morgan, B.J.T. (1995) Optimisation using simulated annealing. *The Statistician*, 44, pp. 241–257.

Buckley, C., and Salton, G. (1995) Optimization of relevance feedback weights. *Proceedings of the 18th Annual ACM 1995 SIGIR Conference*, pp. 351-356.

Buja, A., Cook, D., and Swayne, D.F. (1996) Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1), 78–99.

Buntine, W. (1992) Learning classification trees. *Statistics and Computing*, 2, pp. 63–73.

Buntine, W., Fischer, B., and Pressburger, T. (1999) Towards automated synthesis of data mining programs. In *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, S. Chaudhuri and D. Madigan (eds.), New York, NY: ACM Press, pp. 372–376.

Burges, C.J.C. (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, pp. 121–167.

Burnham, K.P., and Anderson, D.R. (1998) *Model Selection and Inference: a Practical Information Theoretic Approach*. New York: Springer-Verlag.

Böhning, D. (1998) *Computer Assisted Analysis of Mixtures*, Boca Raton, FL: Chapman and Hall.

Cadez, I.V., McLaren, C.E., Smyth, P., and McLachlan, G.J. (1999) Hierarchical models for screening of iron-deficient anemia. *Proceedings of the 1999 International Conference on Machine Learning*, I. Bratko and S. Dzeroski (eds.), San Francisco, CA: Morgan Kaufmann, pp. 77–86.

Cadez, I.V., Heckerman, D., Meek, C., Smyth, P., and White, S. (2000) Visualization of navigation patterns on a Web site using model-based clustering. *Proceedings of the ACM Seventh International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, pp. 280–284.

Card, S.K., MacKinlay, J.D., and Shneiderman, B. (eds.) (1999) *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA: Morgan Kaufmann.

Carmines, E.G., and Zeller, R.A. (1979) *Reliability and Validity Assessment*. Beverly Hills, CA: Sage Publications.

Carr, D.B., Littlefield, R.J., Nicholson, W.L., and Littlefield, J.S. (1987) Scatterplot matrix techniques for large N. *Journal of the American Statistical Association*, 82(398), pp. 424–436.

Casti, J.L. (1990) *Searching for Certainty: What Scientists Can Know about the Future*. New York: Willam Morrow.

Celeux, G., and Govaert, G. (1995) Gaussian parsimonious clustering models. *Pattern Recognition*, 28, pp. 781–793.

Chambers, J.M., Cleveland, W.S., Kleiner, B., and Tukey, P.A. (1983) *Graphical Methods for Data Analysis*. Pacific Grove: Wadsworth and Brooks/Cole.

Chatfield, C. (1996) *The Analysis of Time Series: An Introduction*. London: Chapman and Hall.

Chatterjee, S., Handcock, M.S., and Simonoff, J.S. (1995) *A Casebook for a First Course in Statistics and Data Analysis*. New York: Wiley.

Chaudhuri, S. (1998) An overview of query optimization in relational systems. *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, New York: ACM Press, pp. 34–43.

Chaudhuri, S., and Dayal, U. (1997) An overview of data warehousing and OLAP technology. *Proceedings of the 1997 ACM/SIGMOD Conference*, New York: ACM Press, pp. 65–75.

Cheeseman, P., and Stutz. J. (1996) Bayesian classification (AutoClass): theory and results. In *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.), Cambridge, MA: AAAI/MIT Press, pp. 153–180.

Cheng, X., and Wallace, J.M. (1993) Cluster analysis of the Northern Hemisphere wintertime 500-hPa height field: spatial patterns. *Journal of the Atmospheric Sciences*, 50, pp. 2674–2696.

Cherkassky, V.S., and Muller, F. (1998). *Learning from Data: Concepts, Theory, and Methods*. New York: Wiley.

Chernoff, H. (1973) The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68, pp. 361–368.

Chickering, D.M., and Heckerman, D. (1997) Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29(2/3), pp. 181–244.

Chickering, D.M., Heckerman, D., and Meek, C. (1997) A Bayesian approach to learning Bayesian networks with local structure. *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann, pp. 80–89.

Chipman, H., George, E.I., and McCulloch, R.E. (1998) Bayesian CART model search (with discussion). *Journal of the American Statistical Association*, 93, pp. 935–960.

Clark, P., and Niblett, T. (1989) The CN2 induction algorithm. *Machine Learning*, 3(4), pp. 261–283.

Clearwater, S., and Stern, E. (1991) A rule-learning program in high-energy physics event classification. *Computational Physics Communications*, 67, pp. 159–182.

Cleveland, W.S., and McGill, M.E. (eds.) (1988) *Dynamic Graphics for Statistics*. Belmont, CA: Wadsworth and Brooks/Cole.

Cleveland, W.S., and Devlin, S.J. (1988) Locally-weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83, pp. 597–610.

Cochran, W.G. (1977) *Sampling Techniques*. New York: Wiley.

Cohen, W. (1995) Fast effective rule induction. *Proceedings of the Twelfth International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann, pp. 115–123.

Cook, R.D., and Weisberg, S. (1994) *An Introduction to Regression Graphics*. New York: Wiley.

Cook, R.D., and Weisberg, S. (1999) *Applied Regression Including Computing and Graphics*. New York: Wiley.

Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., and Schrijver, A. (1998) *Combinatorial Optimization*. New York: Wiley.

Corman, T.H., Leiserson, C. E., and Rivest, R.L. (1990) *Introduction to Algorithms*. Cambridge, MA: MIT Press.

Cortes, C., and Pregibon, D. (1998) Giga-mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, R. Agrawal and P. Stolorz (eds.), Menlo Park, CA: AAAI Press, pp. 174–178.

Cox D.R., and Wermuth, N. (1996) *Multivariate Dependencies: Models, Analysis, and Interpretation*. London: Chapman and Hall.

Cox, D.R., and Hinkley, D.V. (1974) *Theoretical Statistics*. London: Chapman and Hall.

Cox, T.F., and Cox, M.A.A. (1994) *Multidimensional Scaling*. London: Chapman and Hall.

Crawford, S.L. (1989) Extensions to the CART algorithm. *International Journal of Man-Machine Studies*, 31, pp. 197–217.

Cressie, N.A.C. (1981) *Statistics for Spatial Data*, New York: Wiley.

Crowder, M. J., and Hand, D. J. (1990) *Analysis of Repeated Measures*. London: Chapman and Hall.

Daly, F., Hand, D.J., Jones, M.C., Lunn, A.D., and McConway, K. (1995) *Elements of Statistics*, Wokingham, U.K.: Addison-Wesley.

Dasarathy, B.V. (ed.) (1991) *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press.

Davidson, M.L. (1983) *Multidimensional Scaling*. New York: Wiley.

Dawes, R.M., and Smith, T.L. (1985) Attitude and opinion measurement. In *The Handbook of Social Psychology*, Volume I (3rd edition), G. Lindzey and E. Aronson (eds.), New York: Random House, pp. 509–566.

Dawid, A.P. (1984) Statistical theory: The prequential approach (with discussion). *Journal of the Royal Statistical Society A*, 147, pp. 178–292.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R. (1990) Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, pp. 391–407.

DeFinetti, B. (1974, 1975) *Theory of Probability*, Vols. 1 and 2. Chichester, U.K.: Wiley.

Della Pietra, S., Della Pietra, V., and Lafferty, J. (1997) Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), pp. 380–393.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39, pp. 1–38.

Devijver, P.A., and Kittler, J. (1982) *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall.

Devroye, L. (1984) *Nonparametric Density Estimation: the L1 View*. New York: Wiley.

Devroye, L., Gyorfi, L., and Lugosi, G. (1996) *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag.

Devroye, L.P., and Wagner, T.J. (1982) Nearest neighbor methods in discrimination. In *Handbook of Statistics*, vol. 2, P.R. Krishnaiah and L.N. Kanal, (eds.) Amsterdam: North-Holland, pp. 193–197.

Diaconis, P., and Shahshahani, M. (1984) On non-linear functions of linear combinations. *SIAM Journal of Scientific Computing*, 5, pp. 175–191.

Diebolt, J., and Robert, C.P. (1994) Bayesian estimation of finite mixture distributions. *Journal of the Royal Statistical Society B*, 56, pp. 363–375.

Dietterich, T.G. (1998) Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7) pp. 1895-1924.

Digby, P., and Kempton, R. (1987) *Multivariate Analysis of Ecological Communities*. London: Chapman and Hall.

Diggle, P.J., Liang, K-Y., and Zeger, S.L. (1994) *Analysis of Longitudinal Data*. Oxford, U.K.: Clarendon Press.

Domingos, P. (1996) Unifying instance-based and rule-based induction. *Machine Learning*, 24, pp. 141–168.

Domingos, P. (1999) A general method for making classifiers cost-sensitive. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 155–164.

Domingos, P., and Hulten, G. (2000) Mining high-speed data streams. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp.71–80.

Domingos, P., and Pazzani, M. (1997) On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29, pp. 103–130.

Draper, N.R., and Smith, H. (1981) *Applied Regression Analysis*, New York: Wiley.

Dryden, I.L., and Mardia, K.V. (1998) *Statistical Shape Analysis*. Chichester, UK: Wiley.

Du Mouchel, W., Volinsky, C., Johnson, T., Cortes, C., and Pregibon, D. (1999) Squashing flat files flatter. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 6–15.

Duda, R.O., and Hart, P.E. (1973) *Pattern Recognition and Scene Analysis*, New York: Wiley.

Duda, R.O., Hart, P.E., and Stork, D.J. (2001) *Pattern Recognition* New York: Wiley.

Dumais, S.T., Platt, J., Heckerman, D., and Sahami, M. (1998) Inductive learning algorithms and representations for text categorization. *Proceedings of the ACM Seventh International Conference on Information and Knowledge Management*, New York: ACM Press, pp. 148–155.

Dunn, G. (1989) *Design and Analysis of Reliability Studies*. London: Arnold.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998) *Biological Sequence*

Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge, U.K.: Cambridge University Press.

Edwards, D. (1995) *Introduction to Graphical Modeling.* New York: Springer Verlag.

Edwards, A.W.F. (1972) *Likelihood.* Baltimore, MD: Johns Hopkins University Press, expanded edition.

Efron, B., and Tibshirani, R.J. (1993) *An Introduction to the Bootstrap,* New York: Chapman and Hall.

Ein-Dor, P., and Feldmesser, J. (1987) Attributes of the performance of central processing units: a relative performance prediction model. *Communications of the ACM,* 30, pp. 308–317.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Science,* 95(25), pp. 14863–68.

Elliott, R.J., Aggoun, L., and Moore, J.B. (1995) *Hidden Markov Models.* New York: Springer-Verlag.

Everitt, B.S. (1981) A Monte Carlo investigation of the likelihood ratio test for the number of components in a mixture of normal distributions. *Multivariate Behavioural Research,* 16, pp. 171–180.

Everitt, B.S., and Hand, D.J. (1981) *Finite Mixture Distributions.* London: Chapman and Hall.

Everitt, B.S., and Dunn, G. (1991) *Applied Multivariate Data Analysis.* New York: Halstead Press.

Everitt, B.S., Gourlay, A.J., and Kendell, R.E. (1971) An attempt at validation of traditional psychiatric syndromes by cluster analysis. *British Journal of Psychiatry,* 138, pp. 336–339.

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. (1994) Efficient and effective querying by image content. *Journal of Intelligent Information Systems,* 3, pp. 231–262.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. (1994) Fast subsequence matching in time-series databases. *Proceedings of the 1994 Annual ACM SIGMOD Conference,* New York, NY: ACM Press, pp. 419–429.

Fan, J., and Gijbels, I. (1996) *Local Polynomial Modeling and its Applications.* London: Chapman and Hall.

Fawcett, T., and Provost, F. (1997) Adaptive fraud detection. *Data Mining and Knowledge Discovery,* 1(3), pp. 291–316.

Fayyad, U.M., Djorgovski S.G., and Weir N. (1996) Automating the analysis

and cataloging of sky surveys. In *Advances in Knowledge Discovery and Data Mining* U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), Menlo Park, CA: AAAI Press, pp. 471–493.

Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P. (1996) From data mining to knowledge discovery: an overview. In *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.). Menlo Park, CA: AAAI Press. pp. 1–34.

Feller, W. (1968) *An Introduction to Probability Theory and its Applications*, Vol. 1 (3rd ed.) New York: Wiley.

Feng, Z.D., and McCulloch, C.E. (1996) Using bootstrap likelihood ratios in finite mixture models. *Journal of the Royal Statistical Society B*, 58(3), pp. 609–617.

Fenton, N.E. (1991) *Software Metrics*. London: Chapman and Hall.

Fine, T.L. (1999) *Feedforward Neural Network Methodology*. NewYork: Springer.

Fisher, R.A. (1936) The use of multiple measurements on taxonomic problems. *Annals of Eugenics*, 7, pp. 179–188.

Fletcher, R. (1987) *Practical Methods of Optimization*. New York: Wiley.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995) Query by image and video content. *IEEE Computer*, 28(9), pp. 23–31.

Florek K., Lukasziwicz J., Perkal J., Steinhaus H., and Zubrzycki S. (1951) Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicum*, 2, pp. 282–285.

Frakes, W.B., and Baeza-Yates, R. (eds.) (1992) *Information Retrieval: Data Structures and Algorithms*, Englewood Cliffs, N.J.: Prentice Hall.

Fraley, C., and Raftery, A.E. (1998) How many clusters? Which clustering method? answers via model-based cluster analysis. *Computer Journal*, 41, pp. 578–588.

Freund, Y., and Schapire, R.E. (1996) Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, San Francisco, CA: Morgan Kaufmann, pp. 148–156.

Friedman, J. (1997) On bias, variance, 0/1 loss, and the curse of dimensionality. *Data Mining and Knowledge Discovery*, pp. 55–77.

Friedman, J.H. (1991) Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19, pp. 1–141.

Friedman, J.H. and Stuetzle, W. (1981) Project pursuit regression. *Journal of*

the American Statistical Association, 76, pp. 817–823.

Friedman, J.H., and Fisher, N.I. (1999) Bump hunting in high dimensional data (with discussion). *Statistics and Computing*, 9, pp. 123–162.

Friedman, J.H.F., Hastie, T., and Tibshirani, R. (2000) Additive logistic regression: a statistical view of boosting, *Annals of Statistics*, 28, 377–386.

Friedman, N., and Goldszmidt, M. (1996) Learning Bayesian networks with local structure. *Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp. 252–262.

Fukuda, T., Morimoto, Y., Morishita, S., and Tokuyama, T. (1996) Mining optimized association rules for numeric attributes. *Proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database and Knowledgebase Systems (PODS'96)*, New York: ACM Press, pp. 182–191.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*, San Diego, CA: Academic Press.

Fukunaga, K., and Flick, T.E. (1984) An optimal global nearest neighbor metric. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 6, pp. 314–318.

Furnival, G.M., and Wilson, R.W. (1974) Regression by leaps and bounds. *Technometrics*, 16, pp. 499–511.

Gaffney, S., and Smyth, P. (1999) Trajectory clustering with mixtures of regression models. In *Proceedings of the ACM 1999 Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, pp. 63–72.

Ganti, V., Gehrke, J., and Ramakrishnan, R. (1999) Mining very large databases. *IEEE Computer*, 32, pp. 38–45.

Garcia-Molina, H., Ullman, J.D., and Widom, J. (1999) *Database System Implementation*. Englewood Cliffs, NJ: Prentice Hall.

Ge, X., and Smyth, P. (2000) Deformable Markov model templates for time series pattern-matching. *Proceedings of the ACM Seventh International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 81–90.

Gehrke, J., Ganti, V., Ramakrishnan, R., and Loh, W-Y. (1999) BOAT—optimistic decision tree construction. *Proceedings of the 1999 ACM SIGMOD conference*. New York: ACM Press, pp. 169–180.

Gehrke, J.E., Ramakrishnan, R., and Ganti, V. (1998) RainForest—a framework for fast decision tree construction of large datasets. *Proceedings of the 24th International Conference on Very Large Databases (VLDB'98)*, pp. 416–427.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1995) *Bayesian Data Analysis*, London: Chapman and Hall.

Geman, S., Bienenstock, E., and Doursat, R. (1992) Neural networks and the bias-variance dilemma. *Neural Computation*, 4(1), pp. 1–58.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (1996) *Markov Chain Monte Carlo in Practice.* London: Chapman and Hall.

Gill, P.E., Murray, W., and Wright, M.H. (1981) *Practical Optimization.* New York: Academic Press.

Glymour, C., Madigan, D., Pregibon, D., and Smyth, P. (1997) Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery.* 1(1), pp. 11–28.

Goer, J.C. (1967) A comparison of some methods of cluster analysis. *Biometrics*, 23, pp. 623–628.

Golden, R.M. (1996) *Mathematical Methods for Neural Network Analysis and Design.* Cambridge, MA: MIT Press.

Goldstein, H. (1995) *Multilevel Statistical Models* (2nd ed.). London: Arnold.

Gordon, A. (1981) *Classification: Methods for the Exploratory Analysis of Multivariate Data.* London: Chapman and Hall.

Gower, J.C. (1974) Maximal predictive classification. *Biometrics*, 30, pp. 643–654.

Gower, J.C., and Hand, D.J. (1996) *Biplots.* London: Chapman and Hall.

Gray, J., Bosworth, A., Layman, A., and Pirahesh, H. (1996) Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *12th International Conference on Data Engineering (ICDE'96),* New Orleans, Louisiana, pp. 152–159.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997) Data Cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery,* 1, pp. 29–53.

Grenander, U. (1996) *Elements of Pattern Theory.* Baltimore, MD: Johns Hopkins University Press.

Grimmett, G.R., and Stirzaker, D.R. (1992) *Probability and Random Processes.* (2nd ed.) Oxford: Clarenden Press.

Gusfield, D. (1997) *Algorithms on Strings, Trees and Sequences.* New York, NY: Cambridge University Press.

Hall, D.J., and Ball, G.B. (1965) ISODATA: A novel method of cluster analysis and pattern classification. Technical Report, Stanford Research Institute, Menlo Park, California.

Halstead, M.H. (1977) *Elements of Software Science.* New York: Elsevier.

Hamilton, J.D. (1994) *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Hamming, R.W. (1991) *The Art of Probability for Scientists and Engineers*, Redwood City, CA: Addison-Wesley.

Han, J., and Fu, Y. (1995) Discovery of multiple-level association rules from large databases, *Proceedings of the Twenty First International Conference on Very Large Data Bases (VLDB'95)*, San Mateo, CA: Morgan Kaufmann, pp. 420–431.

Han, J., and Kamber, M. (2000) *Data Mining: Concepts and Techniques*, San Francisco, CA: Morgan Kaufmann.

Hand, D.J. (1981) *Discrimination and Classification*. Chichester, U.K.: Wiley.

Hand, D.J. (1982) *Kernel Discriminant Analysis*. Chichester, U.K.: Research Studies Press.

Hand, D.J. (1986) Recent advances in error rate estimation. *Pattern Recognition Letters*, 4, pp. 335–346.

Hand, D.J. (1996) Statistics and the theory of measurement (with discussion). *Journal of the Royal Statistical Society, Series A*, 159, pp. 445–492.

Hand, D.J. (1997) *Construction and Assessment of Classification Rules*. London: Wiley.

Hand, D.J., Blunt, G., Kelly, M.G., and Adams, N.M. (2000) Data mining for fun and profit. *Statistical Science*, 15, pp. 111-131.

Hand, D.J., and Crowder, M.J. (1996) *Practical Longitudinal Data Analysis*. London: Chapman and Hall.

Hand, D.J., Daly, F., Lunn, A.D., McConway, K.J., and Ostrowski, E. (eds.) (1994) *A Handbook of Small Data Sets*. London: Chapman and Hall.

Hand, D.J., McConway, K.J., and Stanghellini, E. (1997) Graphical models of applicants for credit. *IMA Journal of Mathematics Applied in Business and Industry*, 8, pp. 143–155.

Hand, D.J., and Yu, K. (1999) Idiot's Bayes—not so stupid after all? Working paper. Department of Mathematics, Imperial College, London.

Harman, D.K. (1993) The First Text Retrieval Conference (TREC-1), NIST SP 500-207, National Institute of Standards and Technology, Gaithersburg, Md.: (annual series, 1993–1999).

Harman, D.K., (1995) *Hypertext—Information Retrieval—Multimedia: Synergieeffekte Elektronischer Informationssysteme, Proceedings of HIM'95*, R. Kuhlen and M. Rittberger (eds.), Konstanz, Germany: Universitaetsforlag Konstanz, pp. 9–28.

Harrison, D. (1993) Backing up. *Neural Computing*, pp. 98–104.

Harvey, A.C. (1989) *Forecasting, Structural Time Series Models, and the Kalman Filter*. Cambridge, UK: Cambridge University Press.

Hastie, T., and Tibshirani, R.J. (1990) *Generalized Additive Models*. London: Chapman and Hall.

Hastie, T., and Tibshirani, R.J. (1996) Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, pp. 607–616.

Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R., and Kadie, C. (2000) Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1, pp. 49–75.

Hendry, D.F. (1995) *Dynamic Econometrics*. New York: Oxford University Press.

Hilden, J. (1984) Statistical diagnosis based on conditional independence does not require it. *Computers in Biology and Medicine*, 14, pp. 429–435.

Hjort, J.S.U. (1993) *Computer Intensive Statistical Methods: Validation, Model Selection, and Bootstrap*. Boca Raton, FL: CRC Press.

Ho, T.K., Hull J.J., and Srihari, S.N. (1994) Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, pp. 66–75.

Hoffmann, T. (1999) Probabilistic latent sematic indexing. *Proceedings of the ACM SIGIR Conference 1999*, New York: ACM Press, pp. 50–57.

Holsheimer, M., Kersten, M., Mannila, H., and Toivonen, H. (1995) A perspective on databases and data mining. *Proceedings of the First International Conference on knowledge discovery and data mining*, Fayyad, U.M., and Uthurusamy, R. (eds.), Menlo Park, CA: AAAI Press, pp. 150–155.

Holte, R.C., (1993) Very simple classification rules perform well on most commonly used data sets. *Machine Learning*, 11, pp. 63–91.

Huba, G.J., Wingard, J.A., and Bentler, P.M. (1981) A comparison of two latent variable causal models for adolescent drug use. *Journal of Personality and Social Psychology*, 40, pp. 180–193.

Huber, P. (1985) Projection pursuit. *Annals of Statistics*, 13(2), pp. 435–475.

Huber, P.J. (1980) *Robust Statistics*. New York: Wiley.

Hunter, J.S. (1980) The national system of scientific measurement. *Science*, 210, 21 November 1980, pp. 869-874.

Hyvarinen, A. (1999) Survey on independent component analysis. *Neural*

Computing Surveys, 2, pp. 94–128.

Imielinski, T., and Mannila, H. (1996) A database perspective on knowledge discovery. *Communications of the ACM*, 39(11), pp. 58–64.

Imielinski, T., and Virmani, A. (1999) MSQL: A query language for database mining. *Data Mining and Knowledge Discovery* 3(4), pp. 373–408.

Imielinski, T., Virmani, A., and Abdulghani, A. (1999) DMajor application programming interface for database mining. *Data Mining and Knowledge Discovery*, 3(4), pp. 347–372.

Jacoby, W.G. (1997) *Statistical Graphics for Univariate and Bivariate Data.* London: Sage Publications.

Jain, A., and Dubes, R. (1988) *Algorithms for Clustering Data.*, Englewood Cliffs, Prentice-Hall.

Jensen, F.V. (1996) *An Introduction to Bayesian Networks.* New York: Springer-Verlag.

Jolliffe, I.T. (1986) *Principal Component Analysis.* New York: Springer-Verlag.

Jordan, M.I. (1999) *Learning in Graphical Models*, Cambridge, MA: MIT Press.

Jordan, M.I., and Jacobs, R.A. (1994) Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, pp. 181-214.

Journal of the American Society for Information Science (1996) Special Issue on Evaluation, 47:1–105.

Karypis, G., and Kumar, V. (1998) A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *Journal of Parallel and Distributed Computing*, 48(1), pp. 71–95.

Kass, R., and Raftery, A. (1995) Bayes factors. *Journal of the American Statistical Association*, 90, pp. 773–795.

Kato, T., Kurita, T., and Shimogaki, H. (1991) Intelligent visual interaction with image database systems—towards the multimedia personal interface. *Information Processing* (Japan), 14, pp. 134–143.

Kaufman, L., and Rousseeuw, P.J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis.* New York: Wiley.

Keim, D.A., and Kriegel, H.-P. (1994) VisDB: database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, September 1994, pp. 40–49.

Kendall, M.G. (1980) *Multivariate Analysis* (2nd ed.). London: Griffin.

Keogh, E., and Smyth, P. (1997) A probabilistic approach to fast pattern matching in time series databases. *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Menlo Park, CA: AAAI Press, pp. 24–30.

Kim, C-J., and Nelson, C.R. (1999) *State-Space Models with Regime Switching: Classical and Gibbs Sampling Approaches with Applications*. Cambridge, MA: MIT Press.

Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P. (1983) Optimization by simulated annealing. *Science*, 220, pp. 671–680.

Kish, L. (1965) *Survey Sampling*. New York: Wiley.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A.I. (1994) Finding interesting rules from large sets of discovered association rules. *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, New York: ACM Press, pp. 401–407.

Knight, K. (2000) *Mathematical Statistics*, Boca Raton, FL: Chapman and Hall.

Knuth, D. (1997). *The Art of Computer Programming: Fundamental Algorithms*, 3rd ed. Reading, MA: Addison Wesley.

Kohavi, R.(1996) Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, OR: AAAI Press, pp. 202–207.

Koontz, W.L.G., Narendra, P.M., and Fukunaga, K. (1975) A branch and bound clustering algorithm. *IEEE Transactions on Computers*, 24, pp. 908–915.

Krantz, D.H., Luce, R.D., Suppes, P., and Tversky, A. (1971) *Foundations of Measurement, Volume 1: Additive and Polynomial Representations*. New York: Academic Press.

Krzanowski, W.J., and Marriott, F.H.C. (1995) *Multivariate Analysis vol. 2: Classification, Covariance Structures, and Repeated Measurements*. London: Arnold.

Lambert, J.M., and Williams, W.T. (1966) Multivariate methods in plant ecology IV: comparison of information analysis and association analysis. *Journal of Ecology*, 54, pp. 635–664.

Lance, G.N., and Williams, W.T. (1967) A general theory of classificatory sorting strategies: 1. Hierarchical systems. *Computer Journal*, 9, pp. 373–380.

Landauer, T.K., and Dumais, S.T., (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2), pp. 211–240.

Lange, K. (1995) A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society B,* 57, pp. 425–437.

Lange, K. (1999) *Numerical Analysis for Statisticians.* New York: Springer-Verlag.

Lapointe, F.J., and Legendre, P. (1994) A classification of pure malt Scotch whiskies. *Applied Statistics,* 43, pp. 237–257.

Lauritzen, S.L. (1996) *Graphical Models.* Oxford: Clarendon Press.

Lavine, M. (1991) Problems in extrapolation illustrated with space shuttle O-ring data. *Journal of the American Statistical Association,* 86, pp. 919–922.

Lavrac, N., and Dzeroski, S. (1994) *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood.

Lawrence, R.D., Almasi, G.S., Kotlyar, V., Viveros, M.S., and Duri, S.S. (2001) Personalization of supermarket product recommendations, *Data Mining and Knowledge Discovery,* to appear.

Leamer, E.E. (1978) *Specification Searches: Ad Hoc Inference with Experimental Data.* New York: Wiley.

Lee, P.M. (1989) *Bayesian Statistics: An Introduction.* London: Edward Arnold.

Lehmann, E.L. (1986) *Testing Statistical Hypotheses.* New York: Wiley.

Lehmann, E.L., and Casella, G. (1998) *Theory of Point Estimation,* New York: Springer-Verlag.

Leighton, G., and McKinlay, P.L. (1930) *Milk Consumption and the Growth of School Children.* London: HMSO.

Leinweber, D. (personal communication) Stupid data miner tricks: Overfitting the S&P 500.

Lewis, H.R., and Papadimitriou, C.H. (1998) *Elements of the Theory of Computation,* second edition. Upper Saddle River, NJ: Prentice-Hall.

Li, M., and Vitanyi, P. (1993) *An Introduction to Kolmogorov Complexity and Its Applications.* New York: Springer.

Lindsey, I. (1994) *Credit Cards: The Authoritative Guide to Payment and Credit Cards.* Leighton Buzzard: Rushmere Wynne.

Lindsey, J.K. (1996) *Parametric Statistical Inference.* Oxford, U.K.: Clarendon Press.

Lindsey, J.K. (1999) *Models for Repeated Measurements,* 2nd ed. Oxford, U.K.: Oxford University Press.

Lindsey, J.K. (1999) Relationships among sample size, model selection and

likelihood regions, and scientifically important differences. *Journal of the Royal Statistical Society, Series D*, 48, pp. 401–411.

Linhart, H., and Zucchini, W. (1986) *Model Selection*. New York: Wiley.

Little, R.J.A., and Rubin, D.B. (1987) *Statistical Analysis with Missing Data*. New York: Wiley.

Looney, C.G. (1997) *Pattern Recognition Using Neural Networks*. Oxford, U.K.: Oxford University Press.

Lovell, M.C. (1983) Data mining. *Review of Economics and Statistics* 65(1), pp. 1–12.

Luce, R.D., Krantz, D.H., Suppes, P., and Tversky, A. (1990) *Foundations of Measurement, Volume 3: Representation, Axiomatization, and Invariance*. San Diego, CA: Academic Press.

Luenberger, D.G. (1984) *Introduction to Linear and Nonlinear Programming*. Menlo Park, CA: Addison-Wesley.

MacDonald, I.L., and Zucchini, W. (1997) *Hidden Markov and Other Models for Discrete-valued Time Series*. London: Chapman and Hall.

MacKay, D.J.C. (1992) A practical Bayesian framework for back-propagation networks. *Neural Computation*, 4, pp. 448–472.

MacMillan, N.A., and Creelman, C.D. (1991) *Signal Detection Theory: A User's Guide*, New York, NY: Cambridge University Press.

MacNaughton-Smith, P., Williams, W.T., Dale, M.B., and Mockett, L.G. (1964) Dissimilarity analysis. *Nature*, 202, pp. 1034–1035.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, L.M. Le Cam, and J. Neyman (eds.) Berkeley: University of California Press, pp. 281–297.

Madigan, D., Raghavan, N., DuMouchel, W., Nason, M., Posse, C., and Ridgeway, G. (in press) Likelihood-based data squashing: A modeling approach to instance construction. *Data Mining and Knowledge Discovery*.

Mangasarian, O. (1997) Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 1(2), pp.183–201.

Mannila, H. (1997) Inductive databases and condensed representations: Concepts for data mining. *International Logic Programming Symposium 1997*, Cambridge, MA: MIT Press, pp. 21–30.

Mannila, H., Toivonen, H., and Verkamo, A.I. (1994) Efficient algorithms for discovering association rules. *Knowledge Discovery in Databases: Papers from the AAAI-94 Workshop (KDD'94)*, Menlo Park, CA: AAAI Press, pp.

181–192.

Mannila, H., Toivonen, H., and Verkamo, A.I. (1997) Discovery of frequent episodes in sequences. *Data Mining and Knowledge Discovery*. 1(3), pp. 259–290.

Maritz, J.S. (1981) *Distribution-Free Statistical Methods*. London: Chapman and Hall.

Marriott, F.H.C. (1971) Practical problems in a method of cluster analysis. *Biometrics*, 27, pp. 501-514.

Maybury, M.T. (ed.) (1997) *Intelligent Multimedia Information Retrieval*. Menlo Park, CA: AAAI Press.

McCullagh, P., and Nelder, J.A. (1989) *Generalized Linear Models*, 2nd ed. London: Chapman and Hall.

McKendrick, A.G. (1926) Applications of mathematics to medical problems. *Proceedings of the Edinburgh Mathematical Society*, 44, pp. 98–130.

McLachlan, G.J. (1987) Error rate estimation in discriminant analysis: recent advances. In *Advances in Multivariate Statistical Analysis*, A.K. Gupta, ed. The Netherlands: Reidel, pp. 233–252.

McLachlan, G.J. (1987) On bootstrapping the likelihood ratio test for the number of components in a normal mixture. *Applied Statistics*, 36, pp. 318–324.

McLachlan, G.J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley.

McLachlan, G.J., and Basford, K.E. (1988) *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker.

McLachlan, G.J., and Krishnan, T. (1998) *The EM Algorithm and Extensions*. New York: Wiley.

McLachlan, G.J., and Peel, D. (1997) On a resampling approach to choosing the number of components in normal mixture models. In *Computing Science and Statistics (Vol 28)*, L. Billard, and N.I. Fisher (eds.). Fairfax Station, VA: Interface Foundation of North America, pp. 260–266.

McLachlan, G.J., and Peel, D. (1998) MIXFIT: An algorithm for the automatic fitting and testing of normal mixture models. *Proceedings of the 14th International Conference on Pattern Recognition*, vol. 1, Los Alamitos, CA: IEEE Computer Society, pp. 553–557.

McLachlan, G.J., and Peel, D. (2000) *Finite Mixture Models*. New York: Wiley.

McLaren, C.E. (1996) Mixture models in haematology: A series of case stud-

ies. *Statistical Methods in Medical Research*, 5, pp. 129–153.

Meilijson, I. (1989) A fast improvement to the EM algorithm on its own terms. *Journal of the Royal Statistical Society B*, 51, pp. 127–138.

Mendell, N.R., Finch, S.J., and Thode, H.C. (1993) Where is the likelihood ratio test powerful for detecting two component normal mixtures? *Biometrics*, 49, pp. 907–915.

Meo, R., Psaila, G., and Ceri, S. (1996) A new SQL-like operator for mining association rules. *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96)*, San Mateo, CA: Morgan Kaufmann.

Michell, J. (1986) Measurement scales and statistics: A clash of paradigms. *Psychological Bulletin*, 100, pp. 398–407.

Michell, J. (1990) *An Introduction to the Logic of Psychological Measurement*. Hillsdale: Lawrence Erlbaum.

Mitchell, M. (1997) *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Mitchell, T. (1997) *Machine Learning*, New York: McGraw-Hill.

Moore, A. (1999) Very fast EM-based mixture model clustering using multiresolution kd-trees. In *Advances in Neural Information Processing Systems 12*, San Francisco, CA: Morgan Kaufmann.

Moore, A.W. (1999) Cached sufficient statistics for automated discovery and data mining from massive data sources. Online white paper, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Moore, A.W., and Lee, M. (1998) Cached sufficient statistics for efficient machine learning with large data sets. *Journal of Artificial Intelligence Research*, 8, pp. 67–91.

Morgan, B.J.T. (1981) Three applications of methods of cluster analysis. *The Statistician*, 30, pp. 205-223.

Morgan, J.N., and Sonquist, J.A. (1963) Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58, pp. 415–434.

Mosteller, F. (1968) Nonsampling errors. In *International Encyclopedia of the Social Sciences*, 5, D.L. Sills (ed.), New York: MacMillan and Free Press, pp. 113–132.

Muggleton, S. (1995) *Foundations of Inductive Logic Programming*, Englewood Cliffs, NJ: Prentice Hall.

Murthy, S.K. (1998) Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2, pp.

345–389.

Myles, J.P., and Hand, D.J. (1990) The multi-class metric problem in nearest neighbour discrimination rules. *Pattern Recognition*, 23, pp. 1291–1297.

Nakhaeizadeh, G., and Taylor, C.C. (eds.) (1997) *Machine Learning and Statistics*. New York: Wiley.

Neal, R. (1996) *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics 118, New York: Springer.

Neal, R., and Hinton, G. (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, Jordan, M.I. (ed.), Cambridge, MA: MIT Press, pp. 355–371.

Nering, E.D., and Tucker, A.W. (1993) *Linear Programs and Related Problems*. Academic Press Inc.

Newcomb, S. (1886) A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics*, 8, pp. 343–366.

Nightingale, F. (1858) *Notes on Matters Affecting the Health, Efficiency, and Hospital Administration of the British Army, founded chiefly on the Experience of the Late War*. London: Harrison.

Oliver, J.J., and Hand, D.J. (1996) Averaging over decision trees. *Journal of Classification*, 13, pp. 281–297.

Papadimitriou, C.H., and Steiglitz, K (1982) *Combinatorial Optimization—Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall.

Park, J.S., Chen, M.S., and Yu, P.S. (1995) An effective hash-based algorithm for mining association rules. *Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD'95)*, New York: ACM Press, pp. 175–186.

Pearl, J. (1984) *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading, MA: Addison-Wesley.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA: Morgan Kaufmann.

Peixoto, J.L. (1990) A property of well-formulated polynomial regression models. *American Statistician*, 44, pp. 26–30.

Pentland, A., Picard, R.W., and Sclaroff, S. (1994) Photobook: Tools for content-based manipulation of image databases. *International Journal of Computer Vision*, 18, pp. 233–254.

Piatetsky-Shapiro, G. (1991) Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*. G. Piatetsky-Shapiro and W. Frawley (eds.), Menlo Park, CA: AAAI Press.

Piatetsky-Shapiro, G. (1999) The data-mining industry coming of age. *IEEE Expert*, 14(6), pp. 32–34.

Platt, J. (1999) Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods—Support Vector Learning*, B. Scholkopf, C.J.C. Burges, and A.J. Smola (eds.), Cambridge, MA: MIT Press, pp. 185–208.

Poulsen, C.S. (1990) Mixed Markov and latent Markov modelling applied to brand choice behavior. *International Journal of Research in Marketing*, 7, pp. 5–19.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1988) *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, UK: Cambridge University Press.

Provost, F., and Kolluri, V. (1999) A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, 3, pp. 131–169.

Provost, F., Jensen, D., and Oates, T. (1999) Efficient progressive sampling. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, pp. 23–32.

Quandt, R.E., and Ramsey, J.B. (1978) Estimating mixtures of normal distributions and switching regressions. *Journal of the American Statistical Association*, 73(364), 730–738.

Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning*, 1, pp. 81–106.

Quinlan, J.R. (1987) Generating production rules from decision trees. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, pp. 304–307.

Quinlan, J.R. (1990) Learning logical definitions from relations, *Machine Learning*, 5, pp. 239–266.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Raghavan, V.V., and Wong, S.K.M. (1986) A critical analysis of the vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5), pp. 100–124.

Ramakrishnan, R., and Gehrke, J. (1999) *Database Management Systems*, Second Edition. New York: McGraw Hill.

Ramsey, J.O., and Silverman, B.W. (1996) *Functional Data Analysis*. New York: Springer-Verlag.

Randles, R.H., and Wolfe, D.A. (1979) *Introduction to the Theory of Nonpara-*

metric Statistics. New York: Wiley.

Rao, M.R. (1971) Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66, pp. 622–626.

Rastogi, R., and Shim, K. (1998) PUBLIC: A decision tree classifier that integrates building and pruning. *Proceedings of the 24th International Conference on Very Large Databases (VLDB'98)*, pp. 405–415.

Redner, R.A., and Walker, H.F. (1984) Mixture densities, maximum likelihood, and the EM algorithm. *SIAM Review*, 26, pp. 195–239.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994) GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, N.C.: ACM Press, pp. 175-186.

Reyment, R., and Jöreskog K.G. (1993) *Applied Factor Analysis in the Natural Sciences*, Cambridge: Cambridge University Press.

Ridgeway, G. (1997) Finite discrete Markov process clustering. Technical Report TR 97-24, Microsoft Research, Redmond, WA.

Ripley, B.D. (1996) *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge University Press.

Rissanen, J. (1987) Stochastic complexity (with discussion). *Journal of the Royal Statistical Society, Series B*, 49, pp. 223–239 and pp. 253–265.

Robbins, H., and Monro, S. (1951) A stochastic approximation method. *Annals of Mathematical Statistics*, 22, pp. 400–407.

Roberts, F.S. (1979) *Measurement Theory with Applications to Decision-making, Utility, and the Social Sciences*. Reading: Addison-Wesley.

Rocchio, J.J. (1971) Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, Salton, G. (ed.). Englewood Cliffs, N.J.: Prentice Hall, pp. 313–323.

Ross, S.M. (1997) *Introduction to Probability Models*. San Diego, CA: Academic Press, 6th ed.

Rui, Y., Huang, T.S., Ortega, M., and Mehrotra, S. (1997) Relevance feedback: a power tool in interactive content-based image retrieval. *Proceedings of the IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), pp. 644–655.

RuleQuest Research (2000) http://www.rulequest.com/cubist-info.html.

Russek, E., Kronmal, R.A., and Fisher, L.D. (1983) The effect of assuming independence in applying Bayes' theorem to risk estimation and classification in diagnosis. *Computers and Biomedical Research*, 16, pp. 537–552.

Salton, G. (ed.) (1971) *The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewood Cliffs, N.J.: Prentice Hall.

Salton, G., and Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513-523.

Salton, G., and Buckley, C. (1990) Improving retrieval performance by relevance feedback. *Journal of the American Society of Information Science*, 41(4), pp. 288-297.

Salton, G., and McGill, M. (1983) *Introduction to Modern Information Retrieval*, New York: McGraw Hill.

Salzberg, S. (1997) On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3), pp. 317–327.

Salzberg, S. (1999) Gene discovery in DNA sequences. *IEEE Expert*, 14(6), pp. 44–48.

Sarawagi, S., Thomas, S., and Agrawal, R. (1998) Integrating mining with relational database systems: Alternatives and implications. *Proceedings of the ACM SIGMOD Conference on Mangement of Data (SIGMOD 1998)*, New York: ACM Press, pp. 343–354.

Sarawagi, S., Thomas, S., and Agrawal, R. (2000) Integrating association rule mining with relational database systems: alternatives and implications. *Data Mining and Knowledge Discovery*, 4, pp. 89–125.

Savasere, A., Omiecinski, E., and Navathe, S. (1995) An efficient algorithm for mining association rules. *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95)*, San Mateo, CA: Morgan Kaufmann, pp. 432–444.

Schafer, J.B., Konstan, J., and Riedl, J. (in press) Electronic commerce recommender applications. *Data Mining and Knowledge Discovery*.

Schaffer, C. (1994) Cross-validation, stacking and bi-level stacking: Meta-methods for classification and learning. In *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and R.W. Oldford (eds.), New York: Springer-Verlag.

Schapire, R.E., Freund, Y., Bartlett, P., and Lee, W.S. (1998) Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5), pp. 1651–1686.

Schervish, M.J. (1995) *Theory of Statistics*. New York: Springer-Verlag.

Schiavo, R., and Hand, D.J. (2000) Ten more years of error rate research. *International Statistical Review*, 68, pp. 295-310.

Scholkopf, B., Burges, C.J.C., and Smola, A.J. (eds.) (1999) *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press.

Schwarz, G. (1978) Estimating the dimension of a model. *Annals of Statistics*, 6, pp. 461–464.

Scott, D.F. (1992) *Multivariate Density Estimation: Theory and Visualization*. New York: Wiley.

Segal, R., and Etzioni, O. (1994) Learning decision lists using homogenous rules. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press, pp. 619–625.

Shafer, G., and Pearl, J. (1990) *Readings in Uncertain Reasoning*. San Mateo: CA, Morgan Kaufman.

Shafer, J.C., Agrawal, R., and Mehta, M. (1996), SPRINT: A scalable parallel classifier for data mining. *Proceedings of the 22nd International Conference on Very Large Databases (VLDB'96)*, San Francisco, CA:Morgan Kaufmann, pp. 544–555.

Shardanand, U., and Maes, P., (1995) Social information filtering: Algorithms for automating "word of mouth." *Proceedings of CHI'95–Human Factors in Computing Systems*, pp. 210-217.

Shaw, S.W., Defigueiredo, R.J.P. (1990) Structural processing of waveforms as trees. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, 38(2), pp. 328–338.

Shepard, R.N., and Arabie, P. (1979) Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, 86, pp. 87–123.

Short, R.D., and Fukunaga, K. (1981) The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27, pp. 622–627.

Shoshani, A. (1997) OLAP and statistical databases: Similarities and differences. *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, New York: ACM Press, pp. 185–196.

Sibson, R. (1973) SLINK: An optimally efficient algorithm for the single link method. *Computer Journal*, 16, pp. 30–34.

Silberschatz, A., and Tuzhilin, A. (1996) What makes patterns interesting in knowledge-discovery systems, *IEEE Transactions on Knowledge and Data Engineering*, 8(6), pp. 970–974.

Silverman, B.W. (1986) *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Simpson, C.H. (1951) The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society, Series B*, 13, pp. 238–241.

Smith, J.R., and Chang, S. (1997) Querying by color regions using VisualSEEk content-based visual query system. *Intelligent Multimedia Information Retrieval*, Maybury, M.T. (ed.). Menlo Park, CA: AAAI Press, pp. 23–41.

Smoliar, S., and Zhang, H. (1994) Content-based video indexing and retrieval. *IEEE Multimedia*, 1, pp. 62–72.

Smyth, P. (1994) Hidden Markov models for fault detection in dynamic systems. *Pattern Recognition*, 27(1), pp.149–164.

Smyth, P. (1997) Clustering sequences using hidden Markov models. In *Advances in Neural Information Processing 9*, M.C. Mozer, M.I. Jordan, and T. Petsche (eds.), Cambridge, MA: MIT Press, pp. 648–654.

Smyth, P. (1999) Probabilistic model-based clustering of multivariate and sequential data. In *Proceedings of the Seventh International Workshop on AI and Statistics*, D. Heckerman, and J. Whittaker eds., San Francisco, CA: Morgan Kaufman, pp. 299–304.

Smyth, P. (2000) Data mining: Data analysis on a grand scale? *Statistical Methods in Medical Research*. 9, pp. 309–327.

Smyth, P. (2000) Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 9, pp. 63–72.

Smyth, P., and Goodman, R. (1992) An information-theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4), pp. 301–306.

Smyth, P., Ide, K., and Ghil, M. (1999) Multiple regimes in northern hemisphere height fields via mixture model clustering. *Journal of the Atmospheric Sciences*, 56(21), pp. 3704–3723.

Sparck Jones, K., and Willett, P. (1997) *Readings in Information Retrieval*. San Francisco: Morgan Kaufmann.

Späth, H. (1979) Clusterwise linear regression. *Computing*, 22(4), pp. 367–73.

Späth, H. (1985) *Cluster Analysis and Dissection*. Chichester, U.K.: Ellis Horwood.

Srikant, R., and Agrawal, R. (1995) Mining generalized association rules. *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95)*, San Mateo, CA: Morgan Kaufmann, pp. 407–419.

Srikant, R., and Agrawal, R. (1996) Mining quantitative association rules in large relational tables. *Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD'96)*, New York: ACM Press, pp. 1–12.

Srikant, R., Vu, Q., and Agrawal, R. (1997) Mining association rules with item constraints. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*, Heckerman, D., Mannila, H., and Pregibon, D. (eds.). Menlo Park, CA: AAAI Press, pp. 67–73.

Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions (with Discussion). *Journal of the Royal Statistical Society, Series B*, 36, pp. 111–147.

Streiner, D.L., and Norman, G.R. (1995) *Health Measurement Scales*, second edition. Oxford: Oxford University Press.

Swanson, D.R. (1987) Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Retrieval*, 38(4), pp. 228–233.

Swanson, D.R., and Smalheiser, N.R. (1994) Assessing a gap in the biomedical literature: Magnesium deficiency and neurologic disease. *Neuroscience Research Communications*, 15, pp. 1–9.

Swanson, D.R., and Smalheiser N.R. (1997) An interactive system for finding complementary literatures: A stimulus to scientific discovery. *Artificial Intelligence*, 91, pp. 183–203.

Suppes, P., Krantz, D.H., Luce, R.D., and Tversky, A. (1989) *Foundations of Measurement, Volume 2: Geometrical, Threshold, and Probabilistic Representations*. San Diego, CA: Academic Press.

Szalay, A.S., Kunszt, P., Thakar, A., and Gray, J. (1999) Designing and mining multi-terabyte astronomy archives: The Sloan Digital Sky Survey. Technical Report MS-TR-99-30, San Francisco, CA: Microsoft Research.

Thall, P.F., and Vail, S.C. (1990) Some covariance models for longitudinal count data with overdispersion. *Biometrics*, 46, pp. 657-671.

Thisted, R.A., (1988) *Elements of Statistical Computing*. London, Chapman and Hall.

Titterington, D.M., Smith, A.F.M., and Makov, U.E. (1985) *Statistical Analysis of Finite Mixture Distributions*. Chichester, U.K.: Wiley.

Toivonen, H. (1996) Sampling large databases for association rules, *Proceedings of the Twenty Second International Conference on Very Large Data Bases (VLDB'96)*, San Mateo, CA: Morgan Kaufmann, pp. 134–145.

Toussaint, G.T. (1974) Bibliography on estimation of misclassification. *IEEE Transactions on Information Theory*, 20, pp. 472–479.

Tsur, D., Ullman, J.D., Abiteboul, S., Clifton, C., Motwani, R., Nestorov, S., and Rosenthal, A. (1998) QueryFlocks: A generalization of association rule mining. *Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD'98)*, New York, NY: ACM Press, pp. 1–12.

Tufte, E.R. (1983) *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.

Tufte, E.R. (1990) *Envisioning Information*. Cheshire, CT: Graphics Press.

Tukey, J.W. (1977) *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Ullman, J.D. (1988) *Principles of Database and Knowledge-Base Systems*, vol. 1. Rockville, MD: Computer Science Press.

Ullman, J.D., and Widom, J. (1997) *A First Course in Database Systems*. Upper Saddle River, NJ: Prentice-Hall.

van Laarhoven, P.J.M., and Aarts, E.H.L. (1987) *Simulated Annealing: Theory and Applications*. Dordrecht, Netherlands: D. Reidel.

Van Rijsbergen, C.J. (1979) *Information Retrieval*. London: Butterworth Press.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag.

Vapnik, V. (1998) *Statistical Learning Theory*. Chichester, U.K.: Wiley.

Wand, M.P., and Jones, M.C. (1995) *Kernel Smoothing*. London: Chapman and Hall.

Wang, J.T., Zhang, K., Jeong, K., and Shasha, D. (1994) A system for approximate tree matching. *IEEE Transactions on Knowledge and Data Engineering*, 6(4), 559–571.

Webb, A. (1999) *Statistical Pattern Recognition*. London: Arnold.

Webb, G. (2000) Efficient search for association rules. *Proceedings of the ACM Seventh International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM Press, pp. 300–304.

Wedel, M., and Kamakura, W.A. (1998) *Market Segmentation: Conceptual and Methodological Foundations*. Boston, MA: Kluwer.

Wegman, E.J. (1990) Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411), pp. 664–675.

Weiss, S., and Indurkhya, N. (1993) Rule-based regression. *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-93*, San Mateo, CA: Morgan Kaufmann, pp. 1072–1078.

Weiss, S., and Indurkhya, N. (1995) Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3, pp. 383–403.

Weiss, S.M., and Indurkhya, N. (1998) *Predictive Data Mining: A Practical Guide*. San Francisco, CA: Morgan Kaufmann.

Whittaker, J. (1990) *Graphical Models in Applied Multivariate Statistics*, Chichester, U.K.: Wiley.

Wilkinson, L. (1999) *The Grammar of Graphics*. New York: Springer-Verlag.

Witten, I.H., and Franke, E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, CA: Morgan Kaufmann.

Witten, I.H., Moffat, A., and Bell, T.C. (1999) *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd ed. San Francisco, CA: Morgan Kaufmann.

Xu, L., Krzyzak, A., and Suen, C.Y. (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, pp. 418–435.

Zamir, O., and Etzioni, O. (1998) Web document clustering: A feasibility demonstration. *Proceedings of the 21st International ACM SIGIR Conference*, New York: ACM Press, pp. 46–54.

Zhang, T., Ramakrishnan, R., and Livny, M. (1997) BIRCH: an efficient data clustering method for very large databases. *Data Mining and Knowledge Discovery*, 1(2), pp. 141–182.

索　引

索引中的页码为英文原书的页码，与书中边栏的页码一致。

A

A Priori algorithm（A Priori 算法），143，147，157-160

A priori relationships（预知关系），197

Absolute error（绝对误差），216

Accurate measurements（精确测量），45

Actual error rate（实际误差率），359

Acyclic directed graphical models（无环有向图模型），290

AD-Tree data structure（AD-树数据结构），425

Additive form（相加形式），189

Additive predictor（相加预报量），393-394

Advanced Scout system（超级侦察系统），14

Agglomerative methods（凝聚方法），308，311-314

Aggregate（聚合量），414

Aggregation（聚合），414

Akaike information criterion (AIC)（Akaike 信息标准），225

Algebra, relational（代数，关系~），409

Algorithm parameters（算法参数），267

Algorithms（算法），参见 Data mining algorithms; *specific types*

Althernative hypothesis（备择假设），124

ANNs（人工神经网络），391-393

Apparent error rate（表观误差率），359

Approximation（近似），169，265，322-323

Artificial neural networks (ANNs)（人工神经网络），391-393

ASCII characters and codes（ASCII 字符和代码），206-207

Association analysis（关联分析），315

Association rules（关联规则），14，158-160，433-435

Assumption（假定），289

AT&T（美国电话电报公司），13，19

Attributes（属性），4，405

Automated recommender systems（自动推荐系统），471-472

Autoregressive models（自回归模型），199-200，202，478

Average of points（各点的平均），297

B

B-trees（B-树），402-403

Backfitting algorithms（反向拟合算法），394

Backpropagation method（反向传播方法），256

Backward elimination（反向消除），380

Backward selection algorithms（反向选择算法），243

Bandwidth（带宽），176，285，350

Basic algorithms for partition-based clustering,（基于划分的基本聚类算法），302-308

Basis functions（基函数），195

Basket data（购物篮数据），405-406

Bayes assumption, first-order（贝叶斯假定，一阶），354

Bayes error rate（贝叶斯误差率），334

Bayes factor（贝叶斯因子），130

Bayes rule（贝叶斯规则），337-338

Bayes theorem（贝叶斯定理），353

Bayesian approximations（贝叶斯近似），322-323

Bayesian estimation（贝叶斯估计），93，96，106，116-124，220，283

Bayesian Information Criterion (BIC)（贝叶斯信息标准）225-227，235，292，380

Bayesian model（贝叶斯模型）120，359，361-362

Beam search strategy（限定范围搜索策略）246，440，

Beam width（束宽），246

Belief networks（信念网络），290

Bernoulli distributions（伯努里分布），487

Best classification tree problem（最佳分类树问题），241

Best unbiased estimators（最佳无偏估计量），107

Beta distributions（β分布），119

Beta posterior（β后验），122-123

Beta prior（β先验），122-123

Between-cluster variation（聚类间差异），297-298

Bias of measurements（测量偏差），45

Bias-variance（偏差-方差），221-224
Trade-off（~折衷），223
Biased estimation（有偏估计），106
Biasing（偏离），283-284
BIC（贝叶斯信息标准），225-227，235，292，380
Binary data（二值数据），36-37
Binomial distributions（二项分布），487
Blind search（盲目搜索），245-246
Bonferroni inequality（Bonferroni 不等式），131
Boolean conditions and propositions（布尔条件和命题），213，411，429，458，461，475
Boosting methods（boosting 方法），358
Bootstrap methods（自展方法），116，360-361
Box,George（Box（人名），George），168
Boxplots（框须图），61-63
Bracketing methods（加括号方法），254-255
Branch-and-bound concept（分枝定界概念），246-247
Breadth-first search（广度优先搜索），245
Brent's method（Brent 方法），254
Brushing（画刷），71
Building models（建模），378-381

C

Canonical discriminant function（正则判别式函数），343
Canonical parameter（正规参数），388
CART algorithms（分类和回归算法），145-151，157，228，335，345
Cartesian product operations（笛卡尔积运算），409，412，
Cases（实例），4
Categorical data（范畴型数据），187，287-292
Categorical variables（范畴型变量），6
Causation（因果关系），101-102
Central limit theorem（中心极限定理），115-116
Centroid of points（各个点的矩心），297
Chaining（链条现象），312-313
Chance（偶然性），93-97。参见 Uncertainty
Chernoff faces（Chernoff 面容），74
Chi-squared distributions（卡方分布）489-490
Chomsky hierarchy of grammars（Chomsky 语法层次），202
City-block metric（城市街区标距），36
Class identifiers（类标识符），367
Class of patterns（模式类），204
Class variables（分类变量），329
Class-conditional approach（分类条件法），335-339
Classical hypothesis testing（经典假设检验），124-130
Classical multidimensional scaling（经典多维缩放），86
Classification document（文档分类），469-470
  in high dimensions,feature selection for（高维中的特征选择），362-363
  maximal predictive（最可能预测分类），301
  multilayer perceptrons for（用于~的多层感知器），153-157
  predictive models for（用于~的预测模型），327-366
    classification models and（分类模型与），329-339
    evaluating and comparing（评估和比较），359-362
    feature selection for high-dimension（高维情况的特征选择），362-363
    linear discriminants and（线性判别式与），341-343
    logistic discriminant analysis（Logistic 判别式分析），352-353
    naive Bayes model（朴素贝叶斯模型），353-356
    nearest neighbor methods（最近邻方法），347-352
    other methods（其他方法），356-359
    overview（概览），180-182，327-329
    perceptrons and（感知器与），339-341
    tree models（树模型），343-347
  text（文本），469-470
Classification And Regression Trees (CART)（分类和回归树），145-151，153，228，335，345
Classification learning（分类学习），169，328-329
Classification models（分类模型）
  background information（背景信息），329-330
  building real classifiers and（建立实际的分类器与），335-339
  decision boundaries（决策边界），330-331
  discriminative（判别），330-331
  probabilistic models and（概率模型与），331-334
Classifiers（分类器）
  building real（建立实际的），335-339
  evaluating and comparing（评估和比较），359-362
Closed form methods，249-250
Cluster analysis（闭合形式的方法），12，293-296，323

Cluster centers（聚类中心），297
Clustering techniques（聚类技术），12-13，279。参见 Hierarchical clustering; partitionbased clustering algorithms;Probabilistic model-based clustering using mixture models
Clusters（聚类），137
Co-occurrence patterns（一起出现的模式），158
Coding,image（编码，图像），166-167
Coefficients（系数），35，37，361
Collaborative filtering（协同过滤），471-472
Collision（冲突），404
Column vectors（列向量），36
Combinatorial optimization（组合优化），236，239
Commensurability（公度性），32
Complete link method（完全链接方法），313
Complexity of models（模型的复杂度）
  nesting and（嵌套），172
  scoring and（评分），220-228
    bias-variance（偏差-方差），221-224
    concepts in comparing,general（比较模型的一般概念），220-221
    penalizing（惩罚复杂性），224-227
    validation and,external（使用外部验证的），227-228
  in selecting predictive models（在选取预测模型中），183
Compression,data（压缩，数据~），166
Computational methods（计算方法），141，235，291
Computational resources（计算资源），268
Concatenation property（结合性），27
Condensed nearest neighbor methods（压缩最近邻方法），352
Conditional density（条件密度），98
Conditional error rate（条件误差率），359
Conditional independence ("naive") Bayes model（条件独立（"朴素"） 贝叶斯模型），353-356
Conditionally independent variables（条件独立变量），99-100，289，354
Confidence（置信度）
  interval（置信区间），115
  itemsets and, frequent（项集，频繁的），430-431
  limits（~边界），115
Confusion matrix（混淆矩阵），361
Conjugate directions（共轭方向），258
Conjugate families of distributions（共轭族分布），122-123
Constrained optimization（约束优化），259-260
Constraints（约束），10
Content retrieval（内容检索），参见 Retrieval by content
Context-free grammar（独立于上下文的语法），202
Contigency table（列联表），188
Contour plots（等高线图），65-67
Convenience samples（顺便样本），21，48
"Cookbook" approach（"菜谱"方法），152-153
Cosine distance（余弦距离），459
Counts（计数），31
Covariance matrix（协方差矩阵），78，299
Covariances（协方差），33-35
Coverage of a pattern（模式的覆盖面），214
Coxcomb plot（锯齿图），11
Credibility interval（可信区间），123
Critical region（临界区），125
Cross-validation（交叉验证） 148-149，227-228，322，360
Cumulative distribution function（累积分布函数），485
Curse of dimensionality（维度效应），19，193-196
Customer transactions（顾客交易），405-406

D

Data（数据），参见 Databases; Graphical data exploration methods; Measurement and data analysis（分析），166-167
  background information（背景信息），25-26
  basket（购物篮），405-406
  binary（二值（二进制）），36-37
  categorical（范畴型），187，287-292
  compression（压缩），166
  cube（立方体），419-420
  defined（定义），25
  experimental（试验），1
  flattened（压平），7，20，43，358
  form of（~的形式），41-44
  geographic（地理的），44
  high-dimensional（高维的），194-196，362-363
  image（图像），44
  market-basket（市场购物篮），158，429-430
  maximum variability in（最大变化性），77
  mode and（最频值），56

model（模型），405
multirelational（多重关系数据），42-43
observational（观察到的），1
orthogonality of（~的正交性），240
"out-of-sample"（"样本外"），227，328，372
quality（质量），44-51
for collection of data（数据群体的），47-51
for individual measurements（单个测量结果的），44-47
poor（低劣），51
repeated measures（重复测量），349-350
sequential（序列），477
spatial（空间~），44
standard（标准），41
structured, models for（结构化的，用于描述结构化数据的模型），197-203
summarizing（总结~），54-57
summary information（归纳信息），52
suspect（可疑的），50-51
in table（表中的），41
time series（时间序列），476-481
transforming（转化），38-41，194-196，363
unordered categorical, joint distributions for（无序范畴型数据，针对~的联合分布），187
warehousing（仓库），417-419
Data management techniques（数据管理技术），17-18，143，296，421-426。参见 Databases
Data matrix（数据矩阵），41，203-206。参见 Data sets
Data mining（数据挖掘），参见 Data mining algorithms
analysis of（分析），144
background information（背景信息），1-4
data sets and（数据集合和～），4-9
databases and（数据库和～），421-426
defined（定义），1
dredging and（和打捞），22-23
fishing and（和垂钓），22-23
interactive（交互式），11，450
keyword spotting and（关键字命中），479
knowledge discovery in databases and（和数据库中的知识发现），3
models and（和模型），1-2，10-11，175，271
patterns and（和模式），1-2，10-11，271
samples and（和样本），93
snooping and（和探查），22-23
statistics and（和统计），18-21
structures（结构），9-11，142
summary information（归纳），23-24
synthesis of（~的合成），144
tasks（任务），11-15，142
visual techniques and（和可视化技术），11
Data mining algorithms（数据挖掘算法），参见 Score functions for data mining algorithms
background information（背景信息），141-145
Classification And Regression Trees（分类和回归树），145-151，153，228，335，345
components of（~的组件），15-18，142-145
defined（定义），141
nonscalable versions of（非伸缩版本），424
reductionist viewpoint（化约主义观点），151-162
A Priori algorithm（A Priori 算法），157-160
background information（背景信息），151-153
multilayer perceptrons for regression and classification and（用于回归和分类的多层感知器），153-157
vector-space for text retrieval and（用于文本检索的向量空间），160-162
scalable versions of（~的可伸缩版本），423-424
summary information（归纳信息），162-164
tuple（组合），146-151，154-155
Data sets（数据集合（数据集））。参见 Databases
data mining and，4-9
defined（已定义的），4，7
heterogeneous（异质的），279
likelihood of（~的似然），108-109
massive（海量），421-426
nature of（~的特征），4-9
pseudo（伪~），425-426
Data-driven hypothesis generation（以数据驱动的方式生成假设），53
Data-squashing（数据挤压），425
Databases（数据库），参见 Data set
aggregation in（~中的聚合），414

background information（背景信息），399-400
data mining and（~和数据挖掘），421-426
data model in（~中的数据模型），405
data warehousing（数据仓库），417-419
index structures（索引结构），402-404
knowledge discovery in（~中的知识发现），3
management of data and（~和数据管理），421-426
manipulating tables and（操纵表），409-412
massive data sets and（海量数据集和~），421-426
memory hierarchy（存储器层次），400-401
multidimensional indexing（多维索引），404-405
online analytical processing（在线分析处理），417-419
operational（业务~），417
purpose of（~的目的），400
query execution and optimization（查询的执行和优化），415-417
relational（关系~），405-409
strategic（策略），417
string（字符串），420-421
Structured Query Language（结构化查询语言），409，413-415
Deciles（十分位点），56
Decision boundaries（决策边界），330-331
Decision region（决策区域），330
Decision surfaces（决策曲面），330-331
Decision trees（决策树），422
Degrees of freedom（自由度），376-377，489-490
Dendrograms（树状图），313
Density estimation（密度估计），12，184
Density function（密度函数），97-98，355，485。参见 Probability distribution and Density function models
Density mixtures（密度混合），279-281
Density models, parametric（密度模型，参数的~），275-279。参见 Probability distribution and density function models
Denumerable domain（不可数的定义域），485
Dependency modeling（依赖模型），12
Dependent variables（依赖变量），35
Depth-first search（深度优先搜索），245
Derived variables（导出变量），198
Descriptive models（描述模型）
background information（背景信息），271-272
cluster analysis（聚类分析），293-296
functions of（~的函数），12-13
goal of（~的目标），12
hierarchical clustering（层次聚类），308-315
agglomerative methods（凝聚方法），308，311-314
background information（背景信息），308-311
divisive methods（分裂方法），308，314-315
nonprobabilistic（非概率的），219
partition-based clustering algorithms（基于划分的聚类算法），296-308
background information（背景信息），296-297
basic algorithms for（基本算法），302-308
for nonprobabilistic predictive models（用于非概率预测模型的~），219
score functions for（用于~的评分函数），296-302
probabilistic model-based clustering（基于模型的概率聚类），315-323
advantages（优点），319
background information（背景信息），315-316
disadvantages（缺点），319-321
examples（例子），316-319
techniques（技术），321-323
probability distribution and density function models and（概率分布和密度函数模型），272-292
background information（背景信息），272-274
Expectation Maximization algorithm for（用于~的期望最大化算法），281-284
joint distributions for categorical data（用于范畴型数据的联合分布），287-292
mixture distributions and densities（混合分布和密度），279-281
nonparametric density estimation（非参数密度估计），284-287
parametric density models（参数密度模型），275-279
score functions for（用于~的评分函数），274-275
score functions for（用于~的评分函数），212，217-219
Deviance of model（模型的偏离度），389-390
Diagnostic methods（诊断方法），10，338，381-384
Dice coefficient（Dice 系数），37
Difference operation（差运算），410

Discovery task, pattern（发现任务，模式），205
Discriminant functions（判别式函数），331
Discriminative approach（判别方法），335-339
Discriminative classification（判别分类），330-331
Disk access, special-purpose algorithms for（磁盘访问，考虑~的有针对性算法），424
Dispersion measurement（离差测量），56
Dispersion parameter（离差参数），388
Dissection（细分），293
Distance（距离）
  cosine（余弦），459
  distance（距离），32-33，85
  edit-distance（编辑-距离），312
  Euclidean（欧氏~），32-33，85，459，480
  Mahalanobis（马氏~），276-277
  measurements（测量），31-38
  minimum（最小化），298
  pairwise（成对的~），312
  between queries and documents（查询和文档间的~），462
  similarity and（相似性和~），15，451
  weighted Euclidean（加权的欧氏~），33
Distortion of samples（样本失真），49-50
Distribution-free tests（独立于分布的检验），129
Distributions（分布），参见 Probability distribution and density function models
  Bernoulli（伯努里），487
  Beta（β），119
  binomial（二项），487
  chi-squared distribution（卡方），489-490
  conjugate families of（共轭族），122-123
  exponential family of（指数族），388
  F（F），490
  finite mixture（有限混合），280
  independently and identically distributed（独立同分布），108
  joint（联合）
    for categorical data（范畴型数据的~），287-292
    for unordered categorical data（无序范畴型数据的~），187
  left-skewed（左-倾斜的），57
  mixture（混合），279-281
  multimodal（多峰型），56，60
  multinominal（多项式），487-488
  multivariate normal（多元正态），490
  Normal（正态），60，113，115-116，118，121-122，127，171，276，350，488
  Poisson（泊松），280-281，388，488
  posterior（后验），117，122-123
  predictive（预测），120-121
  prior（先验），117，122-123
  probability（概率），485，487-490
  relative（相对），459
  right-skewed（右-倾斜的），57
  skewness of（~的倾斜度），56-57
  student's *t*-（学生氏 *t*-），489
Divisive methods（分裂方法），308，314-315
Document（文档），456，461-465，469-470
Dredging（打捞），22-23
Duplicates（重复），411

E

EDA（探索性数据分析），11-12
Edit-distance（编辑-距离），312
Edited nearest neighbor methods（改进的最近邻方法），352
EFFORT (software program)（EFFORT（软件程序）），29-30
EM algorithms（EM 算法），参见 Expectation Maximization algorithms
Entities（实体），4
Episodes（片段），207-208，436-438
Epsem sample（Epsem 样本），134
Errors（误差）
  absolute（绝对），216
  actual error rate（实际误差率），359
  apparent error rate（表观误差率），359
  Bayes error rate（贝叶斯误差率），334
  conditional error rate（条件误差率），359
  defined（定义），373
  estimation（估计），216
  family error rate（族误差率），131
  mean squared（均方~），107，223-224
  misclassification of objects and（误分类对象），359-361

quadratic error function（二次误差函数），340
resubstitution error rate（重新代入误差率），359
risk of（~的风险），45
squared（平方），216
true error rate（真实的误差率），359
Estimation（估计）
Bayesian（贝叶斯），93，96，106，116-124，220，283
biased（有偏的），106
cross-validation（交叉验证），148-149
defined（定义），93
density（密度），12，184
errors（误差），216
maximum likelihood（最大似然），96，106，108-116
nonparametric density（非参数密度），284-287
over（高估），216
parameter（参数），240
probability distribution and density（概率分布和密度），274-275
quasilikelihood（准似然），390
query selectivity（查询选择能力），273
regression and（回归和），13
stochastic（随机的），123，265
unbiased（无偏的），106，227
uncertainty and（不确定性和），105-124
background information（背景信息），105-106
Bayesian（贝叶斯），93，116-124
maximum likelihood and（最大似然和），93，108-116
properties of estimators and,desirable（估计量属性，理想的~），106-108
stochastic（随机性），123-265
under（低估），216
Estimators（估计量），106-109
Euclidean distance（欧氏距离），32-33，85，459，480
Euclidean metric（欧氏标距），36
Euclidean space（欧氏空间），298
Evaluation（评估）
of classifiers（分类器的），359-362
of models and patterns（模型和模式的），229-231
of retrieval systems（检索系统的），452-456
Event-sequence（事件序列），43
"Exclusive-or" structure（"异-或" 结构），71
Expectation Maximization (EM) algorithms（期望最大化算法）
function of（~的功能），21
for mixture models（用于混合模型的），187，281-284
optimization and（优化和），260-265
red blood cell example（血红细胞例子），317-318
Expected value（期望值），486
Experimental data（试验数据），1
Experimental design（试验设计），132
Explainable variation（可解释的变化），179
Explanatory variable（解释变量），168
Exploratory data analysis (EDA)（探索性数据分析），11-12
Exploring data（探索数据），参见 Graphical data exploration methods
Exponential family of distributions（指数族分布），388
Expressive power of model structure（模型结构的表达力），183

F

F distributions（F 分布），490
Factor analysis（因素分析），83
Factor loadings（因素加载），83
Factored form（因式形式），292
Factorization（因式分解），187-193，290
Factors（因素），195
Family error rate（族误差率），131
Family of model structures（模型结构族），238
Fate（天数），93-97。参见 Uncertainty
Feasible region（可行区域），259
Feature extraction approach（特征提取方法），197-198
Feature selection for classification in high dimensions（在高维空间中选择用于分类的特征），362-363
Features（特征），4
Feed-forward neural networks（前馈神经网络），357，391
Fields（场），4，202
File, inverted（文件，倒排），461
Filtering, collaborative（过滤，协同），471-472
Finite mixture distributions（有限混合分布），280
Finite state machine (FSM)（有限状态机），202
First normal form（第一范式），408
First-order autoregressive model（一阶自回归模型），199-201

First-order Bayes assumption（一阶贝叶斯假定），354
First-order Markov property，101
Fisher information（一阶马尔可夫属性），122
Fisher, R.A.（费歇尔，R. A.），341
Fisher's linear discriminant analysis method（费歇尔线性判别式分析方法），331，353，356，362
Fishing（垂钓），22-23
Fitted model（拟合后的模型），10-11
Flattened data（平面数据），7，20，43，358
Forecasting（预报），133
Form of data（数据的形式），41-44
Forward selection algorithms（前向选择），243，379
Freedom, degrees of（自由度），376-377，489-490
Frequency of episode（片段的频率），436-437
Frequent itemsets（频繁项集），429-433
Frequent sets（频繁集），204，431，433-435
Frequentist view of probability（概率的频率论观点），95
From clause（From 子句），413
FSM（有限状态机），202
Function approximation problems（函数近似问题），169
Functional dependency（函数依赖性），206
Furthest neighbor methods（最远邻方法），313

G

Gaussian noise（高斯噪声），199
Generalizations（泛化（推广）），295，377-378，435-436，476
Generalized additive models（推广的相加模型），393-395
Generalized linear models（推广的线性模型），173-353，384-390
Generative models（产生式模型），272
Generic score functions（通用评分函数），16，219
Genetic search（遗传搜索），266-267
Geographic data（地理数据），44
GIGO（垃圾进，垃圾出），44-45
Gini coefficient of performance（性能的 Gini 系数），361
Global models（全局模型），442-443，478-480
Global pattern（局部模式），9
Goodness-of-fit tests（拟合度检验），126，142，372，377
Google system（Google 系统），15
Grades（分级），31
Gradient descent method（梯度下降方法），253
Gradient-based methods（基于梯度的方法），250-251
Grammars（语法），202
Graphical data exploration method（图形化的数据探索方法）
  background information（背景信息），53-55
  hypothesis testing and（假设检验和），53
  multidimensional scaling（多维缩放），84-90
  principal components analysis（主分量分析），74-84
  summarizing data（总结数据），54-57
  visual techniques（可视化技术）
    for more than two variables（用于两个以上变量的），70-74
    for relationships between two variables（用于两个变量间关系的），62-70
    for single variables（用于单个变量的），57-62
Graphical models（图形模型），189-190
Greedy heuristic search methods（贪婪启发搜索方法），241

H

Hash indices（哈希索引），403-404
Hazard（意外），93-97。参见 Uncertainty
Heterogeneous data set（异质数据集），279
Heteroscedasticity（异方差性），381
Heuristic search methods（启发搜索方法），241，244-246，439-440
Hidden Markov models (HMMs)（隐马尔可夫模型），201-202，291
Hidden variables（隐藏变量），187，190-191，195
Hierarchical clustering（层次聚类）
  agglomerative methods（凝聚方法），308，311-314
  background information（背景信息），308-311
  divisive methods（分裂方法），308，314-315
Hierarchical structure（层次结构），44
High-dimensional data（高维数据），194-196，362-363
"Hill-climbing" algorithm（"爬山"算法），244
Histograms（直方图），57-59，61，284
HMMs（隐马尔可夫模型），201-202，291
Homoscedasticity（同方差性），381
Horseshoe effect（马蹄铁效应），88
Hypertetrahedron（超四面体），258

Hypothesis testing（假设检验）
graphical data exploration methods and（图形化的数据探索方法和~），53
random variables and（随机变量和~），99
uncertainty and（不确定性和~），124-132
background information（背景信息），124
classical（经典的~），124-130
in context（数据挖掘中的~），130-132

I

IBM（美国国际商用机器公司），474
Icon plot（图标图），74
Icons（图标），74
Idealization（理想化），95
IDF（文档频率倒数），463
iid（独立同分布），108
Image（图像）
coding（编码），166-167
form of data and（数据的形式），44
invariants（恒定性），475-476
local part of（~的局部），166
queries（查询），473-474
representation（表示），473
retrieval（检索），472-476
understanding（理解），473
whole（整个），166
Improper priors（不合适的先验），122
Independence in high dimensions（高维中的独立性），187-193
Independent variables（独立变量），99，188-189
Independently and identically distribution (iid)（独立同分布），108
Indicator matrix（指示矩阵），429-430
Individual contribution（个体分布），170
Individual preferences, modeling（个人偏好，对~建模）470-472
Individual X variables（单个的 X 变量），194-195
Individuals（个体），4
Inference（推理），377-378
Information retrieval (IR)（信息检索）。参见 Text retrieval
Input variable（输入变量），329
Inspection, model（审查，模型），381-384
Interactive techniques（交互技术），11，456
Interestingness, criteria for（有趣度，~标准），440-441
Interquartile range（四分位值域），56
Intersection operation（交运算），410
Interval scale（区间标度），28-29
Inverse-document-frequency (IDF)（文档频率倒数），463
Inverted file（倒排文件），461
IR（信息检索）。参见 Text retrieval
ISODATA algorithm（ISODATA 算法），307
Itemsets, frequent（项集，频繁~），429-433
Iteratively weighted least square method（迭代加权最小二乘法）258-389

J

Jaccard coefficient（Jaccard 系数），37
Jackknife methods（Jackknife 方法），360-361
Jeffrey's prior（Jeffrey 先验），122
Join operations（联接运算），412
Joint density function（联合密度函数），97-98
Joint distributions（联合分布）
for categorical data（用于范畴型数据的），287-292
for unordered categorical data（用于无序范畴型数据的），187

K

K - means algorithms（*K*-均值算法），298，305
k - nearest neighbor method（*k*-最近邻算法），348-349
Kalman filters（Kalman 滤波器），201-202
KDD（数据库知识发现），3
Kernel density method（核密度估计），284
Kernel estimates（核估计），59-62，176
Kernel function（核函数），285
Kernel methods，176-178
Kernel models（核方法），287
Kernel plots（核曲线），61
Keyword spotting（关键字命中），479
Knowledge discovery in databases (KDD)（数据库知识发现），3
Kolmogorov-Smirnov test statistic（Kolmogorov-Smirnov 检验统计量），129-130
*k*th mixing proportion（第 k 个混合比例），281
*k*th-order Markov model（k 阶马尔可夫模型），200

Kuhn-Tucker conditions（Kuhn-Tucker 条件），260

L

Lagrange multipliers（拉格朗日乘子），259-260
Laplace approximation（拉普拉斯近似），323
Latent semantic indexing (LSI)（隐含语义索引），465-469
Latent variables（隐含变量），187，190-191，195
Least squares fitting（最小二乘拟合）
  computational issues in（~中的计算问题），370-372
  defined（定义），370
  diagnostic methods and（诊断方法），381-384
  generalization and（推广），377-378
  inference and（推理），377-378
  interpreting（解释），375-377
  model building and（建模），378-381
  model inspection and（模型审查和），381-384
Least squares method（最小二乘法），114，211，370
Leaving-one-out method（留一法），360
Lee, M.（Lee, M），425
Left-skewed distributions（左倾斜分布），57
Length variables（长度变量），32
Letters（字母），206。参见 string
Likelihood function（似然函数），105，108-109，274-275
Likelihood ratio（似然率），125-126
Linear algebra methods（线性代数方法），249-250
Linear correlation（线性相关），35
Linear covariance（线性协方差），35
Linear dependencies（线性依赖），35
Linear discriminants（线性判别式），341-343
Linear function（线性函数），9
Linear models（线性模型）
  background information（背景信息），368-370
  diagnostic methods and（诊断方法），381-384
  generalization and（泛化），377-378
  generalized（推广的~），384-390
  global（全局），478
  inference and（推理），377-378
  inspection（审查），381-384
  model building and（建模），378-381
  probabilistic interpretation of（~的概率解释），372-375
Linear predictor（线性预报量），388
Linear programming（线性规划），259
Linear regression models（线性回归模型），参见 Linear models
Linear structure, regression models with（线性结构，具有~的回归模型），169-173
Local exploration（局部探索），243
Local extremum , finding（局部极值，寻找），251
Local improvement（局部改善），241
Local part of image（图像的局部），166
Local piecewise model structures for regression（用于回归的局部分段模型结构），174-175
Locally linear（局部线性），174
Locally weighted regression model（局部加权回归模型），175-176
Location measurements（位置测量），55
Location parameters（位置参数），184
Loess regression model（Loess 回归方法），175-176
Log-likelihood（对数似然），122，274-275
Log-linear models（对数线性模型），292
Logistic discriminant analysis（logistic 判别式分析），352-353
Logistic link function（logistic 连接函数），385
Logistic regression（logistic 回归），384-385
Logit link function（对数连接函数），385
Logit transformation（对数变换），40
"Lower resolution" data samples 11
LSI（"较低分辨率" 数据样本），465-469
Luck（运气），93-97。参见 Uncertainty

M

Mahalanobis distance（马氏距离），276-277
Manhattan metric（曼哈顿标距），36
Manipulation of variables（操纵变量），168
MAP method（最大化后验法），117，226，283，291
Marginal density（边缘密度），98
Marginal likelihoods（边缘似然），130，226
Market-basket data（市场-购物篮数据），158，429-430
Markov chain model（马尔可夫链模型），189-190，202，290
Markov Chain Monte Carlo (MCMC) methods（马尔可夫链蒙特卡罗方法），123，268
Markov linear-switching model（马尔可夫线性切换模型）

479-480
Markov random fields（马尔可夫随机场），202
Massive data sets（海量数据集），421-426
Mathematical programming（数学规划），259
Maximal predictive classification（最可能预测分类），301
Maximum likelihood estimation（最大似然估计），93，106，108-116
Maximum likelihood estimator (MLE)（最大似然估计量），109，113
Maximum a posteriori (MAP) method（最大化后验法），117，226，283，291
Maximum variability in data（数据中的最大变化性），77
MCMC methods（MCMC 方法），123，268
MDL method（MDL 方法），226
Mean squared error (MSE)（均方误差），107，223-224
Measurements（测量）。参见 Data
  accurate（精度），45-46
  amounts and（数量和~），31
  background information（背景信息），25-26
  balances and（余额），31
  bias of（~的偏差），45
  counted fractions and（计份额），31
  counts versus（计数相对于~），31
  dispersion（离差），56
  distance（距离），31-38
  grades and（分级），31
  individual data quality for（数据个体的质量），44-47
  location（位置），55
  metrical versus categorical（标距型测量和范畴型测量），31
  pairs of（~对），327
  precise（精确的），45
  qualitative versus quantitative（定性的和定量的），31
  ranks and（排位和~），31
  reliability of（~可靠性），46
  representational（表示性的），29-31
  summary information（归纳），52
  types of（~的类型），26-31
  validity of（~的有效性），46-47
  variability（变化性），56
Median（中值），55
Memory hierarchy（存储器层次），400-401
Minimum description length (MDL) method（最短描述长度方法），226
Minimum distance（最短距离），298
Minkowski metric（闵可夫斯基标距），36
Missing data, optimization with（残缺数据，存在~时的优化），260-265
Mixture distributions and densities（混合分布和密度），279-281
Mixture models（混合模型）
  autoregressive models（自回归模型），202
  parametric（参数~），185-187
  probabilistic model-based clustering using（利用~的基于模型概率聚类），315-323
    advantages（优点），319
    background information（背景信息），315-316
    disadvantages（缺点），319-321
    examples（例子），316-319
    techniques（技术），321-323
  and radial basis function approaches（~和径向基函数方法），357
MLE（最大似然估计量），109，113
MLP（前馈多层感知器），153-157，357，391
Mode（最频值），56
Model averaging methods（模型平均方法），346
Models（模型），参见 Complexity of models; Patterns; *specific types*
  background information（背景信息），165-167
  building（建立），378-381
  classes of structure（各类结构），235，238
  curse of dimensionality and（维度效应），193-196
  data（数据），405
  data mining and（数据挖掘和），1-2，10-11，175，271
  defined（定义），165
  deviance of（~偏离度），389-390
  evaluation of（~的评估），229-231
  expressive power of（~的表达力），183
  fundamentals（基础），167-168
  generalized linear（推广的线性），173，353，384-390
  generative（产生式），272
  global（全局），442-443，478-480

goal of（~的目标），102
for individual preferences（用于对个人爱好建模的），470-472
inspection of（~的审查），381-384
*k*th order Markov（*k* 阶马尔可夫），200
Markov chain（马尔可夫链），189-190，202，290
parameters of（~的参数），167，276
for prediction（用于预测的），168-183
background information（背景信息），168-169
local piecewise model structures for regression（用于回归的局部分段模型结构），174-175
nonparametric " memory-based " local models(非参数的"基于记忆" 局部模型），175-178
regression models with linear structure（具有线性结构的回归模型），169-173
selecting, of appropriate complexity（选择，合适的复杂度），183
stochastic components of（~的随机分量），178-180
for probability distributions and density（用于概率分布和密度的），184-193
background information（背景信息），184
concepts, general（概念，一般），184-185
factorization and independence in high dimensions（高维中的因式分解和独立性），187-193
joint distributions for unordered categorical data（无序范畴型数据的联合分布），187
mixtures of（~的混合），185-187
search methods for（搜索~的方法），238-241，378-381
background information（背景信息），238-241
branch-and-bound（分枝定界），246-247
heuristic search（启发式搜索），244-246
simple greedy search algorithm（简单的贪婪搜索算法），243-244
state-space formulation（状态空间搜索形式），241-243
systematic search（系统搜索），244-246
for structured data（用于结构数据），197-203
Momentum-based methods（基于冲量的方法），254
Monothetic divisive methods（单分裂方法），315
Monotonic regression（单调回归），87
Monte Carlo Markov Chain (MCMC) methods（Monte Carlo 马尔可夫方法），123，268
Monte Carlo sampling techniques（Monte Carlo 抽样技术），123，226
Morse codes（莫尔斯代码），85
MSE（MSE），107，223-224
Multicollinearity（多重共线性），371
Multidimensional indexing（多维索引），404-405
Multidimensional scaling（多维缩放），84-90
Multidimensional scaling plot（多维缩放曲线），88
Multilayer perceptrons (MLPs)（多层感知器），153-157，357，391
Multimodal distributions（多峰型分布），56，60
Multinomial distributions（多项分布），487-488
Multiple regression（多重回归），368-369
Multirelational data（多重关系数据），42-43
Multivariate function（多元函数），113-114
Multivariate gradient descent method（多元梯度下降方法），256
Multivariate normal distributions（多元正态分布），490
Multivariate parameter optimization（多元参数优化），255-259
Multivariate random variables（多元随机变量），97-102

N

Naive Bayes model（朴素贝叶斯模型），353-356
NASA Earth Observing System（NASA 地球观测系统），19
Natural language processing (NLP)（自然语言理解），457
Natural parameter（自然参数），388
Nearest neighbor methods（最近邻方法）
agglomerative methods and（凝聚方法和），312-313
condensed（压缩的），352
edited（改进的），352
nonparametric " memory-based " local models and（"基于记忆" 的非参数局部模型），176，178
pairwise distances of the members of each cluster and（聚类的成员的两两距离），312-313
parametric models and（参数模型和），351
predictive models for classification and（用于分类的预测模型和），347-352
reduced（简化的），352
Nelder and Mead variant（Nelder 和 Mead 变体），259
Nesting（嵌套），172
Neural networks（神经网络），173
Newton-Raphson (NR) method（Newton-Raphson（NR），方

法），252-253，255，389
Newton's method（Newton 方法），256-257
NIST（NIST），456
NLP（NLP），457
Nominal scales（标称标度），28，31
Non-metric multidimensional scaling（非标距多维缩放），87
Nonlinear function（非线性函数），10，154
Nonlinear global models（非线性全局模型），478-479
Nonparametric density estimation（非参数密度估计），284-287
Nonparametric "memory-based" local models（"基于记忆"的非参数局部模型），175-178
Nonparametric models（非参数模型），185
Nonparametric test（非参数检验），130
Nonprobabilistic descriptive models（非概率描述模型），219
Nonrepresentational procedures（非表示性过程），30
Nonscalable versions of data mining algorithms（数据挖掘算法的非伸缩版本），424
Nonsystematic variation（非系统性变化），179-180
Normal density（正态密度），197，355
Normal distribution（正态分布），60，113，115-116，118，121-122，127，171，276，350，488
Normal posterior，122-123
Normal prior（正态后验），122-123
NR method（NR 方法），252-253，255，389
Null hypothesis（零假设），124-126
Numerical scales（数字标度），31

O

Objects（对象），4
Observational data（观察到的数据），1
Odds ratio（赔率），352-353
OLAP（OLAP），417-419
OLTP（OLTP），417-419
One-tailed test（单边检验），125
Online algorithms（在线算法），265-266
Online analytical processing (OLAP)（在线分析处理），417-419
Online approximation（在线近似），265
Online transaction processing (OLTP)（在线事务处理），417-419
Operational databases（业务数据库），417
Operational procedures（操作性过程），30
Opportunity samples（机会样本），21，48
Optimization（优化）
  background information（背景信息），235-238
  combinatorial（组合），236-239
  as component of data mining algorithms（作为数据挖掘算法的组件），16-17，142-143
  constrained（约束），259-260
  Expectation Maximization algorithm and（期望最大化算法和），260-265
  maximum likelihood estimation and（最大似然估计和），114
  with missing data（存在残缺数据时的），260-265
  online algorithm and（在线算法和），265-266
  parameter optimization methods（参数优化方法），247-260
    background information（背景信息），247-249
    closed form（闭合形式），249-250
    constrained（约束），259-260
    gradient-based（基于梯度的），250-251
    linear algebra（线性代数），249-250
    multivariate（多元），255-259
    univariate（一元），251-255
  query（查询），415-417
  single-scan algorithms and（单扫描算法和），265-266
  stochastic（随机），266-268
Ordinal scales（顺序标度），28，31
Organization of data（数据的组织），参见 Databases
Orthogonality of data（数据的正交性），240
"Out-of-sample" data（"样本外"数据），227，328，372
Overestimation（高估），216
Overfitting（过度拟合），19，183，223

P

p-dimensional space（*p*-维空间），10，12，165，180，277，479
p-dimensional vector（*p*-维向量），9，36，174，329-330，399
PageRank（PageRank），15
Pairs of measurements（测量对），327
Pairwise distance（两两距离），312
Parallel coordinates plots（平行坐标图），74，76
Parameter optimization methods（参数优化方法）

background information（背景信息），247-249
closed form（闭合形式），249-250
constrained（约束），259-260
gradient-based（基于梯度的），250-251
linear algebra（线性代数），249-250
multivariate（多元），255-259
univariate（一元），251-255
Parameters（参数）
algorithm（算法），267
canonical（正规），388
defined（定义），47
dispersion（离差），388
estimation（估计），240
linear function of（~的线性函数），9
location（位置），184
of models（模型的），167，276
natural（自然），388
regression model（回归模型），173
scale（范围），184，388
Parametric models（参数模型）
density（密度），275-279
mixtures of（~的混合），185-187
nearest neighbor methods and（最近邻方法和），351
overview（概览），184
Parents of variables（变量的双亲），189
Partition-based clustering algorithms（基于划分的聚类算法）
background information（背景信息），296-297
basic algorithms for（基本算法），302-308
for nonprobabilistic descriptive models（用于非概率描述模型的），219
score functions for（~使用的评分函数），296-302
Pattern search（模式搜索），259
Patterns（模式）。参见 Models
background information（背景信息），165-167
class of（~类），204
co-occurrence（同现），158
coverage of（~的覆盖面），214
in data matrices（数据矩阵中的），203-206
data mining and（数据挖掘和），1-2，10-11，271
defined（定义），165
detection of（~的探测），102
discovering（发现），13-14，438-441
discovery task（发现任务），205
evaluation of（~的评估），229-231
finding（寻找），427-448
association rules（关联规则），433-435
background information（背景信息），427-428
episodes from sequences（从序列中~片段），436-438
from local patterns to global models（从局部模式到全局模型），442-443
generalizations（推广），435-436
itemsets, frequent（项集，频繁的），429-433
predictive rule induction and（预测规则归纳），443-447
rule representations（规则表示），428-429
selective discovery（选择发现的），438-441
global（全局），9
local, to global models（局部，到全局模型），442-443
primitive（元），204
Q（Q），450，454
scoring（评分），212-215
search methods for（搜索~的方法），238-241，378-381
background information（背景信息），238-241
branch-and-bound（分枝定界），246-247
heuristic search（启发式搜索），241，244-246
simple greedy search algorithm（简单贪婪搜索算法），243-244
state-space formulation（状态空间形式），241-243
systematic search（系统搜索），244-246
for strings（针对字符串的），206-208
structure of（~结构），158
structures（结构），203-208
in data matrices（数据矩阵中的），203-206
for strings（针对字符串的），206-208
text retrieval（文本检索），14
PCA（主分量分析）。参见 Principal components analysis
Penalized likelihood（惩罚似然），321-322
Percentiles（百分位点），56
Perceptrons（感知器），153-157，339-341，357，391
Permutation tests（置换检验），129
Piecewise model structures for regression（用于回归的分段模型

结构），174-175，182
Point estimates（点估计），115，119
Poisson distributions（泊松分布），280-281，388，488
Poisson regression（泊松回归），388
Polysemy（一词多义），457
Polythetic divisive methods（多分裂），315
Population drift（总体漂移），49
Position, sequential（位置，序列），477
Posterior distributions（后验分布），117
Precise functional form（精确的函数形式），176
Precise measurement（精确测量），45
Precision（精度/查准率），121，453-456
Predicted intervals（预测区间），374-375
Predictive distributions（预测分布），120-121
Predictive models（预测模型）
background information（背景信息），168-169
for classification（用于分类的），327-366
classification models and（分类模型和），329-339
evaluating and comparing（评估和比较），359-362
feature selection for high dimension（针对高维的特征选择），362-363
linear discriminants and logistic discriminant analysis（线性判别式分析），341-343，352-353
naive Bayes model（朴素贝叶斯方法），353-356
nearest neighbor methods（最近邻方法），347-352
other methods（其他方法），356-359
overview（概览），180-182，327-329
perceptrons and（感知器和），339-341
tree models（树模型），343-347
examples of（~的例子），14
goal of（~的目标），13
local piecewise model structures for regression（用于回归的局部分段模型结构），174-175
nonparametric " memory-based" local models（"基于记忆"的非参数局部模型），175-178
for regression（用于回归的），367-398
artificial neural networks（人工神经网络），391-393
background information（背景信息），367-368
generalized linear models（推广的线性模型），384-390
least squares fitting）（最小二乘拟合），368-384
Linear models（线性模型），368-384
other highly parameterized models（其他高度参数化的模型），393-397
regression models with linear structure（具有线性结构的回归模型），169-173
score functions for（~使用的评分函数），212，215-217
selecting, of appropriate complexity（选择，具有合适复杂度的），183
stochastic components of（~的随机分量），178-180
Predictive performance（预测性能），196
Predictive rule induction（预测规则归纳），443-447
Predictor variables（预报变量），168，367
PREFERENCE property（PREFERENCE 属性），27
Preferences, modeling individual（偏好，对个人~建模）470-472
PRIM algorithms（PRIM 算法），445-446
Primitive patterns（元模式），204
Principal components（主分量），195
Principal components analysis (PCA)（主分量分析）
graphical data exploration methods and（图形化的数据探索方法），74-84
high-dimensional data and（高维数据），196
Principal coordinates method（主坐标方法），86
Prior distributions（先验分布），117
Priors（先验），122-123
Probabilistic model-based clustering using mixture models（利用混合模型的基于模型概率聚类）
advantages（优点），319
background information（背景信息），315-316
disadvantages（缺点），319-321
examples（例子），316-319
techniques（技术），321-323
Probabilistic models for classification（用于分类的概率模型）331-334
Probabilistic rule（概率规则），213-214，428
Probability（概率），93-97
Probability calculus（概率计算），94-96
Probability distribution and density function models（概率分布和密度函数模型）
background information（背景信息），184

concepts, general（概念，一般），184-185
descriptive models and（描述模型）
background information（背景信息），272-274
Expectation Maximization algorithm for（用于~的期望最大化算法），281-284
joint distributions for categorical data（用于范畴型数据的联合分布），287-292
mixture distributions and densities（混合分布和密度），279-281
nonparametric density estimation（非参数密度估计），284-287
parametric density models（参数密度估计），275-279
score functions for（用于~的评分函数），274-275
estimation（估计），274-275
factorization and independence in high dimensions（高维中的因式分解和独立性），187-193
joint distributions for unordered categorical data（针对无序范畴型数据的联合分布），187
mixtures of（~的混合），185-187
Probability distributions（概率分布），485，487-490
Probability mass function（概率质量函数），485
Probability theory（概率论），94-95
Projection operation（投影运算），411
Projection pursuit methods（投影追踪方法），77，195-196，357，395-397
Proximity（邻近度），32
Pruning（修剪），153，159
Pseudo data sets（伪数据集），425-426

Q

QBIC（根据图像内容查询），15，474
Quadratic discriminant function（二次判别函数），343
Quadratic error function（二次误差函数），340
Quadratic function（抛物线函数），249
Quadratic programming（二次规划），259
Quality of data（数据质量）
for collection of data（数据群体的），47-51
for individual measurements（单个测量的），44-47
poor（低劣的），51
QUALITY OF LIFE property（QUANLITY OF LIFE（生活质量）属性），29
Quantitative variables（定量变量），6
Quartiles（四分位点），56
Quasi-likelihood methods（准-似然方法），180
Quasi-Newton methods（准-Newton 方法），257-258
Quasilikelihood estimation（准似然估计），390
Query（查询）
aggregation in,（聚合），414
execution（执行），415-417
image（图像），473-474
matching（匹配），461-465
optimization（优化），415-417
pattern Q（~模式 Q），450，454
rectangular range（矩形区域），404
selectivity estimation（选择力估计），273
Structured Query Language（结构化查询语言），409，413-415
text（文本），456-457
Query by Image Content (QBIC)（根据图像内容查询），15，474

R

Radial basis function networks（径向基函数网络），393
RAM（RAM），17
Random samples（随机样本），20，54，123
Random variables（随机变量），97-102，485-490
Random variation（随机变化），179-180
Random-access memory (RAM)（随机访问存储器），17
Randomization tests（随机检验），129
Randomness（随机性），93-97。参见 Uncertainty
Range（值域），56，404
Ranks（分等），31
Ratio scales（比例标度），28
Recall（查全率），453-456
Receiver Operating Characteristic (ROC) curve（接受者操作特性曲线），361，454
Reciprocals of variances（方差的倒数），121
Records（记录），4
Rectangular range query（矩形区域查询），404
Reduced nearest neighbor methods（简化的最近邻方法），352
Reductionist viewpoint on data mining algorithms（数据挖掘算

法的化约主义观点）
A Prior algorithm（A Prior 算法），157-160
background information（背景信息），151-153
multilayer perceptrons for regression and classification and（用于回归和分类的多层感知器），153-157
vector-space for text retrieval and（用于文本检索的向量空间方法），160-162
Redundant variables（冗余变量），194
Reference prior（参考先验），122
Regression（回归）
approach（途径），335-339
defined（定义），169，328-329
estimation and（估计和），13
line（直线），368
linear, probabilistic interpretation of（线性，~的概率解释），372-375
local piecewise model structures for（用于~的局部分段模型结构），174-175
locally weighted model（局部加权模型），175-176
loess model（loess 模型），175-176
logistic（logistic），384-385
methods（方法），348
models with linear structure（具有线性结构的模型），169-173
monotonic（单调），87
multilayer perceptrons for（用于~的多层感知器），153-157
multiple（多重），368-369
plane（平面），368-369
Poisson（泊松），388
predictive models for（用于~的预测模型），367-398
artificial neural networks（人工神经网络），391-393
background information（背景信息），367-368
generalized linear models（推广的线性模型），384-390
least squares fitting（最小二乘拟合），368-384
linear models（线性模型），368-384
other highly parameterized models（其他高度参数化的模型），393-397
Projection pursuit（投影追踪），195-197，395-397
rule-based（基于规则的），446
simple（简单的），368
sum of squares（平方和），376
Regular expression E（正则表达式 E），207
Regular grammars（正则语法），202
Regularities（规律），134
Regularized discriminant analysis（正则化判别式分析），343
Reject option（否决选项），350
Rejection region（拒绝区），125
Relation schema（关系模式），405
Relational algebra（关系代数），409
Relational data model（关系数据模型），405
Relational databases（关系数据库），405-409
Relations（关系），405
Relative distributions（相对分布），459
Relevance feedback（相关反馈），462，470-471
Reliability of measurements（测量的可靠性），46
Repeated measures data（重复测量数据），349-350
Representational measurements（表示性测量），29-31
Resampling techniques（二次采样技术），322
Residual sum of squares（残差平方和），376
Residuals（残差），369
Response variable（响应变量），168，367
Resubstitution error rate（重新代入误差率），359
Retesting, effective（重复测试，有效的），46
Retrieval by content（根据内容检索）
applications of（~的应用），15
background information（背景信息），449-452
evaluation of systems（系统评估），452-456
goal of（~的目标），14
image retrieval（图像检索），472-476
sequence retrieval（序列检索），476-481
summary information（归纳信息），481-482
for text（针对文本的），456-470
background information（背景信息），456-457
classification of document and text（文档和文本分类），469-470
latent semantic indexing（隐含语义索引），465-469
matching queries and documents（匹配查询和文档），461-465
patterns（模式），14
representation of text（文本的表示），457-461

time series（时间序列），476-481
Right-skewed distributions（右倾斜分布），57
Risk of error（误差的风险），45
Robust methods（鲁棒方法），231-232
ROC curve（ROC 曲线），361，454
Rocchio's algorithm（Rocchio 算法），470
Root node（根结点），244-245
Rotations, random（旋转，随机），71
Rothamsted Experimental Station（英国洛桑实验站），11-12
Rows（行），36
Rules（规则）
  discovering（发现），13-14，438-441
  finding（寻找）
    association rules（关联规则），433-435
    background information（背景信息），427-428
    episodes from sequences and（从序列中~片段），436-438
    from local patterns to global models（从局部模式到全局模型），442-443
    generalizations（推广），435-436
    itemsets, frequent（项集，频繁的），429-433
    predictive rule induction and（预测规则的归纳），443-447
    rule representations（规则表示），428-429
    selective discovery of（选择发现的~），438-441
  probabilistic（概率），213-214，428
  regression based on（基于~的回归），446
  representations of（~的表示），428-429
  set of（~集合），443
  structure of（~的结构），158

S

Sample correlation coefficient（样本相关系数），35
Sample covariance（样本协方差），35
Sample mean（样本均值），33，35
Sample-based estimate of sample mean（对样本均值的基于样本估计），55
Samples（样本），7。参见 Data set
  convenience（顺便），21，48
  data mining and（数据挖掘），93
  distortion of（~的失真），49-50
  epsem（Epsem），134
  "lower resolution" data（"低分辨率" 数据），11
  opportunity（机会），21，48
  random（随机），20，54，123
  systematic（系统），133-134
  uncertainty and（不确定性），102-105
Sampling fraction（采样率），133
Sampling methods（采样方法），132-138，338
Sampling paradigm（采样模式），128
Scalable versions of data mining algorithms（数据挖掘算法的可伸缩版本），423-424
Scale parameter（范围参数），184，388
Scales（标度），28-29，31
Scatterplot matrix，71-72
Scatterplots（散点图），64-65
Schemas（图式），41-44，405，410
Score functions for data mining algorithms（数据挖掘算法的评分函数）
  background information（背景信息），211-212
  decomposable（可分解的），240
  defined（定义），211，235
  descriptive（描述结构），212，217-219
  with different complexities（针对不同复杂度的），220-228
    bias-variance（偏差-方差），221-224
    concept in comparing, general（比较模型的一般概念），220-221
    penalizing（惩罚），224-227
    validation and, external（验证，外部），227-228
  evaluating（评估），229-231
  function of（~的函数），142
  generic（通用的），16，219
  for partition-based clustering algorithms（用于基于划分聚类算法的），296-302
  patterns, scoring（模式，评分），212-215
  predictive（预测），212，215-217
  for probability distribution and density function models, estimating（用于概率分布和密度函数模型的，估计），274-275
  robust methods（鲁棒方法），231-232
  scoring method versus（评分方法对~），389

Scoring method（评分方法）
complexity of a model and（模型的复杂度和），220-228
bias-variance（偏差-方差），221-224
concepts in comparing, general（比较模型的一般概念），220-221
penalizing（惩罚），224-227
validation and, external（验证和，外部的），227-228
score functions versus（评分函数对~），389
Scree plots（碎石堆图），79-80
Search methods（搜索方法）
background information（背景信息），235-238
blind（盲目），245-246
branch-and-bound（分枝定界），246-247
breadth-first（广度优先），245
as component of data mining algorithms（作为数据挖掘算法的一个组件），16-17，142-143
depth-first（深度优先），245
genetic（通用的），266-267
greedy heuristic（贪婪启发式），241
heuristic（启发式），241，244-246，439-440
for models and patterns（搜索模型和模式），238-241，378-381
simple greedy search algorithm（简单贪婪搜索算法），243-244
state-space formulation（状态空间形式），241-243
stochastic（随机），266-268
systematic（系统），244-246
Search operators（搜索算子），241-242
Search tree（搜索树），244-245，402
Segmentation（区隔），12，293
Select clause（SELECT 子句），413
Selection operation（选择运算），411
Selectivity（选择能力），273
Sequence retrieval（序列检索），476-481
Sequences, episodes from（序列，~中的片段），436-438
Sequential data（序列数据），477
Sequential position（序列位置），477
Set operations（集合运算），410
Set of rules（规则集），443
SEVERITY property（SEVERITY（严重性）属性），27
Severity scale（严重性标度），28
Significance level（显著水平），105，125
Similarity（相似性），15，449，451，480
Simple greedy search algorithm（简单贪婪搜索），243-244
Simple regression models（简单回归模型），368
Simplex algorithm（单纯形算法），258
Simplex search method（单纯形搜索方法），258
Simpson's paradox（辛普森悖论），100-101
Simulated annealing（模拟退火），267-268
Simultaneous test procedures（同步检验过程），131
Single link method（单链接方法），312-313
Single-link criterion（单链接标准），298
Single-scan algorithm（单扫描算法），265
Singular-value decomposition (SVD)（奇异值分解），415，466
Skewness（倾斜度），56-57
SKICAT system（SKICAT 系统），13
Sloan Digital Sky Survey（Sloan 天体数字化调查），19
Snooping（探查），22-23
Spatial data（空间数据），44
Special-purpose algorithms for disk access（考虑磁盘访问的有针对性算法），424
Spline function（样条函数），174
Splines（样条），174-175
Splitting a node（分裂节点），344-345
SQL（结构化查询语言），409，413-415
Squared error（误差平方），216
SRM approach（SRM 方法），226
SSE（误差平方和），155-156，235
Standard data（标准数据），41
Standard deviation（标准差），56，60
Standardization（标准化），38
Star icons（星图标），74
Star plot（星图），75
State space representation（状态空间表示），241
State variables（状态变量），200-201
State-space formulation for search methods（搜索方法的状态空间表示），241-243
Stationarity（平稳），198-199
Statistical inference（统计推理），102-105
Statistics（统计），18-21，47，425-426
Stepwise model（分步模型），130
Stochastic approximation（随机近似），265

Stochastic components of model structures（模型结构的随机分量），178-180
Stochastic estimation（随机估计），123，265
Stochastic search methods（随机搜索方法），266-268
Strategic databases（策略数据库），417
Stratified random sampling（分层随机采样），135
Strings（字符串），43，206-208，420-421
Structural risk minimization (SRM) approach（结构风险最小化），226
Structured data models（结构化数据模型），197-203
Structured Query Language (SQL)（结构化查询语言），409，413-415
Structures, data mining（结构，数据挖掘），9-11，142
Student's*t*- distributions（学生氏分布），489
Subsamples（子样本），360
Subsets problem（子集问题），241
"Sufficient statistic" concept（"充分统计量"概念），112-113
Sufficient statistics（充分统计量），19-20，425-426
Suffix tree data structure（后缀树数据结构），421
Sum of squared errors (SSE)（误差平方和），155-156，235
Sum of squared residuals（残差平方和），376
Summarizing data（总结数据），54-57
Supervised classification（有指导分类），169，328-329
Support（支持度），430
Support vector machines（支持向量机），357
Surrogate document（代理文档），461
Suspect data（可疑数据），50-51
SVD（奇异值分解），415，466
Synonmy（同义词），457
Systematic sampling（系统采样），133-134
Systematic search methods（系统搜索方法），244-246
Systematic variation（系统变化），179

T

T-dimensional "term space"（*T*-维"词条向量"），461
Tables（表），41，188，408-412
Tasks, data mining（任务，数据挖掘），11-15，142
Taylor series（泰勒级数），227，257，369
Temperature schedule（温度调度表），267
Ten-fold cross-validation（十折交叉验证），322
Term（词条），456
Term frequency (TF)（词条频率），463
Test set（检验集合），360
Text retrieval（文本检索）
  background information（背景信息），456-457
  classification of document and text（文档和文本的分类），469-470
  latent semantic indexing（潜在语义索引），465-469
  matching queries and documents（匹配查询和文档），461-465
  patterns（模式），14
  representation of text（文本的表示），457-461
Text retrieval Conferences (TREC)（文本检索会议），456
TF（词条频率），463
Time series data（时间序列数据），476-481
Total sum of squares（总平方和），376
Training data（训练数据），7。参见 Data set
Training data points（训练数据点），346
Transactions（事务（交易）），405-406
Transforming data（数据转化），38-41，195-196，363
TREC（文本检索大会），456
Tree models（树模型），174，343-347
Tree-structured rule sets（树结构规则集），443
Trellis plotting（格架图），71，73-74
Trimmed mean（修整均值），231-232
True error rate（真实误差率），359
True value concept（真实值概念），45
Tuple, algorithm（组合，算法），146，151，154-155

U

Unbiased estimation（无偏估计）106，227
Uncertainty（不确定性）
  background information（背景信息），93
  dealing with（处理），94-97
  estimation and（估计和），105-124
    background information（背景信息），105-106
    Bayesian（贝叶斯），93，116-124
    maximum likelihood and（最大似然和~），93，108-116
    properties of estimators and, desirable（理想估计量的属性），106-108
    stochastic（随机），123，265

hypothesis testing and（假设检验和~），124-132
background information（背景信息），124
classical（经典的），124-130
in context（数据挖掘中的），130-132
multivariate random variables and（多元随机变量和~），97-102
probability and（概率和~），93-97
random variables and（随机变量和~），97-102
samples and（样本和~），102-105
sampling method and（采样方法和~），132-138
statistical inference and（统计推理和~），102-105
summary information（归纳信息），138
Underestimation（低估），216
Unexplainable variation（不可解释的变化），179-180
Union operation（并运算），410
Univariate parameter optimization（一元参数优化），251-255
Univariate random variables（一元随机变量），485-487
Universal table（大全表），408
Unordered categorical data, joint distributions for（无序范畴型数据，~的联合分布），187
U. S. National Institute of Standards and Technology (NIST)（美国国家标准技术研究所），456

V

Validation（验证），227-228
Validation log-likelihood（验证对数似然），275
Validation subset（验证子集），148-149
Validity of measurements（测量的有效性），46-47
Variability measurements（变化性尺度），56
Variables（变量）
categorical（范畴型），6
class（分类），329
conditionally independent（条件独立），99-100，289，354
defined（定义），4
dependent（依赖），35
derived（导出），198
explanatory（解释），168
frequent sets of（~的频繁集），204
hidden（隐藏），187，190-191，195
independent（独立），99，188-189
individual X（单个~X），194-195
input（输入），329
latent（隐藏），187，190-191，195
length（长度），32
linear dependencies between（~间的线性依赖），35
manipulating（操纵），168
multivariate（多元），97-102
parents of（~的双亲），189
predictor（预报），168，367
quantitative（定量的），6
random（随机），97-102，485-490
redundant（冗余），194
response（响应），168，367
selecting（选择），362-363
selection for high-dimensional data（针对高维数据的~选择），194-195
state（状态），200-201
transforming（转化），363
univariate random（一元随机），485-487
visual techniques for displaying（显示~的可视化技术）
more than two（两个以上），70-74
relationships between two（两个~间的关系），62-70
single（单个），57-62
weight（权），32
Variance function（方差函数），388
Variances（方差），56，78，121，221-224
Variations（变化），297-298
Vector space representation（向量空间表示），458
Vector-space algorithms（向量空间算法），160-162
Visual techniques（可视化技术）
data mining and（数据挖掘和），11
for more than two variables（用于两个以上变量的~），70-74
for relationships between two variables（用于显示两个变量间关系的~），62-70
for single variables（用于单个变量的~），57-62

W

Warehousing, data（数据仓库），417-419
WEIGHT property（WEIGHT（重量）属性），26-28
Weight variables（weight（重量）变量），32

Weighted Euclidean distance（加权欧氏距离），33
Weighted least squares solution（加权最小二乘解），382
Where clause（Where 子句），413
Whole image（整幅图像），166
Wilcoxon test statistic（Wilcoxon 检验统计量），129-130
Within-cluster sum- of- squares（聚类内平方和），298
Within-cluster variation（聚类内变化），297-298

Z

Zero skewness（零倾斜度），57



华章教育 未诚奉献

计算机科学丛书

计算机文化（原书第4版）	Parsons /龚波/50.00
离散数学及其应用（原书第4版）	Rosen/袁崇义/75.00
组合数学（原书第3版）	Brualdi/冯舜玺/38.00
程序设计实践	Kernighan/裘宗燕/20.00
程序设计语言概念和结构（原书第2版）	Sethi/裘宗燕/45.00
编码的奥秘	Petzold/伍卫国/24.00
C语言解析教程（原书第4版）	Pohl/麻志毅/48.00
C程序设计教程	Deitel/薛万鹏/33.00
C程序设计语言	Kernighan/徐宝文/28.00
C++程序设计语言（特别版）	Stroustrup/裘宗燕/78.00
《C++程序设计语言》题解	Vandevoorde/裘宗燕/23.00
C++面向对象开发（原书第2版）	Lee/麻志毅/45.00
C++编程思想（原书第2版）	Eckel/刘宗田/59.00
C++精髓：软件工程方法	Shtern/李师贤/85.00
C++语言的设计和演化	Stroustrup/裘宗燕/48.00
C++程序设计教程	Deitel/薛万鹏/22.00
C++编程思想（原书第1版）	Eckel/袁兆山/39.00
Java语言导学（原书第3版）	Campione/马朝晖/59.00
Java编程思想（原书第2版）	Eckel/侯捷/99.00
Java编程思想（原书第1版）	Eckel/京京工作室/60.00
Java程序设计教程（原书第3版）上册（附光盘）	Deitel/袁兆山/55.00
Java程序设计教程（原书第3版）下册（附光盘）	Deitel/袁兆山/69.00
Visual Basic.NET 程序设计专家指南（原书第2版）	Deitel/龚波/118.00
面向对象程序设计——Java语言描述（附光盘）	Kalin/孙艳春/55.00
面向对象程序设计——C++语言描述	Johnsonbaugh/谢君英/48.00
标准C++与面向对象程序设计	Wang /李健/39.00
Java面向对象程序设计教程	Kafura/袁晓华/即将出版
面向对象程序设计——图形应用实例	Laszlo/何玉洁/35.00
数据结构，算法与应用——C++语言描述	Sahni/王广芳/49.00
编译原理及实践	Louden/冯博琴/39.00
Linux操作系统内核实习（附光盘）	Nutt/陆丽娜/29.00
UNIX操作系统设计	Bach/陈葆钰/35.00
UNIX环境高级编程	Stevens/尤晋元/55.00
UNIX编程环境	Kernighan/陈向群/24.00
现代操作系统	Tanenbaum/陈向群/40.00
并行计算机体系结构（原书第2版）	Culler/李晓明/78.00

结构化计算机组成	Tanenbaum/刘卫东/46.00
可扩展并行计算：技术、结构与编程	Hwang/陆鑫达/49.00
并行程序设计	Wilkinson/陆鑫达/43.00
数据库系统概念（原书第4版）	Silberschatz/杨冬青/即将出版
数据库原理、编程与性能（原书第2版）	O'Neil/周傲英/55.00
数据库设计	Stephens/何玉洁 武欣/35.00
数据库系统导论	Date/孟小峰 王珊/66.00
数据库系统实现	Garcia-Molina/杨冬青/45.00
数据库系统概念	Silberschatz/杨冬青/49.00
数据挖掘：概念与技术	Han/范明 孟小峰/39.00
数据仓库（原书第3版）	Inmon/黄厚宽/即将出版
数据仓库	Inmon/黄厚宽/25.00
信息系统原理	Reynolds/张靖/42.00
计算机信息处理	Mandell/尤晓东/38.00
数据通信与网络（原书第2版）	Forouzan/吴时霖/68.00
神经网络设计	Hagan/戴葵/49.00
高性能通信网络（原书第2版）	Walrand/史美林/55.00
数据广播	Tvede/徐良贤/28.00
ISDN、B-ISDN与帧中继和ATM（原书第4版）	Stallings/程时端/48.00
计算机网络（原书第2版）	Peterson/叶新铭 史美林/49.00
计算机网络实用教程（附光盘）	Dean/陶华敏/65.00
计算机网络实用教程实验手册	Dean/陶华敏/15.00
计算机网络与因特网（附光盘）	Comer/徐良贤/40.00
TCP/IP详解 卷I：协议	Stevens/谢希仁/45.00
TCP/IP详解 卷II：实现	Stevens/谢希仁/78.00
TCP/IP详解 卷III：TCP事务协议、HTTP、NNTP和UNIX域协议	Stevens/谢希仁/35.00
数据通信与网络（第1版）	Forouzan/吴时霖/48.00
数据通信与网络教程	Shay/高传善/40.00
最新网络技术基础	Palmer/严伟/20.00
Internet技术基础	Comer/袁兆山/18.00
分布式操作系统：原理与实践	Galli/徐良贤/38.00
分布式计算的安全原理	Bruce/李如豹/35.00
分布式系统设计	Wu/高传善/30.00
软件工程（原书第6版）	Sommerville/程成/49.00
软件工程：实践者的研究方法（原书第5版）	Pressman/梅宏/59.00
设计模式：可复用面向对象软件的基础	Gamma/吕建/35.00
软件工程：实践者的研究方法（原书第4版）	Pressman/梅宏/48.00
软件工程：Java语言实现	Schach/袁兆山/38.00
计算机图形学的算法基础（原书第2版）	Rogers/石教英/55.00
专家系统原理与编程（附光盘）	Giarratano/印鉴/49.00
人工智能	Nilsson/郑扣根/30.00
机器学习	Mitchell/曾华军/35.00
数字逻辑：应用与设计	Yarbrough/李书浩/49.00
嵌入式计算系统设计原理（附光盘）	Wolf/孙玉芳/65.00
计算理论导引	Sipser/张立昂/30.00

经典原版书库

计算机文化（第4版）	Parsons/55.00
具体数学：计算机科学基础（第2版）	Graham/49.00
组合数学（第3版）	Brualdi/35.00
离散数学及其应用（第5版）	Rosen/79.00
离散数学及其应用（第4版）	Rosen/59.00
程序设计语言原理（第5版）	Sebesta/45.00
程序设计语言 概念和结构（第2版）	Sethi/39.00
程序设计实践	Kernighan/22.00
C++编程思想（第2版）（附光盘）	Eckel/58.00
C++语言的设计和演化	Stroustrup/29.00
Java编程思想（第2版）（附光盘）	Eckel/69.00
数据结构算法与应用——C++语言描述	Sahni/66.00
数据结构与STL	Collins/即将出版
编译原理与实践	Louden/58.00
UNIX操作系统教程	Sarwar/49.00
UNIX环境高级编程	Stevens/39.00
Linux操作系统内核实习	Nutt/25.00
现代操作系统（第2版）	Tanenbaum/48.00
计算机体系结构：量化研究方法（第3版）	Hennessy/99.00
计算机组成（第5版）	Hamacher/48.00
结构化计算机组成（第4版）	Tanenbaum/38.00
并行计算机体系结构	Culler/88.00
计算机体系结构：量化研究方法	Hennessy/88.00
计算机组织与设计：硬件/软件接口	Patterson/80.00
可扩展并行计算：技术、结构与编程	Hwang/69.00
高级计算机体系结构	Hwang/59.00
数据库系统导论（第7版）	Date/65.00
数据库系统实现	Garcia-Molina/42.00
数据库系统概念	Silberschatz/65.00
通信网络基础	Walrand/32.00
Internet技术基础（第3版）	Comer/23.00
计算机网络	Peterson/65.00
数据通信与网络教程（第2版）	Shay/69.00
高速网络与因特网——性能与服务质量（第2版）	Stallings/45.00
TCP/IP详解 卷1：协议	Stevens/39.00
TCP/IP详解 卷2：实现	Stevens/69.00
TCP/IP详解 卷3：TCP 事务协议、HTTP、NNTP和UNIX域协议	Stevens/28.00
ISDN、B-ISDN以及帧中继和ATM（第4版）	Stallings/35.00
网络互连：网桥、路由器、交换机和互连协议（第2版）	Perlman/36.00

高性能通信网络	Walrand/64.00
数据通信与网络	Forouzan/59.00
面向对象软件构造（第2版）	Meyer/78.00
面向对象与经典软件工程（第5版）	Schach/59.00
IT项目管理（第2版）	Schwalbe/65.00
Software for Use	Constantine/39.00
编写有效用例	Cockburn/25.00
设计模式：可复用面向对象软件的基础	Gamma, Helm, Johnson, Vlissides/38.00
软件工程：Java语言实现	Schach/51.00
软件工程：实践者的研究方法	Pressman/68.00
系统分析与设计	Satzinger/60.00
计算机图形学原理及实践——C语言描述（第2版）	Foley, Dam, Feiner, Hughes/88.00
计算机图形学的算法基础（第2版）	Rogers/55.00
专家系统原理与编程	Giarratano, Riley/59.00
机器学习	Mitchell/58.00
神经网络设计	Hagan/69.00
人工智能	Nilsson/45.00
计算理论导论	Sipser/39.00
数字逻辑应用与设计	Yarbrough/69.00
人本界面——设计交互式系统的最新指示	Raskin/28.00
VHDL设计、表示和综合（第2版）	Armstrong, Gray/即将出版
电磁场与电磁波	Guru/68.00
DSP算法、应用与设计	Bateman/即将出版

重点大学计算机教材

UNIX操作系统教程	张红光/33.00
Windows CE.NET系统分析及实验教程	陈向群/33.00
Windows内核实验教程	陈向群/25.00
Windows操作系统原理	尤晋元 史美林 陈向群 郑扣根/39.00
信息安全学	刘艺/即将出版
计算机英语	刘艺 王春生/33.00
计算机基础	詹江平/18.00
16/32位微机原理：汇编语言及接口技术	钱晓捷 陈涛/29.00
计算机网络	蔡开裕 范全鹏/28.00
数字电子技术	张英全/19.00
模拟电子技术	张英全/19.00
数值方法	金一庆/25.00

全美经典学习指导系列

C++编程习题与解答（英文版·第2版）	Hubbard/39.50
C++编程习题与解答	Hubbard/徐漫江/39.00
Java编程习题与解答（英文版）	Hubbard/28.00
Java编程习题与解答	Hubbard/王强/29.00
Visual Basic编程习题与解答	Gottfried/向尕/29.00
关系数据库习题与解答（英文版）	Mata- Toledo/26.00
关系数据库习题与解答	Mata-Toledo/周云辉/19.00
SQL编程习题与解答	Mata Toledo/胡志军/29.00
软件工程习题与解答（英文版）	Gustafson/28.00
计算机图形学习题与解答（英文版，第2版）	Xiang/35.00
计算机图形学习题与解答	Xiang/陈泽琳/29.00
计算机体系结构习题与解答（英文版）	Carter/30.00
计算机网络习题与解答（英文版）	Tittel/38.00
数据结构习题与解答——Java语言描述（英文版）	Hubbard/38.00
数据结构习题与解答——Java语言描述	Hubbard/阳国贵/39.00
数据结构习题与解答——C++语言描述（英文版）	Hubbard/40.00
计算机导论习题与解答	Cushman/薛静峰/30.00
计算机科学导论习题与解答（英文版）	Mata-Toledo/30.00
操作系统习题与解答	Harris/须德/19.00
操作系统习题与解答（英文版）	Harris/25.00

软件工程技术丛书

统一软件开发过程	Booch, Rumbaugh, Jacobson/周伯生/45.00
Rational统一过程引论（原书第2版）	Kruchten/周伯生 吴超英/38.00
CMMI 精粹-集成化过程改进实用导论	Ahern, Clouse, Turner/周伯生/35.00
CMM实施指南	Persse/蔡愉祖/39.00
软件过程改进	Zahran/陈新/49.00
面向对象的分析与设计	Haigh/贾爱霞/38.00
编写有效用例	Cockburn/王雷/35.00
系统分析与设计	Satzinger/朱群雄/65.00
用例分析技术（第2版）	Schneider/姚淑珍/35.00
有效需求实践	Young /韩柯/35.00
面向模式的软件体系结构 卷1：模式系统	Buschmann/贲可荣/45.00
面向模式的软件体系结构 卷2：用于并发和网络化对象的模式	Schmidt/贲可荣/即将出版
从规范出发的程序设计	Morgan/裘宗燕/45.00

软件架构：组织原则与模式	Dikel/张恂/35.00
面向对象软件开发原理（原书第2版）	Eliens/袁兆山/48.00
面向对象的软件测试	McGregor/杨文宏/35.00
面向对象的方法：原理与实践（原书第3版）	Ian Graham /袁兆山/即将出版
UML和模式应用	Craig Larman 姚淑珍/48.00
UML用户指南	Booch, Rumbaugh, Jacobson /邵维忠 麻志毅 张文娟 孟详文/48.00
UML与C++：面向对象开发基础（原书第2版）	Lee/麻志毅/45.00
UML参考手册	Booch, Rumbaugh, Jacobson/姚淑珍 唐发根/69.00
软件复用：结构、过程和组成	Jacobson, Griss/韩柯/55.00
软件复用技术：在系统开发过程中考虑复用	Carma/廖泰安/即将出版
软件项目管理——一个统一的框架	Royce/周伯生/48.00
基于项目的软件工程——面向对象研究方法	Stiller/贲可荣/30.00
软件需求管理：统一方法	Leffingwell/蒋慧/35.00
软件需求	Wiegers/陆丽娜/19.00

电子工程丛书

信号处理的小波导引	Mallat/杨力华/55.00
VHDL设计、表示和合成（原书第2版）（附光盘）	Armstrong/戴葵/65.00
电子电路原理（上册）	Burn/董平/49.00
电子电路原理（下册）（附光盘）	Burn/董平/45.00
信号、系统与信号处理（上册）（附光盘）	Ambardar/冯博琴/40.00
信号、系统与信号处理（下册）	Ambardar/冯博琴/40.00
电磁场与电磁波	Guru/周克定/39.00
数字系统设计基础教程	Uyemura/陈怒兴/32.00
Verilog HDL硬件描述语言	Bhasker/徐振林 /19.00
无线通信中的智能天线	Liberti/马凉/38.00
光纤通信技术	Mynbaev, Scheiner/吴时霖/78.00