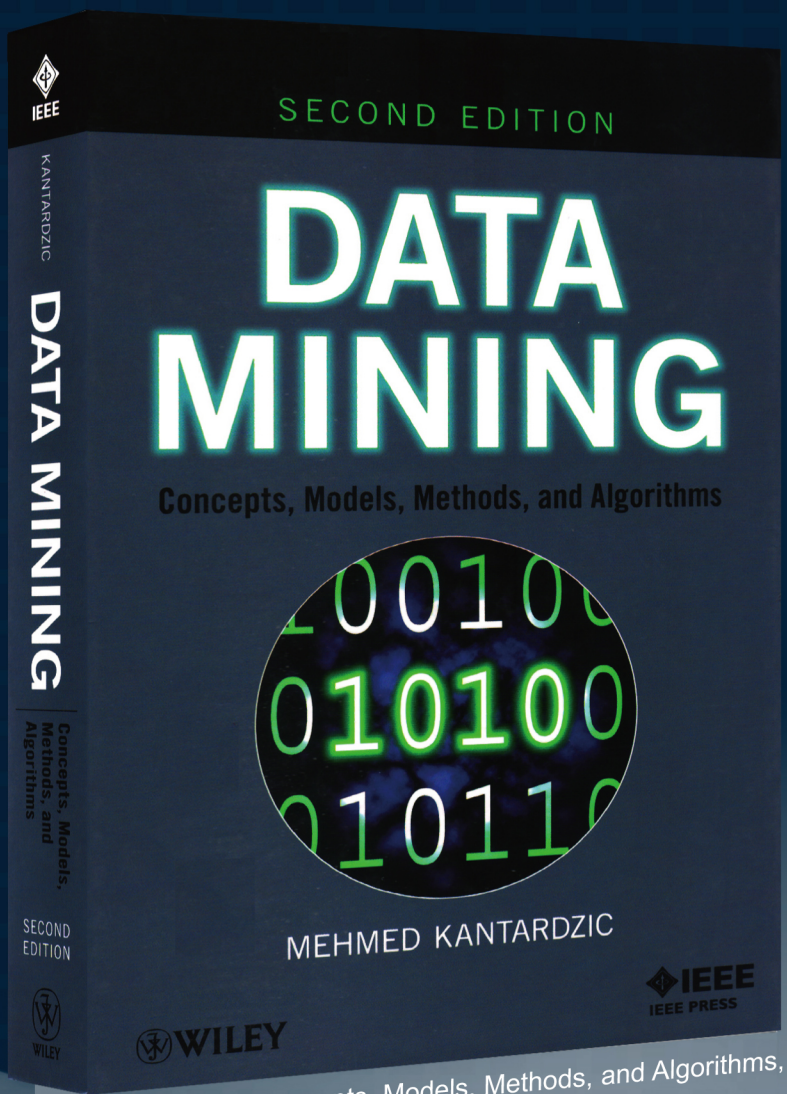
国外计算机科学经典教材

数据挖掘：概念、模型、方法和算法(第2版)

[美] Mehmed Kantardzic 著　　王晓海 吴志刚 译



Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition

WILEY

清华大学出版社

国外计算机科学经典教材

数据挖掘：概念、模型、方法和算法

(第 2 版)

[美] Mehmed Kantardzic 著

王晓海 吴志刚 译

清华大学出版社

北 京

Mehmed Kantardzic
Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition
EISBN：978-0-470-89045-5

图书在版编目(CIP)数据

数据挖掘：概念、模型、方法和算法(第 2 版)/(美)坎塔尔季奇(Kantardzic, M.) 著；王晓海，吴志刚 译. —北京：清华大学出版社，2013.1

(国外计算机科学经典教材)

书名原文：Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition

ISBN 978-7-302-30714-3

Ⅰ. ①数… Ⅱ. ①坎… ②王… ③吴… Ⅲ. ①数据采集—教材 Ⅳ. ①TP274

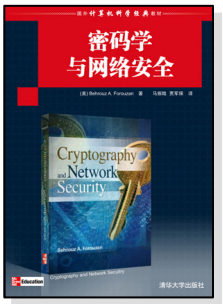
中国版本图书馆 CIP 数据核字(2012)第 278640 号

责任编辑：王　军　韩宏志
装帧设计：思创景点
责任校对：蔡　娟
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：26.25　　**字　　数**：655 千字
版　　次：2013 年 1 月第 1 版　　**印　　次**：2013 年 1 月第 1 次印刷
印　　数：1~4000
定　　价：59.00 元

产品编号：

《国外计算机科学经典教材》丛书

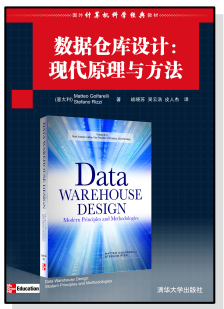


本书延续了 Forouzan 先生一贯的风格，以通俗易懂的方式全面阐述了密码学与计算机网络安全问题所涉及的各方面内容，从全局角度介绍了计算机网络安全的概念、体系结构和模式。本书以 Internet 为框架，以形象直观的描述手法，详细介绍了密码学、数据通信和网络领域的基础知识、基本概念、基本原理和实践方法，堪称密码学与网络安全方面的经典著作。

书　名：密码学与网络安全　　**作　者**：(美) Behrouz A.Forouzan
译　者：马振晗　贾军保　　**ISBN**：9787302185840
定　价：79.8 元　　**出版社**：清华大学出版社

本书特色：

- 本书作者 Behrouz A. Forouzan 是迪安那大学教授，从事计算机信息系统方面的研究工作。著有多部权威性著作，例如《TCP/IP 协议族》系列和《密码学与网络安全》。
- 本书以通俗易懂的方式全面阐述了密码学与计算机网络安全问题所涉及的各方面内容，是密码学与网络安全方面的经典权威之作。
- 本书有配套的中文导读英文版。英文版的章前导读、内容概要等部分译为中文，便于读者阅读。
- 本书最大的特点就是让读者更容易地理解密码学与网络安全。因此，其概念阐述直观、易懂；程序可用性强，便于学生实践；讲述最新的网络安全技术，贴近实际。



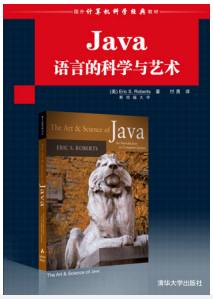
本书介绍最前沿的数据仓库设计技术，指导您构建安全可靠的决策支持基础结构。它阐述遵循成熟可靠的软件工程原理的实用设计方法，分析如何设计最新 ETL 过程，讨论如何将概念模式转换为关系模式，还讲述集成异构数据源、实现星型和雪花模式、管理动态层次结构以及通过具体化和拆分视图来优化性能的方法。

书　名：数据仓库设计：现代原理与方法　　**作　者**：(意大利) Matteo Golfarelli　Stefano Rizzi
译　者：战晓苏　吴云浩　皮人杰　　**ISBN**：9787302230748
定　价：49.80 元　　**出版社**：清华大学出版社

本书特色：

- 本书作者 Matteo Golfarelli 是意大利博洛尼亚大学计算机科学与技术学院副教授，兼任国际杂志 *Data Mining, Modeling and Management* 的编委。
- 本书作者 Stefano Rizzi 是意大利博洛尼亚大学计算机科学与技术学院教授，兼任 *Encyclopedia of Database Systems* 杂志的数据仓库设计编辑。
- 本书内容极其专业、深刻，图文并茂，讲解清晰。
- 本书将广博的数据仓库设计内容浓缩在一卷书中，层次分明，行文严谨，将相关点讲得通通透透，是各层次读者的良师益友。

《国外计算机科学经典教材》丛书



Eric S. Roberts，美国斯坦福大学计算机科学系教授，并担任主管教学事务的系主任。同时还由于教学改革所取得的成就被评为 Charles Simonyi 荣誉教授。他于 1980 年获得哈佛大学应用数学博士学位，并曾在加州 Palo Alto 的 DEC 公司的系统研究中心工作了 5 年。作为一位成功的教育工作者，Roberts 还获得了 1993 年的 Bing Award 奖。

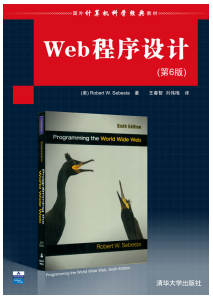
书　名：Java 语言的科学与艺术　　**作　者**：(美) Eric S.Roberts

译　者：付　勇　　**ISBN**：9787302184416

定　价：59.80 元　　**出版社**：清华大学出版社

本书特色：

- ◀ 采用现代面向对象方法，从零开始介绍最有用的类层次结构。
- ◀ 全文使用图形和交互式程序，充分激发学生的学习兴趣。
- ◀ 使用传记简介、引用以及哲学片段来突出计算的历史和理性背景。
- ◀ 着重强调算法和问题解决，而今天的初级教科书通常忽略了这一点。



当前关于 Web 开发的书籍很多，但都是针对专业的开发人员，对于广大学生则往往难以理解和接受。本书由拥有 30 余年计算机教学经验的教授编写，专门针对大学生来讨论 Web 编程，充分考虑了读者的知识背景，比较全面地介绍了建立和维护 Web 服务器站点所必需的工具和技术，包括 XHTML、CSS、JavaScript、XML、Flash 技术、PHP、Ajax、ASP.NET、基于 Web 的数据库访问技术、Ruby、Rails 框架等。

书　名：Web 程序设计(第 6 版)　　**作　者**：(美) Robert W.Sebesta

译　者：王春智　刘伟梅　　**ISBN**：9787302242499

定　价：69.00 元　　**出版社**：清华大学出版社

本书特色：

- ◀ 面向大学生安排内容，充分考虑了他们的知识背景和实际教学需求。
- ◀ 全面覆盖了建立和维护 Web 服务器站点所必需的工具和技术，第 6 版还融入了 NetBeans 6.7、Visual Studio 和 ASP.NET Web Service 等最新技术。
- ◀ 本书介绍的工具和技术都是近年来新涌现出来的并已经成熟和得到广泛应用的，有利于为学生将来从事 Web 开发打下坚实基础，同时也为初学人员快速成长为 Web 程序院提供了一条捷径。
- ◀ Robert W. sebesta 博士拥有 30 余年的计算机教学经验，ACM 和 IEEE 成员，本书是其力作，前 5 版均很受欢迎。

出 版 说 明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，亟需一批门类齐全、具有国际先进水平的计算机经典教材，以适应我国当前计算机科学的教学需要。通过使用国外优秀的计算机科学经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培养出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外多家知名的出版机构 Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 等精选、引进了这套“国外计算机科学经典教材”。

作为世界级的图书出版机构，Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 通过与世界级的计算机教育大师携手，每年都为全球的计算机高等教育奉献大量的优秀教材。清华大学出版社和这些世界知名的出版机构长期保持着紧密友好的合作关系，这次引进的“国外计算机科学经典教材”便全是出自上述这些出版机构。同时，为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从上述这些出版机构出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为该套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部由对应专业的高校教师或拥有相关经验的 IT 专家担任。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外计算机科学经典教材

编审委员会

译 者 简 介

王晓海

曾任总参某部应用研发中心副主任、信息服务中心主任，高级工程师，长期从事数据库应用系统的科研开发工作，负责主持多个大型数据库系统的开发和维护，荣获多项军队科技进步奖，享受军队优秀人才岗位津贴，出版多部论(译)著，在数据库挖掘、数据库应用开发、数据安全保护、数据恢复与数据去密等领域具有丰富的实践经验。

已出版的论著和译著

《Oracle Streams 11g 数据复制》，2012 年，清华大学出版社
《SQL Server 2000 管理、开发及应用实例详解》，2006 年，人民邮电出版社
《空时编码技术》，2004 年，机械工业出版社
《远程通信网络基础》，1996 年，电子工业出版社

吴志刚

工学博士，北京邮电大学副教授，长期从事网络与信息安全技术、数据库技术等领域的学术与科研工作，作为负责人主持过上述领域多项国家 863 计划、发改委产业化示范项目和国家级重大工程项目，获得技术专利 2 项，已在国内外学术期刊和国际会议上发表 20 余篇学术论文。

前　言

从本书第 1 版出版以来的 7 年中，数据挖掘领域在开发新技术和拓展其应用范围方面有了长足的进步。正是数据挖掘领域中的这些变化，令笔者下定决心修订本书的第 1 版，出版第 2 版。本版的核心内容并没有改变，但汇总了这个快速变化的领域中的最新进展，呈现了数据挖掘在学术研究和商业应用领域的最尖端技术。与第 1 版相比，最显著的变化是添加了如下内容：

- 一些新主题，例如集成学习、图表挖掘、时态、空间、分布式和隐私保护等的数据挖掘；
- 一些新算法，例如分类递归树(CART)，DBSCAN (Density-Based Spatial Clustering of Applications with Noise)、BIRCH(Balanced and Iterative Reducing and Clustering Using Hierarchies)、PageRank、AdaBoost、支持向量机(SVM)、Kohonen 自组织映射(SOM)和潜在语义索引(LSI)；
- 详细介绍数据挖掘过程的实用方面和商用理解，讨论验证、部署、数据理解、因果关系、安全和隐私等重要问题；
- 比较数据挖掘模型的一些量化方式方法，例如 ROC 曲线、增益图、ROI 图、McNemar 测试和 K 折交叉验证成对 t 测试。

这是一本教材，所以还增加了一些新习题。这一版也更新了附录中的内容，包含了最近几年的新成果，还反映了某个新主题得到人们的重视时发生的变化。

笔者感谢在课堂上使用本书第 1 版的所有同行，以及支持我、鼓励我和提出建议的所有人，并在新版中采纳了这些建议。笔者真诚地感谢数据挖掘实验室和计算机科学系中的所有同事和同学们，感谢他们审读本书，并提出了许多有益的建议。特别感谢研究生 Brent Wenerstrom、Chamila Walgampaya 和 Wael Emara，他们耐心地校对这个新版本，讨论新章节中的内容，还做了许多校正和增补。Joung Woo Ryu 博士还帮助笔者完成了文字、所有新增图和表格的终稿，笔者对此表示最诚挚的感谢。

本书是面向在校生、毕业生、研究人员和相关从业人员的一本极具价值的指南。本书介绍的广泛主题可以帮助读者了解数据挖掘对现代商业、科学甚至整个社会的影响。

另外，可从 http://www.tupwk.com.cn/downpage 下载本书的汇总参考书目。

Mehmed Kantardzic

作于路易斯维尔

目　录

第 5 章

统 计 方 法

本章目标

- 阐述统计推论在数据挖掘中的一些常用方法。
- 介绍评价数据集的差异的不同统计参数。
- 描述朴素贝叶斯分类和对数回归方法的内容和基本原理。
- 用列联表的相关分析介绍对数线性模型。
- 论述方差分析(ANOVA)和多维样本的线性判别分析的一些概念。

统计学是一门收集、组织数据并从这些数据集中得出结论的科学。描述和组织数据集的一般特性是描述性统计学的主题领域，而怎样从这些数据中推出结论是统计推理的主题。本章将重点介绍统计推理的基本原理；为了加深对这些基本概念的理解，本章也简述了其他的相关内容。

统计数据分析是为数据挖掘制定的最好的一套方法论。历史上，最早在计算机的基础上开发数据分析方面的应用也是有统计人员支持的。从一元到多元数据分析，统计学为数据挖掘提供了各种方法，包括不同类型的回归和判别分析方法。本章只是简要地概括支持数据挖掘过程的统计方法，并没有涵盖所有的方法和方法论；只介绍了现实世界数据挖掘应用中最常用的技术方法。

5.1 统计推断

在统计分析中观测到的所有值，不管其数量是有限还是无限，都称为总体(population)。这个术语适用于任何统计对象，可以是人、物或事件。总体中观测值的数量称为总体的大小。一般来说，总体可能是有限的或无限的，但由于一些有限的总体太大，理论上就把它们假定为无限的。

在统计推断领域，如果观测组成总体的所有值是不可能或不切实际的，就只考虑怎样得出关于总体的结论。例如，试图测定某一品牌的灯泡的平均寿命，但实际上，不可能检测所有这

样的灯泡。因此，在大多数统计分析应用中，必须依据总体中的某个观测值子集。在统计学中，总体的子集称为样本，它描述了一个 n 维向量的有限数据集。本书把总体的这个子集简称为数据集，以免混淆样本的下面两个定义：一个是前面解释过的，表示对总体中单个实体的描述，另一个是这里定义的，表示总体的一个子集。根据已知的数据集，可以建立总体的统计模型，来帮助对该总体作推断。假如从这个数据集中得出的推论是正确的，就得到了能代表总体的样本。选取数据集时，通常选择总体中最简便的数值。但这种方法可能导致对总体的错误推断。如果取样过程得出的推断总是高估或低估总体的某个特性，就称之为偏向。为了消除取样过程出现偏向的可能性，最好是在独立、随机的观察值中选取一个随机的数据集。选取随机样本的主要目的是得到未知总体参数的信息。

数据集和它们所描述的系统之间的关系能用来归纳推理：从所观察的数据来了解(部分)未知的系统。统计推断是与数据分析相关的主要推理形式。统计推断理论包括一些能够对总体进行推断和归纳的方法。这些方法分为两大类：估计和假设检验。

在估计中，为了估计系统的未知参数，需要给出一个置信度或一个置信区间。目的是从数据集 T 中获得信息，来估计现实世界系统 f(X, w)模型的一个或多个参数 w。数据集 T 用变量 X = { $X_1, X_2, \ldots, X_n$} (总体的实体属性)的值从 1～n 的顺序来描述：

$$T = \{(x_{11},...,x_{1n}),(x_{21},...,x_{2n}),...,(x_{m1},...,x_{mn})\}$$

它能组织成表格的形式，作为一组具有相应特征值的样本。对于初始属性集$Y \in X$，只要估计出这个模型的参数，就可以使用它们，根据其他变量或变量集$X^*=X-Y$，来预测随机变量 Y。如果 Y 是数值，就称为回归，如果它是离散的、无序的数据集，就称为分类。

只要从数据集 T 中得到了模型参数 w 的估计值，且知道向量 X*的相应值，就能用所得的模型(以函数 f[X^*, w]的形式给出)预测 Y。预测值 f(X^*, w)和真实值 Y 之间的差称为预测误差，其值最好接近于零。对于 Y 的预测值，模型 f(X^*, w)的自然品质度量指标是整个数据集 T 的期望均值平方误差。

$$E_T[(Y - f[X^*,w])^2]$$

另一方面，在统计检验中，根据对数据集的分析来判断接受还是拒绝对总体特性值的假设。统计假设是关于一个或多个总体的断言或推测。除非检测了整个总体，否则不能完全肯定一个统计假设的真假。当然，在大多数情况下，这是不切实际的，甚至是不可能的。所以可以用随机选取的数据集来检验假设的真假。如果从这些数据集中得出的结果和原假设不一致，就拒绝这个假设，如果得出的证据支持这个假设，就接受它；更确切地讲，这些数据没有充分的证据拒绝它。假设检验的构造可以用“零假设”这个术语来表达，这是指要检验的任何假设，用 H_0 来表示。根据所应用的统计检验，只有已知数据集有足够的证据证明这个假设不成立，才能拒绝 H_0。拒绝 H_0 后，就可以接受总体的备选假设。

本章将详细介绍一些统计估计和假设检验方法。这些方法的主要选择依据是在大量数据集的数据挖掘过程中可应用的技术。

5.2　评测数据集的差异

对于许多数据挖掘任务来说，了解已知数据集中有关中心趋势和数据分布的更一般特性是非常有用的。显然，数据集的这些简单参数是评价不同数据集的差异的描述符。平均数、中位数和众数是反映数据的中心趋势的典型指标，而方差和标准差是反映数据离散程度的指标。

反映数据集中心趋势最常用最有效的数值型指标是平均值(也称为算术平均值)。已知特性 X 有 n 个数值 x_1，x_2，…，x_n，则 X 的平均数是：

$$\text{平均数}=1/n\sum_{i=1}^{n}x_i$$

在现代大多数的统计软件工具中，平均值是一个内嵌函数(和其他描述性统计指标一样)。对于 n 元样本集的每个数值型特征，都可以计算它的平均值，来评价它的中心趋势。有时，数据集中的每个值 x_i 都指定了一个权值 w_i，权值反映了此数据值出现的频率或该值的重要性。这种情况下，加权算术平均数或加权平均数是：

$$\text{平均数}=\sum_{i=1}^{n}w_ix_i \ / \ \sum_{i=1}^{n}w_i$$

在描述一组数据时，平均数是最有用的指标，但它不是唯一的指标。对偏斜数据集来说，中位数能更好地反映它的中心趋势。如果数据集有奇数个有序元素，中位数就是处于正中间的数据值；如果数据集有偶数个有序元素，中位数就是处于正中的两个数据的平均值。用 x_1，x_2，…,x_n 表示一个容量为 n 的数据集，其中的数据按升序排列，则中位数可以定义为：

$$\text{中位数}=\begin{cases}x_{(n+1)/2} & n \text{ 是奇数}\\ (x_{n/2}+x_{(n/2)+1})/2 & n \text{ 是偶数}\end{cases}$$

反映数据集的中心趋势的另一个指标是众数。众数是在数据集中出现频率最高的值。平均数和中位数主要反映了数值型数据集的特性，而众数也适用于分类数据，但是因为这些数据是无序的，所以必须有详细的说明。数据集中出现频率最高的值可能有多个，导致有多个众数。因此，可以把数据集分为单模态(只有一个众数)和多模态(有两个或多个众数)。多模态数据集可以更精确地定义为双模态、三模态等。对于适度倾斜的单模态频率曲线，数值型数据集有下面的经验关系：

$$\text{平均数} - \text{众数} \leqslant 3*(\text{平均数} - \text{中位数})$$

此关系可用来分析数据集的分布，并根据其中两个中心趋势指标估计另一个指标。

例如，分析简单数据集 T 的这 3 个指标，该数据集的数值如下：

$$T = \{3, 5, 2, 9, 0, 7, 3, 6\}$$

排序后得到同一组数据：

$$T = \{0, 2, 3, 3, 5, 6, 7, 9\}$$

反映中心趋势的 3 个描述性统计指标是：

$$\text{平均数}_T = (0+2+3+3+5+6+7+9)/8 = 4.375$$
$$\text{中位数}_T = (3+5)/2 = 4$$
$$\text{众数}_T = 3$$

数值数据分散的程度称为数据的离散度。反映离散度最常用的指标是标准差σ和方差σ^2。n 个数值 $x_1,x_2,\ldots,x_n$ 的方差是：

$$\sigma^2 = (1/(n-1))\sum_{i=1}^{n}(x_i - mean)^2$$

标准差σ是方差σ^2 的平方根。作为反映离散度的指标，标准差σ的基本性质如下：

(1) σ 反映了平均值的离散程度，仅当选取平均值作为反映中心趋势的指标时使用。

(2) 仅当数据中不存在离散，即所有的度量值都相同时，σ=0，否则 σ>0。

对于上例中的数据集，方差 σ^2 和标准差 σ 是：

$$\sigma^2 = 1/7\sum_{i=1}^{8}(x_i - 4.375)^2$$
$$\sigma^2 = 8.5532$$
$$\sigma = 2.9246$$

在许多统计软件工具中，箱图(boxplot)是表示中心趋势和离散度的描述性统计指标最常用的可视化分析工具，通常由数据集的平均值、方差确定，有时也用最小值、最大值。在上例中，T 集中的最大值和最小值是 $min_T=0$，$max_T=9$。它的统计描述图形可以用箱图来表示，如图 5-1 所示。

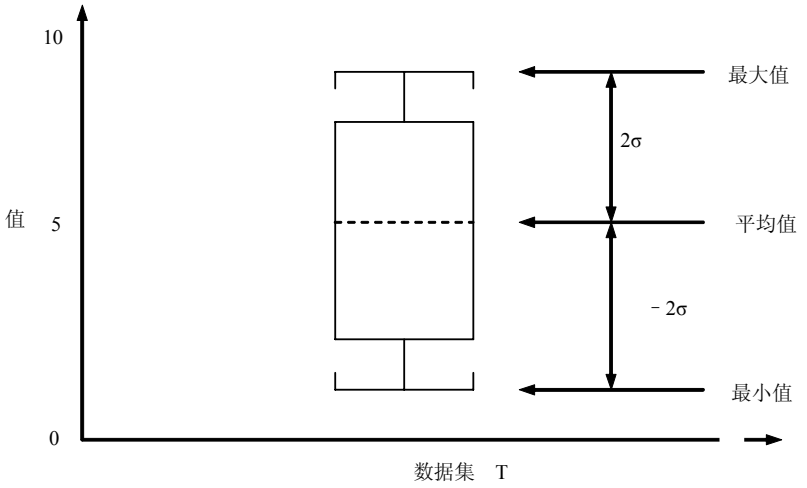


图 5-1 数据集 T 的平均值、方差、最小值和最大值的箱图表示

分析大型数据集需要提前正确地理解数据，这有助于领域专家控制数据挖掘过程，并正确估计数据挖掘应用的结果。数据集的中心趋势指标仅可用于某些具有特定分布的数据值。因此，了解所分析的数据集的分布特性是很重要的。数据集中值的分布根据其离散程度来描述。通常，最好使用直方图来表示它，图 5-2 给出了一个例子。除了量化每个特征值的分布

外，了解分布的全局特点和所有特异性也是很重要的。知道数据集具有经典的钟形分布曲线，可帮助研究人员给数据评估使用许多传统的统计技术。但在许多实际情况中，数据的分布是偏斜的或多模态的，平均值或标准差等概念的传统解释没有什么意义。

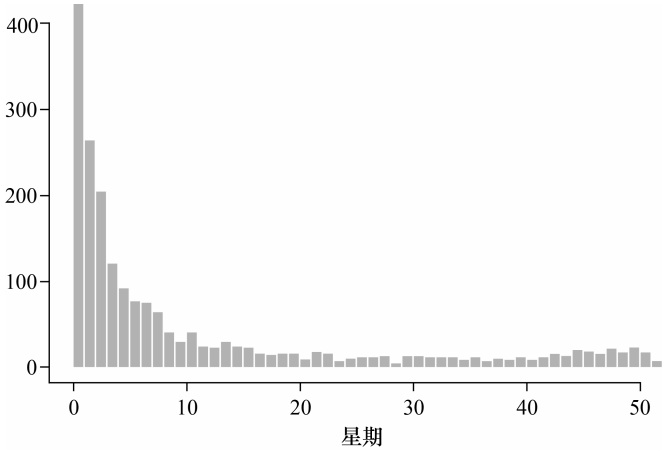


图 5-2　显示单特征分布曲线

评估过程的一部分是确定数据集中特征之间的关系。通过散点图进行简单的可视化，可以初步估计出这些关系。图 5-3 显示了一些比较每对特征的散点图。这个可视化技术可用于大多数集成的数据挖掘工具。这些关系的量化通过相关因子来得到。

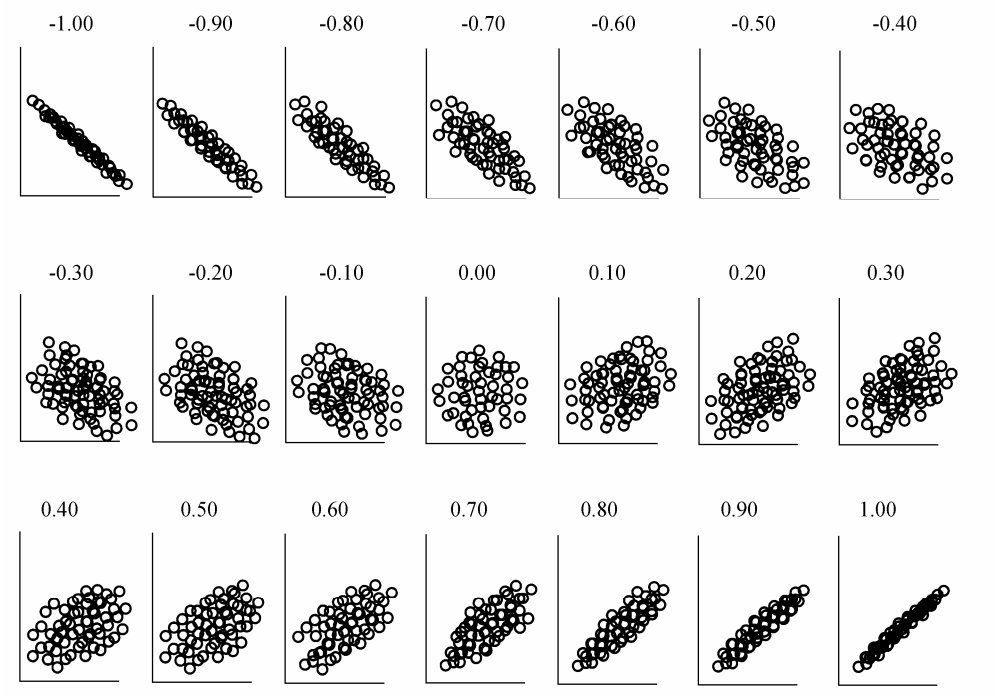


图 5-3　散点图显示了从-1～1 的特征之间的关系

这些可视化过程是数据理解阶段的一部分，在更好地准备数据挖掘时很重要。这个人为的

解释有助于得到数据的一般理解，也可以识别出异常或有趣的特征，例如异常点。

5.3 贝叶斯定理

不难想象，数据并不是总体或待建模系统的唯一可用的信息资源。贝叶斯方法提供了一套将这些外部信息融入数据分析过程的原理方法。这个过程先给出待分析数据集的概率分布。因为这个分布在给出时没有考虑任何数据，所以称为先验分布(prior distribution)。新的数据集将先验分布修正后得到后验分布(posterior distribution)。进行这个修正的基本工具就是贝叶斯定理。

贝叶斯定理为解决归纳-推理分类问题的统计方法提供了理论背景。下面首先解释贝叶斯定理中的基本概念，然后再运用这个定理说明朴素贝叶斯分类过程(或称为简单贝叶斯分类)。

设 X 是一个类标号未知的数据样本，H 为某种假定：数据样本 X 属于某特定的类 C。要求确定 P(H/X)，即给定了观测数据样本 X，假定 H 成立的概率。P(H/X)表示给出数据集 X 后，我们对假设 H 成立的后验概率。相反，P(H)是任何样本的先验概率，不管样本中的数据是什么。后验概率 P(H/X)比先验概率 P(H)基于更多的信息。贝叶斯定理提供了一种由概率 P(H)、P(X)和 P(X/ H)计算后验概率 P(H/X)的方法，其基本关系是：

$$P(H/X) = [\, P(X/\, H) \cdot P(H)] \,/\, P(X)$$

现在假设有 m 个样本 $S = \{S_1, S_2, \ldots, S_m\}$ (训练数据集)，每个样本 S_i 都表示为一个 n 维向量$\{x_1, x_2, \ldots, x_n\}$。$x_i$ 值分别和样本属性 A_1，A_2,…,A_n 相对应。还有 k 个类 $C_1, C_2, \ldots, C_k$，每个样本属于其中一个类。另外给出一个数据样本 X(它的类是未知的)，可以用最高的条件概率 $P(C_i/X)$ 来预测 X 的类，这里 i= 1,…,k。这是朴素贝叶斯分类的基本思想。可以通过贝叶斯定理计算这些概率：

$$P(C_i/X) = [\, P(X \,/\, C_i) \cdot P(C_i)] \,/\, P(X)$$

因为对所有的类，P(X)都是常量，所以仅需要计算乘积 $P(X \,/\, C_i) \cdot P(C_i)$的最大值。类的先验概率用下面的式子计算：

$P(C_i)$ = 类 C_i 的训练样本数量 / m(m 是训练样本的总数)。

因为 $P(X \,/\, C_i)$ 的计算是极其复杂的，特别是对大量的数据集来说，所以要给出零假设：各属性之间是条件独立的。利用这个假设，可以用一个乘积来表示 $P(X \,/\, C_i)$。

$$P(X \,/\, C_i) = \prod_{t=1}^{n} P(x_t \,/\, C_i)$$

式中 x_t 是样本 X 的属性值。概率 $P(x_t/C_i)$ 能够通过训练数据集来估算。

这个简单的例子表明，即使是对大量的训练数据集，朴素贝叶斯分类也是一个简单的计算过程。已知训练数据集包含 7 个四元样本(表 5-1)，需要预测新样本 X={1, 2, 2, class=？}的分类。对每个样本来说，A_1，A_2 和 A_3 是输入维，C 是输出分类。

表 5-1　用朴素贝叶斯分类过程给训练数据集分类

样　本	属性 1	属性 2	属性 3	类
	A_1	A_2	A_3	C
1	1	2	1	1
2	0	0	1	1
3	2	1	2	2
4	1	2	1	2
5	0	1	2	1
6	2	2	2	2
7	1	0	1	1

在这个例子中，因为仅有两个类，所以只需确定乘积 $P(X / C_i) \cdot P(C_i)(i=1, 2)$的最大值。首先计算每个类的先验概率 $P(C_i)$：

$$P(C = 1) = 4/7 = 0.5714$$
$$P(C = 2) = 3/7 = 0.4286$$

然后，用训练数据集为所给新样本 X={1, 2, 2, C =？}(或更精确，$X = \{A_1 =1, A_2 = 2, A_3 = 2, C = ？\}$)的每个属性值计算条件概率 $P(x_t/C_i)$：

$$P(A_1 = 1/C = 1) = 2/4 = 0.50$$
$$P(A_1 = 1/C = 2) = 1/3 = 0.33$$
$$P(A_2 = 2/C = 1) = 1/4 = 0.25$$
$$P(A_2 = 2/C = 2) = 2/3 = 0.66$$
$$P(A_3 = 2/C = 1) = 1/4 = 0.25$$
$$P(A_3 = 2/C = 2) = 2/3 = 0.66$$

假设各属性是条件独立的，则条件概率 $P(X/C_i)$是：

$$P(X/C = 1) = P(A_1 = 1/C = 1) \cdot P(A_2 = 2/C = 1) \cdot P(A_3 = 2/C = 1)$$
$$= 0.50 \cdot 0.25 \cdot 0.25 = 0.03125$$
$$P(X/C = 2) = P(A_1 = 1/C = 2) \cdot P(A_2 = 2/C = 2) \cdot P(A_3 = 2/C = 2)$$
$$= 0.33 \cdot 0.66 \cdot 0.66 = 0.14375$$

最后，用相应的先验概率乘以这些条件概率，得到 $P(C_i/X)$的大约值($\approx$)，并从中找到它们的最大值：

$$P(C_1/X) \approx P(X/C = 1) \cdot P(C = 1) = 0.03125 \cdot 0.5714 = 0.0179$$
$$P(C_2/X) \approx P(X/C = 2) \cdot P(C = 2) = 0.14375 \cdot 0.4286 = 0.0616$$
$$\Downarrow$$
$$P(C_2/X) = \mathrm{Max}\{P(C_1/X), P(C_2/X)\} = \{0.0179,0.0616\}$$

根据朴素贝叶斯分类器算出的最终结果的前两个值，就能预测出，新样本 X 属于类 C=2。这个类的概率乘积 $P(X/C = 2) \cdot P(C = 2)$越大，$P(C = 2/X)$就越大，因为它和计算出的概率乘积成正比。

理论上，与数据挖掘的其他分类方法相比，贝叶斯分类的误差率最小。然而实践中并非总

是如此，因为对属性和类的条件独立性的假设是不准确的。

5.4 预测回归

连续型数值的预测可用称为“回归”的统计技术来建模。回归分析的目的是找到一个联系输入变量和输出变量的最优模型。更确切地讲，回归分析是确定变量 Y 与一个或多个变量之间的相互关系的过程。Y 通常叫做响应输出或因变量，X_i－s 叫做输入、回归量、解释变量或自变量。进行回归分析的常见原因包括：

(1) 测量输出的开销很大，而输入则不是，因此要寻求一种预测输出的廉价方法；

(2) 输入值是已知的，而输出值是未知的，所以需要预测输出值；

(3) 控制输入值，就能够预测相应输出的行为；

(4) 一些输入值和输出值之间可能有因果关系，需要识别这些关系。

在详细解释回归技术之前，先说明插值和回归这两个概念之间的主要区别。在这两个技术中，训练集 $X=\{x^t, r^t\}_{t=1,N}$ 都是已知的，其中 x^t 是输入特征，输出值 $r^t \in R$。

- 如果数据集中没有干扰数据，就进行插值。此时需要找出函数 f(x)，使所有这些训练数据点都满足。在多项插值中，给定 N 个点，就可以使用(N-1)个多项式，给任意输入 x 预测输出 r。
- 在回归技术中，是将干扰因素ε添加到未知函数 f 的输出中，即 $r^t = f(x^t) + \varepsilon$。所谓干扰因素，是指某些隐藏的变量 z^t 是我们观测不到的。因此需要通过模型 $g(x^t)$得到当前训练数据和未来数据的近似输出 $r^t = f(x^t, z^t)$。还必须最小化经验误差：$E(g/x) = 1/N \ \Sigma \ (r^t - g[x^t])^2$ ，$t = 1 \sim N$

广义线性回归模型是目前最常用的统计方法。它用来描述一个变量的变化趋势和其他几个变量值的关系。这类关系的建模叫做线性回归。统计建模的任务并不仅仅是拟合模型，还常常需要从几个可行的模型中选择最优的一个。在不同模型间择优的一种客观方法是方差分析方法(ANOVA)，参见 5.5 节。

拟合一组数据的关系可以用预测模型来表示，这个预测模型叫做回归方程。应用最广泛的回归模型是广义线性模型，表示为：

$$Y = \alpha + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + ... + \beta_n \cdot X_n$$

把这个方程应用到已知的每个样本中，可得到一个新的方程式组：

$$Y_j = \alpha + \beta_1 \cdot x_{1j} + \beta_2 \cdot x_{2j} + \beta_3 \cdot x_{3j} + ... + \beta_n \cdot x_{nj} + \varepsilon_j \qquad j = 1, ..., m$$

式中 ε_j 是 m 个给定样本中各个样本的回归误差。线性模型之所以是线性的，是因为 y_j 的期望值是一个线性函数：输入值的加权和。

只有一个输入变量的线性回归是最简单的回归形式。它将一个随机变量 Y(称为响应变量)建模为另一个随机变量 X(称为预测变量)的线性函数。已知 n 个样本或形如$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$的数据点，其中 $x_i \in X$ ，$y_i \in Y$，则线性回归可表示为：

$$Y = \alpha + \beta \cdot X$$

式中 α 和 β 都是回归系数。假定变量 Y 的方差是一个常量，可以用最小二乘法来计算这些系数，使实际数据点和估计回归直线之间的误差最小。这些残差平方和常常称为回归直线的误差平方和，用 SSE 来表示：

$$\mathrm{SSE} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - y_i')^2 = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

式中 y_i 是所给数据集的真实输出值，而 y_i' 是从模型中得出的响应值。令 SEE 分别对 α 和 β 求微分，得到：

$$\partial(\mathrm{SSE})/\partial\alpha = -2\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)$$

$$\partial(\mathrm{SSE})/\partial\beta = -2\sum_{i=1}^{n}((y_i - \alpha - \beta x_i)\cdot x_i)$$

令微分方程等于零(使总误差最小)，整理得方程组：

$$n\alpha + \beta\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$\alpha\sum_{i=1}^{n} x_i + \beta\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i \cdot y_i$$

解方程组，就得到 α 和 β 的计算式。利用与平均数的标准关系，这种简单优化的回归系数为：

$$\beta = [\sum_{i=1}^{n}(x_i - \mathrm{mean}_x)\cdot(y_i - \mathrm{mean}_y)]/[\sum_{i=1}^{n}(x_i - \mathrm{mean}_x)^2]$$

$$\alpha = \mathrm{mean}_y - \beta\cdot\mathrm{mean}_x$$

这里 mean_x 和 mean_y 是训练数据集中的随机变量 X 和 Y 的平均值。注意，根据给定的数据集，α 和 β 的值只是整个总体的真实参数的估计值，可用方程式 $y=\alpha+\beta x$ 根据输入变量 x_0 来预测均值响应 y_0，没必要从原来的样本集中取值。

例如，以表格的形式给出一个数据集样本(表 5-2)，分析两个变量(预测变量 A 和响应变量 B)之间的线性回归，这个线性回归可以表示如下：

$$B = \alpha + \beta\cdot A$$

表 5-2　回归方法所用到的数据

A	B
1	3
8	9
11	11
4	5
3	2

可以通过上面的公式(用 mean_A=5.4，mean_B=6)计算系数 α 和 β 的值，结果是：

$$\alpha = 0.8$$
$$\beta = 0.92$$

最优的线性回归模型是：

$$B = 0.8 + 0.92 \cdot A$$

图 5-4 把上面的数据集和线性回归表示为一组点和相应的直线。

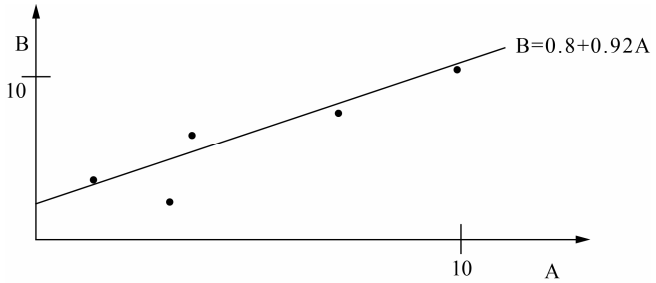


图 5-4　表 5-2 所给的数据集的线性回归图

多元回归是线性回归的扩展，涉及多个预测变量。响应变量 Y 建模为几个预测变量的线性函数。例如，设预测变量是 X_1、X_2 和 X_3，那么多元线性回归可表示为：

$$Y = \alpha + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3$$

式中系数 α、β_1、β_2和β_3的值可以利用最小二乘法求解。对于两个以上输入变量的线性回归模型，可以通过矩阵计算参数 β：

$$Y = \beta \cdot X$$

式中 $\beta = \{\beta_0, \beta_1, ..., \beta_n\}$，$\beta_0 = \alpha$，X 和 Y 是已知训练数据集的输入和输出矩阵。误差平方和 SSE 也可以用矩阵表示：

$$SSE = (Y - \beta \cdot X) \cdot (Y - \beta \cdot X)$$

优化后得

$$\partial(SSE) / \partial\beta = 0 \Rightarrow (X' \cdot X)\beta = X' \cdot Y$$

最后，向量 β 满足矩阵方程式

$$\beta = (X' \cdot X)^{-1}(X' \cdot Y)$$

式中 β 是线性回归的估计系数向量。矩阵 X 和 Y 的维数与训练数据集相同。因此，若只有几百个训练样本，向量 β 的最优解就很容易求得。但在现实世界的数据挖掘问题中，样本数可以达到几百万个。此时，由于矩阵的维数很大，算法的复杂性呈指数增加，因此需要在运算中找到修正值或近似值，或用完全不同的回归方法。

许多非线性回归问题也能转换成一般线性模型的形式。例如，下面的多项式关系：

$$Y = \alpha + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_1X_3 + \beta_4 \cdot X_2X_3$$

设置新的变量$X_4 = X_1 \cdot X_3$和$X_5 = X_2 \cdot X_3$，就能将其转换成线性形式。多项式回归在建模时，也能将新的多项式项添加到基本的线性模型上。三次多项式曲线的形式如下：

$$Y = \alpha + \beta_1 \cdot X + \beta_2 \cdot X_2 + \beta_3 \cdot X^3$$

通过转换到预测变量($X_1 = X$，$X_2 = X^2$ 和 $X_3 = X^3$)，可以将这个模型线性化，将它转变成多元回归问题。这样就可以用最小二乘法来解决它。注意，一般线性回归模型中的线性是指：因变量是未知参数的线性函数。因此，一般的线性模型可能会涉及到自变量的更高次幂，例如 $X_1^2, e^{\beta X}, X_1 \cdot X_2$，1/X 或 X_2^3。然而，最基本的是要对输入变量或它们的合并项选择合适的转换。表 5-3 列出了对回归模型进行线性化的一些有效的转换。

表 5-3　回归线性化的一些有效的转换

函　　数	合适的转换	简易线性回归的形式
指数函数		
$Y=\alpha e^{\beta x}$	$Y^*=\ln Y$	Y^*对 x 的回归
幂函数		
$Y=\alpha x^{\beta}$	$Y^*=\log Y;\ x^*=\log x$	Y^*对 x^*的回归
倒数函数		
$Y=\alpha+\beta(1/x)$	$x^*=1/x$	Y 对 x^*的回归
双曲线函数		
$Y=x/(\alpha+\beta x)$	$Y^*=1/Y;\ x^*=1/x$	Y^*对 x^*的回归

在应用多元回归方法时，主要的任务是从原来的数据集中识别相关的自变量，并用这些相关变量选择回归模型。完成这个任务的两种常用方法是：

(1) 顺序搜索方法(Sequential search approach)——主要是对原来的变量组建立一个回归模型，并选择性地增删变量，直到满足某个整体条件或达到最优。

(2) 组合方法(Combinatorial approach)——实质上，它是一种强力(brute-force)方法，即搜索所有可能的自变量组合，以确定最优的回归模型。

无论使用顺序搜索方法还是组合方法，建模的最大好处来自对应用领域的正确理解。

回归分析过程附加的后续工作是评估这个线性回归模型的性能。相关分析是度量两个变量间的关联程度(在上面的例子中，这种关系用线性回归方程式表达)。两个变量间的线性关联程度可以用一个参数值来表示，该参数称为相关系数 r，它的计算需要用到回归分析中的一些中间结果：

$$r = \beta \cdot \sqrt{(S_{xx} / S_{yy})} = S_{xy} / \sqrt{(S_{xx} \cdot S_{yy})}$$

式中

$$S_{xx}=\sum_{i=1}^{n}(x_i-mean_x)^2$$

$$S_{yy}=\sum_{i=1}^{n}(y_i-mean_y)^2$$

$$S_{xy}=\sum_{i=1}^{n}(x_i-mean_x)(y_i-mean_y)$$

r 的值位于－1～1 之间，r 取负值表示回归直线的斜率为负，正值表示回归直线的斜率为正。在解释 r 值时必须非常谨慎。例如，r 的值等于 0.3 和 0.6 仅仅表示得到了两个正的相关，后者比前者的相关性强。如果认为 r=0.6 表示它的线性关联程度是 r=0.3 的两倍，那就错了。

在本节开头的简单线性回归例子中，得到的模型是 B=0.8+0.92A，可以用相关系数 r 作为评价这个模型性能的指标。由图 4-3 中的数据，可得中间结果：

$$S_{AA}=62$$
$$S_{BB}=60$$
$$S_{AB}=52$$

和最终的相关系数：

$$r=52/\sqrt{62\cdot 60}=0.85$$

相关系数 r =0.85 表示两个变量间有很好的线性关联。此外，还可以这样解释，因为 $r^2=0.72$，所以变量 B 有约 72%的信息和 A 线性关联。

5.5 方差分析

通常，在分析估计回归直线的性能和自变量对最终回归的影响时，使用方差分析(ANOVA)方法。分析的过程是将因变量的总方差细分成几个有意义的组成部分，它们可以用系统的方式观测和处理。方差分析是许多数据挖掘应用中的有力工具。

方差分析主要用于识别线性回归模型中的哪些 β 值非零。假定已通过最小二乘误差算法求出参数 β 的值，残差就是观察到的输出值和拟合值之差。

$$R_i=y_i-f(x_i)$$

对数据集中的 m 个样本，残差的大小和方差 δ^2 的大小有关。假定模型没有过度参数化(overparametrized)，σ^2 可用下式估计：

$$S^2=[\sum_{i=1}^{m}(y_i-f(x_i))^2]/(m-(n-1))$$

分子是残差和，分母是残差的自由度(d.f.)。

S^2 的主要作用是通过它来比较不同的线性模型。如果拟合模型是合适的，那么 S^2 是 σ^2 的一个很好的估计。假如拟合模型包含冗余变量(一些 β 为 0)，S^2 仍接近于 σ^2。仅当拟合模型不包含一个或多个应包含进来的输入变量，S^2 才显著大于 σ^2 的真实值。在 ANOVA 算法中，这些条件是基本的决策步骤，用于分析输入变量对最终模型的影响。首先，给定所有输入，计算这个模型的 S^2，然后，一个一个从模型删除这些输入，若删除了一个有用的输入，S^2 的估计值就会大幅上升。但若删除了一个多余的输入，S^2 的估计值不应有太大的变化。注意从模型中删除一个输入，相当于令相应的 β 值为 0。原则上，在每次迭代中，都要比较两个 S^2 的值，并分析它们之间的不同。为此，引入 F 比率和 F 统计检验，如下：

$$F = S^2_{new} / S^2_{old}$$

若新模型(除去一个或更多输入后)是适合的，F 就接近 1，若 F 的值明显大于 1，就说明这个模型不适合。应用这个迭代 ANOVA 方法，就能识别哪些输入和输出是相关的，哪些是不相关的。只有所比较的模型是嵌套的，换句话说，一个模型是另一个模型的特例时，ANOVA 方法才有效。

设数据集有 3 个输入变量 x_1, x_2, x_3 与一个输出 Y。要使用线性回归方法，必须根据所需输入变量的个数估计出一个最简单的模型。应用 ANOVA 方法后，得出的结果如表 5-4 所示。

表 5-4　含有 3 个输入 x_1、x_2 和 x_3 的数据集的 ANOVA 分析

情　况	输 入 集	S_i^2	F
1	x_1, x_2, x_3	3.56	
2	x_1, x_2	3.98	F_{21}=1.12
3	x_1, x_3	6.22	F_{31}=1.75
4	x_2, x_3	8.34	F_{41}=2.34
5	x_1	9.02	F_{52}=2.27
6	x_2	9.89	F_{62}=2.48

ANOVA 的结果表明，输入变量 x_3 对输出的估计没有影响，因为 F 比值接近 1。

$$F_{21} = S_2 / S_1 = 3.98 / 3.56 = 1.12$$

在其他所有情况中，输入子集使 F 比值显著增加，因此，在不影响模型性能的情况下，减少输入的维数是不可能的。本例最终的线性回归模型是：

$$Y = \alpha + \beta_1 \cdot x_1 + \beta_2 \cdot x_2$$

多元方差分析(MANOVA)是前述 ANOVA 的一个推广，它解决的数据分析问题中，输出不是单个数值，而是一个向量。分析此类数据的一个方法是分别对输出的每个元素建模，但是这忽视了不同输出间可能的关联。换句话说，这种分析基于如下假设：输出是不相关的。MANOVA 是一种不考虑输出间关联的分析方式。已知一组输入和输出变量，就可以用一个多元线性模型

来分析可用的数据集：

$$Y_j = \alpha + \beta_1 \cdot x_{1j} + \beta_2 \cdot x_{2j} + \beta_3 \cdot x_{3j} + ... + \beta_n \cdot x_{nj} + \varepsilon_j \qquad j=1,2,\ldots,m$$

式中 n 是输入的维数，m 是样本数，y_j 是一个 c×1 维的向量，c 是输出个数。该多元模型可以通过和线性模型相同的方法——最小二乘法进行拟合。方法是线性模型与每个 c 维输出拟合，一次只能拟合一个。每一维的相应残差是 $(y_j - y_j')$，其中 y_j 是这个维的真实值，而 y_j' 是估计值。

单元线性模型的残差平方和与多元线性模型的残差平方和矩阵是类似的，这个矩阵 R 定义为：

$$R = \sum_{j=1}^{m} (y_j - y_j')(y_j - y_j')^T$$

矩阵 R 把每个 c 维的残差平方和放在对角线位置，其余的非对角线元素是相应两个向量的叉积的残差平方和。如果要比较两个嵌套的线性模型，来决定某个 β 值是否为 0，就可以另外构造一个平方和矩阵，并利用类似于 ANOVA 的方法—MANOVA。ANOVA 方法中有 F 统计检验，而 MANOVA 基于矩阵 R，有 4 个常用的检验统计方法：Roy 的最大根检验、Lawley-Hotteling 跟踪检验、Pillai 跟踪检验和 Wilks 的 Lambda 检验。本书不介绍这些检验的计算细节，但大多数统计课本都有这些内容。大多数支持 MANOVA 的标准统计包也支持这 4 种统计检验，并说明什么情况下用何种检验。

古典的多元分析也包括主成分分析方法，即将一组样本向量转换为一组维数更少的新样本向量。第 3 章在数据挖掘的预处理阶段中讨论数据归约和数据转换时，讲到了这种方法。

5.6 对数回归

线性回归用于对连续值函数建模。广义回归模型提供了将用线性回归方法给分类响应变量建模的理论基础。广义线性模型的一种常见形式是对数回归。对数回归将某事件发生的概率建模为预测变量集的线性函数。

对数回归方法不是预测因变量的值，而是估计因变量取给定值的概率 p。例如，对数回归方法并不预测某顾客的信用等级是高是低，而是估计顾客的信用等级高的概率。因变量的实际状态通过观察估计概率来决定。假如估计概率大于 0.50，这个预测结果就接近 YES(信用等级高)，否则就接近 NO(信用等级低)。因此，对数回归中的概率 p 称为成功概率。

只有模型的输出变量定义为二元分类变量，才应用对数回归。另一方面，输入变量也应是定量的，所以对数回归支持更一般的输入数据集。假定输出 Y 有两个编码为 0 和 1 的分类值，由这些数据，可以计算出所给输入样本 $P(y_j = 0) = 1 - p_j$ 和 $p(y_j = 1) = P_j$ 取 0 和 1 的概率。拟合这些概率的模型可以用线性回归来表示：

$$\log(p_j / [1 - p_j]) = \alpha + \beta_1 \cdot X_{1j} + \beta_2 \cdot X_{2j} + \beta_3 \cdot X_{3j} + ... + \beta_n \cdot X_{nj}$$

这个方程式称为线性对数模型，函数 $\log(p_j/[1-p_j])$通常写成 *logit(p)*。输出用对数表示的主要原因是避免它的预测概率超出要求的区间[0，1]。假定有一个训练数据集，按照线性回归的步骤对它建模，用线性方程式表示如下：

$$\log it(p) = 1.5 - 0.6 \cdot x_1 + 0.4 \cdot x_2 - 0.3 \cdot x_3$$

再假设有一个新的样本需要分类，其输入值$\{x_1, x_2, x_3\}$={1, 0, 1}。用线性回归模型，可以估计出输出值为 1 的概率 p[Y=1]。首先，计算相应的 logit(p)：

$$\log it(p) = 1.5 - 0.6 \cdot 1 + 0.4 \cdot 0 - 0.3 \cdot 1 = 0.6$$

然后求给定输入的输出值为 1 的概率。

$$\log(p/[1-p]) = 0.6$$
$$p = e^{0.6}/(1+e^{0.6}) = 0.65$$

根据概率 p 的最终结果，可推出输出值 Y=1 的可能性比另一个分类值 Y=0 小。即使这个简单的例子也表明：对数回归在数据挖掘的应用中是一个简易而强大的分类工具。根据一组数据(训练集)就可以建立对数回归模型，再根据另一组数据(检验集)就可以分析在预测分类值时模型的性能。可以把由对数回归方法得到的结果和其他用于分类的数据挖掘方法(如决策规则、神经网络和贝叶斯方法)进行比较。

5.7　对数-线性模型

对数-线性建模是一种分析分类(或数量型)变量间关系的方法。对数-线性模型近似于离散的、多元的概率分布。在这种广义线性模型中，假定输出 Y_i 具有泊松分布，其期望值 μ_j 的自然对数是输入的线性函数：

$$\log(\mu_j) = \alpha + \beta_1 \cdot X_{1j} + \beta_2 \cdot X_{2j} + \beta_3 \cdot X_{3j} + ... + \beta_n \cdot X_{nj}$$

由于所有变量都是分类变量，因此可用表示数据总体分布的频率表来表示它们。对数-线性建模的目的是识别分类变量间的关系。这种关系对应于模型中的相互作用的项。这样，问题就变成确定模型中哪些 β 值为 0 的问题。ANOVA 也阐述了类似的问题。如果在对数-线性模式中，变量间有相互作用，就表示这些变量不是独立的，而是相关的，相应的 β 不等于 0，此时不应把这些分类变量作为这个分析的输出。如果明确指出了输出，就不应使用对数-线性模型，而用对数回归来分析。因此，下面解释的是定义数据集时没有输出变量的对数-线性模型。所给的变量都是分类的，我们需要分析它们之间的关系。这就是一致性分析任务。

一致性分析是分析关联矩阵(也称列联表)中的分类数据。列联表分析的结果回答了“所分析的属性间是否有关系？”这个问题。例如，有一个 2×2 的列联表，包含总计，如表 5-5 所示。

这个表是调查男性和女性对堕胎态度的结果，共有 1100 个样本，每个样本都包括具有相应值的两个分类变量。“性别”变量的取值是男性和女性，“赞同”变量的取值是是或否。所有样本的总计结果都放在列联表的 4 个元素中。

表 5-5　就堕胎态度调查了 1100 个样本的 2×2 列联表

	赞　同		总　计
性别	是	否	
女	309	191	500
男	319	281	600
总计	628	472	1100

“男性和女性赞同堕胎的程度有差异吗？”这个问题可以转变成“两个属性‘性别’和‘赞同’如果相关，其关联程度如何？”假如相关，则男性和女性的观点有很大的差异，否则，他们有相似的观点。

如前所述，对数-线性模型关心的是分类变量间的关联，所以可以用列联表中的数据求这个模型的一些量(指标)，不过这里不这样做，而是根据对两个列联表的比较，定义特征间关联的算法：

(1) 第一步，把所给的列联表转换成一个具有期望值的表，并假定这些变量是相互独立的，计算出这些期望值。

(2) 第二步，用平方距离指标和卡方检验作为评价两个分类变量间关联的标准，对这两个矩阵进行比较。

这两步计算过程对 2×2 的列联表来说是非常简单的，也适用于多维列联表(多于两个值的分类变量分析，如 3×4 或 6×9)。

下面先说明思路。用 $X_{m\times n}$ 来表示这个列联表。这个表中每行的和是：

$$X_{j+}=\sum_{i=1}^{n}X_{ji}$$

这适用于每一行(j=1，…,m)。同样，可以定义每列的和：

$$X_{+i}=\sum_{j=1}^{m}X_{ji}$$

总和定义为每行和的总和

$$X_{++}=\sum_{j=1}^{m}X_{j+}$$

或每列和的总和

$$X_{++}=\sum_{i=1}^{n}X_{+i}$$

假定各行和各列变量间没有关联，就可以用这些总和计算期望值的列联表。期望值如下：

$$E_{ji} = (X_{j+} \cdot X_{+i})/X_{++} \quad j=1,...,m,\ i=1,...,n$$

通过上式可计算出列联表中的每个值。第一步的最终结果是一个只包括期望值的新表，这两个表具有相同的维数。

对于表 5-5 所示的例子，所有的和(行，列和总和)都表示在列联表中。由这些值可以构造期望值的列联表。第一行和第一列交叉处的期望值是：

$$E_{11} = (X_{1+} \cdot X_{+1}) / X_{++} = 500 \cdot 628 / 1100 = 285.5$$

同样，可以计算出其他的期望值，最终的期望值列联表如表 5-6 所示。

表 5-6　表 5-5 中的数据的 2×2 的期望值列联表

	赞　同		总　计
性别	是	否	
女	285.5	214.5	500
男	342.5	257.5	600
总计	628	472	1100

分类变量相关分析的下一步是对关系进行卡方检验。初始假设 H_0 是假设两个变量是不相关的，可以用皮氏卡方公式来检验：

$$\chi^2 = \sum_{j=1}^{m} \sum_{i=1}^{n} ((X_{ji} - E_{ji})^2 / E_{ji})$$

χ^2 的值越大，拒绝假设 H_0 的可能性越大。对于上例，比较表 5-5 和表 5-6，得出如下检验结果：

$$\chi^2 = 8.2816$$

按照 m×n 维表自由度的计算公式，它的自由度为：

$$d.f. = (m-1) \cdot (n-1) = (2-1)(2-1) = 1$$

一般而言，在置信水平 α 下，若 $\chi^2 \geq T(\alpha)$，那么拒绝假设 H_0。其中 $T(\alpha)$ 是 χ^2 分布表的阈值，这在统计课本中常常会涉及到。本例中，选择 $\alpha = 0.05$，可得到阈值：

$$T(0.05) = \chi^2(1-\alpha, d.f.) = \chi^2(0.95,1) = 3.84$$

作简单的比较

$$\chi^2 = 8.2816 \geq T(0.05) = 3.84$$

所以，结论是：拒绝假设 H_0；此调查中分析的属性是高度相关的。换句话说，男性和女性对堕胎的态度有很大的差异。

这个过程可以推广到分类变量有两个以上值的列联表。下面的例子表明，前述步骤可以不

加任何修改地应用于 3×3 列联表。表 5-7(a)中的初值和表 5-7(b)中的估计值相比较，相应的检验结果是 $\chi^2 = 3.229$。注意此情况下的参数：

$$\text{d.f.} = (n-1)(m-1) = (3-1)(3-1) = 4$$

表 5-7　有三个值的分类属性的列联表

(a) 观察值的 3×3 列联表

属性 1		低	中	高	总和
属性 2	优	21	11	4	36
	良	3	2	2	7
	差	7	1	1	9
总和		31	14	7	52

(b) 假设 H_0 下期望值的 3×3 列联表

属性 1		低	中	高	总和
属性 2	优	21.5	9.7	4.8	36
	良	4.2	1.9	0.9	7
	差	5.4	2.4	1.2	9
总和		31	14	7	52

在推导其他结论和进一步分析所给的数据集时必须非常谨慎。显然，这个样本不大。在表的许多单元格中，观察值的数量也很少。这是个严重的问题，必须进行其他统计分析，来检查样本是不是很好地代表了总体。这里不介绍这个分析，因为在大多数实际的数据挖掘问题中，只要数据集足够大，就可以杜绝这些问题。

分析包括分类数据的列联表是一种归纳总结。归纳的另一个方面是对两个以上分类属性的分析。许多高级统计学课本都讲到了三维或高维列联表的分析方法；它们介绍了如何找出要同时分析的几个属性间的关联。

5.8　线性判别分析

线性判别分析(LDA)是解决因变量是分类型(名义类型或顺序类型)、自变量是数值型的分类问题。LDA 的目标是构造一个判别函数，在计算不同输出类中的数据时产生不同的分数。线性判别函数的形式如下：

$$z = w_1 x_1 + w_2 x_2 + ... + w_k x_k$$

式中 $x_1, x_2, ..., x_k$ 是自变量，z 是判别得分，$w_1, w_2, ..., w_k$ 是加权。判别得分的几何解释如图 5-5 所示。数据样本的判别得分表示它在由一组加权参数定义的直线上的投影。

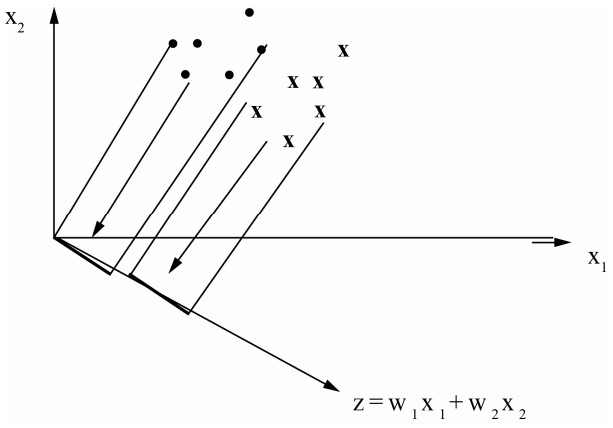


图 5-5　判别得分的几何解释

构建判别函数 z 是为了求出一组权值 w_i，并计算出待分类的样本集的判别得分，使该判别得分的类间方差和类内方差之比最大。构造出判别函数 z 后，就可以用它来预测新样本所属的类。分数线(cutting　scores)是判断每个判别得分的标准。选择分数线要根据各类的样本分布。设 z_a 和 z_b 分别是类 A 和类 B 中待分类样本的平均判别得分。如果两类样本一样大，且服从同一方差分布，那么分数线 $z_{cut\text{-}ab}$ 的最佳选择是：

$$z_{cut-ab} = (z_a + z_b)/2$$

新样本可以根据它的判别得分 $z > z_{cut-ab}$ 或 $z < z_{cut-ab}$ 来分类，当每类样本不一样大时，就把平均判别得分的加权平均数作为最佳分数线：

$$z_{cut-ab} = (n_a \cdot z_a + n_b \cdot z_b)/(n_a + n_b)$$

n_a 和 n_b 表示每类中的样本数。尽管带有多个分数线的判别函数 z 能把样本分为几类，但更复杂的问题还需要使用多重判别分析。多重判别分析应用于每类都构造一个判别函数的情况。这种情况下的分类规则是“选择判别得分最高的一类”，如图 5-6 所示。

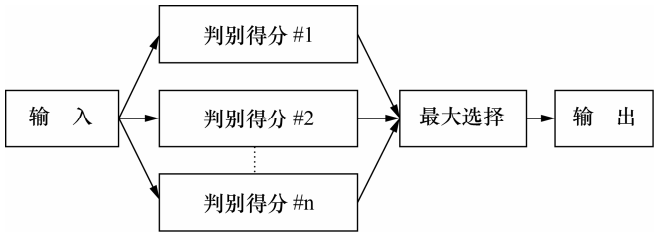


图 5-6　多重判别分析的分类过程

5.9 复习题

1. 统计推断理论中的主要内容是统计检验和估计，它们有什么不同？
2. 分析一组只包括一个属性的数据集 X：

   X = { 7, 12, 5, 18, 5, 9, 13, 12, 19, 7, 12, 12, 13, 3, 4, 5, 13, 8, 7, 6}

   (a) 数据集 X 的平均值是多少？

   (b) 中位数是多少？

   (c) 众数是多少？这个数据集有什么特征？

   (d) 求 X 的标准差。

   (e) 用箱图给出数据集 X 的图形化总结。

   (f) 求数据集 X 的异常点，并讨论这个结果。
3. 由表 5-1 中的训练集，用简单贝叶斯分类法预测下面样本的类别：

   (a) {2,1,1}

   (b) {0,1,1}
4. 已知一组含 X 和 Y 的二维数据集，如表 5-8 所示：

表 5-8 含 X 和 Y 的二维数据集

X	Y
1	5
4	2.75
3	3
5	2.5

   (a) 用线性回归方法计算 $y=\alpha+\beta x$ 中的参数α和β。

   (b) 用相关系数 r 估计(a)中求得的模型的性能。

   (c) 用合适的非线性转换(表 5-3 所示)改进回归结果。改进后的新非线性模型的方程式是什么？讨论相关系数值的简化。
5. 通过对数回归得到的对数函数如下：

$$\text{Logit}(p) = 1.2 - 1.3X_1 + 0.6X_2 + 0.4X_3$$

   求下列样本的输出值为 1 和 0 的概率。

   (a) {1, -1, -1}

   (b) {-1, 1, 0}

   (c) {0, 0, 0}
6. 数据集表示成一个 2×3 的列联表，如表 5-9 所示，分析分类变量 X 和 Y 的关联性。

表 5-9　2×3 的列联表

		Y	
		T	F
X	A	128	7
	B	66	30
	C	42	55

7. 实现一种算法，将输入平面文件中的每个数值型属性用箱图表示。

8. 在线性判别分析中，构造判别函数的基本原则是什么？

9. 实现一种算法，用二维列联表分析分类变量之间的关联。

10. 已知数据集{27, 27, 18, 9}，若

(a) p = 1/3

(b) p = 3/4

求该数据集的 *EMA*(4, 4)，并讨论结果。

11. 若丢失的数据项

(a) 进行相等比较

(b) 进行不等比较

(c) 用默认值代替

(d) 忽略不计

使用贝叶斯分类法确定上述哪个选项正确。

12. 表 5-10 包含了一组数据实例的个数和比率，用于有指导的贝叶斯定理学习。输出变量是性别，其值是男性和女性。假定某人对人寿保险促销选择了“否”，对杂志促销和手表促销选择了“是”，且拥有信用卡保险。使用表中的值和贝叶斯分类法，确定此人为男性的概率。

表 5-10　值和贝叶斯分类法

	杂志促销		手表促销		人寿保险促销		信用卡保险	
	男性	女性	男性	女性	男性	女性	男性	女性
是	4	3	2	2	2	3	2	1
否	2	1	4	2	4	1	4	3

13. 假设某人订阅了至少一本汽车杂志，则他拥有一辆跑车的概率是 40%。另外，3%的成人订阅了至少一本汽车杂志。最后，假设某人没有订阅至少一本汽车杂志，则他拥有一辆跑车的概率是 30%。使用这些信息和贝叶斯定理计算，假设某人拥有一辆跑车，则他订阅了至少一本汽车杂志的概率是多少？

14. 假设本科生抽烟的比例是 15%，研究生抽烟的比例是 23%。如果 1/5 的大学学生是研究生，其余为在校本科生，则一个学生抽烟、且是研究生的概率是多少？

15. 一个 2 × 2 列联表包含 X 和 Y 变量：

		X	
		x1	x2
Y	y1	7	4
	y2	2	8

(a) 确定期望值的列联表。

(b) 如果 χ^2 检验的阈值是 8.28，确定变量 X 和 Y 相互关联的概率。

16. 通过对数回归方式得到的对数函数如下：

Logit(p)=1.2－1.3×1+0.6×2+0.4×3

确定样本{1,−1, −1}的输出值为 0 和 1 的概率。

17. 假定：

- P(好电影|汤姆·克鲁斯参演)=0.01
- P(好电影|汤姆·克鲁斯未参演)=0.1
- P(汤姆·克鲁斯参演了随机选择的电影)=0.01

请计算 P(汤姆・克鲁斯参演|不是好电影)。

18. 下面的训练集有 3 个布尔型输入 x，y，z 和布尔型输出 U。假设要使用贝叶斯分类法预测 U，如表 5-11 所示。

表 5-11　训练集

x	y	z	U
1	0	0	0
0	1	1	0
0	0	1	0
1	0	0	1
0	0	1	1
0	1	0	1
1	1	0	1

(a) 训练完毕后，预测概率 P(U = 0| x = 0, y = 1, z = 0)是多少？

(b) 使用贝叶斯分类训练所得的概率，计算预测概率 P(U = 0|x = 0)？

5.10　参考书目

1. Berthold, M., D. J. Hand, eds., *Intelligent Data Analysis – An Introduction*, Springer, Berlin, 1999.

该书详细介绍了智能数据分析方法的主要分类，其中包括所有常见的数据挖掘技术。前半部分主要论述了古典派的统计观点，内容覆盖了从概率和推断的基本概念到高级的多元分析和

贝叶斯定理。第二部分是根据除统计学之外的其他学科对数据挖掘方法进行了理论解释。该书还列举了大量图表和例子，读者可更多地了解数据挖掘技术的理论和实际评估。

2. Brandt, S., *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*, 3rd edition, Springer, New York, 1999.

该书在统计理论和实际应用之间架起了一座桥梁。它强调简洁但严谨的数学，而且重点在于应用。在介绍了概率和随机变量后，接着讨论了随机数的形成和几种重要分布。后续章节讨论了统计样本、极大似然方法和统计假设的检验。书中包括了几种重要统计方法的详细讨论，例如最小二乘法，方差分析，回归和时间数列分析。

3. Cherkassky, V., F. Mulier, *Learning from Data: Concepts, Theory and Methods*, John Wiley, New York, 1998.

该书统一论述了发现数据相关性的原理和方法。它建立了一个全面的概念体系，在这个体系中，应用了各种学习方法，包括统计学、机器学习和其他学科，这表明几门基本学科是现在提出的大多数新方法的基础。这本基础理论书的另一个特色是它包含大量的案例研究，这些例子简化了统计学习理论的概念，使其更容易理解。

4. Hand, D., Mannila H., Smith P., *Principles of Data Mining*, MIT Press, Cambridge, MA,2001.

该书有 3 部分。第 1 部分是基础知识，概述了数据挖掘算法和应用原理。第 2 部分是数据挖掘算法，说明怎样构建算法，才能按照原理解决具体的问题。第 3 部分说明在解决实际的数据挖掘问题时怎样组合上述分析方法。

5. Nisbet, R., J. Elder, G. Miner, *Handbook of Statistical Analysis and Data Mining Applications*, Elsevier Inc., Amsterdam, 2009.

该书是一本内容全面的专业参考书，适用于商业分析员、科学家、工程师和研究人员(学术和行业)，其中包含数据分析的所有阶段、模型的建立和实现。该书有助于识别技术和商业问题，理解现代数据挖掘算法的优缺点，给实际的应用采用正确的统计方法。通过该书可以用科学的新方法处理大型复杂的数据集，并客观地评估分析结果。该书清晰、直观地解释了使用现代分析技术解决问题的原理和工具，讨论了它们在实际问题上的应用，各行业(包括科学、工程、医药、学术和商业)的从业人员均可接受，并能从中受益。该书包含了初学者理解数据挖掘工具和问题，建立成功数据挖掘方案所需的全部信息。

第 7 章

人工神经网络

本章目标

- 认识人工神经网络(ANN)的基本组成以及它们的属性和功能。
- 描述人工神经网络通常执行的学习任务，如模式关联、模式识别、估计、控制以及过滤。
- 比较不同的人工神经网络结构，如前向型和回馈型网络，并讨论它们的应用。
- 解释神经元层次的学习过程，以及由此扩展的多层、前向型神经网络。
- 比较前向型网络和竞争型网络的学习过程和任务。
- 了解 Kohonen 映射的基本原理及其应用。
- 讨论基于试探式参数调节的人工神经网络的通用性需求。

人脑的计算方式与传统的数字计算机截然不同，这一点一直激励着人工神经网络(Artifical Neural Network, ANN)的研究。对来自不同学科的众多研究者来说，对人脑的计算过程建模是一个巨大的挑战。人脑是一个高度复杂的、非线性的、并行处理信息的系统。它可以组织其组件，执行高质量、高速度的计算，其计算速度在很多情况下比当今最快的计算机要快很多倍。这些处理过程的例子有：模式识别、感知以及控制。自从 20 世纪 50 年代末期 Rosenblatt 第一次将单层感知机应用于模式分类学习以来，人们对人工神经网络的研究已经有 40 多年的历史了。

人工神经网络是人脑的抽象计算模型。人脑大约有 10^{11} 个微处理单元，叫做神经元。这些神经元之间相互连接，连接的数目大约是 10^{15}。和人脑一样，人工神经网络也是由相互连接的人工神经元(或者处理单元)组成的。将这个网络看作一个图表，神经元就可以表示为节点(或顶点)，神经元之间的相互连接表示为边。这些术语在人工神经网络中极其常见，这样的名字还包括“神经网络”、并行分布式处理(PDP)系统、连接模型或者分布式自适应系统，ANN 在文献中还称为神经计算机。

顾名思义，神经网络是一个很多节点通过有向链接组成的网络结构。每个节点代表一个处理单元，节点之间的连接表示所连接的节点之间的因果关系。所有节点都是自适应的，这就意味着这些节点的输出同这些节点的可修改参数值有关。虽然 ANN 概念有多种定义和方法，但

这里采用下面的定义，它将 ANN 看成形式化的自适应机。

定义：人工神经网络是一个大型并行分布式处理器，由简单的处理单元组成。它可以通过调整单元连接的强度，来学习经验知识，并运用这些知识。

显然 ANN 拥有强大的计算能力，首先，它有着庞大的并行分布式结构；其次，它有学习和归纳的能力。归纳是指 ANN 对学习过程中没有遇到过的新输入产生合理的输出。使用神经网络可以提供几种有用的属性和能力：

(1) **非线性**：神经网络作为基本单元，可以是线性的或非线性的处理元素，但是整个 ANN 是高度非线性的。ANN 分布在整个网络中，就这点而言，它是一种特殊的非线性。ANN 对现实世界中高度非线性的机理建模，以生成可供学习的数据时，这一特征尤其重要。

(2) **从样本中学习的能力**：通过应用一系列训练或学习样本，ANN 可以改变它的联接权重。学习过程的最终结果是调整好的网络参数(这些参数分布在所建模型的主要组件上)，它们隐含性地存储了当前问题的知识。

(3) **自适应性**：ANN 内置了随外部环境改变联接权重的能力。特别是，在某个环境下训练好的 ANN，在外部环境改变时，稍加训练就可以适应新的环境。而且，在动态环境中工作时，ANN 可以按照真实环境动态地改变其参数。

(4) **响应验证**：在数据分类环境中，ANN 不仅可以从给定的样本中提供某个类的信息，还可以在决策时提供置信度的信息。后者可以用来剔出模糊的数据，如果将其值增大，则可以提高分类的执行效率或者用神经网络进行建模的其他任务的执行效率。

(5) **容错性**：ANN 有固有的潜在容错能力，即计算的可靠性。它的执行效率在某些不利情形下并不会显著降低，比如，神经元断开了、有干扰数据或者数据丢失了。计算的可靠性有一些经验可以证实，但通常是不受控制的。

(6) **统一的分析和设计**：基本上，ANN 和信息处理器一样具有很好的通用性。在所有涉及 ANN 的应用领域中，都使用了相同的原理、符号，方法上也使用了相同的步骤。

要解释几种不同类型的 ANN 以及它们的基本原理，就必须介绍每个 ANN 的基本组成部分。这种简单的处理单元叫做人工神经元。

7.1 人工神经元的模型

人工神经元就是信息处理单元，它是 ANN 运转的基础。图 7-1 是人工神经元的示意图，它表明神经元是由 3 个基本元素组成的：

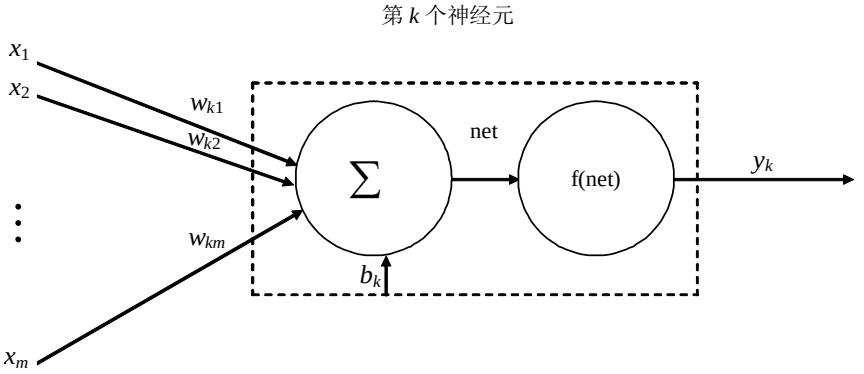


图 7-1　人工神经元模型

(1) 一组连接线。分别来自各个输出 x_i(或者叫做突触)，每条连接线上的权重为 w_{ki}。第一个下标是指当前的神经元，第二个下标是指权重所指向的突触的输入。一般来说，人工神经元的权重既可能是正值，也可能是负值。

(2) 加法器。将输入信号 x_i 与对应的突触权重 W_{ki} 相乘后进行累加。该操作会建立一个线性加法器。

(3) 激活函数 f。限制神经元输出值 y_k 的幅度。

图 7-1 所示的神经元模型还包括一个外部的偏差，用 b_k 来表示。偏差可能会增大或者减小激活函数的净输入，这取决于该偏差是负值还是正值。

在数学术语中，人工神经元是自然神经元的抽象模型，其处理能力用下面的符号来表示。首先，有一些输入 x_i，i=1,…,m。每个输入 x_i 和相应的权重 w_{ki} 相乘，其中 k 是 ANN 中给定神经元的索引。权重模拟了自然神经元中的生物突出强度。在 ANN 文献中，输入和相应权重乘积 x_iw_{ki}(其中，i=1,…,m) 的累加值，通常表示为 net。

$$net_k=x_1w_{k1}+x_2w_{k2}+\ldots+x_mw_{km}+b_k$$

用符号 w_{k0} 表示 b_k，默认输入 $x_0=1$，则 net 求和的统一形式为：

$$net_k=x_0w_{k0}+x_1w_{k1}+x_2w_{k2}+\ldots+x_mw_{km}=\sum_{i=0}^{m}x_iw_{ki}$$

同样，还可以用向量符号将 net 表示成两个 m 维向量的点积：

$$net_k=X\cdot W$$

其中

$X=\{x_0,x_1,x_2,\ldots,x_m\}$

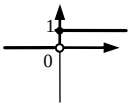
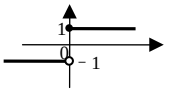
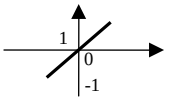
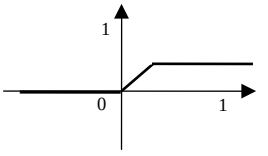
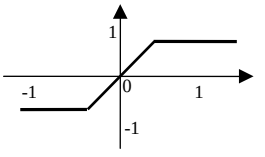
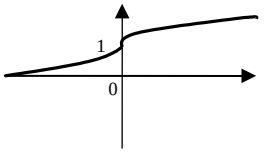
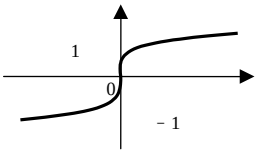
$W=\{w_{k0}, w_{k1}, w_{k2},\ldots,w_{km}\}$

最后，神经元把输出值 y_k 计算为 net_k 值的某个函数：

$$y_k=f(net_k)$$

该函数 f 叫做激活函数。可以定义各种各样的激活函数。一些常用的激活函数如表 7-1 所示。

表 7-1　神经元的常用激活函数

激活函数	输入输出关系	函数图
阶跃	$y=\begin{cases}1 & \text{如果 } net\geq 0\\ 0 & \text{如果 } net<0\end{cases}$	
对称阶跃	$y=\begin{cases}1 & \text{如果 } net\geq 0\\ -1 & \text{如果 } net<0\end{cases}$	
线性函数	y=net	
分段线性函数	$y=\begin{cases}1 & \text{如果 } net>1\\ net & \text{如果 } 0\leq net\leq 1\\ 0 & \text{如果 } net<1\end{cases}$	
对称分段线性函数	$y=\begin{cases}1 & \text{如果 } net>1\\ net & \text{如果} -1\leq net\leq 1\\ 0 & \text{如果 } net<1\end{cases}$	
对数 S 型	$y=1/(1+e^{-net})$	
双曲正切曲线	$y=(e^{net}-e^{-net})/(e^{net}+e^{-net})$	

现在，介绍人工神经元的基本组件及其功能时，可以分析单个神经元中的所有处理阶段。例如，一个神经元有 3 个输入和一个输出，相应的输入值和权重因子以及偏差如图 7-2(a)所示。需要求出各种激活函数的输出值 y，如对称阶跃函数、分段线性函数以及对数 S 型函数。

(1) 对称阶跃函数

$$net=0.5\cdot 0.3+0.5\cdot 0.2+0.2\cdot 0.5+(-0.2)\cdot 1=0.15$$

$$y=f(net)=f(0.15)=1$$

(2) 分段线性函数

net=0.15(计算同情形(1))

$$y=f(net)=f(0.15)=0.15$$

(3) 对数 S 型函数

$$net=0.15(\text{计算同情形(1)})$$
$$y=f(net)=f(0.15)=1/(1+e^{-0.15})=0.54$$

计算单个节点的基本法则可以扩展到有多个节点甚至有多个层的 ANN，如图 7-2(b)所示。假设给定了 3 个节点，所有的偏差均为 0，所有节点的激活函数都是对称阶跃函数。问节点 3 的最终输出 y_3 是多少？

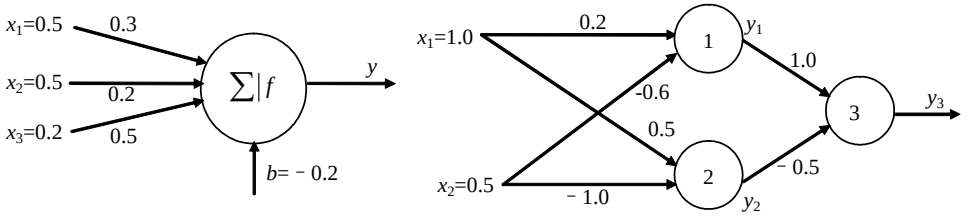
输入数据的处理是分层的。第一步，神经网络执行第一层中节点 1 和节点 2 的计算：

$$net_1=1\cdot0.2+0.5\cdot0.5=0.45\Rightarrow y_1=f(0.45)=0.45$$
$$net_2=1\cdot(-0.6)+0.5\cdot(-1)=-1.1\Rightarrow y_2=f(-1.1)=-1$$

第一层节点的输出 y_1 和 y_2 是第二层中节点 3 的输入：

$$net_3=y_1\cdot1+y_2\cdot(-0.5)=0.45\cdot1+(-1)\cdot(-0.5)=0.95\Rightarrow y_3=f(0.95)=0.95$$

从上面的例子中可以看出，在节点层次上的处理是非常简单的。在人工神经元高度连接的网络中，节点的数目每增加一个，计算的工作量会增加好几倍。处理的复杂性取决于 ANN 的结构。



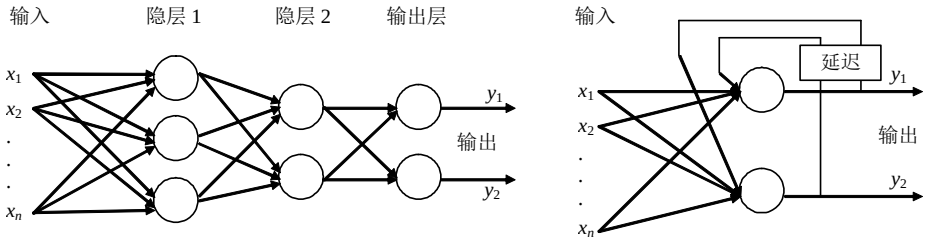
(a) 单个节点　　(b) 三个相互连接的节点

图 7-2　人工神经元以及它们之间的相互连接示例

7.2　人工神经网络的结构

人工神经网络的结构是通过节点的特性以及网络中节点连接的特性来定义的。上一节介绍了单个节点的基本特性，本节将介绍连接的参数。网络结构一般用网络的输入数目、输出数目、基本节点的总数(通常等于整个网络的处理单元数)，以及节点间的组织和连接方式来表示。按照连接的类型，神经网络通常分为两类：前向型和回馈型。

如果处理过程的传播方向是从输入端传向输出端，且没有任何的回环或反馈，该网络就是前向型。在分层的前向型神经网络中，同一层上的节点之间是没有相互连接的；某层上节点的输出总是作为下一层节点的输入。这种形式比较好，因为其具备模块性，即同一层上的节点具有相同的功能，或生成相同层次的输入向量。如果网络中有一个反馈连接，组成了封闭回路(通常有一个延迟单元作为同步组件)，这种网络就是回馈型的。两种类型的神经网络示例如图 7-3 所示。



(a) 前向型网络　　(b) 回馈型网络

图 7-3　人工神经网络的典型结构

虽然很多神经网络模型都可以归为这两类，但是有反向传播学系机理的多层前向型网络仍是在实际中运用得最为广泛的一种模型。可能有超过 90%的商业和工业应用软件都基于此模型。为什么是多层的网络呢？下面简单的示例展示了单层和多层神经网络之间在应用需求上的根本区别。

在神经网络著作中常常用最简单、最著名的分类问题来做示例，即异或问题。其任务是将二进制的输入向量 X 分成类别 0 和类别 1，如果该向量的偶校验值为 1，则为类别 0，反之则为类别 1。异或问题是不可线性分离的；这从图 7-4 中可以明显地看出来，该图的二维输入向量是 $X=\{x_1,x_2\}$。不可能对属于不同类的点进行线性分离。换句话说，不能用单层的网络构建直线(一般来说，它是 n 维空间上的一个线性超平面)，将二维输入空间划分成为两个部分，每个部分都只包含属于同一类的数据点。而使用两层的神经网络就可能解决该问题，如图 7-5 所示，图中展示了联接权重和阈值已知时的一种求解方式。该神经网络可在二维空间中产生一个非线性的分割点。

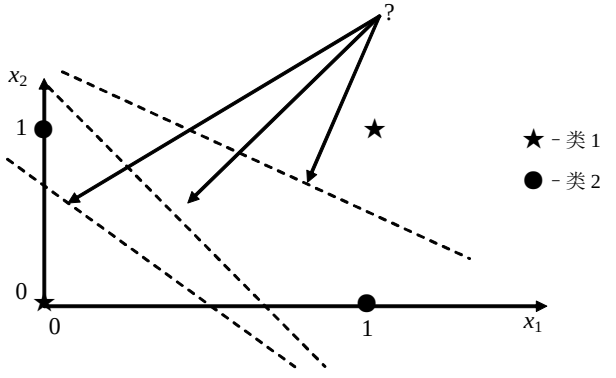


图 7-4　异或问题

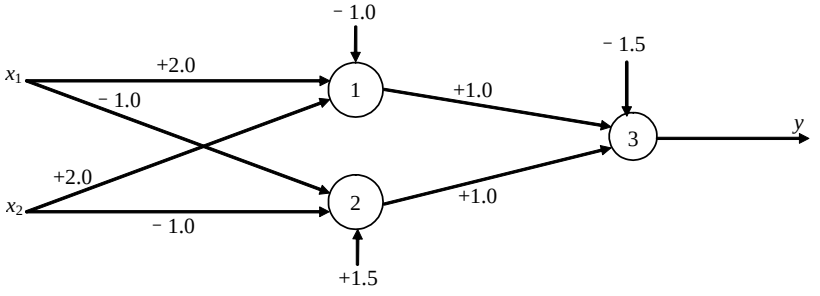


图 7-5　求解异或问题：使用阶跃激活函数的两层 ANN

该例的基本结论是：对基于线性模型的简单问题，单层的神经网络是一个方便的建模工具。但在绝大多数实际问题中，模型都是高度非线性的，多层神经网络是更好的解决方法，甚至可能是唯一的解决方法。

7.3　学习过程

人工神经网络的主要任务是学习现实世界(环境)中内嵌的模型，使所建的模型与真实世界具备高度一致性，以实现相关应用的特定目标。学习过程基于真实世界的数据样本，这是 ANN 设计与经典信息处理系统的根本区别。后者通常先根据环境观察的结果建立数学模型的公式，再用真实的数据验证模型，然后基于此模型构建(规划)系统。相对而言，ANN 的设计直接基于真实的数据，允许数据集“说明自己”。因此，ANN 不仅可以通过学习过程建立隐式的模型，还可以对感兴趣的信息进行处理。

人工神经网络最重要的特性是：网络可以从基于真实样本的环境中学习，并通过学习过程提高执行效率。ANN 通过应用于其连接权重的交互式调整过程来了解环境。理想情况下，学习过程每重复一次，网络对其环境的了解就增加一些。很难对学习这个术语下一个准确的定义。就人工神经网络来说，归纳学习的一个可能定义如下：

定义：学习是一个过程，在该过程中，神经网络的自由参数会随着神经网络所在环境的改变而自动调整。学习的类型是通过参数改变的方式来确定的。

用于解决学习问题的一组指定的、定义明确的规则称为学习算法。学习算法之间的主要不同是算法中调整权重的方式不同。在学习过程中需要考虑的另一个因素是 ANN 结构(节点和连接)建立的方式。

为了演示其中一个学习规则，考虑一个简单的神经元 k，如表 7-2 所示，它构成网络中唯一的运算节点。神经元 k 由输入向量 X(n)驱动，其中 n 表示离散时间，更精确地说，是调整输入权重 w_{ki} 的迭代过程中的时长。ANN 学习(训练)的每个数据样本组成输入向量 X(n)，其相应的输出为 d(n)。

表 7-2　输入和输出

	输入	输出
样本数据 $_k$	x_{k1}，x_{k2}，…，x_{km}	d_k

对输入向量 X(n)进行处理后，神经元 k 生成输出，用 $y_k(n)$表示：

$$y_k = f\left(\sum_{i=1}^{m} x_i w_{ki}\right)$$

它表示这个简单神经网络的唯一输出，并将它同期望响应或者样本中给出的目标输出 $d_k(n)$ 进行比较。输出产生的误差 $e_k(n)$定义如下：

$$e_k(n) = d_k(n) - y_k(n)$$

所产生的误差信号驱动了对学习算法的控制，其目的是对神经元的输入权重进行一系列校准

调节。校准调节的目的是通过一步步的迭代，使输出信号 $y_k(n)$ 越来越接近期望输出 $d_k(n)$。该目标可以通过将成本函数 $E(n)$最小化来实现，其中函数 $E(n)$是误差能量的瞬时值，这个例子的 $E(n)$用误差 $e_k(n)$来定义：

$$E(n)=1/2e_k^2(n)$$

基于成本函数最小化的学习过程称为误差纠正学习方法。特别是，由 $E(n)$的最小化而得到的学习规则通常称为δ规则或 Widrow-Hoff 规则。假设 $w_{kj}(n)$表示神经元 k 在输入为 $x_j(n)$、时长为 n 时的权重因子值。按照δ规则，调整量$\Delta w_{kj}(n)$定义如下：

$$\Delta w_{kj}(n)=\eta \cdot e_k(n)x_j(n)$$

其中η是一个数值为正的常量，它决定了学习率。因此，δ规则可以陈述为：对输入神经元连接的权重因子的调节同误差信号与该连接的输入值之积成正比。

计算调节量$\Delta w_{kj}(n)$之后，突触权重的新值是：

$$w_{kj}(n+1)= w_{kj}(n)+\Delta w_{kj}(n)$$

实际上，$w_{kj}(n)$和 $w_{kj}(n+1)$可以分别看成突触权重的新旧值 w_{kj}。从图 7-6 中可以看出，误差纠正学习是一个闭环反馈系统。由控制理论可知，此类系统的稳定性是由组成反馈环的参数决定的。其中一个重要的参数是学习率η。该参数必须精心选择，才能保证迭代学习过程的收敛稳定性。因此，实际上，该参数是决定误差纠正学习率的一个关键因素。

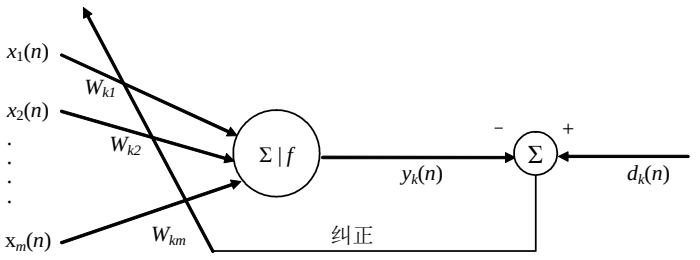
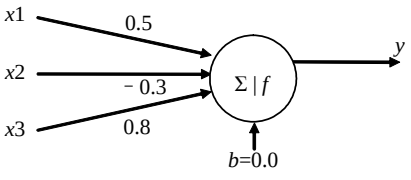


图 7-6　通过调整权重来进行误差纠正学习

下面对图 7-7(a)中单个人工神经元的学习过程作一个简单的分析，图 7-7(b)给出了 3 个训练(学习)样本。



(a) 有反馈的人工神经元

n(样本)	x_1	x_2	x_3	d
1	1.0	1	0.5	0.7
2	-1.0	0.7	− 0.5	0.2
3	0.3	0.3	− 0.3	0.5

(b) 学习过程的训练数据集

图 7-7　单个神经元的误差纠正学习过程的初始化

该神经元在调整权重因子的过程中，将设定学习率η=0.1。神经元的偏差为 0，激活函数是线性的。学习过程的第一次迭代(这里仅列出第一个训练样本的迭代过程)有以下步骤：

$$net(1)=0.5\cdot 1+(-0.3)\cdot 1+0.8\cdot 0.5=0.6$$

$$\Downarrow$$

$$y(1)=f(net(1))=f(0.6))=0.6$$

$$\Downarrow$$

$$e(1)=d(1)-y(1)=0.7-0.6=0.1$$

$$\Downarrow$$

$$\Delta w_1(1)=0.1\cdot 0.1\cdot 1=0.01 \Rightarrow w_1(2)=w_1(1)+\Delta w_1(1)=0.5+0.01=0.51$$
$$\Delta w_2(1)=0.1\cdot 0.1\cdot 1=0.01 \Rightarrow w_2(2)=w_2(1)+\Delta w_2(1)=-0.3+0.01=-0.29$$
$$\Delta w_3(1)=0.1\cdot 0.1\cdot 0.5=0.005 \Rightarrow w_3(2)=w_3(1)+\Delta w_3(1)=0.8+0.005=0.805$$

同样，可以处理第二个和第三次个样本(n=2 和 n=3)。表 7-3 中给出了学习过程中的调节量 Δ w 和新的权重因子 w。

表 7-3　图 7-7(b)中训练样本的权重因子的调整

参　　数	n=2	n=3
x_1	-1	0.3
x_2	0.7	0.3
x_3	-0.5	-0.3
y	-1.1555	-0.18
d	0.2	0.5
e	1.3555	0.68
$\triangle W_1(n)$	-0.14	0.02
$\triangle W_2(n)$	0.098	0.02
$\triangle W_3(n)$	-0.07	-0.02
$W_1(n+1)$	0.37	0.39
$W_2(n+1)$	-0.19	-0.17
$W_3(n+1)$	0.735	0.715

误差纠正学习可以应用在更为复杂的 ANN 结构上，在 7.5 节中将讨论其实现过程，此外，本节还介绍了具有后向传播的多层前向 ANN 的基本原理。本例仅展示了权重因子是怎样随每次训练(学习)样本数据而改变的。我们仅给出了第一次迭代的结果。无论使用新的训练数据还是相同的训练数据，权重的纠正过程都会进行。何时结束迭代过程是由一个或者一组特殊的参数来控制的，该参数叫做停止标准。学习算法可以有不同的停止标准，比如最大数目的迭代次数，或者是连续两次迭代权重因子改变的数值极限。该参数对于最后的学习结果非常重要，我们将在下一节中进行讨论。

7.4 使用 ANN 完成的学习任务

学习算法的选择是由 ANN 需要完成的学习任务决定的。这里明确 6 种使用不同人工神经网络完成的基本学习任务。这些任务是第 4 章介绍的一般学习任务的子任务。

7.4.1 模式联想

从亚里斯多德开始，联想就是人类记忆的一个显著特性，所有的模式识别都使用某种形式的联想作为基本操作。联想一般采取以下两种形式：自联想(autoassociation)和异联想(heteroassociation)。在自联想中，ANN 需要通过反复将模式提供给网络，来储存一组模式。然后，给 ANN 输入局部的描述或者歪曲的、有干扰的原始模式，ANN 必须找回或者回想起特定的模式。异联想与自联想的不同在于，任意一组输入模式都伴随着另外一组输出模式。自联想要使用无指导学习方式，而异联想需要有指导的学习方式。对自联想和异联想这二者来说，在应用 ANN 解决模式关联问题时，有两个主要阶段：

(1) 储存阶段，即神经网络按照给定的模式进行训练的阶段。

(2) 回想阶段，即根据输入网络的、有干扰和歪曲的关键模式，找出某个记忆下来的模式，作为响应。

7.4.2 模式识别

人脑执行模式识别的效率比最强大的电脑高得多。人们通过感知周围的世界获取数据，并能立即识别出数据的来源，而且常常毫不费力。人类通过学习过程执行模式识别；因此，人工神经网络也可以。

模式识别的正式定义是将获取的模式分配给一些数量已知的类中的一个。ANN 进行模式识别时，首先进入学习阶段，在学习阶段，神经网络反复接收一组输入模式以及每个模式所属的分类。然后，在检验阶段，给神经网络输入一个新的模式，神经网络以前没有见过这个新模式，但它属于训练阶段使用的模式所在的样本总体。神经网络能对该模式进行分类，因为它从训练数据中获取了信息。在图形上，模式用多维空间中的点来表示。整个空间叫做决策空间，它分成几个区域，每个区域对应一个类别。决策边界是由训练过程决定的。如果把未分类的新模式输入到神经网络中，必须对它们进行检验。本质上，模式识别是一个标准的分类任务。

函数近似

假设一个非线性的输入- 输出映射用下列函数关系式来描述：

$$Y=f(X)$$

其中向量 X 是输入，向量 Y 是输出。假设向量-数值函数 f 是未知的。给定一组带标记的

样本$\{X_i, Y_i\}$，要求设计一个神经网络，用函数 F 对未知函数 f 进行近似，用公式来表示：

$$|F(X_i)-f(X_i)|<\varepsilon \quad \text{对训练集中的所有 } X_i \text{ 都成立}$$

其中，ε是一个很小的正数。只要训练集足够大，神经网络有足够多的自由参数，近似误差ε就可以足够小。这里描述的近似问题非常适用于有指导学习过程。

控制

控制是人工神经网络可以完成的另一项学习任务。控制应用在系统的某个过程或者关键部分，该过程或者关键部分必须保持在控制的条件下。考虑如图 7-8 所示的有反馈的控制系统。

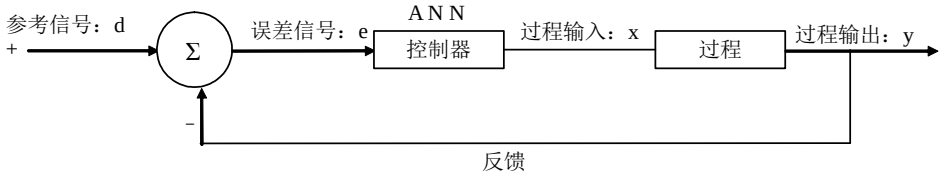


图 7-8　基于 ANN 反馈控制系统的结构图

该系统涉及使用反馈来控制外部源所提供的参考信号 d 层次上的输出 y。系统的控制器可以使用 ANN 技术来实现。误差信号 e 是过程输出 y 与参考信号 d 之差，它在基于 ANN 的控制器中用于调整其自由参数。控制器的主要目标是为过程提供合适的输入 x，使其输出 y 接近参考信号 d。可以通过下面的步骤来训练：

(1) 间接训练——首先在过程中使用实际的输入-输出测量值，在脱机情况下建立 ANN 控制模型。当训练结束时，ANN 控制器就可以放入实时的闭环中。

(2) 直接训练——训练阶段使用真实的数据实时进行，ANN 控制器可以直接从该过程中学习如何调节其自由参数。

过滤

过滤这个术语通常是指从一组杂乱的数据中提取某个量的信息的装置或者算法。在处理实时的、频繁的或者其他领域的系列数据时，可以使用 ANN 作为过滤器，执行下面 3 种基本的信息处理任务：

(1) 过滤——该任务是指通过在 *n* 时刻之前测量的数据(包括 *n* 时刻测量的数据)，在离散时刻 *n* 提取某个量的信息。

(2) 平整——该任务同过滤的不同之处在于，数据不一定仅在 *n* 时刻是可以获取的；晚于时刻 *n* 测量的数据也可以用于获取所需的信息。这就意味着，平整任务在离散时刻 *n* 生成结果时有一个延迟。

(3) 预测——预测的任务是预测将来的数据。其目标是通过在 *n* 时刻之前测量的数据(包括 n 时刻测量的数据)，推导出在将来的 $n+n_0$ 时刻某个量的信息，其中 $n_0>0$。预测可以看成一种建模方式：所做的预测偏差越小，神经网络就越适合作为负责生成数据的实际物理过程的模型。用于预测任务的 ANN 结构图如图 7-9 所示。

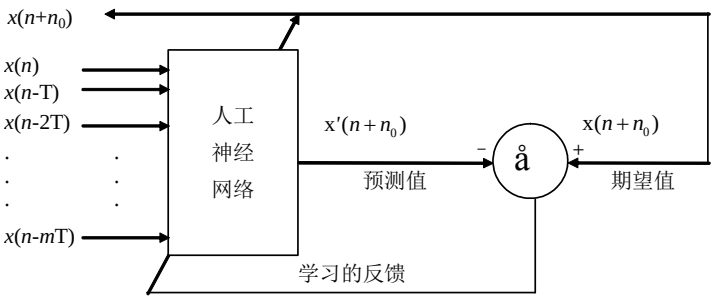


图 7-9　基于 ANN 预测的结构图

7.5　多层感知机

多层前向型神经网络是实际应用中最重要、最流行的一种 ANN。该神经网络一般包含组成网络输入层的一组输入、一个或多个具有计算节点的隐层和一个具有计算节点的输出层。处理过程是一层层地前向进行的。这类人工神经网络通常称为多层感知机(MLP)，MLP 代表简单感知机的概化，简单感知机是本章前面提到的单层神经网络。

多层感知机具有以下 3 个显著的特征：

(1) 神经网络中的每个神经元模型通常包含一个非线性的激活函数，S 型曲线或者双曲线函数。

(2) 神经网络包含神经元的一个或多个隐层，它们不是神经网络的输入或者输出的一部分。这些隐藏节点使神经网络从输入模式中不断获取有意义的特性，来学会高度非线性的复杂任务。

(3) 神经网络中的层与层之间具有高度的连接性。

图 7-10 是一个多层感知机的结构图，该多层感知机有两个用于处理的隐层节点及一个输出层。这个神经网络是全连接的，即神经网络中任何一层的神经元都和上一层的所有节点(神经元)相连接。神经网络中数据流的方向是前向的，从左到右，一层层地流动。

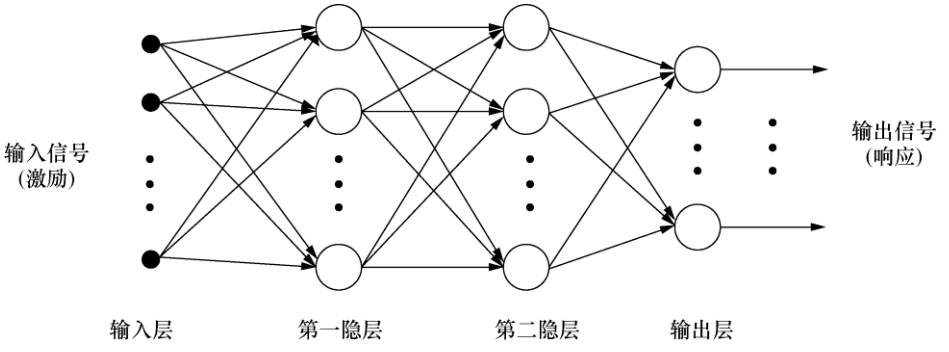


图 7-10　包含两个隐层的多层感知机的结构图

采用最普遍的误差后向传播算法，在监控条件下训练神经网络，多层感知机就可成功应用于解决一些多变的难题。这种算法基于误差纠正学习规则，可以看成由其衍生而来。基本上，误差后向传播学习过程由在神经网络中的不同层上执行的两个阶段组成：前向传播和后向传播。

在前向传播过程中，把训练样本(输入数据向量)应用到神经网络的输入节点，其作用在神经网络中一层一层传播。最后产生一组输出，作为神经网络的实际响应。在前向传播阶段，神经网络中所有突触的权重都是固定不变的。而在后向传播阶段，所有的权重都按照误差纠正规则进行调整。确切地讲，是用神经网络的期望(目标)响应减去实际响应，就生成了误差信号，其中，目标响应是训练样本的一部分。然后，这个误差信号沿着神经网络向后传播，与突触连接的方向相反。对突触的权重进行调整，使神经网络的实际响应逐渐接近期望响应。

在公式化后向传播算法时，先假设在第 *n* 次迭代过程中(即，输入第 *n* 个训练样本)，神经元 j 的输出存在误差信号。这个误差定义为：

$$e_j(n)=d_j(n)-y_j(n)$$

神经元 j 的瞬时误差能量值定义为 $1/2e_j^2(n)$。整个神经网络的总误差能量就是输出层上所有神经元的瞬时能量值之和。这仅仅考虑了误差信号可以直接计算的“可见”神经元。这样，总误差能量就可以写成：

$$E(n)=1/2\sum_{j\in C} e_j^2(n),$$

其中，集合 C 包括神经网络输出层上的所有神经元。用 N 来表示训练数据集中的样本总数，则要计算误差能量的平均方差，就应累加 N 个 E(*n*)，再用 N 进行标准化，表示如下：

$$E_{av}=1/N\sum_{n=1}^{N}E(n)$$

平均误差能量 E_{av} 是神经网络中所有自由参数的函数。对于给定的训练集，E_{av} 表示测量学习效率的成本函数。学习过程的目标是调整神经网络的自由参数，使 E_{av} 最小。要使 E_{av} 最小，在每次迭代时都需要根据每个样本来更新权重，即神经网络的整个训练集都会涉及到。

要使函数 E_{av} 最小，就必须使用节点处理层次上的两个附加关系，本章前面解释过它们：

$$V_j(n)=\sum_{i=1}^{m}w_{ji}(n)x_i(n)$$

和

$$y_j(n)=\varphi(V_j(n))$$

其中 m 是第 j 个神经元的输入数目。另外，使用符号 v 表示前面定义的变量 *net*。后向传播算法将一个纠正量 $\Delta w_{ji}(n)$ 应用到突触的权重 $w_{ji}(n)$，该数值与偏导数 $\partial E(n)/\partial w_{ji}(n)$ 成正比。使用微分链式法则，这个偏微分可以表达成下面的形式：

$$\partial E(n)/\partial w_{ji}(n)=\partial E(n)/\partial e_j(n)\cdot\partial e_j(n)/\partial y_j(n)\cdot\partial y_j(n)/\partial v_j(n)\cdot\partial v_j(n)/\partial w_{ji}(n)$$

偏微分 $\partial E(n)/\partial w_{ji}(n)$ 表示敏感因子，它确定在权重空间的搜索方向。已知下面的关系式是成立的：

$$\partial E(n)/\partial e_j(n)=e_j(n)\qquad(\text{因为 } E(n)=1/2\Sigma e_j^2(n))$$

$$\partial e_j(n)/\partial y_j(n)=-1\qquad(\text{因为 } e_j(n)=d_j(n)-y_j(n))$$

$$\partial y_j(n)/\partial v_j(n)=\varphi'(v_j(n)) \qquad (\text{因为 } y_j(n)=\varphi'(v_j(n)))$$
$$\partial v_j(n)/\partial w_{ji}(n)=x_i(n) \qquad (\text{因为}\Sigma w_{ji}(n)x_i(n))$$

于是，偏微分 $\partial E(n)/\partial w_{ji}(n)$ 可以表示成如下的形式：

$$\partial E(n)/\partial w_{ji}(n)=-e_j(n)\varphi'(v_j(n))x_i(n)$$

使用δ规则将纠正量$\Delta w_{ji}(n)$应用到 $w_{ji}(n)$：

$$\Delta w_{ji}(n)=-\eta\cdot\partial E(n)/\partial w_{ji}(n)=\eta\, e_j(n)\,\varphi'(v_j(n))\,x_i(n)$$

其中，η是后向传播算法的学习率参数，使用负号是考虑到权重空间中的梯度下降方向，即为减小 E(*n*)的数值而改变权重的方向。在学习过程中求出$\varphi'(v_j[n])$，可以解释为何在节点层次上要选择像 S 曲线和双曲线函数这样的连续函数作为标准的激活函数。使用符号 $\delta_j(n)=e_j(n)\cdot\varphi'_j(v_j[n])$，其中$\delta_j(n)$是局部梯度，$w_{ji}(n)$纠正量的最终方程为：

$$\Delta w_{ji}(n)=\eta\cdot\delta_j(n)\,x_i(n)$$

局部梯度$\delta_j(n)$指向突触权重需要的改变方向，按照它的定义，输出神经元 j 的局部梯度$\delta_j(n)$等于该神经元相应的误差信号 $e_j(n)$和相关激活函数的微分$\varphi'(v_j[n])$之积。

对于标准的激活函数，微分$\varphi'(v_j[n])$很容易计算，能够求导是对该函数的唯一要求。如果激活函数是 S 型曲线函数，其形式如下：

$$y_j(n)=\varphi(v_j(n))=1/(1+e^{(--vj(n))})$$

一次求导的结果是：

$$\varphi'\ (v_j[n])=e^{(-vj(n))}/(1+e^{(-vj(n))})^2=y_j(n)(1-y_j(n))$$

最终的权重纠正量为：

$$\Delta w_{ji}(n)=\eta\, e_j(n)\, y_j(n)\,(1-y_j(n))\,x_i(n)$$

最终的纠正量$\Delta w_{ji}(n)$与学习率η成正比，该节点的误差值为 $e_j(n)$，相应的输入和输出值为$x_i(n)$和 $y_j(n)$。因此，对于给定的样本 *n*，计算过程比较简单直接。

如果激活函数是双曲正切函数，执行相似的计算过程，可以得到一阶微分$\varphi'(v_j[n])$的结果：

$$\varphi'(v_j[n])=(1-y_j[n])\cdot(1+y_j[n])$$

和

$$\Delta w_{ji}(n)=\eta\cdot e_j(n)\cdot(1-y_j[n])\cdot(1+y_j[n])\,x_i(n)$$

还有，$\Delta w_{ji}(n)$的实际计算是非常简单的，因为局部梯度求导仅取决于节点的输出值 $y_j(n)$。

一般来说，根据神经元 j 在神经网络中的位置，可以确定$\Delta w_{ji}(n)$的两种不同计算情形。在第一种情形中，神经元 j 是一个输出节点。这种情形很好处理，因为神经网络的每个输出节点都有期望响应，可以直接计算相关的误差信号。前面提到的所有关系式对输出节点来说都是有

效的，不必进行任何的修改。

在第二种情形中，神经元 j 是一个隐含节点。即使隐含的神经元不是可以直接达到的，它们也会影响神经网络的输出结果的误差。隐含神经元 j 的局部梯度$\delta_j(n)$可以重新定义为相关导数$\varphi'(v_j(n))$与局部梯度的权重和之积，该局部梯度是为和神经元 j 相连接的下一层(隐层或者输出层)神经元计算的：

$$\delta_j(n)=\varphi'(v_j(n))\sum_k \delta_k(n)\cdot w_{kj}(n) \qquad k\in D$$

其中 D 表示与节点 j 相连接的下一层上所有节点的集合。由此追溯，在为临近输入层的指定节点计算局部梯度$\delta_j(n)$之前，下一层上所有节点的$\delta_k(n)$都是已知的。

再次分析一下后向传播学习算法的应用，它的两个传递过程对每个训练样本都是截然不同的。第一个传递过程叫做前向传递过程，神经网络的函数信号根据每个神经元来计算，从第一个隐层上的节点开始(输入层上没有计算节点)，然后是第二个隐层，在最终的输出层节点上计算结束。在这个过程中，神经网络根据每个学习样本的给定输入值计算其相应的输出。突触的权重在该过程中保持不变。

而第二个传递过程叫做后向传递过程，它从输出层开始，将误差信号(计算出的输出值和期望输出值之差)沿着神经网络一层一层地向左传递，递归地计算每个神经元的局部梯度δ。这个递归过程允许神经网络中突触的权重按照δ规则进行相应的改变。对于输出层上的神经元，δ等于该神经元的误差信号与其非线性激活函数的一阶导数之积。有了局部梯度δ，就可以直接计算输出节点的每个连接的Δw。若输出层中所有神经元的δ值都已知，就可以在上一层(通常是隐层)用它们来计算节点的局部梯度修改值(还不是最终结果)，然后纠正该层中输入连接的Δw。这个后向过程会重复进行，直到处理了所有的层，神经网络中的所有权重因子都修改了为止。然后，后向传播算法用新的训练样本继续进行。当再也没有新的训练样本时，第一轮的学习过程就结束了。对于同样的样本数据，可能会进行二次、三次、有时甚至是上百次迭代，得到的误差能量 E_{av} 才足够小，才能停止算法。

后向传播算法可以在权重空间中得到最速下降法计算出的“近似”轨线。学习率参数η设定得越小，突触的权重在神经网络的每次迭代中改变得就越小，权重空间的轨线也就越平滑。但是，这种改进是以降低学习效率为代价的。另一方面，如果为了加速学习过程，将学习率参数η设定得过大，由此导致的突触权重变化过大，使神经网络工作不稳定，于是，问题的解逼近最小点时，可能会产生震荡，永远达不到最小点。

为了提高学习率，同时避免震荡，一个简单的方法是修改δ规则，在其中加入一个动力因子：

$$\Delta w_{ji}(n)=\eta\cdot\delta_j(n)\cdot x_i(n)+\alpha\cdot\Delta w_{ji}(n-1)$$

其中α通常是一个正数，叫做动力常数，$\Delta w_{ji}(n-1)$是第$(n-1)$个样本的权重因子的调整量。α实际上通常设定在 0.1~1 之间。添加动力因子会使权重的变化更为平滑，防止权重因梯度干扰或误差平面的高空间频数导致的异常变化。但是，使用动力因子并不总是会加快训练的速度；这或多或少都与应用有一定的关系。动力因子表示一种平均的方法；而不是一种平均的结果。动力因子平均了权重的改变量。动力因子的含义是显而易见的：包括某种权重修正的惯量。若权重因子的修正是高度震荡的，且有符号的变化，则在后向传播算法中包含动力因子，可以起稳定作用。动力因子还可以防止学习过程在错误空间的浅层局部最小点处终止。

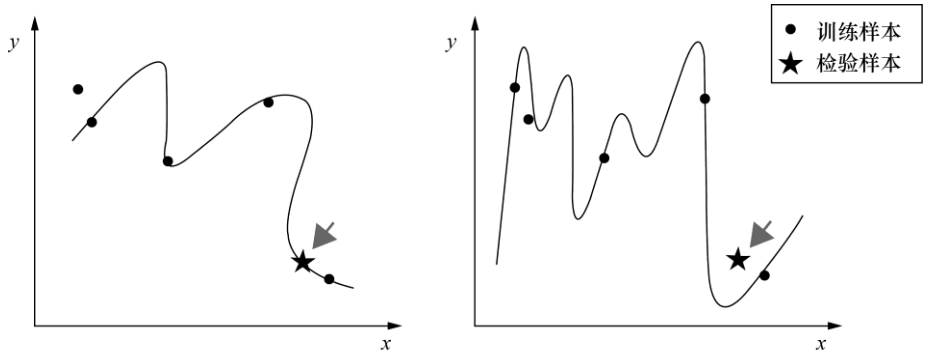
在为给定的任务确定神经网络的最优结构时，3 个参数的值非常重要：隐藏节点数(包括隐

藏层数)，学习率 η 和动力因子 α 。通常，最优结构根据实验决定，但有一些实用的指导方针。如果几个隐藏节点数不同的神经网络在训练后，误差标准方面的结果相近，最好的神经网络结构就是隐藏节点数最少的那个。实际上，这意味着先运用隐藏节点数小的神经网络开始训练过程，增加节点数，然后分析每次得到的误差。如果误差没有因隐藏节点数的增加而降低，最后分析的网络结构就是最优结构。最优的学习和动力常数也可以根据实验决定，但经验表明，η 约为 0.1，α 约为 0.5 时，可求出解。

首次建立人工神经网络时，必须给定初始权重因子。选择这些值的目的是尽快开始学习过程。正确的方法是将初始权重取为非常小的均布随机数。这样，无论其输入值为多少，输出值都处于中间范围，学习过程将随着每个新的迭代过程而更快收敛。

在后向传播的学习中，一般要使用尽可能多的训练样本，通过算法计算突触的权重。这样设计的人工神经网络能概化出最好的结果。对早期创建或训练神经网络时未使用的检验数据来说，当神经网络计算的输入输出映射是正确的时，该神经网络就概化得比较好。在多层感知机中，如果隐藏单元数小于输入数，第一层就进行维归约。每个隐藏单元都可以解释为定义一个模板，分析这些模板，就可以从训练好的人工神经网络中获取知识。在这个解释中，权重定义了模板中的相对重要性。但训练样本最多，使用这些样本进行的学习迭代次数最多，并不一定会产生最好的结论。学习过程会出现其他问题，下面简单地分析它们。

使用人工神经网络的学习过程可以看成曲线拟合问题。这样就允许将概化视为输入数据令人满意的非线性插值，而不是神经网络的理论属性。用于概化的人工神经网络会产生正确的输入输出映射，即使输入与用于训练神经网络的样本略微不同，也是如此，如图 7-11(a)所示。然而，当人工神经网络学了过多的输入输出样本时，就可能停止存储训练数据。这种现象称为超拟合或训练过度。这个问题在第 4 章中讲过。神经网络训练过度后，就会丧失概化类似模式的能力。另一方面，输入输出映射的平稳度和人工神经网络的概化能力密切相关。其要点是以训练数据为基础，为概化选择最简单的函数，也就是为给定的误差标准模拟映射的最平稳的函数。根据所研究现象的范围，平稳度是许多应用的自然要求。因此必须找到平稳的非线性映射，神经网络才能根据训练模式正确地给新模式分类。图 7-11(a)和 7-11(b)分别显示了用于同一组训练数据的一条概化较好的拟合曲线和一条超拟合曲线。



(a) 概化较好的拟合曲线　　(b) 超拟合曲线

图 7-11　曲线拟合问题的一般化

为了克服超拟合问题，可以为人工神经网络(特别是多层感知机)的设计和应用引入另外一

些实用的建议。在人工神经网络中，与所有的建模问题一样，应使用能够充分体现训练数据集的最简单的神经网络。如果小神经网络有效，就不要使用大神经网络！使用最简单神经网络的另一个备选方案是在神经网络超拟合之前就停止训练。还有一个重要的约束是，必须限制神经网络参数的数量。能概化训练集的神经网络，它拥有的参数必须少于训练集中的数据点(非常重要)。如果有大量的输入空间，而训练样本很少，则人工神经网络的概化能力将非常糟糕。

数据挖掘模型(包括人工神经网络)的可解释性，或者模型中输入输出关系的理解方式，是数据挖掘应用研究中的一个重要属性，因为这种研究的目标是获得基本推理机制的知识。这些解释也可用于验证与当前问题的一般理解不一致或相背的结果，它还可以指出数据或模型的问题。

人工神经网络主要在分类和回归问题中研究并成功运用，但它们的可解释性仍很含糊。其缺点是它们是“黑盒子”，即没有明确的机制来确定为什么人工神经网络会做出某个决策。也就是说，给人工神经网络提供输入值，会得到一个输出值，但一般不知道这些输出是如何得到的，输入值和输出值有什么关系，神经网络中大量的权重因子是什么。人工神经网络要成为商业和研究领域中有效的数据挖掘方法，不仅要提供准确的预测，还要提供各种用户(临床医生、决策者、商务策划师、知识分子和非专业人士)都能理解的有意义的知识。如果输入输出关系很明确，人们的理解和接受程度就会大幅提高，最终用户对预测结果也更有信心。

训练好的人工神经网络可以用两种方式解释：简略和详细。简略解释的目的是指出输入神经元对模型预测能力的重要程度如何。这类解释可以按重要性给输入属性排序。简略解释本质上是对神经网络的敏感性分析。该方法并没有指出每个输入神经元的影响效果。因此，无法总结出输入描述符和神经网络输出之间的关系，只能确定输入神经元对模型的影响程度。

详细解释人工神经网络的目标是从人工神经网络模型中找出结构属性的趋向。例如，每个隐含神经元对应的分段超平面有多少个，这些分段超平面可用于逼近目标函数，它们是构建显式 ANN 模型的基本构建块。为了得到逼近 ANN 行为的、更容易理解的系统，模型应不太复杂，但这可能会影响结果的准确性。ANN 复杂结构中隐含的知识可以使用各种方法来发现，以便把 ANN 映射到基于规则的系统上。许多人都致力于将拓扑结构中发现的知识和神经网络的加权矩阵合并为一套符号，其中一些编辑为普通的 if-then 规则集，其他则编辑为命题逻辑公式或非单调逻辑公式，或者模糊规则集。这些转换将训练好的神经网络所发现的隐含知识变成显式知识，允许人类专家理解神经网络如何生成某个结果。要强调的是，只有所提取的规则对人类专家是有意义的、可理解的，从人工神经网络中提取规则的方法才是有价值的。

事实证明，对于用连续型激活函数训练的人工神经网络，基于模糊规则集的系统是其最好的解释。通过这种方式，可以更全面地描述人工神经网络的行为。多层前向型人工神经网络可以看成基于模糊规则集的附加系统。在这些系统中，每个规则的输出都用规则的激活程度作为权重，接着把它们加起来，作为人工神经网络模型的集成表示。大多数通过模糊规则来逼近神经网络的技术的主要缺点是，要获得好的逼近效果，所需的规则数目会呈指数增长。模糊规则表示人工神经网络的输入输出映射，它们使用大量参考文献中描述的不同方法来提取。如果读者对这些方法感兴趣，可从本章末尾的推荐参考书目开始，也可以从第 14 章介绍的模糊系统概念开始。

7.6 竞争网络和竞争学习

竞争神经网络属于一种回馈型网络，它们是以无指导学习算法为基础的，如本节介绍的竞争算法。在竞争学习中，神经网络的输出神经元互相竞争被激活的机会。在多层感知机中，几个输出神经元可以同时激活，而在竞争学习中，一次只能激活一个输出神经元。为了构建带有竞争学习规则的神经网络(这是此类人工神经网络的标准技术)，有 3 个基本元素是必需的：

(1) 一组神经元：它们具有相同的结构、且与最初随机选择的权重连接起来。因此，对于给定的输入样本集，神经元可以有不同的响应。

(2) 一个极限值：决定每个神经元强度的极限值。

(3) 一个机制：允许神经元竞争对给定输入子集的响应，这样每次只能激活一个输出神经元。竞争成功的神经元称为“赢家通吃”神经元。

在竞争学习的最简单形式中，人工神经网络有一个输出神经元层，每个输出神经元和输入节点完全相连。神经网络可以包括神经元之间的反馈连接，如图 7-12 所示。在这个神经网络结构中，反馈连接执行侧向抑制，每个神经元都禁止侧面连接。相反，图 7-12 所示神经网络中的前向突触连接都是可以激活的。

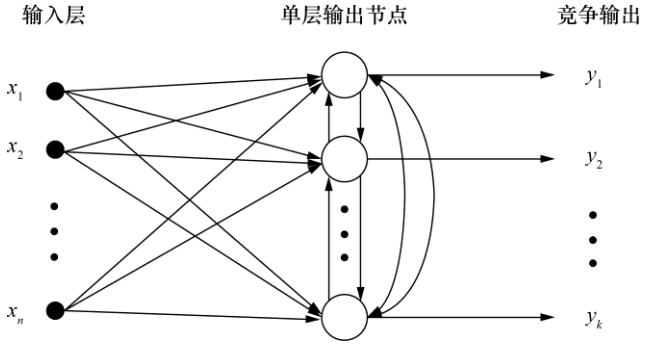


图 7-12　简单竞争性网络的结构图

因神经元 k 竞争成功，所以在神经网络中，对于输入样本 $X=\{x_1,x_2,\ldots,x_n\}$，它的网络值 net_k 必须是所有神经元中最大的。获胜神经元 k 的输出信号 y_k 设为 1；竞争失败的所有其他神经元的输出设为 0。因此：

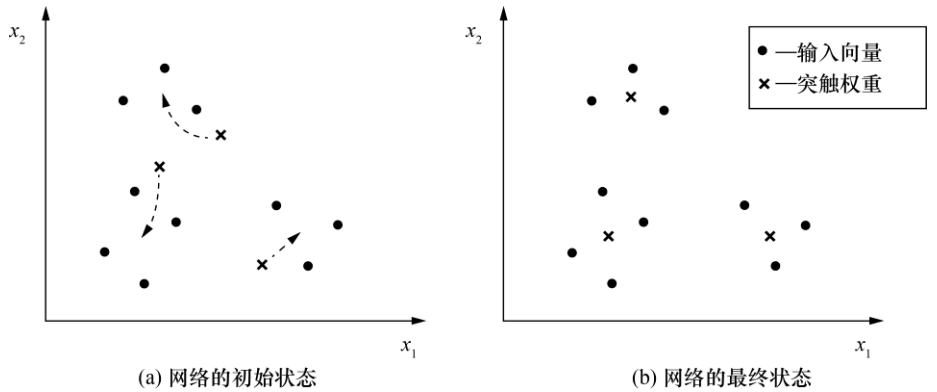
$$y_k=\begin{cases}1 & net_k > net_j \quad \text{对于所有的} j, j\neq k \\ 0 & \text{其他情形}\end{cases}$$

其中，推导出的局部值 net_k 表示输入节点到神经元 k 的所有前向和反馈连接的总和。

设 w_{kj} 表示连接输入节点 j 和神经元 k 的突触权重。于是，神经元将突触权重从非活动输入节点移动到活动输入节点上，以进行学习。如果某个神经元赢得竞争，该神经元的每个输入节点就让渡一定比例的突触权重，接着，把所让渡的权重分配到活动的输入节点中。根据标准的竞争学习规则，运用于突触权重 w_{kj} 的改变量 Δw_{kj} 定义为：

$$\Delta w_{kj}=\begin{cases}\eta(x_j-w_{kj}) & \text{如果神经元 k 赢得竞争} \\ 0 & \text{如果神经元 k 未赢得竞争}\end{cases}$$

其中η为学习率参数。该规则的总体效果是将获胜神经元的突触权重移向输入模式 X。图 7-13 中描述的几何类比法可以说明竞争学习的本质。



(a) 网络的初始状态　　(b) 网络的最终状态

图 7-13　竞争学习的几何解释

在发现一类输入样本时，每个输出神经元就将其突触权重移至所发现的类的重心。图 7-13 说明，神经网络可以通过竞争学习进行聚类。在竞争学习的过程中，神经网络将类似的样本组合起来，用输出上的一个人工神经元来描述。这种基于数据相关性的分组是自动完成的。然而，为了使此功能执行稳定，输入样本必须属于截然不同的组。否则，神经网络可能不稳定。

竞争(或赢家通吃)神经网络常用于聚类输入数据，且事先给定了输出类的数量。根据无人指导的归纳学习进行聚类的著名人工神经网络例子包括 Kohonen 的学习向量量化(learning vector quantization，LVQ)，自组织映射(self-organizing map，SOM)和基于适应回响理论模型的网络。本章讨论的竞争网络与汉明(Hamming)网络密切相关，所以下面复习一下这个常见、又非常重要的人工神经网络类型的关键概念。汉明网络包含两层。第一层是标准的前向层，它将输入向量和预处理的输出向量关联起来。第二层执行竞争，以决定哪个预处理的输出向量最接近输入向量。第二层上具有稳定的正输出的神经元就是竞争的胜利者，其索引就是和输入最吻合的原型向量的索引。

竞争学习可以进行有效的适应性分类，但它有一些方法上的问题。第一个问题是，选择学习率η时，必须平衡学习的速度和最终权重因子的稳定性。学习率接近 0，学习速度就较慢。然而，一旦权重向量达到聚类的中心，它将稳定在中心附近。相反，学习率接近 1，学习速度就较快，但学习的效果不稳定。各个聚类比较接近时，则会产生一个更严重的稳定性问题，权重向量也会非常接近，对于每个新样本，学习过程都会改变它的值及与对应的类。若神经元的初始权重向量离所有输入向量都太远，永远不会赢得竞争，也就永远不会进行学习，此时也可能出现竞争学习的稳定性问题。最后，竞争学习过程的输出神经元数量总是和它的聚类数量相同。这不适用于一些应用，特别是聚类的数量未知或很难事先估计时。

下面的例子将跟踪竞争网络的计算步骤和学习过程。假定一个竞争网络具有 3 个输入和 3 个输出。其任务是把一组三维输入样本集分为 3 类。网络是完全连接的，所有的输入和输出之间都有连接，输出节点之间还有侧面连接。只有局部反馈权重等于 0，这些连接不在最后的网络结构中表示出来。输出节点基于线性激活函数，其中所有节点的偏差值都等于 0。图 7-14 给出了所有连接的权重因子，并假定该网络已经用前面的一些样本训练过。

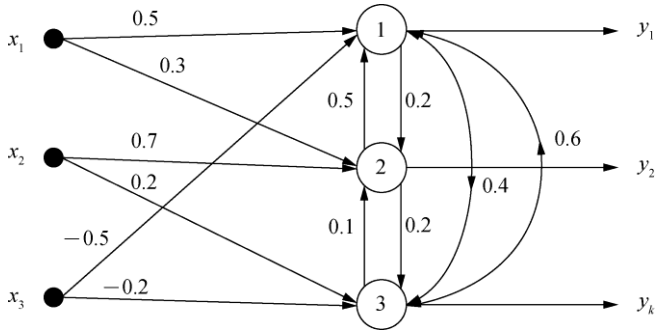


图 7-14　竞争神经网络实例

假定新样本向量 X 的分向量：

$$X=\{x_1,x_2,x_3\}=\{1, 0, 1\}$$

首先，在前向阶段，竞争的临时输出通过它们激活的连接来计算，它们的值是：

$$net_1^* = 0.5\cdot x_1 + (-0.5)\cdot x_3 = 0.5\cdot 1 - 0.5\cdot 1 = 0$$
$$net_2^* = 0.3\cdot x_1 + 0.7\cdot x_2 = 0.3\cdot 1 + 0.7\cdot 0 = 0.3$$
$$net_3^* = 0.2\cdot x_2 + (-0.2)\cdot x_3 = 0.2\cdot 0 - 0.2\cdot 1 = -0.2$$

包括侧面抑制连接后：

$$net_1 = net_1^* + 0.5\cdot 0.3 + 0.6\cdot(-0.2) = 0.03$$
$$net_2 = net_2^* + 0.2\cdot 0 + 0.1\cdot(-0.2) = 0.28(\text{最大值})$$
$$net_3 = net_3^* + 0.4\cdot 0 + 0.2\cdot 0.3 = -0.14$$

输出间的竞争表明，最大的输出值是 net_2，它是胜利者。因此对于已知样本，网络的最终输出是：

$$Y=\{y_1,y_2,y_3\}=\{0, 1, 0\}$$

使用相同的样本，在竞争学习的第二阶段，开始权重因子的纠正过程(仅对获胜节点 y_2)。根据学习率 η=0.2，网络修正的结果是新的权重因子：

$$w_{12} = 0.3 + 0.2(1-0.3) = 0.44$$
$$w_{22} = 0.7 + 0.2(0-0.7) = 0.56$$
$$w_{32} = 0.0 + 0.2(1-0.0) = 0.20$$

网络中的其他权重因子保持不变，因为它们的输出节点不是本例中竞争的胜利者。新权重仅是一个样本的竞争学习过程的结果。对于大型训练数据集，这个过程会不断重复。

7.7　SOM

SOM，通常称为 Kohonen 映射，是由赫尔辛基大学 Teuvo Kohonen 教授提出的一种数据可视化技术。SOM 的主要思想是将 n 维输入数据投影到某些表示上，以获得更直观的理解，

例如二维图像映射。SOM 算法不仅是可用于可视化的试探式模型，而且也探讨了高维数据集的线性和非线性关系。1980 年，SOM 最早应用于语音识别问题，后来逐渐成为基于聚类和分类的各种应用中广泛采用的方法。

数据可视化试图解决的问题是：人类很难形象化地考虑高维数据。SOM 技术能够帮助人们形象化地理解这些高维数据的特点。SOM 的输出能够强调数据的显著特征，并以此自动化地组织相似数据项的聚类。SOM 是一种无监督(无指导)神经网络，通过可视化聚类解决聚类问题。作为一种学习过程的结果，由于能够给出数据的完整的图示，SOM 被当作是一种重要的可视化及数据简化方法。相似数据项转换到低维后，仍然能够自动地分在同一组中。

SOM 实现维度简化的方法是通过建立一到二个维度的输出映射，通过在映射中分组数据项反映数据的相似性。通过上述转换，SOM 实现了两个目标：简化维度并显示相似性。在保留输入样本拓扑关系的同时，以低维空间表示复杂的高维数据。

基本 SOM 可被视为神经网络，其节点可按照各种不同的输入样本模式或模式类以有序方式进行具体地调整。连接边带有权重的节点有时称为神经元，因为 SOM 实际上是一种特殊类型的人工神经网络。SOM 通常是一种单层前馈网络，其输出神经元通常使用二维拓扑网格表示。输出网格可以是矩形或六边形。在前一种情况下，除边和角意外的每个神经元有四个最近邻，第二种情况下包含六个最近邻。尽管六边形网格需要更多的计算工作，但它能够提供更平滑的结果。每个输出神经元都附加一个与输入空间具有相同维数的权重向量。节点 i 对应的权重向量为 $w_i = \{w_{i1},w_{i2},\ldots,w_{id}\}$，其中 d 为输入特征空间的维数。

SOM 输出的结构可以是一维数组或二维矩阵，但也可以是更复杂的三维结构，例如圆柱体或环面体。图 7-15 展示了一个简单的 SOM 架构，包含所有输入和输出之间内部连接的实例。

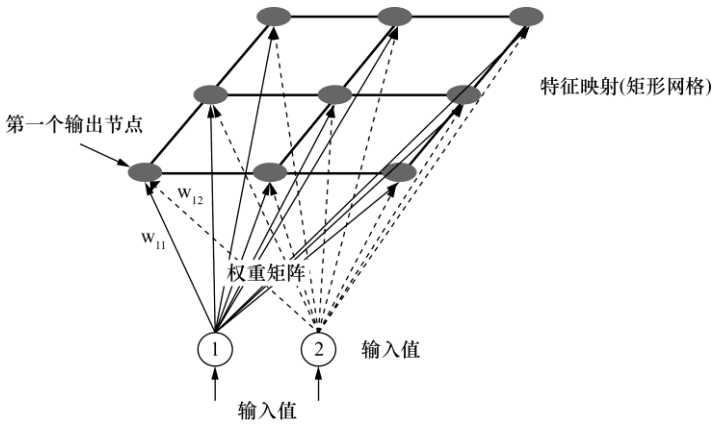


图 7-15　二维输入、3×3 输出的 SOM

SOM 技术一个重要部分是数据。这些数据是 SOM 用来学习的样本。学习过程是竞争性和无监督的，意味着不需要指导来定义给定输入的正确输出。竞争性学习用于训练 SOM，即输出神经元之间展开竞争以共享其输入数据样本。获胜的神经元，权重表示为 w_w，该神经元是在定义的所有 m 个神经元中，其度量最接近输入样本 x 的神经元。

$$d(x,w_w)=\arg\min_{1\leqslant j\leqslant m} d(x,w_j)$$

SOM 基本算法中，对应每个输入，每次仅包含一个输出节点(获胜者)。赢家通吃方法缩减了获胜节点权重向量与当前输入样本的距离。使节点更能“表达”样本。SOM 学习过程是

一个发现网络(权重 w)参数，以便将数据组织到能够保持拓扑结构的聚类的迭代过程。算法获得适当的方法将高维数据投影到低维数据空间上。

SOM 学习的第 1 个步骤是初始化神经元的权重，广泛采用的方法有两种。初始权重被视为从数据集中随机选择的 m 个点的坐标(通常规范化为 0-1 之间的值)，或者从输入数据子空间中按照两个最大的主成分特征向量均匀抽样的随机值。第二种方法可以提高训练速度，但可能会导致局部最小化，丢失数据间的非线性结构。

初始化后执行学习过程，每个训练数据将被顺序地提交给 SOM，通常需要进行几次迭代。每个输出及连接关系称为一个细胞元，是一个包含与输入样本匹配的模板的节点。所有输出节点与同样的输入样本并行地比较，SOM 计算每个细胞元与输入的距离。所有细胞元比较大小，从而确定输入与模板之间距离最近的细胞元，并建立活动输出。每个节点类似针对同样输入样本的不同的译码器或模式监测器，获胜的节点称为最佳匹配单元(BMU)。

在给定输入样本的获胜节点确定后，学习阶段适应 SOM 的权重向量。每个输出对权重向量的适应通过类似竞争学习的过程，只是不包括节点子集适应每个学习的步骤，以便建立拓扑有序的映射。获胜节点 BMU 调整其与训练输入的权重向量，这样能够变得更加敏感，如果在训练后出现在网络上，能提供最大的响应。获胜节点的近邻集合中的节点也必须以同样的方式进行修改以建立能够响应有关样本的节点区域。图 7-16(a)给出了 SOM 的二维矩阵输出实例。对给定的 BMU，其近邻被定义为围绕 BMU 的 3×3 矩阵节点。

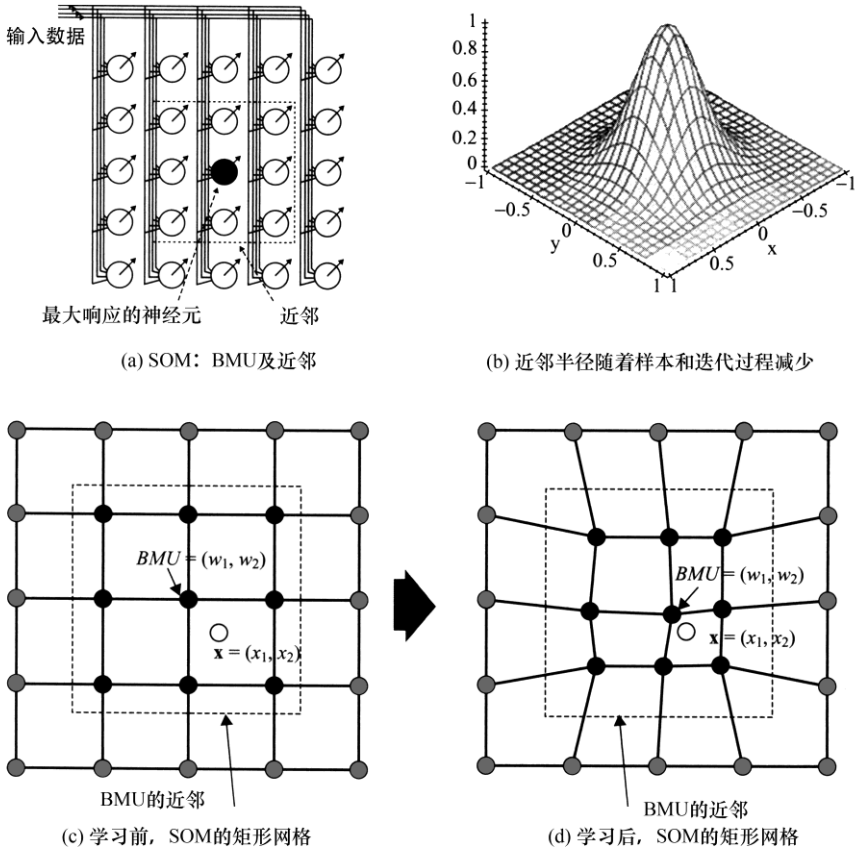


图 7-16　SOM 学习过程特征

属于 BMU 近邻的每个节点(包括 BMU)，在迭代过程中按照如下等式调整其权重向量：

$$w_i(t+1) = w_i(t) + h_i(t)[x(t) - w_i(t)]$$

其中 $hi(t)$是近邻函数。它定义为时间 t 的函数，或更精确地一个训练迭代，定义了第 i 个神经元的近邻区域。实验表明，为获得映射的全局顺序，围绕获胜节点的近邻集合开始时应该较大，以便快速产生粗略的映射。随着对训练集合数据迭代次数的增加，近邻会不断减少以便更能适应网络。这一工作完成后，输入样本可以首先移动到可能的 SOM 区域，然后将更准确地确定位置。该过程大致方式是首先进行粗略调整，然后进行精细调整(图 7-17)。因此 BMU 近邻的半径是动态的。为此，SOM 可以使用多种方法，例如使用在每个新迭代过程中动态缩短半径的指数衰减函数。有关该函数的图形化解释可以参考图 7-16(b)。

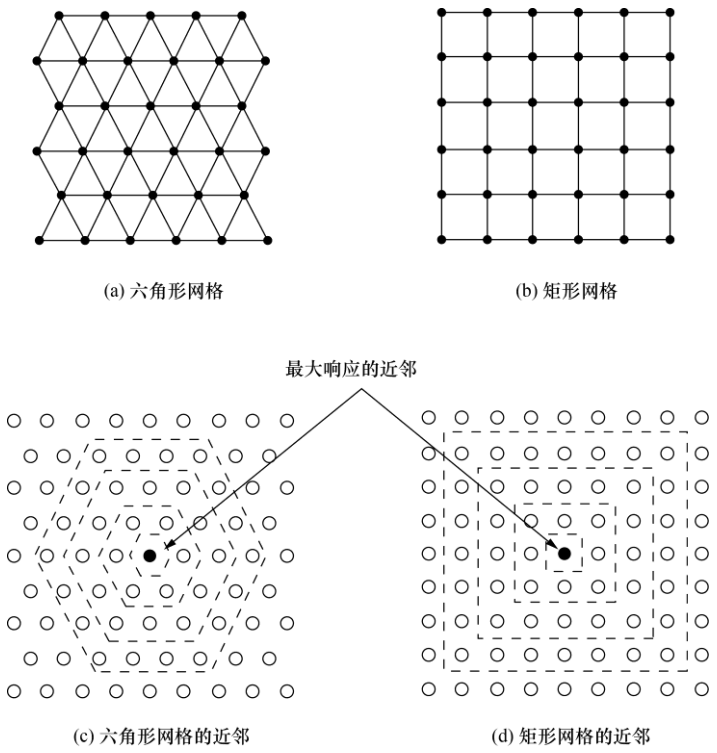


图 7-17　对近邻粗略调整之后的精细调整

最简单的近邻函数涉及围绕 BMU 节点 i 的节点的近邻集合，是一个单调递减的高斯函数。

$$h_i(t)=a(t)\exp\left(\frac{d(i,w)}{2s^2(t)}\right)$$

其中，$\alpha(t)$是学习率(0<α(t)<1)，核 $\sigma(t)$的宽度是随时间的单调递减函数，t 是当前时间步(循环迭代)。过程将适应所有当前近邻区域内的所有权重向量，包括那些获胜的神经元，而近邻以外的神经元保持不变。初始半径设置较大，其中一些值接近映射的宽度和高度。结果导致较早训练阶段中，近邻涉及面广几乎涵盖所有神经元，自组织发生在全局范围内。随着迭代的不断开展，基点逐渐向中心靠近，随着时间推移，近邻数量不断减少。训练结束时，近邻减少到 0，只有 BMU 神经元权重被更新了。通过组织空间上靠近 SOM 输出的相似向量(先

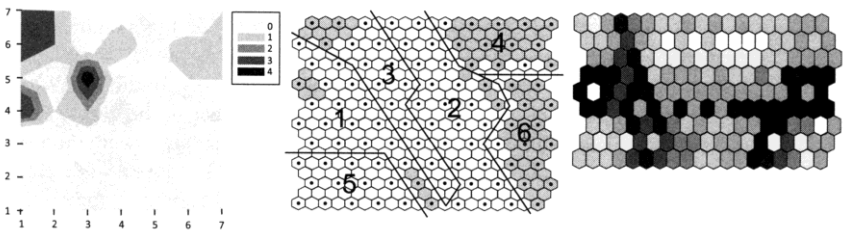
前未见的)的过程对网络进行泛化。

除了能够简化近邻外，如果网络中节点的适应率随着时间减少，SOM 算法还能够更加快速地收敛。初始阶段适应率比较高以生成粗略的聚类节点。一旦建立了粗略的表示，适应率开始降低，每个节点的权重向量的变化减小，映射区域能够对输入训练向量进行精细的调整。包括 BMU 在内的 BMU 近邻内的每个节点在学习过程中调整了权重向量。先前等式中的权重因子更正 $h_i(t)$可以包含“获胜者影响”的指数递减，引入的 $\alpha(t)$仍然是单调递减函数。

SOM(即，映射大小)中的输出神经元数量对于检测数据的误差来说是非常重要的。如果映射尺寸太小，可能难以解释一些输入样本之间存在的重要的差别。 相反，如果映射尺寸太大，差别又太小。在实际应用中，如果不存在额外的试探式规则，SOM 的输出神经元的数量可以根据不同的 SOM 结构的迭代来选择。

SOM 的主要优点在于：表示结果非常容易理解和解释，技术上容易实现。更重要的是，在许多实际应用中起到很好的作用。当然，SOM 也存在一些缺点，SOM 计算复杂度高，对相似性的度量非常敏感，并且不能应用于存在缺失值的数据集中。对 SOM 可以进行几种改进。为减少学习过程的迭代次数，对权重因子的初始化工作进行优化非常重要。输入数据的主要成分可以使 SOM 的计算量显著增加。实际经验表明，六角形网格给出的输出结果质量更好。最后，距离度量方法的选择对任何聚类算法来说都是非常重要的。尽管欧几里德距离几乎可以作为标准，但并不意味着使用它总能获得最佳结果。为提高显示效果(各向同性)，建议采用六角形作为 SOM 的基本单元。

SOM 已经用于大量实际应用，例如自动语音识别、诊断数据分析、监视工业生产和过程的状况，分类卫星图像、分析基因信息、分析脑电信号以及从海量文档中检索。图 7-18 给出了说明性的示例。



(a) 与人类细胞色素结合的药物　　(b) 利率分类　　(c) 购书行为分析

图 7-18　SOM 应用

7.8　复习题

1. 说明人工神经网络设计和“传统”信息处理系统设计之间的基本区别。
2. 为什么容错性是人工神经网络最重要的特性和功能之一？
3. 神经元模型的基本组成是什么？
4. 为什么对数 S 形、正切双曲线等连续函数是人工神经网络实际应用中的常用激活函数？
5. 讨论前向和回馈型神经网络之间的区别。
6. 一个神经元有两个输入和下列参数：偏差 b=1.2，权重因子 W=[w1, w2]＝[3, 2]，输入向

量 X＝$[-5,6]^T$，为下列激活函数计算神经元的输出：

(a) 对称阶跃

(b) 对数 S 形

(c) 正切双曲线

7. 一个神经元有两个输入，其权重因子 W 和输入向量 X 如下：

$$W=[3,2] \quad X=[-5,7]^T$$

要求输出为 0.5，则：

(a) 如果偏差为 0，能否用表 7-1 的某个转换函数完成这项工作？

(b) 用线性转换函数完成这项工作，是否存在偏差？

(c) 用对数 S 形激活函数完成这项工作，偏差是多少？

8. 一个分类问题用表 7-4 所示的三维样本 X 定义，其前两维是输入，第三维是输出。

表 7-4　三维样本 X

X:	I_1	I_2	O
	-1	*1*	*1*
	0	*0*	*1*
	1	*-1*	*1*
	1	*0*	*0*
	0	*1*	*0*

(a) 根据类型画出数据点 X 的标注图。该分类问题可以用单一神经元感知机来解决吗？解释原因。

(b) 画出用于解决问题的感知机图。为所有的网络参数定义初始值。

(c) 运用δ学习算法的一次迭代。权重因子的最终向量是什么？

9. 训练单神经元网络，以对如表 7-5 所示的输入输出样本分类：

表 7-5　输入输出样本

1	*0*	*1*
1	*1*	*-1*
0	*1*	*1*

说明只有该网络使用偏差，才能解决该问题。

10. 考虑以表 7-6 中样本集 X 为基础的分类问题：

表 7-6　样本 X

X:	I_1	I_2	O
	-1	*1*	*1*
	-1	*-1*	*1*
	0	*0*	*0*
	1	*0*	*0*

(a) 根据类别画出数据点的标注图。该问题能否用一个人工神经元来解决？如果可以，画出判别边界。

(b) 设计一个单神经元感知机来解决该问题。确定最终权重因子，使权重向量与判别边界正交。

(c) 用所有的 4 个样本检验解决方案。

(d) 用求出的神经网络对下列样本分类：(－2, 0)，(1, 1)，(0, 1)和(－1, －2)

(e) (d)中的哪些样本总是以相同的方式分类，哪些样本的分类因解决方案而异？

11. 实现执行单层感知机的计算(和学习)的程序。

12. 已知图 7-19 中给定的竞争神经网络：

(a) 如果输入样本是[X1, X2, X3]=[1, -1, -1]，找出输出向量[Y1, Y2, Y3]。

(b) 神经网络中新的权重因子是什么？

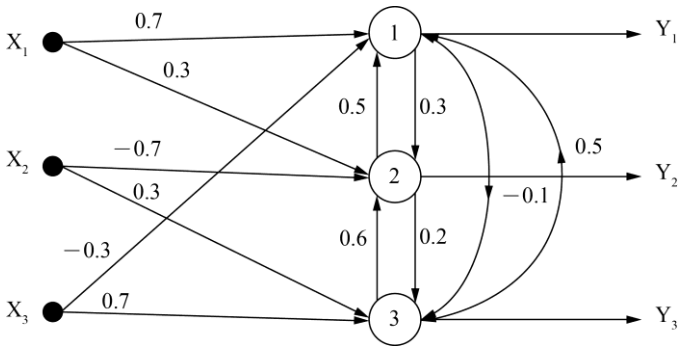


图 7-19　竞争神经网络实例

13. 搜索 Web，找出基于人工神经网络的公用软件或商业软件的基本特征。保存搜寻结果。它们中的哪些软件在学习时需要老师的指导，哪些无需老师的指导?

14. 对于神经网络，哪个结构假设对低度拟合(如高偏差模型)和过度拟合(如高方差模型)之间的平衡影响最大？

(a) 隐含节点数

(b) 学习率

(c) 权重的初始选择

(d) 使用单位为常量的输入

15. 感知机的 Vapnik-Chervonenkis (VC)维小于简单线性支持向量机(SVM)的 VC 维吗？讨论该结果。

7.9　参考书目

1. Engel, A., C. Van den Broeck, *Statistical Mechanics of learning*, Cambridge University Press, Cambridge, UK, 2001.

该书的主题是近十年来研究人员运用统计力学技术获得的机器学习结果。作者对目前仅散

见于论文中的各种重要概念和技术提供了一致的解释。该书包括许多例子和练习，可用于课堂或自学，或作为方便的参考书。

2. Haykin, S., *Neural Networks and Learning Machines*, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2009.

该书首次从工程的角度综合阐述了神经网络，组织严密，语言流畅，具备权威性，内容丰富，覆盖面广，它介绍了神经网络的许多方面，有助于读者了解该技术的起源、功能和潜在的应用。该书涵盖了这个新兴技术的所有重要方面，包括学习过程、后向传播、径向基函数、周期网络、自组织系统、模块网络、时间处理、神经动力学和 VLSI 实现。该书集成了计算机经验，演示了神经网络的实际设计和运转。各章的目标、问题、工作实例、参考书目、照片、插图和一个全面的小词典都有助于概念的理解。新的章节还介绍了其他领域的内容，如 SVM、强化学习/神经动力学编程，还有一整章的案例研究，演示了神经网络的实际应用。该书中的参考书目非常详细，便于参考。该书面向专业工程师和研究科学家。

3. Heaton, J., *Introduction to Neural Networks with Java*, Heaton Research, St. Louis, MO, 2005.

该书向 Java 程序员介绍了神经网络和人工智能，讨论了神经网络结构，如前向结构、后向传播结构、Hopfield 和 Kohonen 网络。还介绍了其他人工智能主题，例如遗传算法和模拟退火算法。每个神经网络都给出了实例，包括旅行商问题、手写识别、模糊逻辑和学习数学函数。所有 Java 源代码都可以在线下载。该书除了向程序员说明如何构建这些神经网络之外，还讨论了 Java Object Oriented Neural Engine (JOONE)，JOONE 是一个开源的 Java 神经引擎。

4. Principe, J. C., R. Mikkulainen, *Advances in Self-Organizing Maps*, Series: *Lecture Notes in Computer Science*, Vol. 5629, Springer, New York, 2009.

2009 年 6 月在佛罗里达州的圣奥古斯汀举办了第 7 届自组织映射的国际高级专题讨论会(WSOM 2009)，该书是该会议的论文集，其中包含 41 篇修订过的完整论文，这些论文是从无数提交给大会的论文中仔细审阅、选择出来的，其主题涉及 SOM 在社会科学、经济、计算生物、工程、时序分析、数据可视化和理论计算机科学等许多领域的应用。

5. Zurada, J. M., *Introduction to Artificial Neural Systems*, West Publishing Co., St. Paul, MN, 1992.

该书是人工神经网络的传统教科书之一。正文来自于为电子工程和计算机科学专业提供的人工神经系统教学成果。作者强调，神经元计算对大规模或超大规模问题的实际意义是非常明显的。

第13章

遗传算法

本章目标

- 认识为大型数据集所描述的优化问题求出近似解的有效算法。
- 比较用遗传算法表达的自然演变和仿真演变的基本原则和概念。
- 用图例描述遗传算法的主要步骤。
- 解释标准和非标准遗传算子，如改进解的机制。
- 讨论带有通配符值的图式概念及其在近似优化上的应用。
- 将遗传算法应用于旅行推销员问题及分类规则的优化。

有一大类有趣的问题，它们的合理快速算法没有开发出来。这些问题的大部分是应用中常见的优化问题。优化的基本方法是设计一个标准的度量——成本函数——来总结决策的效能或价值，并反复地选择可选方案，来提高效能。大多数经典的优化方法都是基于成本函数的梯度或高阶统计产生一连串确定的试探解。总的来说，任何有待完成的抽象任务都可以考虑成问题的求解，把求解看成是在潜在的解空间上搜索。既然是在寻求“最佳”解，就可以把这个任务看成优化过程。对小型数据空间而言，通常经典的穷举搜索法就足够了；对大型数据空间而言，就必须采用特殊技术。在一般条件下，求解技术可表示为生成解序列，这些解序列渐进收敛为最优解，在某些情况下，它们收敛速度快到呈指数级。但是，在待优化函数上加入随机扰动时，这种方法就不能正常运行。而且，在现实条件中，局部最优解常常是不适合的。尽管有这些所谓的“难优化问题”，但常可以找到一种有效的算法来计算出近似最优解。在这些方法中，有一种是以遗传算法为基础，遗传算法是根据自然演变法则开发出来的。

自然演变是一种基于群体的优化过程。在计算机上对这个过程进行仿真，产生了随机优化技术，在应用于解决现实世界中的难题时，这种技术常胜过经典的优化方法。生物学中已经解决了的问题具有混沌、机会、暂时性和非线性交互的特点，而经典优化方法难以处理的问题就具备这些特征。因此，在仿真演变研究中的主要出路就是遗传算法(GA)，它是一种新的、迭代式的优化方法，强调自然演变的某些方面。遗传算法依靠自然随机算法逼近问题的最优解，这些随机算法的研究方法为一些自然现象建立模型，如遗传的继承和达尔文的生存竞争。

13.1 遗传算法的基本原理

遗传算法(GA)是不需要求导的(derivative-free)随机优化方法，它以自然选择和演变过程为基础，但是联系又是不牢靠的。它们最早是由密歇根大学的 John Holland 在 1975 年提出并进行研究。许多生物学家在用计算机对自然遗传系统进行仿真时，都揭示了遗传算法的基本思想。在这些遗传系统中，一个或多个染色体组合成了构造和运转有机体的总遗传法则。染色体由基因构成，基因可以取大量的值，叫做等位基因(allele)值。基因的位置(位点)根据基因的功能来独立地识别。例如，可指明动物眼睛颜色基因的位点是 10，等位基因值是蓝色。

后面几节将详细介绍遗传算法的应用，本节讨论遗传算法的基本原理和组成部分。遗传算法把参数空间或解空间的每个点都编码成一个二进位串，叫做染色体。这些 n 维空间点并不像本书开头定义的那样表示样本。在其他数据挖掘方法中，样本是提前给出的用于训练和检验的数据集，遗传算法中的 n 维点集是遗传算法的一部分，并在优化过程中反复地生成。每个点或二进位串都表示所求问题的一个潜在解。在遗传算法中，优化问题的决策变量编码为一个或多个串的结构，这与自然遗传系统中的染色体类似。编码后的串由一些类似于基因的特征构成。特征位于串中不同的位置，串中每个特征都有自己的位置(位点)和一个确定的等位基因值，这个值的计算遵循所提议的编码方法。染色体中的串结构执行类似于自然演变过程的各种操作，以获得更好的替代解。根据“适合度”值来评估新染色体的质量，而这个适合度值可以看成优化问题的目标函数。自然演变和遗传算法的基本关系在表 13-1 中给出。遗传算法用一个点集作为群体，而不是一个单点。点集反复地演化，以便获得更好的总适合度值。遗传算法在每一代中都使用遗传算子如(交叉和突变)来构造出一个新的群体。成员的适合度越高，其存活和参与交叉或突变运算的可能性就越大。

表 13-1 遗传算法中的基本概念

自然演变中的概念	遗传算法中的概念
染色体	串
基因	串中的特征
位点	串中的位置
等位值	位置值(通常为 0 或 1)
基因型	串结构
表型	特性(特征)集合

遗传算法是一种多用途的优化工具，它已走出了学术界，在很多地方获得了重要应用。对于分析方法不适用的难优化问题，遗传算法往往非常有效。它已经成功地应用在线路敷设、调度、适应控制、博弈、交通问题、旅行推销员问题、数据库查询优化、机器学习等领域。在过去几十年中，由于许多大规模组合优化问题和高约束工程问题只能求得近似解，优化问题的重要性显得越来越突出。遗传算法的目标就是解决这些复杂问题。它们属于概率算法一类，但和随机算法大不相同，因为它们组合了定向搜索和随机搜索两类元素。基于遗传的搜索方法的另

一个属性是它们可以得到一组潜在解，而所有其他的方法都只处理搜索空间的一个点。由于这些特性，遗传算法比已有的定向搜索方法更可靠。

遗传算法很流行，是因为它们不依赖于函数求导，它们具有以下特性：

(1) 遗传算法是并行搜索方法，它能在并行处理机上执行，极大地提高了运行速度。

(2) 遗传算法可应用于连续型优化问题和离散性优化问题。

(3) 遗传算法是随机的，陷入局部小点的可能性较小，而在实际优化应用中，其他方法不可避免要陷入局部小点。

(4) 遗传算法的灵活性便于识别复杂模型中的结构和参数。

遗传算法原理解释了为什么对已知问题的表述可收敛于最优点。但实际应用未必遵循这条原理，主要原因如下：

(1) 对问题的编码常常使遗传算法未在原问题的空间中运行。

(2) 理想的迭代(遗传算法中的代)次数不限，而实际上却是有限制的。

(3) 理想的群体大小不限，而实际上是有限的。

这表示，在某些条件下，遗传算法无法找出最优点，甚至找不到近似最优解。这样的失效通常因为遗传算法过早地收敛于局部最优点。请注意，该问题不仅在其他优化算法中很常见，在其他数据挖掘技术中也很普遍。

13.2　用遗传算法进行优化

首先要注意，在不失一般性的情况下，可以假设所有优化问题都仅能分析为一个求最大值问题。如果优化问题是求函数 f(x)的最小值，就等价于求函数 g(x)=−f(x)的最大值。而且，还可假设目标函数 f(x)在定义域内取正值。否则，就用某个正常量 C 将函数转化成正值，如：

$$\text{man } f^*(x)=\max\{f(x)+C\}$$

如果每个实数变量 x_i 都编码成长度为 m 的二进位串，则初始值和编码信息的关系为：

$$x_i=a+\text{decimal}(\text{binary}-\text{string}_i)\{[b-a]/[2^m-1]\}$$

式中变量 x_i 的取值范围是 $D_i=[a,b]$，m 是使二进位码具有所需精度的最小整数。例如，取值范围为[10,20]的变量 x 是一个二进位编码的串，其长度等于 3，代码为 100，代码的范围在 000～111 之间。那么已编码的变量 x 的实数值是多少？此例中 m=3，相应的精度为：

$$[b-a]/[2^m-1]=(20-10)/(2^3-1)=10/7=1.42$$

这是两个连续 x_i 的值之差，差值可以作为候选极值进行检验。最后，代码为 100 的属性的十进制值为：

$$x = 10+ \text{decimal}(100)\times 1.42=10 + 4\times 1.42 = 15.68$$

把待优化问题中所有特征的二进位码串接起来，就表示一个染色体，作为一个潜在解。染

色体的总长 m 是所有特征的代码长度 m_i 的总和：

$$m = \sum_{i=1}^{k} m_i$$

式中 k 是问题中特征或输入变量的个数。介绍了构建代码的基本原则后，就可以解释遗传算法的主要步骤了。

13.2.1 编码方案和初始化

遗传算法首先为所给问题设计其解的表述。在这里，解是指可以作为可评估的正确解的任何候选值。例如，要使函数 $y=5-(x-1)^2$ 最大，x=2 是一个解，x=2.5 也是一个解，x=3 则是此问题的正确解，它使 y 最大。遗传算法的每个解的表述由设计者负责，它依赖于每个解的形式，以及哪个解的形式便于应用遗传算法。最常见的表述是一个字符串，也就是特征表述的一个代码串，串中的字符来自于固定的字母表。字母表越大，串中每个字符可表示的信息就越多。因此，要编码指定的信息量，串中的元素必须较少。但在大多数现实世界的应用中，遗传算法通常使用二进制编码方案。

编码过程把特征空间中的点转化成位串形式。例如，在三维特征空间中的点(11, 6, 9)，其每一维的取值范围是[0, 15]，这个点可以用一个串接起来的二进位串表示：

$$(11, 6, 9) \Rightarrow (101101101001)$$

其中，每个特征的十进制值通过二进制编码，成为一个四位的基因。

也可以使用其他编码方案，如格雷码，必要时还可以编码负数、浮点数或离散值。编码方案提供了一种把问题专用的知识直接转化成遗传算法框架的方法。这个过程在确定遗传算法的性能时起着重要的作用。而且，遗传算子可以而且应该和编码方案一起设计，用于具体问题。

所有特征的值编码成一个位串后，就代表一个染色体。在遗传算法中，处理的不是一个染色体，而是一个染色体集合，叫做群体。要对群体初始化，可以简单地随机设定染色体群体的大小。群体大小也是遗传算法用户要面对的最重要的选择之一，在很多应用中可能是至关重要的：能得到近似解吗？如果能，速度有多快？如果群体数量太小，遗传算法就可能收敛太快，而只得到一个局部最优解；如果群体数量太大，遗传算法可能浪费计算资源，等待改进的时间也可能太长。

13.2.2 适合度估计

在建立起群体后，下一步是计算群体中每个成员的适合度(fitness)值，因为每个染色体都是最优解的候选。对求最大值的问题来讲，第 i 个成员的适合度 f_i 通常是目标函数在这个成员处(或参数空间中的点)的估计值。解的适合度可以比较不同的解，以确定哪个解更合理。适合度值可以用复杂的分析公式、仿真模型来确定，或者参照实验观察值或日常问题的设定来确定。如果牢记目标函数的选择是高度主观的、主要取决于问题，恰当地确定适合度值可以使遗传

算法正确地工作。

适合度通常为正值，因此，如果目标函数不是严格为正，就必须进行某种数据缩放和/或平移操作。另一个方法是使用群体中成员的等级作为适合度值。这种方法的优点是目标函数不需要很精确，只要它能提供正确的等级信息即可。

13.2.3　选择

在这个阶段，必须从当前的代中建立一个新的群体。选择(Selection)操作确定哪个父染色体会参与繁殖下一代。通常，成员参与选择的概率与成员的适合度值成正比例。实现这种方法最常见的方式是设定选择概率 p 等于：

$$P_i = f_i \Big/ \sum_{k=1}^{n} f_k$$

式中 n 是群体大小，f_i 是第 i 个染色体的适合度值。这种选择方法的作用是让适合度高于平均值的成员进行繁殖，并取代适合度低于平均值的成员。

对选择过程来说(按照适合度的概率分布选择一个新群体)，可使用根据每个染色体的适合度来决定其槽(slot)大小的轮盘赌。轮盘的建立如下：

(1) 计算每个染色体 v_i 的适合度值 $f(v_i)$。

(2) 求出群体的适合度之和。

$$F = \sum_{i=1}^{pop\text{-}size} f(v_i)$$

(3) 计算每个染色体 v_i 的选择概率 P_i。

$$P_i = f(v_i)/F$$

(4) 计算每个被选中的染色体 v_i 的累积概率 q_i。

$$q_i = \sum_{j=1}^{i} P_j$$

式中 q 取值从 0 到最大值 1。取 1 表示群体中的所有染色体都包含在累积概率中。

选择过程的基础是旋转轮盘的次数和群体数目相同。每次都为新群体选择一个染色体。群体数目多大，就重复执行步骤 1 和步骤 2 多少次：

(1) 生成区间[0, 1]内的随机数 r。

(2) 如果 $r < q_1$，选择第一个染色体 v_1；否则选择第 i 个染色体 v_i，使 $q_{i-1} < r \leqslant q_i$。

显然，一些染色体可能被选择多次。这与理论是一致的。遗传算法会维护一组潜在解，进行多维搜索，并促使生成优质解。群体在进行仿真演变——在每一代中“较好”的解会繁殖下去，而“较差”的解死亡。目标函数或评价函数可用来区分不同的解，这些函数担任着环境的角色。

13.2.4 交叉

遗传算法的长处是结构化信息可与高度适合的个体进行交叉(Crossover)组合，因此，交叉算子必须能开发染色体之间重要的相似性。交叉概率 PC 参数定义了期望执行交叉操作的染色体数目——PC • pop-size。可用下面的迭代过程来定义当前群体中的染色体交叉过程。所有染色体必须重复执行步骤 1 和 2：

(1) 生成区间[0, 1]内的随机数 r。

(2) 如果 r < PC，选择指定的染色体进行交叉。

如果 PC 设为 1，群体中所有的染色体都会进行交叉操作；如果 PC=0.5，只有一半染色体会进行交叉，另一半将不变，直接包括到新群体中。

为了利用当前基因库的潜能，可以用交叉算子来生成新的染色体，这些新染色体将会保留前一代的好特征。交叉通常应用于父母对的选择。

单点交叉是最基本的交叉算子，它根据遗传代码随机选择一个交叉点，两个父母染色体在这点相互交换。在两点交叉中，选择两个点，然后把这两点之间的染色体串对换，生成新的一代。单点和两点交叉如图 13-1 所示。

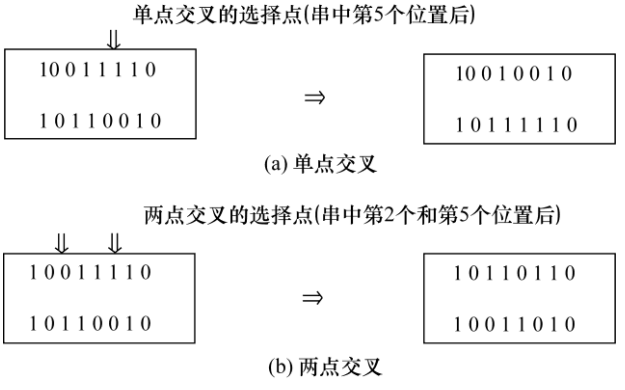


图 13-1 交叉算子

同样，可以定义 n 点交叉，在 1 点、2 点、3 点、4 点直到最后 n-1 点和 n 点之间对换染色体串。交叉的作用与自然演变过程中的交配类似，在自然交配中父母把他们的染色体段传给子女。因此，如果一些孩子获得父母的“优良”基因或基因特点，就能胜过父母。

13.2.5 突变

交叉利用了已有基因的潜力，但是如果群体不包含解决特定问题所需的所有编码信息，再多的基因混合都不能得出令人满意的解。因此，能自发产生新染色体的突变(Mutation)算子就加

入进来了。实现突变的最常见方式是以很低的指定突变率(MR)为概率，来反转染色体中的一位基因。突变算子可以防止任何一位基因在遍历整个群体后收敛于一个值。更重要的是，它可以防止群体收敛并停滞于局部最优点。突变率通常很低，所以通过交叉获得的好染色体不会丢失。如果突变率高(例如大于 0.1)，遗传算法的效果接近于原始随机搜索。图 13-2 列举了一个突变的例子。

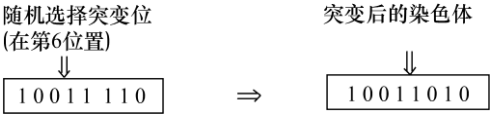


图 13-2　突变算子

在自然演变过程中，选择、交叉和突变会同时出现，以繁衍出后代。这里将它们分割成连续的阶段，以方便遗传算法的实现和实验。注意，本节只大体描述了遗传算法的基础。遗传算法的详细实现区别非常大，但是主要的阶段和迭代过程是一样的。

在本节末尾，可将遗传算法的主要组成部分概括成编码方案、适合度估计、父母选择以及交叉算子和突变算子的应用。这些阶段交迭执行，如图 13-3 所示。

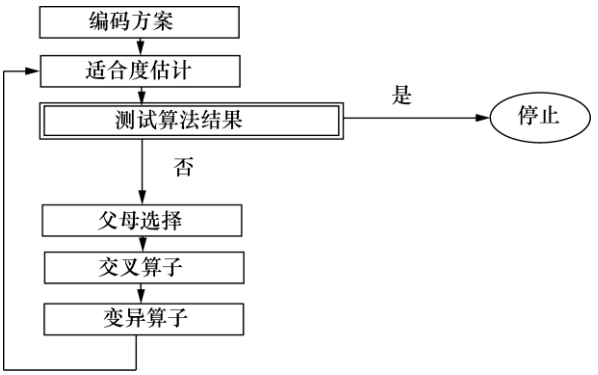


图 13-3　遗传算法的主要阶段

在演变过程中，跟踪最好的个体染色体相对容易。在遗传算法的实现中，按惯例是在独立的位置上存储“曾经是最好”的个体。这样，算法就可以上报在整个过程中找到的最佳值，它刚好在最后的群体中。

约束条件下的优化也属于用遗传算法求正解的一类问题。遗传算法的约束处理技术可以分成几类。一种方法是通过违反约束，来处理遗传算法的候选解，即先在不考虑约束的情况下生成潜在的解，然后通过减少评价函数的适合度，对解进行“惩罚”(penalize)。换句话说，就是把所有违反约束的情况和惩罚联系起来，把有约束的问题转换成无约束的问题。这些惩罚放入评价函数中，实现方式有许多种。一些惩罚函数指定一个约束作为惩罚措施。其他惩罚函数则依赖于约束的违反程度：违反程度越严重，惩罚力度越大。根据违反的程度，惩罚函数的增长可以是对数的、线性的、二次方的、指数的等。在本章的参考书目中介绍了遗传算法在约束条件下优化的几种实现方式(13.9 节)，以便于深入学习。

13.3 遗传算法的简单例证

要对某个问题应用遗传算法，必须定义或选择以下 5 个部分：

(1) 为问题的潜在解选择遗传表述或编码方案。

(2) 一种创建潜在解的初始群体的方法。

(3) 一个评价函数，它扮演着环境的角色，根据“适合度”对解进行评级。

(4) 改变后代成分的遗传算子。

(5) 遗传算法使用的不同参数值(群体大小、应用算子的比率等)。

下面列举一个简单的例子来讨论遗传算法的主要特征。假设要优化一个简单的单变量函数。此函数定义为：

$$f(x) = x^2$$

任务是在取值范围[0, 31]内找出使函数 f(x)最大的 x。选择这个问题是因为很容易对函数 f(x)进行优化分析，用遗传算法比较分析优化的结果，并找到近似的最优解。

13.3.1 表述

遗传算法的第一步是以编码串的形式表示可供选择的解(输入特征的值)。这个串一般是一系列具有值的特征；每个特征的值都可以用离散值集(叫做等位集)中的一个值来编码。根据问题的需要来定义等位集，而找出合适的编码方法是遗传算法的使用艺术的一部分。编码方法必须最简单，但能完整地表述问题的解。本例使用一个二进制向量作为一个染色体，来表示单个变量 x 的实值。向量的长度取决于所要求的精度，本例中，所选精度为 1。因此，为了在取值范围[0,31]内达到所要的精度，至少需要五位代码(串)：

$$(b-a)/(2^m-1) \leqslant \text{要求精度}$$

$$(31-0)/(2^m-1) \leqslant 1$$

$$2^m \geqslant 32$$

$$m \geqslant 5$$

本例中，用下面的关系式来定义从实数到二进制编码的映射(因为 a=0)：

$$\text{编码} = \text{二进制}(x_{\text{十进制}})$$

而从二进制编码到实数值的逆映射也是唯一的：

$$X = \text{十进制}(\text{编码}_{\text{二进制}})$$

它只用于检查优化的中间结果。例如，要把 x=11 转化成二进制串，相应的代码就是 01011。相反，代码 11001 表示十进制值 x=25。

13.3.2　初始群体

初始化过程非常简单：随机创建一个指定长度的染色体(二进制码)群体。假设群体中串的数目等于 4，那么随机生成的一个染色体群体是：

CR_1=01101
CR_2=11000
CR_3=01000
CR_4=10011

13.3.3　评价

表示染色体的二进制向量的评价(Evaluation)函数和初始函数 f(x)是等价的，在 f(x)中，给定的染色体表示实数值 x 的二进制码。如前所述，评价函数扮演着环境的角色，它根据“适合度”对潜在的解进行评级。在本例中，4 个染色体 CR_1 到 CR_4 对应输入变量 x 的值：

$x_1(CR_1)=13$
$x_2(CR_2)=24$
$x_3(CR_3)=8$
$x_4(CR_4)=19$

于是，评价函数对它们进行评级如下：

$f(x_1)=169$
$f(x_2)=576$
$f(x_3)=64$
$f(x_4)=361$

对初始生成的染色体的评价结果可以采用表格的形式给出，如表 13-2 所示。期望繁殖这一列展示了初始群体中染色体的“评价质量”。染色体 CR_2 和 CR_4 参与下一代繁殖的可能性比 CR_1 和 CR_3 更大。

表 13-2　初始群体的评价

CR_i	代码	X	f(x)	f(x)/Σ f(x)	期望繁殖：f(x)/fav
1	01101	13	169	0.14	0.58
2	11000	24	576	0.49	1.97
3	01000	8	64	0.06	0.22
4	10011	19	361	0.31	1.23
Σ			1170	1.00	4.00
平均值			293	0.25	1.00
最大值			576	0.49	1.97

13.3.4 交替

在交替(Alternation)阶段，根据前一次迭代中评价的群体来选择新的群体。显然，本例中染色体 CR_4 是 4 个染色体中最好的，因为它的评价返回值最高。在交替阶段，依赖其目标函数值或适合度值来选择个体。对最大值问题来讲，个体的适合度越高，它参与下一代繁殖的可能性也就越大。在选择阶段可以使用不同的方案，在前面提出的简单遗传算法——轮盘选择技术中，根据为每个个体计算出的选择概率来随机选择个体。选择概率是用个体的适合度值除以群体的适合度之和，这些值都在表 13-2 的第 5 列中给出。

下一步将设计轮盘，对于本例，轮盘如图 13-4 所示。

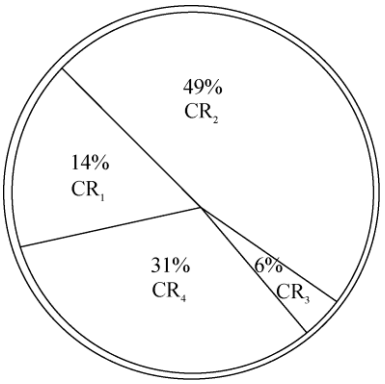


图 13-4　选择下一代群体的轮盘

可以用轮盘选择下一代群体的染色体。假设为下一代群体随机选择的染色体为 CR_1、CR_2、CR_2、CR_4 (染色体的选择要与表 13-2 中第 6 列的期望繁殖相一致)。在下一步中，这 4 个染色体要进行遗传操作：交叉与突变。

13.3.5 遗传算子

交叉未必应用于所有已选的个体对。可根据一个特定的概率 PC(交叉概率)来选择要交叉的个体对，典型的交叉概率在 0.5～1 之间，如果没有进行交叉(PC=0)，后代就只是父母对的复制品。对交叉过程来讲，确定要进行交叉的个体占群体的百分比是必不可少的。对于本例，用下面的遗传算法参数：

(1) 群体大小 pop-size=4(这个参数已经使用了)。

(2) 交叉概率 PC=1。

(3) 突变概率 PM=0.001(在突变操作中将用到这个参数)。

交叉概率为 1 表示进行 100%的交叉——所有的染色体都会进行交叉操作。

在遗传算法的这个阶段，第 2 个参数集是随机选择要进行交叉操作的父母，和在串中进行交叉的位置。假设 CR_1-CR_2 和 CR_2-CR_4 是随机选择的染色体对，两对的交叉位置都在第 3 个位置后。选择的串

第 1 对　　CR_1= 01101

CR_2= 11000

第 2 对　　CR_2= 11000

CR_4= 10011

在交叉后，会变成一个新群体：

CR_1'=01100

CR_2'=11001

CR_3'=11011

CR_4'=10000

第 2 个可应用在遗传算法的每次迭代中的算子是突变。在本例中，突变算子的概率为 0.1%，意思是对 1000 个转换后的位，突变只执行一次。由于只转换了 20 位(把一个 4×5 位的群体转换成另一个群体)，出现突变的概率非常小。因此，可假设串 CR_1'到 CR_4'在第一次迭代的突变操作中将保持不变。每 50 次迭代期望发生一位突变。

在遗传算法的第一次迭代中，最终处理步骤就是如此。其结果是新群体 CR_1'到 CR_4'，下一次迭代将会从评价处理开始，其中会用到新群体。

13.3.6　评价(第二次迭代)

在新群体中重复评价过程。其结果在表 13-3 中给出。

表 13-3　第二代染色体的评价

CR_i	代码	x	f(x)	f(x)/∑f(x)	期望繁殖：f(x)/fav
1	01100	12	144	0.08	0.32
2	11001	25	625	0.36	1.44
3	11011	27	729	0.42	1.68
4	10000	16	256	0.14	0.56
∑			1754	1.00	4.00
平均值			439	0.25	1.00
最大值			729	0.42	1.68

在遗传算法的其他迭代中，优化过程可依照表 13-3 继续进行。本例的计算步骤就到此为止，并对结果进行一些分析，以深入理解遗传算法。

虽然遗传算法中使用的搜索技术是基于一些随机参数的，但它们能利用每个群体中最好的选择方案，从而获得更好的解。比较表 13-2 和 13-3 中的总和、平均值和最大值：

$$\sum\nolimits_1 = 1170 \Rightarrow \sum\nolimits_2 = 1754$$

$$\text{平均值}_1 = 293 \Rightarrow \text{平均值}_2 = 439$$
$$\text{最大值}_1 = 576 \Rightarrow \text{最大值}_2 = 729$$

这些比较说明新的第二代群体更接近函数 f(x)的最大值。前面两次迭代中评价出的最佳染色体是 CR_3'=11011，它对应于特征的值 x=27(理论上 f(x)的最大值是已知的，就是当 x=31 时，f(x)=961)。并不是遗传算法的每次迭代都可以获得这样的增长，但平均来看，经过多次迭代后，最终群体将更接近于解。迭代次数是遗传算法的一个可行的终止条件。其他可能的终止条件是两次连续迭代的结果之差小于指定的阈值，或者获得了恰当的适合度，或者计算时间有限，都可以终止遗传算法的计算。

13.4 图式

遗传算法的理论基础是解的二进制表述，以及图式表示法。图式是指允许在染色体之间进行相似度探测的一个模板。为简要介绍图式的概念，必须先定义一些相关的术语。搜索空间 Ω 是可行染色体或串的完整集合。在一个长度固定为 l 的串中，每一位(基因)都在大小为 k 的字母表 A 中取一个值，得到的搜索空间大小为 k^l。例如，在二进制编码串中，串长为 8，搜索空间大小为 2^8=256。在群体 S 中，串用向量 x∈Ω 来表示。因此，前面的二进制编码串例子中，x 是$\{0,1\}^8$中的一个元素。图式就是在某些位置定义有固定值的串的子集的一个相似模板。

可以通过向基因字母表中引入一个通配符(*)来建立图式。图式中的每个位置都取字母表中的一个值(固定位置)或取一个“通配”符。例如，对于二进制，长度为 1 的图式定义为 $H \in \{0,1,*\}^1$，图式表示除了在‘*’位置之外，在其他所有位置都匹配的所有串。换句话说，图式表示搜索空间的一个子集，或者这个搜索空间中的一个超平面划分。例如，考虑长度为 10 的串和图式。图式

(*111100100)

匹配两个串

{(0111100100),(1111100100)}

图式

(*1*1100100)

匹配 4 个串

{(0101100100),(0111100100),(1101100100),(1111100100)}

当然，图式

(1001110001)

只代表一个串，图式

(**********)

代表所有长度为 10 的串。总之，可行图式的总数为$(k+1)^l$，式中 k 是字母表中的符号数，l 是串的长度。在长度为 10 的二进制编码串例子中，有$(2+1)^{10}=3^{10}=59\,049$ 个不同的串。显然，每个二进制图式刚好匹配 2^r 个串，其中，r 是图式模板中通配符的个数。反过来，每个长度为 m 的串都有 2^m 个不同的图式与其匹配。

在优化定义域为[0,31]的函数 $f(x)=x^2$ 时，可以用图来表示五位代码的不同图式。每个图式代表此二维问题空间的一个子空间。例如，图式 1****减少了图 13-5(a)中子空间上的解的搜索空间。图式 1*0**相应的搜索空间如图 13-5(b)所示。

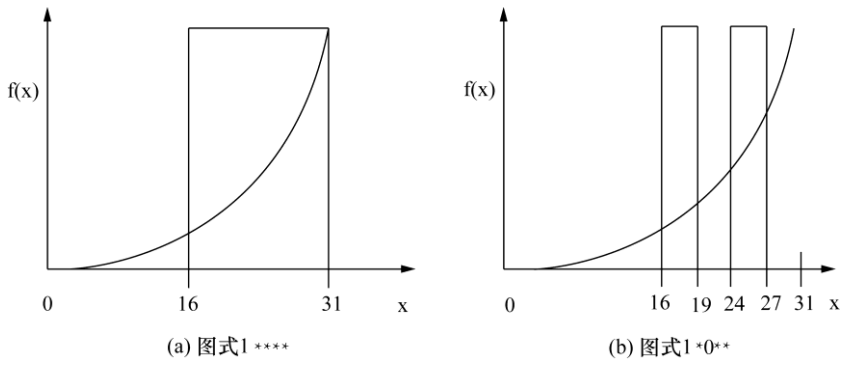


图 13-5　$f(x)=x^2$：不同图式的搜索空间

不同图式具有不同的特性，其中有 3 个重要的属性：顺序(O)、长度(L)和适合度(F)。图式 S 的阶数用 O(S)表示，是图式中 0 和 1 位置的个数，也就是图式中表示的固定位置的个数。这个参数的计算非常简单：模板长度减去通配符的个数。例如，下面是长度均为 10 的 3 个图式：

$$S_1=(***001*110)$$
$$S_2=(*****0**0*)$$
$$S_3=(11101**001)$$

它们的阶数分别为：

$$O(S_1)=10-4=6，O(S_2)=10-8=2，O(S_3)=10-2=8$$

图式 S_2 是最特殊的一个，而 S_2 是最一般化的一个。图式的阶数符号可用于计算图式在突变中的存活概率。

图式 S 的长度表示为 L(S)，它是第一个和最后一个固定串位置之间的距离。它定义了图式中信息的紧密度。例如，上述图式 S_1 到 S_3 的长度参数值为：

$$L(S_1)=10-4=6，L(S_2)=9-6=3，L(S_3)=10-1=9$$

注意，只有一个确定位置的图式其长度为 0。图式的长度参数 L 可用于计算图式在交叉中的存活概率。

图式 S 的另一个属性是它在时刻 t 的适合度 F(S,t)(用于指定的群体)。它定义为群体中匹配此图式 S 的所有串的平均适合度。假设在 t 时刻群体中与图式 S 匹配的串有 p 个串$\{v_1,v_2,\cdots,v_p\}$，

那么：

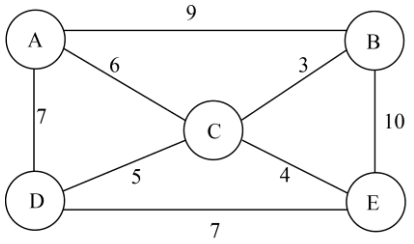
$$F(S,t)=\left[\sum_{i=1}^{p} f(v_i)\right]\Big/P$$

本书给出的图式构建的基本原则解释了一个问题(但没有证据)：即短的(高阶数)、低顺序(低长度)、大于平均数(高适合度)的图式在遗传算法的下一代中匹配的串会呈指数增长。这个原则的一个直接后果就是遗传算法用短的、低顺序的图式来探索搜索空间，这些图式在后来的交叉和突变操作中用于交换信息。因此，遗传算法通过分析这些图式，来寻求近似最优点的过程，叫做搭积木。但要注意，搭积木方法只是一个经验结果，没有任何证据，对一些现实世界的问题来说，很容易违反这些规则。

13.5 旅行推销员问题

本节解释怎样将遗传算法用于求解旅行推销员问题(TSP)。简单地说，旅行推销员必须亲自访问一次自己所负责区域的每个城市，然后回到起点。若在所有城市之间的旅行费用已知，该怎样计划旅行路线，才能尽量降低旅行的总费用？TSP 是一个组合优化问题，出现在大量的应用中。求解这个问题的算法有几种，如分支界限算法、逼近算法和试探式搜索算法。近几年，人们多次尝试用遗传算法逼近 TSP 的解。

TSP 的描述基于数据的图形表述。该问题可以定义为：给定一个不定向的加权图，找出最短的路径，即每个顶点都要访问一次的最短路径，但起点和终点相同。图 13-6 是这种图及其最优解的一个例子。A、B、C 等是要访问的城市，线边的数字是城市间的访问费用。



最优解：A-----►B----►C----►E----►D---►(A)

图 13-6　旅行推销员问题的图形表述和相应的最优解

即使解并不是一个最优解，用城市的排列来表示此问题的每个解是很自然的。最后的城市可以去掉，因为它总与出发城市相同。但在计算每次旅行的总距离时，都要计算到最后城市的距离。

用城市的排列表示每个解时，每个城市都必须访问一次。但并非每个排列都代表一个有效解，因为一些城市没有直接连接(如图 13-6 中的 A 和 E)。一个实用的方法是为没有直接相连的城市人为指定一个很大的距离。这样，就会去除含有连续的、不毗连的城市的无效解，所有的解都可接受。

本问题的目标是尽量减小每次旅行的费用，这个目标可以选择不同的适合度函数来表示。例如，如果总距离为 s，要使适合度函数最小，f(s)就是简单的 f(s)=s；要使适合度函数最大，可以选择 f(s)=1/s、$f(s)=1/s^2$、f(s)=K−s，式中 K 是使 f(s)≥0 的正常量。设计最好的适合度函数没有通用公式，但是，若没有在适合度函数中恰当地反映出解，就无法找到最优解或近似最优解。

在处理城市的排列时，简单的交叉操作会产生无效解。例如，对图 13-6 中的问题而言，两个解的串在第 3 个位置之后的位进行交叉：

⇓
A D E B C
A E C D B
⇑

会得到新串

A D E D B
A E C B C

这两个串都是无效解，因为它们并不代表串中初始元素的排列。要避免这个问题，引入一种修正交叉操作，它直接操作城市的排列，仍然得到一个排列。这就是部分匹配交叉(PMC)操作。它不仅能应用于旅行推销员的问题，也能应用于用排列来表述解的其他问题。下面列举一个例子来说明 PMC 操作的效果。假设两个解用相同符号的不同排列来表示，且 PMC 是两点操作。应用 PMC 操作过程的第一步是选择两个串和两个随机交叉点。

⇓⇓
A D E B C
A E C D B
⇑⇑

交叉点之间的子串称为匹配项。本例中，匹配项有两个元素：第一个串中为 E B，第二个串中为 C D。交叉操作要求交换 E 和 C，表示为有序对(E, C)，同样，B 和 D 也要交换，用(B, D)表示。PMC 操作的下一步是在两个串中交换这两个元素的排列。换句话说，必须在两个串中交换有序对(E, C)位置和(B, D)的位置。在第一个串中，(E, C)交换后的结果为 A D C B E，在进行第二对(B, D)的交换后，最终结果为 A B C D E。对第二个串进行同样的操作，第一次交换后变为 A C E D B，最后变成 A C E B D。如果分析这两个经过 PMC 操作后的串：

⇓⇓
A B C D E
A C E B D
⇑⇑

可以看到，两个串的中间部分像标准交叉操作那样进行了交换，但是得到的两个新串仍然是有效排列，因为实际上每个串中的符号都进行了交换。

用于解决旅行推销员问题的遗传算法的其他步骤都保持不变。基于以上算子的遗传算法要优于对 TSP 的随机搜索，但是，这种算法仍然有很大的提升空间。在解决 100 个随机生成的城

市的 TSP 时，这种算法在经过 20 000 代后，得出的结果高于最小值 9.4%。

13.6 使用遗传算法的机器学习

优化问题是遗传算法最常见的一种应用。一般说来，优化问题试图确定一个解，例如确定机构的最大利润，或者确定生产过程所选特征的值，来计算产品成本的最小值。遗传算法的另一个常见的应用领域是为通常比较复杂的已知系统确定输入-输出映射，这也是所有的机器学习算法要解决的问题。

输入-输出映射的基本观点是得出一个形式合理的函数或模型，该形式往往比原始映射要简单，而原始映射通常用一个输入-输出样本集来表示。函数是这种映射的最好描述方式。“最好”这个词的测度要依赖特定的应用。常见的测度是函数的精度、可靠性和计算效率。通常，要找到一个满足所有这些标准的函数并不容易。所以，遗传算法要确定一个“优秀”的函数，此函数可以成功地应用于模式分类、控制和预测。映射过程可以是自动的，这种使用遗传算法的自动化代表了另一种为归纳型机器学习建模的方法。

本书前面的章节描述了机器学习的不同算法。根据某个已知的样本集开发新模型(输入-输出的抽象关系)也可以在遗传算法领域内实现。有好几种基于遗传算法的学习方法。本节将解释基于图式的技术的基本原理，及其应用于分类问题的可能性。

本节列举一个简单的数据库作为例子。假定训练或学习数据集使用属性集合来描述，每个属性都有自己的类别取值范围：一个可行值的集合。这些属性在表 13-4 中给出。

表 13-4 已知数据集 s 的属性 A_i 和可能的值

属　性	值
A_1	x,y,z
A_2	x,y,z
A_3	y,n
A_4	m,n,p
A_5	r,s,t,u
A_6	y,n

一个分类模型有两类样本 C1 和 C2，该模型可用 if-then 的形式来表示，左边是输入特征值的布尔型表达式。右边是相应的类：

$$([A_1=x]\wedge[A_5=s])\vee([A_1=y]\wedge[A_4=n])\Rightarrow C_1$$
$$([A_3=y]\wedge[A_4=n])\vee(A_1=x)\Rightarrow C_2$$

这些分类规则或分类器都可以用更一般的方式来表示：如指定字母表中的一些串。若数据集有 6 个输入和一个输出，则每个分类器的形式为：

$$(p_1, p_2, p_3, p_4, p_5, p_6)：d$$

式中 P_i 表示表 13-4 中域的第 i 个属性值($1 \leqslant i \leqslant 6$)，d 是两类中的一种。要用指定的形式描述分类规则，必须在每个属性的值集中加入通配符“*”。例如，属性 A_1 的新值集合为{x, y, z,*}。对其他属性也作类似的扩展。前面为类 C_1 和 C_2 指定的规则分解成几段(这些段可以使用 AND 逻辑运算连接起来)，表示为：

(x****s*)：C_1
(y**n**)：C_1
(**yn**)：C_2
(x*****)：C_2

要简化此例，可假设此系统只分两个类：C_1 和非 C_1。任何系统都很容易扩展，来处理多个类(多重分类)。对只有一个规则 C_1 的简单分类来说，d 只接受两个值：d=1(类 C_1 的成员)和 d=0(不是 C_1 的成员)。

假设在学习过程的某些阶段，分类器 Q 在系统中有一个随机生成的小群体，每个分类器的力度 s 为：

Q_1　(***ms*)：1，　S_1=12.3
Q_2　(**y**n)：0，　S_2=10.1
Q_3　(xy****)：1，　S_3=8.7
Q_4　(*z****)：0，　S_4=2.3

力度 S_i 是根据可用的训练数据集计算出来的参数，它们表示规则对训练数据集的适合度，和规则所支持的数据集的百分比成比例。

这里使用遗传算法的基本迭代步骤以及相应的算子，根据训练数据集规则的适合度函数来优化规则集。在这个学习技术中也要使用交叉和突变算子。但是必须对突变做一些修正。第一个属性 A_1 的取值范围为{x, y, z,*}。因而，在突变时，会把突变字符(码)变成其他 3 个概率相同的字符之一。其后代的力度通常和父母一样。例如，如果在随机选择的位置 2 上对规则 Q_3 进行突变，用一个随机选择的值*替代值 y，则突变后的新分类器为：

Q_{3M}　(x*****)：1，　S_{3M}=8.7

交叉操作不需要做任何修正。利用所有分类器的长度相等这个特点，把选出来的父母进行交叉。假设有 Q_1 和 Q_2：

⇓

Q_1　(***ms*)：1
Q_2　(**y**n)：0

则先生成一个随机交叉位置的点，假定在串中标示的第 3 个字符后进行交叉，则后代为：

Q_{1c}　(*****n)：0，
Q_{2c}　(**yms*)：1

后代的力度是父母的力度的(可能是加权)平均值。现在，系统准备继续进行学习过程：开始另一次循环，进一步从训练数据集中接受正的或负的样本，并修正分类器的力度，作为适合度的测度。注意，训练数据集是通过估计每次迭代的图式力度而引入学习过程中的。分类器应收敛于一些力度很高的规则。

这种算法的一种可能的实现方法是 GIL 系统，这种系统使遗传算法更接近符号水平——主要是定义操作二进制串的专门算子。前述的符号分类器都转换成二进制串，对每个属性都产生一个固定长度的二进制串。长度等于指定属性的值的个数。在串中，有要求的值设为 1，其他则设为 0。例如，如果属性 A_1 的值为 z，用二进制串 001 来表示(两个 0 表示 x 和 y)。如果属性的值为*，意味着此属性可能取任何值，在二进制串中所有位置都取 1。

前面的例子中有 6 个属性，所有属性加起来共有 17 个不同的值，分类器用符号表示为：

(x***r*)∨(y**n**)：1

转换成二进制表述：

(100|111|11|111|1000|11∨010|111|11|010|1111|11)

在表达式中，每个属性之间用竖线分开。GIL 系统的算子是根据归纳推理来建模的，它包括各种归纳算子，如规则交换(RuleExchange)、规则复制(RuleCopy)、规则概化(RuleGeneralization)、规则特化(RuleSpecialization)、规则分割(RuleSplit)、SelectorDrop、ReferenceChange、ReferenceExtension 等。下面依次讨论其中一些算子。

13.6.1 规则交换

规则交换算子与经典遗传算法的交叉算子极其相似，因为它将两个父染色体中被选择的复合体相互交换。例如，两个父体(规则)：

⇓

(100|111|11|111|1000|11∨010|111|11|010|1111|11)和

(111|001|01|111|1111|01∨110|100|10|010|0011|01)

⇑

交换后生成后代(新规则)：

(100|111|11|111|1000|11∨110|100|10|010|0011|01)和

(111|001|01|111|1111|01∨010|111|11|010|1111|11)

13.6.2 规则概化

此一元算子概括出复合体的一个随机子集。例如，对于父体：

(100|111|11|111|1000|11∨110 100 10 010 0011 01∨010|111|11|010|1111|11)

选择第二和第三个复合体进行概化，将它们按位或运算，生成后代为：

(100|111|11|111|1000|11∨110|111|11|010|1111|11)

13.6.3　规则特化

此一元算子特化了复合体的一个随机子集。例如，对于父体：

(100|111|11|111|1000|11∨110|100|10|010|0011|01∨010|111|11|010|1111|11)

选择第二和第三个复合体进行特化，将它们按位与运算，生成后代为：

(100|111|11|111|1000|11∨010|100|10|010|0011|01)

13.6.4　规则分割

此算子对单个复合体进行操作，把它分割成很多个复合体。例如，对于父体：

(100|111|11|111|1000|11)

⟹

⇑

它可以产生后代(算子对第二个选择器进行了分割)

(100|011|11|111|1000|11∨100|100|11|111|1000|11)

GIL 系统是一个基于遗传算法原理的复杂归纳学习系统。它需要大量参数，如应用每个算子的概率。它也是一个迭代过程。每次迭代时，所有染色体按照它们的完整度、一致性和适合度标准演变，形成一个新群体，较好的染色体更有可能出现在新群体中。然后把这些算子应用于新群体中，再循环下去。

13.7　遗传算法用于聚类

人们花费了许多精力，用遗传算法替代传统的聚类算法，来提供更好的解。其重点是适当的编码方案、特定的基因算子和对应的适合度函数。还专门为数据聚类提出了几个编码方案，其中 3 个主要类型是二进制、整数和实数编码。

二进制编码方案通常表示为长度为 N 的二进制串，其中 N 是数据集样本的数量。二进制串中的每个位置都对应一个样本。若第 i 个样本是聚类的原型，第 i 个基因的值就是 1，否则为 0。例如，表 13-5 中数据集 s 可以编码为串[0100001010]，其中样本 2、7、9 是类 C_1、C_2 和 C_3 的原型。在该串中，1 的个数等于预先定义的聚类的数量。显然，这个编码方案会得到基于中心点的表示，类的原型匹配数据集中有代表性的样本。用二进制编码方案表示某个数据分区还有另一种方法：使用 k × N 维的矩阵，其中行表示类，列表示样本。采用这种方法时，若第 j

个样本属于第 i 个类，就给(i,j)基因型赋予 1，而同一列上的其他元素均为 0。例如，使用这种表示法，表 13-5 中的数据集就编码为 3 ×10 矩阵，如表 13-6 所示。

表 13-5　为给定的数据集 s 定义的 3 个类

样本	特征 1	特征 2	聚类
1	1	1	C_1
2	1	2	C_1
3	2	1	C_1
4	2	2	C_1
5	10	1	C_2
6	10	2	C_2
7	11	1	C_2
8	11	2	C_2
9	5	5	C_3
10	5	6	C_3

表 13-6　表 13-5 中给出的数据集 s 的二进制编码

1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1

整数编码使用 N 个整数位置的向量，其中 N 是数据集样本的数量。

每个位置都对应一个样本，即第 i 个位置(基因)表示第 i 个数据样本。假定有 k 个聚类，则每个基因都取字母表{1, 2, 3, … , k}中的一个值。这些值定义了类标记。例如，整数向量[1111222233]表示表 13-5 中的聚类。采用整数编码方案表示分区的另一种方式是使用只有 k 个元素的数组为数据集提供基于中心点的表示。采用这种方法时，每个数组元素都表示样本的索引 x_i，$i = 1, 2, \dots, N$ (按照样本在数据集中的顺序)，该索引对应于给定类的原型。例如，数组[1 6 10]可以表示一个分区，其中 1、6、10 是表 13-5 中数据集的类原型(中心点)的索引。整数编码的计算效率通常高于二进制编码方案。

实数编码是第三种编码方案，其基因型由实数组成，表示类重心的坐标。在 n 维空间中，前 n 个位置表示第一个重心的 n 维坐标，接下来的 n 个位置表示第二个重心的 n 维坐标，以此类推。例如，基因型[1.5 1.5 10.5 1.5 5.0 5.5]编码了 3 个重心，分别是表 13-5 中类 C1、C2、C3 的(1.5, 1.5)、(10.5, 1.5)和(5.0, 5.5)。

将遗传算法应用于聚类时，第二个重要决策是选择合适的遗传算子。人们提出了许多交叉和突变算子，来解决遗传算法中与环境无关的重要问题。在聚类问题中使用传统的遗传算子时，它们通常只操作基因值，而没有考虑它们与其他基因的关系。例如，图 13-7 中的交叉操作说明，两对父母表示聚类问题的相同解(标记不同，但整数编码相同)，它们生成的后代表示的聚类解不同于其父母表示的聚类解。而且，假定类的数量提前指定为 $k = 3$，则只有两个类的解就无效。因此，必须为聚类问题专门开发遗传算子。例如，交叉算子应重复应用，或者不定期地进行突

变操作，直到得到有效的后代为止。

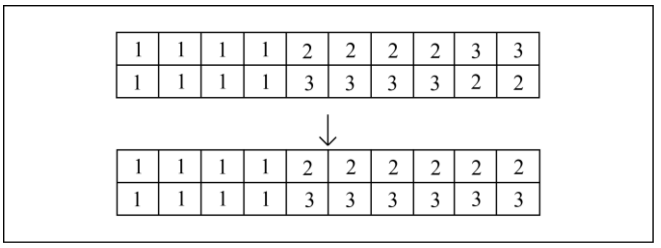


图 13-7　相同的父母通过交叉运算得到不同的后代

不同的聚类验证条件可用作适合度函数，以演变出聚类问题中的数据分区。它们主要取决于编码方案和所选的遗传算子集。这里仅列举聚类适合度函数的一个例子：使用基于中心点的实数编码方案时，适合度函数 f 要最小化样本与其类平均值的欧式距离的平方和。该适合度函数 $f(C_1,C_2,\ldots,C_k)$用公式表示为：

$$f(C_1,C_2,\ldots,C_k)=\sum_{j=1}^{k}\sum_{x_i\in C_j}|x_i-z_j|^2$$

其中$\{C_1,C_2,\ldots,C_k\}$是编码为基因型的 k 个聚类，x_i 是数据集中的一个样本，z_j 是类 C_j 的重心。注意，只有类的数量 k 预先指定，且它使类内距离最小，使类间距离最大，这个条件才成立。一般情况下，适合度函数基于样本与类重心或中心点之间的距离。尽管这类函数应用广泛，但它们通常倾向于发现球形的类，这显然不适合许多实际应用。也可以采用其他方法，包括基于密度的适合度函数。实际上，要成功应用遗传算法解决聚类问题，很大程度上取决于如何根据编码方案、算子、适合度函数、选择过程和初始群体来设计遗传算法的应用方式。

13.8　复习题

1. 已知一个二进制串，它表示 4 个属性值的串联：

{2, 5, 4, 7}={010101100111}

用此例解释遗传算法的基本概念以及它们在自然演变中的等价概念。

2. 如果用遗传算法优化函数 f(x)，x 要求精确到 6 位小数，取值范围为[-1, 2]，则二进制向量(染色体)的长度是多少？

3. 如果两个染色体 v1=(00110011)和 v2=(01010101)，交叉点随机选在第 5 个基因后，得出的两个后代应该是多少？

4. 已知图式(*1*00)，它匹配哪些串？

5. 图式(********)可匹配多少个串？

6. $f(x) = -x^2+16x$ 是定义在区间[0, 63]上的一个函数，用遗传算法的两次迭代，求出 f(x)最大值的近似解。

7. 对题 6 中的函数 f(x)，比较 3 个图式的阶数(O)、长度(L)和适合度(F)

$$S_1 = (*1*1**)$$
$$S_2 = (*10*1*)$$
$$S_3 = (**1***)$$

8. 已知父染色体(1 1 0 0 0 1 0 0 0 1)，若采用如下突变概率，此染色体可能的后代是什么？(举出例子)

(a) p_m=1.0

(b) p_m =0.5

(c) p_m =0.2

(d) p_m =0.1

(e) p_m =0.0001

9. 解释搭积木假设的原理以及其潜在的应用。

10. 在两个串 S1 和 S2 上进行部分匹配交叉(PMC)操作，随机选择交叉点如下：

$$S_1 = \{A\ C\ B\ D\ F\ G\ E\}$$
$$S_2 = \{B\ D\ C\ F\ E\ G\ A\}$$
$$\uparrow \qquad \uparrow$$

11. 在 Web 上搜索基于遗传算法的公用或商业软件工具，找出其基本特性，记录搜索结果。

13.9 参考书目

1. Fogel, D. B., ed., *Evolutionary Computation*, IEEE Press, New York, 1998.

本书集中了 30 篇里程碑式的论文，横跨了演变计算的整个历史——从今天的研究追溯到 40 多年前的起源。

2. Goldenberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.

本书是遗传算法的第一批综合性文章之一。它以非常系统化的方式介绍了大部分技术，这些技术的修正和改进比较小，也是今天近似优化解决方案的一部分。

3. Hruschka, E., R. Campello, A. Freitas, A. Carvalho, A Survey of Evolutionary Algorithms for Clustering, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 39, No. 2, 2009, pp. 133–155.

本文呈现了为分类任务设计的算法的演变过程，反映了这个领域的概貌，主要讨论了文献中比较重视的一些主题。在这方面，有大量有关分区算法的论文讨论了如何解决较难的数据聚类问题，但本文还包含一些重复方法(如软和模糊算法)。本文最后提出了一些重要的问题，以及一些可能成为未来研究主题的开放式问题。

4. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin: Germany, 1999.

本书用通俗易懂的语言解释了遗传算法这个领域，根据一些有趣的检验案例讨论了这种方法的效率。这种技术的重要性在于它可用于解决许多以大量离散数据表示的难优化问题，如旅行推销员问题、调度、分区和控制等问题。