普通高等教育“十一五”国家级规划教材
高等学校规划教材

模 式 识 别

李晶皎　赵丽红　王爱侠　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书系统阐述了模式识别的原理与方法，并在此基础上介绍了模式识别的应用。全书分为：基础部分和应用部分：基础部分主要包括统计模式识别、模糊模式识别、神经网络模式识别等内容；应用部分有车牌识别和语音识别。本书将理论与实践相结合，有利于读者加深对理论方法的理解，可使读者比较系统地掌握模式识别的理论和相关技术。书中给出的两个应用实例，为读者应用模式识别方法来解决实际问题提供了具体思路和方法。附录给出的习题解答，有利于学生学习理解原理与方法。

本书可以作为高等院校自动化、计算机、生物医学工程等学科本科生、研究生的教材或教学参考书，亦可供有关工程技术人员参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

模式识别 / 李晶皎，赵丽红，王爱侠编著．—北京：电子工业出版社，2010.11
高等学校规划教材
ISBN 978-7-121- 04401-4

Ⅰ．①模…　Ⅱ．①李…②赵…③王…　Ⅲ．①模式识别－高等学校－教材　Ⅳ．①O235

中国版本图书馆 CIP 数据核字(2010)第 188521 号

策划编辑：许菊芳
责任编辑：许菊芳　　特约编辑：王　崧
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1092　1/16　印张：20　　字数：512 千字
印　　次：2010 年 11 月第 1 次印刷
定　　价：35.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

模式识别诞生于 20 世纪 20 年代，在 20 世纪 60 年代初发展成为一门学科。随着计算机性能的不断提高，模式识别技术迅速发展，被广泛应用于人工智能、光学字符识别、生物身份认证、DNA 序列分析、人脸识别、手势识别、语音识别、信息检索、数据挖掘和信号处理、图像识别和理解、视频跟踪识别等。

全书共分 10 章。第 1 章为绪论，概述了模式识别的概念、方法和应用；第 2 章为贝叶斯决策理论，主要介绍了贝叶斯理论、正态分布模式、概率密度函数的估计等；第 3 章为线性判别函数，主要介绍了线性判别函数、感知器算法、最小平方误差准则函数、Fisher 线性判别函数等；第 4 章为模式特征提取与选择，主要介绍了离散 K-L 变换、离散傅里叶变换、离散余弦和正弦变换、小波变换等；第 5 章为聚类分析，主要介绍了相似性测度和聚类准则、层次聚类、*K* 均值聚类、ISODATA 聚类等算法；第 6 章为人工神经网络，主要介绍了人工神经网络的构成、多层前馈网络学习算法、联想记忆网络学习算法、Hamming 网络分类学习算法、径向基函数网络等；第 7 章为支持向量机，主要介绍了支持向量机的理论基础、常用的几种支持向量机、支持向量回归机等；第 8 章为核函数方法及应用，主要介绍了核函数的可分性条件、核函数的参数确定、核函数的构造方法、KPCA 等几种核方法；第 9 章为模糊模式识别，主要介绍了模糊数学的基本概念、最大隶属原则和择近原则的模糊模式识别基本方法、模糊 *C* 均值聚类算法、基于模糊等价矩阵的聚类分析等；第 10 章为模式识别应用，详细给出了车牌识别和语音识别的系统组成，以及实现方法和结果；附录 A 给出了模式识别文献中最著名的数据集之一——鸢尾属植物样本 Iris 数据；附录 B 给出了各章习题的详细解答。

本书是作者在结合多年教学实践和相关科研成果的基础上编写的。第 1、5、9、10 章由李晶皎、吴秀丽、宋光杰、吴鹏编写，第 6、7、8 章由赵丽红、王骄、杜玉远、李景宏编写，第 2、3、4 章由王爱侠、闫爱云、李贞妮、马学文编写。全书由李晶皎负责整理与统稿。

由于作者学识有限，书中难免有错误和不准确之处，恳请广大读者批评指正。

作　者

2010 年 8 月于东北大学

目　录

第 1 章　绪论 ····· (1)
1.1　模式和模式识别的概念 ····· (1)
1.2　模式识别的研究方法 ····· (1)
1.2.1　识别方法 ····· (1)
1.2.2　模式识别系统的组成 ····· (2)
1.3　模式识别的应用 ····· (3)
参考文献 ····· (5)

第 2 章　贝叶斯决策理论 ····· (6)
2.1　基于最小错误率的贝叶斯判别法 ····· (6)
2.2　基于贝叶斯公式的几种判别规则 ····· (10)
2.2.1　基于最小风险的贝叶斯决策 ····· (10)
2.2.2　最小最大决策 ····· (13)
2.3　正态分布模式的统计决策 ····· (15)
2.3.1　正态分布概率密度函数的定义及性质 ····· (15)
2.3.2　多元正态概率模型的贝叶斯判别函数 ····· (20)
2.4　概率密度函数的估计 ····· (24)
2.4.1　最大似然估计 ····· (25)
2.4.2　贝叶斯估计 ····· (28)
2.5　离散情况的贝叶斯决策 ····· (31)
2.6　贝叶斯分类器的错误率 ····· (33)
习题 2 ····· (37)
参考文献 ····· (37)

第 3 章　线性判别函数 ····· (39)
3.1　线性判别函数 ····· (39)
3.2　广义线性判别函数 ····· (42)
3.3　感知器算法 ····· (44)
3.3.1　基于赏罚概念的感知器训练算法 ····· (44)
3.3.2　梯度下降法 ····· (46)
3.4　最小平方误差准则函数 ····· (47)
3.5　多类问题 ····· (50)
3.5.1　多类问题的基本概念 ····· (50)
3.5.2　决策树简介 ····· (51)

3.6 Fisher 线性判别函数 ····· (54)
习题 3 ····· (56)
参考文献 ····· (57)
第 4 章 模式特征提取与选择 ····· (58)
4.1 离散 K-L 变换 ····· (58)
4.1.1 离散 K-L 展开式 ····· (59)
4.1.2 基于 K-L 变换的数据压缩 ····· (60)
4.1.3 基于 K-L 变换的特征提取 ····· (62)
4.2 离散傅里叶变换 ····· (64)
4.2.1 一维离散傅里叶变换 ····· (64)
4.2.2 二维离散傅里叶变换 ····· (65)
4.3 离散余弦和正弦变换 ····· (67)
4.3.1 余弦变换 ····· (67)
4.3.2 正弦变换 ····· (69)
4.4 Hadamard 变换 ····· (70)
4.5 Haar 变换 ····· (72)
4.6 小波变换 ····· (73)
4.6.1 连续小波变换 ····· (73)
4.6.2 离散小波变换 ····· (75)
4.6.3 多分辨率分析 ····· (75)
4.6.4 正交小波包 ····· (78)
习题 4 ····· (79)
参考文献 ····· (80)
第 5 章 聚类分析 ····· (81)
5.1 相似性测度和聚类准则 ····· (82)
5.1.1 相似性测度 ····· (82)
5.1.2 聚类准则 ····· (83)
5.2 聚类算法 ····· (86)
5.2.1 聚类算法的分类 ····· (86)
5.2.2 层次聚类算法 ····· (87)
5.2.3 *K* 均值算法 ····· (90)
5.2.4 核聚类 ····· (93)
5.2.5 ISODATA 算法 ····· (95)
5.3 聚类有效性 ····· (99)
习题 5 ····· (101)
参考文献 ····· (102)
第 6 章 人工神经网络 ····· (103)
6.1 人工神经网络的构成 ····· (103)

6.1.1 神经元的结构模型 ····· (103)
6.1.2 人工神经网络的连接方式 ····· (105)
6.1.3 神经网络模型分类 ····· (107)
6.1.4 神经网络学习规则 ····· (108)
6.2 多层前馈网络学习算法 ····· (109)
6.2.1 前馈网络模型 ····· (109)
6.2.2 感知器分类学习算法 ····· (113)
6.2.3 BP 网络分类学习算法 ····· (115)
6.3 联想记忆网络学习算法 ····· (118)
6.3.1 反馈网络模型 ····· (119)
6.3.2 联想记忆分类学习算法 ····· (124)
6.4 海明网络分类学习算法 ····· (127)
6.4.1 海明神经网络结构 ····· (127)
6.4.2 海明网络分类学习算法 ····· (128)
6.5 特征映射网络分类学习算法 ····· (130)
6.5.1 特征映射网络结构 ····· (130)
6.5.2 特征映射分类学习算法 ····· (131)
6.6 前馈网络分类机理 ····· (133)
6.6.1 前馈网络分类的几何机理 ····· (133)
6.6.2 前馈网络分类的代数机理 ····· (137)
6.7 径向基函数网络 ····· (139)
6.7.1 径向基函数 ····· (139)
6.7.2 径向基函数网络的特点 ····· (140)
6.7.3 径向基函数网络的正则化 ····· (142)
习题 6 ····· (145)
参考文献 ····· (146)

第 7 章 支持向量机 ····· (149)
7.1 最优分类超平面 ····· (149)
7.2 支持向量机的理论基础 ····· (153)
7.2.1 支持向量机的三种分类形式 ····· (153)
7.2.2 统计学习理论 ····· (160)
7.2.3 优化理论 ····· (166)
7.3 常用的几种支持向量机 ····· (168)
7.3.1 *C*-支持向量分类机 ····· (168)
7.3.2 *C*-支持向量机的变形 ····· (174)
7.3.3 广义支持向量机 ····· (175)
7.3.4 *v*-支持向量机 ····· (176)
7.4 支持向量回归机 ····· (178)

7.4.1 回归问题 ····· (178)
7.4.2 线性回归 ····· (179)
7.4.3 非线性回归 ····· (182)
7.4.4 ε-支持向量回归机 ····· (184)
7.4.5 v-支持向量回归机 ····· (185)
习题 7 ····· (187)
参考文献 ····· (187)

第 8 章 核函数方法及应用 ····· (189)
8.1 核函数的可分性条件 ····· (190)
8.1.1 输入空间中样本点线性可分的判别条件 ····· (190)
8.1.2 特征空间中样本点线性可分的判别条件 ····· (191)
8.2 核函数的参数确定 ····· (195)
8.3 核函数的构造方法 ····· (196)
8.3.1 基于特征变换的核函数构造 ····· (196)
8.3.2 利用 Mercer 核函数的性质组合核函数 ····· (197)
8.3.3 借助其他领域知识构造核函数 ····· (198)
8.4 几种核方法 ····· (198)
8.4.1 KPCA 的基本思想 ····· (198)
8.4.2 基于类内散布的最优 kernel PCA 展开方法 ····· (201)
8.4.3 融合先验类别信息的非线性主元分析算法 ····· (202)
8.4.4 PKPCA 与 KPCA 和 KFD 的关系 ····· (205)
习题 8 ····· (205)
参考文献 ····· (206)

第 9 章 模糊模式识别 ····· (207)
9.1 模糊数学的基本理论 ····· (207)
9.1.1 模糊集合 ····· (207)
9.1.2 模糊关系 ····· (210)
9.1.3 模糊集合的度量 ····· (213)
9.2 模糊模式识别的基本方法 ····· (217)
9.2.1 最大隶属原则 ····· (217)
9.2.2 择近原则 ····· (218)
9.3 模糊聚类分析方法 ····· (220)
9.3.1 基于模糊等价矩阵的聚类分析 ····· (220)
9.3.2 模糊 C 均值聚类算法 ····· (224)
9.3.3 模糊聚类的有效性 ····· (228)
习题 9 ····· (232)
参考文献 ····· (233)

第 10 章　模式识别应用 ····· (235)
10.1　车牌识别 ····· (235)
10.1.1　车牌图像预处理 ····· (235)
10.1.2　车牌定位 ····· (239)
10.1.3　字符分割 ····· (246)
10.1.4　字符识别 ····· (247)
10.2　语音识别 ····· (252)
10.2.1　语音识别研究的发展与现状 ····· (252)
10.2.2　语音识别方法简介 ····· (254)
10.2.3　DHMM 语音识别系统 ····· (256)
参考文献 ····· (280)
附录 A　鸢尾属植物样本数据(Iris Data) ····· (283)
附录 B　习题解答 ····· (285)
习题 2 ····· (285)
习题 3 ····· (288)
习题 4 ····· (289)
习题 5 ····· (292)
习题 6 ····· (298)
习题 7 ····· (299)
习题 8 ····· (302)
习题 9 ····· (303)

第1章 绪 论

1.1 模式和模式识别的概念

模式识别诞生于20世纪20年代。随着20世纪40年代计算机的出现，20世纪50年代人工智能的兴起，模式识别在20世纪60年代迅速发展成为一门学科。在20世纪60年代以前，模式识别主要限于统计学领域的理论研究，计算机的出现增加了对模式识别实际应用的需求，也推动了模式识别理论的发展。

在日常生活中，我们经常进行模式识别活动，例如，收听广播就是在做语音识别，阅读报纸就是在做文字识别，看照片就是在做图像识别。人通过自己的感觉器官从外界获取信息，经过思维、分析、判断，建立对客观世界各种事物的认识；人通过视觉获得形状、大小、色彩等信息，通过听觉获得各种声音的信息，通过触觉获得温度、湿度、材质等信息。人从各个方面获取信息，进行综合思维，认识各种客观事物。随着计算机的发展，人们一直希望计算机能够具有人的能力。

模式识别(Pattern Recognition)就是研究用计算机实现人类的模式识别能力的一门学科，目的是利用计算机将对象进行分类。这些对象与应用领域有关，它们可以是图像、信号，或者是任何可测量且需要分类的对象，对象的专业术语就是模式(Pattern)。按照广义的定义，存在于时间和空间中可观察的事物，如果可以区别它们是否相同或相似，都可以称为模式。

模式识别是一个多领域的交叉学科，它涉及人工智能、统计学、计算机科学、工程学、医学等众多的研究问题。例如，语音识别、字符识别、医学图像识别、医疗诊断、商品销售分析等，吸引了众多的研究人员，且人们提出了许多新方法。在20世纪80年代，基于知识的系统和神经网络发展迅速。近年来，在概率和统计交叉的领域取得重大进展，例如，核函数方法的核贝叶斯计算方法。到目前为止，模式识别的理论和技术还远未完善，尚有很多课题有待人们去研究和探索。

1.2 模式识别的研究方法

1.2.1 识别方法

根据所采用的数学模型，模式识别的分类主要如下。

(1)统计模式识别。主要是指依据模式特征数据的统计分析而建立的数学模型的方法。

(2)结构模式识别。主要依据模式内部结构关系数据和模式之间的结构关系数据，采用语言结构分析方法进行识别。

模式识别仍然是一门发展中的学科，新的理论和方法不断出现。在模式识别中应用模糊数学和人工神经网络的方法，取得了良好效果。

模糊模式识别就是利用模糊数学的理论和方法来分析解决模式识别问题。这种方法既有模糊数学基础，又接近于人的思维方法，因此适合于分类识别对象本身或要求识别结果具有模糊性的场合。典型的模糊模式识别方法有模糊 K 均值和模糊 ISODATA 算法。

人工神经网络是对人脑结构和功能的简单模拟和近似，它是由大量神经元相互连接构成的非线性动力学系统。人工神经网络在自学习、自组织、联想记忆及容错方面具有较大的能力，可以处理一些环境信息十分复杂、背景知识不清楚、推理规则不明确的识别问题。

1．统计模式识别

在统计模式识别中，被研究的模式用特征向量来描述，特征向量中的每一个元素代表模式的一个特征，特征向量构成的空间称为特征空间。一般情况下，合理的假设是：同类模式在特征空间中相距较近，而不同类模式在特征空间中相距较远。这是因为相距较近的模式的各个特征相差不多，属于同一类的可能性较大。如果用某种方法分割特征空间，使得同一类模式大体上都在特征空间的同一个区域中，对于待分类的模式，就可以根据它的特征向量位于特征空间中的哪一个区域来判定它属于哪一类模式。统计模式识别的任务就是用不同的方法划分特征空间，从而达到识别的目的。

2．结构模式识别

结构模式识别方法主要分析模式的结构信息。由于模式是由一些模式基元按一定的结构规则组合而成的，因此结构分析的内容就是分析模式如何由基元构成的规则。目前比较成功的是句法结构模式识别方法，它通过检查代表这个模式的句子是否符合事先给定的某一类文法规则来达到识别的目的，即如果符合，那么该模式就属于这个文法所代表的那个模式类。

介绍模式识别最新方法和理论的中英文期刊很多，主要有《模式识别与人工智能》、《中国图像图形学报》、*Pattern Recognition Letters*、*Pattern Recognition*、*IEEE Transactions on Pattern Analysis and Machine Intelligence* 等，以及有关的图像处理和模式识别应用类杂志。

1.2.2 模式识别系统的组成

本节以统计模式识别为例，介绍模式识别系统的组成。图1.1给出了一个统计模式识别系统的简单框图，它主要由信息获取、预处理、特征提取与选择、分类器设计、分类决策这五个模块组成。

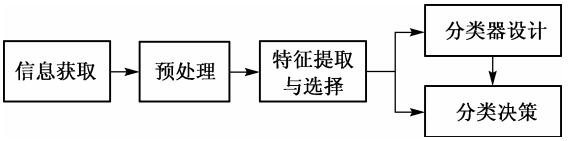


图 1.1 模式识别系统的基本构成

1．信息获取

为了使计算机能够分类识别对象(模式)，必须首先将对象用计算机所能接受的形式表示。目前待识别模式大多是非电量输入模式，如灰度、色彩、声音、压力、温度等，需要将

这些以各种不同形式表现的信息通过相应的传感器转换成电信号，然后经过 A/D 变换，最终转换为计算机能接受的数字量。通常输入对象的信息有下列三种类型：

(1) 二维图像。如指纹、照片、文字等。

(2) 一维图像。如语音信号、心电图、机械振动波等。

(3) 物理参数和逻辑值。如体温、各种实验数据等。

通过测量、采样和量化，可以用矩阵或向量表示二维图像或一维波形。这就是信息的获取过程。

2．预处理

预处理的目的是去除信息获取过程中掺入的干扰和噪声，人为地加强有用信息，并对各种因素造成的退化现象进行复原。

3．特征提取与选择

由信息获取部分得到的原始数据量一般都很大。例如，一幅文字图像可以有几千个数据，100 ms 的语音信号也有几千个数据。为了有效地实现分类识别，要对原始数据进行选择和变换，得到最能反映分类本质的特征，而由这些特征组成的向量则称为特征向量。上述过程就是特征提取与选择过程。一般我们把原始数据组成的空间称为测量空间，把经过特征提取与选择后得到的特征向量空间称为特征空间。特征空间中的一个点就是一个特征向量，它代表一个模式或样本，特征向量的每一个分量就是模式的一个特征。在模式识别中，特征提取与选择占有重要的地位，但尚无通用的理论指导，只能通过分析具体识别对象来确定选取何种特征。

4．分类器设计

为了把待识别模式分配到各自的模式类中，必须设计出一套分类判别规则。基本做法是：用一定数量的样本(或训练样本集)确定出一套分类判别规则，使得按这套分类规则对待识别模式进行分类所造成的错误识别率最小或引起的损失最小。这就是分类器设计过程。

5．分类决策

分类器按照已经确定的分类规则将待识别模式进行分类判别，输出分类结果。这就是分类器的使用过程，也称为分类决策。

在模式识别系统中，信息获取和预处理部分通常是数字信号处理和数字图像处理等课程的研究课题，本书只讨论特征提取与选择、分类器设计以及分类决策的理论和方法。

1.3　模式识别的应用

随着模式识别的迅速发展，模式识别技术已在越来越多的领域中得到应用。

1．文字识别

迄今为止，在模式识别领域中发展最成熟、应用最广泛的一个方面就是文字识别。各种成熟的光学字符识别(Optical Character Recognition, OCR)系统已经在使用，如可识别手写体

阿拉伯数字的邮政信函自动分拣系统，可以识别数字及数字个数的银行支票机器识别系统，等等。

按照识别对象划分，文字识别分为英文字符识别、阿拉伯数字识别和汉字识别。按照书写方式划分，文字识别可分为印刷体识别和手写体识别。由于汉字结构复杂、种类多，因而汉字识别难度最大。联机手写汉字识别，由于利用书写板输入了汉字的笔顺信息，因此降低了识别难度，故已有实际应用；而脱机手写汉字识别还处于实验室阶段。

2．语音识别

由于语音是人类最自然的沟通和交换信息的方式，因而成为计算机人机接口的关键技术。从原理上看，语音识别虽然实现起来并不困难，但在实际实现时会遇到很多困难，主要表现如下：①发音的多变性。不同人发同一个音、同一个人在不同的条件下发同一个音等，都会有不同的发音特征参数；②发音的模糊性。在实际的连续语音流中，语音声学变量与音素变量之间不存在一一对应的关系，语音流中存在变化多端的音变现象，这些音变对人类的听觉系统来说很容易辨认，但机器识别却很不容易。

语音识别的应用很广，如声控打字、用声音控制计算机等。如将语音识别与语音合成结合起来，则可以实现甚低比特率的语音通信。目前从事语音识别研究的组织很多，如 IBM、Microsoft、Motorola、Siemens、Nokia、Toshiba 等公司，以及美国的卡内基·梅隆大学(CMU)、麻省理工学院(MIT)和我国的清华大学等。一些中、小词汇量的孤立词或连续语音识别系统已经进入市场。据预测，带有语音功能的计算机将很快成为大众化产品，语音识别将可能取代键盘和鼠标成为计算机的主要输入手段。

3．指纹识别

手掌及其手指、脚、脚趾内侧表面的皮肤凹凸不平所产生的纹路会形成各种各样的图案，而这些皮肤的纹路在图案、断点和交叉点上各不相同，具有唯一性。依靠这种唯一性，就可以将一个人与其指纹对应起来，通过将其指纹和预先保存的指纹进行比较，便可以验证他的真实身份。从 20 世纪 60 年代开始，随着计算机技术的发展，人们开始研究利用计算机处理指纹，自动指纹识别系统在法律实施方面的研究和应用已在世界许多国家展开。20 世纪 80 年代，随着计算机和光学扫描技术的迅速发展，使得它们作为指纹取像工具成为现实，从而指纹识别得到了广泛应用。20 世纪 90 年代后期，随着低价位的取像设备的飞速发展，以及可靠的比对识别算法的实现，指纹识别在个人身份识别中得到了广泛应用。

4．生物医学应用

模式识别已经广泛应用于生物医学，例如，心电图和心电向量图的分析，脑电图的分析，染色体的自动分类，癌细胞分类，血相分析，X 光片、CT 片、磁共振片等医学图片的分析。

5．其他方面的应用

(1)对地球资源和环境的调查研究。在这类应用中，人们利用遥感卫星或飞机获取的大量信息，经过图像处理、模式识别来调查资源情况，例如调查林业资源、水资源及矿藏资源的分布，监测大气变化，调查环境污染，调查土地规划和利用状况，进行农作物长势监测，虫

灾监测，等等。

(2) 生产自动化过程中的应用。在这类应用中，人们使用模式识别来对自动化生产线上的产品进行质量检验。例如，在大规模集成电路的生产中，芯片内的缺损检验以及芯片上引出端的自动识别和引出线焊接等。

(3) 军事应用。在这类应用中，人们使用模式识别来对可见光、雷达、红外线图像进行分析与识别，检出和鉴别目标的出现，判断目标的类别，并对运动中的目标进行监视和跟踪；采用地形匹配方法校正飞行器轨道，提高导弹命中率；等等。

参考文献

[1] 边肇祺编著. 模式识别(第二版). 北京：清华大学出版社，2000.

[2] 杨光正编著. 模式识别. 合肥：中国科学技术大学出版社，2007.

[3] 钟珞编著. 模式识别. 武汉：武汉大学出版社，2006.

[4] Andrew R. Webb 著, 王萍译. 统计模式识别(第二版). 北京：电子工业出版社，2004.

[5] Richard O. Duda 著, 李宏东译. 模式分类. 北京：机械工业出版社，2003.

[6] 蔡元龙编. 模式识别. 西安：西北电讯工程学院出版社, 1986.

[7] J. P. Marques de Sa 著, 吴逸飞译. 模式识别——原理、方法及应用. 北京：清华大学出版社，2002.

[8] 苏宁编著. 模式识别的理论与方法. 武汉：武汉大学出版社，2004.

[9] Sergios Theodoridis 著, 李晶皎译. 模式识别(第三版). 北京：电子工业出版社，2006.

[10] 张宏林编著. Visual C++数字图像模式识别技术及工程实践. 北京：人民邮电出版社，2003.

[11] Lawrence Rabiner. *Fundamentals of Speech Recognition.* 北京：清华大学出版社，1999.

[12] Alan Bovik. *Handbook of Image and Video Processing, Second Edition.* 北京：电子工业出版社，2006.

第 2 章　贝叶斯决策理论

在模式识别方法中，通过模式分类将特征空间分割成若干区域，使每个区域对应一个模式类别。在理想情况下，基于这些区域分割而进行的决策不应该产生错误，如果在实际中做不到这一点，则要求分类的错误代价尽量小；如果一些错误比另一些错误要付出更多的代价，则要求分类错误的平均代价最小[1]。

模式识别的一种主要处理方法是贝叶斯(Bayes)决策理论。这种方法的基本思路是，在假设决策问题可以用概率的形式描述，并且所有有关的概率结构均已知的前提下，决策者根据已经获得的历史资料数据以及主观知识(包括经验、直觉、判断等)，对未来事件发生的概率做出主观估计(即先验概率)，最后根据期望值的计算结果做出决策选择。由于先验状态分布与实际情况存在一定的误差，所以它很难准确地反映客观真实情况，而且有时候决策结果对先验概率又非常敏感，所以必须通过市场调查等方法收集有关自然状态的补充信息，以修正对事件的先验概率估计(得到后验概率)，最后用后验状态分布进行决策。贝叶斯决策理论提供了一种修正先验概率的科学方法。

用贝叶斯理论进行分类时要求满足两点：第一，要决策的类别数是一定的。例如两类样本(正常状态 ω_1 和异常状态 ω_2)，或 L 类样本 ω_1，ω_2，…，ω_L；第二，各类别总体的概率分布是已知的，即每一类样本出现的先验概率 $p(\omega_i)$ 以及各类概率密度函数 $p(\boldsymbol{x}\mid\omega_i)$ 是已知的。显然，$0\leq p(\omega_i)\leq 1, i=1,2,\cdots,L$，且 $\sum_{i=1}^{L}p(\omega_i)=1$[2]。对于两类故障诊断问题，相当于在识别前已知正常状态 ω_1 的概率 $p(\omega_1)$ 和异常状态 ω_2 的概率 $p(\omega_2)$，它们是由先验知识确定的状态先验概率。如果不做进一步的仔细观测，仅依靠先验概率做决策，那么就会给出这样的决策规则：若 $p(\omega_1)>p(\omega_2)$，则做出状态属于 ω_1 类的决策；反之，则做出状态属于 ω_2 类的决策。例如，某设备在 365 天运行中，发生故障是少见的，无故障是经常的，有故障的概率远小于无故障的概率。因此，若无特别明显的异常状况，就应判断为无故障。由于只利用先验概率提供的分类信息太少了，所以这样判断对于某一个实际的待检状态根本达不到诊断的目的。为此，我们还要对系统状态进行状态检测，分析所观测到的信息。根据观测信息，结合先验概率再对状态进行归类。

如今贝叶斯理论广泛应用于各个领域，如工程技术、管理科学、系统运筹、医疗诊断等。本章介绍这种方法的基本内容。

2.1　基于最小错误率的贝叶斯判别法

在模式分类中，要尽量减少分类错误的概率。从这样的要求出发，利用概率论中的贝叶斯公式，能得到使错误率最小的分类规则，这称为基于最小错误率的贝叶斯判别法[3]。

假定一个两类问题，先验概率分别为 $p(\omega_1)$ 和 $p(\omega_2)$。令 $\boldsymbol{x}$ 为 N 维向量，$\boldsymbol{x}$ 的类条件概率密度为 $p(\boldsymbol{x}\mid\omega_i), i=1,2$。根据全概率公式，模式样本 $\boldsymbol{x}$ 出现的全概率密度为

$$p(\boldsymbol{x})=\sum_{i=1}^{2}p(\boldsymbol{x}\mid\omega_i)p(\omega_i) \tag{2.1}$$

根据贝叶斯公式，在模式 $\boldsymbol{x}$ 出现的条件下，两个类的后验概率为

$$p(\omega_i\mid\boldsymbol{x})=\frac{p(\boldsymbol{x}\mid\omega_i)p(\omega_i)}{p(\boldsymbol{x})}=\frac{p(\boldsymbol{x}\mid\omega_i)p(\omega_i)}{\sum_{i=1}^{2}p(\boldsymbol{x}\mid\omega_i)p(\omega_i)} \tag{2.2}$$

因此，贝叶斯公式实际上是通过观察 $\boldsymbol{x}$，把状态的先验概率 $p(\omega_i)$ 转化为状态的后验概率 $p(\omega_i\mid\boldsymbol{x})$。要判断 $\boldsymbol{x}$ 是属于 ω_1 类还是属于 ω_2 类，从概率统计的观点来看，$\boldsymbol{x}$ 来自于哪类的概率大，就属于哪类，这样能够使错误概率最小。因此基于最小错误率的贝叶斯判别规则为

$$\text{如果 } p(\omega_1\mid\boldsymbol{x})>p(\omega_2\mid\boldsymbol{x})\text{，那么 }\boldsymbol{x}\in\omega_1\text{；否则 }\boldsymbol{x}\in\omega_2 \tag{2.3}$$

由于 $p(\omega_1\mid\boldsymbol{x})+p(\omega_2\mid\boldsymbol{x})=1$，因此上述规则可写成

$$\begin{cases}\text{如果 } p(\omega_1\mid\boldsymbol{x})>\dfrac{1}{2}\text{，那么}\boldsymbol{x}\in\omega_1\\ \text{如果 } p(\omega_2\mid\boldsymbol{x})>\dfrac{1}{2}\text{，那么}\boldsymbol{x}\in\omega_2\end{cases} \tag{2.4}$$

由贝叶斯定理，上述规则可进一步表示成

$$\begin{cases}\text{如果 } p(\omega_1)p(\boldsymbol{x}\mid\omega_1)>p(\omega_2)p(\boldsymbol{x}\mid\omega_2)\text{，那么}\boldsymbol{x}\in\omega_1\\ \text{如果 } p(\omega_1)p(\boldsymbol{x}\mid\omega_1)<p(\omega_2)p(\boldsymbol{x}\mid\omega_2)\text{，那么}\boldsymbol{x}\in\omega_2\end{cases} \tag{2.5}$$

或者可以写成

$$\begin{cases}\text{如果 } l_{12}(\boldsymbol{x})=\dfrac{p(\boldsymbol{x}\mid\omega_1)}{p(\boldsymbol{x}\mid\omega_2)}>\dfrac{p(\omega_2)}{p(\omega_1)}=\theta_{21}\text{，那么}\boldsymbol{x}\in\omega_1\\ \text{如果 } l_{12}(\boldsymbol{x})=\dfrac{p(\boldsymbol{x}\mid\omega_1)}{p(\boldsymbol{x}\mid\omega_2)}<\dfrac{p(\omega_2)}{p(\omega_1)}=\theta_{21}\text{，那么}\boldsymbol{x}\in\omega_2\end{cases} \tag{2.6}$$

其中，$l_{12}(\boldsymbol{x})$ 称为似然比函数，θ_{21} 称为似然比的判决阈值[4]。

【例 2.1】 对一大批人进行癌症普查，设 ω_1 类代表正常人，ω_2 类代表癌症患者。已知先验概率如下：正常状态 $p(\omega_1)=0.9$，异常状态 $p(\omega_2)=0.1$。以一个化验结果作为特征 $\boldsymbol{x}$:{阳性，阴性}，正常人和癌症患者的化验结果为阳性的概率分别为

$$p(\boldsymbol{x}=\text{阳性}\mid\omega_1)=0.2\text{，}\quad p(\boldsymbol{x}=\text{阳性}\mid\omega_2)=0.4$$

现有一人化验结果为阳性，问此人是否为癌症患者[2]?

解：利用贝叶斯公式，分别计算出 ω_1 及 ω_2 的后验概率如下：

$$p(\omega_1\mid\boldsymbol{x})=\frac{p(\boldsymbol{x}\mid\omega_1)p(\omega_1)}{\sum_{j=1}^{2}p(\boldsymbol{x}\mid\omega_j)p(\omega_j)}=\frac{0.2\times0.9}{0.2\times0.9+0.4\times0.1}=0.818$$

$$p(\omega_2\mid\boldsymbol{x})=1-p(\omega_1\mid\boldsymbol{x})=0.182$$

根据贝叶斯决策规则公式，有

$$p(\omega_1 \mid x) > p(\omega_2 \mid x)$$

所以合理的判别是把 $\boldsymbol{x}$ 归类于 ω_1，属于正常状态。

从这个例子可以看出，决策结果取决于实际观察到的先验概率 $p(\omega_i)$ 和类条件概率密度 $p(\boldsymbol{x} \mid \omega_i)$，根据先验概率 $p(\omega_i)$ 和类条件概率密度 $p(\boldsymbol{x} \mid \omega_i)$ 计算后验概率 $p(\omega_1 \mid \boldsymbol{x})$ 和 $p(\omega_2 \mid \boldsymbol{x})$，基于最小错误率的贝叶斯决策理论就是根据后验概率的大小进行分类决策的。因为 $p(\omega_1 \mid \boldsymbol{x}) > p(\omega_2 \mid \boldsymbol{x})$，所以做出 $\boldsymbol{x} \in \omega_1$ 的决策；而如果 $p(\omega_1 \mid \boldsymbol{x}) < p(\omega_2 \mid \boldsymbol{x})$，则 $\boldsymbol{x} \in \omega_2$。在这个例子中，由于状态 ω_1 的先验概率比 ω_2 的先验概率大几倍，故先验概率在最后决策中起了主导作用。

在前面我们只是给出了最小错误率贝叶斯决策规则，而没有证明按照这种规则进行分类能够使错误率最小。现在仅以一维情况为例来完成证明，其结果不难推广到多维情形。

所谓错误率，是指平均错误率，它以 $P(e)$ 来表示，其定义为

$$P(e) = \int_{-\infty}^{+\infty} p(e, \boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{-\infty}^{+\infty} p(e \mid \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{2.7}$$

其中 $\int_{-\infty}^{+\infty} (\,) \mathrm{d}\boldsymbol{x}$ 表示在整个特征空间上的积分。

对于两类别问题，由式(2.3)可知，如果 $p(\omega_1 \mid \boldsymbol{x}) < p(\omega_2 \mid \boldsymbol{x})$，那么判断结果应为 $\boldsymbol{x} \in \omega_2$。显然在做出决策 $\boldsymbol{x} \in \omega_2$ 时，$\boldsymbol{x}$ 的条件错误概率为 $p(\omega_1 \mid \boldsymbol{x})$；反之，$\boldsymbol{x}$ 的条件错误概率应为 $p(\omega_2 \mid \boldsymbol{x})$。从而 $\boldsymbol{x}$ 的条件错误概率可表示为

$$p(e \mid \boldsymbol{x}) = \begin{cases} p(\omega_1 \mid \boldsymbol{x}), & \text{当} p(\omega_1 \mid \boldsymbol{x}) < p(\omega_2 \mid \boldsymbol{x}) \\ p(\omega_2 \mid \boldsymbol{x}), & \text{当} p(\omega_1 \mid \boldsymbol{x}) > p(\omega_2 \mid \boldsymbol{x}) \end{cases} \tag{2.8}$$

如果令 t 为 ω_1，ω_2 两类的分界面，则当特征向量 $\boldsymbol{x}$ 是一维向量时，t 为 X 轴上的一点，而且 t 点将 X 轴分为两个区域 R_1 和 R_2。R_1 的范围为 $(-\infty, t)$，R_2 的范围为 $(t, +\infty)$，这样就有

$$\begin{aligned} P(e) &= \int_{-\infty}^{t} p(\omega_2 \mid \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \int_{t}^{+\infty} p(\omega_1 \mid \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \int_{-\infty}^{t} p(\boldsymbol{x} \mid \omega_2) p(\omega_2) \mathrm{d}\boldsymbol{x} + \int_{t}^{+\infty} p(\boldsymbol{x} \mid \omega_1) p(\omega_1) \mathrm{d}\boldsymbol{x} \end{aligned} \tag{2.9}$$

上式可以写为

$$\begin{aligned} P(e) &= p(\boldsymbol{x} \in R_1, \omega_2) + p(\boldsymbol{x} \in R_2, \omega_1) \\ &= p(\boldsymbol{x} \in R_1 \mid \omega_2) p(\omega_2) + p(\boldsymbol{x} \in R_2 \mid \omega_1) p(\omega_1) \\ &= p(\omega_2) \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}x + p(\omega_1) \int_{R_2} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}x \\ &= p(\omega_2) P_2(e) + p(\omega_1) P_1(e) \end{aligned} \tag{2.10}$$

所以总的错误率是两种分类错误率的加权和。

图2.1[1]说明了一维模式的情况。t 为 R_1 和 R_2 区域的分界。显然 t 的位置不同，则错误率也不同。图中两个画线(斜线和纹线)部分之和，包括黑色区域 A，为总的错误率。当把决策面 t 左移时，可以减小代表误分类的三角区域 A 的面积，从而减小分类错误概率。若选取决策面 t 使得

$$p(\omega_2)p(\boldsymbol{x}\mid\omega_2)=p(\omega_1)p(\boldsymbol{x}\mid\omega_1)\text{，即 } t=t_0$$

则可以消除面积 A，从而得到最小的分类错误率。显而易见，也只有这种划分才能使对应的错误率区域面积最小。这正是贝叶斯决策理论得到的结果。

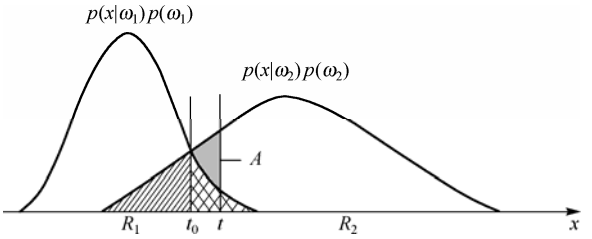


图 2.1　一维模式下的错误率

贝叶斯决策式(2.3)实际上是对每个特征值 $\boldsymbol{x}$ 都使 $p(e|\boldsymbol{x})$取较小值，这就使得公式(2.7)表示的整个积分也必然达到最小，即平均错误率 $P(e)$达到最小。这就证明了最小错误率贝叶斯判别法确实使错误率最小。以上一维情况下的讨论不难推广到 N 维情形。

根据两类问题的讨论，不难得到多类问题(c 类)基于最小错误率的贝叶斯判别法，即

$$\text{如果 } p(\omega_i\mid\boldsymbol{x})>p(\omega_j\mid\boldsymbol{x}),\ j=1,2,\cdots,c,\quad \forall j\neq i\text{，那么 }\boldsymbol{x}\in\omega_i \tag{2.11}$$

或者表示为

$$\text{如果 } p(\omega_i\mid\boldsymbol{x})=\max_{j=1,2,\mathrm{L},c}p(\omega_j\mid\boldsymbol{x})\text{，那么 }\boldsymbol{x}\in\omega_i \tag{2.12}$$

也可以表示为

$$\text{如果 } p(\omega_i)p(\boldsymbol{x}\mid\omega_i)>p(\omega_j)p(\boldsymbol{x}\mid\omega_j),\quad j=1,2,\cdots,c,\forall j\neq i\text{，那么 }\boldsymbol{x}\in\omega_i \tag{2.13}$$

由式(2.6)，多类模式的最小错误率贝叶斯判别法也可以表示为

$$\text{如果 } l_{ij}=\frac{p(\boldsymbol{x}\mid\omega_i)}{p(\boldsymbol{x}\mid\omega_j)}>\frac{p(\omega_j)}{p(\omega_i)}=\theta_{ji},\quad j=1,2,\mathrm{L},c,\forall j\neq i\text{，那么 }\boldsymbol{x}\in\omega_i \tag{2.14}$$

只要给出各类的类条件概率密度函数 $p(\boldsymbol{x}\mid\omega_i)$(或似然比 l_{ij})和各类的先验概率 $p(\omega_i)$(或判别阈值 θ_{ji})，就可对任意模式样本 $\boldsymbol{x}$ 按照最小错误率判别法进行分类。

在多类决策过程中，要把特征空间分割成 $R_1, R_2, \cdots, R_c$ 个区域，可能错分的情况很多。平均错误概率 $P(e)$将由 $c(c-1)$项组成，即

$$\begin{aligned}P(e)=&\underbrace{\begin{aligned}&[p(x\in R_2\mid\omega_1)+p(x\in R_3\mid\omega_1)+\cdots+p(x\in R_c\mid\omega_1)]p(\omega_1)+\\&[p(\boldsymbol{x}\in R_1\mid\omega_2)+p(\boldsymbol{x}\in R_3\mid\omega_2)+\cdots+p(\boldsymbol{x}\in R_c\mid\omega_2)]p(\omega_2)+\\&\cdots+\\&[p(\boldsymbol{x}\in R_1\mid\omega_c)+p(\boldsymbol{x}\in R_2\mid\omega_c)+\cdots+p(\boldsymbol{x}\in R_{c-1}\mid\omega_c)]p(\omega_c)\end{aligned}}_{\text{每行}c-1}\Big\}c\text{行}\\=&\sum_{i=1}^{c}\sum_{\substack{j=1\\j\neq i}}^{c}[p(x\in R_j\mid\omega_i)]p(\omega_i)\end{aligned} \tag{2.15}$$

由式(2.15)可见直接求 $P(e)$的计算量比较大，如果先计算平均正确分类概率 $P(c)$，

$$P(c)=\sum_{j=1}^{c}p(\boldsymbol{x}\in R_j\mid\omega_j)p(\omega_j)=\sum_{j=1}^{c}\int_{R_j}p(\boldsymbol{x}\mid\omega_j)p(\omega_j)\mathrm{d}x \tag{2.16}$$

上式中求和号内只有 c 项，而 $P(e)=1-P(c)$，则要比直接计算 $P(e)$简单得多。

在介绍了最小错误率贝叶斯决策法的几种等价形式之后，下面简单说明判别函数和决策面的概念。

对于 m 类问题，按照决策规则可以把多维特征空间分成 m 个类别区域，划分这些区域的界面称为决策面，在数学上可以表示为决策面方程的形式，用于描述决策规则的某种函数称为判别函数。相邻的两个类别在决策面上的判别函数值相等。如果 ω_i 与 ω_j 相邻，则分割它们的决策面应为[5]

$$d_i(\boldsymbol{x})=d_j(\boldsymbol{x})\text{ 或 }d_i(\boldsymbol{x})-d_j(\boldsymbol{x})=0 \tag{2.17}$$

其中判别函数

$$d_k(\boldsymbol{x})=p(\omega_k)p(\boldsymbol{x}\mid\omega_k),\quad k=i,j \tag{2.18}$$

例如对于两类问题，由式(2.5)和式(2.17)可以得到决策面方程为

$$p(\omega_1)p(\boldsymbol{x}\mid\omega_1)-p(\omega_2)p(\boldsymbol{x}\mid\omega_2)=0$$

2.2 基于贝叶斯公式的几种判别规则

2.1 节介绍了基于最小错误率的贝叶斯决策规则，并且证明了应用这种决策规则时，能够使平均错误率最小。但当接触实际问题时，会发现使错误率最小并不一定是一个普遍适用的最佳选择。本节介绍另外两种贝叶斯决策规则。

2.2.1 基于最小风险的贝叶斯决策

在实际应用中，有时需要考虑一个比错误率更为广泛的概念——风险(或称为决策代价、决策损失)。事实上，在同一个问题中，不同的判断会产生不同的损失，特别是错误的判断会带来风险，不同的错误判断产生的风险不同[4]。

以癌细胞识别为例。我们对细胞的分类不仅要考虑到尽可能做出正确的判断，而且还要考虑做出错误判断时会带来什么后果。诊断中把正常细胞判为异常固然会给病人带来精神上的负担，而将本来就是异常的情况错判为正常，就会使早期的癌变患者失去进一步检查的机会，进而造成严重的后果。显然这两种不同的错误判断所造成的损失严重程度是明显不同的，后者的损失比前者的要严重得多。最小风险贝叶斯决策正是考虑各种错误造成的不同损失而提出的一种决策规则[1]。

考虑一个 c 类问题。假设用 $\omega_j(j=1,2,\cdots,c)$ 表示 c 个类别，决策方法有 p 种，即 $\{a_1,a_2,\cdots,a_p\}$，定义损失函数 $\lambda_{ij},i=1,2,\cdots,p;\ j=1,2,\cdots,c$，这个函数表示本应属于 ω_j 类的样本在采取决策 a_i 时所带来的损失代价(或称惩罚因子)。用决策表可以一目了然地表示各种情况下的决策损失，如表 2.1 所示。

下面我们在已知先验概率 $p(\omega_j)$ 和类条件概率密度 $p(\boldsymbol{x}|\omega_j)$，$j$ =1, 2, …, c 的条件下讨论最小风险贝叶斯决策。

表 2.1　一般决策表

状态 / 决策	自然状态					
	ω_1	ω_2	…	ω_j	…	ω_c
a_1	λ_{11}	λ_{12}	…	λ_{1j}	…	λ_{1c}
a_2	λ_{21}	λ_{22}	…	λ_{2j}	…	λ_{2c}
⋮	⋮	⋮	…	⋮	⋮	⋮
a_i	λ_{i1}	λ_{i2}	…	λ_{ij}	…	λ_{ic}
⋮	⋮	⋮	…	⋮	⋮	⋮
a_p	λ_{p1}	λ_{p2}	…	λ_{pj}	…	λ_{pc}

根据贝叶斯公式，后验概率 $p(\omega_j|\boldsymbol{x})$ 为

$$p(\omega_j|\boldsymbol{x})=\frac{p(\boldsymbol{x}|\omega_j)p(\omega_j)}{p(\boldsymbol{x})}=\frac{p(\boldsymbol{x}|\omega_j)p(\omega_j)}{\sum_{i=1}^{c}p(\boldsymbol{x}|\omega_i)p(\omega_i)} \tag{2.19}$$

当引入“损失代价”的概念后，考虑错判所造成的后果时，就不能只根据后验概率的大小来做决策，而必须考虑所采取的决策能否使损失最小。对于给定的 N 维随机向量 $\boldsymbol{x}$，从决策表中可以看出，如果采取决策 a_i，λ 可以在 c 个 $\lambda_{ij}, j=1,2,\cdots,c$ 中任取一个，其相应的概率为 $p(\omega_j|\boldsymbol{x})$。因此，在采取决策 a_i 的情况下，条件期望损失(也称为条件风险) $R(a_i|\boldsymbol{x})$ 定义为 $\lambda_{ij}(j=1,2,\cdots,c)$ 与对应概率的加权和，即

$$R(a_i|\boldsymbol{x})=E\{\lambda_{ij}\}=\sum_{j=1}^{c}\lambda_{ij}p(\omega_j|\boldsymbol{x}),\ i=1,2,\cdots,p \tag{2.20}$$

因为 $\boldsymbol{x}$ 是随机向量的观察值，对于 $\boldsymbol{x}$ 的不同观察值，采取决策 a_i 时，其条件风险不同。所以究竟采取哪一种决策将根据 $\boldsymbol{x}$ 的取值来决定。决策 a_i 可以视为随机向量 $\boldsymbol{x}$ 的函数，记为 $a_i(\boldsymbol{x})$，它本身也是一个随机变量，于是可以定义期望风险 R 为

$$R=\int R(a_i(\boldsymbol{x})|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.21}$$

式中，$\mathrm{d}\boldsymbol{x}$ 是特征空间的体积元，积分在整个特征空间进行。

期望风险 R 反映了对于整个特征空间，所有 $\boldsymbol{x}$ 的取值采取相应的决策 $a_i(\boldsymbol{x})$时所带来的平均风险；而条件风险 $R(a_i|\boldsymbol{x})$ 只反映了某一个 $\boldsymbol{x}$ 的取值在采取决策 a_i 时所带来的风险。显然需要采取一系列决策 a 使期望风险 R 最小。如果在采取每一个决策时都使条件风险最小，则对所有的 $\boldsymbol{x}$ 做出决策时，其期望风险也必然最小，这样的决策称为最小风险贝叶斯决策[3]。

若不考虑拒识，则最小风险贝叶斯决策为

$$\text{如果 } R(a_k|\boldsymbol{x})=\min_{j=1,\perp,p}R(a_j|\boldsymbol{x})\text{，则 } a=a_k \tag{2.22}$$

也可以表示为

$$\text{如果 } R(a_i|\boldsymbol{x})<R(a_j|\boldsymbol{x})\text{，}j=1,2,\cdots,p,\ \forall j\neq i\text{，那么 }\boldsymbol{x}\in\omega_i \tag{2.23}$$

在处理实际问题时，最小风险贝叶斯决策可按如下步骤进行[3]。

(1)在已知 $p(\omega_j)$，$p(\boldsymbol{x}|\omega_j)$，$j = 1, 2, \cdots, c$ 及给出待识别模式 $\boldsymbol{x}$ 的情况下，根据贝叶斯公式计算出后验概率：

$$p(\omega_j|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_j)p(\omega_j)}{\sum_{i=1}^{c} p(\boldsymbol{x}|\omega_i)p(\omega_i)}, j = 1, \cdots, c$$

(2)利用计算出的后验概率及决策表，由式(2.20)计算出采取决策 a_i ($i = 1, 2, \cdots, p$)时的条件风险 $R(a_i|\boldsymbol{x})$：

$$R(a_i|\boldsymbol{x}) = \sum_{j=1}^{c} \lambda_{ij} p(\omega_j|\boldsymbol{x}),\ i = 1, 2, \cdots, p$$

(3)对(2)中得到的 p 个条件风险值 $R(a_i|\boldsymbol{x})$，$i = 1, 2, \cdots, p$ 进行比较，找出使条件风险最小的 a_k，即

$$R(a_k|\boldsymbol{x}) = \min_{j=1,2,\cdots,p} R(a_j|\boldsymbol{x})$$

则 a_k 就是最小风险贝叶斯决策。

可以看出，最小风险贝叶斯决策除了要有符合实际情况的先验概率 $p(\omega_j)$ 和类条件概率密度函数 $p(\boldsymbol{x}|\omega_j)$，$j = 1, 2, \cdots, c$ 外，还必须有合适的损失函数 λ_{ij}，$j = 1,2, \cdots, c$，$i = 1, 2, \cdots, p$。在实际工作中，要根据具体问题分析错误决策带来的损失代价，与有关专家共同商讨来确定。

下面我们来简单讨论一下最小错误率贝叶斯决策与最小风险贝叶斯决策之间的关系。

设损失函数为

$$\lambda_{ij} = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases} \quad i, j = 1, 2, \cdots, c \tag{2.24}$$

即正确分类决策时无损失；当采取错误决策时，损失代价都为 1，即损失程度一致，这样的损失函数称为 0-1 损失函数。此时条件风险为

$$R(a_i|\boldsymbol{x}) = \sum_{j=1}^{c} \lambda_{ij} p(\omega_j|\boldsymbol{x}) = \sum_{j=1, j\neq i}^{c} p(\omega_j|\boldsymbol{x}) = 1 - p(\omega_i|\boldsymbol{x}) \tag{2.25}$$

式中，$\sum_{j=1, j\neq i}^{c} p(\omega_j|\boldsymbol{x})$ 表示将 $\boldsymbol{x}$ 分类到 ω_i 时的条件错误率，$p(\omega_i|\boldsymbol{x})$ 是决策正确时的条件概率。所以在采用 0-1 损失函数时，使 $R(a_k|\boldsymbol{x}) = \min_{j=1,2,\cdots,c} R(a_j|\boldsymbol{x})$ 的最小风险贝叶斯决策就等价于

$$\sum_{j=1, j\neq i}^{c} p(\omega_j|\boldsymbol{x}) = \min_{i=1,2,\cdots,c} \sum_{j=1, j\neq i}^{c} p(\omega_j|\boldsymbol{x}) = \min_{i=1,2,\cdots,c} (1 - p(\omega_i|\boldsymbol{x}))$$

时的最小错误率贝叶斯决策。所以最小错误率贝叶斯决策是最小风险贝叶斯决策的一个特例。

【例 2.2】 对一大批人进行癌症普查，设 ω_1 类代表正常人，ω_2 类代表癌症患者。已知先验概率如下：正常状态，$p(\omega_1) = 0.9$；异常状态，$p(\omega_2) = 0.1$。

以一个化验结果作为特征 $\boldsymbol{x}$:{阳性，阴性}，正常人和癌症患者化验结果为阳性的概率分别为 $p(\boldsymbol{x}=\text{阳性}\,|\,\omega_1)=0.2$ 和 $p(\boldsymbol{x}=\text{阳性}\,|\,\omega_2)=0.4$。已知判别代价分别为 $\lambda_{11}=0$，$\lambda_{22}=0$，$\lambda_{12}=6$，$\lambda_{21}=1$，现有一人化验结果为阳性，问此人是否患癌症[2]？

解：根据例 2.1 的计算结果可知后验概率为

$$p(\omega_1\,|\,\boldsymbol{x})=0.818,\quad p(\omega_2\,|\,\boldsymbol{x})=0.182$$

再按式(2.20)计算条件风险

$$R(a_1\,|\,\boldsymbol{x})=\sum_{j=1}^{2}\lambda_{1j}\,p(\omega_j\,|\,\boldsymbol{x})=\lambda_{12}\,p(\omega_2\,|\,\boldsymbol{x})\ =1.092$$

$$R(a_2|\boldsymbol{x})=\lambda_{21}\,p(\omega_1\,|\,\boldsymbol{x})=0.818$$

由于 $R(a_1\,|\,\boldsymbol{x})>R(a_2\,|\,\boldsymbol{x})$，即决策为 ω_2 的条件风险小于决策为 ω_1 的条件风险，因此我们采取决策行动 a_2，即判断待识别特征 $\boldsymbol{x}$ 为 ω_2 类——癌症患者。

在本例中，首先根据先验概率和条件概率计算后验概率，然后根据决策表计算条件风险，基于最小风险的贝叶斯决策采取使条件风险最小的决策行动，因为 $R(a_1\,|\,\boldsymbol{x})>R(a_2\,|\,\boldsymbol{x})$，所以采取决策 a_2。本例的判断结果与例2.1正好相反，这是因为影响决策结果的因素又多了一个，即“损失”，而且两类错误决策所造成的损失相差悬殊，因此“损失”起了主导作用。

2.2.2　最小最大决策

从最小错误率和最小风险贝叶斯决策的介绍中可以看出，其最后做出的决策都与先验概率 $p(\omega_i)$ 有关。如果对于给定的 $\boldsymbol{x}$，其 $p(\omega_i)$ 不变，按照贝叶斯决策规则，可以使错误率或风险最小。但如果 $p(\omega_i)$ 改变或者事先对先验概率一无所知，再按照固定 $p(\omega_i)$ 条件下的决策规则进行决策，就不可能得到最小风险或最小错误率。本节所要介绍的最小最大决策就是在考虑 $p(\omega_i)$ 变化的情况下，如何使最大可能的风险最小，也就是在最差的条件下争取最好的结果[5]。

在最小风险决策中，$R(a_i\,|\,\boldsymbol{x})$ 仅反映单个样本 $\boldsymbol{x}$ 的决策风险，不能反映整个模式空间划分成某种类别空间时的总风险。对于多种划分 R_i 来说，能从总体上考查哪一种划分最优，这就是所要研究的最小最大决策，其实质是即使 $p(\omega_i)$ 变化，由此决策带来的最大可能的风险也最小。

根据式(2.21)，总风险 R 为

$$R=\int_R R(a(\boldsymbol{x})\,|\,\boldsymbol{x})\,p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

现在来分析总风险 R 与先验概率 $p(\omega_i)$ 之间的关系。在两类情况下有

$$R(a_1\,|\,\boldsymbol{x})=\lambda_{11}p(\omega_1\,|\,\boldsymbol{x})+\lambda_{12}p(\omega_2\,|\,\boldsymbol{x})$$

$$R(a_2\,|\,\boldsymbol{x})=\lambda_{22}p(\omega_2\,|\,\boldsymbol{x})+\lambda_{21}p(\omega_1\,|\,\boldsymbol{x})\tag{2.26}$$

所以

$$
\begin{aligned}
R &= \int_{R_1} R(a_1 \mid \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \int_{R_2} R(a_2 \mid \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{R_1} [\lambda_{11} p(\boldsymbol{x} \mid \omega_1) p(\omega_1) + \lambda_{12} p(\boldsymbol{x} \mid \omega_2) p(\omega_2)] \mathrm{d}\boldsymbol{x} + \\
&\quad \int_{R_2} [\lambda_{21} p(\boldsymbol{x} \mid \omega_1) p(\omega_1) + \lambda_{22} p(\boldsymbol{x} \mid \omega_2) p(\omega_2)] \mathrm{d}\boldsymbol{x} \\
&= \lambda_{11} p(\omega_1) \int_{R_1} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x} + \lambda_{12} p(\omega_2) \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} + \\
&\quad \lambda_{21} p(\omega_1) \int_{R_2} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x} + \lambda_{22} p(\omega_2) \int_{R_2} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x}
\end{aligned} \tag{2.27}
$$

对于两类问题，满足

$$
\begin{aligned}
p(\omega_1) + p(\omega_2) &= 1 \\
\int_{R_2} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} &= 1 - \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} \\
\int_{R_1} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x} &= 1 - \int_{R_2} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x}
\end{aligned} \tag{2.28}
$$

将式(2.28)代入式(2.27)，整理得

$$
\begin{aligned}
R &= \lambda_{22} + [\lambda_{12} - \lambda_{22}] \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} + \{[\lambda_{11} - \lambda_{22}] + [\lambda_{21} - \lambda_{11}] \int_{R_2} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x} - \\
&\quad [\lambda_{12} - \lambda_{22}] \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x}\} \times p(\omega_1) \\
&= A + B \times p(\omega_1)
\end{aligned} \tag{2.29}
$$

其中，

$$
A = \lambda_{22} + [\lambda_{12} - \lambda_{22}] \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} \tag{2.30}
$$

$$
B = [\lambda_{11} - \lambda_{22}] + [\lambda_{21} - \lambda_{11}] \int_{R_2} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x} - [\lambda_{12} - \lambda_{22}] \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} \tag{2.31}
$$

上式表明，一旦确定了 R_i，则 A，B 为常数，总风险 R 与先验概率 $p(\omega_1)$ 呈线性关系。因为 0 ≤ $p(\omega_1)$ ≤1，R 值的变化范围为 $A \sim A + B$。所以可以对 $p(\omega_1)$ 取若干不同的值，分别按照最小风险贝叶斯决策方法确定其相应的两类区域，从而计算出相应的最小风险 R，就可以得到最小风险 R 与先验概率 $p(\omega_1)$ 的关系曲线，如图2.2 所示[3]。曲线上的 A 点纵坐标 R_a^* 是对应于先验概率 $p_a^*(\omega_1)$ 时的最小风险，而过 A 点的直线 CD 则是对应于式(2.29)的直线，直线上点的纵坐标则是对应于 $p(\omega_1)$ 变化时的风险值。

由式(2.29)可知，$\partial R / \partial p(\omega_1) = B$，如果 $B = 0$，则 $R = A$。在图2.2(b)中 B 点的横坐标 $p_b^*(\omega_1)$ 对应于决策方案，使系数 B = 0，纵坐标对应于其贝叶斯风险，过 B 点的切线 $C'D'$ 与横轴平行，即此时 $R = A$ 表示的直线与曲线相切，且平行于 $p(\omega_1)$ 坐标轴。不管 $p(\omega_1)$ 如何变化，其风险都不再变化，其最大风险等于 A，这时最大可能风险达到最小值。所以最小最大决策为

$$
(\lambda_{11} - \lambda_{22}) + (\lambda_{21} - \lambda_{11}) \int_{R_2} p(\boldsymbol{x} \mid \omega_1) \mathrm{d}\boldsymbol{x} - (\lambda_{12} - \lambda_{22}) \int_{R_1} p(\boldsymbol{x} \mid \omega_2) \mathrm{d}\boldsymbol{x} = 0 \tag{2.32}
$$

由此来确定 R_1 和 R_2。

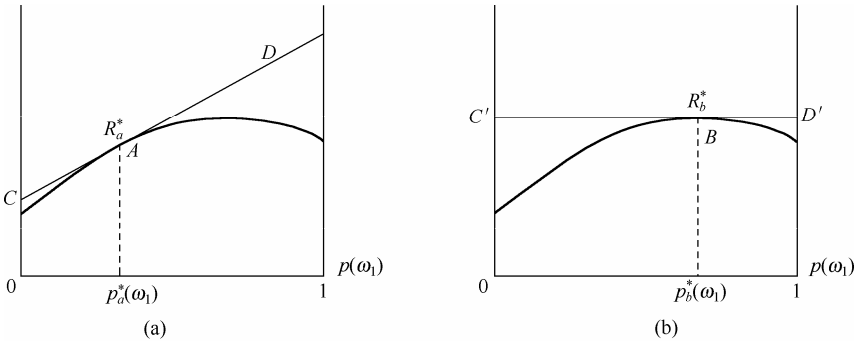


图 2.2　最小最大决策

因此，最小最大决策的任务就是寻找使贝叶斯风险为最大时的决策域 R_2 和 R_1，它对应于方程(2.32)的解。在求出使贝叶斯风险最大时的决策域 R_2 和 R_1 以及相对应的先验概率 $p_b^*(\omega_1)$ 后，最小最大决策就和最小风险贝叶斯决策规则相似。

最小最大决策在博弈论(game theory)中的作用比较大。在博弈论中，你会有一个对手以对你最不利的方式与你竞争。因此，对于你来说，如何采取一种行为(例如做出一种分类)使你所付出的代价(由你对手的对策行为所产生的)最小，具有十分重要的意义。

2.3　正态分布模式的统计决策

在前面论述的贝叶斯决策理论中，先验概率和条件概率密度函数都很重要。通常，先验概率的估计并不困难，所以贝叶斯决策主要取决于条件概率密度函数 $p(\boldsymbol{x}\,|\,\omega_i)$。在所研究的概率密度函数中最引人注意的是正态分布密度函数，原因如下。

(1)在自然现象和社会现象中，大量的随机变量都服从或近似地服从正态分布。

(2)即使统计总体不服从正态分布，但是其许多重要的样本特征可能是渐近正态分布的。

(3)正态分布在数学上分析起来比较方便。

因此，本节着重介绍正态分布的定义和性质，并讨论正态分布概率模型下的贝叶斯决策判别函数。

2.3.1　正态分布概率密度函数的定义及性质

(1)单变量正态分布

单变量正态分布概率密度函数定义为

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{2.33}$$

式中，μ 为随机变量 x 的期望，σ^2 为 x 的方差，σ 也称为标准差，并且

$$\mu=E(x)=\int_{-\infty}^{\infty}xp(x)\mathrm{d}x$$

$$\sigma^2=E[(x-\mu)^2]=\int_{-\infty}^{\infty}(x-\mu)^2p(x)\mathrm{d}x \tag{2.34}$$

由概率知识可以知道，概率密度函数应满足下列关系式：

$$p(x)\geq 0,\ -\infty<x<+\infty \tag{2.35}$$

$$\int_{-\infty}^{\infty}p(x)\mathrm{d}x=1 \tag{2.36}$$

单变量正态分布概率密度函数 $p(x)$由两个参数 μ 和 σ^2 就可以完全确定。为了简化起见，我们常记为 $p(x)\sim N(\mu,\sigma^2)$，它表示 x 是均值为 μ、方差为 σ^2 的正态分布的随机变量。正态分布的样本主要集中在均值附近，其分散程度可以用标准差来表征，σ 愈大分散程度就愈大。正态分布概率密度函数 $p(x)$ 如图2.3[2]所示。从正态分布的总体中抽取样本，约有 95%的样本落在区间 $|x-\mu|\leq 2\sigma$ 中。

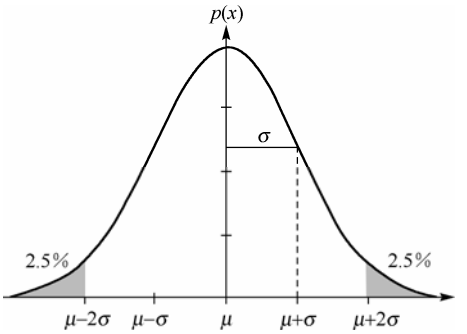


图 2.3 正态分布概率密度函数

(2) 多元正态分布

(a) 多元正态分布的概率密度函数。多元正态分布的概率密度函数定义为

$$p(\boldsymbol{x})=\frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{2.37}$$

其中，$\boldsymbol{x}$ 是 n 维列向量；$\boldsymbol{\mu}$ 是 n 维均值向量；$\boldsymbol{\Sigma}$ 是 $n\times n$ 维协方差矩阵；$(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}$ 是 $(\boldsymbol{x}-\boldsymbol{\mu})$ 的转置；$\boldsymbol{\Sigma}^{-1}$ 是 $\boldsymbol{\Sigma}$ 的逆矩阵；$|\boldsymbol{\Sigma}|$ 是 $\boldsymbol{\Sigma}$ 的行列式。$\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 分别是向量 $\boldsymbol{x}$ 和矩阵 $(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}$ 的期望，即

$$\boldsymbol{\mu}=E\{\boldsymbol{x}\}=(\mu_1,\mu_2,\cdots,\mu_n)^{\mathrm{T}} \tag{2.38}$$

$$\boldsymbol{\Sigma}=E\{(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\}=[\sigma_{ij}^2]_{n\times n} \tag{2.39}$$

更具体地说，若 x_i 是 $\boldsymbol{x}$ 的第 i 个元素，μ_i 是 $\boldsymbol{\mu}$ 的第 i 个元素，σ_{ij}^2 是 $\boldsymbol{\Sigma}$ 的第 i 行、第 j 列元素，则

$$\mu_i=E\{x_i\}=\int_{-\infty}^{\infty}x_i p(x_i)\mathrm{d}x_i \tag{2.40}$$

其中 $p(x_i)$为边缘分布：

$$p(x_i)=\int_{-\infty}^{\infty}\mathrm{L}\int_{-\infty}^{\infty}p(\boldsymbol{x})\mathrm{d}x_1\mathrm{d}x_2\mathrm{L}\ \mathrm{d}x_{i-1}\mathrm{d}x_i\mathrm{L}\ \mathrm{d}x_n \tag{2.41}$$

而

$$\begin{aligned}\sigma_{ij}^2&=E[(x_i-\mu_i)(x_j-\mu_j)^{\mathrm{T}}]\\&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x_i-\mu_i)(x_j-\mu_j)^{\mathrm{T}}p(x_i,x_j)\mathrm{d}x_i\mathrm{d}x_j\end{aligned} \tag{2.42}$$

不难证明，协方差矩阵 $\boldsymbol{\Sigma}$ 是对称半正定的，我们只考虑正定的情况。这时

$$\boldsymbol{\Sigma}=\begin{bmatrix}\sigma_{11}^2 & \sigma_{12}^2 & \mathrm{L} & \sigma_{1n}^2\\ \sigma_{12}^2 & \sigma_{22}^2 & \mathrm{L} & \sigma_{2n}^2\\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M}\\ \sigma_{1n}^2 & \sigma_{2n}^2 & \mathrm{L} & \sigma_{nn}^2\end{bmatrix} \tag{2.43}$$

$\boldsymbol{\Sigma}$ 的行列式$|\boldsymbol{\Sigma}|>0$，x_i 的方差是主对角线元素 σ_{ii}^2。非主对角线上的元素 σ_{ij}^2 是 x_i 和 x_j 的协方差。

(b) 多元正态分布的性质。多元正态分布有不少易于分析的性质，这里仅介绍几个与我们关系密切的性质。

① 参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 对分布的确定性。

多元正态分布由均值向量 $\boldsymbol{\mu}$ 和协方差矩阵 $\boldsymbol{\Sigma}$ 完全确定。由式(2.38)和式(2.39)可见，均值向量 $\boldsymbol{\mu}$ 由 n 个分量组成，协方差矩阵 $\boldsymbol{\Sigma}$ 由于其对称性，故其独立元素只有 $n(n+1)/2$ 个，所以，多元正态分布是由 $n+n(n+1)/2$ 个参数来完全确定的。为简单起见，多元正态分布概率密度函数常记为 $p(\boldsymbol{x})\sim N(\boldsymbol{\mu},\boldsymbol{\Sigma})$。

② 等密度点的轨迹为超椭球面。

从正态分布总体中抽取的样本大部分落在由 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 确定的一个区域里，如图2.4所示[3]。这个区域的中心由均值向量 $\boldsymbol{\mu}$ 决定，区域的大小由协方差矩阵 $\boldsymbol{\Sigma}$ 决定。从多元正态概率密度函数表达式(2.37)可以看出，当指数项为常数时，概率密度函数 $p(\boldsymbol{x})$ 的值不变，因此等密度点应是使式(2.37)中的指数项为常数的点，即应满足

$$(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})=\text{常数} \tag{2.44}$$

可以证明式(2.44)的解是一个超椭球面，且它的主轴方向由 $\boldsymbol{\Sigma}$ 矩阵的特征向量决定，长度与相应的协方差矩阵 $\boldsymbol{\Sigma}$ 的本征值成正比。

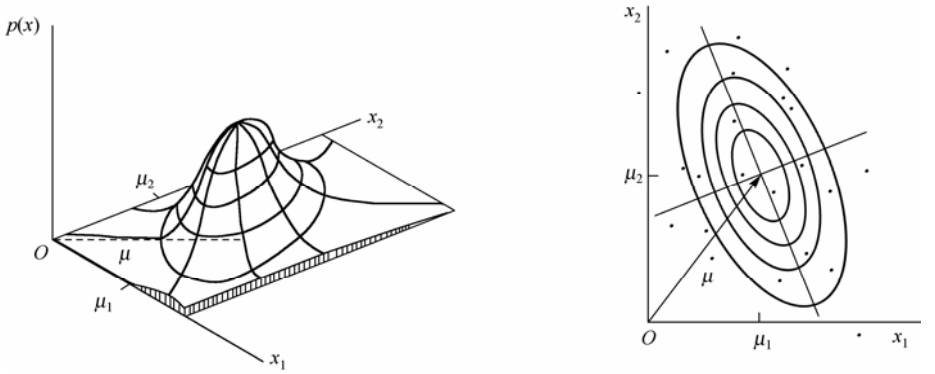


图 2.4　正态分布的等密度点的轨迹为超椭球面

在数理统计中，式(2.44)所表示的数量

$$\gamma^2=(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{2.45}$$

称为 $\boldsymbol{x}$ 到 $\boldsymbol{\mu}$ 的 Mahalanobis 距离的平方[5]。所以等密度点轨迹是 $\boldsymbol{x}$ 到 $\boldsymbol{\mu}$ 的 Mahalanobis 距离为常数的超椭球面。这个超椭球体大小是样本对于均值向量的离散度度量。可以看出，对应于 Mahalanobis 距离为 γ 的超椭球体积是

$$V=V_d\,|\boldsymbol{\Sigma}|^{\frac{1}{2}}\,\gamma^d \tag{2.46}$$

其中 V_d 是 d 维单位超球体的体积，

$$V_d=\begin{cases}\dfrac{\pi^{d/2}}{\left(\dfrac{d}{2}\right)!}, & d\text{ 为偶数}\\ \dfrac{2^d\pi^{(d-1)/2}\left(\dfrac{d-1}{2}\right)!}{d!}, & d\text{ 为奇数}\end{cases} \tag{2.47}$$

所以，对于给定的维数，样本离散度随$|\boldsymbol{\Sigma}|^{\frac{1}{2}}$的变化而改变。

③不相关性等价于独立性。

在数理统计中，一般来说若两个随机变量x_i和x_j之间不相关，并不意味着它们之间一定独立。下面给出不相关与独立的定义。

若

$$E\{x_i\times x_j\}=E\{x_i\}E\{x_j\} \tag{2.48}$$

则定义随机变量x_i和x_j是不相关的。

若

$$p(x_i,x_j)=p(x_i)p(x_j) \tag{2.49}$$

则定义随机变量x_i和x_j是独立的。

从定义可以看出独立性是比不相关性更强的条件，独立性要求式(2.49)对于所有的x_i和x_j都成立，而不相关性说的是两个随机变量的积的期望等于两个随机变量的期望的积，它反映了x_i和x_j总体的性质。若x_i和x_j相互独立，则它们之间一定不相关；反之则不一定成立。

对多元正态分布的任意两个分量x_i和x_j，若x_i和x_j互不相关，可以证明它们之间一定独立。这就是说在正态分布下不相关性等价于独立性，证明参见文献[3]。

④ 边缘分布和条件分布的正态性[3]。

多元正态分布的边缘分布和条件分布仍然是正态分布。我们只以二元情况为例来证明这一点，一般情况与此相似，只是情况比较复杂。

二元正态分布协方差矩阵$\boldsymbol{\Sigma}$及其逆矩阵$\boldsymbol{\Sigma}^{-1}$为

$$\boldsymbol{\Sigma}=\begin{bmatrix}\sigma_{11}^2 & \sigma_{12}^2\\ \sigma_{12}^2 & \sigma_{22}^2\end{bmatrix},\qquad \boldsymbol{\Sigma}^{-1}=\frac{1}{|\boldsymbol{\Sigma}|}\begin{bmatrix}\sigma_{22}^2 & -\sigma_{12}^2\\ -\sigma_{12}^2 & \sigma_{11}^2\end{bmatrix} \tag{2.50}$$

根据边缘分布定义

$$\begin{aligned}p(x_1)&=\int_{-\infty}^{\infty}p(x_1,x_2)\mathrm{d}x_2\\ &=\frac{1}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\int_{-\infty}^{\infty}\exp\left\{-\frac{1}{2|\boldsymbol{\Sigma}|}\left[\sigma_{22}^2(x_1-\mu_1)^2+\sigma_{11}^2(x_2-\mu_2)^2-2\sigma_{12}^2(x_1-\mu_1)(x_2-\mu_2)\right]\right\}\mathrm{d}x_2\end{aligned}$$

$$=\frac{1}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\int_{-\infty}^{\infty}\exp\left\{-\frac{\sigma_{11}^2}{2|\boldsymbol{\Sigma}|}\left[(x_2-\mu_2)^2+\frac{\sigma_{22}^2}{\sigma_{11}^2}(x_1-\mu_1)^2-2\frac{\sigma_{12}^2}{\sigma_{11}^2}(x_1-\mu_1)(x_2-\mu_2)\right]\right\}\mathrm{d}x_2$$

$$=\frac{1}{(2\pi)^{\frac{1}{2}}\sigma_{11}}\exp\left\{-\frac{1}{2}\left(\frac{x_1-\mu_1}{\sigma_{11}}\right)^2\right\}\times\frac{\sigma_{11}}{(2\pi)^{\frac{1}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\int_{-\infty}^{\infty}\exp\left\{\frac{-\sigma_{11}^2}{2|\boldsymbol{\Sigma}|}\left[(x_2-\mu_2)-\frac{\sigma_{12}^2}{\sigma_{11}^2}(x_1-\mu_1)\right]^2\right\}\mathrm{d}x_2$$

$$=\frac{1}{(2\pi)^{\frac{1}{2}}\sigma_{11}}\exp\left\{-\frac{1}{2}\left(\frac{x_1-\mu_1}{\sigma_{11}}\right)^2\right\} \tag{2.51}$$

其中 $|\boldsymbol{\Sigma}|=\sigma_{11}^2\sigma_{22}^2-\sigma_{12}^4$，所以 x_1 的边缘分布为

$$p(x_1)\sim N(\mu_1,\sigma_{11}^2) \tag{2.52}$$

也就是说，边缘分布 $p(x_1)$服从以均值为 μ_1、方差为 σ_{11}^2 的正态分布。同理可以推出 x_2 的边缘分布为 $p(x_2)\sim N(\mu_2,\sigma_{22}^2)$。

对于给定 x_1 条件下 x_2 的分布，有定义

$$p(x_2\mid x_1)=\frac{p(x_1,x_2)}{p(x_1)} \tag{2.53}$$

根据边缘分布的推导过程可以写出

$$p(x_2\mid x_1)=\frac{\sigma_{11}}{(2\pi)^{\frac{1}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left\{-\frac{\sigma_{11}^2}{2|\boldsymbol{\Sigma}|}\left[(x_2-\mu_2)-\frac{\sigma_{12}^2}{\sigma_{11}^2}(x_1-\mu_1)\right]^2\right\}$$

$$=K\exp\left\{-\frac{\sigma_{11}^2}{2|\boldsymbol{\Sigma}|}\left[x_2-\left(\mu_2+\frac{\sigma_{12}^2}{\sigma_{11}^2}(x_1-\mu_1)\right)\right]^2\right\} \tag{2.54}$$

其中 $K=\dfrac{\sigma_{11}}{(2\pi)^{\frac{1}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}$。

同理可以写出给定 x_2 条件下 x_1 的分布为

$$p(x_1\mid x_2)=\frac{\sigma_{22}}{(2\pi)^{\frac{1}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left\{-\frac{\sigma_{22}^2}{2|\boldsymbol{\Sigma}|}\left[(x_1-\mu_1)-\frac{\sigma_{12}^2}{\sigma_{22}^2}(x_2-\mu_2)\right]^2\right\} \tag{2.55}$$

可见，$p(x_2\mid x_1)$ 和 $p(x_1\mid x_2)$ 也服从正态分布。

⑤ 线性变换的正态性。

多元正态随机向量的线性变换仍为多元正态分布的随机向量。设

$$\boldsymbol{x}=\left(x_1,x_2,\cdots,x_n\right)^{\mathrm{T}}$$

是均值向量为 $\boldsymbol{\mu}$、正定协方差矩阵为 $\boldsymbol{\Sigma}$ 的正态随机向量。若对 $\boldsymbol{x}$ 做线性变换，即

$$\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x} \tag{2.56}$$

其中 $\boldsymbol{A}$ 是线性变换矩阵，且是非奇异阵，则 $\boldsymbol{y}$ 服从以均值向量为 $\boldsymbol{A\mu}$ 、协方差矩阵为 $\boldsymbol{A\Sigma A}^{\mathrm{T}}$ 的多元正态分布，即

$$p(\boldsymbol{y}) \sim N(\boldsymbol{A\mu}, \boldsymbol{A\Sigma A}^{\mathrm{T}}) \tag{2.57}$$

从线性变换的正态性可以看出，用非奇异阵 $\boldsymbol{A}$ 对 $\boldsymbol{x}$ 做线性变换后，原来的正态分布正好变成另一参数不同的正态分布。由于 $\boldsymbol{\Sigma}$ 是对称阵，根据线性代数知识总可以找到某个 $\boldsymbol{A}$，使得变换后 $\boldsymbol{y}$ 的协方差阵 $\boldsymbol{A\Sigma A}^{\mathrm{T}}$ 为对角阵。这就意味着 $\boldsymbol{y}$ 的各个分量间是相互独立的。

⑥ 线性组合的正态性。

若 $\boldsymbol{x}$ 为多元正态随机向量，则线性组合 $\boldsymbol{y} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}$ 是一维的正态随机变量，

$$p(\boldsymbol{y}) \sim N(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\mu}, \boldsymbol{a}^{\mathrm{T}}\boldsymbol{\Sigma a}) \tag{2.58}$$

其中 $\boldsymbol{a}$ 是与 $\boldsymbol{x}$ 同维的向量。

这一性质的证明是不难的，只要利用性质⑤做线性变换 $\boldsymbol{y} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}$，则 $p(\boldsymbol{y}) \sim N(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\Sigma A})$，其中 $\boldsymbol{A} = [\boldsymbol{a}, \boldsymbol{A}_1]$ 为非奇异阵，$\boldsymbol{A}_1$ 为 $n \times (n-1)$ 维矩阵，$\boldsymbol{y} = \begin{bmatrix} y \\ \boldsymbol{Y}_1 \end{bmatrix}$，这时可写为

$$\boldsymbol{y} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}^{\mathrm{T}} \\ \boldsymbol{A}_1^{\mathrm{T}} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \\ \boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{x} \end{bmatrix}$$

根据性质⑤，$\boldsymbol{y}$ 是服从均值向量为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu}$ 。协方差阵为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\Sigma A}$ 的多元正态分布的随机向量。

又根据性质④，$\boldsymbol{y}$ 的边缘分布的正态性，可以得出 $\boldsymbol{y} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}$ 服从正态分布，其概率密度函数为 $p(\boldsymbol{y}) \sim N(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\mu}, \boldsymbol{a}^{\mathrm{T}}\boldsymbol{\Sigma a})$ 。

2.3.2 多元正态概率模型的贝叶斯判别函数

根据 2.1 节中给出的最小错误率贝叶斯判别函数 $d_i(\boldsymbol{x}) = p(\omega_i)p(\boldsymbol{x} \mid \omega_i)$，即式(2.18)，为了便于分析取对数形式，则有

$$g_i(\boldsymbol{x}) = \ln d_i(\boldsymbol{x}) = \ln p(\omega_i) + \ln[p(\boldsymbol{x} \mid \omega_i)] \tag{2.59}$$

由于在多元正态概率模型 $p(\boldsymbol{x} \mid \omega_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$，$i = 1, \cdots, c$ 下，$\boldsymbol{x}$ 的类概率密度函数为

$$p(\boldsymbol{x} \mid \omega_i) = \frac{1}{(2\pi)^{n/2} \mid \boldsymbol{\Sigma}_i \mid^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\} \tag{2.60}$$

将式(2.60)代入式(2.59)得

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln \mid \boldsymbol{\Sigma}_i \mid + \ln[p(\omega_i)] \tag{2.61}$$

式(2.61)即对应于多元正态分布模型的贝叶斯判别函数[5]。由式(2.17)，可知分割 ω_i 类和 ω_j 类的决策面为 $d_i(\boldsymbol{x}) = d_j(\boldsymbol{x})$，即

$$-\frac{1}{2}[(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) - (\boldsymbol{x} - \boldsymbol{\mu}_j)^{\mathrm{T}} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)] - \frac{1}{2}\ln\frac{\mid \boldsymbol{\Sigma}_i \mid}{\mid \boldsymbol{\Sigma}_j \mid} + \ln\frac{p(\omega_i)}{p(\omega_j)} = 0 \tag{2.62}$$

为了进一步理解多元正态分布下的判别函数和决策面，下面对一些特殊情况进行讨论。

(1) 第一种情况：$\boldsymbol{\Sigma}_i = \sigma^2 \boldsymbol{I}$，$i = 1, 2, \cdots, c$

这种情况中每类的协方差矩阵都相等，而且类内各个特征相互独立，具有相等的方差 σ^2 。下面再分两种情况讨论。

(a) 先验概率 $p(\omega_i)$ 与 $p(\omega_j)$ 不相等。

此时各类的协方差矩阵为

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma^2 & \mathrm{L} & 0 \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ 0 & \mathrm{L} & \sigma^2 \end{bmatrix}$$

从几何上看，相当于各类样本落入以 $\boldsymbol{\mu}_i$ 为中心的同样大小的一些超球体内。由于

$$\begin{aligned} |\boldsymbol{\Sigma}_i| &= \sigma^{2n} \\ \boldsymbol{\Sigma}_i^{-1} &= \frac{1}{\sigma^2}\boldsymbol{I} \end{aligned} \tag{2.63}$$

将式(2.63)代入式(2.61)，得到其判别函数

$$g_i(\boldsymbol{x}) = -\frac{(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{\mu}_i)}{2\sigma^2} - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln\sigma^{2n} + \ln p(\omega_i) \tag{2.64}$$

由于上式中的第二项、第三项与类别 ω_i 无关，在判决函数中是常量，所以可忽略，于是 $g_i(\boldsymbol{x})$ 简化为

$$g_i(\boldsymbol{x}) = -\frac{1}{2\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \ln p(\omega_i) \tag{2.65}$$

其中，

$$(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{\mu}_i) = \|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2 = \sum_{j=1}^{n}(x_j-\mu_{ij})^2, i = 1,\cdots,c \tag{2.66}$$

为 $\boldsymbol{x}$ 到 ω_i 类的中心(均值向量 $\boldsymbol{\mu}_i$)的欧氏距离的平方。

判别函数 $g_i(\boldsymbol{x})$ 还可以进一步简化。由式(2.65)可知判别函数是 $\boldsymbol{x}$ 的二次函数，但是 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ 与类别 ω_i 无关，也可以省略，则有

$$g_i(\boldsymbol{x}) = -\frac{1}{2\sigma^2}(-2\boldsymbol{\mu}_i^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{\mu}_i^{\mathrm{T}}\boldsymbol{\mu}_i) + \ln p(\omega_i) = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} + \omega_{i0} \tag{2.67}$$

其中，

$$\begin{aligned} \boldsymbol{w}_i &= \frac{1}{\sigma^2}\boldsymbol{\mu}_i \\ \omega_{i0} &= -\frac{1}{2\sigma^2}\boldsymbol{\mu}_i^{\mathrm{T}}\boldsymbol{\mu}_i + \ln p(\omega_i) \end{aligned} \tag{2.68}$$

决策规则要求对某个待分类的 $\boldsymbol{x}$，分别计算 $g_i(\boldsymbol{x})$，$i = 1,\mathrm{L}\ ,c$，若

$$g_k(x) = \max_i g_i(\boldsymbol{x})\text{，则有 } \boldsymbol{x} \in \omega_k \tag{2.69}$$

由式(2.67)可以看出，判别函数 $g_i(\boldsymbol{x})$ 是 $\boldsymbol{x}$ 的线性函数，分割 ω_i 类与 ω_j 类的决策面是由线性方程 $g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = 0$ 所确定的一个超平面(如果决策域 R_i 与 R_j 相毗邻)。

在 $\boldsymbol{\Sigma}_i = \sigma^2 \boldsymbol{I}$ 的特殊情况下，这个方程可改写为

$$\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_0) = 0 \tag{2.70}$$

其中，

$$\boldsymbol{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$$

$$\boldsymbol{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \frac{\sigma^2}{\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|^2} \ln \frac{p(\omega_i)}{p(\omega_j)} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

满足式(2.70)的 $\boldsymbol{x}$ 的轨迹为 ω_i 类与 ω_j 类间的决策面，它是一个超平面。

(b) $p(\omega_i) = p(\omega_j)$ 时的情况。

如 c 类的先验概率 $p(\omega_i)$, $i = 1, \cdots, c$ 都相等，则可忽略式(2.65)中的 $\ln p(\omega_i)$ 项，使最小错误率贝叶斯决策规则表达得相当简单，若对模式 $\boldsymbol{x}$ 进行分类，只要计算 $\boldsymbol{x}$ 到各类中心(均值 $\boldsymbol{\mu}_i$)的欧氏距离的平方 $\left\|\boldsymbol{x} - \boldsymbol{\mu}_i\right\|^2$，然后把 $\boldsymbol{x}$ 归于具有 $\min\limits_{i=1,\mathrm{L},c} \left\|\boldsymbol{x} - \boldsymbol{\mu}_i\right\|^2$ 的类。这种分类规则称为最小距离分类规则。

当 $p(\omega_i) = p(\omega_j)$ 时，式(2.70)表示的超平面通过 $\boldsymbol{\mu}_i$ 与 $\boldsymbol{\mu}_j$ 的连线的中点，并与连线正交。

(2)第二种情况：$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$

这也是一种比较简单的情况，它表示各类的协方差矩阵都相等，从几何上看，相当于各类样本集中于以该类均值 $\boldsymbol{\mu}_i$ 点为中心的同样大小和形状的超椭球内。

由于 $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \cdots = \boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$，即 $\boldsymbol{\Sigma}$ 与 i 无关，则其判别函数式(2.61)可简化为

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) + \ln p(\omega_i) \tag{2.71}$$

若 c 类先验概率 $p(\omega_i)$ 都相等，则判别函数可进一步简化为

$$g_i(\boldsymbol{x}) = \gamma^2 = (\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) \tag{2.72}$$

这时决策规则为：为了对 $\boldsymbol{x}$ 进行分类，计算出 $\boldsymbol{x}$ 到每类的均值点 $\boldsymbol{\mu}_i$ 的 Mahalanobis 距离的平方 γ^2，最后把 $\boldsymbol{x}$ 归于 γ^2 最小的类别。

将式(2.71)展开，忽略与 i 无关的 $\boldsymbol{x}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{x}$ 项，则判别函数可写成下面的形式：

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x} + \omega_{i0} \tag{2.73}$$

其中

$$\begin{aligned} \boldsymbol{w}_i &= \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i \\ \omega_{i0} &= -\frac{1}{2} \boldsymbol{\mu}_i^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i + \ln p(\omega_i) \end{aligned} \tag{2.74}$$

由式(2.73)可见，判别函数 $g_i(\boldsymbol{x})$ 也是 $\boldsymbol{x}$ 的线性判别函数，因此决策面仍是一个超平面。如果决策域 R_1 和 R_2 毗邻，则决策面方程应满足

$$g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = 0$$

即

$$\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_0) = 0 \tag{2.75}$$

其中，

$$\boldsymbol{w}=\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j) \tag{2.76}$$

$$\boldsymbol{x}_0=\frac{1}{2}(\boldsymbol{\mu}_i+\boldsymbol{\mu}_j)-\frac{\ln\dfrac{p(\omega_i)}{p(\omega_j)}}{(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j) \tag{2.77}$$

由式(2.76)可见，$\boldsymbol{w}=\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)$ 通常不在 $\boldsymbol{\mu}_i-\boldsymbol{\mu}_j$ 方向，$(\boldsymbol{x}-\boldsymbol{x}_0)$ 为通过 $\boldsymbol{x}_0$ 点的向量。$\boldsymbol{w}$ 与 $(\boldsymbol{x}-\boldsymbol{x}_0)$ 的点积为零则表示 $(\boldsymbol{x}-\boldsymbol{x}_0)$ 与 $\boldsymbol{w}$ 正交，所以决策面通过 $\boldsymbol{x}_0$ 点，但不与 $\boldsymbol{\mu}_i-\boldsymbol{\mu}_j$ 正交。

若各类的先验概率相等，则式(2.77)可写为

$$\boldsymbol{x}_0=\frac{1}{2}(\boldsymbol{\mu}_i+\boldsymbol{\mu}_j) \tag{2.78}$$

此时 $\boldsymbol{x}_0$ 点为 $\boldsymbol{\mu}_i$ 与 $\boldsymbol{\mu}_j$ 连线的中点，根据前面的讨论，决策面应通过这一点。

若先验概率不相等，$\boldsymbol{x}_0$ 就不在 $\boldsymbol{\mu}_i$ 与 $\boldsymbol{\mu}_j$ 连线的中点上，而是在连线上向先验概率小的均值点偏移。

(3)第三种情况：各类的协方差阵不相等

这是多元正态分布的一般情况，即

$$\boldsymbol{\Sigma}_i\neq\boldsymbol{\Sigma}_j,\quad i,j=1,2,\cdots,c \tag{2.79}$$

此时，判别函数式(2.61)只有第二项 $\frac{n}{2}\ln 2\pi$ 与 i 无关，可以忽略，简化后得

$$\begin{aligned}g_i(\boldsymbol{x})&=-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)-\frac{1}{2}\ln|\boldsymbol{\Sigma}_i|+\ln p(\omega_i)\\&=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_i\boldsymbol{x}+\boldsymbol{w}_{ic}^{\mathrm{T}}\boldsymbol{x}+\omega_{i0}\end{aligned} \tag{2.80}$$

其中，

$$\boldsymbol{w}_i=-\frac{1}{2}\boldsymbol{\Sigma}_i^{-1}\ (n\times n\text{ 矩阵}) \tag{2.81}$$

$$\boldsymbol{w}_{ic}=\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i\ \ (n\text{ 维列向量}) \tag{2.82}$$

$$\omega_{i0}=-\frac{1}{2}\boldsymbol{\mu}_i^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i-\frac{1}{2}\ln|\boldsymbol{\Sigma}_i|+\ln p(\omega_i) \tag{2.83}$$

这时判别函数式(2.80)将 $g_i(\boldsymbol{x})$ 表示为 $\boldsymbol{x}$ 的二次型。若决策域 R_i 与 R_j 毗邻，则决策面应满足

$$g_i(\boldsymbol{x})-g_j(\boldsymbol{x})=0$$

即

$$\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{w}_i-\boldsymbol{w}_j)\boldsymbol{x}+(\boldsymbol{w}_{ic}-\boldsymbol{w}_{jc})^{\mathrm{T}}\boldsymbol{x}+\omega_{i0}-\omega_{j0}=0 \tag{2.84}$$

由式(2.84)所决定的决策面为超二次曲面，随着 $\boldsymbol{\Sigma}_i$，$\boldsymbol{\mu}_i$，$p(\omega_i)$ 的不同而呈现为某种超二次曲面，即超球面、超椭球面、超抛物面、超双曲面或超平面。图2.5[2]给出了在二元正态情况下决策面的形式，在图2.5(a)至图2.5(e)五种形式中，变量 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 是类条件独立的，所以协方差矩阵为对角阵。如果再假定各先验概率相等，那么不同的决策面只是由于方差项的差异而引起的。

在图中以标号 1 和 2 的等概率密度轮廓线来表征相应类别的方差，在图2.5(a)中，$p(\boldsymbol{x}|\omega_2)$ 的方差比 $p(\boldsymbol{x}|\omega_1)$ 的小，因此来自类 ω_2 的样本更加可能在该类的均值附近找到，同时由于圆的对称性，决策面是包围着 $\boldsymbol{\mu}_2$ 的一个圆。若把 $\boldsymbol{x}_2$ 轴伸展，如图2.5(b)所示，此时决策面就伸展为一个椭圆。在图 2.5(c)中两类的密度在 $\boldsymbol{x}_1$ 方向上具有相同的方差，但在 $\boldsymbol{x}_2$ 方向上 $p(\boldsymbol{x}|\omega_1)$ 的方差比 $p(\boldsymbol{x}|\omega_2)$ 的方差大，这时 $\boldsymbol{x}_2$ 值大的样本可能来自类 ω_1，并且决策面为抛物线。若对 $p(\boldsymbol{x}|\omega_2)$ 在 $\boldsymbol{x}_1$ 方向上加大其方差，如图2.5(d)所示，则决策面就变为双曲线。最后图2.5(e)示出了特殊的对称性情况，使决策面由双曲线退化为一对直线[2]。

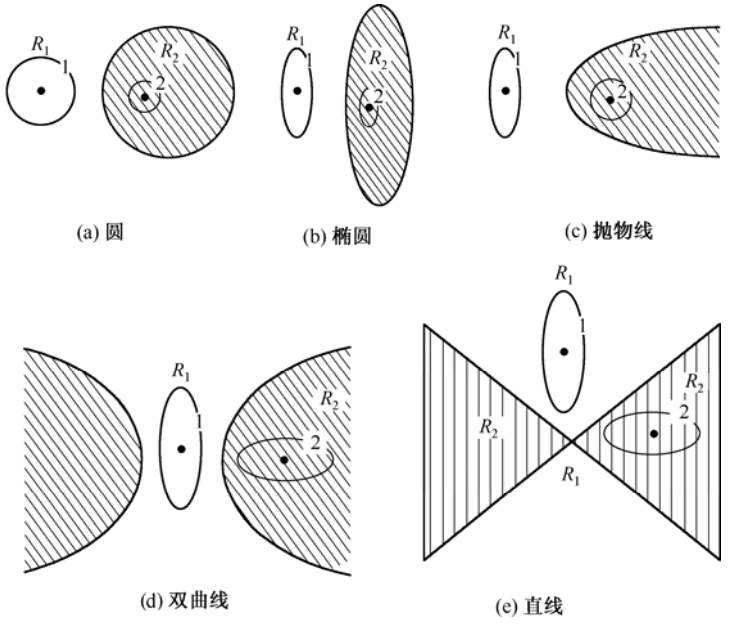


图 2.5 正态分布下的几种决策面形式

2.4 概率密度函数的估计

在前三节中，我们介绍了如何根据先验概率 $p(\omega_i)$ 和类条件概率 $p(\boldsymbol{x}|\omega_i)$ 来进行样本分类。但是，在模式识别的实际应用中，通常得不到有关问题的概率结构的全部知识，往往只有一些模糊而笼统的知识，再加上设计样本，这些样本是待分类模式的一个特定子集。因此，所要解决的问题就是寻找某种有效的方法，利用现有的这些信息进行正确的分类。

我们的解决办法是利用这些训练样本来估计问题中所涉及的先验概率和条件概率密度函数，并把这些估计的结果当做实际的先验概率和条件概率密度，对样本进行分类。在模式识别问题中，估计先验概率通常没有太大的困难，最大的困难在于估计类条件概率密度，包括类条件概率密度函数的形式和参数。其中主要的问题有两个：(1)在很多情况下，已有的训练样本数量总是显得太少；(2)当表示特征的向量 $\boldsymbol{x}$ 的维数较大时，会产生严重的计算复杂度问题(算法的时间、系统的资源开销等)[7]。

当不知道概率密度函数的形式但是能估计出一些参数时，就要采用非参数估计的方法，这种方法也称为总体推断，这类方法有 Parzen 窗法、有限项正交函数级数逼近法、随机逼近法等。但是，如果我们事先已经知道概率密度函数的类型，并且先验知识允许我们能够把条

件概率密度函数进行参数化，那么问题的难度就可以显著地降低。例如我们可以假设 $p(\boldsymbol{x}|\omega_i)$ 是一个多元正态分布，其均值为 $\boldsymbol{\mu}_i$，协方差矩阵为 $\boldsymbol{\Sigma}_i$（这两个参数的具体值是未知的）。这样，我们就把问题从估计完全未知的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 转化为估计参数 $\boldsymbol{\mu}_i$ 和 $\boldsymbol{\Sigma}_i$。解决参数估计问题有两类方法：一类方法是将参数作为非随机量处理，如矩法估计、最大似然估计就属于这类方法；另一类方法是将参数作为随机变量，贝叶斯估计就属于此类方法[6]。

矩法估计只能用于各阶矩的估计，它对于原点矩的估计是无偏的，对中心矩的估计一般不是无偏估计量。其算法简单，宜用于样本容量 N 较大的情况，以保证它的优良性。最大似然估计要求知道总体概率密度函数的具体形式，它也认为被估计量是确定性的，但最大似然估计的适用范围比矩法估计的适用范围更宽一些。在满足某些条件时，最大似然估计 $\hat{\boldsymbol{\theta}}$ 的方差最小，且以概率 1 收敛于真值 $\boldsymbol{\theta}$，最大似然估计 $\hat{\boldsymbol{\theta}}$ 在 $p(\boldsymbol{x},\ \boldsymbol{\theta})$ 满足一定的正则条件下，渐近正态分布，其均值为 $\boldsymbol{\theta}$。最大似然估计的另一个优点是在实际中计算 $p(\boldsymbol{x}|\boldsymbol{\theta})$ 比计算 $p(\boldsymbol{\theta}|\boldsymbol{x})$ 要容易得多，这里不涉及 $\boldsymbol{\theta}$ 的概率密度函数。矩法估计在计算时不需要总体的概率密度类型，而最大似然估计必须运用总体的概率密度类型。虽然最大似然估计次于贝叶斯估计，但由于在某些条件下，最大似然估计能获得方差最小的估计或渐近性很好的估计，因此有着广泛的应用意义。

所以在这里我们介绍最常用的两种方法：最大似然估计和贝叶斯估计。最大似然估计把待估计的参数视为确定性的量，只是其取值未知。最佳估计就是使产生已观测到的样本概率最大的那个值。与此不同的是，贝叶斯估计则把待估计的参数视为符合某种先验概率分布的随机变量。对样本进行观测的过程，就是把先验概率密度转化为后验概率密度的过程，这样就利用样本的信息修正了对参数的初始估计值。在贝叶斯估计中，一个典型的效果就是，每得到新的观测样本，都使得后验概率密度函数变得更加尖锐，使其在待估参数的真实值附近形成最大的尖峰。无论使用哪种参数估计，在参数估计完成后，我们都使用后验概率作为分类准则。

2.4.1　最大似然估计

假定每个样本的类别是已知的，那么就可以把它们按类别分成 c 组 $R_1, R_2, \cdots, R_c$，其中 R_i 中的样本都属于 ω_i 类，而且它们都是独立地根据类条件概率密度函数 $p(\boldsymbol{x}|\omega_i)$ 抽取的，因此我们说每一个样本集 R_i 中的样本都是独立同分布的随机变量。如果能假定 $p(\boldsymbol{x}|\omega_i)$ 的函数形式，并且把它的参数视为未知向量，记为 $\boldsymbol{\theta}_i$，则只要 $\boldsymbol{\theta}_i$ 确定，那么类条件概率密度函数就完全确定了。例如，假设可以认为 $p(\boldsymbol{x}|\omega_i)$ 服从正态分布，即 $p(\boldsymbol{x}|\omega_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$，则参数向量 $\boldsymbol{\theta}_i$ 就由分量 $\boldsymbol{\mu}_i$ 和 $\boldsymbol{\Sigma}_i$ 组成。为了强调 $p(\boldsymbol{x}|\omega_i)$ 依赖于参数向量 $\boldsymbol{\theta}_i$，我们可以把类条件概率密度函数 $p(\boldsymbol{x}|\omega_i)$ 记为 $p(\boldsymbol{x}|\omega_i, \boldsymbol{\theta}_i)$。现在问题就变成利用样本提供的信息来估计参数向量 $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_c$。

为了使问题简化，我们假定在样本集 R_i 中的样本不包含关于 $\boldsymbol{\theta}_j\ (i \neq j)$ 的信息，也就是说，假定不同类的参数是无关的。这样，每个参数向量只对自己类别中的样本起作用，这就允许我们对每一类模式分别进行处理。为了符号的简洁，我们将不在表达式中注出类别标志，也就是说，把整个参数估计问题按模式类分成 c 个独立的问题，每个问题都可以表述成如下格式：已知样本集 $R = \{x_i\}$，$i = 1, 2, \cdots, N$，其中每一个样本都是根据已知形式的概率密度函数 $p(\boldsymbol{x}|\boldsymbol{\theta})$ 独立抽取的，要求使用这些样本，估计未知概率密度函数中的参数向量 $\boldsymbol{\theta}$。

设样本集 R 包含 N 个样本，

$$R = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \mathrm{L}\ \ ,\boldsymbol{x}_N\}$$

由于假定样本是独立抽取的，所以

$$p(R\,|\,\boldsymbol{\theta}) = \prod_{k=1}^{N} p(\boldsymbol{x}_k\,|\,\boldsymbol{\theta}) \tag{2.85}$$

因为现在样本集 R 已知，所以可把 $p(R\,|\,\boldsymbol{\theta})$ 视为 $\boldsymbol{\theta}$ 的函数，也称为关于样本集 R 的似然函数。参数向量 $\boldsymbol{\theta}$ 的最大似然估计，就是求使 $p(R\,|\,\boldsymbol{\theta})$ 达到最大值的那个参数向量 $\hat{\boldsymbol{\theta}}$，或者理解成参数向量 $\boldsymbol{\theta}$ 的最大似然估计就是最符合已知观测样本集的那一个 $\boldsymbol{\theta}$ 值，如图2.6所示，使似然函数 $p(R\,|\,\boldsymbol{\theta})$ 达到最大值的点 $\hat{\boldsymbol{\theta}}$ 就是 $\boldsymbol{\theta}$ 的最大似然估计。

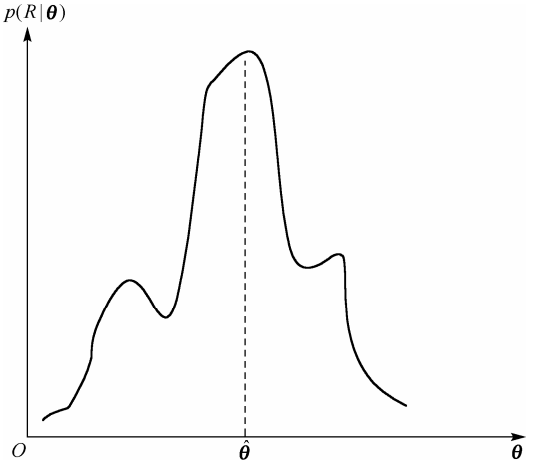


图 2.6　最大似然估计用例

为了分析上的方便，一般使用似然函数的对数函数，因为底数大于 1 的对数函数是单调递增函数，所以使对数似然函数最大的参数向量 $\hat{\boldsymbol{\theta}}$，同样能使似然函数也达到最大值。如果 $p(R\,|\,\boldsymbol{\theta})$ 对 $\boldsymbol{\theta}$ 是可微的，则 $\hat{\boldsymbol{\theta}}$ 可以由典型的求极值方法求出，如果实际的待求参数的个数为 r，则参数向量 $\boldsymbol{\theta}$ 可以写成 r 个分量的列向量

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \mathrm{L}\ \ ,\boldsymbol{\theta}_r)^{\mathrm{T}}$$

用 ∇_θ 表示梯度算子，即

$$\nabla_\theta = \begin{bmatrix} \dfrac{\partial}{\partial \boldsymbol{\theta}_1} \\ \dfrac{\partial}{\partial \boldsymbol{\theta}_2} \\ \mathrm{M} \\ \dfrac{\partial}{\partial \boldsymbol{\theta}_r} \end{bmatrix} \tag{2.86}$$

令 $l(\theta)$ 为似然对数，即

$$l(\boldsymbol{\theta}) = \ln p(R\,|\,\boldsymbol{\theta}) \tag{2.87}$$

则使对数似然函数最大的那个$\boldsymbol{\theta}$值表示为

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) \tag{2.88}$$

结合式(2.85)和式(2.88)，可以得到

$$l(\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\boldsymbol{x}_k \mid \boldsymbol{\theta}) \tag{2.89}$$

$$\nabla_{\boldsymbol{\theta}} l = \sum_{k=1}^{N} \nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{x}_k \mid \boldsymbol{\theta}) \tag{2.90}$$

所以$\boldsymbol{\theta}$的最大似然估计必须满足 r 个方程式组成的方程组

$$\nabla_{\boldsymbol{\theta}} l = 0 \tag{2.91}$$

需要注意的是，我们必须记住得到的$\hat{\boldsymbol{\theta}}$只是对于真实值的一个估计，其对于真实值的接近程度受训练样本个数的制约，训练样本越多，其中的样本就越具有代表性，估计值$\hat{\boldsymbol{\theta}}$也就越接近于真实值[7]。

在正态分布情况下，均值向量$\boldsymbol{\mu}$和协方差矩阵$\boldsymbol{\Sigma}$都是未知的，这些未知参数构成$\boldsymbol{\theta}$的各个分量。现在我们把上面得到的结论应用到训练样本服从多元正态分布的参数估计中。设从均值向量为$\boldsymbol{\mu}$、协方差矩阵为$\boldsymbol{\Sigma}$的正态总体中抽取样本，我们观察以下不同的情况。

(1)首先考虑一种简单的情况，假设$\boldsymbol{\Sigma}$已知，未知的只是均值向量$\boldsymbol{\mu}$，这时有

$$\begin{aligned} \ln p(\boldsymbol{x}_k \mid \boldsymbol{\mu}) &= -\frac{1}{2}\ln[(2\pi)^d |\boldsymbol{\Sigma}|] - \frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}) \\ \nabla_{\boldsymbol{\mu}} \ln p(\boldsymbol{x}_k \mid \boldsymbol{\mu}) &= \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}) \end{aligned} \tag{2.92}$$

这里未知参数是$\boldsymbol{\mu}$，它就是参数向量$\boldsymbol{\theta}$，所以$\boldsymbol{\mu}$的最大似然估计必须满足方程

$$\sum_{k=1}^{N} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}) = 0 \tag{2.93}$$

两边乘以协方差矩阵$\boldsymbol{\Sigma}$，进行整理得

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{x}_k \tag{2.94}$$

这表明总体的未知均值的最大似然估计正是样本的算术平均——样本均值。它的几何解释是：若把 N 个样本视为一群质点，则样本均值便是它们的质心。

(2)考虑一般的一维正态的情况。在实际应用中，更一般的情况是多元正态分布的均值$\boldsymbol{\mu}$和协方差矩阵$\boldsymbol{\Sigma}$都未知，这样，参数向量$\boldsymbol{\theta}$的组成成分为$\boldsymbol{\theta}_1 = \boldsymbol{\mu}$，$\boldsymbol{\theta}_2 = \sigma^2$，从而单个训练样本的对数似然函数为

$$\begin{aligned} \ln p(\boldsymbol{x}_k \mid \boldsymbol{\theta}) &= -\frac{1}{2}\ln 2\pi\boldsymbol{\theta}_2 - \frac{1}{2\boldsymbol{\theta}_2}(\boldsymbol{x}_k - \boldsymbol{\theta}_1)^2 \\ \nabla_{\theta} \ln p(\boldsymbol{x}_k \mid \boldsymbol{\theta}) &= \begin{bmatrix} \dfrac{1}{\boldsymbol{\theta}_2}(\boldsymbol{x}_k - \boldsymbol{\theta}_1) \\ -\dfrac{1}{2\boldsymbol{\theta}_2} + \dfrac{(\boldsymbol{x}_k - \boldsymbol{\theta}_1)^2}{2\boldsymbol{\theta}_2^2} \end{bmatrix} \end{aligned} \tag{2.95}$$

由式(2.95)可知，$\boldsymbol{\theta}_1$ 和 $\boldsymbol{\theta}_2$ 的最大似然估计应满足方程组

$$\sum_{k=1}^{N}\frac{1}{\hat{\boldsymbol{\theta}}_2}(\boldsymbol{x}_k-\hat{\boldsymbol{\theta}}_1)=0$$

$$\sum_{k=1}^{N}\frac{-1}{2\hat{\boldsymbol{\theta}}_2}+\sum_{k=1}^{N}\frac{(\boldsymbol{x}_k-\hat{\boldsymbol{\theta}}_1)^2}{2\hat{\boldsymbol{\theta}}_2^2}=0 \tag{2.96}$$

其中 $\hat{\boldsymbol{\theta}}_1$ 和 $\hat{\boldsymbol{\theta}}_2$ 分别是 $\boldsymbol{\theta}_1$ 和 $\boldsymbol{\theta}_2$ 的最大似然估计。

将 $\hat{\boldsymbol{\mu}}=\hat{\boldsymbol{\theta}}_1$ 和 $\sigma^2=\hat{\boldsymbol{\theta}}_2$ 代入这个方程组，即得均值和方差的最大似然估计

$$\hat{\boldsymbol{\mu}}=\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}_k$$

$$\sigma^2=\frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})^2 \tag{2.97}$$

(3)对于多维正态分布的情况，计算方法完全类似，只是复杂一些而已。对于多元正态分布的均值 $\boldsymbol{\mu}$ 和协方差矩阵 $\boldsymbol{\Sigma}$ 的最大似然估计结果是

$$\hat{\boldsymbol{\mu}}=\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}_k \tag{2.98}$$

$$\hat{\boldsymbol{\Sigma}}=\frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})\,(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})^{\mathrm{T}} \tag{2.99}$$

可以看出，实际均值向量的最大似然估计就是样本均值，而协方差矩阵的最大似然估计是 N 个矩阵 $(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})\,(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})^{\mathrm{T}}$ 的算术平均[8]。由于真实的协方差矩阵是 $(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})\,(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})^{\mathrm{T}}$ 的数学期望，所以这是一个令人满意的结果。要注意的是，协方差矩阵的最大似然估计是有偏的，就是说，$\hat{\boldsymbol{\Sigma}}$ 的期望值不等于 $\boldsymbol{\Sigma}$，$\boldsymbol{\Sigma}$ 的无偏估计应当是

$$C=\frac{1}{N-1}\sum_{k=1}^{N}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})\,(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}})^{\mathrm{T}} \tag{2.100}$$

当 N 比较大时，这两个估计实际上是一致的。

2.4.2 贝叶斯估计

这一节介绍利用我们所掌握的先验知识计算后验概率的贝叶斯估计方法。虽然使用贝叶斯估计方法得到的结果和最大似然估计的结果很相似，但这两种方法在本质上是不同的：在最大似然估计方法中，我们把需要估计的参数向量 $\boldsymbol{\theta}$ 视为一个确定而未知的参数，使获得实际观测样本的概率最大的估计值视为最好的估计值；而在贝叶斯估计方法中，将概率密度函数的参数估计量 $\boldsymbol{\theta}$ 视为随机变量，根据这些估计量统计特性的先验知识，粗略地给出这些估计量的密度函数，再通过训练模式样本集 $R=\{\boldsymbol{x}_i\}$，利用贝叶斯公式进行迭代运算过程把参数 $\boldsymbol{\theta}$ 的初始密度转化为后验概率密度。

设 $R=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$ 为用于估计未知参数 $\boldsymbol{\theta}$ 密度函数的样本，利用贝叶斯定理，可以得到在逐一给定 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$ 之后，$\boldsymbol{\theta}$ 的条件密度函数的迭代公式

$$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N)=\frac{p(\boldsymbol{x}_N\mid \boldsymbol{\theta},\boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_{N-1})p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_{N-1})}{p(\boldsymbol{x}_N\mid \boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_{N-1})} \tag{2.101}$$

式中，对于 $p(\boldsymbol{\theta}|\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N)$ 而言， $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{N-1})$ 是它的先验概率密度。在加入新的样本 $\boldsymbol{x}_N$ 后，得到新的概率密度 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$。 $p(\boldsymbol{\theta})$ 应是最早的先验概率密度。首先给出第一个样本 $\boldsymbol{x}_1$，按照贝叶斯定理计算后验概率密度 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1)$。将 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1)$ 作为下一步计算的先验概率密度，读入样本 $\boldsymbol{x}_2$，计算得到后验概率密度 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2)$，依此类推可以计算得到最后的 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$ 值。

在观察具体的训练样本之前，我们已知的关于参数向量 $\boldsymbol{\theta}$ 的全部知识可以用先验概率密度函数 $p(\boldsymbol{\theta})$ 来体现。概率 $p(\boldsymbol{x}_N\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{N-1})$ 则按下式计算

$$p(\boldsymbol{x}_N\mid \boldsymbol{x}_1,\cdots,\boldsymbol{x}_{N-1})=\int_x p(\boldsymbol{x}_N\mid \boldsymbol{\theta},\boldsymbol{x}_1,\cdots,\boldsymbol{x}_{N-1})p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\cdots,\boldsymbol{x}_{N-1})\mathrm{d}\boldsymbol{\theta} \tag{2.102}$$

它与未知量 $\boldsymbol{\theta}$ 无关，可认为是一个定值。

我们通过单变量正态密度函数的例子来说明这种方法。

设一个模式样本集的类概率密度函数为单变量正态分布 $N(\boldsymbol{\theta},\sigma^2)$，其中 σ^2 已知，均值 $\boldsymbol{\mu}=\boldsymbol{\theta}$ 待求，即

$$p(\boldsymbol{x}\mid \boldsymbol{\theta})=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{x}-\boldsymbol{\theta}}{\sigma}\right]^2\right] \tag{2.103}$$

要求计算后验概率密度 $p(\boldsymbol{\theta}\big|\{\boldsymbol{x}_i\})$ 和最终要求的类条件概率密度函数 $p(\boldsymbol{x}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$。

给定 N 个训练样本 $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，最初的先验概率密度 $p(\boldsymbol{\theta})$ 服从 $N(\boldsymbol{\theta}_0,\sigma_0^2)$，$\boldsymbol{\theta}_0$ 是根据先验知识对 $\boldsymbol{\theta}$ 的推测，其不确定性由 σ_0^2 表示。这里由于均值的估计量是样本的线性函数，而样本 $\boldsymbol{x}$ 服从正态分布，所以 $p(\boldsymbol{\theta})$ 服从正态分布[5]。

由初始条件 $p(\boldsymbol{\theta})\sim N(\boldsymbol{\theta}_0,\sigma_0^2)$和 $p(\boldsymbol{x}_1\mid \boldsymbol{\theta})\sim N(\boldsymbol{\theta}_0,\sigma_0^2)$，根据贝叶斯公式 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1)=ap(\boldsymbol{x}_1\mid \boldsymbol{\theta})$ $p(\boldsymbol{\theta})$，得

$$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1)=a\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{x}_1-\boldsymbol{\theta}}{\sigma}\right]^2\right]\times\frac{1}{\sqrt{2\pi}\sigma_0}\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{\theta}-\boldsymbol{\theta}_0}{\sigma_0}\right]^2\right] \tag{2.104}$$

其中 a 为定值。根据贝叶斯法则

$$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_N)=\frac{p(\boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_N\mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\varphi}p(\boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_N\mid \boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}} \tag{2.105}$$

其中，φ 表示整个模式的样本空间。每一次迭代运算从样本子集中逐一给出一个样本，N 次运算独立地给出 N 个样本，因此有

$$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\cdots,\boldsymbol{x}_N)=a\left\{\prod_{k=1}^{N}p(\boldsymbol{x}_k\mid \boldsymbol{\theta})\right\}p(\boldsymbol{\theta})$$

代入 $p(\boldsymbol{x}_k\mid \boldsymbol{\theta})$ 和 $p(\boldsymbol{\theta})$ 值，得

$$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_N)=a\left\{\prod_{k=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{x}_k-\boldsymbol{\theta}}{\sigma}\right]^2\right]\right\}\times\frac{1}{\sqrt{2\pi}\sigma_0}\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{\theta}-\boldsymbol{\theta}_0}{\sigma_0}\right]^2\right]$$

$$=a'\exp\left[-\frac{1}{2}\left\{\sum_{k=1}^{N}\left[\frac{\boldsymbol{\theta}-\boldsymbol{x}_k}{\sigma}\right]^2+\left[\frac{\boldsymbol{\theta}-\boldsymbol{\theta}_0}{\sigma_0}\right]^2\right\}\right]$$

$$=a''\exp\left[-\frac{1}{2}\left\{\left[\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}\right]\boldsymbol{\theta}^2-2\left[\frac{1}{\sigma^2}\sum_{k=1}^{N}\boldsymbol{x}_k+\frac{\boldsymbol{\theta}_0}{\sigma_0^2}\right]\boldsymbol{\theta}\right\}\right] \tag{2.106}$$

式中与 $\boldsymbol{\theta}$ 无关的因子均并入常数项 a' 和 a''。可见 $p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$ 是 $\boldsymbol{\theta}$ 平方函数的指数函数，仍然是正态密度函数，可将它写成 $N(\boldsymbol{\theta}_N,\sigma_N^2)$，即

$$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\mathrm{L}\ ,\boldsymbol{x}_N)=\frac{1}{\sqrt{2\pi}\sigma_N}\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{\theta}-\boldsymbol{\theta}_N}{\sigma_N}\right]^2\right]$$

$$=a''\exp\left[-\frac{1}{2}\left[\frac{\boldsymbol{\theta}^2}{\sigma_N{}^2}-2\frac{\boldsymbol{\theta}_N\boldsymbol{\theta}}{\sigma_N^2}\right]\right] \tag{2.107}$$

将式(2.107)与式(2.106)比较，得到

$$\frac{1}{\sigma_N^2}=\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}$$

$$\frac{\boldsymbol{\theta}_N}{\sigma_N^2}=\frac{1}{\sigma^2}\sum_{k=1}^{N}\boldsymbol{x}_k+\frac{\boldsymbol{\theta}_0}{\sigma_0^2}=\frac{N}{\sigma^2}\hat{\boldsymbol{\mu}}_N+\frac{\boldsymbol{\theta}_0}{\sigma_0^2}$$

整理得

$$\begin{cases}\boldsymbol{\theta}_N=\dfrac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\hat{\boldsymbol{\mu}}_N+\dfrac{\sigma^2}{N\sigma_0^2+\sigma^2}\boldsymbol{\theta}_0\\ \sigma_N^2=\dfrac{\sigma_0^2\sigma^2}{N\sigma_0^2+\sigma^2}\end{cases} \tag{2.108}$$

即由训练样本集 $\{\boldsymbol{x}_i\}$，$i=1,\cdots,N$，可求得均值 $\boldsymbol{\theta}$ 的后验概率密度 $p(\boldsymbol{\theta}\mid\{\boldsymbol{x}_i\})$ 服从 $N(\boldsymbol{\theta}_N,\sigma_N^2)$。其中 $\boldsymbol{\theta}_N$ 是根据 N 个样本对均值的估计，是先验信息 $(\boldsymbol{\theta}_0,\sigma_0^2,\sigma^2)$ 与训练样本的信息(上式中的 N，$\hat{\boldsymbol{\mu}}_N$)相结合的结果，是利用 N 个训练样本信息对均值先验估计 $\boldsymbol{\theta}_0$ 的补充。σ_N^2 是对这个估计不确定性的度量。由于 σ_N^2 随 N 增加而单调减小，故当 $N\to\infty$ 时，它趋于零。$\boldsymbol{\theta}_N$ 是 $\hat{\boldsymbol{\mu}}_N$ 和 $\boldsymbol{\theta}_0$ 的线性组合，两者的系数非负，其和为 1，故 $\boldsymbol{\theta}_N$ 值在 $\hat{\boldsymbol{\mu}}_N$ 和 $\boldsymbol{\theta}_0$ 之间。只要 $\sigma_0\neq 0$，当 $N\to\infty$ 时，$\boldsymbol{\theta}_N$ 趋于样本均值的估计量 $\hat{\boldsymbol{\mu}}_N$。在正态密度函数均值的估计过程中，每增加一次样本，都减小对 $\boldsymbol{\theta}$ 估计的不确定性，$p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$ 随着样本的增加其曲线愈显“尖锐”，均值与估计量 $\hat{\boldsymbol{\mu}}_N$ 之间的偏差绝对值亦愈来愈小，图2.7[1]就是它的一个例子。

采用上述方法的目的是为了通过 N 个训练样本来估计模式样本的类概率密度函数 $p(\boldsymbol{x}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$。由于

$$p(\boldsymbol{x}\mid \boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)=\int_{\varphi}p(\boldsymbol{x}\mid\boldsymbol{\theta})p(\boldsymbol{\theta}\mid \boldsymbol{x}_1,\cdots,\boldsymbol{x}_N)\mathrm{d}\boldsymbol{\theta}$$

式中，$p(\boldsymbol{x}\mid\boldsymbol{\theta})\sim N(\boldsymbol{\theta},\sigma^2)$，$p(\boldsymbol{\theta}\mid\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)\sim N(\boldsymbol{\theta}_N,\sigma_N^2)$。不难推断，上述两个正态分布密度函数之积对 $\boldsymbol{\theta}$ 的积分结果也是正态密度函数，即

$$p(\boldsymbol{x}\mid\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)\sim N(\boldsymbol{\theta}_N,\sigma_N^2+\sigma^2)$$

在采用训练样本之前。均值 $\boldsymbol{\theta}$ 未知，经过采用 N 个样本进行估计之后，概率密度函数服从 $N(\boldsymbol{\theta}_N,\sigma_N^2+\sigma^2)$，这就得到了均值 $\boldsymbol{\theta}_N$ 值的估计，同时原来的方差 σ^2 修正为 $\sigma_N^2+\sigma^2$。

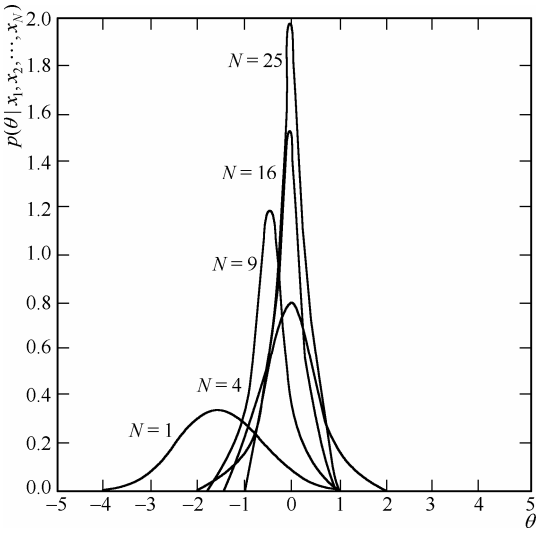


图 2.7　贝叶斯估计举例

2.5　离散情况的贝叶斯决策

上面几节介绍的都是特征向量 $\boldsymbol{x}$ 为连续随机向量情况下的贝叶斯决策理论。但在许多实际的模式识别问题中，特征向量 $\boldsymbol{X}$ 是离散型随机变量，仅可取 c 个离散值 $V_1, V_2, \cdots, V_c$ 中的一个。此时，我们仍然可以利用贝叶斯法则进行计算

$$P(\omega_j\mid\boldsymbol{X})=\frac{P(\boldsymbol{X}\mid\omega_j)P(\omega_j)}{P(\boldsymbol{X})} \tag{2.109}$$

式中

$$P(\boldsymbol{X})=\sum_{i=1}^{c}P(\boldsymbol{X}\mid\omega_i)P(\omega_i) \tag{2.110}$$

可见，贝叶斯决策规则仍然不变，最小错误概率的贝叶斯决策法则仍为

$$\text{如果 } P(\omega_i\mid\boldsymbol{X})>P(\omega_j\mid\boldsymbol{X})\text{，对于一切 } j\neq i \text{ 成立，则决策 } \omega_i \tag{2.111}$$

最小风险的贝叶斯决策法则仍是

$$\text{如果 } R(a_k\mid\boldsymbol{X})=\min_{i=1,2,\cdots,c}R(a_i\mid\boldsymbol{X})\text{，则对应的决策 } a=a_k \tag{2.112}$$

这里条件风险的定义也没有改变。

下面我们以最小错误率的贝叶斯决策规则为例来进行讨论。多数情况下，可以得到以下等价的判别函数

$$d_i(\boldsymbol{X}) = P(\omega_i \mid \boldsymbol{X})$$

$$d_i(\boldsymbol{X}) = P(\boldsymbol{X} \mid \omega_i)P(\omega_i)$$

$$g_i(\boldsymbol{X}) = \ln P(\boldsymbol{X} \mid \omega_i) + \ln P(\omega_i)$$

对于二类模式的分类问题，通常采用下述形式的判别函数

$$d(\boldsymbol{X}) = P(\omega_1 \mid \boldsymbol{X}) - P(\omega_2 \mid \boldsymbol{X})$$

$$d(\boldsymbol{X}) = \ln \frac{P(\boldsymbol{X} \mid \omega_1)}{P(\boldsymbol{X} \mid \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

作为离散型模式分类的例子，我们考虑一个两类模式的分类问题。设特征向量为

$$\boldsymbol{X} = (x_1, x_2, \cdots, x_d)^{\mathrm{T}}$$

它的各个分量都是或为 0 或为 1 的二值特征，并且各个特征相互独立，令

$$p_i = P(x_i = 1 \mid \omega_1)$$

$$q_i = P(x_i = 1 \mid \omega_2)$$

这实际上是对模式的每个特征给出一个“是”或“否”的答案的模型。“是”表示该模式具有对应的特征，其值为 1；“否”表示不具有对应的特征，其值为 0。

因为假定模式中各个特征互相独立，所以可以把条件概率 $P(\boldsymbol{X} \mid \omega_i)$ 写成 $\boldsymbol{X}$ 的分量的概率之积的形式

$$P(\boldsymbol{X} \mid \omega_1) = \prod_{i=1}^{d} p_i^{x_i} (1 - p_i)^{1 - x_i} \tag{2.113}$$

$$P(\boldsymbol{X} \mid \omega_2) = \prod_{i=1}^{d} q_i^{x_i} (1 - q_i)^{1 - x_i} \tag{2.114}$$

从而似然比为

$$\frac{P(\boldsymbol{X} \mid \omega_1)}{P(\boldsymbol{X} \mid \omega_2)} = \prod_{i=1}^{d} \left(\frac{p_i}{q_i}\right)^{x_i} \left(\frac{1 - p_i}{1 - q_i}\right)^{1 - x_i} \tag{2.115}$$

如果采用对数形式的判别函数，则有

$$d(\boldsymbol{X}) = \sum_{i=1}^{d} \left[x_i \ln \frac{p_i}{q_i} + (1 - x_i) \ln \frac{1 - p_i}{1 - q_i} \right] + \ln \frac{P(\omega_1)}{P(\omega_2)} \tag{2.116}$$

式(2.116)关于 x_i 是线性的，因而可以改写成线性判别函数的形式

$$d(\boldsymbol{X}) = \sum_{i=1}^{d} w_i x_i + w_0 \tag{2.117}$$

式中，

$$w_i = \ln \frac{p_i (1-q_i)}{q_i (1-p_i)}$$

$$w_0 = \sum_{i=1}^{d} \ln \frac{1-p_i}{1-q_i} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

根据决策法则，如果 $d(\boldsymbol{X}) > 0$，则把 $\boldsymbol{X}$ 归到 ω_1 类；否则归到 ω_2 类。式(2.117)说明 $d(\boldsymbol{X})$是 $\boldsymbol{X}$ 的分量的线性组合，系数是权 w_i，它的值表示在做分类决策时对特征 x_i 做“是”回答的关联程度。如果 $p_i = q_i$，则 $w_i = 0$，说明 x_i 不能给出任何关于分类的信息；如果 $p_i > q_i$，则 $1-p_i < 1-q_i$，从而 $w_i > 0$，在这种情况下，特征 x_i 对于 ω_1 类回答“是”的频率要高于 ω_2 类；最后，如果有 $p_i < q_i$，则 $w_i < 0$，此时特征 x_i 对于 ω_1 类回答“是”的频率要低于 ω_2 类。在判别中，先验概率仅对阈值权 w_0 起作用。若 $P(\omega_1)$ 增加，则 w_0 增加，判决就偏向于 ω_1 类。当 $P(\omega_1)$ 减小时，结果正相反。

几何上，所有样本都位于 d 维超立方体的顶点。由 $d(\boldsymbol{X}) = 0$ 定义的决策面是一个把属于 ω_1 类的顶点和属于 ω_2 类的顶点分开的超平面。显然，在离散情况下，我们可以移动这个决策面，只要它不和任何顶点相交，并且不改变分类错误概率就可以。

2.6　贝叶斯分类器的错误率

在分类过程中，任何一种决策规则都有对应的错误率或误差。当类条件概率密度和先验概率已知时，采用指定的决策规则进行分类的错误率是固定的，在决策规则确定后，通常总以错误率来评价其性能。特别是当同一个分类问题使用几种不同的分类方案时，通常总以错误率作为方案比较的标准[3]。因此，在模式识别的理论和实践中，错误率是非常重要的参数。尽管错误率的概念简单，但是计算过程复杂，这促使人们研究错误率计算与估计的方法：一是按照理论公式进行计算，二是直接进行实验估计。

从 2.1 节中介绍的平均错误率 $P(e)$的概念可以看出，当 $\boldsymbol{x}$ 是多维向量时，计算 $P(e)$实际上要进行多重积分计算。所以，从表面上看，错误率的理论计算公式不复杂，但在多维情况下，类条件概率密度函数的表达式又比较复杂时，计算错误率是相当困难的。因此，一般情况下采用实验估计错误率，只在一些特殊情况下用理论公式进行计算。下面先介绍在一种特殊情况下的错误率的理论计算，然后讨论实验估计。

(1)一种特殊情况下的错误率的理论计算

假设为两类情况，模式服从正态分布，而且两类的协方差矩阵相等，即

$$\begin{aligned} p(\boldsymbol{x} \mid \omega_1) &= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_1) \right] \\ p(\boldsymbol{x} \mid \omega_2) &= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (\boldsymbol{x}-\boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_2) \right] \end{aligned} \tag{2.118}$$

其中 $\boldsymbol{\mu}_1$ 和 $\boldsymbol{\mu}_2$ 分别为两类的期望向量，$\boldsymbol{\Sigma}$ 为协方差矩阵。下面用理论公式求错误率。

由 2.1 节可知，最小错误率贝叶斯决策规则可以表示为

$$\begin{cases} 若\dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \dfrac{p(\omega_2)}{p(\omega_1)}, & 则\boldsymbol{x}\in\omega_1 \\ 若\dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} < \dfrac{p(\omega_2)}{p(\omega_1)}, & 则\boldsymbol{x}\in\omega_2 \end{cases} \tag{2.119}$$

两边取负对数，进一步改写为

$$\begin{cases} 若-\ln p(\boldsymbol{x}|\omega_1)+\ln p(\boldsymbol{x}|\omega_2) < \ln\dfrac{p(\omega_1)}{p(\omega_2)}, & 则\boldsymbol{x}\in\omega_1 \\ 若-\ln p(\boldsymbol{x}|\omega_1)+\ln p(\boldsymbol{x}|\omega_2) > \ln\dfrac{p(\omega_1)}{p(\omega_2)}, & 则\boldsymbol{x}\in\omega_2 \end{cases} \tag{2.120}$$

令 $h(\boldsymbol{x})=-\ln p(\boldsymbol{x}\mid\omega_1)+\ln p(\boldsymbol{x}\mid\omega_2)$，$t=\ln\dfrac{p(\omega_1)}{p(\omega_2)}$，称 $h(\boldsymbol{x})$ 为负对数似然比，则决策规则简化为

$$\begin{cases} 若h(\boldsymbol{x})<t, & 则\ \boldsymbol{x}\in\omega_1 \\ 若\ h(\boldsymbol{x})>t, & 则\boldsymbol{x}\in\omega_2 \end{cases} \tag{2.121}$$

令 $h(\boldsymbol{x})$为 $\boldsymbol{x}$ 的函数，$\boldsymbol{x}$ 是随机向量，则 $h(\boldsymbol{x})$也是随机变量，将它的分布密度函数表示为 $p(h\mid\omega_i)$。由于它是一维密度函数，因此易于积分，所以用它计算错误率有时比较方便。这样，

$$P_1(e)=\int_{R_2} p(\boldsymbol{x}\mid\omega_1)\mathrm{d}\boldsymbol{x}=\int_t^{\infty} p(h\mid\omega_1)\mathrm{d}h \tag{2.122}$$

从上式可以看到，只要知道 $h(\boldsymbol{x})$密度函数的形式就可以算出 $P_1(e)$ 和 $P_2(e)$。将式(2.118)代入 $h(\boldsymbol{x})$，可得

$$\begin{aligned} h(\boldsymbol{x}) &= -\ln p(\boldsymbol{x}\mid\omega_1)+\ \ln p(\boldsymbol{x}\mid\omega_2) \\ &= -\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln|\boldsymbol{\Sigma}|\right]+ \\ &\quad \left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln|\boldsymbol{\Sigma}|\right] \\ &= (\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}+\frac{1}{2}(\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2) \end{aligned} \tag{2.123}$$

从上式看出，$h(\boldsymbol{x})$是 $\boldsymbol{x}$ 的线性函数。$\boldsymbol{x}$ 是服从正态分布的随机向量，根据 2.3.1 节介绍的内容可知，$h(\boldsymbol{x})$服从一维正态分布，对于 $p(h\mid\omega_1)$ 可以算出决定一维正态分布的参数均值 η_1 和方差 σ_1^2。

$$\begin{aligned} \eta_1 &= E[h(\boldsymbol{x})\mid\omega_1] \\ &= (\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1+\frac{1}{2}(\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2) \\ &= -\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) \end{aligned}$$

令 $\eta=\dfrac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)$，则有

$$\eta_1 = -\eta$$

$$\sigma_1^2 = E[(h(\boldsymbol{x}) - \eta_1)^2 \mid \omega_1] = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = 2\eta = \sigma^2$$

同理可得 $p(h \mid \omega_2)$ 的参数均值 η_2 和方差 σ_2^2：

$$\eta_2 = E[h(\boldsymbol{x}) \mid \omega_1] = \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = \eta$$

$$\sigma_2^2 = E[(h(\boldsymbol{x}) - \eta_2)^2 \mid \omega_2] = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = 2\eta = \sigma^2$$

因此，可以利用 $p(h \mid \omega_1)$ 和 $p(h \mid \omega_2)$ 计算出 $P_1(e)$ 和 $P_2(e)$。

$$\begin{aligned} P_1(e) &= \int_t^{\infty} p(h/\omega_1)\,\mathrm{d}h \\ &= \int_t^{\infty} \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \exp\left[-\frac{1}{2}\left(\frac{h+\eta}{\sigma}\right)^2\right] \mathrm{d}h \\ &= \int_t^{\infty} \frac{1}{(2\pi)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\left(\frac{h+\eta}{\sigma}\right)^2\right] \mathrm{d}\left(\frac{h+\eta}{\sigma}\right) \\ &= \int_{\left(\frac{t+\eta}{\sigma}\right)}^{\infty} (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\xi^2\right) \mathrm{d}\xi \end{aligned} \tag{2.124}$$

$$\begin{aligned} P_2(e) &= \int_{-\infty}^{t} p(h/\omega_2)\mathrm{d}h \\ &= \int_{-\infty}^{t} \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} \exp\left[-\frac{1}{2}\left(\frac{h-\eta}{\sigma}\right)^2\right] \mathrm{d}h \\ &= \int_{-\infty}^{t} \frac{1}{(2\pi)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\left(\frac{h-\eta}{\sigma}\right)^2\right] \mathrm{d}\left(\frac{h-\eta}{\sigma}\right) \\ &= \int_{-\infty}^{\left(\frac{t-\eta}{\sigma}\right)} (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\xi^2\right) \mathrm{d}\xi \end{aligned} \tag{2.125}$$

其中 $t = \ln\left[\dfrac{p(\omega_1)}{p(\omega_2)}\right], \sigma = \sqrt{2\eta}$。式(2.124)和式(2.125)的计算可以查标准正态分布 $N(0, 1)$ 的累积分布函数表得到，这样就通过理论计算得到了特殊情况下的错误率。

(2)实际分类器错误率的估计

在处理实际问题时，更多地依赖于实验，即利用样本来估计错误率[4]。下面分两种情况讨论分类器错误率的估计。

(a)先验概率未知时的分类器错误率——随机抽样计算

如果分类器已经给定，先验概率 $p(\omega_i)$ (i =1, 2, …, c)未知，我们可以简单地随机抽取 N 个样本，用来检验给定的分类器。假定错分的样本数目为 τ，可以认为错误率的估计值等于被错分的样本数目与样本总数之比，即

$$\hat{\varepsilon}=\frac{\tau}{N} \tag{2.126}$$

由于每次任意抽取 N 个样本，每次试验的错分样本数就可能不同，所以 τ 是一个离散的随机变量。下面从理论上说明上式的估计是最好的。

设 ε 是真实的错误率。在给定 τ 后， ε 的密度函数为

$$P(\tau\mid\varepsilon)=C_N^{\tau}\varepsilon^{\tau}(1-\varepsilon)^{N-\tau} \tag{2.127}$$

ε 的最大似然估计 $\hat{\varepsilon}$ 就是下列方程的解：

$$\frac{\partial\ln P(\tau\mid\varepsilon)}{\partial\varepsilon}=\frac{\tau}{\varepsilon}-\frac{N-\tau}{1-\varepsilon}=0$$

解此方程得 $\hat{\varepsilon}=\frac{\tau}{N}$，这说明 $\frac{\tau}{N}$ 是错误率 ε 的最大似然估计。

$\hat{\varepsilon}$ 的物理意义很直观，其估计的统计特性为

期望： $E(\hat{\varepsilon})=E[\tau]/N=N\varepsilon/N=\varepsilon$

方差： $\mathrm{Var}(\hat{\varepsilon})=\mathrm{Var}[\tau]/N^2=\varepsilon(1-\varepsilon)/N$

显然，它是无偏估计量。

如果没有给定分类器，只给出已知分布的 N 个样本，则这些样本既要用来设计分类器，又要用来检验分类器的分类错误率。

(b)先验概率已知时的分类器错误率——选择抽样计算

如果先验概率 $p(\omega_i)$ 已知，可分别从不同的类别中抽取 $N_i=p(\omega_i)\times N$ 个样本，总的样本数 $N=N_1+N_2+\cdots+N_c$，这种样本抽取法称为选择抽取。

设 k_i 是本属于 ω_i 类而被错分的样本数，因 $k_1, k_2, \cdots, k_C$ 是独立的，其联合概率为

$$p(k_1,k_2,\cdots,k_c)=p(k_1)p(k_2)\cdots p(k_c)=\prod_{i=1}^{c}C_{N_i}^{k_i}\varepsilon_i^{k_i}(1-\varepsilon_i)^{N_i-k_i} \tag{2.128}$$

其中 ε_i 为 ω_i 类的真实错误率。利用同样的方法可得 ε_i 的最大似然估计为

$$\hat{\varepsilon}_i=\frac{k_i}{N_i}(i=1,2,\cdots,c) \tag{2.129}$$

而总的错误率估计为

$$\hat{\varepsilon}=\sum_{i=1}^{C}p(\omega_i)\hat{\varepsilon}_i \tag{2.130}$$

$\hat{\varepsilon}$ 的统计特征为

$$\text{期望：}\quad E[\hat{\varepsilon}]=p(\omega_1)E[\hat{\varepsilon}_1]+p(\omega_2)E[\hat{\varepsilon}_2]+\mathrm{L}+p(\omega_c)E[\hat{\varepsilon}_c]=\sum_{i=1}^{c}p(\omega_i)E[\hat{\varepsilon}_i] \tag{2.131}$$

协方差：　$$\mathrm{Var}[\hat{\varepsilon}]=\frac{1}{N}\sum_{i=1}^{c}p(\omega_i)\varepsilon_i(1-\varepsilon_i)\tag{2.132}$$

显然它也是无偏估计。

习题 2

2.1　两类问题的最小错误率贝叶斯决策的内容是什么？

2.2　两类问题的最小风险贝叶斯决策的内容是什么？

2.3　简述密度函数的参数估计与非参数估计方法的主要差别。

2.4　简述最大似然估计与贝叶斯估计的基本思想及主要差别。

2.5　设在一维特征空间中两类样本服从正态分布，$\sigma_1=\sigma_2=1$，$\mu_1=0$，$\mu_2=3$，两类先验概率之比 $p(\omega_1)/p(\omega_2)=e$，试求基于最小错误率贝叶斯决策原则的决策分界面的 $\boldsymbol{x}$ 值。

2.6　设有两类正态分布的样本集，第一类均值 $\boldsymbol{\mu}_1=(2,0)^{\mathrm{T}}$，方差 $\boldsymbol{\Sigma}_1=\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}$，第二类均值 $\boldsymbol{\mu}_2=(2,2)^{\mathrm{T}}$，方差 $\boldsymbol{\Sigma}_2=\begin{bmatrix}1 & -1/2\\ -1/2 & 1\end{bmatrix}$，先验概率 $p(\omega_1)=p(\omega_2)$，按最小错误率贝叶斯决策求两类的分界面。

2.7　已知某一正态分布二维随机变量的协方差矩阵为 $\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}$，均值向量为零向量。试求其 Mahalanobis 距离为 1 的点轨迹。

2.8　对两类问题，若损失函数 $\lambda_{11}=\lambda_{22}=0,\lambda_{12}\neq 0,\lambda_{21}\neq 0$，试求基于最小风险贝叶斯决策分界面处的两类错误率与 $\lambda_{12},\lambda_{21}$ 的关系。

2.9　设一个二维空间中的两类样本服从正态分布，其参数分别为 $\boldsymbol{\mu}_1=(1,0)^{\mathrm{T}}$，$\boldsymbol{\Sigma}_1=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$，$\boldsymbol{\mu}_2=(-1,0)^{\mathrm{T}}$，$\boldsymbol{\Sigma}_2=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$，先验概率 $p(\omega_1)=p(\omega_2)$。试证明其基于最小错误率的贝叶斯决策分界面为圆，并求其方程。

2.10　将上题推广到一般情况，若 $\boldsymbol{\Sigma}_1=\sigma^2\boldsymbol{I}$，$\boldsymbol{\Sigma}_2=k\boldsymbol{\Sigma}_1$，试判断先验概率相等条件下，基于最小错误率的贝叶斯决策面是否是超球面。

参考文献

[1]　杨光正等. 模式识别. 合肥：中国科学技术大学出版社，2001:30, 34, 59.

[2]　钟珞等. 模式识别. 武汉：武汉大学出版社，2006:6, 12, 17, 26.

[3]　边肇祺等. 模式识别. 第二版. 北京：清华大学出版社，1999:9, 14, 26, 28, 35.

[4]　温熙森等. 模式识别与状态监控. 长沙：国防科技大学出版社，1997:170, 171, 177.

[5]　舒宁等. 模式识别的理论与方法. 武汉：武汉大学出版社，2004:8, 13, 18, 21, 30.

[6]　孙即祥. 现代模式识别. 长沙：国防科技大学出版社，2001:128.

[7] Richard O. Duda Peter E. Hart David G. Stork 著，李宏东等译. 模式分类. 北京：机械工业出版社，2003:67, 70.

[8] 肖健华. 智能模式识别方法. 广州：华南理工大学出版社，2006.

[9] Sergios Theodoridis. *Pattern Recognition*. 李晶皎译. 模式识别(第三版). 北京：电子工业出版社，2006.

[10] C. E. Thomaz and D. F. Gillies. *Small sample size: a methodological problem in Bayes plug-in classifier for image recognition*, Technical Report, Department of Computing, Imperial College of Science and Technology, UK, 2001.

第 3 章　线性判别函数

前面所讨论的贝叶斯准则指明了根据统计参数进行分类决策的方向，具有理论指导意义。贝叶斯准则是在已知类条件概率密度 $p(\boldsymbol{x}\mid\omega_i)$ 的参数表达式和先验概率 $p(\omega_i)$ 的前提下，利用样本估计 $p(\boldsymbol{x}\mid\omega_i)$ 的未知参数，再用贝叶斯定理将其转换成后验概率 $p(\omega_i\mid\boldsymbol{x})$，并根据后验概率的大小进行分类决策的方法。但是在许多实际问题中，由于样本特征空间的类条件概率密度的形式常常很难确定，利用 Parzen 窗等非参数方法估计分布又往往需要大量的样本，而且随着特征空间维数的增加，所需样本数急剧增加。因此，在实际问题中，可以考虑另外一种分类方法，即根据训练样本集提供的信息，直接进行分类[1]。这种方法省略了统计分布状况分析与参数估计环节，而是直接对特征空间进行划分，是当前模式分类中主要使用的方法之一。由于决策域的分界面是用数学表达式描述的，如线性函数或各种非线性函数等，所以分界面方程的确定主要包括函数类型选择和最佳参数确定两部分。一般来说，函数类型是由设计者选择的，但其参数的确定则是依据一定的准则函数，通过一个学习过程来实现优化的[2]。本章介绍线性判别函数的形式以及确定判别函数依据的一些准则函数。

3.1　线性判别函数

设模式 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 是 n 维的，如果可以用线性方程 $d(\boldsymbol{x})=0$ 将分别属于不同类别的 $\boldsymbol{x}$ 划分开，则这个线性方程称为线性判别函数，其一般形式是

$$d(\boldsymbol{x})=\omega_1x_1+\omega_2x_2+\mathrm{L}\ +\omega_nx_n+\omega_{n+1}=\boldsymbol{w}_0^{\mathrm{T}}\boldsymbol{x}+\omega_{n+1} \tag{3.1}$$

式中，$\boldsymbol{w}_0=(\omega_1,\omega_2,\mathrm{L}\ ,\omega_n)^{\mathrm{T}}$ 称为参数向量或权向量。若令 $\boldsymbol{w}=(\omega_1,\omega_2,\cdots,\omega_n,\omega_{n+1})^{\mathrm{T}}$，$\boldsymbol{x}=(x_1,x_2,\mathrm{L}\ ,x_n,1)^{\mathrm{T}}$，则式(3.1)可写为

$$d(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \tag{3.2}$$

其中 $\boldsymbol{x}$ 和 $\boldsymbol{w}$ 分别称为增广特征(模式)向量和增广权向量，此时的增广特征向量的全体称为增广特征空间，引入它们是为了叙述和计算方便。在给出了线性判别函数的一般形式以后，下面分别给出不同情况下的判别规则。

(1)两类别问题

以二维空间为例来说明两类模式的情况。在二维空间中存在线性判别函数

$$d(\boldsymbol{x})=\omega_1x_1+\omega_2x_2+\omega_3=0$$

可以很明显地看到属于 ω_1 类的任一模式代入 $d(\boldsymbol{x})$ 后为正值，而属于 ω_2 类的任一模式代入 $d(\boldsymbol{x})$ 后为负值，如图3.1所示。

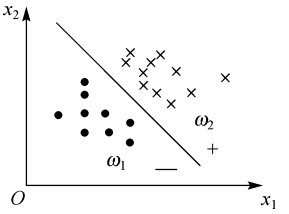


图 3.1　两类模式的线性判别函数

多维空间和二维空间情况相同，只是在三维空间判别界面为平面，在三维以上的多维空

间，判别界面为超平面。所以，对于待识别模式的增广特征向量 $\boldsymbol{x}$，可以通过下面的判别规则进行分类判别。设 $d(\boldsymbol{x})$为判别函数，则

$$d(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\begin{cases}>0, & \boldsymbol{x}\in\omega_1\\ <0, & \boldsymbol{x}\in\omega_2\\ =0, & \boldsymbol{x}\in\omega_i\text{或拒分}\end{cases}\tag{3.3}$$

其中 $d(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}=0$ 为判别边界。

(2) 多类别问题

在实际应用中，模式类别往往不止两个。前面介绍的两类别判别方法可以推广应用到类别数大于 2 的多维情况。假设有 $\omega_1,\omega_2,\cdots,\omega_c$ 类模式，在讨论判别问题时，分为以下三种情况。

第一种情况： ω_i 和 $\overline{\omega}_i$ 可分的情况

在这种情况下，每一个模式类与其他模式类之间都可用单个判别界面分开，即所确定的判别函数将属于 ω_i 类和不属于 ω_i 类的模式划分开是处理这种情况的基本思想。这样，c 类问题就转变为 $c-1$ 个两类问题。如果模式是线性可分的，一般需要建立 $c-1$ 个独立的判别函数：

$$d_i(\boldsymbol{x})=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\ ,\quad i=1,2,\cdots,c$$

通过训练，其中每个判别函数都具有以下性质：

$$d_i(\boldsymbol{x})=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\begin{cases}>0, & \boldsymbol{x}\in\omega_i,i=1,2,\cdots,c\\ \le 0, & \boldsymbol{x}\notin\omega_i\end{cases}\tag{3.4}$$

式中，$\boldsymbol{w}_i=(\omega_{i1},\omega_{i2},\mathrm{L}\ ,\omega_{in},\omega_{i(n+1)})^{\mathrm{T}}$ 为第 i 类判别函数的权向量。判别界面 $d_i(\boldsymbol{x})=0$ 将特征空间划分成两个子区域，其中一个子区域为包含 ω_i 的类域 Ω_i，另一个子区域为包含不属于 ω_i 的类域；同样，另一个判别界面 $d_j(\boldsymbol{x})=0$ 也将特征空间划分成两个子区域，其中一个子区域为包含 ω_j 的类域 Ω_j，另一个子区域为包含不属于 ω_j 的类域。

这种情况的模式类别分布如图3.2所示，每一类都可用单个判别边界与其他类别区别开来。

例如图3.2中某一模式 $\boldsymbol{x}_p$ 属于 ω_1，可见有 $d_1(\boldsymbol{x}_p)>0$，而 $d_2(\boldsymbol{x}_p)<0$，$d_3(\boldsymbol{x}_p)<0$，ω_1 类与其他类由边界 $d_1(\boldsymbol{x})=0$ 分开。

使用这类判别函数，可能会同时出现两个或两个以上的判别式都大于零或所有的判别式都小于零的情况。对于出现在这种区域中的点，不能判别出它们的类别，我们称这样的区域为不确定区，用 IR 表示，类别越多，不确定区域也就越多。$d_i(\boldsymbol{x})>0$ 只表明 $\boldsymbol{x}$ 位于 ω_i 类所在的半空间中，而这个半空间还可能含有其他的类域，因此仅用一个判别函数 $d_i(\boldsymbol{x})>0$ 不能可靠地判别出 $\boldsymbol{x}\in\omega_i$，还必须有 $d_j(\boldsymbol{x})<0$，$\forall j\ne i$，通过多个不等式的联立，使判别区域变小，从而判别结果更准确。所以，对于 m 类问题，判决规则为

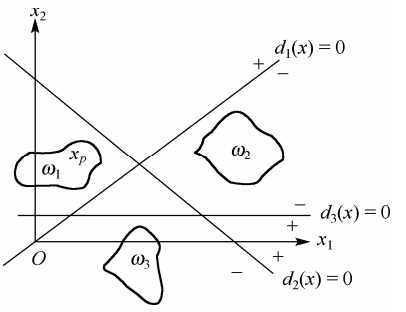


图 3.2 第一种情况分布

$$\text{如果}\quad \begin{cases} d_i(\boldsymbol{x})>0 \\ d_j(\boldsymbol{x})\leqslant 0,\ \forall j\neq i,\ j=1,2,\cdots,c \end{cases} \text{则 } \boldsymbol{x}\in\omega_i \tag{3.5}$$

【例 3.1】 设有一个三维三类问题，判别函数已经求得如下：

$$d_1(\boldsymbol{x})=-x_1+x_2+x_3+1,\quad d_2(\boldsymbol{x})=x_1+x_2-3,\quad d_3(\boldsymbol{x})=-x_1-x_3+2$$

有模式 $\boldsymbol{x}=(7,4,1)^{\mathrm{T}}$，判断其类别。

解：可将该模式特征向量代入 $d_1(\boldsymbol{x})$，$d_2(\boldsymbol{x})$ 和 $d_3(\boldsymbol{x})$ 中，得到

$$d_1(\boldsymbol{x})=-1<0\rightarrow \boldsymbol{x}\notin\omega_1(\boldsymbol{x}\in\omega_2\text{或}\omega_3)$$
$$d_2(\boldsymbol{x})=8>0$$
$$d_3(\boldsymbol{x})=-6<0\rightarrow \boldsymbol{x}\notin\omega_3(\boldsymbol{x}\in\omega_1\text{或}\omega_2)$$

根据上面的计算结果，最后判断 $\boldsymbol{x}\in\omega_2$。

第二种情况：ω_i/ω_j 可分的情况

在这种情况下，不能将每个模式类别与其他类别用单个判别界面完全分隔，但对于 c 类中任意两类 ω_i 和 ω_j 都可以分别建立一个判别函数，将属于 ω_i 类的模式和属于 ω_j 类的模式分开，该判别函数对其他类不提供模式分类信息。这种情况如图 3.3 所示，判别界面 $d_{12}(\boldsymbol{x})=0$ 通过 ω_3 类，只能将 ω_1 类和 ω_2 类分开，而不能将 ω_1 类与 ω_2 类和 ω_3 类完全分开。既然每两个类之间存在一个判别界面，而从 c 元中取 2 元的组合数为 $c(c-1)/2$，所以对于 c 类模式来说，需要 $c(c-1)/2$ 个判别函数将各个模式类分开。

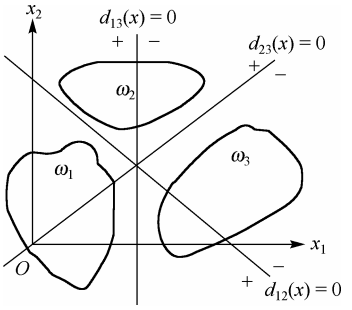


图 3.3　第二种情况分布

通过训练得到区分 ω_i 和 ω_j 两类的判别函数为[4]

$$d_{ij}(\boldsymbol{x})=\boldsymbol{w}_{ij}^{\mathrm{T}}\boldsymbol{x},\quad i,j=1,2,\mathrm{L}\ ,c;\ i\neq j$$

其具有如下性质：

$$d_{ij}(\boldsymbol{x})=\boldsymbol{w}_{ij}^{\mathrm{T}}\boldsymbol{x}\begin{cases}>0, & \boldsymbol{x}\in\omega_i\\ <0, & \boldsymbol{x}\in\omega_j\end{cases}$$

$$d_{ij}(\boldsymbol{x})=-d_{ji}(\boldsymbol{x}) \tag{3.6}$$

虽然 $d_{ij}(\boldsymbol{x})$ 具有这样的属性，但是不能仅仅根据 $d_{ij}(\boldsymbol{x})$ 的正负来判定 $\boldsymbol{x}$ 是否属于 ω_i 类，只能判断出 $\boldsymbol{x}$ 是位于包含 ω_i 类的半空间中，还是位于含有 ω_j 类的半空间中，这是因为在某个半空间中还可能含有其他的类域(或一部分)。所以除了关心 $d_{ij}(\boldsymbol{x})$ 的正负之外，还要考虑其他类域的判别函数才能做出正确的判决。所以这种情况的判别规则是

$$\text{如果}\quad d_{ij}(\boldsymbol{x})>0,\forall j\neq i,\ j=1,2,\cdots,c,\quad \text{则 } \boldsymbol{x}\in\omega_i \tag{3.7}$$

在这种方法中依然存在不确定区。

【例 3.2】 设有一个三维三类问题，三个判别函数分别为

$$d_{12}(\boldsymbol{x}) = x_1 - 2x_2 + 3x_3$$
$$d_{13}(\boldsymbol{x}) = -x_2 - x_3 + 2$$
$$d_{23}(\boldsymbol{x}) = -x_1 - 2x_2 - 3x_3 + 6$$

判断 $\boldsymbol{x} = (7, 3, 2)^{\mathrm{T}}$ 属于哪一类。

解：将 $\boldsymbol{x} = (7, 3, 2)^{\mathrm{T}}$ 代入上述判别函数有

$$d_{12}(\boldsymbol{x}) = 7\ ;\quad d_{13}(\boldsymbol{x}) = -3\ ;\quad d_{23}(\boldsymbol{x}) = -13$$

利用 $d_{ij}(\boldsymbol{x}) = -d_{ji}(\boldsymbol{x})$ 的性质，有

$$d_{21}(\boldsymbol{x}) = -7\ ;\quad d_{31}(\boldsymbol{x}) = 3\ ;\quad d_{32}(\boldsymbol{x}) = 13$$

由于

$$\begin{cases} d_{21}(\boldsymbol{x}) = -7 < 0 \\ d_{23}(\boldsymbol{x}) = -13 < 0 \end{cases},\quad \begin{cases} d_{12}(\boldsymbol{x}) = 7 > 0 \\ d_{13}(\boldsymbol{x}) = -3 < 0 \end{cases}$$

而 $d_{3j}(\boldsymbol{x}) > 0, j = 1,2$，所以判断 $\boldsymbol{x} \in \omega_3$。

对于 $c > 3$ 的多类模式来说，ω_i / ω_j 可分的情况比 $\omega_i / \overline{\omega}_i$ 可分的情况需要更多的判别函数，但是 $\omega_i / \overline{\omega}_i$ 可分的情况下是将 ω_i 类和其余 c–1 类区分开，而 ω_i / ω_j 可分的情况下是将 ω_i 类和 ω_j 类分开，显然 ω_i / ω_j 可分的情况使模式更容易线性可分。

3.2 广义线性判别函数

对于多类模式，其特征空间的分布情况往往比较复杂，但只要它们的分布不重叠或重叠度不大，总可以确定其判别边界，而这些判别边界往往都是曲线或曲面等非线性界面。设 n 维模式特征向量集 $\{\boldsymbol{x}_i\}$ 在特征空间 X^n 中是非线性可分的，对各个模式 $\boldsymbol{x}$ 做非线性变换：

$$X^n \to Y^m, m > n,\ \boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}\ ,\quad \boldsymbol{y} = (y_1, y_2, \cdots, y_m)^{\mathrm{T}} \tag{3.8}$$

其中 $y_i = f_i(\boldsymbol{x}), i = 1,2,\cdots,m$ 为 $\boldsymbol{x}$ 的单值实函数，选取适当的函数 $f_i(\boldsymbol{x})$，使 $\{y_j = (f_1(x_j), f_2(x_j), \cdots, f_m(x_j))^{\mathrm{T}}\}$ 在特征空间 Y^m 中线性可分，即分类界面是线性的。这样，在原始特征空间 X^n 中的非线性判别函数在变换特征空间 Y^m 中成为线性判别函数，我们称之为广义线性判别函数。X^n 中非线性判别函数 $d(\boldsymbol{x})$ 的一般形式和 Y^m 中相应的线性判别函数 $d(\boldsymbol{y})$ 的关系表示如下：

$$\begin{aligned} d(\boldsymbol{x}) &= \omega_1 f_1(\boldsymbol{x}) + \omega_2 f_2(\boldsymbol{x}) + \mathrm{L}\ + \omega_n f_n(\boldsymbol{x}) + \omega_{n+1} \\ &= \omega_1 y_1 + \omega_2 y_2 + \mathrm{L}\ + \omega_m y_m + \omega_{m+1} \\ &= \boldsymbol{w}^{\mathrm{T}} \boldsymbol{y} = d(\boldsymbol{y}) \end{aligned} \tag{3.9}$$

式中，$\boldsymbol{w} = (\omega_1, \omega_2, \cdots, \omega_{m+1})^{\mathrm{T}}$，$\boldsymbol{y} = (y_1, y_2, \cdots, y_m, 1)^{\mathrm{T}} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}), 1)^{\mathrm{T}}$，$y_i = f_i(\boldsymbol{x})$，$i = 1, 2, \cdots, m$ 为 $\boldsymbol{x}$ 的单值实函数。

显然，在 3.1 节中介绍的线性判别函数，实际上是广义线性判别函数在 $y_i = f_i(\boldsymbol{x}) =$

$x_i, i=1,2,\cdots,n$ 下的特例。由于相当多的实际问题不是线性可分的，任何非线性判别函数总可以变换成广义线性判别函数，而线性判别技术已经有了比较全面的理论和方法，因此研究广义线性判别函数具有重要的意义。

很明显，如果 $f_i(\boldsymbol{x})$ 为一次多项式，则 $d(\boldsymbol{x})$为线性判别函数。若 $f_i(\boldsymbol{x})$ 为二次多项式，例如 $\boldsymbol{x}$ 为二维模式，即 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$，此时有

$$d(\boldsymbol{x})=\omega_{11}x_1^2+\omega_{12}x_1x_2-\omega_{22}x_2^2+\omega_1x_1+\omega_2x_2+\omega_3 \tag{3.10}$$

则令

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_6)^{\mathrm{T}}=(x_1^2,x_1x_2,x_2^2,x_1,x_2,1)^{\mathrm{T}}$$
$$\boldsymbol{w}=(\omega_{11},\omega_{12},\omega_{22},\omega_1,\omega_2,\omega_3)^{\mathrm{T}}$$

于是 $d(\boldsymbol{x})=d(\boldsymbol{y})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}$，这时的判别函数已经线性化。

推广到 n 维，即 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，若其判别函数为

$$d(\boldsymbol{x})=\sum_{j=1}^{n}\omega_{jj}x_j^2+\sum_{j=1}^{n-1}\sum_{k=j+1}^{n}\omega_{jk}x_jx_k+\sum_{j=1}^{n}\omega_jx_j+\omega_{n+1} \tag{3.11}$$

对其进行线性化，令

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_m),\qquad m=(n+1)(n+2)/2 \tag{3.12}$$

则 $\boldsymbol{y}$ 的各分量为

$$y_i=f_i(\boldsymbol{x})=\boldsymbol{x}_p^s\boldsymbol{x}_q^t,\qquad p,q=1,2,\cdots,n;\ s,t=0,1 \tag{3.13}$$

有 $d(\boldsymbol{x})=d(\boldsymbol{y})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}$，其中 $\boldsymbol{w}$ 的各分量对于二次项为

$$\omega_{pq},\qquad p=1,2,\cdots,n;\ q=p,p+1,\cdots,n$$

对于一次项为 $\omega_p,\ p=1,2,\cdots,n$，对于零次项为 ω_{n+1}。

若 $f_i(\boldsymbol{x})$ 为 r 次多项式，$\boldsymbol{x}$ 为 n 维模式，则

$$f_i(\boldsymbol{x})=(\boldsymbol{x}_{p_1})^{s_1}(\boldsymbol{x}_{p_2})^{s_2}\cdots(\boldsymbol{x}_{p_r})^{s_r} \tag{3.14}$$

式中，$p_1, p_2,\cdots, p_r = 1, 2,\cdots, n$；$s_1, s_2,\cdots, s_r = 0, 1$。这时，判别函数 $d(\boldsymbol{x})$可以按下列递推关系写出：

常数项　$d_{(0)}(\boldsymbol{x})=\omega_{n+1}$

一次项　$d_{(1)}(\boldsymbol{x})=\sum_{p=1}^{n}\omega_p\boldsymbol{x}_p+d_{(0)}(\boldsymbol{x})$

二次项　$d_{(2)}(\boldsymbol{x})=\sum_{p=1}^{n}\sum_{q=p}^{n}\omega_{pq}\boldsymbol{x}_p\boldsymbol{x}_q+d_{(1)}(\boldsymbol{x})$

…

r 次项　$d_{(r)}(\boldsymbol{x})=\left\{\sum_{p_1=1}^{n}\sum_{p_2=p_1}^{n}\cdots\sum_{p_r=p_{r-1}}^{n}\omega_{p_1p_2\cdots p_r}\boldsymbol{x}_{p_1}^{s_1}\boldsymbol{x}_{p_2}^{s_2}\cdots\boldsymbol{x}_{p_r}^{s_r}\right\}+d_{(r-1)}(\boldsymbol{x})$

由此可以推出任意有限次的判别函数，并可以将非线性判别函数线性化。

3.3 感知器算法

根据 3.1 节的讨论，如果从过去的经验中能够知道两类模式的分布规律，并在两类线性可分的情况下，就可以找到它们的分界面。但是在大多数情况下，我们所能掌握的只是按上述分布抽出的样本，我们的任务就是根据一组样本集找出这些模式类的分界面，并且希望以后抽出的样本也能被这个分界面正确地分类。感知器算法(Perception Approach)是通过训练模式样本集的“学习”来得到判别函数系数解的方法之一。“感知器”一词，是借用 20 世纪 50 年代到 60 年代初期人们对一种分类学习机模型的称呼。

根据 3.1 节的讨论，在两类线性可分的情况下，对于样本 $\boldsymbol{x}$，如果有 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}>0$，则判别 $\boldsymbol{x}$ 样本属于 ω_1，如果 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}<0$，就判别 $\boldsymbol{x}$ 样本属于 ω_2。为了简化训练过程，我们把属于 ω_2 的样本用负号来表示，而使所有样本都满足 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}>0$ 的权向量 $\boldsymbol{w}^{*}$ 称为解向量。下面介绍求解解向量的两种方法。

3.3.1 基于赏罚概念的感知器训练算法

基于赏罚概念的感知器算法是在利用样本进行训练求解向量的过程中采取的赏罚策略，对正确分类的模式则“赏”(这里用“不罚”)；错误分类的模式则“罚”，具体训练步骤如下。

已知两个训练模式集，它们分别属于 ω_1 和 ω_2 类，确定权向量的初始值为 $\boldsymbol{w}(1)$(任意取值)，然后开始用全部训练模式集进行第一轮迭代。如果第 k 次判断中，取 $\boldsymbol{x}_k\in\omega_1$，而 $\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k\leq 0$，则认为对第 k 次实验中所采用的模式 $\boldsymbol{x}_k$ 进行了错误判别，其中 $\boldsymbol{w}^{\mathrm{T}}(k)$表示第 k 次判别所采用权向量。由于判别失误，所以需要校正权向量，使

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+c\boldsymbol{x}_k \tag{3.15}$$

这里 c 为校正增量，如 $\boldsymbol{x}_k\in\omega_2$，$\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k\geq 0$，同样分类错误，需要按照下式校正权向量：

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)-c\boldsymbol{x}_k \tag{3.16}$$

如果在第 k 次判别时，不符合以上情况，则表明第 k 次迭代得到正确的分类，则权向量保持不变：

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k) \tag{3.17}$$

按照这种方式计算下去，直到所有样本都能进行正确的分类判断，得到最后的正确权向量 $\boldsymbol{w}^{*}$，只要有一个样本没有正确分类，就必须进行下一轮迭代。

这就是基于赏罚概念的感知器算法的基本原理[4]。

若对属于 ω_2 类的模式样本乘以(−1)，则校正权向量表达式(3.15)与式(3.16)完全一致。这样，式(3.16)和式(3.17)表示的算法可以统一写成

$$\boldsymbol{w}(k+1)=\begin{cases}\boldsymbol{w}(k), & \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k>0\\ \boldsymbol{w}(k)+c\boldsymbol{x}_k, & \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k\leq 0\end{cases} \tag{3.18}$$

式中 c 为正的校正增量。也就是说，基于赏罚概念的感知器算法对于正确分类的模式权，向量 $\boldsymbol{w}$ 不变；对错误分类的模式，则修正权向量。

【例 3.3】 图3.4中有 4 个模式样本：$\boldsymbol{x}_1=(0,0)^{\mathrm{T}}$，$\boldsymbol{x}_2=(1,1)^{\mathrm{T}}$，$\boldsymbol{x}_3=(2,0)^{\mathrm{T}}$，$\boldsymbol{x}_4=(2,2)^{\mathrm{T}}$。其中 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 属于 ω_1，$\boldsymbol{x}_3$ 和 $\boldsymbol{x}_4$ 属于 ω_2，用基于赏罚概念的感知器算法求判别函数中权向量的解。

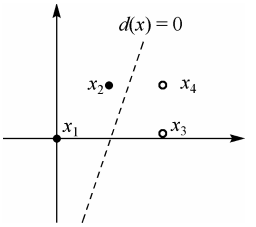


图 3.4　模式样本分布

解：根据式(3.18)求解。首先将属于 ω_2 的训练样本乘以(−1)，并将所有训练样本写成增广向量的形式：$\boldsymbol{x}_1=(0,0,1)^{\mathrm{T}}$，$\boldsymbol{x}_2=(1,1,1)^{\mathrm{T}}$，$\boldsymbol{x}_3=(-2,0,-1)^{\mathrm{T}}$，$\boldsymbol{x}_4=(-2,-2,-1)^{\mathrm{T}}$。取 $c=1$，$\boldsymbol{w}(1)=0$。

第一轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(1)\boldsymbol{x}_1=(0,0,0)(0,0,1)^{\mathrm{T}}=0$，所以 $\boldsymbol{w}(2)=\boldsymbol{w}(1)+\boldsymbol{x}_1=(0,0,1)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(2)\boldsymbol{x}_2=(0,0,1)(1,1,1)^{\mathrm{T}}=1>0$，故 $\boldsymbol{w}(3)=\boldsymbol{w}(2)$。

$\boldsymbol{w}^{\mathrm{T}}(3)\boldsymbol{x}_3=(0,0,1)(-2,0,-1)^{\mathrm{T}}=-1<0$，那么 $\boldsymbol{w}(4)=\boldsymbol{w}(3)+\boldsymbol{x}_3=(-2,0,0)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(4)\boldsymbol{x}_4=(-2,0,0)(-2,-2,-1)^{\mathrm{T}}=4>0$，因此 $\boldsymbol{w}(5)=\boldsymbol{w}(4)$。

由于第一步和第三步错误分类，而只有对全部模式分类都正确时，权向量才是正确的解，所以需要进行第二轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(5)\boldsymbol{x}_1=0$，故 $\boldsymbol{w}(6)=\boldsymbol{w}(5)+\boldsymbol{x}_1=(-2,0,1)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(6)\boldsymbol{x}_2=-1<0$，因此 $\boldsymbol{w}(7)=\boldsymbol{w}(6)+\boldsymbol{x}_2=(-1,1,2)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(7)\boldsymbol{x}_3=0$，因此 $\boldsymbol{w}(8)=\boldsymbol{w}(7)+\boldsymbol{x}_3=(-3,1,1)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(8)\boldsymbol{x}_4=3>0$，那么 $\boldsymbol{w}(9)=\boldsymbol{w}(8)$。

仍有分类错误，需要进行第三轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(9)\boldsymbol{x}_1=1>0$，于是 $\boldsymbol{w}(10)=\boldsymbol{w}(9)$。

$\boldsymbol{w}^{\mathrm{T}}(10)\boldsymbol{x}_2=-1<0$，因此 $\boldsymbol{w}(11)=\boldsymbol{w}(10)+\boldsymbol{x}_2=(-2,2,2)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(11)\boldsymbol{x}_3=2>0$，所以 $\boldsymbol{w}(12)=\boldsymbol{w}(11)$。

$\boldsymbol{w}^{\mathrm{T}}(12)\boldsymbol{x}_4=-2<0$，所以 $\boldsymbol{w}(13)=\boldsymbol{w}(12)+\boldsymbol{x}_4=(-4,0,1)^{\mathrm{T}}$。

继续进行第四轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(13)\boldsymbol{x}_1=1>0$，所以 $\boldsymbol{w}(14)=\boldsymbol{w}(13)$。

$\boldsymbol{w}^{\mathrm{T}}(14)\boldsymbol{x}_2=-3<0$，所以 $\boldsymbol{w}(15)=\boldsymbol{w}(14)+\boldsymbol{x}_2=(-3,1,2)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(15)\boldsymbol{x}_3=4>0$，所以 $\boldsymbol{w}(16)=\boldsymbol{w}(15)$。

$\boldsymbol{w}^{\mathrm{T}}(16)\boldsymbol{x}_4=2>0$，所以 $\boldsymbol{w}(17)=\boldsymbol{w}(16)$。

继续进行第五轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(17)\boldsymbol{x}_1=2>0$，所以 $\boldsymbol{w}(18)=\boldsymbol{w}(17)$。

$\boldsymbol{w}^{\mathrm{T}}(18)\boldsymbol{x}_2=0$，所以 $\boldsymbol{w}(19)=\boldsymbol{w}(18)+\boldsymbol{x}_2=(-2,2,3)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(19)\boldsymbol{x}_3=1>0$，所以 $\boldsymbol{w}(20)=\boldsymbol{w}(19)$。

$\boldsymbol{w}^{\mathrm{T}}(20)\boldsymbol{x}_4=-3<0$，所以 $\boldsymbol{w}(21)=\boldsymbol{w}(20)+\boldsymbol{x}_4=(-4,0,2)^{\mathrm{T}}$。

继续进行第六轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(21)\boldsymbol{x}_1=2>0$，所以 $\boldsymbol{w}(22)=\boldsymbol{w}(21)$。

$\boldsymbol{w}^{\mathrm{T}}(22)\boldsymbol{x}_2=-2<0$，所以 $\boldsymbol{w}(23)=\boldsymbol{w}(22)+\boldsymbol{x}_2=(-3,1,3)^{\mathrm{T}}$。

$\boldsymbol{w}^{\mathrm{T}}(23)\boldsymbol{x}_3=3>0$，所以 $\boldsymbol{w}(24)=\boldsymbol{w}(23)$。

$\boldsymbol{w}^{\mathrm{T}}(24)\boldsymbol{x}_4=1>0$，所以 $\boldsymbol{w}(25)=\boldsymbol{w}(24)$。

继续进行第七轮迭代：

$\boldsymbol{w}^{\mathrm{T}}(25)\boldsymbol{x}_1 = 3 > 0$，所以 $\boldsymbol{w}(26) = \boldsymbol{w}(25)$。

$\boldsymbol{w}^{\mathrm{T}}(26)\boldsymbol{x}_2 = 1 > 0$，所以 $\boldsymbol{w}(27) = \boldsymbol{w}(26)$。

$\boldsymbol{w}^{\mathrm{T}}(27)\boldsymbol{x}_3 = 3 > 0$，所以 $\boldsymbol{w}(28) = \boldsymbol{w}(27)$。

$\boldsymbol{w}^{\mathrm{T}}(28)\boldsymbol{x}_4 = 1 > 0$，所以 $\boldsymbol{w}(29) = \boldsymbol{w}(28)$。

到此为止，所有分类都正确，所以解向量 $\boldsymbol{w}^* = (-3, 1, 3)^{\mathrm{T}}$，相应的判别函数为

$$d(\boldsymbol{x}) = -3\boldsymbol{x}_1 + \boldsymbol{x}_2 + 3$$

图3.4中的虚线即为判别界面 $d(\boldsymbol{x}) = 0$。

3.3.2 梯度下降法

我们知道一个函数的梯度指明了当其自变量增加时，该函数增大率最大的方向，负梯度则表明在同样条件下函数下降最快的方向。基于梯度函数的这一重要性质，并且参考感知器算法的修正规则方程(3.18)，我们试图定义一个准则函数 $J(\boldsymbol{w}, \boldsymbol{x})$，使该函数的最小值对应着最优解向量 $\boldsymbol{w}^*$[8]。这样，就将求解问题简化成一个标量函数的最小化问题。

梯度下降法的原理比较简单，首先从一个随意选择的权向量 $\boldsymbol{w}(1)$ 开始，计算其梯度向量 $\nabla J_{\boldsymbol{w}=\boldsymbol{w}(1)}$，下一个值 $\boldsymbol{w}(2)$ 由 $\boldsymbol{w}(1)$ 向下降最陡的方向移一段距离得到，即沿梯度的负方向[5]。则 $\boldsymbol{w}(k+1)$ 由等式

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \rho(\nabla J)_{\boldsymbol{w}=\boldsymbol{w}(k)} \tag{3.19}$$

计算。其中 ρ 为正比例因子，如果对于所有的 $\boldsymbol{x}$，在 $\boldsymbol{w} = \boldsymbol{w}(k)$时梯度 ∇J 为最小值，则得到权向量的最优解 $\boldsymbol{w}^*$。

我们选择准则函数 J 为

$$J(\boldsymbol{w}, \boldsymbol{x}) = k(|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}| - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) \tag{3.20}$$

其中 $k > 0$。显然，只要满足 $|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}| - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = 0$，该准则函数均达到最小值 $J_{\min}(\boldsymbol{w}, \boldsymbol{x})$，此时有 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \geq 0$，因此得到最优解 $\boldsymbol{w}^*$。

假定 $\boldsymbol{x}$ 为定值，$J(\boldsymbol{w}, \boldsymbol{x})$与 $\boldsymbol{w}$ 的关系如图3.5所示。在坐标原点的左侧，$J(\boldsymbol{w}, \boldsymbol{x})$是一条下降的斜线，斜率取决于梯度；在坐标原点的右侧，$J(\boldsymbol{w}, \boldsymbol{x}) = 0$，是 $\boldsymbol{w}$ 的解区。

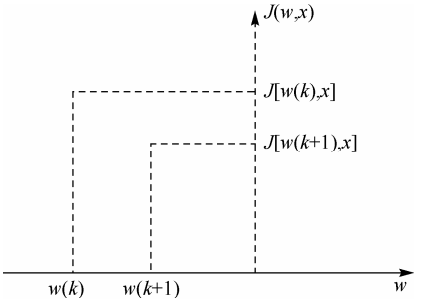


图 3.5 梯度法示意图

令 $k = 1/2$，根据式(3.20)求梯度 ∇J，即

$$\nabla J = \frac{\partial J}{\partial \boldsymbol{w}} = \frac{1}{2}[\boldsymbol{x}\,\mathrm{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) - \boldsymbol{x}] \tag{3.21}$$

式中

$$\mathrm{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} > 0 \\ -1, & \text{其他} \end{cases} \tag{3.22}$$

将式(3.21)代入式(3.19)，得到

$$\begin{aligned}\boldsymbol{w}(k+1)&=\boldsymbol{w}(k)-\frac{\rho}{2}\left\{\boldsymbol{x}_k\,\mathrm{sgn}[\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k]-\boldsymbol{x}_k\right\}\\&=\begin{cases}\boldsymbol{w}(k), & \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k>0\\ \boldsymbol{w}(k)+\rho\boldsymbol{x}_k, & \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k\leq 0\end{cases}\end{aligned}\tag{3.23}$$

当 $\rho=c>0$ 为常数时，上式与基于赏罚概念的感知器算法的修正公式(3.18)一致。可见，基于赏罚概念的感知器算法只是梯度下降法的一种特殊情况，并且把 ρ 为常数的梯度法称为固定增量法。

当 k 只记录迭代运算过程中对权向量修正的次数时，式(3.23)简化为

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+\rho_k\boldsymbol{x}_k\tag{3.24}$$

在上式中，用 ρ_k 代替 ρ，表示在迭代中 ρ 随 k 变化，称为可变增量法。在迭代过程中，对于某个 k，若要求

$$\boldsymbol{w}^{\mathrm{T}}(k+1)\boldsymbol{x}_k>0\tag{3.25}$$

将式(3.25)代入式(3.24)，可以得到

$$\rho_k\geq\frac{|\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}_k|}{\|\boldsymbol{x}_k\|^2}\tag{3.26}$$

此时的算法称为可变增量的绝对修正法。

可见，参数 ρ 的选择很重要。一般来说，ρ 选大一些，收敛速度快，但迭代过程可能会不稳定，甚至会引起发散。

与基于赏罚概念的感知器算法一样，梯度下降法只适用于线性可分的情况，否则，算法将会在解区边界两侧来回摆动而始终不收敛。

这里只介绍了线性感知器算法，对于非线性算法，请读者参见第 7 章。

3.4　最小平方误差准则函数

在 3.3 节介绍的感知器算法是在已知模式集线性可分的基础上采用的，但是对于给定的模式集往往不能预先知道是否线性可分。本节介绍的最小平方误差准则函数(LMSE)可以在训练过程中判定训练模式集是否线性可分[4]。最小平方误差准则函数是在对准则函数引进最小均方差的基础上建立起来。

3.3 节介绍的准则函数都是基于解线性不等式组的方法，共同之处是企图找到一个解向量 $\boldsymbol{w}^*$，使得满足 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}>0$ 的数目最大，从而使错分的样本数最少，在不等式组一致的情况下，得到一个解区中的解向量。

我们把不等式组写成如下形式：

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i=b_i>0\tag{3.27}$$

其中 b_i 是任意给定的正常数。将上式写成联立方程组的形式，即为

$$\boldsymbol{X}\boldsymbol{w}=\boldsymbol{b}\tag{3.28}$$

其中，

$$\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}}\\ \boldsymbol{x}_2^{\mathrm{T}}\\ \mathrm{M}\\ \boldsymbol{x}_d^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{x}_{11}\boldsymbol{x}_{12}\mathrm{L}\ \ \boldsymbol{x}_{1n}\\ \boldsymbol{x}_{21}\boldsymbol{x}_{22}\mathrm{L}\ \ \boldsymbol{x}_{2n}\\ \mathrm{M}\\ \boldsymbol{x}_{d1}\boldsymbol{x}_{d2}\mathrm{L}\ \ \boldsymbol{x}_{dn}\end{bmatrix}$$

是一个 $d\times n$ 矩阵，$\boldsymbol{x}_i$ 是规范化增广样本向量，$\boldsymbol{b}=(b_1,b_2,\cdots,b_d)^{\mathrm{T}}$ 是一个 d 维向量，$b_i>0$，$i=1,2,\cdots,d$。这样就可以用解一组线性方程组的问题来代替原来的解一组线性不等式的问题。

通常样本数 d 总是大于维数 n，因此 $\boldsymbol{X}$ 是长方阵，一般为满秩阵。实际上方程个数多于未知数的情况，一般为矛盾方程组，一般没有精确的解存在。

我们定义一个误差向量

$$\boldsymbol{e}=\boldsymbol{X}\boldsymbol{w}-\boldsymbol{b} \tag{3.29}$$

由于最小平方误差准则以最小均方误差为准则，因而定义准则函数

$$J(\boldsymbol{w})=\|\boldsymbol{e}\|^2=\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{b}\|^2=\sum_{i=1}^{d}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{b}_i)^2 \tag{3.30}$$

然后找一个使 $J(\boldsymbol{w})$ 极小化的 $\boldsymbol{w}$ 作为问题的解，即求解使 $J(\boldsymbol{w})$ 的梯度为 0 的 $\boldsymbol{w}$ 值。

首先对式(3.30)中的 $J(\boldsymbol{w})$ 求梯度，

$$\nabla J(\boldsymbol{w})=\sum_{i=1}^{d}2(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{b}_i)\boldsymbol{x}_i=2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{b}) \tag{3.31}$$

令 $\nabla J(\boldsymbol{w})=0$，得

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{b} \tag{3.32}$$

得到

$$\boldsymbol{w}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{b}=\boldsymbol{X}^{*}\boldsymbol{b} \tag{3.33}$$

其中 $n\times d$ 矩阵 $\boldsymbol{X}^{*}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}$ 称为矩阵 $\boldsymbol{X}$ 的规范逆矩阵，$\boldsymbol{w}$ 就是式(3.27)的最小平方误差准则函数的解。

可见 $\boldsymbol{w}$ 的解依赖于向量 $\boldsymbol{b}$，$\boldsymbol{b}$ 的不同选择可以赋予解不同的性质，而且式(3.33)的计算量很大，因为要求解 $n\times n$ 维矩阵的逆。为了避免上述缺点，可以采用梯度下降法。由式(3.31)可知其梯度为

$$\nabla J(\boldsymbol{w})=\sum_{i=1}^{n}2(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{b}_i)\boldsymbol{x}_i=2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{b})$$

则梯度下降法计算过程为

(1) 首先任意指定初始权向量

$$\boldsymbol{w}(1)=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{b}(1),\quad \boldsymbol{b}(1)>0 \tag{3.34}$$

(2) 如第 k 步不能满足 $\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}(k)-\boldsymbol{b}(k))=0$，则按下式计算第 $k+1$ 步的权向量：

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)-\rho\boldsymbol{X}^{\mathrm{T}}[\boldsymbol{X}\boldsymbol{w}(k)-\boldsymbol{b}(k)] \tag{3.35}$$

其中 ρ 为修正系数，则这个算法产生的权向量 $\boldsymbol{w}(k)$, $k=1,2,\cdots$ 收敛于满足方程 $\nabla J(\boldsymbol{w})=0$，且不管 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 是否为奇异阵，这个梯度下降算法总能产生一个解。

每次迭代时的误差向量 $\boldsymbol{e}(k)$，

$$\boldsymbol{e}(k)=X\boldsymbol{w}(k)-\boldsymbol{b}(k) \tag{3.36}$$

$$\boldsymbol{e}(k)=X\boldsymbol{w}(k)-\boldsymbol{b}(k) \tag{3.36}$$

是研究样本集线性可分性的重要指标。只有满足

$$\boldsymbol{e}(k)\geqslant 0\text{(即每一个分量均为正值或零)} \tag{3.37}$$

系统才线性可分，如果 $\boldsymbol{e}(k)<0$，则不能收敛。

下面以一个实例来说明这种情况。

【例 3.4】 假设有两类样本集 $\omega_1:(0,0)^{\mathrm{T}},(0,-1)^{\mathrm{T}}$ 和 $\omega_2:(-1,0)^{\mathrm{T}},(-1,-1)^{\mathrm{T}}$，如图3.6所示。

显然它们是线性可分的。将 ω_2 中样本乘以(–1)，并将所有样本写成增广矩阵(对 ω_1,ω_2 所有样本增 1，并对 ω_2 取负)得样本矩阵为

$$\boldsymbol{X}=\begin{bmatrix}0 & 0 & 1\\ 0 & -1 & 1\\ 1 & 0 & -1\\ 1 & 1 & -1\end{bmatrix}$$

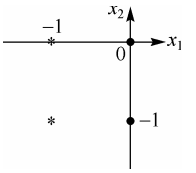


图 3.6　两类样本集示意图

则 $\boldsymbol{X}$ 的规范逆矩阵为

$$\boldsymbol{X}^{*}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}=\begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\\ 0.5 & -0.5 & -0.5 & 0.5\\ 0.75 & 0.25 & -0.25 & 0.25\end{bmatrix}$$

取 $\boldsymbol{b}(1)=(1,1,1,1)^{\mathrm{T}}$ 和 $\rho=1$，得

$$\boldsymbol{w}(1)=\boldsymbol{X}^{*}\boldsymbol{b}(1)=(2,0,1)^{\mathrm{T}}$$

因为

$$\boldsymbol{X}\boldsymbol{w}(1)=\begin{bmatrix}0 & 0 & 1\\ 0 & -1 & 1\\ 1 & 0 & -1\\ 1 & 1 & -1\end{bmatrix}\begin{bmatrix}2\\ 0\\ 1\end{bmatrix}=(1,1,1,1)^{\mathrm{T}}$$

这时

$$\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}(k)-\boldsymbol{b}(k))=\boldsymbol{X}\boldsymbol{w}(1)-\boldsymbol{b}(1)=0$$

$$\boldsymbol{E}(1)=0$$

因而 $\boldsymbol{w}(1)$ 即为所求的解，$\boldsymbol{w}^{*}=(2,0,1)^{\mathrm{T}}$，因此决策面方程为 $2x_1+1=0$。

现在把上述 4 个样本重新变换一下，使其出现不可分的状态：

$$\omega_1:(0,0)^{\mathrm{T}},(-1,-1)^{\mathrm{T}},\quad \omega_2:(0,-1)^{\mathrm{T}},(-1,0)^{\mathrm{T}}$$

此时的样本矩阵为

$$\boldsymbol{X}=\begin{bmatrix}0 & 0 & 1\\ -1 & -1 & 1\\ 0 & 1 & -1\\ 1 & 0 & -1\end{bmatrix}$$

则 $\boldsymbol{X}$ 的规范逆矩阵为

$$\boldsymbol{X}^{*}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}=\begin{bmatrix}0.5 & -0.5 & -0.5 & 0.5\\ 0.5 & -0.5 & 0.5 & -0.5\\ 0.75 & -0.25 & -0.25 & -0.25\end{bmatrix}$$

取 $\boldsymbol{b}(1)=(1,1,1,1)^{\mathrm{T}}$ 和 $\rho=1$，得

$$\boldsymbol{w}(1)=\boldsymbol{X}^{*}\boldsymbol{b}(1)=(0,0,0)^{\mathrm{T}}$$

$$\boldsymbol{E}(1)=X\boldsymbol{w}(1)-\boldsymbol{b}(1)=(-1,-1,-1,-1)^{\mathrm{T}}$$

$\boldsymbol{E}(1)$是一个全负分量的向量，说明系统是线性不可分的。

3.5 多类问题[1]

前面几节讨论了两类问题的线性判别方法。无论是在理论上还是在具体算法上，它们都有一 定的特殊性。但实际上，我们遇到的不只是两类问题，还经常遇到多类问题。因此，研究多类分类算法是很重要的。

3.5.1 多类问题的基本概念

利用线性判别函数解决多类问题的判别有多种方法。例如，可以把 c 类问题视为 $c-1$ 个两类问题，其中第 i 个问题是用线性判别函数把属于 ω_i 类的点和不属于 ω_i 类的点分开，如图3.7(a)所示。再麻烦一些的方法是用 $c(c-1)/2$ 个线性判别函数，把样本分为 c 个类别，每个线性判别函数只对其中的两个类别分类，如图3.7(b)所示。但是这两种方法都会产生如图3.7所示的阴影区域，对这个阴影区域中的点无法确定其类别。

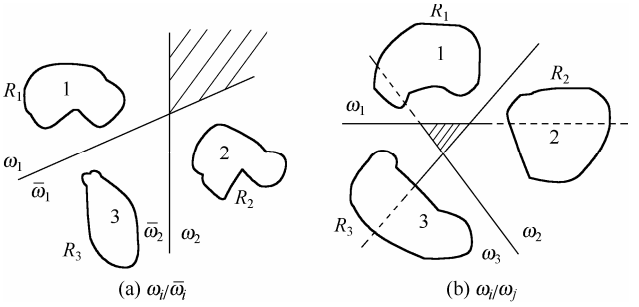


图 3.7 把多类问题转化为多个两类问题的两种情况

为此，我们定义 c 个判别函数

$$d_i(\boldsymbol{x}) = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{w}_{i0}, i = 1,2,\cdots,c \tag{3.38}$$

如果对一切 $j \neq i$，都满足 $d_i(\boldsymbol{x}) > d_j(\boldsymbol{x})$，则把 $\boldsymbol{x}$ 归于 ω_i 类；如果 $d_i(\boldsymbol{x}) = d_j(\boldsymbol{x})$，则拒绝决策。这样得到的判别规则把特征空间分为 c 个决策区域 $\Omega_1, \Omega_2, \cdots, \Omega_c$，当 $\boldsymbol{x}$ 在 Ω_i 中时，$d_i(\boldsymbol{x})$ 具有最大值。如果 Ω_i 和 Ω_j 相邻．则它们的分界面就是超平面 $\boldsymbol{H}_{ij}$ 的一部分，其定义为

$$d_i(\boldsymbol{x}) = d_j(\boldsymbol{x}) \tag{3.39}$$

或

$$(\boldsymbol{w}_i - \boldsymbol{w}_j)^{\mathrm{T}}\boldsymbol{x} + (\boldsymbol{w}_{i0} - \boldsymbol{w}_{j0}) = 0 \tag{3.40}$$

由此可知，$\boldsymbol{w}_i - \boldsymbol{w}_j$ 是 $\boldsymbol{H}_{ij}$ 的法向量，从 $\boldsymbol{x}$ 到 $\boldsymbol{H}_{ij}$ 的代数距离为

$$r = \frac{d_i(\boldsymbol{x}) - d_j(\boldsymbol{x})}{\left\|\boldsymbol{w}_i - \boldsymbol{w}_j\right\|} \tag{3.41}$$

因此，对线性机器来说，重要的是权向量的差而不是权向量本身。这时应该有 $c(c–1)/2$ 个超平面。但在实际问题中，出现在分界面上的超平面的个数往往少于 $c(c–1)/2$。下面我们在图3.8中说明在二维情况下，决策面的分布情况。

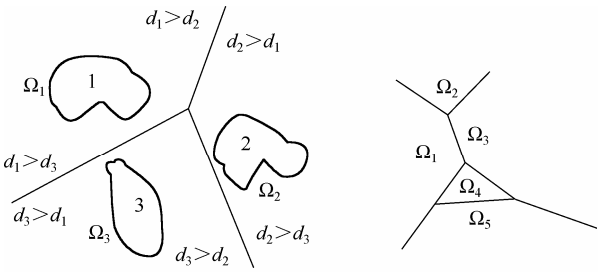


图 3.8　多类线性决策面的例子

很明显，决策面是凸的，决策区域单连通。除了这一点限制外，这种方法有很多优点，最突出的优点就是分析起来比较简单。

对于多类问题判别方法的设计，本章前面讲到的关于两类问题的准则函数和算法一般都可以推广到多类情况。但是由于计算起来非常复杂，对此我们不准备详细讨论。下面我们只介绍一下解决多类问题的决策树方法。

3.5.2　决策树简介

1．基本概念

决策树是一种模式识别中进行分类的有效方法，对于多类或多峰分布问题，这种方法尤为方便。利用决策树可以把一个复杂的多类别分类问题转化为若干简单的分类问题来解决。这种方法不是企图用一种算法、一个决策规则把多个类别一次分开，而是采用分级的形式，使分类问题逐步得到解决。图3.9所示就是一个决策树的例子。

一般来说，一个决策树由一个根节点 n_1、一组非终止节点 n_i 和一些终止节点 t_j 组成，可对 t_j 标以各种类别标签，有时不同的终止节点上可以出现相同的类别标签。如果用 T 表示决策树，那么一个决策树 T 对应于特征空间的一种划分，它把特征空间分成若干区域，在每个区域中，某个类别的样本占优势，因此可以标以该类样本的类别标签。

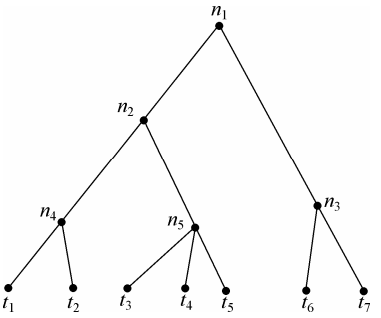


图 3.9 决策树示意图

数学上可以对决策树分类规则做如下表述。

给定样本集 R，其中的样本属于 c 个类别，用 R_i 表示 R 中属于第 i 类的样本集。定义一个指标集 $I=\{1,2,\cdots,c\}$ 和一个 I 的非空子集的集合：

$$\tau=\{I_1,I_2,\mathrm{L}\ ,I_p\}$$

我们可以令当 $i\neq j$ 时，$I_i\cap I_j=\varnothing$。一个广义决策规则 f 就是 R 到 τ 的一个映射（记为 $f:R\to\tau$）。若 f 把第 i 类的某个样本映射到包含 i 的那个子集 I_k 中，则识别正确。

设 $T(R, I)$ 是由样本集 R 和指标集 I 所形成的所有可能映射的集合，则 $T(R, I)$ 可表示为由对 (a_i,τ_i) 所组成的集合，元素 (a_i,τ_i) 称为一个节点，a_i 是该节点上表征这种映射的参数，$\tau_i=\{I_{i1},I_{i2},\cdots,I_{ip_i}\}$ 是该节点上指标集 I_i 的非空子集的集合。令 n_i 和 n_j 是 $T(R, I)$ 的两个元素，其中，

$$n_i=(a_i,\tau_i),\tau_i=\{I_{i1},I_{i2},\cdots,I_{ip_i}\}$$

$$n_j=(a_j,\tau_j),\tau_j=\{I_{j1},I_{j2},\cdots,I_{jp_j}\}$$

若 $\underset{1\leq\ l\leq\ p_j}{\cup} I_{jl}=I_{ik},\ 1\leq\ k\leq\ p_i$，则称 n_i 为 n_j 的父节点，或称 n_j 为 n_i 的子节点。

设 $B\subset T(R,I)$ 是节点的有限集，且 $n\in B$。若 B 中没有一个元素是 n 的父节点，则称 n 是 B 的根节点。当 $B\subset T(R,I)$ 满足下列条件时，它就是一个决策树分类规则：

① B 中有一个并且只有一个根节点。

② 设 n_i 和 n_j 是 B 中的两个不同元素，则 $\underset{1\leq\ k\leq\ p_i}{\cup} I_{ik}\neq\underset{1\leq\ l\leq\ p_j}{\cup} I_{jl}$。

③ 对于每一个 $i\in I$，B 中存在一个节点 $n'=(a',\tau'),\tau'=\left\{I'_1,I'_2,\mathrm{L}\ ,I'_p\right\}$，且 τ' 中有一个元素是 i（与它对应的 n' 的子节点叫叶节点，又称终止节点）。

注意，在这样定义的决策树中，每个类别标签只出现在一个叶节点上；我们当然也可以使每个类别标签出现在几个不同的叶节点上。这时，前述的当 $i\neq j$ 时 $I_i\cap I_j=\varnothing$ 的限制条件就不再成立了。

二叉树是决策树的一种简单形式。所谓二叉树，是指除叶节点外，树的每个节点仅分为两个分支，也就是说，每个节点 n_i 都有且只有两个子节点 n_{il} 和 n_{ir}。二叉树结构决策树可以把一个复杂的多类别分类问题化为多级多个两类问题来解决，在每个节点 n_i，都把样本集分为左右两个子集。分成的每一部分可能仍然包含多个类别的样本，可以把每一部分再分成两个子集，直至分成的每一部分只包含同一类别的样本，或某一类样本占优势为止。

这种二叉树结构决策树概念简单、直观，便于解释，而且在各个节点上可以选择不同的特征和采用不同的决策规则，因此设计方法灵活多样，便于利用先验知识，获得一个较好的决策树。图3.10所示是一个二叉决策树的例子。

在这个例子中，每个节点上只选择一个特征，并给出了相应的决策阈值。对于未知样本 $\boldsymbol{x}$，只要从根节点到叶节点，顺序把 $\boldsymbol{x}$ 的某个特征观察值与相应的阈值相比较，就可做出决策，把 $\boldsymbol{x}$ 分到相应的分支，最后分到合适的类别。

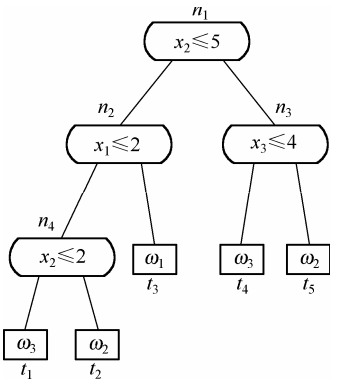


图 3.10　一个二叉决策树的例子

2．决策树设计需要考虑的基本问题

设计一个决策树，主要应解决下面几个问题：

① 选择一个合适的树结构，即合理安排树的节点和分支；

② 确定在每个非终止节点上要使用的特征；

③ 在每个非终止节点上选择合适的决策规则。

这三个问题解决了，决策树的设计也就完成了。二叉树的设计也不例外。

把一个多类别分类问题转化为两类问题的形式是多种多样的，因此，对应的二叉树的结构也将各不相同。我们的目的是要找一个最优的决策树。

显然，一个性能良好的决策树结构应该错误率小而且决策代价低。但是由于很难把错误率的解析表达式和树的结构联系起来，在每个节点上采用的决策规则也仅仅是在该节点上采用的特征观察值的函数，因此，即使每个节点上的性能都达到最优，也不能说整个决策树的性能达到最优。所以在实际问题中，人们往往提出其他一些优化准则，例如极小化整个树的节点数目，或极小化从根节点到叶节点的最大路程长度，或极小化从根节点到叶节点的平均路程长度等，然后采用动态规划方法，力争设计出能最好地满足某种准则的“最优”决策树。

3．决策树设计方法

下面举几个决策树设计方法的例子，供读者在应用决策树解决具体问题时参考。

① 穷举决策树设计方法。这是从减少错误率的要求出发的一种设计方法。

设有一个决策规则 f，它对每个样本给予一个明确的类别，即 $f: R \to I$。与此相联系的可以定义一个错误矩阵 ε_f，其中的每个元素 $\varepsilon_f(i,j)$，表示第 i 类的样本由决策规则分为第 j 类的条件概率。假使

$$\tau = \{I_1, I_2, \cdots, I_p\}$$

是 I 的互不相交的 p 个子集的集合，则从决策规则 f 可以形成一个广义决策规则 $f': R \to \tau$，使得若 $f(x) = i$，则 $f'(x) = I_{j(i)}$，$I_{j(i)}$ 是 τ 中包含 i 的一个元素。显然，决策规则 f' 的正确识别率为

$$P_{f'} = \sum_{i \in I} P(\omega_i) \sum_{l \in I_j(i)} \boldsymbol{\varepsilon}_f(i,l)$$

这里 $P(\omega_i)$ 是 ω_i 类的先验概率。我们对所有可能的树结构都进行计算以得到最大的正确识别率。当类别数很大时，这种方法计算量非常大。例如，在类别数为 15 时，不同的树的结构达到 1 386 298 975 种，因此实际实现时有很大的困难。但是若规定树结构的阶数(父节点所具有的子节点的数目)，例如规定阶数等于 2，即二叉树，则树的结构总共只有 16 384 种。对于高速计算机来说，这样做还是可行的。

② 软决策的决策树设计方法。一般的决策树希望用误识率和特征测量的平均代价的加权和作为设计的最优化准则。这样一来，在不同节点上不同决策规则的特征选择和树结构的设计就不可分割地结合在一起，从而造成了巨大的计算复杂性。当实际问题中要处理的特征数和类别数都很多时，这样做几乎是不可能的。因此提出了不少把树结构设计和特征选择(即使是局部的)加以分开的方法，有的限制每个节点只用一个特征，有的借助于启发式知识进行设计和选择特征，有的则把树结构限制为二叉树形式，在每一个节点上只把某一个类别和其他类分开等。所有这些方法的特点都是在每个非终止节点上做出唯一的决策。在对一个样本进行识别时，必须从根节点开始，沿树的路径一个节点一个节点地检验，直到叶节点上为止。在叶节点上得到它最后决策的类别标签。因此它的可靠性和根节点以及中途所有非终止节点上的决策可靠性都有关。为了克服这个缺点，在每个节点上都产生某个条件后验概率的估计向量而不做明确的分类决策。决策树和通常的单级贝叶斯分类的不同之处在于，它的条件后验概率是通过一系列步骤计算出来的。对每个节点来说，输出是输入类别集合 I' 的一个互不相交的子集集合 $\{I'_1, I'_2, \mathrm{L}\ , I'_p\}$，即 $I' = \underset{i=1,2,..,p}{\cup} I'_i$，且当 $i \neq j$ 时，$I'_i \cap I'_j = \varnothing$。子集 I'_i 的先验概率是 I'_i 中包含各类别的先验概率之和，即 $P(I'_i) = \sum_{i \in I'_i} P(i)$。数学上可以证明，在某个节点上，样本 $\boldsymbol{x}$ 对某个输出子集 I'_i 的条件后验概率可按照下式估计：

$$P(I'_i \mid I', \boldsymbol{x}) = P(i \mid \boldsymbol{x}) \frac{1}{P(I' \mid \boldsymbol{x})}$$

假设对于各个节点的后验概率估计精确度都很高，这样的决策树的性能对子树结构来说具有渐近的“鲁棒”性。我们可以选择各个节点上输出的互不相交的子集集合，使得 p 个 $P(I'_i \mid I', \boldsymbol{x})$ 的估计具有最大的可信度。从原则上说，这样做需要检验 I 的各种可能的子集集合 $\{I_1, I_2, \cdots, I_p\}$，显然计算量是非常大的。对于子类别互相不重叠的情况，可以认为，后验概率估计的可靠性和 I'_i 与 I'_j 之间的距离是成比例的，因此有可能用分级聚类的方法来确定集合 $\{I_1, I_2, \cdots, I_p\}$，从而大大简化计算的复杂度。

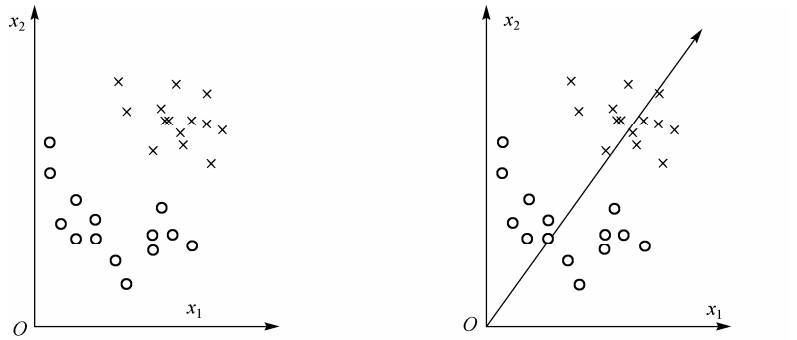
还有一些其他的决策树设计方法，如二叉树，它们用多元逐步回归选择特征，并在对应的两个分支上类别数相等的条件下使余数最小。

3.6 Fisher线性判别函数

在应用统计模式识别方法时，经常会遇到“维数灾难”问题，在低维空间里适用的方法在高维空间可能完全不适用。于是人们开发了一些降低特征空间维数的方法，Fisher 线性判别方法就是其中之一。

把多维特征空间的所有点投影到一条过原点的直线上，就能把特征维数压缩到 1。这在数学上是很容易办到的。但是，在高维空间中很容易分开的样本，把它们投影到任意一条直线上，不同类别的样本可能混杂在一起，无法区分，如图3.11(a)所示两类二维模式的分布，它们的投影无论在 x_1 或 x_2 轴上都是混杂的，因此单纯取它们在 x_1 或 x_2 轴上的投影不容易分类。但是如果把直线绕原点转动一下，就有可能找到一个方向，使这些样本点的投影能很好地区分开，如图3.11(b)所示。因此直线方向的选择很重要。如何找到最好的直线方向使样本

在这条直线上的投影很容易分开，以及如何实现最好方向投影的变换，正是 Fisher 算法要解决的基本问题，这个投影变换正是我们所寻求的解向量 $\boldsymbol{w}^*$ [6]。



(a) 样本投影到x_1或x_2无法分类　(b) 绕原点转动找到一个方向投影样本可分

图 3.11　Fisher 线性判别原理示意图

设给定两类模式样本集 ω_1 和 ω_2，它们各有 n_1 和 n_2 个 d 维样本。我们的目标就是找到这样一条直线，使得模式样本在这条直线上的投影最有利于分类。设 $\boldsymbol{w}$ 为这条直线正方向的单位向量。于是 ω_1 和 ω_2 对直线的投影组成集合 Y_1 和 Y_2，每个 $\boldsymbol{y} \in Y_i$ 就是 $\boldsymbol{x} \in \omega_i$ 在单位向量 $\boldsymbol{w}$ 上的投影。从而有

$$\boldsymbol{y} = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} \tag{3.42}$$

为了找到最有利于分类的投影方向，需要建立一个能反映不同类别模式在这条直线上投影分离程度好坏的准则函数。

为了使样本容易分类，应使各类模式投影均值彼此间相距尽可能大。设 $\boldsymbol{\mu}_i$ 是各类样本的均值向量，n_i 是 ω_i 类的样本个数，

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x \in \omega_i} \boldsymbol{x}, \quad i = 1, 2 \tag{3.43}$$

则这些样本在直线 $\boldsymbol{w}$ 上的投影均值是

$$\tilde{\mu}_i = \frac{1}{n_i} \sum_{\boldsymbol{y} \in Y_i} \boldsymbol{y} = \frac{1}{n_i} \sum_{\boldsymbol{x} \in \omega_i} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} = \boldsymbol{w}^{\mathrm{T}} \mu_i \ , \ i = 1, 2 \tag{3.44}$$

而投影均值差为

$$\tilde{\mu}_1 - \tilde{\mu}_2 \tag{3.45}$$

为了得到更好的分类效果，应该使投影均值差即式(3.45)尽量大。

为了使类别分类效果好，还应该使同类模式的投影尽量密集。我们可以用类内离散度来度量这个密集程度。定义一类模式投影的类内离散度为

$$\tilde{S}_i^2 = \sum_{\boldsymbol{y} \in Y_i} (\boldsymbol{y} - \tilde{\mu}_i)^2 \tag{3.46}$$

则总的类内离散度为

$$\tilde{S}_1^2 + \tilde{S}_2^2 \tag{3.47}$$

它代表整个样本集合中各类样本投影的密集程度。为了得到更好的分类结果，应选择直线 $\boldsymbol{w}$ 使得类内总离散度尽可能小。

综合以上讨论，我们定义 Fisher 准则函数为

$$J(\boldsymbol{w})=\frac{(\tilde{\mu}_1-\tilde{\mu}_2)^2}{\tilde{S}_1^2+\tilde{S}_2^2} \tag{3.48}$$

显然应该寻找使 $J(\boldsymbol{w})$ 的分子尽可能大，分母尽可能小，也就是使 $J(\boldsymbol{w})$ 尽可能大的 $\boldsymbol{w}$ 作为投影方向。

通过推导变换（具体推导过程参见文献[12]），解得使 $J(\boldsymbol{w})$ 取得最大值的 $\boldsymbol{w}^*$ 为

$$\boldsymbol{w}^*=\boldsymbol{S}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) \tag{3.49}$$

其中 $\boldsymbol{S}=\boldsymbol{S}_1+\boldsymbol{S}_2$，$\boldsymbol{S}_i=\sum_{\boldsymbol{x}\in\omega_i}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}, i=1,2$ 为第 i 类的离散度矩阵。$\boldsymbol{w}^*$ 就是 d 维 X 空间到一维 Y 空间的最好投影方向。有了 $\boldsymbol{w}^*$，利用式(3.42)，就可以把 d 维样本 $\boldsymbol{x}_i$ 投影到一维，即直线 $\boldsymbol{w}^*$ 上，变成一维样本 $\boldsymbol{y}_i$。这样 d 维分类问题就转化为一维分类问题了。

现在只要确定阈值 $\boldsymbol{y}_0$，将投影 $\boldsymbol{y}_i$ 与 $\boldsymbol{y}_0$ 相比较，就可以做出决策。$\boldsymbol{y}_0$ 的选取有不同的方案，比较常用的三种分别是

$$\boldsymbol{y}_0=\frac{\tilde{\mu}_1+\tilde{\mu}_2}{2} \tag{3.50}$$

$$\boldsymbol{y}_0=\frac{n_1\tilde{\mu}_1+n_2\tilde{\mu}_2}{n_1+n_2}=\tilde{\mu} \tag{3.51}$$

$$\boldsymbol{y}_0=\frac{\tilde{\mu}_1+\tilde{\mu}_2}{2}+\frac{\ln(P(\omega_1)/P(\omega_2))}{n_1+n_2-2} \tag{3.52}$$

其中 $P(\omega_1)$ 和 $P(\omega_2)$ 分别为 ω_1 类和 ω_2 类样本的先验概率[7]。

这样，对于任意给定的未知样本 $\boldsymbol{x}$，只要计算它的投影点 $\boldsymbol{y}$，$\boldsymbol{y}=\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}$，再根据决策规则

$$\begin{cases}\boldsymbol{y}>\boldsymbol{y}_0 & \Rightarrow \boldsymbol{x}\in\omega_1\\ \boldsymbol{y}<\boldsymbol{y}_0 & \Rightarrow \boldsymbol{x}\in\omega_2\end{cases} \tag{3.53}$$

进行分类，就可以判别出样本 $\boldsymbol{x}$ 的类别。

习题 3

3.1 设五维空间的线性方程为 $55x_1+68x_2+32x_3+16x_4+26x_5+10=0$，试求出其权向量与样本向量点积的表达式 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{w}_0=0$ 中的 $\boldsymbol{w}$ 和 $\boldsymbol{x}$，以及相应的增广权向量与增广特征向量。

3.2 给出一组三类问题的判别函数：

$$g_1(\boldsymbol{x})=-x_1,\quad g_2(\boldsymbol{x})=x_1+x_2-1,\quad g_3(\boldsymbol{x})=x_1-x_2-1$$

① 假设每一模式类与其他模式类之间可用单个判别平面分隔；

② 每两类模式之间都可分别用判别平面分隔开，且

$$g_{12}(\boldsymbol{x})=g_1(\boldsymbol{x}),\quad g_{13}(\boldsymbol{x})=g_2(\boldsymbol{x}),\quad g_{23}(\boldsymbol{x})=g_3(\boldsymbol{x})$$

对于以上两种情况，分别求出每类的判别边界和区域。

3.3　设两类样本的类内离散矩阵分别为 $\boldsymbol{S}_1=\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}$，$\boldsymbol{S}_2=\begin{bmatrix}1 & -1/2\\ -1/2 & 1\end{bmatrix}$，各类样本均值分别为 $\boldsymbol{\mu}_1=(2,\ 0)^{\mathrm{T}}$，$\boldsymbol{\mu}_2=(2,\ 2)^{\mathrm{T}}$，试用 fisher 准则求其决策面方程。

参考文献

[1] 边肇祺等．模式识别．第二版．北京：清华大学出版社，1999: 83, 112, 114.

[2] 钟珞等．模式识别．武汉：武汉大学出版社，2006: 53.

[3] 蔡元龙．模式识别．西安：西安电子科技大学出版社，1992: 37, 38.

[4] 舒宁等．模式识别的理论与方法．武汉：武汉大学出版社，2004: 41, 47, 50.

[5] Richard O. Duda, Peter E. Hart, David G. Stork 著，李宏东等译．模式分类．北京：机械工业出版社，2003:184.

[6] 模式识别与智能计算：MATLAB 技术实现. 北京：电子工业出版社，2008: 111.

[7] 温熙森等．模式识别与状态监控．长沙：国防科技大学出版社，1997: 213.

[8] 李金宗．模式识别导论．北京：高等教育出版社，1994: 146.

[9] 干晓蓉．模式识别．昆明: 云南人民出版社，2006.

[10] 关新平等．信号处理与模式识别．北京：机械工业出版社，2006.

[11] 李兰友，杨淑莹．图像模式识别：VC++技术实现．北京：清华大学出版社，北京交通大学出版社，2005.

[12] 杨光正等．模式识别．合肥：中国科技技术大学出版社，2007: 25.

第 4 章　模式特征提取与选择

在一个较完善的模式识别系统中，或者明显或者隐含地有特征提取与选择的环节，通常其处于对象特征数据采集和分类识别两个环节之间，特征提取与选择方法的优劣极大地影响着分类器的设计和性能，它是模式识别三大核心问题之一。

通常能描述对象的元素很多，为了节约资源，节省计算机存储空间，提高运行速度，有时更是为了可行性，需要进行特征提取与选择。

特征提取：原始特征的数量可能很大，或者说样本处于一个高维空间中，通过映射(或变换)的方法可以用低维空间来表示样本，这个过程称为特征提取。映射后的特征称为二次特征，它们是原始特征的某种组合(通常是线性组合)。所谓特征提取，在广义上就是指一种变换。若 Y 是测量空间，X 是特征空间，则变换 $A:Y \to X$ 就称为特征提取器[1]。

特征选择：从一组特征中挑选出一些最有效的特征以达到降低特征空间维数的目的，这个过程称为特征选择[1]。

特征提取与选择的基本任务是，在保证分类识别正确率的条件下，研究如何从众多特征中选择出那些对分类识别最有效、数量最少的特征，从而实现特征空间维数的压缩。在具体实施特征提取与选择时，可以在一定的准则下从 n 维特征中选取 m 维特征来反映原来的模式，即直接选择法。但是直接选择法简单删去的 $n-m$ 维的特征不一定是无用的信息，这种方法总是不十分理想。因为一般来说，原来的 n 个数据各自在不同程度上反映识别对象的某些特征，简单删去可能会丢失较多的有用信息。如果将原来的特征做变换，获得的每个数据都是原来 n 个数据的线性组合，得到新的变换过的模式，这样可以保证在信息损失最小的情况下获得有利于分类的特征。离散 K-L 变换、离散傅里叶变换、正弦余弦变换和小波变换都是常用的变换方法。

本章分别介绍几种特征提取和选择的相关理论和方法。

4.1　离散K-L变换

利用正交变换将原始特征变换为新模式，可以更多地保留所有特征提供的分类信息，以获得有利于分类的特征。本节介绍常用的正交变换：离散 K-L(Karhunen-Loeve)变换，又称主成分分析，它是一种基于目标统计特性的正交变换。它具有如下重要且优良的性质：使变换后产生的新分量正交或不相关；以部分新的分量表示原向量使得均方误差最小；使变换向量更趋确定、能量更趋集中等，它适用于任意概率密度函数[2]。这些性质使离散 K-L 变换在特征提取、数据压缩等方面都有着极为重要的应用。例如，可以用在汽车车牌字符识别中[14]、路标识别中[15]以及人脸自动识别中[1]。

4.1.1　离散K-L展开式

假设 $\boldsymbol{x}$ 为 n 维的随机向量，$\boldsymbol{x}$ 可以用 n 个基向量的加权和来表示：

$$\boldsymbol{x}=\sum_{j=1}^{n}a_j\boldsymbol{\varphi}_j \tag{4.1}$$

式中，$\boldsymbol{\varphi}_j$ 为基向量，a_j 为加权系数。

式(4.1)可以写成矩阵形式表示为

$$\boldsymbol{x}=(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\mathrm{L}\ ,\boldsymbol{\varphi}_n)\begin{bmatrix}a_1\\a_2\\\mathrm{M}\\a_n\end{bmatrix}=\boldsymbol{\Phi a} \tag{4.2}$$

其中 $\boldsymbol{\Phi}=(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\cdots,\boldsymbol{\varphi}_n)$，$\boldsymbol{a}=(a_1,a_2,\cdots,a_n)^{\mathrm{T}}$。取基向量为正交向量，即

$$\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\varphi}_k=\begin{cases}1, & j=k\\0, & j\neq k\end{cases} \tag{4.3}$$

$\boldsymbol{\Phi}$ 由正交向量构成，所以为正交矩阵，即

$$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}=\boldsymbol{I} \tag{4.4}$$

将式(4.2)两边左乘 $\boldsymbol{\Phi}^{\mathrm{T}}$，并考虑到 $\boldsymbol{\Phi}$ 为正交矩阵，得

$$\boldsymbol{a}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x} \tag{4.5}$$

即

$$\boldsymbol{\alpha}_j=\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{x},\ j=1,2,\mathrm{L}\ ,n \tag{4.6}$$

K-L 展开式的根本性质是将随机向量 $\boldsymbol{x}$ 展开为另一组正交向量 $\boldsymbol{\varphi}_j$ 的线性和，且其展开式系数 a_j（即系数向量 $\boldsymbol{a}$ 的各个分量）具有不相关的性质。那么在上述条件下怎样确定正交向量集 $\{\boldsymbol{\varphi}_j\}$ 呢？

设随机向量 $\boldsymbol{x}$ 的总体自相关矩阵为

$$\boldsymbol{R}=E\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\} \tag{4.7}$$

将式(4.2)代入式(4.7)，有

$$\boldsymbol{R}=E\{\boldsymbol{\Phi}\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\}=\boldsymbol{\Phi}E\{\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}\}\boldsymbol{\Phi}^{\mathrm{T}} \tag{4.8}$$

我们希望向量 $\boldsymbol{a}$ 的各分量互不相关，即应使 $(a_1\mathrm{L}\ a_j\mathrm{L}\ a_n)$ 满足

$$E\{a_ja_k\}=\begin{cases}\lambda_j, & j=k\\0, & j\neq k\end{cases} \tag{4.9}$$

写成矩阵形式，应使

$$E\{\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}\}=\begin{bmatrix}\lambda_1 & & & 0\\ & \mathrm{O} & & \\ & & \lambda_j & \\ & & & \mathrm{O}\\ 0 & & & \lambda_n\end{bmatrix}=\boldsymbol{D}_{\lambda} \tag{4.10}$$

则 $\boldsymbol{R}=\boldsymbol{\Phi}\boldsymbol{D}_{\lambda}\boldsymbol{\Phi}^{\mathrm{T}}$。

因为 $\boldsymbol{\Phi}$ 为正交矩阵，上式两边右乘 $\boldsymbol{\Phi}$，有

$$\boldsymbol{R}\boldsymbol{\Phi}=\boldsymbol{\Phi}\boldsymbol{D}_{\lambda}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}=\boldsymbol{\Phi}\boldsymbol{D}_{\lambda} \tag{4.11}$$

即

$$\boldsymbol{R}\varphi_j=\boldsymbol{\lambda}_j\boldsymbol{\varphi}_j,\ j=1,2,\mathrm{L}\ ,n \tag{4.12}$$

可见 $\boldsymbol{\Phi}$ 实际上是由矩阵 $\boldsymbol{R}$ 的本征向量组成的，$\boldsymbol{\varphi}_j$ 对应自相关矩阵 $\boldsymbol{R}$ 的本征向量，$\boldsymbol{\lambda}_j$ 是自相关矩阵 $\boldsymbol{R}$ 的本征值。

所以当需要写出 $\boldsymbol{x}$ 的 K-L 展开式时，展开式系数计算步骤如下。

(1) 首先计算 $\boldsymbol{x}$ 的自相关矩阵 $\boldsymbol{R}$。

(2) 然后求 $\boldsymbol{R}$ 的本征向量 $\boldsymbol{\varphi}_j$，$j=1,2,\mathrm{L}\ ,n$，得到矩阵 $\boldsymbol{\Phi}=(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\mathrm{L}\ ,\boldsymbol{\varphi}_n)$。

(3) 最后由式(4.5)可以确定展开式系数 $\boldsymbol{a}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}$。

4.1.2 基于K-L变换的数据压缩

K-L 展开式用于特征选择时相当于一种线性交换。如果从自相关矩阵 $\boldsymbol{R}$ 的 n 个本征向量中取出 m 个组成变换矩阵 $\boldsymbol{\Phi}$，即

$$\boldsymbol{\Phi}=(\boldsymbol{\varphi}_1\boldsymbol{\varphi}_2\mathrm{L}\ \boldsymbol{\varphi}_m),\ m<n \tag{4.13}$$

这时 $\boldsymbol{\Phi}$ 是一个 $n\times m$ 矩阵。令 $\boldsymbol{x}$ 为 n 维向量，经过 $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}$ 变换，得到降维为 m 的新模式。现在的问题是怎样选取变换矩阵 $\boldsymbol{\Phi}$，使降维的新模式在最小均方误差的条件下接近原来的向量 $\boldsymbol{x}$。

对于式(4.1)，取其中 m 项，对略去的项用常数 b 代替，这时对 $\boldsymbol{x}$ 的估计值为

$$\hat{\boldsymbol{x}}=\sum_{j=1}^{m}a_j\boldsymbol{\varphi}_j+\sum_{j=m+1}^{n}b\boldsymbol{\varphi}_j \tag{4.14}$$

则产生的误差为

$$\Delta\boldsymbol{x}=\boldsymbol{x}-\hat{\boldsymbol{x}}=\sum_{j=m+1}^{n}(a_j-b)\boldsymbol{\varphi}_j \tag{4.15}$$

$\Delta\boldsymbol{x}$ 的均方误差为

$$\bar{\varepsilon}^2=E\left\{\|\Delta\boldsymbol{x}\|^2\right\}=\sum_{j=m+1}^{n}E\left\{(a_j-b)^2\right\} \tag{4.16}$$

要使 $\bar{\varepsilon}^2$ 最小，对 b 的选择应满足

$$\begin{aligned}\frac{\partial}{\partial b}\left[E(a_j-b)^2\right]&=\frac{\partial}{\partial b}\left[E(a_j^2-2a_jb+b^2)\right]\\&=-2\left[E(a_j)-b\right]\\&=0\end{aligned}\tag{4.17}$$

所以

$$b=E[a_j]\tag{4.18}$$

也就是说，对于省略掉的那些分量，可以用它们的期望值来代替，这时的误差为

$$\begin{aligned}\bar{\varepsilon}^2&=\sum_{j=m+1}^{n}E\left[(a_j-E\{a_j\})^2\right]\\&=\sum_{j=m+1}^{n}\boldsymbol{\varphi}_j^{\mathrm{T}}E\left[(\boldsymbol{x}-E\{\boldsymbol{x}\})(\boldsymbol{x}-E\{\boldsymbol{x}\})^{\mathrm{T}}\right]\boldsymbol{\varphi}_j\\&=\sum_{j=m+1}^{n}\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\Sigma}_x\boldsymbol{\varphi}_j\end{aligned}\tag{4.19}$$

其中 $\boldsymbol{\Sigma}_x$ 是 $\boldsymbol{x}$ 的协方差矩阵。

现在，我们还需要确定 $\boldsymbol{\varphi}_j$ 以使 $\bar{\varepsilon}^2$ 最小。由于 $\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\varphi}_j=1$，利用拉格朗日乘数法求 $\boldsymbol{\varphi}_j$，使 $\bar{\varepsilon}^2$ 最小：

$$\bar{\varepsilon}_0^{\,2}=\bar{\varepsilon}^2-\sum_{j=m+1}^{n}\lambda_j\left[\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\varphi}_j-1\right]=\sum_{j=m+1}^{n}\left[\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\Sigma}_x\boldsymbol{\varphi}_j-\lambda_j(\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\varphi}_j-1)\right]$$

式中，λ_j 为拉格朗日乘数。利用二次梯度运算公式，可得

$$\nabla_{\boldsymbol{\varphi}}[\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Sigma}_x\boldsymbol{\Phi}]=2\boldsymbol{\Sigma}_x\boldsymbol{\Phi}\,,\quad \nabla_{\boldsymbol{\varphi}_j}[\bar{\varepsilon}_0^2]=2\boldsymbol{\Sigma}_x\boldsymbol{\varphi}_j-2\lambda_j\boldsymbol{\varphi}_j=0$$

即 $\boldsymbol{\Sigma}_x\boldsymbol{\varphi}_j=\lambda_j\boldsymbol{\varphi}_j$。这说明 λ_j 是协方差矩阵 $\boldsymbol{\Sigma}_x$ 的第 j 个本征值，而 $\boldsymbol{\varphi}_j$ 是与 λ_j 对应的本征向量。将上式代入 $\bar{\varepsilon}^2$，则得最小均方误差为

$$\bar{\varepsilon}_0^2=\sum_{j=m+1}^{n}\boldsymbol{\varphi}_j^{\mathrm{T}}\boldsymbol{\Sigma}_x\boldsymbol{\varphi}_j=\sum_{j=m+1}^{n}\lambda_j\tag{4.20}$$

可见，λ_j 越小，误差也越小。

从以上分析可得出以下结论。

(1) 按照 K-L 展开式的性质和最小均方差的准则来选择特征，应使式(4.14)中的 $b=0$。由式(4.18)可知，相当于 $E[a_j]=0$。因为 $E[\boldsymbol{a}]=E[\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}]=\boldsymbol{\Phi}^{\mathrm{T}}E[\boldsymbol{x}]$，故应使 $E[\boldsymbol{x}]=0$。正因为这个条件，在将模式总体做 K-L 变换前，先将其均值作为新坐标轴的原点。

(2) 计算自相关矩阵 $\boldsymbol{R}$，求出 $\boldsymbol{R}$ 的本征值 λ_j，$j=1,2,\cdots,n$ 及其对应的本征向量 $\boldsymbol{\varphi}_j$，$j=1,2,\cdots,n$。由式(4.20)可看出，为使误差最小，不采用的本征向量对应的本征值应尽可能小。因此，将本征值按照大小次序编号，即

$$\lambda_1>\lambda_2>\mathrm{L}\ >\lambda_m>\mathrm{L}\ >\lambda_n\geq\ 0$$

取前 m 个大的本征值对应的本征向量构成变换矩阵为

$$\boldsymbol{\Phi}=(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\mathrm{L}\ ,\boldsymbol{\varphi}_m)$$

(3)将 n 维的原向量变换成 m 维新向量：

$$\boldsymbol{y}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x} \tag{4.21}$$

K-L 变换是在均方误差最小的意义下获得数据压缩的最佳变换，且不受模式分布的限制。对于一种类别的模式特征提取，显然它不存在分类问题，只是怎样用低维 m 个特征来表示原来高维 n 个特征，使其误差最小，也就是使其整个模式分布结构尽可能保持不变。

【例 4.1】[23] 给出样本数据如下：

$$\begin{bmatrix}-5\\-5\end{bmatrix},\begin{bmatrix}-5\\-4\end{bmatrix},\begin{bmatrix}-4\\-5\end{bmatrix},\begin{bmatrix}-5\\-6\end{bmatrix},\begin{bmatrix}-6\\-5\end{bmatrix},\begin{bmatrix}5\\5\end{bmatrix},\begin{bmatrix}5\\6\end{bmatrix},\begin{bmatrix}6\\5\end{bmatrix},\begin{bmatrix}5\\4\end{bmatrix},\begin{bmatrix}4\\5\end{bmatrix}$$

试用 K-L 变换做一维数据压缩。

解：

(1)求样本总体均值向量

$$\boldsymbol{\mu}=\frac{1}{10}\left[\begin{bmatrix}-5\\-5\end{bmatrix}+\begin{bmatrix}-5\\-4\end{bmatrix}+\mathrm{L}\ +\begin{bmatrix}4\\5\end{bmatrix}\right]=\begin{bmatrix}0\\0\end{bmatrix}$$，不用做坐标平移。

(2)求自相关矩阵

$$\boldsymbol{R}=\frac{1}{10}\left[\begin{bmatrix}-5\\-5\end{bmatrix}(-5,-5)+\mathrm{L}\ +\begin{bmatrix}4\\5\end{bmatrix}(4,5)\right]=\begin{bmatrix}25.4 & 25.0\\25.0 & 25.4\end{bmatrix}$$

(3)求本征值和本征向量。解本征值方程

$$\begin{vmatrix}25.4-\lambda & 25.0\\25.0 & 25.4-\lambda\end{vmatrix}=0$$

得 $\lambda_1=50.4,\lambda_2=0.4$, 由 $\boldsymbol{R}\boldsymbol{\varphi}_j=\boldsymbol{\lambda}_j\boldsymbol{\varphi}_j$ 得到本征向量为

$$\boldsymbol{\varphi}_1=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix},\boldsymbol{\varphi}_2=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

(4)因为要求做一维压缩，所以取 $\boldsymbol{\varphi}_1=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$ 作为变换矩阵 $\boldsymbol{\Phi}$，由 $\boldsymbol{y}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}$ 将原样本变换为新的样本：

$$\left(-\frac{10}{\sqrt{2}}\right),\left(-\frac{9}{\sqrt{2}}\right),\left(-\frac{9}{\sqrt{2}}\right),\left(-\frac{11}{\sqrt{2}}\right),\left(-\frac{11}{\sqrt{2}}\right),\left(\frac{10}{\sqrt{2}}\right),\left(\frac{11}{\sqrt{2}}\right),\left(\frac{11}{\sqrt{2}}\right),\left(\frac{9}{\sqrt{2}}\right),\left(\frac{9}{\sqrt{2}}\right)$$

4.1.3 基于K-L变换的特征提取

K-L 变换能在信息损失最小的情况下获得互不相关的新特征。由于这种变换是沿着方差较大的几个方向选择新坐标的轴系，变换的结果突出了差异性，而减小了相关性，当进一步考虑提取有利于鉴别分类的特征时，这一变换特性就成为分析的依据。

在进行K-L 变换时，可以根据不同的散布矩阵求本征向量，以构成变换矩阵，而这样做的结果对提取鉴别各类模式的信息具有不同的效应。

(1) 按总体散布矩阵做 K-L 变换

这是把多类模式合并起来视为一个总体分布，按其协方差矩阵做 K-L 展开，采用与大本征根对应的向量组成变换矩阵，使降维后的模式能在均方差最小的条件下逼近原来的模式。把这种散布矩阵称为总体散布矩阵 $\boldsymbol{S}_t$，即

$$\boldsymbol{S}_t = E\{(\boldsymbol{x}-\boldsymbol{\mu}_0)(\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\} \quad \boldsymbol{x}\in\forall\omega_i;\quad i=1,2,\mathrm{L}\ ,c \tag{4.22}$$

采用总体散布矩阵能保留模式原有分布的主要结构。如果原来的多类模式在总体分布上存在可分性好的特征，用总体散布矩阵的 K-L 变换能尽量多地保留可分性信息。

(2) 按类内散布矩阵做 K-L 变换

采用类内散布矩阵 $\boldsymbol{S}_w$ 做 K-L 变换，即

$$\boldsymbol{S}_w = \sum_{i=1}^{c} p(\omega_i)E\{(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\},\ \ x\in\omega_i \tag{4.23}$$

它等于各类模式的协方差矩阵之和，为了突出各类模式的主要特征分量，可选用对应于大本征根的本征向量组成变换矩阵；反之，为使同一类模式能聚类于最小的特征空间范围，也可选用对应于小本征根的本征向量组成变换矩阵。

图 4.1 显示了不同形式的两类模式分布。如采用类内散布矩阵做 K-L 变换，则图 4.1(a) 宜采用小本征根向量组成变换矩阵，提取一维的可分特征；图 4.1(b) 宜采用大本征根向量组成变换矩阵；图4.1(c) 用小本征根还是大本征根的向量，需要试探一下；图4.1(d) 所求的类内散布矩阵，其大小本征根的差异不明显，需要用其他散布矩阵来确定变换矩阵。从以上四种模式分布的情况分析，类内散布矩阵适用于各类模式的分布都比较相似且某一维特征分量的可分性较好的场合[3]。

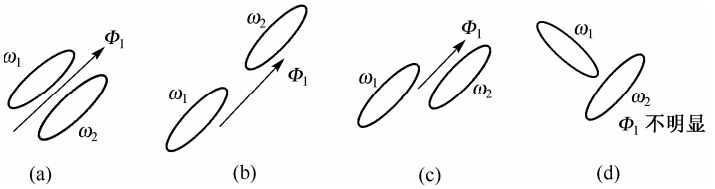


图 4.1　两类模式的不同分布形式与广义 K-L 变换

(3) 按类间散布矩阵做 K-L 变换

为了强调不同类别之间的差异，类别之间的平均距离是重要的指标，因此可以利用类间散布矩阵

$$\boldsymbol{S}_B = \sum_{i=1}^{c} p(\omega_i)(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)^{\mathrm{T}} \tag{4.24}$$

做 K-L 变换进行特征提取。式中，$\boldsymbol{\mu}_0$ 为 c 类模式总体的均值向量。

$\boldsymbol{S}_B$ 由不大于 $c-1$ 个独立向量组成，只有 $c-1$ 个非零本征值。通常，模式维数 n 大于类别数 c，所以 $\boldsymbol{S}_B$ 虽也是对称正定的，但是为奇异矩阵。同样，求出 $\boldsymbol{S}_B$ 的本征根，排列成

$\lambda_1 > \lambda_2 > \mathrm{L} > \lambda_{c-1}$，并且 $\lambda_c, \mathrm{L}, \lambda_n = 0$，选出 m 个与大本征根对应的本征向量组成变换矩阵。

一般说来，类间距离比类内距离大得多的多类问题，采用类间散布矩阵比较合适，例如图4.1(d)所示的分布用类间散布矩阵就比较合适。

4.2 离散傅里叶变换

离散傅里叶变换(Discrete Fourier Transform, DFT)建立了离散时域与离散频域之间的关系，使时域和频域有关的计算都限制在有限空间中。如果信号直接在时域上进行，则计算量大，且随着样点数目的增加而急剧增加，难以实时处理。而采用离散傅里叶变换的方法，将输入的数字信号首先做离散傅里叶变换，把时域中的卷积或相关运算简化为频域的相乘处理，然后再做离散傅里叶逆变换，恢复为时域信号，可以大大减少计算量，提高处理速度。另外，DFT 还有一个明显的优点就是存在快速算法，即 FFT 算法。FFT 算法极大地提高了 DFT 的计算速度，随样点数的增加，其优越性愈加显著[4]。

因为许多书籍和资料都对离散傅里叶变换有详细的介绍，所以这里只简单的介绍一下它的定义和性质。

4.2.1 一维离散傅里叶变换

给定 N 个输入样本 $f(0), f(1), \mathrm{L}, f(N-1)$，其离散傅里叶变换(DFT)产生一个包含 N 个样本的新的样本集 $F(u)$，定义为

$$F(u) = \sum_{k=0}^{N-1} f(k) \exp\left(-\mathrm{j}\frac{2\pi}{N}ku\right) = \sum_{k=0}^{N-1} f(k) W_N^{ku}, u = 0, 1, \mathrm{L}, N-1, \mathrm{j} = \sqrt{-1} \tag{4.25}$$

$F(u), u = 0, 1, \mathrm{L}, N-1$ 的离散傅里叶逆变换(IDFT)定义为

$$f(k) = \frac{1}{N}\sum_{u=0}^{N-1} F(u) \exp\left(\mathrm{j}\frac{2\pi}{N}ku\right) = \frac{1}{N}\sum_{u=0}^{N-1} F(u) W_N^{-ku}, k = 0, 1, \mathrm{L}, N-1 \tag{4.26}$$

式中，

$$W_N \equiv \exp\left(-\mathrm{j}\frac{2\pi}{N}\right) \tag{4.27}$$

DFT 具有很多性质，这里介绍几种常用的性质。

(1)线性性质

函数 $af_1(k) + bf_2(k)$ 满足

$$af_1(k) + bf_2(k) \Leftrightarrow aF_1(u) + bF_2(u) \tag{4.28}$$

(2)时移性质

如果序列 $f(k)$ 向右(或向左)移动 i 位，则有

$$f(k-i) \Leftrightarrow F(u) W_N^{ui} \tag{4.29}$$

即位移后的 DFT 是位移前的 DFT 与一个指数相乘。

(3)频移性质

对任意实整数，有

$$f(k)W_N^{-ki} \Leftrightarrow F(u-i) \tag{4.30}$$

即将 $f(k)$ 与一指数相乘，相当于其变换后的频域中心移动到新的位置。

(4)时间卷积定理

离散卷积的定义为

$$y(k) = f(k) * g(k) = \sum_{i=0}^{N-1} f(i)g(k-i) \tag{4.31}$$

其中 $f(k)$ 和 $g(k)$ 是具有相同周期 N 的周期函数。如果 $f(k)$ 和 $g(k)$ 的 DFT 分别为 $F(u)$ 和 $G(u)$，则离散卷积的 DFT 为

$$y(k) = f(k) * g(k) \Leftrightarrow F(u)G(u) \tag{4.32}$$

(5)频率卷积定理

频率卷积为

$$Y(u) = \sum_{i=0}^{N-1} F(i)G(u-i) = F(u) * G(u) \tag{4.33}$$

因为 $F(u)$ 和 $G(u)$ 是周期性的，所以上式是在频率平面中的一个循环卷积。此表达式的逆 DFT 为

$$Y(u) = F(u) * G(u) \Leftrightarrow f(k) * g(k) \tag{4.34}$$

4.2.2　二维离散傅里叶变换

设二维离散信号为 $\{f(x,y) \mid x = 0,1,\mathrm{L}\ ,M-1; y = 0,1,\mathrm{L}\ ,N-1\}$，则其二维 DFT 定义为

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$
$$u = 0,1,\mathrm{L}\ ,M-1; \quad v = 0,1,\mathrm{L}\ ,N-1 \tag{4.35}$$

其逆傅里叶变换(IDFT)为

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\exp\left[\mathrm{j}2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$
$$x = 0,1,\mathrm{L}\ ,M-1; y = 0,1,\mathrm{L}\ ,N-1 \tag{4.36}$$

如果要处理的图像信号为方阵，即 $M = N$，则 DFT 变换可以简化为

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\exp\left(-\mathrm{j}2\pi\frac{ux+vy}{N}\right) \tag{4.37}$$

$$f(x,y) = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)\exp\left(\mathrm{j}2\pi\frac{ux+vy}{N}\right) \tag{4.38}$$

式中 $u,v,x,y = 0,1,\mathrm{L}\ ,N-1$。

二维 DFT 除了具有一维 DFT 变换的性质外，还具有以下性质[4]。

(1)变换的可分离性

二维 DFT 正反变换都可以分为两次一维 DFT 运算：

$$F(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\exp\left(-\mathrm{j}2\pi\frac{ux+vy}{N}\right)$$

$$=\sum_{x=0}^{N-1}\left[\sum_{y=0}^{N-1}f(x,y)\exp\left(-\mathrm{j}2\pi\frac{vy}{N}\right)\right]\exp\left(-\mathrm{j}2\pi\frac{ux}{N}\right),\qquad u,v=0,1,\mathrm{L}\ ,N-1 \tag{4.39}$$

$$f(x,y)=\frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}F(u,v)\exp\left(\mathrm{j}2\pi\frac{ux+vy}{N}\right)$$

$$=\frac{1}{N}\left[\sum_{u=0}^{N-1}\frac{1}{N}\sum_{v=0}^{N-1}F(u,v)\exp\left(\mathrm{j}2\pi\frac{vy}{N}\right)\right]\exp\left(\mathrm{j}2\pi\frac{ux}{N}\right),\qquad x,y=0,1,\mathrm{L}\ ,N-1 \tag{4.40}$$

上式分离后，二维 DFT 就分解为水平和垂直两部分运算。上式中方括号中的项表示在图像的行上计算 DFT，方括号外的求和则为数组在列上的 DFT。所以，二维 DFT 就分成了两个一维 DFT 来实现。

(2) 旋转不变性

引入极坐标，使

$$\begin{cases}x=r\cos\theta\\ y=r\sin\theta\end{cases},\qquad \begin{cases}u=\omega\cos\varphi\\ v=\omega\sin\varphi\end{cases}$$

则 $f(x,y)$ 和 $F(u,v)$ 分别表示为 $f(r,\theta)$和$F(\omega,\varphi)$。在极坐标中，存在以下变换对：

$$f(r,\theta+\theta_0)\Leftrightarrow F(\omega,\varphi+\theta_0) \tag{4.41}$$

这表明，如果图像旋转一个角度θ_0，则它的频率也旋转同样的角度。

(3) 去相关性

当输入的像素高度相关时，变换系数趋于不相关。变换系数的协方差矩阵中的对角元素值比非对角元素值大得多。对于一个信号来说，如果它的各个分量之间完全不相关，则表示该数据中没有冗余。因此正交变换的去相关性有利于图像数据的压缩。

(4) 熵保持性

如果把 $f(x,y)$ 视为一个具有一定熵值的随机函数，那么变换函数 $F(u,v)$ 的熵值和原来图像信号 $f(x,y)$ 的熵值相等。

【例 4.2】 图像的二维离散傅里叶变换。

图 4.2(a)所示为一幅原始图像，图 4.2(b)所示为该图像的离散傅里叶频谱。在图 4.2(b)中可以看到图像的低频能量都集中在中心部分，而高频能量则集中在四周，这样就便于以后对图像频谱进行各种处理(如滤波、降噪等)。

在图像处理的广泛领域中，傅里叶变换起着非常重要的作用，包括图像的效果增强、图像分析、图像复原和图像压缩等。在图像数据的数字处理中常用的是二维离散傅里叶变换，它能把空间域的图像转变到频率域上进行研究，从而简化处理过程，增强处理效果。



图 4.2　二维图像的傅里叶变换

4.3　离散余弦和正弦变换

在模式识别领域中，除了前面介绍的 DFT 之外，还有许多种离散正交变换被广泛应用，如离散正弦变换(Discrete sine Transform, DST)和离散余弦变换(Discrete Cosine Transform, DCT)就是其中的两种方法。在数字图像处理领域中，DCT 与 DFT 相比有着许多优点，其中最显著的是 DFT 是复数域的计算，尽管借助 FFT 可以提高运算速度，但是在实际应用中特别是在实际处理中，会带来了不便，而离散余弦变换是实数变换[4]。

4.3.1　余弦变换

给定 N 个输入样本 $x(0), x(1), \mathrm{L}\ , x(N-1)$，一维离散余弦变换(DCT)定义为

$$y(k) = a(k)\sum_{n=0}^{N-1} x(n)\cos\left(\frac{\pi(2n+1)k}{2N}\right),\ k = 0,1,\mathrm{L}\ ,N-1 \tag{4.42}$$

逆 DCT 为

$$x(n) = \sum_{k=0}^{N-1} a(k)y(k)\cos\left(\frac{\pi(2n+1)k}{2N}\right),\ n = 0,1,\mathrm{L}\ ,N-1 \tag{4.43}$$

其中，

$$a(k) = \begin{cases} \sqrt{\dfrac{1}{N}}, & k = 0 \\ \sqrt{\dfrac{2}{N}}, & k \neq 0 \end{cases} \tag{4.44}$$

上式写成向量形式为 $\boldsymbol{y} = \boldsymbol{C}^{\mathrm{T}}\boldsymbol{x}$。其中矩阵 $\boldsymbol{C}$ 的元素是

$$c(n,\ k) = 1/\sqrt{N},\ k = 0, 0 \leq\ n \leq\ N-1$$

$$c(n,\ k) = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi(2n+1)k}{2N}\right),\ 1 \leq\ k \leq\ N-1,\ 0 \leq\ n \leq\ N-1$$

余弦变换具有以下性质[5]。

(1)余弦变换是实变换，也是正交变换，即

$$\boldsymbol{C}^{-1}=\boldsymbol{C}^{\mathrm{T}} \tag{4.45}$$

(2)余弦变换是一种快速变换。对包含 N 个元素的向量进行余弦变换，采用 N 点 FFT 时，其计算复杂度为 $O(N\ \mathrm{lb}\ N)$。

(3)余弦变换对于高度相关的数据具有很强的能量压缩能力。这是由下面的性质决定的。

(4)余弦变换的基向量(也就是矩阵 $\boldsymbol{C}$ 的各行)是对称三角阵 $\boldsymbol{Q}_c$ 的特征向量。$\boldsymbol{Q}_c$ 定义如下：

$$\boldsymbol{Q}_c=\begin{bmatrix}1-a & -a & & & 0\\ -a & 1 & \mathrm{O} & & \\ & \mathrm{O} & \mathrm{O} & \mathrm{O} & \\ & & \mathrm{O} & 1 & -a\\ 0 & & & -a & 1-a\end{bmatrix}$$

(5)长度为 N 的一阶平稳马尔可夫序列，当它的相关系数 ρ 接近 1 时，$N\times N$ 的余弦变换与 K-L 变换非常接近。

根据一维 DCT，我们可以推导出二维 DCT。设 $\{f(x,y)\mid x=0,1,\mathrm{L}\ ,M-1;y=0,1,\mathrm{L}\ ,N-1\}$ 为二维信号序列集合，则其二维 DCT 定义为

$$F(u,v)=a(u)a(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cos\left(\frac{\pi(2x+1)u}{2M}\right)\cos\left(\frac{\pi(2y+1)v}{2N}\right) \tag{4.46}$$
$$u=0,1,\mathrm{L}\ ,M-1;v=0,1,\mathrm{L}\ ,N-1$$

其逆 DCT 为

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}a(u)a(v)F(u,v)\cos\left(\frac{\pi(2x+1)u}{2M}\right)\cos\left(\frac{\pi(2y+1)v}{2N}\right) \tag{4.47}$$
$$x=0,1,\mathrm{L}\ ,M-1;\ \ y=0,1,\mathrm{L}\ ,N-1$$

其中，

$$a(u)=\begin{cases}\sqrt{\dfrac{1}{M}}, & u=0\\ \sqrt{\dfrac{2}{M}}, & u\neq 0\end{cases},\quad a(v)=\begin{cases}\sqrt{\dfrac{1}{N}}, & v=0\\ \sqrt{\dfrac{2}{N}}, & v\neq 0\end{cases}$$

从式(4.46)和式(4.47)可以看出，二维 DCT 的正反变换都具有可分离性，因而可通过两次一维变换实现一个二维变换。

【例 4.3】 二维图像及其离散余弦变换频谱。

图4.3(a)所示为一幅原始图像，图4.3(b)所示为该图像的离散余弦变换频谱。在图4.3(b)中可以看出图像的低频能量都集中在左上角区域，而高频能量则集中在右下角区域。与例 4.2 的离散傅里叶频谱图进行比较，可以发现高低频的能量集中在不同的区域，这主要是因为离散傅里叶变换的变换核是复数，而离散余弦变换的变换核实际上是取其实部。

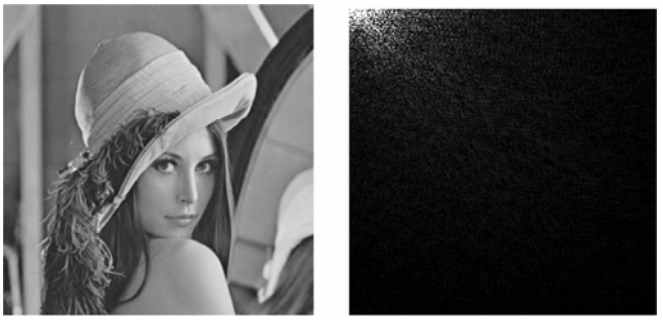


图 4.3　二维图像及其离散余弦变换频谱的显示

离散余弦变换在图像处理中占有重要位置，尤其是在图像的变换编码中有着非常成功的应用。静止图像压缩标准 JPEG 就采用了离散余弦变换。离散余弦变换实际上是傅里叶变换的实数部分，但是它比傅里叶变换有更强的信息集中能力。对于大多数自然图像，离散余弦变换能将大多数的信息放到较少的系数上去，因此能够提高编码的效率。

4.3.2　正弦变换

给定 N 个输入样本 $x(0), x(1), \mathrm{L}\ , x(N-1)$，它们的离散正弦变换(DST)定义为

$$y(k)=\sqrt{\frac{2}{N+1}}\sum_{n=0}^{N-1}x(n)\sin\left(\frac{\pi(n+1)(k+1)}{N+1}\right),\ k=0,1,\mathrm{L}\ ,N-1 \tag{4.48}$$

其逆 DST 为

$$x(n)=\sqrt{\frac{2}{N+1}}\sum_{k=0}^{N-1}y(k)\sin\left(\frac{p(n+1)(k+1)}{N+1}\right),\ n=0,1,\mathrm{L}\ ,N-1 \tag{4.49}$$

上式写成向量形式为 $\boldsymbol{y}=\boldsymbol{S}^{\mathrm{T}}\boldsymbol{x}$。其中矩阵 $\boldsymbol{S}$ 的元素是

$$s(k,\ n)=\sqrt{\frac{2}{N+1}}\sin\left(\frac{\pi(n+1)(k+1)}{N+1}\right),\ k,\ n=0,1,\mathrm{L}\ ,N-1$$

正弦变换具有以下性质[5]。

(1) 正弦变换是实变换、对称变换和正交变换，即

$$\boldsymbol{S}^{*}=\boldsymbol{S}=\boldsymbol{S}^{-1}=\boldsymbol{S}^{\mathrm{T}} \tag{4.50}$$

因此正向和逆向正弦变换是完全一样的。

(2) 正弦变换是一种快速变换。通过 $2(N+1)$ 点 FFT，一个 N 元向量的正弦变换计算复杂度为 $O(N\ \mathrm{lb}\ N)$。

(3) 长度为 N 的一阶平稳马尔可夫序列，当它的相关系数 ρ 属于区间 $(-0.5, 0.5)$ 时，正弦变换非常接近于 K-L 变换。一般来说，它对图像能量的压缩性能可以达到“好”至“非常好”的等级。

4.4 Hadamard变换

下面介绍的 Hadamard 变换和 Haar 变换与前面提到的 DFT、DCT、DST 相比，具有计算方面的优点，它们的单位矩阵由±1 组成，并且变换只是加法和减法运算，没有乘法运算。因此，与那些需要乘法运算的变换相比，节省了处理器运算时间。

Hadamard 变换矩阵 $\boldsymbol{H}_n$ 是 $N \times N$ 矩阵，其中 $N = 2^n$, $n = 1, 2, 3$。可以很容易地由核矩阵

$$\boldsymbol{H}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.51}$$

和 Kronecker 的积递归生成

$$\boldsymbol{H}_n = \boldsymbol{H}_{n-1} \otimes \boldsymbol{H}_1 = \boldsymbol{H}_1 \otimes \boldsymbol{H}_{n-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{H}_{n-1} & \boldsymbol{H}_{n-1} \\ \boldsymbol{H}_{n-1} & \boldsymbol{H}_{n-1} \end{bmatrix} \tag{4.52}$$

以 $n=3$ 为例，Hadamard 矩阵为

$$\boldsymbol{H}_3 = \boldsymbol{H}_1 \otimes \boldsymbol{H}_2$$

$$\boldsymbol{H}_2 = \boldsymbol{H}_1 \otimes \boldsymbol{H}_1$$

由此可得

$$H_3 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$N \times 1$ 维向量 $\boldsymbol{u}$ 的 Hadamard 变换记为

$$\boldsymbol{v} = \boldsymbol{H}\boldsymbol{u} \tag{4.53}$$

反变换为

$$\boldsymbol{u} = \boldsymbol{H}\boldsymbol{v} \tag{4.54}$$

其中 $\boldsymbol{H} = \boldsymbol{H}_n$，$n = \mathrm{lb}\, N$ 。写成级数的形式，变换记为

$$v(k) = \frac{1}{\sqrt{N}}\sum_{m=0}^{N-1} u(m)(-1)^{b(k,m)},\quad 0 \le k \le N-1 \tag{4.55}$$

$$u(m) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} v(k)(-1)^{b(k,m)},\quad 0 \le m \le N-1 \tag{4.56}$$

式中，

$$b(k,m)=\sum_{i=0}^{n-1}k_i m_i;\quad k_i,m_i=0,1$$

式中$\{k_i\}$和$\{m_i\}$分别是 k 和 m 的二进制表示，即

$$k=k_0+2k_1+\mathrm{L}\ +2^{n-1}k_{n-1}$$
$$m=m_0+2m_1+\mathrm{L}\ +2^{n-1}m_{n-1}$$

二维 Hadamard 变换表示为

$$\boldsymbol{Y}=\boldsymbol{H}_n\boldsymbol{X}\boldsymbol{H}_n,\quad \boldsymbol{X}=\boldsymbol{H}_n\boldsymbol{Y}\boldsymbol{H}_n \tag{4.57}$$

Hadamard 变换具有以下性质[5]。

(1) Hadamard 变换是实变换、对称变换和正交变换，即

$$\boldsymbol{H}=\boldsymbol{H}^*=\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{H}^{-1}$$

(2) Hadamard 变换是一种快速变换。式(4.52)给出的一维变换可通过$O\ (N\ \mathrm{lb}\ N)$次加法和减法运算完成。

因为 Hadamard 变换只包含±1 两个值，所以在变换中不需要计算乘法。更进一步，加法和减法的次数还可以由 N^2 次减少到大约 $N\ \mathrm{lb}\ N$ 次。这是因为 $\boldsymbol{H}_n$ 可以写为 n 个稀疏矩阵的乘积，即

$$\boldsymbol{H}=\boldsymbol{H}_n=\prod_{p=1}^{n}\boldsymbol{H}_p^{(n)},\ n=\mathrm{lb}\ N$$

其中，

$$\boldsymbol{H}_p^{(n)}=\frac{1}{\sqrt{2}}\begin{bmatrix}
1 & 1 & 0 & 0 & & & & \\
0 & 0 & 1 & 1 & 0 & 0 & & \\
\mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & & & & \\
0 & 0 & & \mathrm{L} & & & 1 & 1 \\
1 & -1 & 0 & 0 & & & & \\
0 & 0 & 1 & -1 & & & & \\
\mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} & & & & \\
0 & 0 & & \mathrm{L} & & & 1 & -1
\end{bmatrix}$$

因为 $\boldsymbol{H}_p^{(n)}$每行只包含两个非零值，所以变换

$$\boldsymbol{v}=\boldsymbol{H}_n^{(n)}\boldsymbol{u},\ n=\mathrm{lb}\ N$$

可以通过对 u 进行 n 次 $\boldsymbol{H}_p^{(n)}$变换实现。由于 $\boldsymbol{H}_p^{(n)}$的结构，每次对向量进行 $\boldsymbol{H}_p^{(n)}$操作只需要 N 次加法或减法，即总共要进行 $N_n=N\ \mathrm{lb}\ N$ 次加法或减法。

(3) 对于高度相关的图像，Hadamard 变换的能量压缩性能可以达到“好”到“很好”的程度。

【例 4.4】 二维图像的 Hadamard 变换。

图4.4(a)所示为二维原始图像，对其进行 Hadamard 变换后的结果如图4.4(b)所示。



(a) 原始图像　　(b) Hadamard 变换后的结果

图 4.4　二维图像的 Hadamard 变换

4.5　Haar变换

Haar 函数 $h_k(x)$ 定义在[0, 1]连续闭区间内，且 $k = 0, 1, \cdots, N-1$，其中 $N = 2^n$ 。整数 k 可以唯一地分解为两个整数 p 和 q 的表达式：

$$k = 2^p + q - 1 \tag{4.58}$$

其中 $0 \leq p \leq n-1$，且当 $p=0$ 时，$q=0, 1$；当 $p \neq 0$ 时，$1 \leq q \leq 2^p$。例如，当 $N=4$ 时，有

k	0	1	2	3
p	0	0	1	1
q	0	1	1	2

用 (p, q) 表示 k，Haar 函数可定义为

$$h_0(x) = h_{0,0}(x) = \frac{1}{\sqrt{N}}, x \in [0,1] \tag{4.59}$$

$$h_k(x) = h_{p,q}(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2}, & \dfrac{q-1}{2^p} \leq x \leq \dfrac{q-1/2}{2^p} \\ -2^{p/2}, & \dfrac{q-1/2}{2^p} \leq x < \dfrac{q}{2^p} \\ 0, & x \in [0,1] \end{cases} \tag{4.60}$$

令 x 取离散值 m/N，$m = 0,1,\mathrm{L}\ ,N-1$，代入式(4.60)得到的行组成 Haar 变换矩阵。例如 $N = 8$ 时的 Haar 变换为

$$\boldsymbol{H}_r = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \tag{4.61}$$

从式(4.61)表示的 $\boldsymbol{H}_r$ 结构可以看出，Haar 变换体现了输入样本向量之间的差异或是各样本的局部均值之间的差异。因此二维 Haar 变换系数 $y(k,l)$，除了 $k=l=0$ 的情况，体现了图像像素的局部均值沿着行和列方向的差异。所以它可以对原始图像进行边缘提取。

Haar 变换具有以下性质[5]。

(1) Haar 变换是实变换和正交变换，因此

$$\boldsymbol{H}_r = \boldsymbol{H}_r^*$$
$$\boldsymbol{H}_r^{-1} = \boldsymbol{H}_r^{\mathrm{T}}$$

(2) Haar 变换的速度非常快。对于一个 $N\times 1$ 向量，其计算复杂度为 $O(N)$。

(3) Haar 变换对图像的能量压缩性能比较差。

【例 4.5】 二维图像的 Haar 变换。

图4.5(a)所示为原始图像，对其进行 Haar 变换后的结果如图4.5(b)所示。



(a) 原始图像　　(b) Haar 变换后的结果

图 4.5　二维图像的 Haar 变换

4.6　小波变换

长期以来，傅里叶分析一直被认为是最完美的数学理论和最实用的方法之一。但是用傅里叶分析只能获得信号的整个频谱，而难以获得信号的局部特性，特别是对于突变信号和非平稳信号难以获得希望的结果。

为了克服经典傅里叶分析本身的弱点，人们发展了信号的时频分析法。1946 年 Gabor 提出的加窗傅里叶变换就是其中的一种，但是加窗傅里叶变换还没有从根本上解决傅里叶分析的固有问题。小波变换的诞生，正是为了克服经典傅里叶分析本身的不足。

4.6.1　连续小波变换

1．一维连续小波变换

给定一个基本函数 $\psi(t)\in L^1 \cap L^2$，且 $\hat{\psi}(0)=0$，令

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right),\quad a,b\in R,\quad a\neq 0 \tag{4.62}$$

显然，$\psi_{a,b}(t)$ 是基本函数 $\psi(t)$ 先做移位再做伸缩以后得到的。b 的作用是确定对 $x(t)$ 分析的

时间位置，也即时间中心。尺度因子 a 的作用是把基本小波 $\psi(t)$ 做伸缩， a,b 不断地变化，可得到一组函数 $\{\psi_{a,b}(t)\}$，称之为小波基函数，或简称小波基。给定平方可积的信号 $x(t)$，即 $x(t)\in L^2(R)$，则 $x(t)$ 的小波变换(Wavelet Transform , WT)定义为

$$W_x(a,b)=\frac{1}{\sqrt{a}}\int x(t)\psi^*\left(\frac{t-b}{a}\right)\mathrm{d}t=\int x(t)\psi^*_{a,b}(t)\mathrm{d}t=\langle x(t),\psi_{a,b}(t)\rangle \tag{4.63}$$

式中 a,b 和 t 均是连续变量，因此该式又称为连续小波变换(CWT)。如无特别说明，式中及以后各式中的积分域都为 $(-\infty,+\infty)$。信号 $x(t)$ 的小波变换 $W_x(a,b)$ 是 a 和 b 的函数，b 是时移，a 是尺度因子。若 $x(t)$ 是实信号， $\psi(t)$ 也是实信号，则 $W_x(a,b)$ 也是实信号；反之， $W_x(a,b)$ 为复函数。显然，式(4.63)的 W 又可解释为信号 $x(t)$ 和一族小波基的内积。ψ^* 表示 ψ 的复共轭。这一变换存在的前提是

$$C_\psi=\int_R\frac{|\hat{\psi}(\omega)|^2}{|\omega|}\mathrm{d}\omega<\infty \tag{4.64}$$

式中， $\hat{\psi}(\omega)$ 是 $\psi(t)$ 的傅里叶变换。由 $W_x(a,b)$ 重构 $x(t)$ 的小波逆变换为

$$x(t)=\frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_x(a,b)\psi_{a,b}(t)\frac{\mathrm{d}a}{a^2}\mathrm{d}b \tag{4.65}$$

式中 C_ψ 由(4.64)给出。

小波变换有如下性质。

(1)小波变换是一种满足能量守恒方程的线性运算，它把一个信号分解成对空间和尺度(即时间和频率)的独立贡献，同时又不失原信号所包含的信息。

(2)小波变换相当于一个具有放大、缩小和平移等功能的数学显微镜，通过检查不同放大倍数下信号的变化来研究其动态特性。

(3)小波变换不一定要求是正交的，而且小波基不唯一，小波函数系的时宽-带宽积很小，且在时间和频率轴上都很集中，即展开系数的能量很集中。

(4)小波变换巧妙地利用了非均匀的分辨率，较好地解决了时间和频率分辨率的矛盾；在低频段用高频率分辨率和低时间分辨率(宽的分析窗口)，而在高频段则用低频率分辨率和高时间分辨率(窄的分析窗口)，这与时变信号的特征一致。

(5)小波变换将信号分解为在对数坐标中具有相同大小频带的集合，这种以非线性的对数方式而不是以线性方式处理频率的方法，对时变信号具有明显的优越性。

(6)小波变换是稳定的，是一个信号的冗余表示。由于 a 和 b 是连续变化的，相邻分析窗的绝大部分是相互重叠的，因而相关性很强。

(7)小波变换同傅里叶变换一样，具有统一性和相似性，其正反变换具有完美的对称性。

2．二维小波变换

在图像处理中，应用的小波变换是二维小波变换。二维函数 $f(x,y)$ 的连续小波变换的定义如下：

$$W_f(a,b_x,b_y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\psi^*_{a,b_x,b_y}(x,y)\mathrm{d}x\mathrm{d}y \tag{4.66}$$

式中，b_x 和 b_y 分别表示在 x 轴和 y 轴上的平移。二维连续小波逆变换定义为

$$f(x,y)=\frac{1}{c_\psi}\int_0^\infty\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_f(a,b_x,b_y)\psi_{a,b_x,b_y}(x,y)\mathrm{d}b_x\mathrm{d}b_y\frac{\mathrm{d}a}{a^3} \tag{4.67}$$

式中，c_ψ 为系数，$\psi_{a,b_x,b_y}(x,y)$ 为

$$\psi_{a,b_x,b_y}(x,y)=\frac{1}{|a|}\psi\left(\frac{x-b_x}{a},\frac{y-b_y}{a}\right) \tag{4.68}$$

而 $\psi(x,y)$ 是一个二维小波基。

4.6.2　离散小波变换

连续小波主要用于理论分析方面，而把连续小波离散化则更有利于实际应用。对 a 和 b 按照如下规律采样：

$$a=a_0^m,\quad b=nb_0a_0^m \tag{4.69}$$

式中，$a_0>1$，$b_0\in R$，$m,n\in Z$。则由式(4.62)得离散小波基

$$\psi_{m,n}(t)=a_0^{-m/2}\psi[a_0^{-m}t-nb_0)],\qquad m,n\in Z \tag{4.70}$$

对给定的信号 $x(t)$，离散小波变换(Discrete Wavelet Transform, DWT)为

$$W_x(m,n)=\int x(t)\psi_{m,n}^*(t)\mathrm{d}t \tag{4.71}$$

如果 $x(t)$ 也是离散的，记为 $x(k)$，则有

$$W_x(m,n)=\sum_k x(k)\psi_{m,n}^*(k) \tag{4.72}$$

小波逆变换的离散形式为

$$x(k)=\sum_{m,n}W_x(m,n)\psi_{m,n}(k) \tag{4.73}$$

由于离散小波变换是对连续小波变换的伸缩因子和平移因子按一定规则采样得到的，故连续小波变换所具有的性质，离散小波变换一般仍具备。

小波分析的应用是与小波分析的理论研究紧密地结合在一起的。现在，它已经在电子信息产业领域取得了令人瞩目的成就。电子信息技术是六大高新技术领域之一，其重点在于图像和信号处理。现在，对于时间不变的信号，处理的理想工具仍然是傅里叶分析，但是在实际应用中，绝大多数信号是时变的非稳定信号，因而小波分析就成为分析非稳定信号的有力工具。

小波分析的应用领域十分广泛，包括数学领域的许多学科，信号分析、图像处理，量子力学、理论物理，军事电子对抗与武器的智能化，计算机分类与识别，音乐与语言的人工合成，医学成像与诊断，地震勘探数据处理，大型机械的故障诊断，等等。

4.6.3　多分辨率分析

S. Mallat 提出的多分辨率分析(Multi Resolution Analysis, MRA)的概念，在泛函分析的框

架下统一了各种具体的小波构造方法，给出了构造正交小波基的一般方法及其快速算法，并将小波变换应用于图像分解和重建，这是小波理论上的一个突破性进展。

多分辨率分析建立在函数空间分解概念之上，将信号在不同尺度的函数空间进行分解，然后比较各个空间所包含的信号信息。多分辨率分析的目的是在不同尺度(频域空间)对信号进行观察。在大尺度下，观察信号的全貌或信号的缓变成分，或对信号进行粗略逼近；在小尺度下，观察信号的局部或信号的快速变化成分[6]。

Mallat 给出了多分辨率分析的定义。

设 $\{V_j\}, j \in Z$ 是 $L^2(R)$ 空间中的一系列闭合子空间，如果它们满足如下五个性质，则说 $\{V_j\}$， $j \in Z$ 是一个多分辨率近似。这五个性质如下所示。

(1)固定尺度下的平移不变性

$\forall (j,k) \in Z^2$，若 $x(t) \in V_j$，则

$$x(t - 2^j k) \in V_j \tag{4.74}$$

也就是说，空间 V_j 对于正比于尺度 2^j 的位移具有不变性，也即函数的时移不改变其所属的空间。若令 $a = 2^j$，则 b 应取 $b = 2^j k b_0$，将 b_0 归一化为 1，则

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) = a^{-j/2} \psi(2^{-j}t - k) = \psi_{j,k}(t) \tag{4.75}$$

所以，式(4.74)实际上应等效为

$$\forall j \in Z，若 x(t) \in V_j，则 x(t-k) \in V_j \tag{4.76}$$

这是因为若 $\forall j \in Z$，则必有 $2^j \in Z$。

(2)一致单调性

$$\forall j \in Z，V_j \supset V_{j+1}，即 V_0 \supset V_1 \supset V_2 \text{L } V_j \supset V_{j+1} \text{L} \tag{4.77}$$

性质 2 说明，在尺度 2^j (或 j)时，对 $x(t)$ 做的是分辨率为 2^{-j} 的近似，其结果将包含在较低一级分辨率 2^{-j-1} 时对 $x(t)$ 近似的所有信息，这就是空间的包含，也即式(4.77)。

(3)尺度伸缩性

$$\forall j \in Z，若 x(t) \in V_j，则 x(t/2) \in V_{j+1} \tag{4.78}$$

性质 3 是性质 2 的直接结果。在 V_{j+1} 中，函数做了二倍的扩展，分辨率降为 2^{-j-1}，所以 $x(t/2)$ 应属于 V_{j+1}。

(4)逼近性

$$\lim_{j\to\infty} V_j = \bigcap_{j=-\infty}^{\infty} V_j = \{0\}，\quad \lim_{j\to\infty} V_j = \text{Closure}\left(\bigcup_{j=-\infty}^{\infty} V_j\right) = L^2(R) \tag{4.79}$$

性质 4 说明当 $j \to \infty$ 时，分辨率 $2^{-j} \to 0$，这时我们将会失去 $x(t)$ 的所有信息，即

$$\lim_{j\to\infty} P_j x(t) = 0$$

从空间上讲，所有 $V_j, j = -\infty \sim +\infty$ 的交集为零空间。

⑸ 正交基存在性

存在一个基本函数 $\theta(t)$，使得 $\{\theta(t-k)\}$，$k \in Z$ 是 V_0 中的 Riesz 基。

性质 5 是性质 4 的另一面，即当 $j \to -\infty$ 时，分辨率 $2^{-j} \to \infty$，那么信号 $x(t)$ 在该尺度下的近似收敛于自身，即

$$\lim_{j \to -\infty} \left| P_j x(t) - x(t) \right| = 0 \tag{4.80}$$

从空间上讲，即所有 $V_j, j = -\infty \sim +\infty$ 的并集收敛于整个 $L^2(R)$ 空间。

性质 5 说明了 V_0 中 Riesz 基的存在性问题，并将由此引出 $V_0, V_1, \mathrm{L}, V_j$ 中正交基的存在性问题，因此需要着重加以解释。

设 V_0 是一能量有限的空间 $L^2(R)$，$\{\boldsymbol{\theta}_k = \boldsymbol{\theta}(t-k)\}, k \in Z$ 是 V_0 中的一组向量，其个数与 V_0 的维数一致。自然，V_0 中的任一元素 $\boldsymbol{x}$ 都可表示为 $\boldsymbol{\theta}_k$ 的线性组合，即

$$\boldsymbol{x}(t) = \sum_{k=-\infty}^{\infty} c_k \boldsymbol{\theta}(t-k) \tag{4.81}$$

我们知道，若 ⑴ $\{\boldsymbol{\theta}_k = \boldsymbol{\theta}(t-k)\}, k \in Z$ 之间是线性无关的，⑵ 并且存在常数 $0 < A \leq B < \infty$ 使得

$$A\|\boldsymbol{x}\|^2 \leq \sum_{k=-\infty}^{\infty} |c_k|^2 \leq B\|\boldsymbol{x}\|^2 \tag{4.82}$$

则 $\boldsymbol{\theta}(t-k), k \in Z$ 是 V_0 中的 Riesz 基。

若将信号 $\boldsymbol{x}(t) \in L^2(R)$ 按以下空间进行分解：

$$L^2(R) = \left[\sum_{j=-\infty}^{J} W_j \right] \oplus V_J \tag{4.83}$$

其中 J 为任意尺度。$\boldsymbol{x}(t)$ 分解成 W_j（$j = -\infty \sim J$）的投影和 V_J 空间的投影，即

$$\boldsymbol{x}(t) = \sum_{j=-\infty}^{J} \sum_{k=-\infty}^{\infty} d_{j,k} \boldsymbol{\psi}_{j,k}(t) + \sum_{k=-\infty}^{\infty} c_{J,k} \boldsymbol{\phi}_{J,k}(t) \tag{4.84}$$

当分解尺度 J 趋于无穷大时，V_J 空间趋向{0}，式(4.84)可改写为

$$\boldsymbol{x}(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \boldsymbol{\psi}_{j,k}(t) \tag{4.85}$$

【例 4.6】 二维图像的小波变换。

图4.6(a)所示为原始图像，图4.6(b)所示为用 MATLAB 提供的正交小波 db2 对图像做一级小波分解后的图像。可以看出，经过一级小波分解，原始图像被分解成几个子图像，每个子图像包含了原始图像中不同的频率成分。左上角子图包含了原始图像的低频分量，即图像的主要特征，低频分量可以再次分解；右上角子图包含了图像的垂直分量，即包含了较多的垂直边缘信息；左下角子图包含了图像的水平分量，即包含了较多的水平边缘信息；右下角子图包含了图像的对角分量，即同时包含了垂直和水平边缘信息。从图 4.6(b)中可以看出，

经过小波变换，原始图像的全部信息被重新分配到了 4 个子图中，左上角子图包含了原始图像的低频信息，但失去了一部分边缘细节信息，这些失去的细节信息被分配到了其他三个子图中。由于失去了部分细节信息，所以左上角子图比原始图像模糊了一些，不仅如此，其长宽尺寸也降低到原来的一半，即分辨率降低到原来的 1/4。一种最容易理解的图像压缩方法就是，丢弃三个细节子图，只保留并编码低频子图。但实际上，并不是通过这么简单的处理来进行图像压缩的，三个细节子图不会被丢掉，而是与低频子图一起编入码流，这样才可能在解码时恢复出完整的原始图像，当然，如果用户只需要一个小尺寸的图像，那就只需从码流中解码出低频子图即可。低频子图可以进一步分解，经过二级分解后，系数矩阵所对应的图像如图4.6(c)所示。在图4.6(c)中，低频子图的尺寸降到了原始图像的 1/16，可见每一级小波分解都是对空间分辨率和频率分量的进一步细分。从此例可以看出，小波变换为在一个码流中实现图像多级分辨率提供了基础。

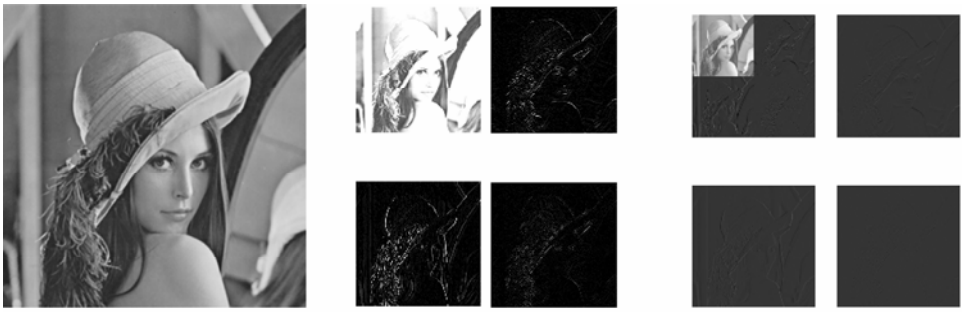


图 4.6 图像的小波变换

4.6.4 正交小波包

上节讨论的多分辨率分析，信号子空间仅仅针对尺度空间(信号的近似部分) V_J 展开，并没有对小波空间(信号的细节部分) W_j 再次剖分，其空间剖分如图4.7所示。

<table>
<tr><td colspan="4">V_0</td></tr>
<tr><td colspan="3">V_1</td><td>W_1</td></tr>
<tr><td colspan="2">V_2</td><td>W_2</td><td>W_1</td></tr>
<tr><td>W_2</td><td>W_3</td><td>W_2</td><td>W_1</td></tr>
</table>

图 4.7 多分辨率分析过程的空间剖分

显然，小波空间的分辨率不够。如果感兴趣的信号落在小波空间 W_j 中，同时在空间 W_j 中所占的区域(或频率范围)又很小(例如窄带信号)，则该信号的特征可能会被空间 W_j 中的其他成分掩盖。针对小波变换的这个缺陷，20 世纪 90 年代初 Coifman, Meyer, Wickerhauster 等提出了小波包分解理论，对小波空间分解进行扩展。理想的小波包时-域空间分解如图4.8 所示[6]。

图4.8 分解子空间 $U_{j,n}$ 中的 j 表示分解层次，相应尺度 $a=2^j$，n 表示第 j 层分解的子空间编号，第 j 层共有 2^j 组正交基。第 j 层的第 n 个编号子空间由 $\{U_{j,n}(t-k), k\in Z\}$ 构成。

<table>
<tr><td colspan="8">V_0</td></tr>
<tr><td colspan="4">$U_{1,0}$</td><td colspan="4">$U_{1,1}$</td></tr>
<tr><td colspan="2">$U_{2,0}$</td><td colspan="2">$U_{2,1}$</td><td colspan="2">$U_{2,2}$</td><td colspan="2">$U_{2,3}$</td></tr>
<tr><td>$U_{3,0}$</td><td>$U_{3,1}$</td><td>$U_{3,2}$</td><td>$U_{3,3}$</td><td>$U_{3,4}$</td><td>$U_{3,5}$</td><td>$U_{3,6}$</td><td>$U_{3,7}$</td></tr>
</table>

图 4.8　小波包分析过程的空间剖分

小波包分解过程要求同一层之间的各子空间相互正交。同时，同一层的所有子空间之和应该等于整个信号空间。各子空间的正交基$\{U_{j,n}(t-k), k \in Z\}$之间必须满足

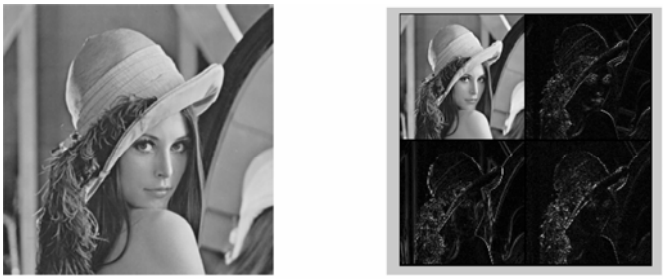
$$\{U_{j,n1}(t-k), k \in Z\} \cap \{U_{j,n2}(t-k), k \in Z\} = 0, \quad n_1 \neq n_2 \tag{4.86a}$$

$$\bigcup_{\oplus i} \{U_{j,ni}(t-k), k \in Z\} = L^2(R) \tag{4.86b}$$

采用小波包分解之后，信号可由不同的子空间组合构成。针对每一类待分析信号，采用选定的小波包分解函数，都存在一个最优树。

【例 4.7】 对二维图像进行小波包分析。

对图4.9(a)所示的原始图像，利用 MATLAB 对图像进行小波包分析(小波基函数为 db2)，得到的结果如图4.9(b)所示。



(a) 原始图像　　(b) 小波包分析结果

图 4.9　二维图像的小波包分析

习题 4

4.1　已知有两个数据集分别为ω_1:(0, 0, 1)，(1, 1, 1)，(1, 0, 1)，(1, 0, 0)和ω_2:(0, 0, 0)，(1, 1, 0)，(0, 1, 0)，(0, 1, 1)。

(1)将该 8 个数据作为一个数据集对其进行 K-L 变换

(2)求这两个数据集的类内离散矩阵，并以此作为其产生矩阵进行 K-L 变换。

4.2　已知一组数据的协方差矩阵为$\begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$，试问

(1)协方差矩阵中各元素的含义。

(2)K-L 变换的最佳准则是什么？

(3)为什么说经 K-L 变换后，消除了各分量之间的相关性？

4.3　求如下序列的离散傅里叶变换：

(1) $x_1(n)=\delta(n-3)$ (2) $x_2(n)=\frac{1}{2}\delta(n+1)+\delta(n)+\frac{1}{2}\delta(n-1)$

(3) $x_3(n)=\begin{cases}(1/2)^n, & n=0,2,4,\cdots\\ 0, & \text{其他}\end{cases}$

4.4 简述离散傅里叶变换的性质及在图像处理中的应用。

4.5 小波变换有哪些特点？

4.6 取一幅实际的图像，对其进行傅里叶变换并观察变换后的图像，从图像上来分析两者之间的关系。

参考文献

[1] 边肇祺等. 模式识别. 第二版. 北京：清华大学出版社，1999:177.

[2] 钟珞等. 模式识别. 武汉：武汉大学出版社，2006:131.

[3] 蔡元龙. 模式识别. 西安：西安电子科技大学出版社，1992:122.

[4] 何小海. 图像通信. 西安：西安电子科技大学出版社， 2005:30, 32, 34.

[5] Anil K. Jain 著，韩博等译. 数字图像处理基础. 北京：清华大学出版社，2006:121, 124, 126, 129.

[6] 祁才君. 数字信号处理技术的算法分析与应用. 北京：机械工业出版社，2005:280, 296.

[7] 舒宁等. 模式识别的理论与方法. 武汉：武汉大学出版社，2004.

[8] 黄智编著. 天津：天津科学技术出版社，1989.

[9] 蔡元龙. 模式识别. 西安：西安电子科技大学出版社，1992.

[10] 沈庭芝. 数字图像处理及模式识别. 北京：北京理工大学出版社，1998.

[11] 贾永红编著. 计算机图像处理与分析. 武汉：武汉大学出版社，2001.

[12] 朱秀昌. 数字图像处理与图像通信. 北京：北京邮电大学出版社，2002.

[13] 高守传. VISUAL C++实践与提高：数字图像处理与工程应用篇. 北京：中国铁道出版社，2006.

[14] 刘榴娣. 实用数字图像处理. 北京：北京理工大学出版社，2003.

[15] 赵先锋. 离散 K L 变换在汽车车牌字符识别中的应用一例. 仪器仪表学报，25(4), 2004-08.

[16] 李坤，朱焕伟，杨俊明. 离散 K-L 变换在路标识别中的应用. 中国民族大学学报，2008-05, 17(2).

[17] Dy, JG and CE Brodley (2004). *Feature Selection for Unsupervised Learning*. Journal of Machine Learning Research, 5, 845–889.

[18] R. O. Duda, P. E. Hart. *Pattern Classification and Scene Analysis*. NewYork: John wiley & sons, 1973.

[19] Nello Cristianini & Jogn Shawe-Jaylor. *An Introduction to Support Vector Machines and other Kernel-based learning method*. Cambridge University Press 2000.

[20] 王朝英. 信号处理原理学习指导. 北京：清华大学出版社，北京交通大学出版社，2006.

[21] 胡学龙. 数字图像处理. 北京：电子工业出版社，2006.

[22] 方勇. 数字信号处理——原理与实践. 北京：清华大学出版社，2006.

[23] 李朝晖，张弘. 数字图像处理及应用. 北京：机械工业出版社，2004.

[24] 张奎，黄凤岗. 模式识别. 哈尔滨：哈尔滨工程大学出版社，1998.

第5章　聚 类 分 析

聚类(Clustering)就是按照一定的要求和规律对事物进行区分和分类的过程，在这一过程中没有任何关于分类的先验知识，仅靠事物间的相似性作为类属划分的准则，因此是无监督分类。聚类分析是指用数学的方法研究和处理给定对象的分类。聚类是一个古老的问题，它伴随着人类社会的产生和发展而不断深化，人类认识世界就必须区分不同的事物，并认识事物间的相似性。

多年来，人们提出了许多关于“聚类”的定义[1-4]，但一直没有通用的定义。温熙森[5]给出的聚类分析定义是：“聚类分析是统计模式识别的另一重要工具，它把模式归入到这样的类别或聚合类：同一个聚合类的模式比不同聚合类中的模式更相近”。它的基本原理就是在没有先验知识的情况下，基于“物以类聚”的观点，用数学方法分析各模式向量之间的距离及分散情况，按照样本的距离远近划分类别。

为了能够准确描述聚类的定义，希腊的 Sergios Theodoridis 给出了聚类的数学定义[4]。设 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N\}$ 是待聚类样本的数据集，定义 $\boldsymbol{X}$ 的 K 聚类就是将 $\boldsymbol{X}$ 分割成 K 个集合(聚类) $C_1,C_2,\mathrm{L}\ ,C_K$，使其满足下面三个条件：

(1) $C_1\neq\varnothing, i=1,\mathrm{L}\ ,K$

(2) $C_1\ \mathrm{U}\ C_2\ \mathrm{UL}\ \mathrm{U}\ C_K=\boldsymbol{X}$

(3) $C_i\ \mathrm{I}\ C_j\neq\varnothing, i\neq j; i,j=1,\mathrm{L}\ ,K$

与聚类相关的概念是聚集(Clumping)，聚集允许一个模式属于多个类。例如，根据单词的意思将其分类，某些单词有多个意思，可以属于几个不同的类。但是，在本章中，我们只讨论聚类问题。

聚类分析是无监督分类方法，它把一个没有类别标记的样本集按照某种准则划分成若干个子集，使相似的样本尽可能归为一类，不相似的样本尽量划分到不同的类中。在实际应用中，很多情况下无法预先知道样本的类别，只能用没有样本类别标记的样本集进行分类器设计，这就是无监督分类方法。监督分类方法和无监督分类方法的区别主要如下：

(1)监督分类方法有训练样本集，在训练样本集中给出不同类别的训练样本，用这些训练样本可以找出区分不同类样本的方法，从而在特征空间中划定决策域。

(2)监督分类方法由训练阶段和测试阶段组成。训练阶段利用训练集中的训练样本进行分类器设计，确定分类器参数；测试阶段将待识别样本输入，根据分类的决策规则，确定待识别样本的所属类别。

(3)无监督分类方法可用来分析数据的内在规律，它没有训练样本；如聚类分析、主分量分析、数据拟合等方法都是无监督分类方法。

对样本集进行聚类分析要考虑的问题如下：

(1)相似性测度。如何度量样本间的相似性。

(2)聚类准则。如何聚类取决于聚类的准则函数，使某种聚类准则达到极值。

(3)聚类算法。用什么算法找出使准则函数取极值的最好聚类结果。

(4)聚类有效性。判定聚类在多大程度上反映了样本集的真实结构，应如何确定样本集中正确的类别数。

5.1 相似性测度和聚类准则

5.1.1 相似性测度

为了将样本集划分成不同的类别，需要定义一种相似性测度(Similarity Measure)来度量同一类样本之间的相似性和不同样本之间的差异性。具体选用什么样的测度进行分类，要依据样本之间的实际情况做适当的选择。

1．欧氏距离(Euclidean Distance)

对两个样本 $\boldsymbol{x}$ 和 $\boldsymbol{y}$，其欧氏距离定义为

$$D=\|\boldsymbol{x}-\boldsymbol{y}\|=\left[(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})\right]^{\frac{1}{2}}=\sqrt{\sum_{i=1}^{d}(x_i-y_i)^2} \tag{5.1}$$

欧氏距离是最常用的相似性测度。如果选用欧氏距离作为相似性度量，需要特征空间是各向同性的。也就是说，由欧氏距离所确定的样本具有平移和旋转不变性。

2．马氏距离(Mahalanobis Distance)

$$D_2=(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m}) \tag{5.2}$$

其中 $\boldsymbol{x}$ 是样本的特征向量，$\boldsymbol{m}$ 是样本的均值向量，$\boldsymbol{\Sigma}$ 是协方差矩阵。

若 $\boldsymbol{\Sigma}$ 为单位阵，则马氏距离与欧氏距离相似。马氏距离使用的难点在于，只有当已知类别的样本集给定时，才能计算出协方差矩阵 $\boldsymbol{\Sigma}$，这往往是难以做到的，因为待分类的样本本身是无类别的。

3．明氏距离(Minkowski Distance)

$$D_m(\boldsymbol{x},\boldsymbol{y})=\left[\sum_i|x_i-y_i|^m\right]^{\frac{1}{m}} \tag{5.3}$$

其中 m 为正整数，x_i 和 y_i 分别表示 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的第 i 个分量。当 $m=2$ 时，为欧氏距离；当 $m=1$ 时，为绝对距离，也称市区距离(City Block Distance)：

$$D_1(\boldsymbol{x},\boldsymbol{y})=\sum_i|x_i-y_i| \tag{5.4}$$

4．Tanimoto 测度

Tanimoto 测度也称为 Tanimoto 距离，Tanimoto 测度可用于实向量测量，也可用于离散值向量测量，定义为

$$s_T(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}} \tag{5.5}$$

向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 越相似，$s_T(\boldsymbol{x}, \boldsymbol{y})$ 值越大。

5．角度相似性函数

相似性测度不一定只限于距离，可以用向量夹角余弦反映几何相似性。在模式向量具有扇形分布时，经常采用这种测度：

$$s(\boldsymbol{x}, \boldsymbol{y}) = \cos(\boldsymbol{x} \cdot \boldsymbol{y}) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|} \tag{5.6}$$

其中 $s(\boldsymbol{x}, \boldsymbol{y})$ 是向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间夹角的余弦。

如图5.1所示，$\boldsymbol{x}$ 和 $\boldsymbol{x}_2$ 同属于一类，$s(\boldsymbol{x}, \boldsymbol{x}_2) = \cos\theta_1$；$\boldsymbol{x}_3$ 属于另一类，$s(\boldsymbol{x}, \boldsymbol{x}_3) = \cos\theta_2$，此时余弦值越大，相似性就越大。

距离和角度相似性函数作为相似性的测度各有其局限性。距离对于坐标系的旋转和位移是不变的，对于放大缩小并不具有不变性的性质。角度相似性函数对于坐标系的旋转、放大、缩小是不变的，但对于位移不具有不变性的性质。用角度相似性函数作为相似性的测度还有一个缺点，即当不同类的样本分布在从模式空间原点出发的一条直线上时，所有样本之间角度相似性函数几乎都等于 1，造成归为一类的错误。

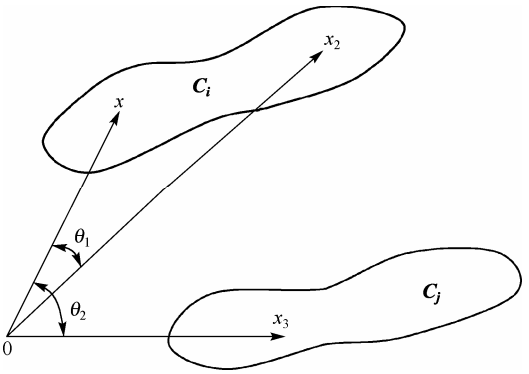


图 5.1　角度相似性函数

5.1.2　聚类准则

有了相似性测度，就能聚类相似的模式样本。而要剔除相异的样本，就需要有数值的聚类准则，用聚类准则衡量对样本集的一种划分结果的好坏。

假定有一组样本 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \mathrm{L}\ , \boldsymbol{x}_N\}$，我们要把它划分成 K 个不相交的子集 $C_1, C_2, \mathrm{L}\ , C_K$，每个子集代表一个聚类。同一类中的样本比不同类中的样本相似性高一些，于是存在多种分类方法。那么到底哪种分类方法最好呢？这时，需要定义一个准则函数，有了准则函数，聚类问题就变成求解准则函数最优问题。下面介绍几种准则函数。

1．误差平方和准则

误差平方和准则是聚类问题中最简单而又广泛应用的准则。准则函数为

$$J = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in \boldsymbol{C}_i} \|\boldsymbol{x} - \boldsymbol{m}_i\|^2 \tag{5.7}$$

其中 K 是聚类类别数目，$\boldsymbol{C}_i$ 是第 i 类聚类中心域的样本集合，n_i 是 $\boldsymbol{C}_i$ 中的样本数，$\boldsymbol{m}_i$ 是第 i 类的样本的均值向量：

$$\boldsymbol{m}_i = \frac{1}{n_i}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\boldsymbol{x} \tag{5.8}$$

J 度量了用 m 个聚类中心 $\boldsymbol{m}_1, \boldsymbol{m}_2, \mathrm{L}\ , \boldsymbol{m}_K$ 代表的 K 个子集 $C_1, C_2, \mathrm{L}\ , C_K$ 所产生的总误差平方。对于不同的聚类，J 极小的聚类是误差平方和准则下的最优结果。

经过简单的代数运算，可以将式(5.7)中的均值向量 $\boldsymbol{m}_i$ 消去，得到另一种准则函数表示形式

$$J = \sum_{i=1}^{K} n_i s_i \tag{5.9}$$

其中 K 是聚类类别数目，n_i 是第 i 个聚类 $\boldsymbol{C}_i$ 中的样本数，s_i 是相似性算子：

$$s_i = \frac{1}{n_i^2}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\sum_{\boldsymbol{x}'\in\boldsymbol{C}_i} \| \boldsymbol{x} - \boldsymbol{x}' \|^2 \tag{5.10}$$

s_i 是第 i 类中各对点之间距离平方的平均，以欧氏距离作为相似性测度。

若 s_i 以无量纲的相似性函数 $s(\boldsymbol{x}, \boldsymbol{x}')$ 来取代相似性算子 s_i 中的欧氏距离，则有

$$s_i = \frac{1}{n_i^2}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\sum_{\boldsymbol{x}'\in\boldsymbol{C}_i} \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}'}{\|\boldsymbol{x}\|\cdot\|\boldsymbol{x}'\|} \tag{5.11}$$

把式(5.11)代入式(5.9)即准则函数 J 的表示式中，可得到准则函数的另一种表示形式。

这种准则函数适用于同类的样本很密集且类别间分离明显的情况。如果类别间距离小或各类别中样本数目相差很大时，可能发生错误。图5.2(a)中所示的模式分类，使用这种准则进行聚类可获得最好的效果。而图5.2(b)中的模式分布，使用这种准则得到的效果就不理想。

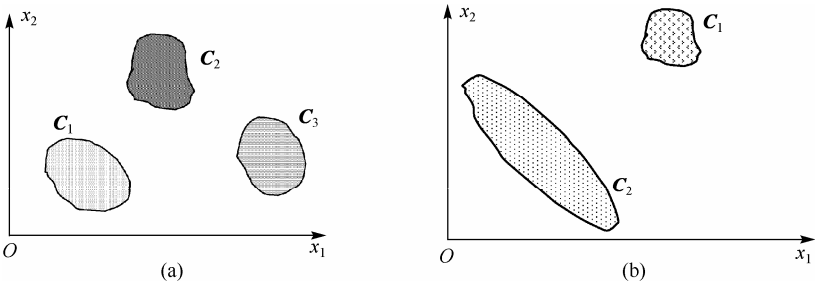


图 5.2　模式分布图

当不同类别中的样本数相差很大而类别之间距离较小时，样本数多的一类有可能被一分为二，这样聚类的结果是：将样本数大的类拆分，得到的误差平方和准则函数 J 比保持完整时小(如图5.3所示)。因此，有可能将 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$ 分错［见图5.3(a)］，发生错误聚类。

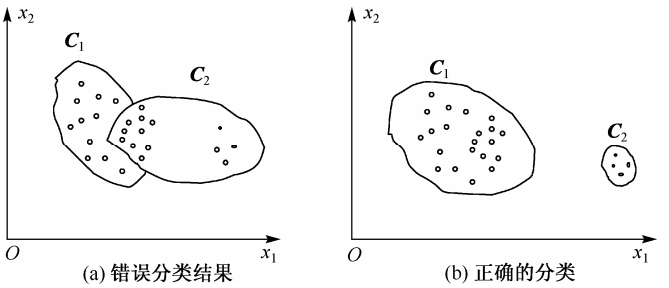


图 5.3　把大类错误地拆分的问题

2．离散度准则

离散度准则也称为散布准则，它不仅能反映同类样本的聚集程度，也能反映不同类之间的分离程度。在介绍离散度准则之前，首先定义离散度矩阵。

第 i 类的均值向量(第 i 类的中心)

$$\boldsymbol{m}_i = \frac{1}{n_i}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\boldsymbol{x} \tag{5.12}$$

总平均向量

$$\boldsymbol{m} = \frac{1}{N}\sum_{i=1}^{K} n_i\boldsymbol{m}_i \tag{5.13}$$

第 i 类的离散度矩阵

$$\boldsymbol{S}_i = \sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^{\mathrm{T}} \tag{5.14}$$

类内离散度矩阵

$$\boldsymbol{S}_W = \sum_{i=1}^{K}\boldsymbol{S}_i \tag{5.15}$$

类间离散度矩阵

$$\boldsymbol{S}_B = \sum_{i=1}^{K} n_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}} \tag{5.16}$$

总离散度矩阵

$$\boldsymbol{S}_T = \sum_{\boldsymbol{x}\in\boldsymbol{X}}(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}} \tag{5.17}$$

其中 K 是聚类类别数目，$\boldsymbol{C}_i$ 是第 i 类聚类中心域的样本集合，n_i 是 $\boldsymbol{C}_i$ 中的样本数，$\boldsymbol{m}_i$ 是第 i 类的样本的均值向量。

根据上述定义可以证明，总离散度矩阵等于类内离散度矩阵与类间离散度矩阵之和，即

$$\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B$$

总离散度矩阵与如何划分类别无关，仅与全部样本有关，但是类内和类间离散度矩阵都与类别划分有关。为了更准确地度量类内和类间离散度矩阵，需要引入一个标量来衡量离散度矩阵的大小。下面介绍两种度量矩阵的标量，即矩阵的迹和矩阵的行列式。

(1)迹准则

方阵的主对角线元素之和称为这个方阵的迹，迹是度量离散度矩阵大小的最简单的标量方法。大致来说，迹是离散度半径的平方和，它正比于数据在各个坐标轴方向上的方差之和。所以最小化矩阵的对角线元素之和，最小化类内离散度矩阵 $\boldsymbol{S}_W$ 的迹可以作为一种准则函数。事实上，可以证明这个准则与前面介绍的误差平方和准则是一致的。由式(5.14)和式(5.15)可得

$$\mathrm{tr}\boldsymbol{S}_W = \sum_{i=1}^{K}\mathrm{tr}\boldsymbol{S}_i = \sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\|\boldsymbol{x}-\boldsymbol{m}_i\|^2 = J_e \tag{5.18}$$

因为 $\mathrm{tr}\boldsymbol{S}_T = \mathrm{tr}\boldsymbol{S}_W + \mathrm{tr}\boldsymbol{S}_B$，而 $\boldsymbol{S}_T$ 与样本如何划分类别无关，所以最小化类内离散度矩阵迹准则函数 $\mathrm{tr}\boldsymbol{S}_W$ 的同时，也最大化了类间离散度矩阵迹准则函数 $\mathrm{tr}\boldsymbol{S}_B$，即

$$\mathrm{tr}\boldsymbol{S}_B = \sum_{i=1}^{K} n_i \| \boldsymbol{m}_i - \boldsymbol{m} \|^2 \tag{5.19}$$

(2)行列式准则

矩阵的行列式也可作为离散度矩阵的另一种标量度量。这种度量大致上反映了离散体积的平方，这是因为它正比于数据在各个主轴方向上的方差之积。但是，当类别数小于或等于特征向量的维数时，类间离散度矩阵 $\boldsymbol{S}_B$ 是奇异矩阵，即行列式的值为 0。所以不能选择 $\boldsymbol{S}_B$ 的行列式作为准则函数，一般选择类内离散度矩阵 $\boldsymbol{S}_W$ 的行列式作为准则函数。假定类内离散度矩阵 $\boldsymbol{S}_W$ 是非奇异的，于是得到准则函数为

$$J_d = |\boldsymbol{S}_W| = \left| \sum_{i=1}^{K} \boldsymbol{S}_i \right| \tag{5.20}$$

尽管类内离散度矩阵 $\boldsymbol{S}_W$ 有时也会是奇异的，比如当样本数与类别数之差小于维数时，但是使得样本数与类别数之差大于维数的条件易于满足，也就是说，使 $\boldsymbol{S}_W$ 非奇异比使 $\boldsymbol{S}_B$ 非奇异的条件容易满足。

5.2 聚类算法

5.2.1 聚类算法的分类

聚类算法是试图识别数据集合聚类的特殊性质的学习过程。聚类算法主要包括以下几种[4]。

(1)顺序算法(Sequential algorithms)。这类算法产生一个独立的聚类，并至少将所有特征向量使用一次或几次，是直接、快速的算法；最后的结果与特征向量参与算法的顺序有关。这类算法适用于致密和超球面或超椭圆面形状的聚类，并与使用的距离测度有关。

(2)层次聚类算法(Hierarchical clustering algorithms)，也称分级聚类算法。具体有：

- 合并算法(Agglomerative algorithms)。这类算法在每一步都减少类数，聚类结果是合并前一步的两个类。
- 分裂算法(Divisive algorithms)。这类算法的原理与合并算法的原理相反，在每一步增加类数，聚类结果是将前一步的一个类分裂成两个类。

(3)基于准则函数最优的聚类算法。这类算法用准则函数定量地判断聚类结果，通常类数是固定的，通过迭代求最优准则函数，不断产生并调整聚类结果，当准则函数局部最优时，迭代结束。具体有：

- 硬聚类算法(Hard clustering algorithms)。如果一个特征向量属于某一类，就不能属于另一类；根据选择的准则函数，当准则函数局部最优时，将特征向量分到各个类中。最著名的算法是 ISODATA [6]。
- 概率聚类算法(Probabilistic clustering algorithms)。它是硬聚类算法的特例，采用贝叶

斯分类方法，并且每个特征向量 $\boldsymbol{x}$ 被分到使 $P(\boldsymbol{C}_i|\boldsymbol{x})$ 最大的聚类 $\boldsymbol{C}_i$ 中，通过适当地定义优化任务完成概率估计。

- 模糊聚类算法(Fuzzy clustering algorithms)。在这类算法中，计算特征向量属于某类的隶属度，当隶属度大于给定阈值时，特征向量属于该类。
- 可能聚类算法(Possibilistic clustering algorithms)。这类算法测量特征向量 $\boldsymbol{x}$ 属于聚类 $\boldsymbol{C}_i$ 的可能性。

(4) 其他算法。

- 分支和约束聚类算法(Branch and bound clustering algorithms)。对于给定的类数，这类算法使用给定准则，不需要考虑所有可能的聚类就可以提供全局最优聚类。然而，它们的计算量大。
- 遗传聚类算法(Genetic clustering algorithms)。这类算法用一个可能的聚类作为初始种群，迭代生成新的种群，按照相关的准则，新种群一般比原来的种群具有更优的聚类。
- 随机松弛算法(Stochastic relaxation algorithms)。在指定准则情况下，这种方法保证在一定的条件下概率收敛于全局最优聚类，但是计算量大。
- 谷点搜索聚类算法(Valley-seeking clustering algorithms)。这类算法把特征向量当做一个多维任意变量 $\boldsymbol{x}$ 的实例，它基于一个假设，即很多特征向量驻留的 $\boldsymbol{x}$ 区域对应着 $\boldsymbol{x}$ 的概率密度函数值增加的区域。所以，对概率密度函数的估计可能使得聚类形成的区域更加显著。
- 竞争学习算法(Competitive learning algorithms)。这类算法是迭代算法，不使用准则函数。根据某种距离度量，产生一些聚类，而且收敛于最可判断的一个。其典型代表是基本的竞争学习算法和漏洞学习算法。
- 基于形态学变换技术的算法(Algorithms based on morphological transformation techniques)。这类算法使用形态学变换，以获得更好的聚类划分。
- 基于密度的算法(Density-based algorithms)。该类算法把聚类视为一维空间中数据较为密集的区域。从这个角度看，该类算法与谷点搜索算法相似。
- 子空间聚类算法(Subspace clustering algorithms)。该类算法非常适合于处理高维数据集。在某些应用中，特征空间的维数甚至可以达到几千。
- 基于核的方法(Kernel-based methods)。该类算法的基础是在非线性支持向量机中，采用“核方法”实现原始空间 $\boldsymbol{X}$ 的映射，把 $\boldsymbol{X}$ 转化为高维空间，在高维空间中再使用广义最优分类方法对样本进行划分。

5.2.2　层次聚类算法

层次聚类算法(Hierarchical clustering algorithms)也称系统聚类算法、分级聚类算法或树聚类，是实际应用中采用得最多的算法之一。层次聚类算法的基本思想是：将 N 个样本自成一类，然后计算类与类之间的距离，再将距离最近的两类合并，减少类别数，直至达到分类要求为止。

算法描述如下：

(1) 初始分类。N 个模式样本自成一类，即 $G_i^{(0)} = \{\boldsymbol{x}_i\}, i = 1, 2, \mathrm{L}\ , N$。

(2)计算各类间的距离 D_{ij}，得到一个对称的 $m\times m$(初始时为 $N\times N$)距离矩阵 $\boldsymbol{D}^{(l)}$，l 为逐次聚类合并的次数，其中 m 为类别数。

(3)找出 $\boldsymbol{D}^{(l)}$中的最小元素，将其对应的两类合并为一类，由此建立新的分类 $G_1^{(l+1)}$，$G_2^{(l+1)}$,L 。

(4)转至(2)，重复计算及合并。

结束条件：

- 检查类别数，如果类别数为 1，则停止。
- 设定一个距离阈值 D_{T}，当 $\boldsymbol{D}^{(l)}$中的最小元素超过给定阈值 D_{T}时，则停止。

在层次聚类算法中，关键技术是类间距离的选择和类别数的确定。选择不同的类间距离，将得到不同的聚类过程和结果。下面介绍几种类间距离的计算方法。

(1)最短距离法。如果 H 和 K 是两个聚类，则两类间的最短距离定义为

$$D_{HK}=\min\left\{D\left(\boldsymbol{x}_H,\boldsymbol{x}_K\right)\right\},\ \boldsymbol{x}_H\in H,\ \boldsymbol{x}_K\in K$$

其中 $D(\boldsymbol{x}_H,\boldsymbol{x}_K)$表示 H 类中的样本 $\boldsymbol{x}_H$ 和 K 类中的样本 $\boldsymbol{x}_K$ 之间的欧氏距离，D_{HK}表示 H 类中所有样本与 K 类中所有样本之间的最小距离，如图5.4(a)所示。

如果 K 类由 I 和 J 两类合并而成，则

$$D_{HI}=\min\left\{D\left(\boldsymbol{x}_H,\boldsymbol{x}_I\right)\right\},\ \boldsymbol{x}_H\in H,\ \boldsymbol{x}_I\in I$$

$$D_{HJ}=\min\left\{D\left(\boldsymbol{x}_H,\boldsymbol{x}_J\right)\right\},\ \boldsymbol{x}_H\in H,\ \boldsymbol{x}_J\in J$$

得到递推公式 $D_{HK}=\min\left\{D_{HI},D_{HJ}\right\}$。

在图5.4(b)中，D_{HK}是 D_{HI}和 D_{HJ}中最小的，$D_{HK}=D_{HI}$。

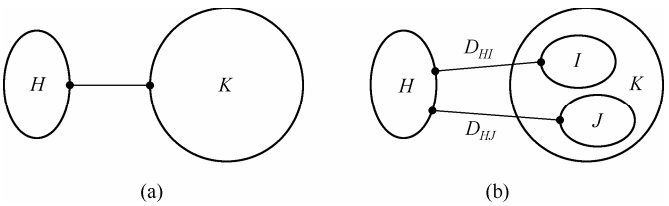


图 5.4 最短距离法

(1)最长距离法。与最短距离法类似，H 类和 K 类之间的距离定义为

$$D_{HK}=\max\left\{D\left(\boldsymbol{x}_H,\boldsymbol{x}_K\right)\right\},\ \boldsymbol{x}_H\in H,\ \boldsymbol{x}_K\in K$$

如果 K 类由 I 和 J 两类合并而成，则

$$D_{HI}=\max\left\{D\left(\boldsymbol{x}_H,\boldsymbol{x}_I\right)\right\},\ \boldsymbol{x}_H\in H,\ \boldsymbol{x}_I\in I$$

$$D_{HJ}=\max\left\{D\left(\boldsymbol{x}_H,\boldsymbol{x}_J\right)\right\},\ \boldsymbol{x}_H\in H,\ \boldsymbol{x}_J\in J$$

得到递推公式 $D_{HK}=\max\left\{D_{HI},D_{HJ}\right\}$。

(3)均值距离。设 H 和 K 是两个聚类，则两类间的均值距离定义为

$$D_{HK}=\sqrt{\frac{1}{n_H n_K}\sum_{\substack{i\in H\\ j\in K}}d_{ij}^2}$$

其中 d_{ij}^2 表示 H 类中任一样本 x_i 和 K 类中的任一样本 x_j 之间的欧氏距离的平方，n_H, n_K 分别是 H 类和 K 类中的样本数目。

如果 K 类由 I 类和 J 两类合并而成，则可以得到 H 类和 K 类之间距离的递推式，如下所示：

$$D_{HK}=\sqrt{\frac{n_I}{n_I+n_J}D_{HI}^2+\frac{n_J}{n_I+n_J}D_{HJ}^2}$$

【例 5.1】 已知 6 个二维样本 $X=\{x_i, i=1,2,\mathrm{L}\ ,6\}$，如图 5.5 所示，其中 x_1 = [0.2, 0.3]$^{\mathrm{T}}$, $x_2=[0.5,1]^{\mathrm{T}}$，$x_3=[3,3]^{\mathrm{T}}$，$x_4=[3.4,2.1]^{\mathrm{T}}$，$x_5=[4.1,3.5]^{\mathrm{T}}$，$x_6=[4.5,2.5]^{\mathrm{T}}$。试按照最短距离进行层次聚类。

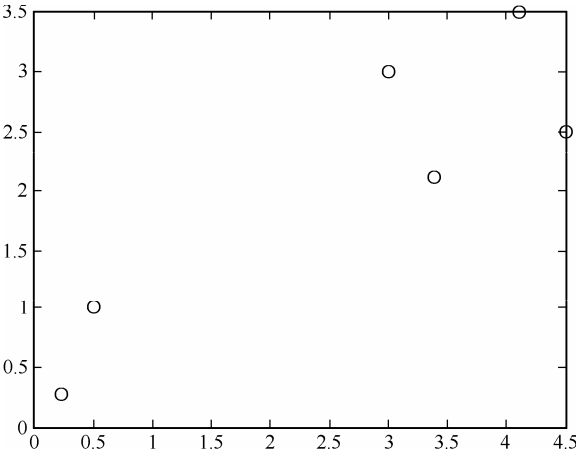


图 5.5　例 5.1 的样本

解：(1)初始将每个样本视为一类，得

$$G_1^{(0)}=\{x_1\}, G_2^{(0)}=\{x_2\}, G_3^{(0)}=\{x_3\}, G_4^{(0)}=\{x_4\}, G_5^{(0)}=\{x_5\}, G_6^{(0)}=\{x_6\}$$

计算各类间的欧氏距离，得到距离矩阵 $\boldsymbol{D}^{(0)}$，如下表所示。

$\boldsymbol{D}^{(0)}$	$G_1^{(0)}$	$G_2^{(0)}$	$G_3^{(0)}$	$G_4^{(0)}$	$G_5^{(0)}$	$G_6^{(0)}$
$G_1^{(0)}$	0	0.7616	3.8897	3.6715	5.0448	4.8301
$G_2^{(0)}$	0.7616	0	3.2016	3.1016	4.3829	4.2720
$G_3^{(0)}$	3.8897	3.2016	0	0.9849	1.2083	1.5811
$G_4^{(0)}$	3.6715	3.1016	0.9849	0	1.5652	1.1705
$G_5^{(0)}$	5.0448	4.3829	1.2083	1.5652	0	1.0770
$G_6^{(0)}$	4.8301	4.2720	1.5811	1.1705	1.0770	0

(2)将最短距离 0.7616 对应的类 $G_1^{(0)}$ 和 $G_2^{(0)}$ 合并为一类，得到新的分类：

$$G_{12}^{(1)}=\{G_1^{(0)}, G_2^{(0)}\}$$

$$G_2^{(1)}=\{G_2^{(0)}\}, G_3^{(1)}=\{G_3^{(0)}\}, G_4^{(1)}=\{G_4^{(0)}\}, G_5^{(1)}=\{G_5^{(0)}\}, G_6^{(1)}=\{G_6^{(0)}\}$$

计算各类间的欧氏距离，得到距离矩阵 $\boldsymbol{D}^{(1)}$，如下表所示。

$\boldsymbol{D}^{(1)}$	$G_{12}^{(1)}$	$G_3^{(1)}$	$G_4^{(1)}$	$G_5^{(1)}$	$G_6^{(1)}$
$G_{12}^{(1)}$	0	3.2016	3.1016	4.3829	4.2720
$G_3^{(1)}$	3.2016	0	0.9849	1.2083	1.5811
$G_4^{(1)}$	3.1016	0.9849	0	1.5652	1.1705
$G_5^{(1)}$	4.3829	1.2083	1.5652	0	1.0770
$G_6^{(1)}$	4.2720	1.5811	1.1705	1.0770	0

(3)将最短距离 0.9849 对应的类 $G_3^{(1)}$和 $G_4^{(1)}$合并为一类，得到距离矩阵 $\boldsymbol{D}^{(2)}$，如下表所示。

$\boldsymbol{D}^{(2)}$	$G_{12}^{(2)}$	$G_{34}^{(2)}$	$G_5^{(2)}$	$G_6^{(2)}$
$G_{12}^{(2)}$	0	3.1016	4.3829	4.2720
$G_{34}^{(2)}$	3.1016	0	1.2083	1.1705
$G_5^{(2)}$	4.3829	1.2083	0	1.0770
$G_6^{(2)}$	4.2720	1.1705	1.0770	0

(4)将最短距离 1.0770 对应的类 $G_5^{(2)}$和 $G_6^{(2)}$合并为一类，得到距离矩阵 $\boldsymbol{D}^{(3)}$，如下表所示。

$\boldsymbol{D}^{(3)}$	$G_{12}^{(3)}$	$G_{34}^{(3)}$	$G_{56}^{(3)}$
$G_{12}^{(3)}$	0	3.1016	4.2720
$G_{34}^{(3)}$	3.1016	0	1.1705
$G_{56}^{(3)}$	4.2720	1.1705	0

(5)将最短距离 1.1705 对应的类 $G_{34}^{(3)}$和 $G_{56}^{(3)}$合并为一类，得到距离矩阵 $\boldsymbol{D}^{(4)}$，如下表所示。

$\boldsymbol{D}^{(4)}$	$G_{12}^{(4)}$	$G_{3456}^{(4)}$
$G_{12}^{(4)}$	0	3.1016
$G_{3456}^{(4)}$	3.1016	0

设定一个距离阈值为 $D_{\mathrm{T}} = 2$，$\boldsymbol{D}^{(4)}$中的最小元素为 3.1016，超过给定阈值，则聚类结束。结果为

$$G_1 = \{x_1, x_2\}, G_2 = \{x_3, x_4, x_5, x_6\}$$

如果无阈值条件，继续聚类，最终全部样本归为一类。

上述层次聚类过程可用图5.6所示的分类树表示，左边的数据为类间的最短距离。

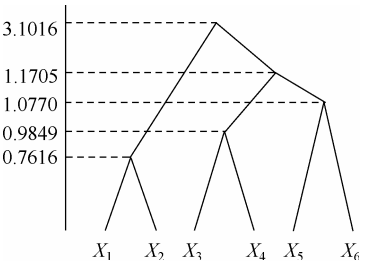


图 5.6 例 5.1 的层次聚类法分类树

5.2.3 *K*均值算法

K 均值算法(*K*-mean Algorithm，也称为 *C* 均值算法)是基于准则函数最优的聚类算法，它能够使各类样本到聚类中心的距离平方和取得极小值。

已知样本集合 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \mathrm{L}\ , \boldsymbol{x}_n\}$，$\boldsymbol{x}_j$是 d 维特征向量，$j = 1, 2, \mathrm{L}\ , n$；已知类别数 K 和初始聚类中心 $\boldsymbol{C}_i$；相似性测度可以采用欧氏距离；聚类准则采用误差平方和准则，其准则函数为

$$J=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\|\boldsymbol{x}-\boldsymbol{C}_i\|^2,\quad \boldsymbol{C}_i\text{是第 }i\text{ 类的聚类中心} \tag{5.21}$$

K 均值算法就是通过不断调整聚类中心，使得误差平方和准则函数 J 取得极小值。

K 均值算法如下：

(1)初始化：给定类别数 K，初始化聚类中心 $\boldsymbol{C}_i(l)$，$i=1,2,\mathrm{L}\ ,K,l=1$。

(2)第 l 次迭代的修正：逐个将样本 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_n\}$ 按照最小距离原则分配给 K 个聚类中心的某一个。若 $\|\boldsymbol{x}_j-\boldsymbol{C}_p(l)\|<\|\boldsymbol{x}_j-\boldsymbol{C}_i(l)\|$，$i,p=1,\mathrm{L}\ ,K,i\neq p$，则 $\boldsymbol{x}\in\boldsymbol{C}_p(l)$，$\boldsymbol{x}$ 是聚类中心为 $\boldsymbol{C}_p(l)$ 的样本集。

(3)计算新的聚类中心：

$$\boldsymbol{C}_i(l+1)=\frac{1}{N_i}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i(l)}\boldsymbol{x},\quad i=1,2,\mathrm{L}\ ,K$$

其中 N_i 为第 i 个聚类所包含的样本个数。

用均值向量作为新的聚类中心，可使准则函数

$$J_i=\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\|\boldsymbol{x}-\boldsymbol{C}_i\|^2,\quad i=1,2,\mathrm{L}\ ,K$$

最小。

在这一步要分别计算 K 个聚类的样本均值向量，所谓的 K 均值算法就由此特点得名。

(4)若 $\boldsymbol{C}_i(l+1)\neq\boldsymbol{C}_i(l)$，$i=1,2,\mathrm{L}\ ,K$，令 $l=l+1$，转(2)；将样本逐个重新分类，重复迭代计算。

若 $\boldsymbol{C}_i(l+1)=\boldsymbol{C}_i(l)$，$i=1,2,\mathrm{L}\ ,K$，算法收敛，计算完毕。

K 均值算法的计算复杂度是 $O(ndKl)$，其中 n 是样本数量，d 是样本维数，K 是类别数，l 是迭代次数。

K 均值算法是以确定类别数和选定的初始聚类中心为前提，使样本到其所在类中心的距离之和最小。分类结果受到类别数和初始聚类中心的影响，得到的聚类结果是局部最优的。但是该方法简单，聚类结果可以令人满意，因而应用比较普遍。

当类别数 K 未知时，在使用 K 均值算法时假设类别数逐步增加。可采用试探法，令 $K=2,3,4,\mathrm{L}$，对应算出准则函数 J，作出 J 与 K 的关系曲线(见图5.7)，J 随 K 的增加而逐步减少。通常情况下，当 J 下降变慢时，对应的 K 较为合适。如图5.7所示，从 5 至 6 下降较小，因而认为 5 较合适，故 K = 5。但并非所有情况都能找到这样的转折点，当无明显转折点时，可利用对该问题的先验知识分析选取合理的类别数。

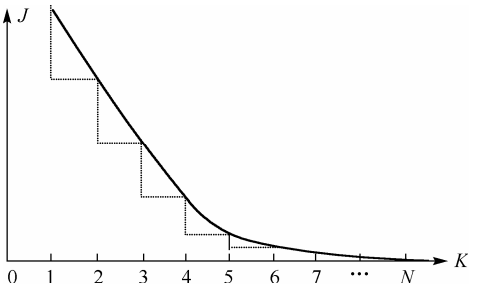


图 5.7　J 与 K 的关系曲线

初始聚类中心的选择，有以下几种方法：

(1)凭先验知识，将样本集大致分类，取代表点。

(2)将全部样本随机地分为 K 类，计算每类的中心，作为初始聚类中心。

(3) 以每个样本为中心，某个正数 r 为半径，在球内落入的点的个数作为密度，取最大密度点为 $\boldsymbol{C}_1(0)$，然后再找出与 $\boldsymbol{C}_1(0)$ 的距离大于 r 的次大密度作为新的聚类中心，依次选定。

(4) 选择给定样本集的前 K 个样本作为聚类中心。

(5) 从 $K-1$ 类问题得出 $K-1$ 个聚类中心，再找出一个最远点。

【例 5.2】 图5.8给出了 20 个二维样本，用 K 均值算法进行分类，取 $K=2$。

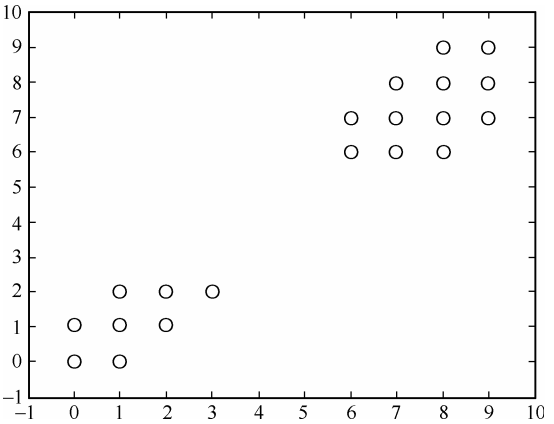


图 5.8 例 5.2 的 K 均值算法样本

解：

(1) 令 $K=2$，选择 $\boldsymbol{C}_1(1)=\boldsymbol{x}_1=[0,0]^{\mathrm{T}}$，$\boldsymbol{C}_2(1)=\boldsymbol{x}_2=[1,0]^{\mathrm{T}}$。

(2) 因为 $\|\boldsymbol{x}_1-\boldsymbol{C}_1(1)\|<\|\boldsymbol{x}_1-\boldsymbol{C}_i(l)\|$ 和 $\|\boldsymbol{x}_3-\boldsymbol{C}_1(1)\|<\|\boldsymbol{x}_1-\boldsymbol{C}_i(l)\|$，$i=2$，故 $X_1(1)=\{x_1,x_3\}$，$N_1=2$。同样，剩余的样本接近于 $\boldsymbol{C}_2(1)$，故 $X_2(1)=\{x_2,x_4,x_5,\mathrm{L}\ ,x_{20}\}$，$N_2=18$。

(3) 更新聚类中心，如图5.9所示。

$$\boldsymbol{C}_1(2)=\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{C}_1(1)}\boldsymbol{x}=\frac{1}{2}(x_1+x_3)=\begin{bmatrix}0\\0.5\end{bmatrix}$$

$$\boldsymbol{C}_2(2)=\frac{1}{N_2}\sum_{\boldsymbol{x}\in\boldsymbol{C}_2(1)}\boldsymbol{x}=\frac{1}{18}(x_2+x_4+x_5+\mathrm{L}\ +x_{20})=\begin{bmatrix}5.7\\5.3\end{bmatrix}$$

(4) 因为 $\boldsymbol{C}_j(2)\neq\boldsymbol{C}_j(1),j=1,2$，转到(2)。

(5) 因为 $\|\boldsymbol{x}_l-\boldsymbol{C}_1(2)\|<\|\boldsymbol{x}_l-\boldsymbol{C}_2(2)\|$，$l=1,2,\mathrm{L}\ ,8$，所以 $\boldsymbol{C}_1(2)=\{x_1,x_2,x_3,\mathrm{L}\ ,x_8\}$，$N_1=8$。又因为 $\|\boldsymbol{x}_l-\boldsymbol{C}_2(2)\|<\|\boldsymbol{x}_l-\boldsymbol{C}_1(1)\|,l=9,10,11,\mathrm{L}\ ,20$，故 $\boldsymbol{C}_2(2)=\{x_9,x_{10},x_{11},\mathrm{L}\ ,x_{20}\}$，$N=12$。

(6) 更新聚类中心：

$$\boldsymbol{C}_1(3)=\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{C}_1(2)}\boldsymbol{x}=\frac{1}{8}(x_1+x_2+\mathrm{L}\ +x_8)=\begin{bmatrix}1.2\\1.1\end{bmatrix}$$

$$\boldsymbol{C}_2(3)=\frac{1}{N_2}\sum_{\boldsymbol{x}\in\boldsymbol{C}_2(2)}\boldsymbol{x}=\frac{1}{18}(x_9+x_{10}+x_{11}+\mathrm{L}\ +x_{20})=\begin{bmatrix}7.7\\7.3\end{bmatrix}$$

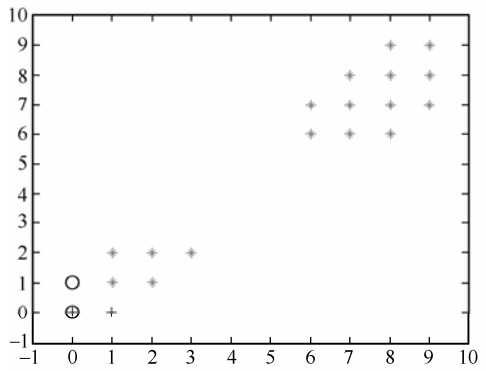
(7) 因为 $\boldsymbol{C}_j(3)\neq\boldsymbol{C}_j(2),j=1,2$，转到(2)。

(8) 本次分类结果与前一次迭代产生的结果相同，$\boldsymbol{C}_1(4)=\boldsymbol{C}_1(3)$，$\boldsymbol{C}_2(4)=\boldsymbol{C}_2(3)$。

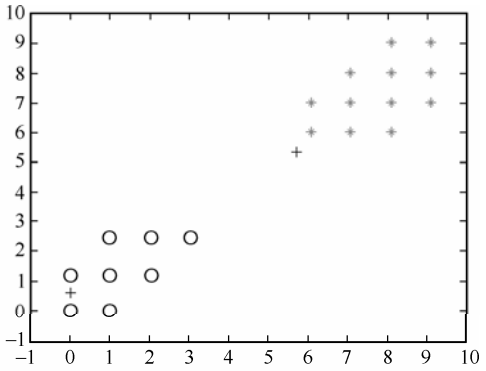
(9) 本次的聚类中心也与前一次的相同。

因为 $\boldsymbol{C}_j(4)=\boldsymbol{C}_j(3), j=1,2$，故算法收敛。聚类中心为

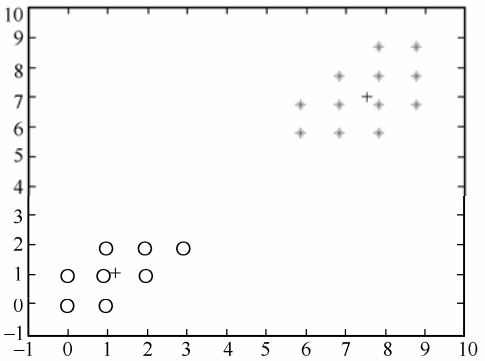
$$\boldsymbol{C}_1=\begin{bmatrix}1.2\\1.1\end{bmatrix},\quad \boldsymbol{C}_2=\begin{bmatrix}7.7\\7.3\end{bmatrix}$$



(a) 第 1 次迭代结果



(b) 第 2 次迭代结果



(c) 第 3 次迭代结果

图 5.9　例 5.2 的聚类过程

5.2.4　核聚类

在 K 均值算法中，仅用一个聚类中心点作为一类的代表，而一个点往往不能充分地反映出该类的样本分布结构；当类的分布是球状或近似球状时，才能有较好的效果。对于图5.10 的样本分布，它是各个分量方差不等的正态分布，K 均值算法的分类效果不好[10, 11, 12]。在图5.10 中，A 点依概率密度应属于 ω_1 类，但由于它离 ω_2 类的均值 $\boldsymbol{m}_2$ 更近，用 K 均值算法就会将 A 点分到 ω_2 类。

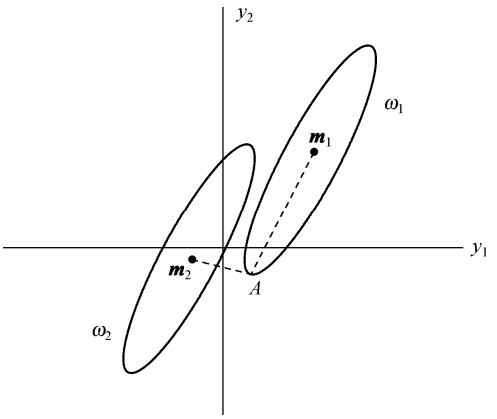


图 5.10　各分量方差不等的正态分布

如果已知各类样本分布，则可以利用它们进行聚类。已知样本集合 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_n\}$，类别数 c，基于样本与核的相似性测度准则函数为

$$J=\sum_{i=1}^{c}\sum_{\boldsymbol{x}\in\boldsymbol{\Gamma}_i} d(\boldsymbol{x},\boldsymbol{K}_i) \tag{5.21}$$

其中 $\boldsymbol{K}=\{\boldsymbol{K}_1,\boldsymbol{K}_2,\mathrm{L}\ ,\boldsymbol{K}_c\}$ 为核集，$\boldsymbol{\Gamma}=\{\boldsymbol{\Gamma}_1,\boldsymbol{\Gamma}_2,\mathrm{L}\ ,\boldsymbol{\Gamma}_c\}$ 为 $\boldsymbol{X}$ 划分为 c 类的子集，d 表示某种相似性测度。则聚类就是不断调整核集 $\boldsymbol{K}$ 和子集$\boldsymbol{\Gamma}$，最终使准则函数 J 取得极小值的过程。相应的算法如下：

(1) 初始化，将样本集 $\boldsymbol{X}$ 划分成 c 类，并确定每类的初始核 $\boldsymbol{K}_j$，$j=1,2,\mathrm{L}\ ,c$。

(2) 计算每个样本与所有核的相似性测度 $d(\boldsymbol{x}_j,\boldsymbol{K}_j)$，$j=1,2,\mathrm{L}\ ,n$，$i=1,2,\mathrm{L}\ ,c$，如果 $d(\boldsymbol{x}_j,\boldsymbol{K}_1)=\min\limits_{i=1,2,\mathrm{L}\ ,c} d(\boldsymbol{x}_j,\boldsymbol{K}_i)$，$j=1,2,\mathrm{L}\ ,n$，则 $\boldsymbol{x}_j\in\boldsymbol{\Gamma}_l$，将每个样本分到相应的类别中。

(3) 重新修正核 $\boldsymbol{K}_i$，$i=1,2,\mathrm{L}\ ,c$。若所有的核 $\boldsymbol{K}_i$ 都保持不变，则算法结束；否则，转到(2)。

实践证明，只要选择合适的核函数，核聚类算法通常都能取得合理的聚类，比较好地拟合不同样本的数据分布。实际上，K 均值算法是基于核 $\boldsymbol{K}_i$ 聚类算法的特例，是用样本均值替代核 $\boldsymbol{K}_i$。

为了进一步说明核聚类算法，下面介绍两种核函数和相似性测度。

(1) 正态分布函数为核函数

当已知某类的分布为正态分布时，可以用正态分布函数作为核函数：

$$\boldsymbol{K}_i(\boldsymbol{x}_k,\boldsymbol{V}_i)=\frac{1}{(2\pi)^{d/2}\left|\hat{\boldsymbol{\Sigma}}_i\right|^{1/2}}\exp\left[-\frac{1}{2}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}}_i)\right]$$

其中 $\boldsymbol{V}_i=(\hat{\boldsymbol{\mu}}_i,\hat{\boldsymbol{\Sigma}}_i)$，$\hat{\boldsymbol{\mu}}_i$ 为子集 $\boldsymbol{\Gamma}_i$ 的均值，$\hat{\boldsymbol{\Sigma}}_i$ 为 $\boldsymbol{\Gamma}_i$ 的协方差矩阵，d 是样本 $\boldsymbol{x}_k$ 的维数。

$$\hat{\boldsymbol{\mu}}_i=\frac{1}{n_i}\sum_{l=1}^{n_i}\boldsymbol{x}_l^{(i)},\quad i=1,2,\mathrm{L}\ ,c$$

$$\hat{\boldsymbol{\Sigma}}_i=\frac{1}{n_i}\sum_{l=1}^{n_i}(\boldsymbol{x}_l^{(i)}-\hat{\boldsymbol{\mu}}_i)(\boldsymbol{x}_l^{(i)}-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}},\quad i=1,2,\mathrm{L}\ ,c$$

相似性测度可定义为

$$d(\boldsymbol{x}_k,\boldsymbol{K}_i)=\frac{1}{2}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}(\boldsymbol{x}_k-\hat{\boldsymbol{\mu}}_i)+\frac{1}{2}\lg\left|\hat{\boldsymbol{\Sigma}}_i\right|$$

(2) 主轴核函数

当已知各类样本分别在相应的主轴附近分布时，如图 5.11 所示，第一类样本在 D_1 表示的主轴方向上集中，第二类样本在 D_2 表示的主轴方向上集中。在这种情况下，可定义核函数为

$$\boldsymbol{K}_j(\boldsymbol{x},\boldsymbol{V}_j)=\boldsymbol{U}_j{}^{\mathrm{T}}\boldsymbol{x}$$

式中 $\boldsymbol{U}_j=\{\boldsymbol{u}_1,\boldsymbol{u}_2,\mathrm{L}\ ,\boldsymbol{u}_{dj}\}$ 是样本协方差矩阵 $\hat{\boldsymbol{\Sigma}}_j$ 的 dj 个最大特征值所对应的本征向量系统。

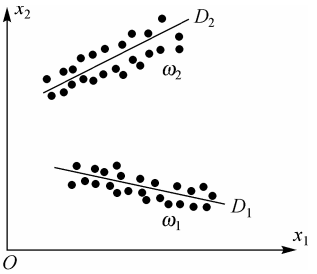


图 5.11 主轴核函数

任一样本 $\boldsymbol{x}$ 与一个轴 $\boldsymbol{U}_j$ 之间的相似程度可以用 $\boldsymbol{x}$ 与 ω_j 类主轴之间的欧氏距离的平方来度量：

$$d_L^2(\boldsymbol{x}, \boldsymbol{K}_j) = [(x - \hat{\boldsymbol{\mu}}_j) - \boldsymbol{U}_j \boldsymbol{U}_j^{\mathrm{T}}(x - \hat{\boldsymbol{\mu}}_j)]^{\mathrm{T}}[(x - \hat{\boldsymbol{\mu}}_j) - \boldsymbol{U}_j \boldsymbol{U}_j^{\mathrm{T}}(x - \hat{\boldsymbol{\mu}}_j)]$$

式中 $\hat{\boldsymbol{\mu}}_j$ 是ω_j类样本的均值向量。

5.2.5　ISODATA算法

ISODATA（Iterative Self-Organizing Data Analysis Techniques Algorithm，迭代自组织数据分析算法）[8-9]是基于准则函数最优的聚类算法，它与 K 均值算法有相似之处，即聚类中心同样是通过样本均值的迭代运算得到的。ISODATA 在迭代过程中，增加了产生和消除某些类别的方法，能自动合并和分裂类，因而有自动寻找类别数 K 的能力。ISODATA 的特点是计算简单，适用于识别致密聚类。

合并主要发生在某一类样本数太少的情况，或者两类聚类中心之间距离太小的情况。为此，设置每一类中最少样本数和两类聚类中心之间的最小距离参数。

分裂主要发生在某一类的某分量出现类内方差过大的现象时，适合将其分裂成两类，使类内方差比较合理。为此，设置类内某个分量方差的参数，用以决定是否需要将某一类分裂成两类。

ISODATA 算法中的参数如下：

K——要找的聚类中心数；

θ_N——每一类中至少应具有的样本个数；

θ_s——类内的样本标准差阈值；

θ_c——两个聚类中心之间的最小距离；

L——在一次迭代运算中可合并的最多对数；

I——允许迭代的次数。

ISODATA 算法如下：

(1) 初始化：任意选定 c 个聚类中心$\boldsymbol{z}_1(1), \boldsymbol{z}_2(1), \mathrm{L}\ , \boldsymbol{z}_c(1)$；定义算法参数$k, \theta_N, \theta_S, \theta_c, L, I$。其中 c 不一定等于所要求解的聚类中心数 K。

(2) 分配 N 个样本到 c 类中。按最近邻原则计算，若$\|\boldsymbol{x} - \boldsymbol{z}_i\| < \|\boldsymbol{x} - \boldsymbol{z}_j\|$，$i = 1, 2, \mathrm{L}\ , c$，$i \neq j$，则 $\boldsymbol{x} \in \boldsymbol{X}_i$，其中 $\boldsymbol{X}_i$ 表示分到聚类中心 $\boldsymbol{z}_i$ 的样本子集，N_i 为 $\boldsymbol{X}_i$ 中的样本数。

(3) 对任意 i，若 $N_i < \theta_N$，则去除 X_i，使 $c = c-1$，也就是将样本数比θ_N少的样本子集去除。

(4) 修正聚类中心 $\boldsymbol{z}_i$：

$$\boldsymbol{z}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in \boldsymbol{X}_i} \boldsymbol{x}\ ,\quad i = 1, 2, \mathrm{L}\ , c$$

(5) 计算 $\boldsymbol{X}_i$ 中样本与各个聚类中心间的平均距离：

$$\bar{d}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in \boldsymbol{X}_i} \|\boldsymbol{x} - \boldsymbol{z}_i\|\ ,\quad i = 1, 2, \mathrm{L}\ , c$$

(6) 计算总体的平均距离：

$$\bar{d} = \frac{1}{N} \sum_{i=1}^{c} N_i \bar{d}_i$$

其中 N 为样本集中的样本总数。

(7)判断分裂、合并及迭代运算步骤。

① 若是最后一次迭代，置 $\theta_c=0$，转到(11)，算法结束。

② 若 $c\leqslant\frac{K}{2}$，转到(8)，将已有的聚类分裂。

③ 若 $c \geqslant 2K$，或是偶次迭代，直接转到(11)。

④ 若②、③不满足，继续。

(8)计算标准差 σ_{ij}：

$$\sigma_{ij}=\sqrt{\frac{1}{N_i}\sum_{\boldsymbol{x}\in\boldsymbol{X}_i}(\boldsymbol{x}_{ik}-\boldsymbol{z}_{ij})^2}\ ,\quad i=1,2,\mathrm{L}\ ,d\ ;\ j=1,2,\mathrm{L}\ ,c$$

其中 d 是样本的维数，x_{ik} 是 $\boldsymbol{X}_k$ 中第 i 个样本的第 k 个分量，z_{ij} 是 $\boldsymbol{z}_j$ 的第 i 个分量。

$$\sigma_j=\{\sigma_{1j},\sigma_{2j},\mathrm{L}\ ,\sigma_{dj}\}^{\mathrm{T}}$$

(9)找出 σ_j 中的最大分量 $\sigma_{j\max}$，$j=1,2,\mathrm{L}\ ,c$。

(10)如果 $\sigma_{j\max}>\theta_s$，$j=1,2,\mathrm{L}\ ,c$，同时满足以下条件之一：

① $\overline{d_j}>d$ 和 $N_j>2(\theta_N+1)$

② $c\leqslant\frac{K}{2}$

则将 $\boldsymbol{X}_j$ 分成两类，出现两个新的聚类中心 $\boldsymbol{z}_j^+$ 和 $\boldsymbol{z}_j^-$，删去 $\boldsymbol{z}_j$，并使 $c=c+1$。在对应于 $\sigma_{j\max}$ 的 $\boldsymbol{z}_j$ 的分量上加上一个给定量 γ_j，而 $\boldsymbol{z}_j$ 的其他分量保持不变构成 $\boldsymbol{z}_j^+$；在对应于 $\sigma_{j\max}$ 的 $\boldsymbol{z}_j$ 的分量上减去 γ_j，而 $\boldsymbol{z}_j$ 的其他分量保持不变构成 $\boldsymbol{z}_j^-$。规定 γ_j 是 $\sigma_{j\max}$ 的一部分，$\gamma_j=\alpha\sigma_{j\max}$，$0<\alpha\leqslant1$。选择 α 的基本要求是，使任意样本到这两个新的聚类中心 $\boldsymbol{z}_j^+$ 和 $\boldsymbol{z}_j^-$ 之间有一个足够可检测的距离差别，但又不能太大。

如果完成分裂，迭代次数加 1，$l=l+1$，转到(2)。否则，继续进行(11)。

(11)计算全部的聚类中心的两两距离 d_{ij}：

$$d_{ij}=\left\|\boldsymbol{z}_i-\boldsymbol{z}_j\right\|,\quad i\neq j\ ;\ \ i,j=1,2,\mathrm{L}\ ,c$$

(12)比较：

如果 $d_{ij}\geqslant\theta_c$，转到(14)；否则，如果 $d_{ij}<\theta_c$，将 $d_{ij}<\theta_c$ 的值按升序排序，即 $d_{i1j1}<d_{i2j2}<\mathrm{L}<d_{iljl}$，$l\leqslant L$。

(13)从 d_{i1j1} 开始，逐对合并，算出新的聚类中心：

$$\boldsymbol{z}_l^*=\frac{1}{N_{il}+N_{jl}}[N_{il}z_{il}+N_{jl}z_{jl}],\quad l=1,2,\mathrm{L}\ ,L$$

删去 $\boldsymbol{z}_{il}$ 和 $\boldsymbol{z}_{jl}$，并使 $c=c-1$。注意，只允许一对对合并，并且一个聚类中心只能合并一次。

(14)迭代处理：

如果是最后一次迭代，$l=I$，算法结束。否则，

① 不修改参数，$l=l+1$，转到(2)。

② 需要人工参与修改参数，$l=l+1$，转到(1)。

每次回到算法的第一步或第二步就计为一次迭代。

【例 5.3】 如图 5.12 所示，有 8 个二维的样本 $\boldsymbol{X}=\{\boldsymbol{x}_i, i=1,2,\mathrm{L}\ ,8\}$，其中 $x_1=[0.5,0.5]^{\mathrm{T}}$，$x_2=[1,1]^{\mathrm{T}}$，$x_3=[1.5,2]^{\mathrm{T}}$，$x_4=[5,5]^{\mathrm{T}}$，$x_5=[5,3]^{\mathrm{T}}$，$x_6=[4,4]^{\mathrm{T}}$，$x_7=[5,4]^{\mathrm{T}}$，$x_8=[6,5]^{\mathrm{T}}$。用 ISODATA 算法进行聚类。

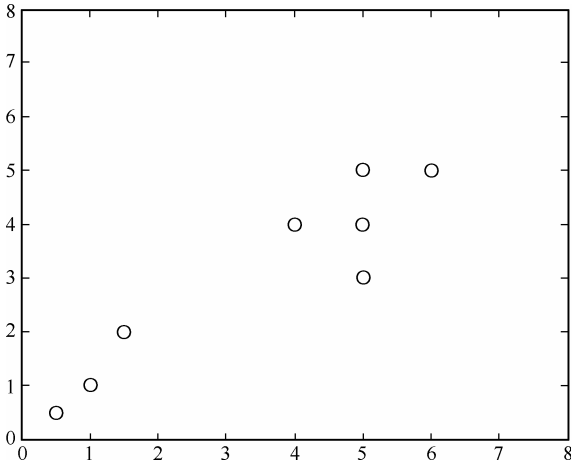


图 5.12　例 5.3 的样本

解：假设初始聚类中心数 $c=1$，$\boldsymbol{z}_1=\boldsymbol{x}_1=(1,1)^{\mathrm{T}}$。

第一次迭代

(1) 初始化：聚类中心数 $K=2$，每一类中至少应具有的样本个数 $\theta_N=1$，类内的样本标准差阈值 $\theta_s=1$，两个聚类中心之间的最小距离 $\theta_c=4$，在一次迭代运算中可合并的最多对数 $L=2$，允许迭代的次数 $I=4$。若分析的模式中无法得到先验信息，则任意选择这些参数，然后通过算法在逐次迭代中进行调整。

(2) 因为只有一个聚类中心 $\boldsymbol{z}_1$，所以 $\boldsymbol{X}_1=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_8\}$，$N_1=8$。

(3) 由于 $N_1>\theta_N$，故无子类要去除。

(4) 更新聚类中心 $\boldsymbol{z}_1$：

$$\boldsymbol{z}_1=\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}\boldsymbol{x}=\frac{1}{8}(\boldsymbol{x}_1+\boldsymbol{x}_2+\mathrm{L}\ +\boldsymbol{x}_8)=\begin{bmatrix}3.50\\3.0625\end{bmatrix}$$

(5) 计算 $\overline{d}_i$：

$$\overline{d}_1=\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}\|\boldsymbol{x}-\boldsymbol{z}_1\|=\left(\|\boldsymbol{x}_1-\boldsymbol{z}_1\|+\mathrm{L}\ +\|\boldsymbol{x}_8-\boldsymbol{z}_8\|\right)/8=2.4246$$

(6) 计算 $\overline{d}$，此时 $\overline{d}=\overline{d}_1=2.4246$。

(7) 因为这不是最后一次迭代，且 $c=\dfrac{K}{2}$ $(K=2,c=1)$，所以转到(8)。

(8) 求 $\boldsymbol{X}_1$ 的标准差：

$$\sigma_{11}=\sqrt{\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}(\boldsymbol{x}_{1k}-\boldsymbol{x}_{11})^2}=\sqrt{\frac{1}{8}[(0-3.38)^2+\mathrm{L}\ +(6-3.38)^2]}=2.0156$$

$$\sigma_{21}=\sqrt{\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}(\boldsymbol{x}_{2k}-\boldsymbol{x}_{21})^2}=\sqrt{\frac{1}{8}[(0-2.75)^2+\mathrm{L}\ +(5-2.75)^2]}=1.6286$$

$$\sigma_1=\begin{bmatrix}2.0156\\1.6286\end{bmatrix}$$

(9) 取 σ_1 的最大值 $\sigma_{1\max}=2.0156$。

(10) 因为 $\sigma_{1\max}>\theta_s$，$c=\dfrac{K}{2}$，则 $\boldsymbol{z}_1$ 分裂成两个新的聚类中心。

令 $\gamma_1=0.5\sigma_{1\max}=0.5\times2.0156\approx1.0078$，则 $\boldsymbol{z}_1^+=\begin{bmatrix}4.5078\\3.0625\end{bmatrix}$，$\boldsymbol{z}_1^-=\begin{bmatrix}2.4922\\3.0625\end{bmatrix}$。为方便起见，令 $\boldsymbol{z}_1^+,\boldsymbol{z}_1^-$ 分别为 $\boldsymbol{z}_1$ 和 $\boldsymbol{z}_2$，$c=c+1=2$，转到(2)。

第二次迭代

(2) 将样本重新分配给 $\boldsymbol{z}_1$ 和 $\boldsymbol{z}_2$，现在样本集为

$$\boldsymbol{X}_1=\{\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6,\boldsymbol{x}_7,\boldsymbol{x}_8\},\ N_1=5,\quad \boldsymbol{X}_2=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3\},N_2=3$$

(3) 因为 $N_1>\theta_N$ 和 $N_2>\theta_N$，故无子集要去除。

(4) 更新聚类中心：

$$\boldsymbol{z}_1=\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}\boldsymbol{x}=\begin{bmatrix}5.0\\4.2\end{bmatrix}$$

$$\boldsymbol{z}_2=\frac{1}{N_2}\sum_{\boldsymbol{x}\in\boldsymbol{X}_2}\boldsymbol{x}=\begin{bmatrix}1.0000\\1.1667\end{bmatrix}$$

(5) 计算 $\overline{d}_i$，$i=1,2$：

$$\overline{d_1}=\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}(\boldsymbol{x}-\boldsymbol{z}_1)=0.9001$$

$$\overline{d_2}=\frac{1}{N_2}\sum_{\boldsymbol{x}\in\boldsymbol{X}_2}(\boldsymbol{x}-\boldsymbol{z}_2)=0.6573$$

(6) 计算 $\overline{d}$：

$$\overline{d}=\frac{1}{N}\sum_{i=1}^{c}N_i d_i=\frac{1}{8}\sum_{i=1}^{2}(5\times0.9001+3\times0.6573)=0.8090$$

(7) 因为是偶次迭代，转到(11)。

(11) 计算两两距离 d_{12}：

$$d_{12}=\|\boldsymbol{z}_1-\boldsymbol{z}_2\|=5.0201$$

(12) $d_{12}>\theta_c(\theta_c=4)$，不发生合并。

(14) 因为不是最后一次迭代，所以需要判定是转到(1)，还是转到(2)。此例中：①已得到 $K=2$ 的聚类数；②聚类间的分离度大于样本分离的标准差；③每一个聚类中包括一定数

量的样本总数。因此，得出的聚类中心 $\boldsymbol{z}_1$ 和 $\boldsymbol{z}_2$ 具有代表性。下一次迭代不需要更改算法参数，于是转到(2)。

第三次迭代

(2)~(6)同第二次迭代，产生同样的结果。

(7)所有条件都不满足，继续进行计算。

(8)计算 $\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$ 的标准差：

$$\sigma_{11}=\sqrt{\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}(\boldsymbol{x}_{1k}-\boldsymbol{z}_{11})^2}=0.6325$$

$$\sigma_{21}=\sqrt{\frac{1}{N_1}\sum_{\boldsymbol{x}\in\boldsymbol{X}_1}(\boldsymbol{x}_{2k}-\boldsymbol{z}_{21})^2}=0.7483$$

$$\boldsymbol{\sigma}_1=\begin{bmatrix}0.6325\\0.7483\end{bmatrix}$$

$$\sigma_{12}=\sqrt{\frac{1}{N_2}\sum_{\boldsymbol{x}\in\boldsymbol{X}_2}(\boldsymbol{x}_{1k}-\boldsymbol{z}_{12})^2}=0.4082$$

$$\sigma_{22}=\sqrt{\frac{1}{N_2}\sum_{\boldsymbol{x}\in\boldsymbol{X}_2}(\boldsymbol{x}_{2k}-\boldsymbol{z}_{22})^2}=0.6236$$

$$\boldsymbol{\sigma}_2=\begin{bmatrix}0.4082\\0.6236\end{bmatrix}$$

(9)这里 $\sigma_{1\max}=0.7483$ 和 $\sigma_{2\max}=0.6263$。

(10) $\sigma_{1\max}<\theta_s,\sigma_{2\max}<\theta_s$，不满足分裂条件，继续计算。

(11)得到与上次迭代时相同的结果：$D_{12}=\|\boldsymbol{z}_1-\boldsymbol{z}_2\|=5.0201$。

(10)得到与上次迭代时相同的结果。

(12)无归并。

(13)除标准差计算外，在本次迭代中，无新的增加，转到(2)。

第四次迭代

(2)~(6)与上次迭代一样，产生同样的结果。

(7)因 $I=4$ 是最后一次迭代，置 $\theta_c=0$，转到(11)。

(11)得到与上次迭代时相同的结果：$D_{12}=5.0201$。

(12)得到与上次迭代时相同的结果。

(13)无归并发生。

(14)因 $I=4$ 是最后一次迭代，故算法结束。

5.3　聚类有效性

聚类分析是无监督分类方法，聚类算法都是以假定类别数目已知为前提，来通过迭代或某种搜索方法求得各个类别的聚类中心的。即使样本集没有自然的聚类，聚类算法也可以将

样本集分类。不同的聚类算法会得到不同的分类结果，如何才能确定分类结果是否适合样本集？这就需要对聚类结果进行评价的工具。对聚类算法的结果进行定量评价的方法，一般称为聚类有效性(Cluster Validity)。

在聚类分析中，类别数的选择是一个非常困难的问题，至今还没有令人满意的方法。在相关文献中，报道了大量关于聚类有效性的方法。类别数的有效性评价方法可分为如下3类[14]：

- **有效性指标**，包括 Silhouette 指标、Davies-Bouldin 指标、Calinski-Harabasz 指标等。
- 检测稳定聚类结构的**稳定性方法**。
- 模拟热力学系统的**系统演化方法**。

这些方法都有各自的优点，例如有效性指标容易使用，稳定性方法能处理更复杂的数据，系统演化方法能提供各聚类之间的距离或可分离程度信息。

1．平均 Silhouette(Sil)指标

设 $a(\boldsymbol{x})$ 为聚类 $\boldsymbol{C}_j$ 中的样本 $\boldsymbol{x}$ 与类内所有其他样本的平均不相似度或距离，$d(\boldsymbol{x}, \boldsymbol{C}_i)$ 为样本 $\boldsymbol{x}$ 到另一个聚类 $\boldsymbol{C}_i$ 的所有样本的平均不相似度或距离，则 $b(\boldsymbol{x}) = \min\{d(\boldsymbol{x}, \boldsymbol{C}_i)\}, i = 1, 2, \text{L}$，$k$(类别数)，$i \neq j$。Sil 指标计算每个样本与同一聚类中样本的不相似度，以及与其他聚类中样本的不相似度，其每个样本 $\boldsymbol{x}$ 的计算公式如下：

$$\text{Sil}(\boldsymbol{x}) = \frac{b(\boldsymbol{x}) - a(\boldsymbol{x})}{\max\{a(\boldsymbol{x}), b(\boldsymbol{x})\}}$$

一般以一个样本集的所有样本的平均 Sil 值来评价聚类结果的质量，Sil 指标越大表示聚类质量越好，其最大值对应的类别数作为最优的聚类个数。

2．Davies-Bouldin(DB)指标

DB 指标是基于样本的类内散度与各聚类中心间距的测度，进行类数估计时其最小值对应的类数作为最优的聚类个数。DW_i 表示聚类 $\boldsymbol{C}_i$ 的所有样本到其聚类中心的平均距离，DC_{ij} 表示聚类 $\boldsymbol{C}_i$ 和聚类 $\boldsymbol{C}_j$ 中心之间的距离，则 DB 指标如下：

$$\text{DB}(k) = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1\sim k, j\neq i}\left(\frac{DW_i + DW_j}{DC_{ij}}\right)$$

3．Calinski-Harabasz(CH)指标

CH 指标(伪 F 统计量)是基于全部样本的类内离差矩阵与类间离差矩阵的测度，其最大值对应的类别数作为最优的聚类个数。CH 指标为

$$\text{CH}(k) = \frac{\text{tr}\boldsymbol{SB}(K)/(K-1)}{\text{tr}\boldsymbol{SW}(K)/(n-K)}$$

其中，$\boldsymbol{SW}_i$ 为类 $\boldsymbol{C}_i$ 的离差矩阵，且 $\boldsymbol{SW}(k) = SW_1 + SW_2 + SW_k$，$\boldsymbol{SB}(K)$ 为 K 个聚类的类间离差矩阵，tr$\boldsymbol{SW}$ 表示矩阵 $\boldsymbol{SW}$ 的迹。

4．稳定性方法

稳定性方法对一个样本数据集进行两次重采样产生 2 个样本数据子集，再用相同的聚类算法对 2 个样本数据子集进行聚类，产生 2 个具有 K 个聚类的聚类结果，计算 2 个聚类结果的相似度的分布情况。2 个聚类结果具有高的相似度说明 K 个聚类反映了稳定的聚类结构，其相似度可以用来估计聚类个数。

5．系统演化方法

系统演化方法将一个样本数据集视为伪热力学系统，当样本数据集被划分为 K 个聚类时，称系统处于状态 K。系统由初始状态 $K = 1$ 出发，经过分裂过程和合并过程，系统将演化到它的稳定平衡状态 Ko，其所对应的聚类结构决定了最优类数 Ko。系统演化方法能提供关于所有聚类之间的相对边界距离或可分程度，它适用于明显分离的聚类结构和轻微重叠的聚类结构。

有效性指标方法均是基于数据的类内与类间的离散矩阵或(不)相似度的明显差别进行统计计算的。但对于 2 个聚类很靠近或重叠的情况，会有许多样本的类内相似度与类间相似度相差不大，这种不明显的差别会使有效性指标的性能变差。而稳定性方法对重采样的 2 个样本数据子集进行相同的聚类，则靠近或重叠的聚类只表明 2 次聚类时的相同条件，并不影响性能。系统演化方法则分析任意 2 个聚类之间的相对边界距离或可分程度，因此，能判别靠近或轻微重叠的聚类。

在实际应用中，确定类别数的一种简单方法是在数据的低维表示中观察数据，可以使用线性和非线性投影方法将高维空间的数据降维。当使用准则函数最优进行聚类时，通常假设类别数分别是 1, 2, 3 等情况进行聚类，观察准则函数与类别数之间的关系，如图5.7所示，找到拐点确定最佳类别数。但是，不是所有情况都能找到拐点的。

可以使用统计检验方法，进行聚类有效性的判定在给定类别数 K 的条件下，得到最佳聚类后，对所得到的 K 个样本子集分别做单峰分布的 α 显著性检验，只要还有一个子集不满足显著性检验，则说明仍存在可分性，令 $K = K + 1$ 重新进行聚类，直到有 K 个子集均不具备类可分性，此时对应的 K 为最佳类别数。这种方法属于系统演化方法。

习题 5

5.1　简述监督分类方法和无监督分类方法的区别。

5.2　证明欧氏距离满足三角不等式。

5.3　证明总离散度矩阵等于类内离散度矩阵与类间离散度矩阵之和，即 $\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B$。

5.4　已知 6 个二维样本 $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, 2, \mathrm{L}\ , 6\}$，其中 $\boldsymbol{x}_1 = [0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2 = [0, 1]^{\mathrm{T}}$, $\boldsymbol{x}_3 = [1, 1.5]^{\mathrm{T}}$, $\boldsymbol{x}_4 = [4, 3]^{\mathrm{T}}$, $\boldsymbol{x}_5 = [4.5, 3]^{\mathrm{T}}$, $\boldsymbol{x}_6 = [5, 4]^{\mathrm{T}}$。试按照最短距离进行层次聚类。

5.5　已知 13 个二维样本 $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, 2, \mathrm{L}\ , 13\}$，其中 $\boldsymbol{x}_1 = [0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2 = [0, 1]^{\mathrm{T}}$, $\boldsymbol{x}_3 = [1, 0]^{\mathrm{T}}$, $\boldsymbol{x}_4 = [0.5, 4]^{\mathrm{T}}$, $\boldsymbol{x}_5 = [1, 3]^{\mathrm{T}}$, $\boldsymbol{x}_6 = [1, 5]^{\mathrm{T}}$, $\boldsymbol{x}_7 = [1.5, 4.5]^{\mathrm{T}}$, $\boldsymbol{x}_8 = [6, 4]^{\mathrm{T}}$, $\boldsymbol{x}_9 = [6.5, 5]^{\mathrm{T}}$, $\boldsymbol{x}_{10} = [7, 4]^{\mathrm{T}}$, $\boldsymbol{x}_{11} = [7.5, 7]^{\mathrm{T}}$, $\boldsymbol{x}_{12} = [8, 6]^{\mathrm{T}}$, $\boldsymbol{x}_{12} = [8, 7]^{\mathrm{T}}$。用 K 均值算法进行分类，分别取 $K = 2$ 和 $K = 4$。

5.6 用 K 均值算法对鸢尾属植物(Iris)样本数据进行分类，取 $K=3$。

5.7 对习题 5.5 中的样本集 $\boldsymbol{X}$，用 ISODATA 算法进行聚类分析。

5.8 用 ISODATA 算法对鸢尾属植物(Iris)样本数据进行聚类分析，并与习题 5.6 的结果进行对比分析。

参考文献

[1] Johnson S. C. *Hierarchical clustering schemes*, Psychometrika, 1967, 32: 241-254.

[2] Wallace C. S. , Boulton D.M. *An information measure for classification*, Computer Journal, 1968, 11: 185.194.

[3] Everitt B., Landau S., Leesse M. *Cluster Analysis*, Arnold, 2001.

[4] Sergios Theodoridis, Konstantinos Koutroumbas. *Pattern Recognition* (Third Edition), 北京：电子工业出版社，2006.

[5] 温熙森. 模式识别与状态监控, 北京：科学出版社，2007: 228.

[6] Lioy S. P., *Least squarea quantization in PCM*. IEEE Transaction on Information Theory, 1982, 28(2).

[7] Everitt B., Landau S., Leesse M. *Cluster Analysis*. Arnold, London, 2001.

[8] Venkateswarlu N B, Raju PSVSK. *Fast ISODATA Clustering Algorithms*. Pattern Recognition, 1992, 25(23): 335.342.

[9] Carman CS, Merickel MB. *Supervising ISODATA with an Information Theoretic Stopping Rule*. Pattern Recognition, 1990, 23(1/2): 185.197.

[10] 边肇祺, 张学工. 模式识别. 北京：清华大学出版社, 2000: 239-241.

[11] 钟珞, 潘昊. 模式识别. 武汉：武汉大学出版社, 2006: 117-118.

[12] 孙即祥. 现代模式识别(第二版). 北京：高等教育出版社, 2008: 49-50.

[13] 齐敏, 李大健, 郝重阳. 模式识别导论. 北京：清华大学出版社, 2009: 14-36.

[14] 王开军, 李健, 张军英, 过立新. 聚类分析中类数估计方法的实验比较. 计算机工程. 2008.34(9): 198-199.

第 6 章　人工神经网络

人工神经网络(Artificial Neural Network, ANN)是一门活跃的边缘性交叉学科，它作为一门新思想得到了广泛传播。人工神经网络理论是巨量信息并行处理和大规模并行计算的基础，可用来描述认知、决策及控制的智能行为。它的中心问题是智能的认知和模拟。人工神经网络与生物神经网络(Biological Neural Network, BNN)并不完全相同，本章神经网络指的是人工神经网络。

6.1　人工神经网络的构成

人工神经网络是由大量神经元(处理单元)互相连接而成的网络。为了模拟大脑的基本特性，在现代神经科学研究的基础上，人们提出了人工神经元网络模型。人工神经网络并没有完全地真正反映大脑的功能，它只是对生物神经网络进行某种抽象、简化和模拟。人工神经网络的信息处理由神经元之间的相互作用来实现；知识与信息的存储表现为网络元件互连间分布式的物理联系；人工神经网络的学习和识别决定于各神经元连接权的动态演化过程。

6.1.1　神经元的结构模型

不论什么样的神经网络模型，其中一个最小的信息处理单元就是神经元。到目前为止，人们已经建立了数百种人工神经元模型[1]，最常用的神经元模型仍然是最早提出的 MP 模型。神经元是一个多输入单输出的信息处理单元，是神经元网络的基本处理单元，其一般的结构模型如图6.1 所示。其中 u_i 为神经元 i 的内部状态；θ_i 为阈值；x_i 为输入信号；w_{ij} 表示神经元 i 与神经元 j 的连接权值，其正、负分别表示兴奋和抑制；s_i 表示某一外部输入的控制信号。

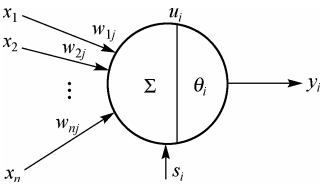


图 6.1　神经元模型

神经元模型常用一阶微分方程来描述：

$$\begin{aligned}\frac{\mathrm{d}u_i}{\mathrm{d}t} &= -u_i(t) + \sum_j w_{ij}x_j(t) - \theta_i \\ y_i(t) &= f(u_i(t))\end{aligned} \tag{6.1}$$

神经元的输出由函数 f 表示。作用函数反映了神经元的特征，根据作用函数不同，可将神经元分为多种类型。常用以下函数表达其非线性特征。

(1)线性函数：

$$f(u_i) = u_i \tag{6.2}$$

(2)半线性函数：

$$f(u_i) = \begin{cases} 0, & u_i < 0 \\ u_i, & u_i \geq 0 \end{cases} \tag{6.3}$$

(3)分段线性函数：

$$f(u_i)=\begin{cases}-1, & u_i<-t\\ u_i, & -t\le u_i<t\\ 1, & u_i\ge t\end{cases} \tag{6.4}$$

(4)硬限函数：

$$f(u_i)=\begin{cases}1, & u_i\ge 0\\ -1, & u_i<0\end{cases} \tag{6.5}$$

(5)Sigmoid 函数：

$$f(u_i)=\mathrm{th}(u_i)=\frac{1}{1+\exp(-u_i)} \tag{6.6}$$

上述神经元作用函数的取值范围有时会略做改变，但并不改变相应神经网络的性质。例如，分段线性函数的取值范围也可为[0, 1]，硬限函数的取值也可为{0, 1}。

神经网络就是由多个神经元加权连接而成的网络。虽然单个神经元只能进行十分简单的信息处理，但多个神经元连接而成的网络却具有强大的计算能力。神经网络计算表现为神经元之间的互相作用。改变神经元之间的连接方式和连接强度就可以改变神经网络的计算效果，其中，两个神经元之间的连接强度大小由一个实数表示，该实数称为连接权值。神经元之间的连接形式和连接权值通常由神经网络学习过程决定。根据神经元类型、神经元连接方式和学习方式的不同，人们设计形成了各种不同的神经网络模型。

神经网络可以视为如图6.2所示的信息处理系统，每当网络接收到外部信息时，便输出一个经过处理后的结果。人们可以通过修改网络连接方式和连接权值，使网络具有某种特殊的信息处理能力，但却不能控制网络的每一个计算步骤。

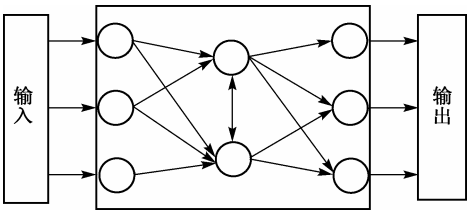


图 6.2 神经网络的信息处理

神经网络具有如下的计算特征：

(1)每个神经元独立于其他神经元进行工作，且每个神经元在某时刻的输出信息只依赖于该时刻来自与其连接的其他神经元的有效信息。

(2)每个神经元只能对来自局部的信息进行处理，一个神经元只能处理与其连接的神经元发送来的信息，不连接的神经元状态信息，不影响其处理进程。

(3)大量的连接必然带来信息的分布式冗余表示。

前两个特征使神经网络具备了大规模并行计算能力，后一个特征则使得神经网络具备了很好的容错和泛化能力(或称推广能力)。这种计算特性使得神经网络在下述三种基本情形下具有特别的优势：

(1)将大量数据根据某种属性分为较少的类，或利用大量数据进行具有较少可能结果的决策。例如在图像、声音信号处理时经常遇到这种情况。

(2)某些非线性函数映射事先并不知道，且会随环境适当变化，须自动在实践中获得。在机器人控制中，这种情况是常见的。

(3)某些组合优化问题需要在应用时实时求近似最优解，如计算机处理中的任务排工、网络通信中的路由计算等。

6.1.2　人工神经网络的连接方式

一般神经网络具有严格的分层结构形式，一个神经网络分为几层，每层有多少个神经元，层内和层间神经元是怎样连接的，称为神经网络的拓扑结构。神经网络的结构主要是指构成网络的各神经元之间相互连接的逻辑关系而不是空间排列关系。尽管神经网络是由大量的神经元组成的，但是组成网络的神经元并不需要空间位置的描述，各神经元之间也没有空间距离的概念。在描述和理解神经网络的逻辑结构时有两个概念非常重要，它们是“层”和“反馈”。

“层”是指可以进行并行操作的神经元集合。同一层中的所有神经元因其具有相同的输入输出关系而在网络中处于等同的地位。相同输入是指同一层中的所有神经元的输入向量完全相同，但是不同神经元的权值向量却可以不同，因此对于同一层中的神经元来说它们的净输入不相同。即同一层的神经元在同一时刻所处理的信息是相同的，但是对相同的信息处理的结果却不一定相同。正是因为处理的信息相同，才能使它们可以“互不干扰”地同时执行，从而实现并行操作。而相同输出则是指同一层的神经元其输出的目的相同：要么成为另一层神经元的输入，要么成为神经网络的系统输出。如果某一层神经元输出不再构成其他层的输入，而是直接成为神经网络的系统输出，那么称这个层为输出层；如果该层中神经元的输出构成其他层的输入，则称其为隐层；而为网络中第一个隐层提供输入的集合有时也称为输入层。只有输出层的网络称为单层网络，带有隐层的网络统称为多层网络。为了简洁和直观，在画神经网络结构示意图时通常会把同一层的神经元排列在一起。同一层的神经元虽然具有相同的输入输出结构，但是不一定具有相同的传输函数。也就是说，它们虽然同时处理相同的信息，但是处理的方式可以完全不同，这也是神经网络具有强大计算功能的原因之一。

“反馈”是使神经网络具有动力学行为的一种途径。如果神经网络中信息的流向不再是由输入层“直线”地到输出层，而是出现“折回”现象，就说网络中有反馈结构。两种情况可以产生反馈，一种是后一层神经元的输出成为前层神经元的输入，另一种则是同一层内的神经元之间相互连接。通常所说的反馈网络主要是指由后者构成的网络。反馈神经网络是一个典型的非线性动力系统，它比前馈网络具有更强的计算能力和更广泛的应用前景。一个典型的代表就是 Hopfield 神经网络模型，它是一个单层神经网络，其中每一个神经元都与其他所有神经元相连接。

根据连接方式的不同，神经网络的神经元之间的连接有如下几种形式。

(1) 前馈网络

前馈网络如图6.3所示，神经元分层排列，分别组成输入层、中间层(隐层)和输出层。每一层的神经元只接受来自前一层神经元的输入，后面的层对前面的层没有信号反馈。

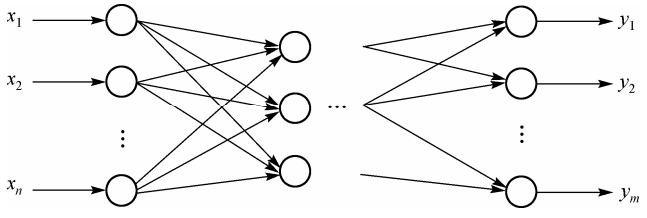


图 6.3　前馈网络

输入模式经过各层次的顺序传输，最后在输出层上得到输出。感知器和误差反向传播算法所采用的网络属于前馈网络类型。

(2)有反馈的前向网络

有反馈的前向网络如图6.4所示，从输出层到输入层有信息反馈。这种网络可以用来存储某种模式序列，如神经认知机即属此类。

(3)层内有相互结合的前馈网络

层内有相互结合的前馈网络如图6.5所示，通过层内神经元的相互结合，可以实现同一层内神经元之间的横向抑制或兴奋机制。这样可以限制每层内能同时动作的神经元数，或者把每层内的神经元分成若干组，让每组作为一个整体来运作。例如，可以利用横向抑制机理把某层内具有最大输出的神经元挑选出来，而抑制其他神经元处于无输出的状态。

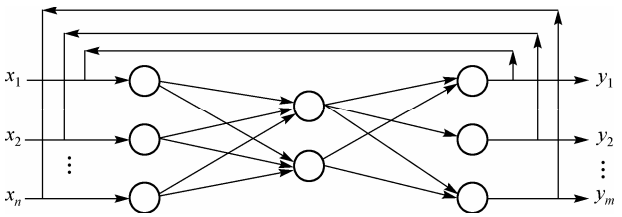


图 6.4 有反馈的前馈网络

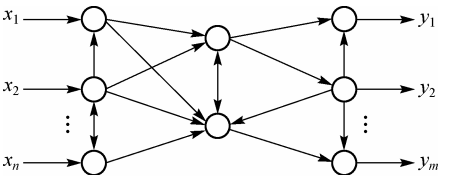


图 6.5 相互结合型网络

学习能力是神经网络最有吸引力的特征之一。学习方法的好坏对于神经网络在应用中的实际计算效果具有十分显著的影响。有关神经网络学习的若干概念如下。

1．输入与输出模式

神经网络的任何操作均离不开数据。输入和输出模式通常是以向量形式表示的神经网络的输入数据和输出结果。某些神经网络并不需要输入模式和输出模式同时存在，这种只需单一模式的网络通常又称为自联想网络，同时需要输入和输出模式的网络称为异联想网络。通常在实际应用中，模式表达了来自实际物理模型并带有不同属性的实际数据。输入、输出模式的选择，对神经网络的实际处理效果有很大影响。参与神经网络计算的数据模式称为样本模式或训练样本。

2．连接权值

神经元之间的连接强度由一个实数来表达。因此神经网络的拓扑结构可以表示为一个边带权的有向图，设某网络由 n 个神经元组成，并分别编号为$1, 2, \mathrm{L}\ , n$，则由节点 i 到节点 j 的连接权值记为 w_{ij}。神经网络学习算法的基本问题是确定神经网络的连接权值。

3．处理单元

处理单元即神经元，是神经网络中最基本的信息处理单位。神经元所处理的只是来自与其连接的其他神经元送来的信号。目前人们已经提出了许多神经元的信号处理方法。最常用的神经元信号处理过程分为两步，如下所示。

(1)将外来输入信号进行加权求和，这一步也体现了连接权值的作用：

$$u_i = \sum_{j=1}^{n} w_{ij} x_j - \theta_i \tag{6.7}$$

(2)将加权求和的结果做非线性变换后向其他神经元输出：

$$y_i = f(u_i) \tag{6.8}$$

神经网络结构确定后，神经网络的学习就是计算神经元网络之间的连接权值。

6.1.3　神经网络模型分类

按照网络的性能可分为离散型和连续型神经网络，又可分为确定型和随机型神经网络。按照网络结构可分为前馈和反馈神经网络。按照学习方式可分为监督学习和无监督学习神经网络。按照连接突触性质可分为一阶关联网络和高阶非线性关联网络。按照对生物神经系统的不同组织层次模拟，神经网络模型又可分为神经元层次模型、组合式模型、网络层次模型、神经系统层次模型、智能模型等。

一些有代表性的神经网络模型如下[2]。

(1)自适应共振理论(Adaptive Resonance Theory, ART)：包括 ART1 和 ART2，可对多个复杂的二维模式进行自组织、自稳定的大规模并行处理。ART1 用于二进制离散输入，ART2 用于连续信号输入，主要用于模式识别。缺点是对转换、失真及规模变化较敏感。

(2)雪崩网络(Avalanche Network)：主要用于学习、识别和重演时空模式的神经网络，如连续语言识别和教学机器人。缺点是连接权值调节较为困难。

(3)双向联想存储器(Bidirectional Associative Memory, BAM)：由相同神经元构成的双向联想记忆式单层网络，具有学习功能。缺点是存储容量较小且需要编码。

(4)BP 网络：多层前馈神经网络，采用误差反向传播算法学习，是使用较为广泛的网络。可用于语言综合、语音识别、自适应控制等。缺点是对于稍复杂的网络，训练时间长。

(5)Boltzman 机/Cauchy 机：用噪声过程代替代价函数的全局极小值，主要用于模式识别。缺点是训练时间长，且有噪声。

(6)盒中脑模型：具有最小均方差的单层自联想网络，可用于从数据库中提取知识。缺点是仅为单步决策。

(7)Hopfield 网络：由相同元件构成的单层反馈自联想网络。缺点是需要对称连接，存储容量较小。

(8)Madaline：具有最小方差学习功能的线性网络，它是 Adaline 的扩展，具有较强的学习功能，主要用于自适应控制。缺点是输入输出仅为线性关系。

(9)神经认知机：多层结构的字符识别网络。缺点是需要大量处理元件和连接。

(10)感知器：由硬限函数神经元组成的多层神经网络。单层感知器目前已很少使用，多层感知器的学习算法是这类网络实用化的关键。

(11)特征映射网络：描述某种最优映射，可在工作中自动抽取数据特征。

(12)细胞神经网络：单层反馈神经网络，神经元的作用函数采用分段线性函数。主要用于模式识别、文字识别与噪声控制。

(13)RECURRENT 网络：多层反馈神经网络，用于连续信号识别与处理。

6.1.4 神经网络学习规则

对于神经网络的学习问题，若神经网络结构确定后，学习算法的任务就是计算网络神经元之间的连接权值。按照连接权值的计算规则，可将目前常见的神经网络学习算法分类简述如下[2]。

(1) Hebb 规则学习

D. O. Hebb 提出了生物神经元突触权值的变化规则[3]：若一个神经元 A 是兴奋的，且引起与其相连接的另一个神经元 B 也兴奋时，则 A 和 B 之间的连接强度增加。这种网络所能记忆的模式数目受到网络输入模式维数的限制[4-6]。Hopfield 于 1982 年提出了 Hopfield 神经网络模型，他利用能量函数的思想证明了网络稳定性[7, 8]。Hopfield 模型是一种单层的反馈神经网络，连接权值不加限制，而神经元输出值限制为{0, 1}或{ – 1, +1}，当用做联想存储器时，连接权值的计算方法同样采用了 Hebb 规则。后来的分析表明以 Hebb 规则学习为基础的联想记忆网络所能存储的模式数目也是有限的。若网络的连接权值也限制为只取{0, 1}值，则网络的连接权值计算采用从 Hebb 规则演化出来的逻辑计算规则[9-12]。这种网络所能存储的信息量较大，但需要对二进制向量稀疏编码。

δ 规则学习也可以用 Hebb 规则解释，因此是 Hebb 规则学习的特殊形式。在前馈网络学习时，由某模式得到网络实际输出和外部输出的差值δ，网络连接权值恰好按照δ的一定比例进行修正。单层感知器网络和 BP 网络的学习算法均是采用δ规则进行权值计算的。

(2) 因素元件学习

某些神经网络的学习算法可以产生一组有关输入模式的因素元件作为网络的连接权值。一组数据的因素元件是由模式集合的协方差矩阵求得最小正交向量集合而得到的。求得基本集合后，可以利用线性变换得到所有的向量。

(3) 竞争学习

竞争学习通常是一个两步过程，第一步，神经元根据输入模式和现有状态进行竞争，竞争结果只有一个神经元获胜；第二步，取胜神经元修正连接权值，权值向量修正的方向是输入模式向量的方向。竞争学习有许多变种算法，如自组织特征映射网络[13, 14]、ART1 和 ART2 均采用竞争学习规则[15, 16]。近年来人们已将基于竞争学习的神经网络应用于各种实际问题中，大量实验表明竞争学习的收敛性质，但这类学习算法收敛性的理论证明仍未得到彻底解决。

(4) 最小–最大学习

最小–最大学习分类器是模糊神经网络分类器，系统利用由两个向量组成的二元组表示一类模式[18]。每一类模式均由一个神经元输出表示，一个神经元由两个连接权值向量 $\boldsymbol{v}_j$ 和 $\boldsymbol{w}_j$ 表达，$\boldsymbol{v}_j$ 称为最小向量，$\boldsymbol{w}_j$ 称为最大向量。每个神经元的最小权值向量和最大权值向量分别通过最小和最大过程计算得到。一般最小向量和最大向量的每个分量均限制在 0 和 1 之间，最小向量和最大向量的每个分量的值界定了一个超立方体区域，若网络输入向量位于该区域中的程度由一个模糊成员函数确定，则相应的最小最大向量就形成了一个模糊集[6, 18]。

(5) 误差修正学习

误差修正学习用于多层前馈神经网络的学习过程。若多层前馈网络的神经元采用连续的神经元作用函数，则这种网络就可以采用误差修正法计算其连接权值。最典型也最著名的误差修正学习算法是 BP 网络[19, 20]。这种学习算法需要首先设计一个误差函数，网络连接权值

是该函数的变量，连接权值的计算过程是误差函数的减小过程。根据误差函数的构造方法和神经元信息处理方法的不同，人们设计了多种类型的前馈网络误差修正学习算法。

(6)随机学习

随机神经网络模型的神经元输出值是根据概率分布函数随机地在一定范围内取值的。随机学习利用随机处理、概率和能量关系调整网络的连接权值。利用随机方法模拟系统的能量下降过程，由此收集每个神经元的评价值，并根据该评价值随机调整连接权值，这是随机学习的基本步骤。这种方法又称为模拟退火算法[21]。Ackley 等[22]于 1985 年提出的 Boltzman 机首次采用模拟退火算法实现了学习过程。模拟退火学习算法可在理论上保证求得与样本模式达到最优匹配的网络连接权值，但在实际应用中，该算法的学习速度太慢。

6.2　多层前馈网络学习算法

感知器网络[23-25]是最早提出的前馈神经网络。Widrow 与 Hoff 给出了单层感知器的学习算法[27]，该算法在固定网络结构的基础上利用δ规则计算神经网络的连接权值，固定网络结构就是固定单层网络的神经元个数。算法首先赋给每个网络连接权一个随机实数，然后不断对网络连接权值进行修正，直到对所有样本模式输入，网络都给出正确输出为止。虽然从多层感知器可以获得任意的非线性映射，但该算法却无法推广到多层感知器的学习。由于单层感知器只能实现线性可分的分类问题，因此只有当参与网络学习的样本模式线性可分时，该算法才会学习成功。

若多层前馈网络的神经元均采用 Sigmoid 作用函数，则相应的神经网络称为 BP 网络。BP 网络因 1986 年 Rumelhart 等提出的 BP 算法而得名。BP 算法采用了与 Widrow-Hoff 算法相同的计算思想，因此 BP 算法所采用的方法又称广义δ规则。该算法计算网络的连接权值，需构造一个误差函数 $E(\boldsymbol{W}, U)$，$\boldsymbol{W}$ 表示所有连接权值变量形成的向量，U 表示所有样本模式组成的集合。首先赋予每个连接权值一个随机初始量，然后利用梯度下降法计算连接权值的修正量：

$$\Delta \boldsymbol{W} = -\eta \frac{\partial E}{\partial \boldsymbol{W}} \tag{6.9}$$

算法不断根据式(6.9)修正网络连接权值，直到误差函数取值达到所要求的范围。BP 算法首次提供了多层前馈神经网络的学习方法，其巧妙地通过误差反传获得了梯度 $\partial E / \partial \boldsymbol{W}$ 的计算方法。近年来许多学者对 BP 算法做了广泛深入的研究，从各个方面探索了该算法的性质[27-29]。根据 BP 算法的计算思想，可得到多种形式的误差反传学习算法，如下所示。

(1)前馈神经网络的神经元采用其他类型的作用函数，若该作用函数是连续可导的，则几乎用同样的方法就可得到 $\partial E / \partial \boldsymbol{W}$ 的计算方法[27]。

(2)BP 算法的误差函数 E 采用平方误差函数，若将该函数换为其他类型的误差函数，如概率误差函数，同样可利用误差反传方法计算 $\partial E / \partial \boldsymbol{W}$ 。因而有概率误差反传算法[27, 30, 31]。

6.2.1　前馈网络模型

没有反馈连接的神经网络就是前馈神经网络。通常前馈网络分为输入层、中间层和输出层。因输入层不需要任何非线性变换，通常该层并不算前馈网络的一层。多层感知器、径向

基函数网络、BP 网络等都属于前馈网络，且都采用广义 δ 规则设计其学习算法。前馈网络的输入输出关系是一种映射关系，从系统观点看，这种映射是高度非线性映射，其信息处理能力也来自简单非线性函数的多次复合。关于前馈神经网络所能实现的映射能力，不加证明给出下述结论。

【定理 6.1】 m 维单位立方体$E^m=[0,1]^m$中的任意一个连续函数$\boldsymbol{\Phi}$, $E^m \to R^n, y=\boldsymbol{\Phi}(\boldsymbol{x})$都可以用三层神经网络去精确地实现。

【定理 6.2】 设$\boldsymbol{\Phi}$为有界非线性单调递增函数，K 为 R^n 的紧致子集(有界闭子集)，$f(\boldsymbol{x})=f(x_1,x_2,\mathrm{L}\ ,x_n)$ 为 K 上的实值连续函数，则对任意 $\varepsilon>0$，存在整数 N 和实常数 C_i，$\theta_i\left(i=1,2,\mathrm{L}\ ,N\right)$和 $w_{ij}\left\{1\leq i,j\leq N\right\}$，使

$$\hat{f}(x_1,\mathrm{L}\ ,x_n)=\sum_{i=1}^{N}C_i\boldsymbol{\Phi}\left(\sum_{j=1}^{n}w_{ij}x_j-\theta_i\right) \tag{6.10}$$

满足$\max\limits_{\boldsymbol{x}\in K}\left|\hat{f}(x_1,\mathrm{L}\ ,x_n)-f(x_1,\mathrm{L}\ ,x_n)\right|<\varepsilon$。也就是说，对任意给定的$\varepsilon>0$，存在一个 3 层网络，其隐单元输出函数为$\boldsymbol{\Phi}(\boldsymbol{x})$，输入和输出单元的输出函数是线性的，使网络输出与函数 $f(\cdot)$ 的实际值的误差不超过ε。

上述结论保证了前馈神经网络逼近任意映射的能力。

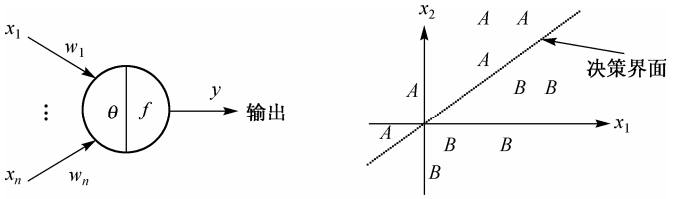
1．感知器网络

Rosenblatt 于 1959 年提出了感知器的概念[23, 24, 32]。利用 δ 规则，容易给出单层感知器的学习算法。单层感知器的映射能力有限，多层感知器则可以实现任意映射 $F:R\to\{0,1\}^m$。但多层感知器的映射能力并不能直接套用定理 6.1、定理 6.2 的结论，因为感知器神经元的作用函数均采用硬限函数，并不连续可导。

(1) 单层感知器

单层感知器既可用于连续值输入，也可用于二进制值输入。图6.6给出一个判断输入到底属于 A, B 两类中哪一类的单层感知器模型。在图6.6中，

$$y=f\left(\sum_{i=1}^{n}w_ix_i-\theta\right)=f(u)=\begin{cases}0, & u<0\\ 1, & u\geq 0\end{cases} \tag{6.11}$$



(a) 神经元　　(b) 分类域

图 6.6　硬限函数神经元及分类域

该神经元的作用相当于一个 n–1 维超平面对 n 维空间的分割。若输入 $\boldsymbol{x}$ 落在超平面上方，则 $y=1$，表示输入属于 A 类；若输入 $\boldsymbol{x}$ 落在超平面下方，则 $y=0$，表示输入属于 B 类。超平面即图中的决策界面，该决策界面是由突触权值 $\boldsymbol{w}$ 和阈值 θ 决定的，显然，若 A 类区域和

B 类区域可以被一个超平面分开，则能够用单层感知器实现分类。利用单层感知器容易实现逻辑与、或运算，与、或函数都是线性可分的函数映射。但对非线性可分函数，单层感知器无能为力。

【定理 6.3】 单层感知器不能实现异或(XOR)函数。

证明： 若有单层感知器可以实现布尔变量 x_1 与 x_2 的异或，即存在 w_1, w_2 和 θ 使 $f(w_1x_1 + w_2x_2 - \theta) = x_1 \oplus x_2$，则有

$$\begin{cases} w_1 - \theta \geq 0 \\ w_2 - \theta \geq 0 \\ w_1 + w_2 - \theta < 0 \\ -w_1 - w_2 - \theta < 0 \end{cases} \tag{6.12}$$

由式(6.12)的第3式和第4式得 $\theta > 0$，由式(6.12)的第1式、第2式、第3式得 $2\theta \leq w_1 + w_2 < 0$，矛盾。

单层感知器不能实现异或运算的几何解释如图6.7所示。显然找不到一条直线将点集 $\{(0,1),(1,0)\}$ 分在直线一边，而将点集 $\{(0,0),(1,1)\}$ 分在另一边。

(2) 多层感知器

多层感知器是在输入和输出节点之间含有一层或多层隐含节点的前馈网络。可以证明，只要隐层神经元个数足够多，多层感知器可以实现任意的函数映射 $F: R^n \to \{0,1\}^m$。因此多层感知器可以实现异或运算。

只要在输入与输出之间加一个隐层，形成一个两层的感知器就可完成异或逻辑门。神经网络结构与决策域如图6.8所示。

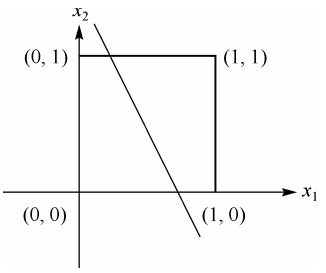
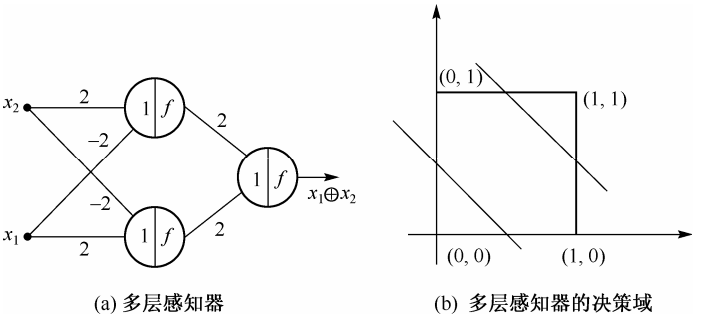


图 6.7　几何解释



(a) 多层感知器　　(b) 多层感知器的决策域

图 6.8　异或运算的多层感知器实现及其决策域

多层感知器的映射能力来自各节点中的非线性函数 $f(\cdot)$。由图 6.8(b)可知，异或函数实现的决策域是一个凸区域。图6.9给出了一、二、三层感知器的决策域形状。分析表明，单层感知器形成一个半平面决策域；二层感知器能形成任何可能的无界的由输入模式张成的凸区域；三层感知器可形成任意复杂的决策域。

凸域指该区域中任意两点连线中的所有点仍在该区域中。凸域是由多层感知器第一层的各个节点所形成的半平面相互重叠而成的。第一层中的各个节点如同单层感知器一样，当输入点落在由权值和阈值形成的超平面之上时,节点输出为 1，否则为 0。若第一层 N_1 个节点到输出节点的所有权值均为 1，且输出节点的阈值为 $N_1 - \varepsilon$ $(0 < \varepsilon < 1)$，则只有当第一层节点均

输出 1 时，输出节点才输出 1，相当于逻辑与运算。最终的决策域是由第一层所形成的所有半平面重叠而得出的。此凸区域的边界数不会超过第一层中的节点个数。

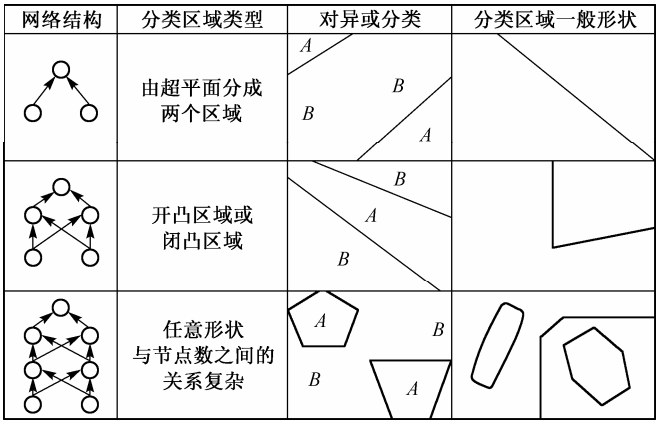


图 6.9 感知器网络结构与分类区域的关系

一个三层感知器可以形成任意复杂的决策区域。下面用构造法简单予以说明。将所希望的决策域分成若干小的凸区域，每个凸区域可由两层感知器决策，将所有凸区域的决策结果进行逻辑或操作，即为所希望决策域的输出结果。逻辑或只需要一个神经元实现。

三层感知器可以实现任意复杂的决策域，因此不必采用三层以上的感知器就可以达到所需目的。从上面的分析可以了解利用三层感知器解决具体问题时应该怎样选择节点个数。如果决策域是互不相连的或是网状的，并且不能由一个凸面形成，则第二层中的节点个数一定要大于 1。最坏情况下，第二层中的节点个数等于输入分布中互不相连的区域个数。第一层中的节点个数必须充分多，使得能为每个第二层节点所产生的凸区域提供多条边。

【例 6.1】 用多层感知器网络实现回归问题，数据是从噪声正弦函数产生的，网络有 3 个隐层单元，权值衰减系数为 0.01。初始化网络之后，使用标准共轭梯度算法进行 100 次训练。数据、函数和网络输出结果如图6.10所示。

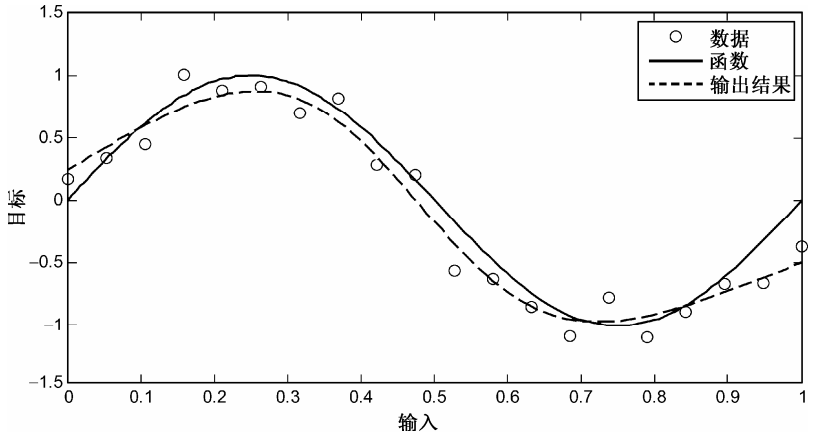


图 6.10 多层感知器实现回归问题的数据、函数和输出结果

2．BP 神经网络

BP 网络是一种多层前馈神经网络，因使用误差反向传播算法即 BP 算法进行学习而得名。该网络神经元的作用函数采用 Sigmoid 函数。该网络的学习算法即为反向传播算法，简称 BP 算法。基本思想是根据样本数据构造一个误差函数，再利用梯度下降法求该函数的最小值，由此得到神经网络的连接权值。

由定理 6.1、定理 6.2 可知，多层的 BP 网络可以实现任意的非线性映射。图6.11 给出了 BP 网络的一般拓扑结构。

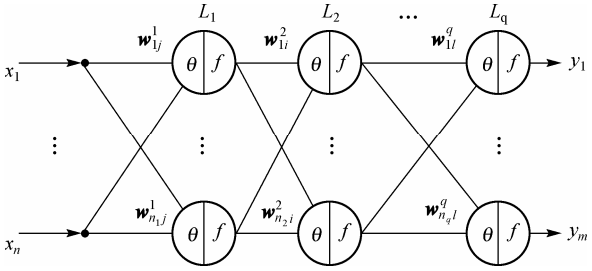


图 6.11　BP 神经网络模型

在图中，设网络输入向量为 $(x_1, x_2, \mathrm{L}\ , x_n)^{\mathrm{T}}$，输出向量为 $(y_1, y_2, \mathrm{L}\ , y_m)^{\mathrm{T}}$，网络的 q 层神经元分别有 $n_1, n_2, \mathrm{L}\ , n_q$ 个节点，第 i 层网络的神经元输出分别为 $x_i^i, \mathrm{L}\ , x_n^i$，则有

$$\sigma_i^1 = \sum_{j=1} w_{ij}^1 x_j - \theta_i^1, \qquad i = 1, 2, \mathrm{L}\ , n_1 \tag{6.13}$$

$$x_i^1 = f(\sigma_i^1), \qquad i = 1, 2, \mathrm{L}\ , n_1 \tag{6.14}$$

$$\sigma_i^s = \sum_{j=1}^{n_s - 1} w_{ij}^s x_j^{s-1} - \theta_i^s, \qquad i = 1, 2, \mathrm{L}\ , n_s,\ \ s = 2, \mathrm{L}\ , q \tag{6.15}$$

$$x_i^s = f(\sigma_i^s), \quad i = 1, 2, \mathrm{L}\ , n_s,\ s = 2, \mathrm{L}\ , q \tag{6.16}$$

$$y_i = w_i^q, \quad i = 1, 2, \mathrm{L}\ , m = n_q \tag{6.17}$$

其中神经元作用函数 $f(\cdot)$ 取 Sigmoid 函数式(6.6)。Rumelhart 等提出的 BP 网络学习算法，使 BP 网络乃至一般多层前馈网络的实际应用变得不再困难。

6.2.2　感知器分类学习算法

学习问题是由训练样本集合描述的。设感知器的模式分类问题由 N 个样本组成的训练样本集合描述：$U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \mathrm{L}\ , \boldsymbol{u}_N\}$，其中 $\boldsymbol{u}_t = (\boldsymbol{x}_t, \boldsymbol{d}_t)$，$\boldsymbol{x}_t = (x_{t1}, x_{t2}, \mathrm{L}\ , x_{tm})^{\mathrm{T}} \in R^n$，$\boldsymbol{d}_t = (d_{t1}, d_{t2}, \mathrm{L}\ , d_{tm})^{\mathrm{T}} \in R^n$，$t = 1, 2, \mathrm{L}\ , N$。学习的目的是建立感知器网络使当以 $\boldsymbol{x}_t$ 输入该网络时，网络的输出为 $\boldsymbol{d}_t$，$\boldsymbol{d}_t$ 代表了样本 $\boldsymbol{u}_t$ 或输入 $\boldsymbol{x}_t$ 的类别。显然，实现 U 所描述的分类问题的神经网络学习算法是有监督的学习算法。

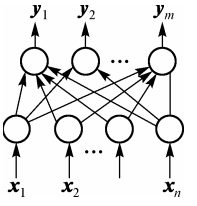


图 6.12　单层感知器

对于单层感知器的学习，训练样本给定后，感知器的拓扑结构也就确定了。只需由 m 个 n 维输入的神经元组成单层感知器，结构如图6.12 所示。

关键问题是如何确定网络的连接权值。设第 i 个输出神经元节点连接权值向量为 $\boldsymbol{w}_i=(w_{i1},w_{i2},\mathrm{L}\ ,w_{in})^{\mathrm{T}}$，阈值为 θ_i，则第 i 个神经元对于模式输入 $\boldsymbol{x}_t$ 的实际输出为

$$y_{ti}=f(\sigma_{ti})=f\left(\sum_{j=1}^{n}w_{ij}x_{tj}-\theta_i\right) \tag{6.18}$$

定义平方误差代价函数如下：

$$E=\frac{1}{2N}\sum_{t=1}^{N}\sum_{i=1}^{m}(d_{ti}-y_{ti})^2 \tag{6.19}$$

先将 $f(\cdot)$ 当做单调增的连续可微函数，得到

$$\frac{\mathrm{d}E}{\mathrm{d}w_{ij}}=-\frac{1}{N}\sum_{t=1}^{N}(d_{ti}-y_{ti})f'(\sigma_{ti})x_{tj} \tag{6.20}$$

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_i}=\frac{1}{N}\sum_{t=1}^{N}(d_{ti}-y_{ti})f'(\sigma_{ti}) \tag{6.21}$$

因此，确定网络连接权值和阈值的基本思想是，先任意给出其初始值，然后根据式(6.20)和式(6.21)不断修正 w_{ij} 和 θ_i，直到误差函数 E 的取值小到一定范围为止。在式(6.20)和式(6.21)中，因 $f(\cdot)$ 是单调增函数，故 $f'(\sigma_{ii})$ 按照梯度下降法将 w_{ij} 和 θ_i 的修正设计为

$$\Delta w_{ij}=-\eta\frac{\mathrm{d}E}{\mathrm{d}w_{ij}}=\frac{\eta}{N}\sum_{t=1}^{N}(d_{ti}-y_{ti})x_{tj} \tag{6.22}$$

$$\Delta\theta_i=-\eta\frac{\mathrm{d}E}{\mathrm{d}\theta_i}=-\frac{\eta}{N}\sum_{t=1}^{N}(d_{ti}-y_{ti}) \tag{6.23}$$

式(6.22)和式(6.23)是由连续可微的神经元作用函数得到的，但其中并不包括求导计算，因此适用于单层感知器的学习计算，η 表示连接权值和阈值每次修正量的步长控制量。在感知器中 $f(\cdot)$ 采用硬限函数，其学习算法的连接权值修正量计算仍由式(6.22)和式(6.23)给出，单层感知器的学习算法形式描述如下：

【算法 6.1】 单层感知器学习算法

(1)给定 w_{ij} 和 θ_i 的初始值，选定 E 的终止值 ε、步长控制量 η。

(2)根据式(6.19)计算误差函数值 E。

(3)若 $E<\varepsilon$，则算法结束。否则，

(4)按式(6.22)和式(6.23)计算 Δw_{ij} 和 $\Delta\theta_i$，并做权值修正：$w_{ij}=w_{ij}+\Delta w_{ij}$ 和 $\theta_i=\theta_i+\Delta\theta_i$。

(5)转到(2)。

上述算法在计算权值修正量时，每次都要用到所有训练样本。在实际算法编程时，可按照一定顺序每次迭代只使用一个训练样本，若所有样本模式都使用完一次后，误差函数值仍不满足要求，则重复使用，直到结束。但该算法的计算是否结束与参与训练的样本模式有关。算法 6.1 对连接权值和阈值的计算是一个误差函数 E 不断减小的过程，当每个样本模式的输出向量与对应网络实际输出向量相等时，$E=0$，算法结束，称为收敛。算法收敛意味着学习

成功而结束，后面的 BP 算法同样存在收敛问题。对于不依赖于误差修正技术的算法，收敛性仅仅表示学习成功而结束。

【定理 6.4】 若待分类的样本模式在空间内是线性可分的，则算法 6.1 可在有限次迭代后收敛。

定理 6.4 的证明很容易，在此略去。

单层感知器只能进行线性可分的函数计算。因此，只有训练样本满足线性可分条件时，上述算法才会学习成功。否则，算法不结束，在实际实现时可通过控制算法迭代次数强制算法结束。

因每次都要根据样本输出和网络实际输出的差值δ来计算权值修正量，因此算法6.1 又称为δ规则学习，这种δ规则学习不能推广到多层感知器的学习问题，原因在于感知器的神经元采用的硬限作用函数并不连续可导。

6.2.3　BP网络分类学习算法

BP 网络的拓扑结构与多层感知器相同。只是 BP 网络模型的神经元作用函数均采用 Sigmoid 函数。BP 网络可以实现一个映射 $F: R^n \to R^m$，因此描述 BP 网络分类问题的训练样本集合可给出为 $U=\{\boldsymbol{u}_1, \boldsymbol{u}_2, \mathrm{L}\ , \boldsymbol{u}_N\}$，其中 $\boldsymbol{u}_p=(\boldsymbol{x}_p, \boldsymbol{d}_p)$，$\boldsymbol{x}_p=(x_{p1}, x_{p2}, \mathrm{L}\ , x_{pm})^{\mathrm{T}} \in R^n$，$\boldsymbol{d}_p=(d_{p1}, d_{p2}, \mathrm{L}\ , d_{pm})^{\mathrm{T}} \in R^n$，$p=1, 2, \mathrm{L}\ , N$。

单层感知器的学习方法不能推广到多层感知器，但能推广到多层 BP 网络的学习，关键是如何计算网络连接权值和神经元阈值的修正量。首先要求网络结构是已知的，即网络共有几层，每层有多少个节点，都是事先确定好的。算法利用误差函数 E 衡量学习是否成功，通过迭代不断修正网络的连接权值，以寻找误差函数 E 的最小值。网络连接权值的修正量利用梯度下降法得到。

1．标准 BP 算法

根据 BP 网络结构，如图 6.11 所示，假设网络共有 q 层。因此定义误差函数

$$E=\frac{1}{2}\sum_{p=1}^{N}\sum_{i=1}^{m}(d_{pi}-y_{pi})^2=\sum_{p=1}^{N}E_p \tag{6.24}$$

其中，

$$E_p=\frac{1}{2}\sum_{i=1}^{m}(d_{pi}-y_{pi})^2 \tag{6.25}$$

$$y_{pi}=x_{pi}^{q} \tag{6.26}$$

利用梯度下降法寻找 E 的局部最小值，每个连接权值均需沿着 E 对连接权值导数的反方向修正。因此必须首先计算 $\partial E/\partial w_{ij}^t$ 和 $\partial E/\partial \theta_i^t$。假设网络输入样本向量 $\boldsymbol{x}_p$ 时，网络 t 层的第 i 个神经元的激励总和为 $\sigma_{pi}^t=\sum_{j=1}^{n_{t-1}} w_{ij}^t x_j^{t-1}-\theta_i^t$，令

$$\delta_{pi}^t=-\frac{\partial E_p}{\partial \sigma_{pi}^t} \tag{6.27}$$

下面给出误差函数对连接权值和阈值的计算公式：

$$\delta_{pi}^{q}=(d_{pi}-x_{pi}^{q})f'(\sigma_{pi}^{q}) \tag{6.28}$$

$$\frac{\partial E_p}{\partial w_{ij}^{q}}=-\delta_{pi}^{q}x_{pj}^{q-1} \tag{6.29}$$

$$\frac{\partial E_p}{\partial \theta_{i}^{q}}=\delta_{pi}^{q} \tag{6.30}$$

$$\delta_{pi}^{t}=\left(\sum_{k=1}^{n_{t+1}}\delta_{pk}^{t+1}w_{ki}^{t+1}\right)f'(\sigma_{pi}^{t+1}),\qquad t=1,2,\mathrm{L}\ ,q-1 \tag{6.31}$$

$$\frac{\partial E_p}{\partial w_{ij}^{t}}=-\delta_{pi}^{t}x_{pj}^{t-1} \tag{6.32}$$

$$\frac{\partial E_p}{\partial \theta_{i}^{t}}=\delta_{pi}^{t} \tag{6.33}$$

由于假定 $f(\cdot)$ 是 Sigmoid 函数，所以其导数可以按如下公式计算：

$$x_{pi}^{t}=f(\sigma_{pi}^{t})=\frac{1}{1+\mathrm{e}^{-\mu\sigma_{pi}^{t}}} \tag{6.34}$$

$$f'(\sigma_{pi}^{t})=\mu f(\sigma_{pi}^{t})(1-f(\sigma_{pi}^{t}))=\mu x_{pi}^{t}(1-x_{pi}^{t}) \tag{6.35}$$

于是得到网络连接权值和神经元阈值的修正量计算公式为

$$\Delta w_{ij}^{t}=\sum_{p=1}^{N}\delta_{pi}^{t}x_{pj}^{t-1},\qquad t=1,2,\mathrm{L}\ ,q,\quad i=1,2,\mathrm{L}\ ,n_t \tag{6.36}$$

$$\Delta \theta_{i}^{t}=-\sum_{p=1}^{N}\delta_{pi}^{t},\qquad t=1,2,\mathrm{L}\ ,q,\quad i=1,2,\mathrm{L}\ ,n_t \tag{6.37}$$

$$\delta_{pi}^{q}=(d_{pi}-x_{pi}^{q})\mu x_{pi}^{q}(1-x_{pi}^{q}),\qquad i=1,2,\mathrm{L}\ ,n_q=m \tag{6.38}$$

$$\delta_{pi}^{t}=\left(\sum_{k=1}^{n_{t+1}}\delta_{pk}^{t+1}w_{ki}^{t+1}\right)\mu x_{pi}^{t}(1-x_{pi}^{t}),\qquad t=1,2,\mathrm{L}\ ,q-1,\quad i=1,2,\mathrm{L}\ ,n \tag{6.39}$$

要计算网络连接权值的修正量，需首先正向计算网络每一层的节点输出 x_{pi}^{t}，然后反向计算 δ_{pi}^{t}，这就是反向传播计算思想。反向传播或 BP 算法形式化给出如下：

【算法 6.2】 反向传播(Back Propagation)学习算法

(1)给定 w_{ij} 和 θ_i 的初始值，选定 E 的终止值 ε，步长控制量 η。

(2)对每个训练样本 $\boldsymbol{u}_p$，正向计算 x_{pi}^{t}，并根据式(6.24)、式(6.25)计算误差函数值 E。

(3)若 $E<\varepsilon$，则算法结束。否则，

(4)对每个训练样本，反向计算 δ_{pi}^{t}，根据式(6.32)和式(6.33)计算 Δw_{ij}^{t} 和 $\Delta\theta_{i}^{t}$。

(5)权值修正：$w_{ij}^t = w_{ij}^t + \Delta w_{ij}^t$，$\theta_i^t = \theta_i^t + \Delta\theta_i^t$。

(6)转到(2)。

由以上讨论看出，对于给定的样本集合，目标误差函数 E 是全体连接权值和全体神经元阈值的函数，BP 算法实际是对误差函数 E 的寻优计算。E 函数又是关于连接权值和阈值的一个非常复杂的超曲面，由于寻优参数太多，BP 算法的一个最大问题就是收敛速度慢。BP 算法的另一个缺陷是局部极值问题，E 的超曲面可能存在多个极值点，但上述算法一般收敛到极值附近的局部最小点，这可以通过重新选择初始值予以解决。只要有足够多的隐层和隐层节点，BP 网络就可以逼近任意映射。BP 算法提供了一种确定网络连接权值的全局逼近方法，因而 BP 网络和 BP 算法在实践中得到了广泛的应用。在应用中，如何根据特定问题确定网络结构是十分重要的问题，若采用 BP 算法学习，只能凭经验和试验的方法确定网络结构。

BP 网络能够实现输入、输出之间的非线性映射，但它并不依赖于由输入和输出向量描述的具体模型。其输入与输出之间的关系分布存储在连接权值中。由于连接权的个数很多，个别神经元的损坏只对输入输出关系有较小的影响，因而 BP 网络显示了很好的容错性能。

2．BP 算法的改进

人们对 BP 算法进行了广泛研究，提出了许多改进的 BP 算法。下面介绍典型的几种。

(1)引入动量项

标准 BP 算法实质上是一种简单的最速下降静态寻优算法，在修正网络连接权值 $\boldsymbol{w}$ 时，只是按照当时的梯度反方向进行修正，而没有考虑以前积累的经验，即以前时刻的梯度方向，从而常使学习过程发生振荡，收敛缓慢。在网络权值修正量中考虑当前时刻和前一时刻的梯度，即

$$\Delta \boldsymbol{w}(t) = \alpha\left[(1-\eta)\left(-\frac{\partial E}{\partial \boldsymbol{w}(t)}\right) + \eta\left(\frac{\partial E}{\partial \boldsymbol{w}(t-1)}\right)\right] \tag{6.40}$$

其中，$\boldsymbol{w}(t)$ 表示第 t 次权值修正时网络的某个连接权值或神经元阈值。α 为学习率，n 为动量项因子，$\alpha > 0$，$0 \leq \eta < 1$。该方法加入的动量项相当于阻尼项，它减小了学习过程的振荡趋势，改善了收敛性。

(2)变尺度法

标准 BP 算法采用一阶梯度法是导致收敛速度慢的另一个原因。若采用二阶梯度法，收敛性可以得到大大改善。二阶梯度法的权值修正量计算公式为

$$\Delta \boldsymbol{w}(t) = -\alpha\left(\frac{\partial^2 E}{\partial \boldsymbol{w}^2}\right)^{-1}\frac{\partial E}{\partial \boldsymbol{w}(t)} \tag{6.41}$$

虽然二阶梯度法有较好的收敛性，但是需要计算 E 对 $\boldsymbol{w}$ 的二阶导数，二阶导数的计算量很大，所以一般不直接采用二阶梯度法，而常常采用变尺度法或共轭梯度法。这样既能加快算法的收敛速度，又无须直接计算二阶梯度。

(3)变步长法

一阶梯度法寻优收敛速度慢的一个原因是学习率 α 选择得不恰当。α 选得太小，则收敛速度慢，α 选得太大，则有可能修正过头，导致振荡甚至发散。变步长法即是针对这一问题

提出的。变步长法的权值修正量计算如下：

$$\Delta \boldsymbol{w}(t) = -\alpha(t)\frac{\partial E}{\partial \boldsymbol{w}(t)} \tag{6.42}$$

$$\alpha(t) = 2^{\lambda}\alpha(t-1) \tag{6.43}$$

$$\lambda = \operatorname{sgn}\left[\frac{\partial E}{\partial \boldsymbol{w}(t)}\frac{\partial E}{\partial(\boldsymbol{w}(t-1)}\right] \tag{6.44}$$

上述算法说明，当连续两次迭代梯度方向相同时，表明下降太慢，因而步长加倍；当连续两次迭代梯度方向相反时，表明下降过头，因而步长减半。当需要引入动量项时，式(6.42)可修改为

$$\Delta \boldsymbol{w}(t) = \alpha(t)\left[(1-\eta)\left(-\frac{\partial E}{\partial \boldsymbol{w}(t)}\right)+\eta\left(-\frac{\partial E}{\partial \boldsymbol{w}(t-1)}\right)\right] \tag{6.45}$$

在使用该算法时，由于步长在迭代过程中自适应进行调整，因此对于不同的连接权值实际上采用了不同的学习率。也就是说，误差代价函数 E 在超曲面上于不同方向按照各自比较合理的步长向极小点逼近。

6.3 联想记忆网络学习算法

联想记忆神经网络又称联想存储器，前馈网络和反馈网络都可以用做联想记忆网络。Hopfield 联想存储器是根据 Hebb 规则学习得到的单层反馈神经网络，这是目前最有影响力的联想记忆网络。Hopfield 联想记忆网络的神经元个数即样本模式维数，Hebb 学习方法能简单而快速地计算网络的连接权值，因此基于 Hebb 规则的联想记忆网络学习算法是一种结构学习算法。

容错能力和存储容量是衡量联想记忆网络好坏的两个重要指标。Hopfield 联想记忆网络由上述指标衡量时，并不十分理想。分析表明，若所有样本模式均是两两正交的，则 n 个神经元的 Hopfield 网络可以存储 n 个模式，一般情况下，n 个神经元的 Hopfield 网络可存储 $O(n/\log\ n)$ 个模式[33, 34]。当样本模式之间的海明距离较小时，会造成存储样本吸引域太小而出现联想错误，又由于有与存储样本模式数目相当的假吸引中心存在，使得该网络的稳定性和容错能力均不理想。近年来人们提出了多种新的神经网络联想记忆模型[34-36]及相应学习算法，在网络存储容量和稳定性方面有所改进，但均未能消除假吸引中心的存在问题。

利用 BP 网络可以将前馈网络用做自联想的联想存储器，只需将样本模式相应变为输入、输出相同的样本训练 BP 网络即可。但 BP 网络用做联想存储器，非样本模式的稳定输出均为假吸引中心，一般认为这样造成的假吸引中心更多。BP 算法学习时间复杂度太高是其另一缺陷。分析表明，对于网络连接权值不随网络收敛而变化的反馈网络，其最大存储容量可表示为

$$\text{存储容量} = O(c\times\text{网络规模}) \tag{6.46}$$

其中网络规模即网络的神经元个数，c 为不大于 1 的正常数。Hopfield 联想记忆网络并未达到最好的存储容量，其他的改进模型也都未能消除假吸引中心的存在性，容错能力亦受到限制。

因此研究容错能力和存储容量俱佳的联想记忆网络学习算法仍具有十分重要的意义。

6.3.1　反馈网络模型

反馈网络是动态神经网络，一般需要工作一段时间才会达到稳定。最典型的反馈神经网络是 Hopfield 网络，Hopfield 本人首先将该模型应用于联想记忆和优化计算。

根据网络输出是离散量还是连续量，或根据网络所采用神经元的不同，Hopfield 网络可分为离散和连续两种网络模型。

1．离散型 Hopfield 网络

(1) 网络结构和工作方式

离散型 Hopfield 网络是单层反馈神经网络，单个神经元采用硬限作用函数。网络结构如图6.13所示。每个神经元都有对其他神经元的反馈连接，但没有自反馈连接。每个神经元的阈值以外部输入形式出现，用来控制神经网络的初始状态。对于每个神经元，其工作方式与硬限函数神经元完全相同，其中作用函数通常取{0, 1}或{–1, 1}值。

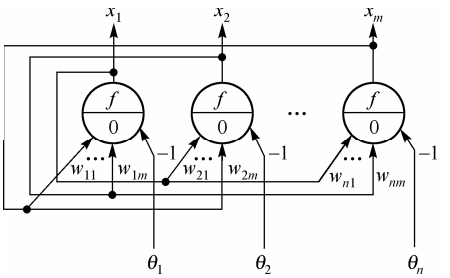


图 6.13　离散 Hopfield 网络

Hopfield 网络有如下两种工作方式。

① 异步方式。每次只有一个节点进行状态调整计算，其他节点的状态保持不变。网络的动力学描述为

$$\begin{cases} x_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij}x_j(t) - \theta_i\right) \\ x_j(t+1) = x_j(t), \qquad j \neq i \end{cases} \tag{6.47}$$

每次由哪个神经元调整状态可以随机选定，也可以按照规定的次序进行。

② 同步方式。所有节点同时调整状态。其动力学描述为

$$x_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij}x_j(t) - \theta_i\right), \qquad i = 1, 2, \text{L}\ , n \tag{6.48}$$

上述计算也可以写成矩阵形式

$$\boldsymbol{x}(t+1) = f(\boldsymbol{W}\boldsymbol{x}(t) - \theta) \tag{6.49}$$

其中，$\boldsymbol{x} = (x_1, x_2, \text{L}\ , x_n)^{\text{T}}$ 是状态向量，$\boldsymbol{\theta} = (\theta_1, \theta_2, \text{L}\ , \theta_n)^{\text{T}}$ 是阈值向量，$\boldsymbol{W}$ 是由 w_{ij} 所组成的 $n \times n$ 矩阵，$f(\cdot)$ 是向量函数，表示 $\boldsymbol{f}(\sigma) = (f(\sigma_1), f(\sigma_2), \text{L}\ , f(\sigma_n))^{\text{T}}$。该网络是动态的反馈网络，初始状态为 $\boldsymbol{x}(0) = (x_1(0), x_2(0), \text{L}\ , x_n(0))^{\text{T}}$。网络的输入即为其初始状态，输出为网络的稳定状态：$\lim\limits_{t\to\infty} \boldsymbol{x}(t)$。网络是否有稳定的输出状态是该网络必须考虑的问题。

(2) 网络的稳定性

离散 Hopfield 网络是离散的非线性动力学系统。如果系统是稳定的，则可以从任意一个初态收敛到一个稳定状态；若系统是不稳定的，由于网络输出点只有两种状态，因而系统不可能出现无限发散，只可能出现限幅的自持振荡或极限环。若将稳态视为一个记忆样本，则

初态到稳态的收敛过程就是寻找记忆样本的过程。初态可认为是给定样本的部分信息，网络改变的过程是从部分信息到全部信息的联想过程。因而 Hopfield 神经网络的重要应用是联想存储器。若将稳态与某种优化计算的目标函数相对应，并作为目标函数的极小点，那么初态到稳态的收敛过程是优化计算的过程。该优化计算是在网络演变过程中自动完成的。

【定义 6.1】 若网络的状态 $\boldsymbol{x}$ 满足 $\boldsymbol{x}=f(\boldsymbol{W}\boldsymbol{x}-\theta)$，则称 $\boldsymbol{x}$ 为网络的稳定点或吸引子。

【定理 6.5】 对于离散 Hopfield 神经网络，若按异步工作方式调整状态，且网络的连接矩阵 $\boldsymbol{W}$ 对称，则对于任意初态，网络都最终收敛到一个吸引子。

证明： 定义网络的能量函数为

$$E(t)=-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\boldsymbol{x}(t)+\boldsymbol{x}^{\mathrm{T}}(t)\theta \tag{6.50}$$

由于神经元的状态只取两种状态 1 和–1（1 和 0），因此上述定义的能量函数 $E(t)$ 是有界的。令 $\Delta E(t)=E(t+1)-E(t)$，$\Delta\boldsymbol{x}(t)=\boldsymbol{x}(t+1)-\boldsymbol{x}(t)$，则

$$\begin{aligned}\Delta E(t)&=E(t+1)-E(t)\\&=-\frac{1}{2}[\boldsymbol{x}(t)+\Delta\boldsymbol{x}(t)]^{\mathrm{T}}\boldsymbol{W}[\boldsymbol{x}(t)+\Delta\boldsymbol{x}(t)]+[\boldsymbol{x}(t)+\Delta\boldsymbol{x}(t)]^{\mathrm{T}}\theta-\left[-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\boldsymbol{x}(t)+\boldsymbol{x}^{\mathrm{T}}(t)\theta\right]\\&=\Delta\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\boldsymbol{x}(t)-\frac{1}{2}\Delta\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\Delta\boldsymbol{x}(t)+\Delta\boldsymbol{x}^{\mathrm{T}}(t)\theta\\&=-\Delta\boldsymbol{x}^{\mathrm{T}}(t)[\boldsymbol{W}\boldsymbol{x}(t)-\theta)-\frac{1}{2}\Delta\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\Delta\boldsymbol{x}(t)\end{aligned} \tag{6.51}$$

由于假定为异步工作方式，可假设在 t 时刻只有第 i 个神经元调整状态，即 $\Delta\boldsymbol{x}(t)=[0,0,\mathrm{L}\ ,0,\Delta x_i(t),0,\mathrm{L}\ ,0]$，代入上式得

$$\begin{aligned}\Delta E(t)&=-\Delta x_i(t)\left[\sum_{j=1}^{n}w_{ij}x_j(t)-\theta_i\right]-\frac{1}{2}\Delta x_i^2 w_{ii}\\&=-\Delta x_i(t)\left[\sigma_i(t)+\frac{1}{2}\Delta x_i(t)w_{ii}\right]\\&=-\Delta x_i(t)\sigma_i(t)\end{aligned} \tag{6.52}$$

设神经元节点取 1 和–1 两种状态，考虑 $-\Delta x_i(t)$ 可能出现的各种情况，可以得到 $\Delta x_i(t)\sigma(t)\geqslant 0$，可见在任何情况下均有 $\Delta E(t)\leqslant 0$。由于 $E(t)$ 有下界，所以 $E(t)$ 将收敛到一个常数。另外，当 $\Delta E(t)=0$ 时，有以下两种情况之一：

情况 1：$x_i(t+1)=x_i(t)=1$ 或 $x_i(t+1)=x_i(t)=-1$

情况 2：$x_i(t)=-1$，$x_i(t+1)=1$，$\sigma_i(t)=0$

第一种情况表明网络已经进入稳态。对于第二种情况，若 x_i 由 1 再变回 – 1，则有 $\Delta E<0$，与 $E(t)$ 已经收敛到常数矛盾。所以网络最终将收敛到吸引子。

上述分析假设 $w_{ii}=0$，实际上，当 $w_{ii}>0$ 时结论也成立，而且收敛速度更快。同时假设每个节点取 1 和 – 1 两种状态，不难验证当取 1 和 0 两种状态时结论也成立。

【定理 6.6】 对于离散 Hopfield 网络，若按同步方式调整状态，且连接矩阵 $\boldsymbol{W}$ 为非负

定对称阵，则对于任意初态，网络最终收敛到一个吸引子。

证明：利用定理 6.1 的能量函数，得

$$\begin{aligned}\Delta E(t) &= E(t+1) - E(t) \\ &= -\Delta \boldsymbol{x}^{\mathrm{T}}(t)[\boldsymbol{W}\boldsymbol{x}(t) - \theta] - \frac{1}{2}\Delta \boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\Delta \boldsymbol{x}(t) \\ &= -\sum_{i=1}^{n}\Delta x_i(t)\sigma_i(t) - \frac{1}{2}\Delta \boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\Delta \boldsymbol{x}(t)\end{aligned} \tag{6.53}$$

对 $\forall i$ 有 $-\Delta x_i(t)\sigma_i(t) \leqslant 0$，因为 $\boldsymbol{W}$ 非负定，所以 $\Delta E(t) \leqslant 0$，也即 $E(t)$ 将最终收敛到一个常数值，按照与上面同样的分析可知网络将最终收敛到一个吸引子。

可见对于同步方式，要使网络收敛，则对连接矩阵的要求更高了。若不满足 $\boldsymbol{W}$ 非负定的条件，则网络也可能收敛到一个两个状态的极限环。异步工作方式的稳定性较好，但失去了神经网络并行计算的优点。

(3) 联想记忆

对于给定的一组二进制数据，若能设计一个 Hopfield 神经网络，使给定数据成为该网络的吸引子，则网络成为存储给定数据的联想存储器，给定的数据称为存储样本模式。Hopfield 网络用做联想记忆神经网络，亦称为 Hopfield 联想存储器，简称 HAM。 HAM 采用 Hebb 规则确定神经网络的连接权值，所确定的神经网络具有对称的连接矩阵。

为了实现正确的联想记忆，要求每个存储模式都是网络的吸引子，每个吸引子都有一定的吸引范围，吸引范围称为联想存储器的吸引域。一般认为一个模式向量 $\boldsymbol{x}^{(k)}$ 的吸引域是以该向量为中心的球体，该球体内的向量 $\boldsymbol{x}^{(s)}$ 满足 $d_H(\boldsymbol{x}^{(s)}, \boldsymbol{x}^{(k)}) \leqslant \alpha n (0 \leqslant \alpha \leqslant 0.5)$，称 α 为吸引半径，n 为神经元个数。HAM 的吸引子具有如下性质。

性质 1：若 $\boldsymbol{x}$ 是网络的一个吸引子，且对 $\forall i$，$\theta_i = 0$，$\sum_{j=1}^{n} w_{ij}x_j \neq 0$，则 $-\boldsymbol{x}$ 也是该网络的吸引子。

证明：由于 $\boldsymbol{x}$ 是吸引子，即 $\boldsymbol{x} = f(\boldsymbol{W}\boldsymbol{x})$，从而有 $f(\boldsymbol{W}(-\boldsymbol{x})) = f(-\boldsymbol{W}\boldsymbol{x}) = -f(\boldsymbol{W}\boldsymbol{x}) = -\boldsymbol{x}$，即 $-\boldsymbol{x}$ 也是网络的吸引子。

性质 2：若 $\boldsymbol{x}^{(a)}$ 是网络的吸引子，则 $\boldsymbol{x}^{(a)}$ 与海明距离 $d_H(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)}) = 1$ 的 $\boldsymbol{x}^{(b)}$ 一定不是该网络的吸引子，海明距离定义为两个向量中不相同的元素个数。

证明：不失一般性，设 $x_1^{(a)} \neq x_1^{(b)}$，$x_i^{(a)} = x_i^{(b)}$，$i = 2,\mathrm{L}\ ,n$。因为 $w_{11} = 0$，所以有

$$x_1^{(a)} = f\left[\sum_{j=2}^{n} w_{1j}x_j^{(a)} - \theta_1\right] = f\left[\sum_{j=2}^{n} w_{1j}x_j^{(b)} - \theta_1\right] \neq x_1^{(b)} \tag{6.54}$$

所以 $\boldsymbol{x}^{(\mathrm{b})}$ 一定不是网络的吸引子。

推论：若 $\boldsymbol{x}^{(a)}$ 是网络的一个吸引子，且对 $\forall i, \theta_i = 0$，$\sum_{j=1}^{n} w_{ij}x_j \neq 0$，则 $\boldsymbol{x}^{(a)}$ 与海明距离 $d_H(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)}) = n - 1$ 的 $\boldsymbol{x}^{(b)}$ 一定不是该网络的吸引子。

证明：若 $d_H(\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)}) = n - 1$，则 $d_H(-\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)}) = 1$。根据性质 1，$\boldsymbol{x}^{(a)}$ 是网络的吸引子，$-\boldsymbol{x}^{(a)}$ 也是网络的吸引子，根据性质 2，$\boldsymbol{x}^{(b)}$ 一定不是吸引子。

由上述性质可知，若要使存储样本成为 HAM 的吸引子，则必然产生同样多的假吸引子。

Hopfield 神经网络最多能存储多少个样本是 HAM 的记忆容量或存储容量。记忆容量不仅与网络节点个数有关，还与连接权的设计有关。对于 Hebb 规则设计连接权值的网络，如果记忆样本是正交的，则可以获得最大的记忆容量。实际问题的样本不可能都是正交的，所以在研究记忆容量时通常假设样本向量是随机的。记忆容量还与要求的吸引域大小有关，要求的吸引域越大，则记忆容量越小。对于给定的网络，严格分析并确定其记忆容量并非易事。Hopfield 给出的实验结果为

$$m \leqslant 0.15n \tag{6.55}$$

按照记忆模式为随机分布的假设所做的理论分析表明，当 $n \to \infty$ 时，HAM 的记忆容量为[28,33, 37]

$$m \leq \frac{(1-2\alpha)^2 n}{2\ln n} \tag{6.56}$$

其中 α 为要求的吸引半径。

(4) 双向联想记忆

在实际应用中，许多情况下需要双向联想记忆。双向联想记忆网络的原理几乎与 Hopfield 联想记忆网络相同。

2．连续 Hopfield 神经网络

(1) 网络结构和工作方式

连续 Hopfield 神经网络也是单层的反馈网络，其结构仍如图 6.13 所示。对于每个神经元，工作方式为

$$\begin{cases} \sigma_i = \sum_{j=1}^{n} w_{ij} x_j - \theta_i \\ \dfrac{\mathrm{d}y_i}{\mathrm{d}t} = -\dfrac{1}{\tau_i} y_i + \sigma_i \\ x_i = f(y_i) \end{cases} \tag{6.57}$$

同样假定 $w_{ij} = w_{ji}$。与离散 Hopfield 网络比较，式(6.57)中增加了一个微分方程，相当于一个惯性环节。σ_i 是该环节的输入，y_i 是该环节的输出。对于离散的 Hopfield 网络，中间的式子也可视为 $y_i = \sigma_i$，差别在于神经元作用函数。连续 Hopfield 神经网络的神经元作用函数取 Sigmoid 函数。对应于离散型网络，当 $x_i \in \{-1, 1\}$ 时，此处可取神经元作用函数：

$$f(y_i) = \frac{1 - \mathrm{e}^{-\mu y_i}}{1 + \mathrm{e}^{-\mu y_i}} \tag{6.58}$$

当 $x_i \in \{0, 1\}$ 时，此处取神经元作用函数：

$$f(y_i) = \frac{1}{1 + \mathrm{e}^{-\mu y_i}} \tag{6.59}$$

式(6.58)和式(6.59)都是连续的单调上升函数。连续 Hopfield 网络是一个连续非线性动力学系统，可用一组非线性微分方程来描述。给定初始状态 $x_i(0)$，$i=1,2,\mathrm{L}\ ,n$，通过求解非线性微分方程组可求得网络状态的运动轨迹。若系统是稳定的，则最终将收敛到一个稳定状态。求解对应的微分方程组的过程由硬件实现电路自动完成，速度是非常快的。

考虑连续 Hopfield 网络的稳定性，我们有如下定理。

【定理 6.7】 对于连续 Hopfield 神经网络，若 $w_{ij}=w_{ji}$, $\tau_i>0$，$f(\cdot)$ 单调递增，则网络是稳定的。

证明：定义该网络的能量函数为

$$\begin{aligned}E(t)&=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}x_i(t)x_j(t)+\sum_{i=1}^{n}x_i(t)\theta_i+\sum_{i=1}^{n}\frac{1}{\tau_i}\int_0^{\tau_i}f^{-1}(\eta)\mathrm{d}\eta\\&=-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{W}\boldsymbol{x}(t)+\boldsymbol{x}^{\mathrm{T}}(t)\theta+\sum_{i=1}^{n}\frac{1}{\tau_i}\int_0^{\tau_i}f^{-1}(\eta)\mathrm{d}\eta\end{aligned}\tag{6.60}$$

根据 Lyapunov 关于非线性动力系统稳定性理论，只需证明 $\dfrac{\mathrm{d}e}{\mathrm{d}t}\leqslant 0$ 即可说明网络的稳定性：

$$\begin{aligned}\frac{\mathrm{d}E}{\mathrm{d}t}&=\sum_{i=1}^{n}\frac{\partial E}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}t}=\sum_{i=1}^{n}\left[-\sum_{j=1}^{n}w_{ij}x_j+\theta_i+\frac{1}{\tau_i}f^{-1}(x_i)\right]\frac{\mathrm{d}x_i}{\mathrm{d}t}\\&=\sum_{i=1}^{n}-\frac{\mathrm{d}y_i}{\mathrm{d}t}\frac{\mathrm{d}x_i}{\mathrm{d}t}=-\sum_{i=1}^{n}\frac{\mathrm{d}y_i}{\mathrm{d}x_i}\left(\frac{\mathrm{d}x_i}{\mathrm{d}t}\right)^2\end{aligned}\tag{6.61}$$

因函数 $f(\cdot)$ 单调递增，所以 $y_i=f^{-1}(x_i)$ 单调递增。所以 $(\mathrm{d}y_i/\mathrm{d}x_i)>0$，同时 $(\mathrm{d}x_i/\mathrm{d}t)^2\geqslant 0$，因而有

$$\frac{\mathrm{d}E}{\mathrm{d}t}\leqslant 0\tag{6.62}$$

故网络一定是渐进稳定的。在网络的稳定态，能量函数一定取极小值，因此可以用 Hopfield 网络进行优化计算。

3．Boltzmann机

Boltzmann 机的拓扑结构与 Hopfield 网络相同，是单层反馈的神经网络。但这种网络的工作方式设计为一个随机收敛过程。该网络的随机动力学可描述如下：

$$\begin{cases}\sigma_i(t)=\displaystyle\sum_{j=1}^{n}w_{ij}x_j(t-1)-\theta_i\\[2ex]P(x_i(t)=1)=\dfrac{1}{1+\mathrm{e}^{-\sigma_i(t)/T}}\\[2ex]P(x_i(t)=0)=\dfrac{\mathrm{e}^{-\sigma_i(t)/T}}{1+\mathrm{e}^{-\sigma_i(t)/T}}\end{cases}\tag{6.63}$$

其中，T 称为温度。可以构造该网络的能量函数为

$$E=-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\theta} \tag{6.64}$$

则从概率意义上看，网络的上述能量是随时间而不断下降的，因而网络在概率意义下是稳定的。

Boltzman 机采用模拟退火算法进行学习，可用于模式分类、预测、组合优化等方面的实际问题。

6.3.2 联想记忆分类学习算法

前馈网络的分类记忆是由当前的输入和网络的连接权值决定的，而反馈网络总是将以前的输出返回到输入端，其输出不但取决于当前的输入，而且还取决于过去的输出。联想记忆是对反馈网络计算收敛性的一种描述，实质上是一种分类记忆。联想记忆通常包括自联想和异联想。自联想是指网络将缺损、畸变或受干扰的模式通过回忆联想出来。异联想是将缺损、畸变或受干扰的模式通过存储器中的一对一关系，映射找到对应模式的属性。衡量容错记忆性能的指标主要有容错性和存储容量。容错性是指对失真的样本仍然能给出其原本的属性。存储容量是指网络中存储互不干扰样本的最大数目。神经网络联想记忆的容错性与它的非局域存储方式有关，网络的存储容量则与网络结构、学习方式和网络设计参数有关。

1．Hopfield联想记忆网络学习算法

利用 Hopfield 网络做联想记忆同样需要根据训练样本确定具体的网络连接。其学习过程就是形成网络连接和计算网络连接权值的过程。

联想记忆学习问题由训练样本集合$U=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N\}$给出，其中$\boldsymbol{x}_t=(x_{t1},x_{t2},\mathrm{L}\ ,x_{tm})^{\mathrm{T}}\in\{-1,+1\}^n$。学习的目的是形成离散 Hopfield 神经网络，使每个样本模式都成为网络的吸引子。

既然训练样本都是 n 维的二进制向量，因此网络结构就只需由 n 个硬限作用函数神经元组成。关键问题是如何确定网络的连接权值。Hopfield 根据 Hebb 规则设计采用了如下十分简单的连接权值计算方法：

$$w_{ij}=\begin{cases}\sum_{p=1}^{N}x_{pi}x_{pj}, & i\neq j\\ 0, & i=j\end{cases},\quad 1\leqslant i,j\leqslant n \tag{6.65}$$

通常用连接矩阵描述 Hopfield 网络的所有连接权值。将式(6.65)写成矩阵形式，有

$$\boldsymbol{W}=\sum_{p=1}^{N}\boldsymbol{x}_p\boldsymbol{x}_p^{\mathrm{T}}-I \tag{6.66}$$

Hopfield 网络的连接一旦确定，只要输入某个样本模式，网络就不断演化，直至网络收敛到吸引子，此吸引子是状态空间中的定态，即网络的最终输出态。Hopfield 网络联想记忆分类的学习算法如下。

【算法 6.3】 联想记忆分类学习算法

(1) 根据训练样本集合$(\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N)$设计连接权值矩阵。$k=0$，开始迭代。

(2) 计算网络连接权值$w_{ij}(k+1)=w_{ij}(k)+x_{(k+1),i}x_{(k+1),j}$。

(3) $k=k+1$，若$k>N$，则算法结束，否则转到(2)。

在学习阶段，因连接权值矩阵是死记硬背式学习，其学习是简单的。网络对输入任意给定未知模式，一般需要经过若干次状态演化，网络才会输出稳定的值，达到收敛。如输入的未知样本是原存储模式受到干扰形成的，最后输出的是原存储模式，即为联想记忆。

【定理 6.8】 设 N 个样本模式 $(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathrm{L}\ , \boldsymbol{x}_N)$ 满足

$$\varepsilon n \leq d_H(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq (1-\varepsilon)n, i \neq j, 0 < \varepsilon < \frac{1}{2}$$

若 $N \leq 1 + \dfrac{1}{1-2\varepsilon}$，则当 N 足够大时，$\boldsymbol{x}_1, \boldsymbol{x}_2, \mathrm{L}\ , \boldsymbol{x}_N$ 都是根据算法 6.3 所得网络的吸引子。

在离散联想记忆网络设计中，网络节点的作用函数采用硬限函数。上述讨论均假设网络节点的输出取–1 和+1 两种状态。若网络节点的状态取值为{0, 1}时，按照 Hebb 规则，Hopfield 联想记忆网络的连接权值应按式(6.67)计算：

$$w_{ij} = \begin{cases} \sum_{p=1}^{N} (2x_{pi} - 1)(2x_{pj} - 1), & i \neq j \\ 0, & i = j \end{cases}, \quad 1 \leq i, j \leq n \tag{6.67}$$

算法 6.3 中的计算公式也应做相应改动。

2．双向联想记忆学习算法

双向联想记忆网络是由 Kosko 于 1988 年提出的，简称 BAM。其拓扑结构如图6.14所示，两层网络分别称为 X 层和 Y 层。

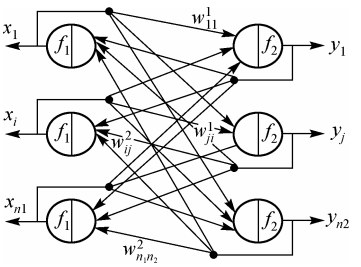


图 6.14　BAM 网络拓扑结构

该网络是两层的反馈网络。网络以前馈和反馈的方式来处理信息，即网络存在输入→输出→输出→输入两层连接权值。在网络的一端输入信号，则可在另一端得到输出，该输出又反馈回来，直至网络达到稳态。

BAM 网络主要用于任意二值模式的联想记忆。该网络既能实现自联想记忆，也能实现异联想记忆。假设网络的 X 层和 Y 层分别有 n_1 和 n_2 个神经元节点，并用 $f_1(\cdot)$ 和 $f_2(\cdot)$ 分别表示两层网络节点各自的作用函数，则网络的动力学描述为

$$y_j(t+1) = f_2\left(\sum_{i=1}^{N_1} w_{ji}^t x_i(t)\right), \qquad j = 1, 2, \mathrm{L}\ , n_2 \tag{6.68}$$

$$x_i(t+1) = f_1\left(\sum_{i=1}^{N_1} w_{ij}^t y_j(t)\right), \qquad i = 1, 2, \mathrm{L}\ , n_1 \tag{6.69}$$

式(6.68)和式(6.69)并未考虑神经元的阈值，在 BAM 网络中一般假设所有神经元的阈值为 0，下面讨论 BAM 网络的学习算法。

BAM 网络联想记忆的学习问题由下述训练样本集合给出：$U = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \mathrm{L}\ , (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$，其中 $\boldsymbol{x}_p = \{+1, -1\}^{n1}$，$\boldsymbol{y}_p = \{+1, -1\}^{n2}$，$p = 1, 2, \mathrm{L}\ , N$。学习目标是形成双向联想记忆网络，使所有训练样本成为网络的稳态或吸引子。

根据训练样本中 $\boldsymbol{X}$ 向量和 $\boldsymbol{Y}$ 向量的维数，立即可设计出网络的结构，网络两层分别由 n_1 和 n_2 个神经元组成，神经元作用函数均采用硬限函数，并假设每个神经元的阈值为 0。因此学习算法只需考虑网络连接权值的计算即可。通常将所有连接权值用两个连接权矩阵 $\boldsymbol{W}^1$ 和 $\boldsymbol{W}^2$ 表示。$\boldsymbol{W}^1$ 表示由 X 层到 Y 层的连接权值矩阵，$\boldsymbol{W}^2$ 表示由 Y 层到 X 层的连接权值矩阵。双向联想记忆的学习算法思想仍然来自于 Hebb 规则，有

$$\boldsymbol{W}^1 = \sum_{p=1}^{N} \boldsymbol{y}_p \boldsymbol{x}_p^{\mathrm{T}} \tag{6.70}$$

$$\boldsymbol{W}^2 = \sum_{p=1}^{N} \boldsymbol{x}_p \boldsymbol{y}_p^{\mathrm{T}} \tag{6.71}$$

显然，网络的两个连接权值矩阵是对称的，即 $\boldsymbol{W}^1 = (\boldsymbol{W}^2)^{\mathrm{T}}$。式(6.70)和式(6.71)即是 BAM 的连接权值计算方法。

【定理 6.9】 若 BAM 学习的训练样本中，所有 $\boldsymbol{X}$ 向量和所有 $\boldsymbol{Y}$ 向量是各自相互正交的，则所有训练样本都会成为由式(6.70)和式(6.71)所得网络的吸引子。

证明：设 $(\boldsymbol{x}_l, \boldsymbol{y}_l)$ 是任意一个训练样本，t 时刻以向量 $\boldsymbol{x}_l$ 输入 BAM 的 X 端时，则 Y 端的输出计算如下：

$$\boldsymbol{y}(t+1) = f_2(\boldsymbol{W}^1 \boldsymbol{x}_l) = f_2\left(\sum_{p=1}^{N} \boldsymbol{y}_p \boldsymbol{x}_p^{\mathrm{T}} \cdot \boldsymbol{x}_l\right) = f_2(\boldsymbol{y}_l \boldsymbol{x}_l^{\mathrm{T}} \boldsymbol{x}^l) = f_2(N_1 \boldsymbol{y}_l) = \boldsymbol{y}_l \tag{6.72}$$

上述计算利用了训练样本中 $\boldsymbol{X}$ 向量的正交性，即

$$\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j = \begin{cases} n_1, & i = j \\ 0, & i \neq j \end{cases} \tag{6.73}$$

类似地，若 t 时刻以向量 $\boldsymbol{y}_l$ 输入 BAM 的 Y 端时，则 X 端的输出为

$$\boldsymbol{x}(t+1) = \boldsymbol{x}_l \tag{6.74}$$

故 $(\boldsymbol{x}_l, \boldsymbol{y}_l)$ 是网络的吸引子。

【例 6.2】 假定有三个训练样本 $(\boldsymbol{x}_1, \boldsymbol{y}_1)$，$(\boldsymbol{x}_2, \boldsymbol{y}_2)$，$(\boldsymbol{x}_3, \boldsymbol{y}_3)$，其中 $\boldsymbol{x}_1 = (1, -1, -1, 1)^{\mathrm{T}}$，$\boldsymbol{x}_2 = (-1, 1, 1, -1)^{\mathrm{T}}$，$\boldsymbol{x}_3 = (1, -1, 1, -1)^{\mathrm{T}}$，$\boldsymbol{y}_1 = (-1, 1, 1)^{\mathrm{T}}$，$\boldsymbol{y}_2 = (-1, -1, -1)^{\mathrm{T}}$，$\boldsymbol{y}_3 = (1, -1, 1)^{\mathrm{T}}$。按照式(6.70)和式(6.71)计算得

$$\boldsymbol{W}^1 = (\boldsymbol{W}^2)^{\mathrm{T}} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & -3 & 3 \\ 3 & -3 & -1 & 1 \end{bmatrix} \tag{6.75}$$

不难验证，$(\boldsymbol{x}_1, \boldsymbol{y}_1)$，$(\boldsymbol{x}_2, \boldsymbol{y}_2)$，$(\boldsymbol{x}_3, \boldsymbol{y}_3)$ 都是网络的吸引子即稳态。

3．HAM与BAM网络模式分类器

HAM 和 BAM 用做模式分类器，需满足如下两个条件：

(1) 所有待分类模式为网络的稳定态，当这些模式作为网络输入时，网络能自动收敛到它们的联想输出状态。

(2) 对于任意输入的模式样本，网络收敛后能自动确定该输入模式为哪个存储模式。

条件(1)可以通过误差校正和正交归一化技术满足。条件(2)联想记忆网络本身则不能解决。因为网络输出端的输出只是输入模式样本的联想记忆结果。由于学习规则和存储模式的限制，输入模式往往收敛到假吸引中心。为使网络的联想输出能够按已存储模式类别判别出来，需要在网络的输出端增加一个后处理器。

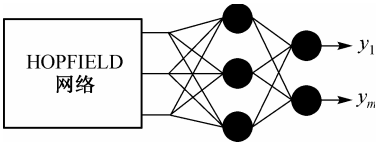
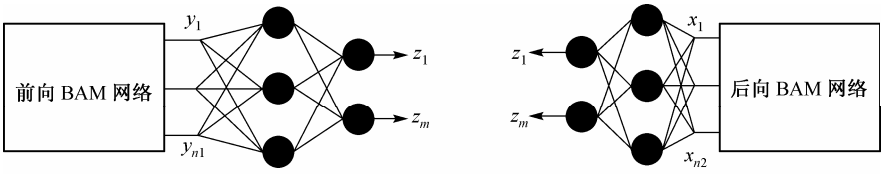


图 6.15　HAM 模式分类器

最直接的方法是在网络的输出端串接一个前馈网络。图6.15 和图6.16 分别是 HAM 加前馈网络和 BAM 加前馈网络形成的模式分类器。



(a) 前向 BAM 模式分类器　　(b) 后向 BAM 网络分类器

图 6.16　BAM 模式分类器

网络在进行实际模式分类时，由前端的联想记忆网络对输入模式进行预处理，再由后端的前馈网络做进一步的模式属性匹配，最后输出结果。

上述两种联想记忆分类器通过在后端增加多层前馈网络而构成。前端的联想记忆网络可采用前面介绍的学习算法形成。后端的前馈网络可以采用 BP 网络或多层感知器。若采用 BP 网络，需要确定两个常数 $C_1(0.8)$ 和 $C_2(0.2)$ 使神经元输出二值化，实际输出大于等于 C_1 时表示 1；小于等于 C_2 时表示 0 即可。后端的 BP 网络需要根据前端联想记忆网络的稳定态及对应的输出模式进行训练。

6.4　海明网络分类学习算法

6.4.1　海明神经网络结构

海明(Hamming)网络可以认为是一种带有局部反馈的前馈神经网络。该模型是 1987 年由 Lippmann 等人提出的[38-40]，常用于各种检测问题。网络的拓扑结构如图6.17所示。网络的输入端节点数是样本模式向量的维数 n，隐层或输出层节点数 m 代表所有样本共分为 m 个类别，由输出节点的输出值确定输入向量的类别。

网络的第一层类似于一个单层感知器，称为匹配子网络，其功能是将输入模式与存储在网络中的标准模式进行模式匹配，确定哪个标准模式与输入模式距

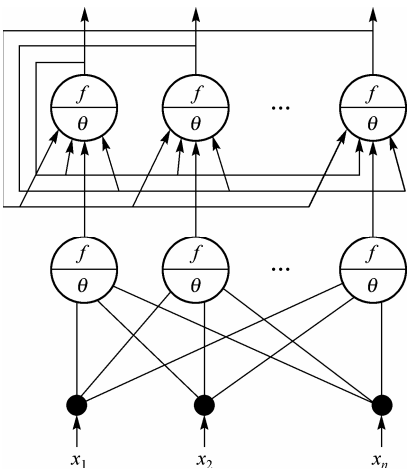


图 6.17　海明神经网络的拓扑结构

离最近。网络第二层称为竞争子网络，竞争子网络与 Hopfield 结构相同，但连接权值的取值不同，其功能是迭代寻找匹配子网络中的最大匹配值输出。海明网络的运行机理是，由第一层网络通过学习将存储模式记忆在网络连接权中，在工作阶段计算输入模式与各样本模式的匹配程度，把结果送入竞争子网络，竞争子网络选择匹配子网络中最大值输出，即选择最佳匹配模式输出，从而实现对离散输入模式进行在海明距离最小意义下的识别及分类。海明网络两个子网络的神经元作用函数均取分段线性函数，即

$$f(\sigma)=\begin{cases}0, & \sigma<0\\ \sigma, & 0\leq\sigma\leq\sigma_c\\ \sigma_c, & \sigma>\sigma_c\end{cases} \tag{6.76}$$

要从海明网络的竞争子网络获得稳定的输出，竞争子网络必须具有竞争收敛性。可以证明，若竞争子网络的连接权值满足

$$w_{ij}=\begin{cases}1, & i=j\\ -\eta, & i\neq j\end{cases},\quad i,j\in\{1,2,\mathrm{L}\ ,m\} \tag{6.77}$$

则竞争子网络的竞争必是收敛的。

6.4.2 海明网络分类学习算法

1．分类原理

二进制信息在信道中传递时，由于各种噪声干扰，可能发生比特位变化。假设二进制位取{+1, –1}两种状态，则发生状态变化的条件概率为

$$\begin{cases}P\left(\dfrac{-1}{+1}\right)=\varepsilon\\ P\left(\dfrac{+1}{+1}\right)=1-\varepsilon\\ P\left(\dfrac{+1}{-1}\right)=\rho\\ P\left(\dfrac{-1}{-1}\right)=1-\rho\end{cases} \tag{6.78}$$

在噪声干扰较小时，有$0\leq\varepsilon<0.5$，$0\leq\rho<0.5$。分类误差最小的分类器是最佳分类器。假设有 M 类模式，每类的代表模式为 $\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_M$。根据最佳分类原则，可利用如下规则判断测试模式 $\boldsymbol{x}$ 属于哪一类：

$$\text{若 } P\left(\frac{\boldsymbol{x}}{\boldsymbol{x}_i}\right)=\max_{j=1}^{M}\left\{P\left(\frac{\boldsymbol{x}}{\boldsymbol{x}_j}\right)\right\},\quad \text{则 } \boldsymbol{x}\in\boldsymbol{x}_i$$

其中，$P\left(\dfrac{\boldsymbol{x}}{\boldsymbol{x}_i}\right)$表示模式 $\boldsymbol{x}$ 的最大输出类条件概率。在 $\varepsilon=\rho$ 时，条件概率

$$P(\boldsymbol{x}\mid\boldsymbol{x}_j)=\varepsilon^{d_H(\boldsymbol{x},\boldsymbol{x}_j)}(1-\varepsilon)^{N-d_H(\boldsymbol{x},\boldsymbol{x}_j)}=\left(\frac{\varepsilon}{1-\varepsilon}\right)^{d_H(\boldsymbol{x},\boldsymbol{x}_j)}(1-\varepsilon)^N \tag{6.79}$$

式中，$d_H(\boldsymbol{x},\boldsymbol{x}_j)$ 表示模式 $\boldsymbol{x}_j$ 与测试模式 $\boldsymbol{x}$ 之间的海明距离。因 ε < 0.5，所以 $\frac{1}{1-\varepsilon}<1$，故 $d_H(\boldsymbol{x},\boldsymbol{x}_j)$ 越小，则 $P\left(\frac{\boldsymbol{x}}{\boldsymbol{x}_i}\right)$ 越大。由此可知，测试模式 $\boldsymbol{x}$ 与已存样本模式 $\boldsymbol{x}_j$ 的最小海明距离对应的输出，就是 $\boldsymbol{x}$ 的类别属性。

要使 $d_H(\boldsymbol{x},\boldsymbol{x}_j)$ 最小，就要使 N–$d_H(\boldsymbol{x},\boldsymbol{x}_j)$ 最大。$N-d_H(\boldsymbol{x},\boldsymbol{x}_j)$ 的计算可以用单层前馈网络实现。欲用单层前馈网络来拟合该变量，需要有

$$N-d_H(\boldsymbol{x},\boldsymbol{x}_j)=\sum_{i=1}^{n}w_{ji}\boldsymbol{x}_i+\theta_j,\quad j=1,2,\mathrm{L}\ ,N \tag{6.80}$$

若二进制信息的模式分量取+1 和 – 1 两个状态，则有

$$N-d_H(\boldsymbol{x},\boldsymbol{x}_j)=\frac{N}{2}+\frac{1}{2}\sum_{i=1}^{n}x_{ji}\boldsymbol{x}_i \tag{6.81}$$

因此 w_{ji} 和 θ_j 可如下取值：

$$\begin{cases}w_{ji}=\dfrac{x_{ji}}{2}\\ \theta_j=\dfrac{N}{2}\end{cases},\quad i=1,2,\mathrm{L}\ ,n,\quad j=1,2,\mathrm{L}\ ,N \tag{6.82}$$

若二进制信息模式分量取 0 和+1 状态，则有

$$N-d_H(\boldsymbol{x},\boldsymbol{x}_j)=n-\sum_{i=1}^{n}x_{ji}+\sum_{i=1}^{n}(2x_{ji}-1)x_i \tag{6.83}$$

因此 w_{ji} 和 θ_j 可如下取值：

$$\begin{cases}w_{ji}=2x_{ji}-1\\ \theta_j=n-\sum\limits_{i=1}^{n}x_{ji}\end{cases},\quad i=1,2,\mathrm{L}\ ,n,\quad j=1,2,\mathrm{L}\ ,N \tag{6.84}$$

这样就将最小海明距离等价地用单层感知器网络输出端对应的最大输出值表示出来。

2．学习算法

海明网络分类问题描述如下：给定训练样本集合 $U=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N\}$，$\boldsymbol{x}_i=(x_{i1},x_{i2},\mathrm{L}\ ,x_{in})^{\mathrm{T}}$，其中，$N$ 个样本模式是样本空间中 N 类模式的代表。欲建立神经网络存储上述 N 个模式，并当网络输入任意一个模式时，网络输出应指明与该输入模式海明距离最近的一类样本模式，即指明输入模式的类别。根据上面的分析可知，这种分类方法的分类误差最小。

海明网络分类器的匹配子网络是由 n 个分段线性作用函数神经元组成的单层前馈网络。因有 N 个样本模式，所以竞争子网络是由 N 个神经元组成的单层竞争反馈网络，每个神经元节点的作用函数均采用分段线性函数，竞争子网络也是输出层网络。通过连接权值的设计实现竞争。设竞争子网络第 j 个神经元的输入连接权值记为 $v_{j1},v_{j2},\mathrm{L}\ ,v_{jN}$，则竞争连接权值可如下给出：

$$v_{jk}=\begin{cases}1, k=j\\ -\eta, k\neq j\end{cases},\ j,k=1,2,\mathrm{L}\ ,N \tag{6.85}$$

可以证明，当 $\eta < 1/N$ 时，竞争子网络的竞争是收敛的。竞争结束时，对应有最大输入的神经元输出大于 0，其他神经元输出均为 0。由此产生输入模式的分类。海明网络的学习和测试算法如下：

【算法 6.4】 海明网络分类器学习和测试

学习阶段

(1) 根据训练样本集合 $U=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N\}$，利用式(6.82)或式(6.84)计算匹配子网络的连接权值和神经元阈值 w_{ji} 和 θ_j，$i=1,2,\mathrm{L}\ ,N$，$j=1,2,\mathrm{L}\ ,n$。

(2) 按照式(6.85)计算竞争子网络的连接权值 $v_{j1},v_{j2},\mathrm{L}\ ,v_{jN}$。

(3) 竞争子网络与匹配子网络间的所有连接权值为 1。

测试阶段

(1) 输入任意测试模式 $\boldsymbol{x}=(x_1,x_2,\mathrm{L}\ ,x_n)$ 至网络。

(2) 在匹配子网络输出端得到 $\boldsymbol{x}$ 与 $\boldsymbol{x}_j$ 的匹配程度输出：

$$y_j(0)=f\left(\sum_{j=1}^{N}w_{ij}x_j-\theta_i\right) \tag{6.86}$$

(3) 开始竞争，直到只有一个竞争顶点保持正值为止：

$$y_j(t+1)=f\left(y_j(t)-\eta\sum_{k\neq j}y_k(t)\right) \tag{6.87}$$

(4) 竞争过程结束时，取胜神经元代表测试模式类别。

(5) 返回(1)继续测试。

利用训练样本集合设计得到海明网络后，输入任一模式 $\boldsymbol{x}$。在匹配子网络输出端得到 $\boldsymbol{x}$ 与 N 类模式的匹配程度输出，由此形成竞争子网络的初始状态。竞争子网络引用横向抑制机制开始竞争，直到输出端只有一个节点输出大于 0。该节点表示输入 $\boldsymbol{x}$ 与样本 $\boldsymbol{x}_j$ 在海明距离意义下是最接近的模式。

6.5 特征映射网络分类学习算法

6.5.1 特征映射网络结构

神经网络的基本目标之一是，研究人类怎样发现、学习和识别自己所处的环境。特征映射网络是一种自组织的神经网络，可以自动地向环境学习，可以对复杂的二维模式进行自组织、自稳定的大规模并行处理。自组织特征映射理论体现了人脑的后天学习过程，如果将感知外界信息的视网膜作为神经网络的输入层，将做出刺激反应的大脑皮层视为网络输出层，则输出层对不同输入模式的反应能力就是自组织能力。也就是说，输出层哪个节点对应于哪个输入模式不是事先确定的，而是通过输出层节点之间的竞争确定的。网络通过节点之间的竞争，输出层获胜的节点就代表输入模式的类别。海明网络也有一个竞争层网络，对于具体

的输入模式，哪个节点竞争获胜实际是人为规定好的。但 Kohonen 的自组织特征映射神经网络则不同。图 6.18 是 Kohonen 特征映射网络的示意图。

网络的输入层为输入模式样本的向量，节点数为输入样本的维数 n。网络输出层将神经元排成了一个节点矩阵，输出节点数为 m。输出节点之间存在局部相互连接。这种网络将输入样本映射到输出层上，形成特征图，它们之间的连接权值是通过无导师竞争学习实现的。该网络输出层特征图的节点之间，在一定邻域存在相互连接，也称为侧向反馈抑制连接。如图6.18所示，侧向反馈的作用表现为类似“墨西哥帽”的函数：

$$g(x)=(1-x^2)\exp\left(-\frac{x^2}{2\alpha}\right) \tag{6.88}$$

式中 α 为控制函数分布的形状参数。“墨西哥帽”抑制函数如图6.19所示。

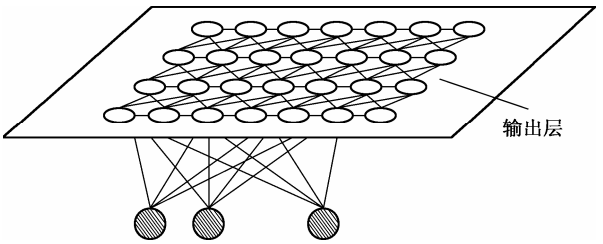


图 6.18　特征映射神经网络

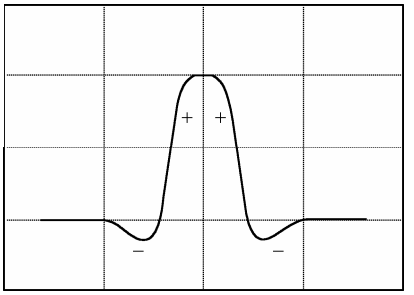


图 6.19　“墨西哥帽”抑制函数

该函数表示在邻域中心附近的节点是相互“激励”的，远一些的神经元则相互抑制，更远的节点之间其作用可以忽略不计。Kohonen 自组织特征映射的基本原理是，当某类模式输入时，其输出层某一节点得到最大刺激而获胜，获胜节点周围的一些节点因侧向作用也受到较大刺激。这时网络进行一次学习操作，获胜节点及其周围节点的连接权值向量向输入模式的方向做相应的修正。当输入模式类别发生变化时，二维平面上的获胜节点也从原来的节点转移到其他节点。这样，网络通过自组织方式用大量训练样本数据来调整网络的连接权值，最后使网络输出层特征图能够体现样本数据的分布情况。根据 Kohonen 网络的输出状况，不仅能判断输入模式所属的类别，使输出节点代表某类模式，而且能够得到整个数据区域的大体分布情况，即从样本数据得到所有数据的分布特征。

6.5.2　特征映射分类学习算法

1．分类原理

假设训练样本集合 $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\mathrm{L}\ ,\boldsymbol{x}_N\}$ 来自 C 类模式集合，其属性和对应的概率密度函数 $f_C(\boldsymbol{x})$ 都是未知的。如果由这些样本模式训练神经网络形成分类器，则在网络收敛后，对于输入的任意模式，网络通过自组织竞争获胜的输出节点就是样本模式的类别属性。自组织学习利用网络连接权值反映样本数据的概率分布，并从样本数据的概率分布中得到输入数据的类别属性。因此期望学习达到如下效果：输入样本按其分布 $f_C(\boldsymbol{x})$ 出现的概率越高，则使其对应的输出值越大。网络收敛后，其对应的权值应成为反映 $f_C(\boldsymbol{x})$ 分布的本质特征向量，这个向量就成为判决输入样本属性的依据。图6.20 说明了通过自组织学习进行模式分类的原理。

在图6.20所示神经网络中，C 个输出层神经元表示所有模式共有 C 个类别，每个输出层神经元的作用函数均采用分段线性函数。输入节点数为 n 表示输入模式为 n 维向量。网络按照如下规则学习：对于相似的输入模式，使同一个输出节点受到最大刺激而激活，输出幅度增加，对于属性特征差别较大的输入模式，则给予不同节点最大刺激。对于输入模式 $\boldsymbol{x}_k$，若输出层第 j 个节点获胜，再次利用 Hebb 规则可给出与该节点相连的权值向量计算方法为

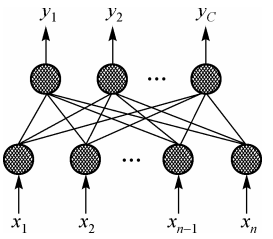


图 6.20 单层神经网络

$$\Delta w_{ji} = \eta y_j x_{ki}, \quad i = 1, 2, \mathrm{L}\ , n, \quad j = 1, 2, \mathrm{L}\ , C \tag{6.89}$$

其中，$0 < \eta < 1$。若网络再次输入向量 $\boldsymbol{x}_k$，则网络的输出变为

$$y_j' = \sum_{i=1}^{n} (w_{ji} + \Delta w_{ji}) x_{ki} = (1+\eta) y_j, \quad j = 1, 2, \mathrm{L}\ , C \tag{6.90}$$

上式表明，如果以后再遇到 $\boldsymbol{x}_k$ 或与 $\boldsymbol{x}_k$ 接近的模式时，节点 j 将以更大的可能性获胜。为了避免训练样本模式的不断增加导致网络连接权值幅度的过度增长，通常对取胜神经元的连接权值修正量的计算方法改为

$$\Delta w_{ji} = \eta y_j (x_{ki} - w_{ji}) \tag{6.91}$$

一般认为，经过按照规则式(6.89)学习的网络权值向量，将不断逼近模式分布的主要能量方向。

2．Kohonen自组织特征映射算法

Kohonen 提出生物视网膜接受外界信息的自适应方程为

$$\frac{\mathrm{d}w_{ji}(t)}{\mathrm{d}t} = \alpha(t)\{\eta_j(t) \cdot x_i(t) - \gamma[\eta_j(t)] \cdot w_{ji}(t)\} \tag{6.92}$$

其中，$\alpha(t)$ 为学习常数，通常 $0 < \alpha(t) < 1$，且随时间单调下降；$\eta_j(t)$ 表示神经元顶点的触发频率；$\gamma(\cdot)$ 为非线性函数。假定在节点 j 的邻域 $NE_j(t)$ 内，$\eta_j(t) = 1$，否则 $\eta_j(t) = 0$。设非线性函数 $\gamma(\cdot)$ 的取值为 $\gamma(0) = 0$ 和 $\gamma(1) = 1$，则式(6.92)进一步写为

$$\begin{cases} \dfrac{\mathrm{d}w_{ji}(t)}{\mathrm{d}t} = \alpha(t)[x_i(t) - w_{ji}(t)], & j \in NE_j(t) \\ \dfrac{\mathrm{d}w_{ji}(t)}{\mathrm{d}t} = 0, & j \notin NE_j(t) \end{cases} \tag{6.93}$$

将式(6.93)写成离散形式，则有

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \alpha(k)[x_i(k) - w_{ji}(k)], & j \in NE_j(t) \\ w_{ji}(k+1) = w_{ji}(k), & j \notin NE_j(t) \end{cases} \tag{6.94}$$

自组织学习基本方法是无监督的学习方法。上述学习方法表明了取胜神经元受到最大激励。一般来说，选择取胜神经元的准则为：若神经元节点 j 取胜，则 j 满足

$$\left\|\boldsymbol{x}(t)-\boldsymbol{w}_j(t)\right\|=\min\{\left\|\boldsymbol{x}(t)-\boldsymbol{w}_i(t)\right\|\} \tag{6.95}$$

自组织特征映射学习算法如算法 6.5 所示。

【算法 6.5】 Kohonen 自组织特征映射算法

(1) $k=0$，对网络连接权随机赋值，取 $\alpha(t)$ 和 $NE_{j^*}(t)$ 随时间变化的形式。

(2) 对网络输入模式 $\boldsymbol{x}=(x_1,x_2,\mathrm{L}\ ,x_N)^{\mathrm{T}}$。

(3) 计算 $\boldsymbol{x}$ 与所有输出节点连接权向量 $\boldsymbol{w}_j$ 的距离：

$$d_j=\sum_{i=1}^{n}(x_i-w_{ji})^2,\quad j=1,2,\mathrm{L}\ ,C \tag{6.96}$$

(4) 选择具有最小距离的节点 j^* 为获胜节点：$d_{j^*}=\min\limits_j\{d_j\}$

(5) 调整节点 j 及其几何邻域 $NE_{j^*}(t)$ 内节点所连接的权值向量：

$$\boldsymbol{w}_j(k+1)=\boldsymbol{w}_j(k)+\alpha(k)[\boldsymbol{x}(k)-\boldsymbol{w}_j(k)],\quad j\in NE_{j^*}(t) \tag{6.97}$$

(6) 对新的样本数据，转到(2)继续训练，若无样本，则结束。

上述算法实际可在网络分类计算的同时进行。对获胜节点邻域的选择，在开始时应选择较大的邻域半径，随着训练过程的进行，邻域半径逐渐缩小，最后只包含获胜神经元。Kohonen 自组织特征映射算法无须知道训练样本的属性，只需通过自组织方式进行模式聚类。当学习成功时，网络输出层的特征图相当于对模式数据进行了特征提取。自组织特征映射算法是否对任意分布的模式数据均能得到对应模式属性的特征图是非常重要的问题。这表现为自组织分类学习的收敛性，许多实验结果验证了上述收敛性。但自组织学习是否对任意分布的样本模式都是收敛的，目前并没有给出严格证明[41]。

6.6　前馈网络分类机理

多层感知器和多层 BP 网络具有相同的拓扑结构，差别在于神经元不同。BP 算法与单层感知器学习算法的基本计算规则是按相同的思路得到的，BP 算法又称广义 δ 规则学习算法，实际上，前馈神经网络都可以采用广义的 δ 规则实现其学习过程。但基于梯度下降方法来训练前馈神经网络需要花费很长的时间。有监督的前馈网络训练时间长短与算法、网络规模、网络结构及分类样本模式的特征均有关系。

单层网络的训练时间要比多层网络的训练时间少得多，感知器可在模式空间形成超平面分割，而多层感知器在期望输出与实际输出于误差平方和最小的情况下，其输出端的输出值对应模式样本后验概率的估计[42]。这一结论揭示了多层感知器分类与传统贝叶斯分类的等价性。另外，有监督学习的前馈网络分类器是一种特殊的函数逼近器，网络学习是约束网络连接权向量，将不同类别样本模式映射到输出端对应的监督信号过程。

6.6.1　前馈网络分类的几何机理

1．超平面分割

用于模式分类的有监督学习前馈网络是按如下原理进行分类计算的：第一隐层的作用是在模式空间实现各种超平面分割；第二隐层实现第一隐层输出的逻辑与运算，即将分割的超平

面空间按类别进行空间划分；输出层又对第一隐层的输出值进行逻辑或运算，即将经过与运算的、属于同一类的超平面进行归类。图6.21是多层感知器的单个神经元非线性作用示意图。

假设该神经元是第一隐层的第 i 个节点，该节点接受到的总激励输入为

$$\sigma_i = \sum_{j=1}^{n} w_{ij}x_j - \theta_i = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} - \theta_i \tag{6.98}$$

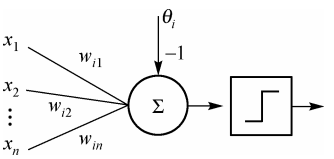


图 6.21 隐层顶点非线性作用示意图

其中，$\boldsymbol{w}_i = (w_{i1}, w_{i2}, \mathrm{L}\ , w_{in})^{\mathrm{T}}$，$\boldsymbol{x} = (x_1, x_2, \mathrm{L}\ , x_n)^{\mathrm{T}}$。假设隐层单元的非线性作用函数为硬限函数，即

$$y_i = f(\sigma_i) = \begin{cases} 1, & \sigma_i \geq 0 \\ 0, & \sigma_i < 0 \end{cases} \tag{6.99}$$

考察式(6.98)，$\sigma_i = 0$ 表示模式空间中的一个超平面，将该模式空间分割成两部分。如果只考虑二维情况，则 $\sigma_i = 0$ 是二维空间中的一条直线，如图6.22所示。此时，$\boldsymbol{w}_i$ 表示直线的方向向量，θ 为该直线在纵轴上的截距。

基于广义 δ 规则的前馈网络学习算法对神经元权值和阈值的计算，均采用了先给出一个初始值然后逐渐修正的计算步骤。实质上，上述学习过程就是自适应调整 $(\boldsymbol{w}_i, \theta_i)$ 的过程，也就等效为超平面位置与取向不断改变的过程。这样，多个神经元节点将在空间中形成多个超平面，这些超平面将模式空间划分为若干区域，图6.23给出了一种形象的说明。

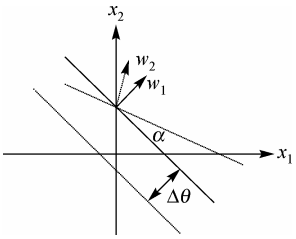


图 6.22 二维空间直线分割

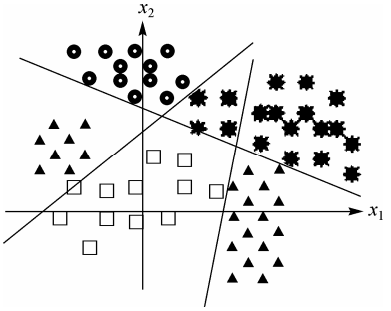


图 6.23 二维空间中训练点被超平面分割

若设 $\boldsymbol{w}_i' = (w_{i1}, w_{i2}, \mathrm{L}\ , w_{in}, \theta_i)^{\mathrm{T}}$，$\boldsymbol{x}' = (x_1, x_2, \mathrm{L}\ , x_n, -1)^{\mathrm{T}}$，则式(6.98)可写成

$$\sigma_i = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} - \theta_i \tag{6.100}$$

该超平面与坐标原点的距离为 $\dfrac{|\sigma_i|}{\|\boldsymbol{w}_i'\|}$。设前馈网络的输入模式为 n 维向量。在增维的模式空间 R^{n+1} 中定义一个以 $x_0 \in R^{n+1}$ 为中心的超球 $S(\boldsymbol{x}_0', r)$：

$$S(\boldsymbol{x}_0', r) = \{\boldsymbol{x}' : \| \boldsymbol{x}' - \boldsymbol{x}_0' \| \leq r\}, r \in R^+ \tag{6.101}$$

若取 $\boldsymbol{x}_0'$ 为 R^{n+1} 中的坐标原点，则 $\sigma_i = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} - \theta_i = 0$ 是经过坐标原点的一个超平面 HP_w。它与超球面 $S(\boldsymbol{x}_0', r)$ 之间的关系如图6.24所示。

对于任一模式向量 $\boldsymbol{x}$，设与其垂直的超平面为 HP_x，则 $\boldsymbol{x}'$ 与 $\boldsymbol{w}'$ 之间的夹角为

$$\theta = \arccos\frac{\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}'}{\| \mathbf{w}' \| \| \boldsymbol{x}' \|}, 0 \leq \theta \leq 180° \tag{6.102}$$

若 $0 \leq \theta \leq 90°$，则 $\sigma_i > 0$，对应于超平面 HP_w 的正侧 HP_w^+，若 $90° \leq \theta \leq 180°$，$\sigma_i < 0$，对应于超平面的负侧 HP_w^-。

由此可知，多层感知器在训练中，第一层权值决定了超平面在空间的取向。网络在训练中不断修正连接权值和阈值，标志着由超平面所形成的对模式空间的分割在不断地改变。在隐层单元足够多的情况下，由超平面所形成的线性可分区域将无穷多。

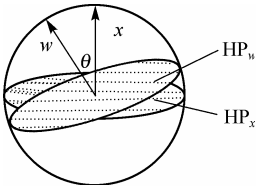


图 6.24　超球与超平面

2．输入层-隐层映射的定性解释

线性可分性是指对于空间上存在的若干模式类别，若能找到一些超平面，使每个超平面都不会经过任何类别模式区域，而将不同类别模式分割开来。

对于两类模式，线性可分是指可以找到一个超平面，使一类模式位于超平面一侧，另一类模式位于超平面另一侧。定理 6.4 指出，只有对于线性可分的模式分类问题，单层感知器才能学习成功。这表明了单层感知器本身具备的映射能力，同时条件也是十分苛刻的。

若样本模式是线性可分的，则只需用一层网络就可以实现模式分类，由单层网络的输出表达输入模式的类别。但实际模式分类问题，一般不能保证样本模式的线性可分性，因此一层超平面只能将同类模式分割在不同区域中。这样就必须使用多层感知器才能完成模式分类，第一层网络对应的超平面完成模式分割，每个区域只有一类模式，而由网络的后级将同类而不同区域的模式联合划归为一类。

Mirchandani 和 Cao 导出基于超平面分割所得的最大线性可分区域与隐层节点数 N 及输入模式向量维数 n 的关系为[44]

$$R(N, n) = \sum_{k=0}^{n} C_N^k \tag{6.103}$$

这是空间中所有超平面均相交的情况。对于存在平行超平面的情况，Mehrotya 给出了更一般的分析结果。图6.25画出了输入向量维数为 n = 6, 7, 8 的情况下所形成的最大线性可分区域与隐单元数的关系曲线。显然，隐层节点数增多，由超平面所形成的最大线性可分区域也明显增加。由 N 个超平面围成的每个线性可分区域称为一个有效判别区域。

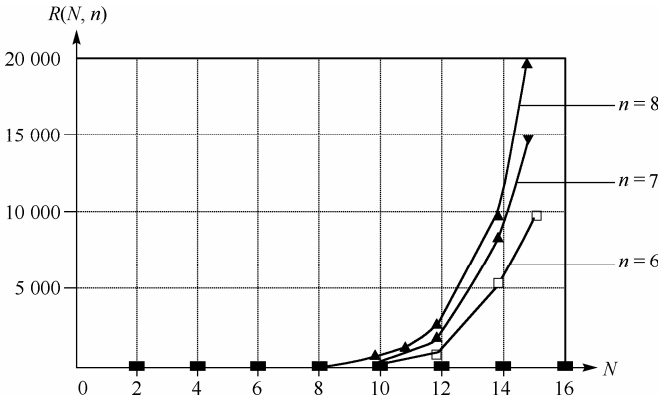


图 6.25　$R(N, n)$ 与隐层节点数 N 的关系曲线

在实际问题的学习计算中，每个有效判别区域的形成有一定随机性，同时，有效判别区

域的形成还取决于多层感知器是如何训练的。如果使用序贯方法来逐次训练神经网络，则在训练的初始阶段，由先参与训练的样本决定的判决区域具有某种取向，而其他判决区域不表示任何信息，可以认为是随机的。但是随着训练的进行，形成的空间有效判决区域逐渐趋于某一确定的位置，每一个区域至少对应一个训练样本。相反，若使用批处理方法来训练网络，那么所形成的有一定取向的判决区域数为批处理的样本数。随着训练的继续，网络隐层所形成的有效判决区域也趋向于确定的位置。在广义δ规则学习过程中，将输出端残留的误差信号反馈回到各层的连接权值上，调整每个超平面的空间取向，也就是调整隐层单元决定的判决区域位置，来使输出误差信号的平方和接近终止精度。所以对于多层感知器，第一层网络的连接权值在训练过程中存在一定的随机性，但当网络收敛后，其取值是确定的，对应的有效判决区域是一个由随机到确定的过程。

一般在多层感知器中，隐层单元越多，则实现的非线性映射能力越强。因此，若网络的隐层单元数扩展到无限多，则可以实现任意的映射。在实际应用中，可以通过增加隐层单元数来增强网络的容错能力。若网络节点数超过了一定的数量，将会因空间中超平面数过多而引起噪声。

3．超曲面分割网络

感知器的神经元首先对输入量进行线性加权求和获得神经元激励总和，然后通过硬限作用函数完成非线性变换。其线性的加权求和计算使中间层神经元对应的输入/输出特征表现为模式空间中超平面对模式的分割，不同类别的模式样本被分割在不同的超空间区域内。然而，在实际应用中，某些不规则的空间分布模式的分类，用超平面分割往往要用很多隐层节点，而且需要很长的训练时间才能形成稳定的超空间，有时甚至很难从模式空间中将某个模式分割出来，表现为网络训练不收敛。这是由于由超平面所构成的超空间体与模式的空间分布区域相差甚远。例如，某些模式的空间分布边界凹凸不匀，表现为非常复杂的空间超曲面，这时，若仍用超平面去分割与逼近就不能满足要求。所以，必须寻找其他超空间分割方法，如超球面分割、超椭球面分割或超抛物面分割等。只需改变神经元由输入分量计算输入总量的方法即可获得其他形式的分割，如下所示。

(1)超球面分割

$$\sigma_i = \sum_j (x_j - w_{ij})^2 + \theta_i \tag{6.104}$$

(2)超椭球面分割、超抛物面分割或超双曲面分割

$$\sigma_i = \sum_j w_{ij} x_j^2 + \theta_i \tag{6.105}$$

(3)任意多项式分割

$$\sigma_i = \sum_j \sum_k (w_{ijk} x_j x_k + x_j^{\alpha} + x_k^{\beta} + \theta_i) \tag{6.106}$$

(4)一般曲线分割

$$\sigma_i = \sum_j \Phi(x_j, w_j) + \theta_i \tag{6.107}$$

超曲面分割网络对模式的旋转变化具有不变性，这类网络在字母、图像识别中将具有广泛的应用前景。

6.6.2　前馈网络分类的代数机理

1．前馈网络的输入输出映射

前馈网络无论是用于模式识别还是用于函数逼近，都是将输入模式空间映射到输出响应空间内，对于模式识别问题，输出响应信号即为外监督信号。若前馈网络的神经元均采用线性作用函数，则这种网络称为线性前馈网络。不论线性前馈网络有多少层，都可以用单层的线性前馈网络等价地实现。从数据所含的信息量来考虑多层的线性前馈网络的映射作用实际是一种数据的压缩变换。每一隐层对前一层的输入模式进行 Fisher 压缩变换[27]，使同类别的模式散布最小，不同类别的模式散布最大，使经变换后的模式有利于输出层的分类。

用线性网络来解决模式分类问题，只能实现线性映射，其能力是有限的，通常所用神经网络具有非线性单元网络，而且实际使用的人工神经网络完全超出生物神经网络的范畴。可以根据问题的特点，在隐层构造特殊的非线性传输函数来适合问题的需要。

2．代数分类机理

线性前馈网络的分类能力是有限的，所完成的只是一种数据压缩变换。为使前馈神经网络能够实现任意的映射，需在前馈网络的中间层增加非线性单元。考虑两层的前馈神经网络，设中间层和输出层网络的神经元分别有 N 和 M 个神经元节点，则网络的输入/输出关系可以表示为

$$y_k = f_k\left(\sum_{j=1}^{M} w_{kj}^{(2)} g_j\left(\sum_{i=1}^{N} w_{ji}^{(1)} x_i - \theta_i^{(1)}\right) - \theta_k^{(2)}\right) \tag{6.108}$$

式中，$w_{kj}^{(2)}$ 表示第 j 个隐层节点到第 k 个输出节点的连接权值，也称为第二层连接权；$w_{ji}^{(1)}$ 表示第 i 个输入节点到第 j 个隐层节点的连接权值，也称为第一层连接权；$\theta_j^{(1)}$ 表示第 j 个隐层节点对应的阈值；$\theta_k^{(2)}$ 表示第 k 个输出节点对应的阈值；$f_k(\cdot)$ 表示第 k 个输出节点对应的非线性作用函数；$g_j(\cdot)$ 表示第 j 个隐层节点对应的非线性传输函数。

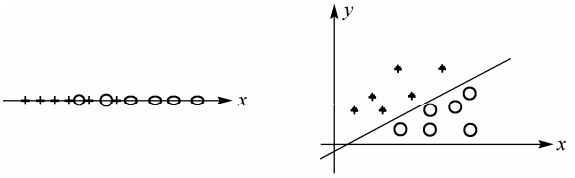
一般来说，$f_k(\cdot)$ 或者是线性的，或者是 sigmoid 函数；$g_j(\cdot)$ 一般取 sigmoid 函数，或其他平方可积的函数形式等。下面分析这种网络的结构特征和分类机理。

(1)输入模式空间到隐层变换空间的映射

线性前馈网络的映射是一种数据压缩分类，会降低样本的维数，存在牺牲分类信息的缺陷。对神经网络分类器来说，因其大规模并行处理的特点，一般无需使用降维处理方法。升维处理却是一般前馈神经网络最常用的方法。升维处理可能会增加计算量，但对于低维中不可分的模式，通过扩展模式样本的拓扑维数，有可能在高维中变得线性可分。

图6.26 是一组两类模式在一维空间不能线性可分而升到两维后线性可分的示意图。

图 6.26(a)表示两类模式在一维数轴上存在部分重叠的情况，如果将它们扩展到二维空间，如图6.26(b)所示，显然，两类模式变得线性可分了。这说明了升维处理的必要性。



(a) 原始模式分布　　(b) 映射到高维空间的模式分布

图 6.26　映射前后的模式分布

假设两类 R^n 空间的模式样本 $\boldsymbol{x}, \boldsymbol{y}$，如果将它们扩展到 R^m 空间内（$m > n$），分别表示为

$$\begin{cases} \boldsymbol{x} = (x_1 x_2, \mathrm{L}\ , x_n, \underbrace{\times, \mathrm{L}\ , \times}_{m-n})^{\mathrm{T}} \\ \boldsymbol{y} = (\underbrace{\times, \mathrm{L}\ , \times}_{m-n}, y_1, y_2, \mathrm{L}\ , y_n)^{\mathrm{T}} \end{cases} \tag{6.109}$$

式中的×为不确定项，它们由网络的结构和学习算法来自适应确定。如果这两类模式样本在 R^n 空间是重叠的，经过式(6.107)的变换在 R^m 空间内可能变得线性可分。神经网络正是基于这种原理对模式信息进行分类的。网络的隐层构造成非线性单元后，与直通形式的线性节点不同，这些隐节点通过调整其阈值和选择非线性作用函数，几乎都变成独立的。

在下面的讨论中，为了简化符号，避免复杂性，忽略权值 $\boldsymbol{w}$ 与阈值 θ 的区别。设隐层有 M 个非线性单元，样本模式总数为 N。另设模式分类问题共有 C 类模式，第 k 类模式给出 N_k 个训练样本模式，则 $N = N_1 + \mathrm{L}\ + N_C$，将训练样本写成矩阵形式为

$$\boldsymbol{X} = \left[\boldsymbol{x}_1^{(1)}, \mathrm{L}\ , \boldsymbol{x}_{N_c}^{(c)} \right]^{\mathrm{T}} \tag{6.110}$$

则经第一层连接权值 $\boldsymbol{W}^{(1)}$ 变换的隐层输入端的总线性仿射函数 $\boldsymbol{Z} \in R^{N \times M}$：

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}^{(1)} \tag{6.111}$$

$$\boldsymbol{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \mathrm{L} & w_{1M}^{1} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ w_{N_1 1}^{(1)} & w_{N_1 2}^{(1)} & \mathrm{L} & w_{N_1 M}^{(1)} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ w_{11}^{(c)} & w_{12}^{(c)} & \mathrm{L} & w_{1M}^{(c)} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ w_{N_1 1}^{(c)} & w_{N_1 1}^{(c)} & \mathrm{L} & w_{N_1 1}^{(c)} \end{bmatrix} \in R^{N \times M} \tag{6.112}$$

该线性仿射函数矩阵 $\boldsymbol{Z}$ 经隐层非线性函数 $g(\cdot)$ 作用后，隐层输出为

$$\boldsymbol{X} = g(\boldsymbol{Z}) = \begin{bmatrix} x_{11}^{(1)} & x_{12}^{(1)} & \mathrm{L} & x_{1M}^{1} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ x_{N_1 1}^{(1)} & x_{N_1 2}^{(1)} & \mathrm{L} & x_{N_1 M}^{(1)} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ x_{11}^{(c)} & x_{12}^{(c)} & \mathrm{L} & x_{1M}^{(c)} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ x_{N_c 1}^{(c)} & x_{N_c 2}^{(c)} & \mathrm{L} & x_{N_c M}^{(c)} \end{bmatrix} \in R^{N \cdot M} \tag{6.113}$$

$\boldsymbol{X}$ 中的前 N_1 行表示第一类模式在隐层的输出，记为 $\boldsymbol{X}^{(1)}$，依次至最后 N_C 行表示第 C 类模式在隐层的输出，记为 $\boldsymbol{X}^{(C)}$。若输入端至隐层的连接权值矩阵 $\boldsymbol{W}^{(1)}$ 发生变化，则经隐层非线性变换后的这些类别的矩阵也将自适应地调整，来跟踪对应的变化。

(2) 隐层变换空间到输出监督空间的映射

输入端训练集样本经隐层映射后，在隐层输出端张成新的模式空间，待分类的 C 个类别分别由 $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \mathrm{L}\ , \boldsymbol{X}^{(C)}$ 矩阵的行空间组成。因非线性的作用，原模式的特性经变换后可能发生变化，但与线性判别相比，这种变换没有损失任何分类信息，不仅如此，而且模式被扩展到更高维空间后，有可能使原非线性可分的模式变得线性可分。

分析表明[28-45]，不论输出层单元的作用函数是线性的还是非线性的，隐层至输出层的映射总可以用线性映射分析，即

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}^{(2)} \tag{6.114}$$

式中的 $\boldsymbol{W}^{(2)} \in R^{M\times C}$ 为隐层至输出层的连接权值矩阵。网络连接权值的训练学习是使隐层模式空间不断逼近输出期望的外监督信号空间。输入层至隐层连接权值的调整，决定着输入模式变换到隐层后的空间几何分布，隐层至输出层连接权值的调整，决定了隐层模式空间逼近外监督信号空间的精度。假设在无限训练时间的前提下，终止逼近精度的大小将与隐层节点数的多少、非线性函数的形式及外监督信号的特征有关。式 (6.112) 只是在最小均方差意义上成立，即

$$\min_{\boldsymbol{W}} \left\| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}^{(2)} \right\|_F^2 \tag{6.115}$$

该式是通常在网络输出端定义的目标函数。网络训练的目的是调整两层权值矩阵 $\left\{\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}\right\}$，使式 (6.115) 在最小均方差意义上最小。

6.7　径向基函数网络[50]

6.7.1　径向基函数

径向基函数 (Radial Basis Function, RBF) 是在多维空间中插值的传统技术，广泛应用在图像处理、信号处理和控制工程等领域[45-47]。RBF 函数最早是由 Powell [46]在 1985 年提出的。问题描述如下。

给定一个点集及 n 维空间的相应实值 $\left\{\boldsymbol{x}_i, y_i\right\}$，$i = 1, 2, \mathrm{L}\ , m$，要求满足插值条件的函数 $f(\boldsymbol{x})$：

$$f(\boldsymbol{x}_i) = y_i,\quad i = 1, \mathrm{L}\ , m \tag{6.116}$$

插值函数 $f(\boldsymbol{x})$ 的形式为

$$f(\boldsymbol{x}) = \lambda_0 + \sum_{i=1}^{m} \lambda_i \phi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_i\right\|\right) \tag{6.117}$$

在 RBF 方法中，$\phi\left(\left\|\boldsymbol{x} - \boldsymbol{x}_i\right\|\right)$，$i = 1, 2, \mathrm{L}\ , m$ 称为基函数，函数 $f(\boldsymbol{x})$ 从该基函数导出。基函数径向对称，每一个基函数的中心都位于给定的数据点上，$\phi(\cdot)$ 为线性函数，例如

$$\phi(\alpha)=(\alpha^2+c^2)^{-\frac{1}{2}} \tag{6.118}$$

将插值条件 $f(\boldsymbol{x}_i)=y_i$，$i=1,2,\mathrm{L}\ ,m$ 代入插值函数表达式，可得到具有 m 个未知系数 λ_i 的 m 个方程。如果矩阵 $\boldsymbol{A}=(\boldsymbol{A}_{ij})_{m\times n}$ 奇异，则方程没有唯一解。其中，

$$\boldsymbol{A}_{ij}=\varphi\left(\left\|\boldsymbol{x}_i-\boldsymbol{c}_j\right\|\right),\ \ i,j=1,\mathrm{L}\ ,m \tag{6.119}$$

只要数据点互不相同，一般此矩阵都是非奇异的。

6.7.2 径向基函数网络的特点

RBF 方法和神经网络之间有着很强的联系。运用神经网络通过复杂的数据建模可以视为在多维空间实现一条曲线，Broomhead 和 Lowe [48]根据这个观点揭示了多层前向网络和 RBF 网络之间的关系，他们提出可以人为地选择径向基函数中的可变因素，并允许训练数据点和基函数的个数不同，这样就可以根据大大少于数据点的信息来降低复杂性；其次，基函数的中心可以在数据点上，也可以不在数据点上，这将提高径向基函数的推广能力[50]。

放宽内插条件后的 RBF 网络可以视为一个具有线性参数的两层网络。网络结构见图 6.27[50]。假定选择了合适的 RBF 中心和非线性变换函数，则隐层实现了将输入空间映射到新的空间，每一个基函数成为隐层神经元的激活函数。输出层在新的空间实现线性组合。隐层到输出层的可调权值可以用线性最小二乘方法得到。

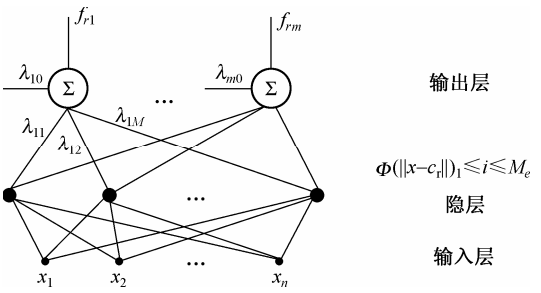


图 6.27 径向基函数网络

常用的隐层激活函数 $\Phi(\cdot)$ 函数如下。

(1) 薄板样条函数：

$$\Phi(v)=v^2\log(v) \tag{6.120}$$

(2) 高斯函数：

$$\Phi(v)=\exp\left(\frac{-v^2}{\beta}\right) \tag{6.121}$$

(3) MQ 函数 (mutiquadric function)：

$$\Phi(v)=(v^2+\beta^2)^{\frac{1}{2}} \tag{6.122}$$

其中 β 是实常数。

最常用的函数 $\Phi(\cdot)$ 是高斯函数，其表达式描述如下：

$$O_j=\exp\left(\frac{-\left\|\boldsymbol{x}-\boldsymbol{c}_j\right\|^2}{\sigma_j^2}\right),\qquad j=1,\mathrm{L}\ ,N_r \tag{6.123}$$

其中 O_j 是隐层第 j 个单元的输出，$\boldsymbol{x}$ 是输入模式，$\boldsymbol{c}_j$ 是隐层第 j 个单元基函数的中心，也可以视为该单元的权向量，σ_j^2 是第 j 个隐节点的归一化参数，它决定该中心点对应的基函数的作

用范围。隐节点输出在 0 到 1 之间，输入与中心的距离越近，隐节点的响应就越大。高斯函数径向对称，对于与基函数中心径向距离相同的输入，隐节点都产生相同的输出。

在径向基函数网络里，非线性映射由非单调增减的高斯函数的线性组合得到，而不像 BP 网络那样是单调增减 Sigmoid 函数。由隐层作用函数产生的表面，形似一系列山丘。这些山丘的高度可由第二层神经元的标量权值调节。如果提供足够数目的隐层神经元，则通过选择合适的中心、归一化参数和输出权，就可以很好地逼近所要描述的非线性函数。一般地，RBF 网络中所利用的非线性函数形式对网络性能的影响并不是至关重要的，关键是基函数中心的选取。从数据点中任意选取 RBF 中心构造出来的 RBF 网络的性能是不能令人满意的，这些中心还可能因为靠得太近而产生近似线性相关，从而带来数值上的病态问题[50]。

多层前馈网络和径向基函数网络从理论上说，都能以任意精度逼近任意非线性映射，只是其基函数的性质不同。Sigmoid 函数在输入空间的无限大区域内都是非零的，而 RBF 函数只覆盖局部区域，一些问题可以利用 Sigmoid 基函数有效地解决，一些问题则更适合用局部覆盖的径向基函数来解决。径向基函数网络又有它独特的优势。径向基函数网络和前馈多层网络相比，后者因为网络输出与参数之间是高度非线性的，网络权值必须通过某种非线性优化技术如梯度下降法进行学习。这就不可避免地存在局部极小问题，即在网络权值学习过程中使参数估计陷入优化指标函数的某一个局部极小。除此而外，BP 网络还有收敛速度慢、网络拓扑难以确定等缺点。遗传算法、学习自动机、模拟退火算法等虽然可以避免局部极小，但一般都需要巨大的计算量，从而极大地限制了前馈网络的实时应用。径向基函数理论为多层前馈网络提供一种崭新且有效的方法。首先，径向基函数网络输出对网络参数是部分线性的，即当网络基函数中心确定之后，网络输出对输出层的权参数是线性的。这样就可采用线性优化中常用的最小二乘等算法，因而不存在局部极小问题，也避免了反向传播算法那样繁琐、冗长的计算，使学习可以比通常的 BP 算法快 10^3~10^4 倍[49]。

【例6.3】 用径向基函数网络解决回归问题，数据是由噪声正弦函数产生的。比较三种不同的激活函数的影响，首先用高斯激活函数。使用两个阶段的训练算法，首先用 EM 迭代算法定位中心，然后用设计矩阵的伪逆寻找第二层的权值。第二种 RBF 网络是薄板样条激活函数，使用和高斯激活函数相同的中心，因此只需要计算第二层的权值。第三种 RBF 网络是 $r^4 \log r$ 激活函数。数据、函数和网络输出比较结果如图6.28所示。

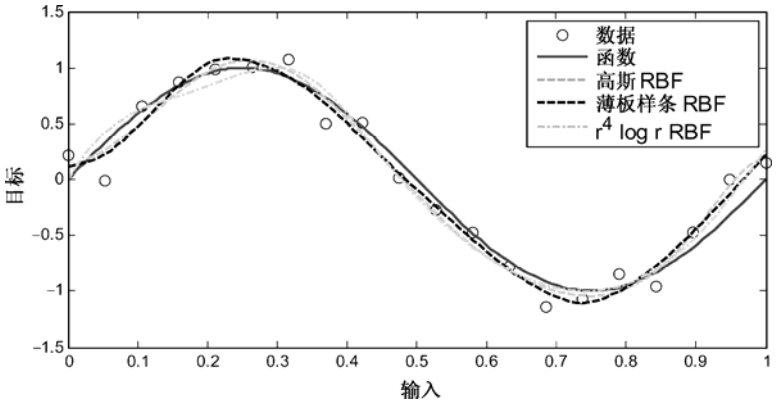


图 6.28　RBF 实现回归问题的数据、函数和输出结果

6.7.3 径向基函数网络的正则化

通过样本学习的输入、输出映射可将神经网络视为多维函数的逼近，即超平面重构问题。学习问题与经典的逼近技术如广义样条、正则化理论密切相关。

1．正则化理论

用一组数据 $S=\{(x_i, y_i)\in R^n\times R, i=1,\mathrm{L}\ , N\}$ 来逼近函数 f，由正则化方法确定函数 f 使得下面的泛函最小：

$$H(f)=\sum_{i=1}^{N}(y_i-f(\boldsymbol{x}_i))^2+\lambda\|Pf\|^2 \tag{6.124}$$

P 为一约束算子(通常为一微分算子)，$\|\cdot\|$ 是一个在函数空间的范数(一般是 L^2 范数)。λ 为正实数，称为正则化参数。算子 P 的结构体现了关于解的先验知识，因此依赖于待解问题的性质。最小化函数 H，可得到相关联的欧拉–拉格朗日方程：

$$\hat{P}Pf(\boldsymbol{x})=\frac{1}{\lambda}\sum_{i=1}^{N}(y_i-f(\boldsymbol{x}_i))\delta(\boldsymbol{x}-\boldsymbol{x}_i) \tag{6.125}$$

其中 $\hat{P}$ 是微分算子 P 的伴随矩阵。式(6.125)中等号右边是对 H 关于 f 的泛函微分，是一个偏微分方程，它的解能被写成其右边项的一个积分变换。它具有由微分算子 $\hat{P}P$ 的格林函数所确定的核，格林函数满足下面的分布式偏微分方程：

$$\hat{P}PG(\boldsymbol{x};\boldsymbol{y})=\delta(\boldsymbol{x}-\boldsymbol{y}) \tag{6.126}$$

由于存在 δ 函数，积分变换是离散和的形式：

$$f(\boldsymbol{x})=\frac{1}{\lambda}\sum_{i=1}^{N}(y_i-f(\boldsymbol{x}))G(\boldsymbol{x};\boldsymbol{x}_i) \tag{6.127}$$

上式说明正则化问题的解位于平滑函数空间中的一个 N 维子空间上，该子空间的基由 N 个函数 $G(\boldsymbol{x};\boldsymbol{x}_i)$ 给出。由于格林函数通常是平移不变的，即 $G=G(\boldsymbol{x}-\boldsymbol{x}_i)$，这时 $G(\boldsymbol{x})$ 和 $G(\boldsymbol{x}-\boldsymbol{x}_i)$ 在一个将 $\boldsymbol{x}_i$ 映射到原点的坐标变换下是等价的。故称 $G(\boldsymbol{x};\boldsymbol{x}_i)$ 为中心点在 $\boldsymbol{x}_i$ 的格林函数，称点 $\boldsymbol{x}_i$ 为展开中心。若令 $\boldsymbol{c}_i=(y_i-f(\boldsymbol{x}_i))/\lambda$，将 N 个数据点代入式(6.127)，可直接得到下面线性系统：

$$(G+\lambda\boldsymbol{I})\boldsymbol{c}=\boldsymbol{y} \tag{6.128}$$

其中 $\boldsymbol{I}$ 是单位矩阵，且 $(\boldsymbol{y})_i=y_i$，$(\boldsymbol{c})_i=y_i$，$(\boldsymbol{G})_{ij}=G(\boldsymbol{x}_i;\boldsymbol{x}_j)$。因此可得正则化问题的解为

$$f(\boldsymbol{x})=\sum_{i=1}^{N}c_iG(\boldsymbol{x}_i;\boldsymbol{x}_j) \tag{6.129}$$

其中 $\boldsymbol{c}$ 由式(6.128)得出。

式(6.127)不是最小化问题的完全解，事实上，位于算子 P 的零空间的函数对于泛函方程(6.124)中的平滑项都是“不可见”的，故上面的展开式是完全解对位于 P 的零空间中某一项取模。这一项的形式取决于所选择的稳定子(stabilizer)和边界条件，因而取决于要解决的具体问题。对于一个稳定子，它是一个同构的旋转不变 n 维算子，零空间是 $2n-1$ 维的多项式空间。

算子 $\hat{P}P$ 是自伴随的，它的格林函数对称，即 $G(\boldsymbol{x};\boldsymbol{y})=G(\boldsymbol{y};\boldsymbol{x})$。因而矩阵 $\boldsymbol{G}$ 对称，其特征值为实数，矩阵 $\boldsymbol{G}+\lambda\boldsymbol{I}$ 满秩(除非 $-\lambda$等于它的某个特征值)。这样，线性系统总有解。在$\lambda=0$ 的情况下，解的存在性(对应于纯粹的插值)依赖于格林函数的性能。如果格林函数正定，上式的展开则内插数据点，不存在零空间项；若格林函数在某些阶条件正定，近似理论的结论保证了对式(6.127)添加适当阶次的多项式，即 P 的零空间的多项式使得它总能插值数据点。格林函数的条件正定及其他性能由稳定子 P 的结构决定。若 P 平移不变，$\boldsymbol{G}$ 将依赖于其自变量的差；若 P 旋转平移不变，$\boldsymbol{G}$ 为一个径向基函数，$\boldsymbol{G}=G\left(\|\boldsymbol{x}-\boldsymbol{y}\|\right)$，这一种情况的正则化解由下面的展开给出：

$$f(\boldsymbol{x})=\sum_{i=1}^{N}c_i G\left(\|\boldsymbol{x}-\boldsymbol{x}_i\|\right) \tag{6.130}$$

这样就产生了径向基函数方法[49]。

2．正则化网络

正则化网络可以由只含一层隐层单元的简单网络来实现，如图6.28所示。网络第一层的单元数等于问题的独立变量数。第二层由非线性隐层单元组成。每个数据点 $\boldsymbol{x}_i$ 对应于一个隐层单元，输入单元与第 i 个隐层单元之间的连接由第 i 个数据点的坐标确定。隐单元的激活函数是格林函数，则第 i 个隐单元的输出为 $G(\boldsymbol{x};\boldsymbol{x}_i)$。输出层包括一个或多个线性单元，其权值是展开式(6.127)的未知系数。求解可以用梯度下降法使数据点上插值误差最小，此时令 $\lambda=0$。如果格林函数是正定的，则该解是使泛函$\|Pf\|^2$最小的最优插值；如果格林函数是条件正定的，要得到最优插值则应对网络添加适当的多项式单元。

正则化网络完全取决于所学习的问题，其输入与隐层之间的连接权是已知的。正则化网络具有三个期望的性质：

(1)如果有足够多的单元，正则化网络就能在紧域上任意逼近多变量连续函数。代数和三角多项式同样具有这一性能。

(2)因为从正则化理论得出的未知系数是线性的，容易证明它具有最佳逼近性能。也就是说，给定函数 f，总是存在一组系数选择，要比其他所有选择都能更好地逼近 f。一些传统的逼近方法，如多项式逼近和具有固定节点的样条逼近等，都具有最佳逼近性质，它们的逼近解与未知参数呈线性关系。

(3)用正则化网络计算出来的解是最优的。这就消除了那些在数据点处精确插值而在无数据点处振荡的解。这一性能对样条插值是典型的，但多项式插值并不具备。

在正则化网络中，输出单元也可能计算一个固定的非线性可逆函数。这对分类问题非常有用，非线性函数常常选用 sigmoid 函数。输入单元也可以是一个非线性函数。适当的输入、输出处理在某些情况下具有优势。用双有理函数、指数函数、对数函数和这三个函数的组合来作为非线性处理函数，因为它们能达到必要的域和范围的转换，并使得输入、输出空间的代数结构破坏最小。输入、输出的这种编码通过利用要逼近映射的域和范围的先验信息尽可能线性化这个逼近。

3．正则化方法的推广

超基函数网络是更一般化的正则化网络，它可以实现基于目标的分类并降低维数。在前面讨论的基础上可做两点推广：

(1) 与式(6.129)相关的网络的复杂度(单元数目)与输入空间的维数无关，而是依赖于训练集的维数(样本数目)，通常样本的数目非常庞大。对式(6.127)进行近似使得单元数目远远小于样本数目，并使展开中心在学习过程中被修正。这个方案还能进一步推广到使式(6.129)由不同类型的函数 G 叠加而成的情形，如不同高度的高斯函数。

(2) 范数 $\|\boldsymbol{x}-\boldsymbol{x}_i\|$ 可视为加权范数：

$$\|\boldsymbol{x}-\boldsymbol{x}_i\|_W^2=(\boldsymbol{x}-\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{x}_i) \tag{6.131}$$

式中 $\boldsymbol{W}$ 为方阵。当 $\boldsymbol{W}$ 为对角阵时，对角元 W_{ij} 为每个输入坐标分配一个权，若 $\boldsymbol{W}$ 为单位矩阵，就化为一般的欧几里得范数。当输入为不同类型时，它起着重要的作用。

① 滑动中心：正则化解近似

正则化方法对于逼近问题在当训练样本数目很大时计算代价很高，而且也可能出现病态问题。因此引入正则化解的近似。

寻找变分问题近似解的一般方法是，将解在有限基上展开。近似解有如下形式：

$$f^*(\boldsymbol{x})=\sum_{i=1}c_i\phi_i(\boldsymbol{x}) \tag{6.132}$$

其中 $\{\phi_i\}_{i=1}^n$ 是一组线性无关函数。系数 c_i 通常是按照能保证与真正解偏差最小的规则来确定的。在标准正则化情况下，待最小化的函数由式(6.122)给出，这时若 $n=N$，并且 $\{\phi_i\}_{i=1}^{nN}=\{G(\boldsymbol{x};\boldsymbol{x}_i)\}_{i=1}^N$ (G 为算子 $\hat{P}P$ 的格林函数)，则这种方法给出的就是准确解，这时可将式(6.130)代入正则化泛函求出展开式中的未知系数。此时正则化泛函为 $H[f^*]=H^*(c_1,\mathrm{L}\ ,c_N)$，通过将正则化泛函对系数极小化，可求出系数：

$$\frac{\partial H\left[f^*\right]}{\partial c_i}=0\qquad i=1,\mathrm{L}\ ,N \tag{6.133}$$

如果格林函数在所考虑的区域边界上为零，则方程组(6.133)与方程组(6.128)是等价的。对更一般的情况，基函数应扩大，以包括产生算子 P 的零空间项，由此得到正确解。对准确解的近似是

$$\boldsymbol{f}^*(\boldsymbol{x})=\sum_{\alpha=1}^n c_\alpha G(\boldsymbol{x};\boldsymbol{t}_\alpha) \tag{6.134}$$

中心 t_α 和系数 c_α 是未知参数。中心的数目一般小于数据点的数目，即 $n\leqslant N$。

② 不同类型的基函数

在式(6.134)中，可以利用不同类型的函数 G 相叠加，如不同高度的高斯函数。待逼近的函数 f 可视为 p 个分量 f^m 的和，$m=1,\mathrm{L}\ ,p$，每个分量具有不同的先验概率。因此欲最小化的泛函就有 p 个稳定子 P^m，可写成

$$H[f]=\sum_{i=1}^N\left(\sum_{m=1}^p f^m(\boldsymbol{x}_i)-\boldsymbol{y}_i\right)^2+\sum_{m=1}^p\lambda_m\left\|P^m f^m\right\|^2 \tag{6.135}$$

通过分析与上式相关的欧拉–拉格朗日方程可知，使式(6.135)泛函最小的函数是与稳定子 P^m 对应的格林函数的线性叠加的线性叠加。对该变分问题的一个近似解形如

$$f^*(\boldsymbol{x})=\sum_{m=1}^{p}\sum_{\alpha=1}^{K_m}c_\alpha^m G^m(\boldsymbol{x};\boldsymbol{t}_\alpha^m) \tag{6.136}$$

其中 $K_m < N$，系数 c_α^m 和中心 $\boldsymbol{t}_\alpha^m$ 待定。

这种方法能够产生大范围的径向基函数来重构函数 f。如果已知欲逼近函数有 p 个范围为 $\sigma_1,\mathrm{L}\ ,\sigma_p$ 的元素，就可以用这个信息选择 p 个稳定子，其格林函数选做方差为 $\sigma_1,\mathrm{L}\ ,\sigma_p$ 的高斯函数。这样，解将是不同方差的高斯函数的线性叠加的线性叠加。

③ 加权范数与正则化

如果 $\boldsymbol{x}$ 的各个分量属于不同类型，就可以利用加权范数，定义为

$$\|\boldsymbol{x}\|_{\boldsymbol{W}}^2=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{x} \tag{6.137}$$

正则化原则在于寻找使泛函

$$H_{\boldsymbol{W}}[f]=\sum_{i=1}^{N}(y_i-f(x_i))^2+\lambda\|Pf\|_y^2 \tag{6.138}$$

最小的 f，式中假设 P 关于变量 $\boldsymbol{y}$ 径向对称，而 $\boldsymbol{y}=\boldsymbol{W}\boldsymbol{x}$。这意味着平滑性约束是在原 $\boldsymbol{x}$ 空间的一个仿射变换空间中给出的。与上式对应的格林函数为 $G\left(\|\boldsymbol{y}\|^2\right)=G\left(\|\boldsymbol{x}\|_w^2\right)$。如果与滑动中心方案结合起来，则正则化问题的近似解有如下形式：

$$f^*(\boldsymbol{x})=\sum_{\alpha=1}^{n}c_\alpha G\left(\|\boldsymbol{x}-\boldsymbol{t}_\alpha\|_{\boldsymbol{w}}^2\right) \tag{6.139}$$

如果 $\boldsymbol{W}$ 参数是未知的，则问题为寻找使泛函 $H_{\boldsymbol{w}}[f]$ 最小的 f 和 $\boldsymbol{W}$，故寻找最优的 $\boldsymbol{W}$ 对应于在坐标系中寻找最佳稳定子。

最简单的情况是 $\boldsymbol{W}$ 为对角阵且 $G(x)=\mathrm{e}^{-x^2}$。此时，

$$G\left(\|\boldsymbol{x}\|_w^2\right)=\mathrm{e}^{-x_1^2w_1^2}\mathrm{e}^{-x_2^2w_2^2}\ \mathrm{L}\ \mathrm{e}^{-x_n^2w_n^2} \tag{6.140}$$

这样 $\boldsymbol{W}$ 的对角元 w_i 等价于多维高斯函数各分量的方差 σ 的倒数。

习题 6

6.1　人工神经元网络也可用来进行模式识别，它与统计模式识别原理上是否相同？

6.2　人工神经元网络用于模式识别所能做的，是否用传统模式识别方法都可以做？

6.3　证明如果隐单元的激活函数是线性的，那么三层网络等价于二层网络。

6.4　利用习题 6.3 的结论解释为什么具有线性隐单元的三层网络不能解决某个非线性可分问题，如 XOR 问题。

6.5　考虑具有 d 个输入单元、n 个隐单元、c 个输出单元及偏置的一个标准三层 BP 网络。(a) 网络中有多少权值？(b) 考虑权值对称。证明：如果将每一个权值的符号反向，网络功能不变。(c) 考虑隐单元的对称交换。隐单元上没有标记，因此它们可以相互交换(沿着对应权值)而使网络功能不受影响。证明该等价标记数——对称交换因子为 $n2^n$，在 n 等于 10 的情况下估计该因子的值。

6.6　设计一个 2 层的感知器网络实现 A XOR B。

参考文献

[1] 刘增良, 刘有才. 模糊逻辑与神经网络. 北京：北京航空航天大学出版社, 1996.

[2] 朱大铭. 人工神经网络的结构学习算法及问题求解研究[D]. 中国科学院计算技术研究所, 1999.

[3] Hebb D. O., *The Organization of Behavior*. John Wiley, New York, NY, 1949.

[4] Simpson, P. K., *Artificial Neural Systems: Foundations, paradigms, Applications and Implementations*. Pergamon press, Elmsford, NY, 1990.

[5] Simpson, P. K., *Fuzzy min-max neural networks-part 1: Classification*. IEEE Transactions on Neural Networks, 1992, 3 (5):776-786.

[6] Simpson, P K., *Fuzzy min-max neural networks-part 2*: *clustering*. IEEE Transactions on Fuzzy Systems, 1993, 1 (1):32-45.

[7] Hopfield, J. J., *Neural networks and physical systems with emergent collective computational abilities*. Proc. Nat. Acad. Sci., 1982, 79:2554-2558.

[8] Hopfield, J. J., *Neurons with graded response have collective computational properties like those of two-state neurons*. Proc. Natl. Acad. Sci., 1984, 81:3088-3092.

[9] Steinbuch, K, and U. Piske, *Learning matrixes and their applications*. IEEE Transactions on Electronic Computers, 1963, EC-12:846-862.

[10] Willshaw D., *Holography, associative memory, memory, and inductive generalization*. In parallel Models of Associative Memory, J. Anderson and Q Hinton, Eds., Lawrence Erl-baum, Hillsdale, NJ, 1980.

[11] Hecht-Nielsen, R., *Neurocomputing*. Addison-Wesley. Reading, MA., 1990.

[12] Oja, E., *A simplified neuron model as a principal component analyzer*. Journal of Mathematical Biology, 1982, 15:2Cr7-273.

[13] Kohonen T., *Self-Organization and Associative Memory* (2^{nd} Edition). Springer-Verlag, Berlin, 1988.

[14] Kohonen T., *Tutorial: Self-organizing feature maps*. IEEE International Joint Conference on Neural Networks, Washington, DC, 1989.

[15] Carpenter G A. and S. A. Grossberg., *A massively parallel architecture for a self-organizing netual pattern recognition machine*. Computer Vision, Graphics, and Image Understanding, 1987, 37:54-115.

[16] Carpenter G. A., and S. A. Grossberg., *ART2: self-organization of stable category recognition codes for analog input patterns*. Applied Optics, 1987, 26 (23):4919-4930.

[17] R. Eberthart, P. Simpson and R. bobbins, *Computational Intelligence PC Tools*, AP Professional press, Boston London, 1996.

[18] Simpson, p., and T. Brotherton, *Fuzzy neural network machine prognosis.* proc. Aero-sense 95: Applications of Fuzzy Logic Technology II, SPIE-The International Society for Optical Engineering, Bellingham, WA, 1995, 2493: 21-27.

[19] Rumelhart, D.E., G.E. Hinton and R. J. Williams, *Learning representations by back-propagating errors*. Nature, 1986, 323 (9): 533-536.

[20] Rumelhart, D.E., and J. L. McClelland, *Parallel Distributed processing, Explorations in the Microstructure of Cognition*, Vol. 1: Foundations. MIT Press, Cambridge, MA, 1986.

[21] R. A. Jacobs, *Increased rates of convergence through learning rate adaptation*, Neural Networks, Vol. 1, 1988.

[22] Ackley D., G Hinton, and T. Sejnowski, *A learning algorithm for Boltzmann machines*. Cognitive Science, 1985, 9:147-169.

[23] Rosenblatt F., The perceptron: *A probabilistic model for inforation storage and organization in the brain*. Psychological Review, 1958, 65:386-408.

[24] Rosenblatt F., *Principles of Neurodynamics*. Spartan Books, Washington, DC, 1962.

[25] Rosenblatt F., *A model for experiential storage in neural networks*. In Computer and Information Sciences: Collected papers in Learning, Adaptation and Control in Information Systems, J. T. Tou and R. H. Wilcox, Eds. Spartan Books, Washington, DC, 1964.

[26] Widrow B. and M. E. Hoff, *Adaptive switching circuits*. 1960 IRE WESCON Convention Record: part 4, Computers: Man-Machine Systems, Los Angeles, 1960, CA:96-104.

[27] 黄德双. 神经网络模式识别系统理论. 北京：电子工业出版社, 1996.

[28] 孙增圻. 智能控制理论与技术. 北京：清华大学出版社，南宁：广西科学技术出版社，1997.

[29] L. W. Chan and F. Fallside, *An adaptive training algorithm for back-propagation networks*. Computer Speech and Language, No. 2, 1987.

[30] R. L. Bankert, P. Rabindra and S. K. Sengupta, *A probabilistic neural networks approach to cloud classification*, AD-A247916, Oct., 1991.

[31] R. L. Streit and T. E. Luginbuhl, *Maxmum liklihood training of probabilistic neural networks*, IEEE Trans. NN, 1994, 5(5):764-783.

[32] 钟义信, 潘新安, 杨义先. 智能理论与技术——人工智能与神经网络，北京：人民邮电出版社，1992

[33] Mahdad Nouri Shirazi, etc. *The Capacity of Associative Memories with Malfunctioning*. IEEE Trans. Neural. 1993, 4(4).

[34] Y Hirai, *A model of Human Associative Processor*. IEEE Trans. Syst, Man, Cybern. 1983, VoLSMC-13:851-857.

[35] Qing Ma, *Adaptive Associative Memories Capable of Pattern Segmentation*. IEEE Trans. Neural. 1996, 7(6): 1439-1449.

[36] Zheng-ou Wang, *A Bidirectional Associative Memory Based on Optimal Linear Associative Memory*. IEEE Trans. Computer, 1996, 9(34).

[37] R. J. McEliece, etc. *The capacity of Hopfield Associative Memory*. IEEE Trans. Inform Theory. 1987, Vol IT 33:461-482.

[38] R. P. Lippmann, *An introduction to computing with neural nets*, IEEE ASSp Magzine, 4-22, apr, 1987.

[39] R. P. Lippmann, *Pattern classification using networks*, IEEE Communications Magazine, 47-64, Nov., 1989.

[40] R. P. Lippmann, B. Gold and M. L. *Malpass, A comparison of Hamming and Hopfield neural nets for pattern classification*, AD-A182-255, May, 1987.

[41] 朱大铭, 马绍汉. 自组织特征映射神经网络学习收敛性分析，计算机研究与发展，1997, 34(2)99-106.

[42] D. W. Ruck, S. K. Rogers, et al., *The multilayer perceptron as an approximation to a Bayes optimal discrimination function*, IEEE Trans. NN, 1990, I(4): 296-298.

[43] Q Mirchandani and W. Cao, *On hidden nodes for neural nets*, IEEE Trans. CAS, 1989, 36(5):661-664.

[44] D. E. Cotter, *The Stone-Weistrass theorem and its application to neural networks*, IEEE Trans. Neural Networks, Dec. 1990.

[45] C. A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Constructive Approximation, Vol.2, pp.11-22, 1986.

[46] M. I. D. Powell, *Radial basis functions for multivariable interpolation: a review*, IMA Conf. on Algorithms for the Approximation of Functions and Data, RMCS Shrivenham, 1985.

[47] M. I. D. Powell, *Radial basis function approximations to polynomials*, 12th Biennial Numerical Analysis Conf., Dundee, pp.232-241, 1987.

[48] D. S. Broomhead and D. Lowe, *Multivariable functional interpolation and adaptive networks*, Complex Systems, Vol.2, pp.321-355, 1988.

[49] 陈小红. 径向基函数神经网络及其在非线性控制中的应用[D]，浙江大学, 1996.

第 7 章　支持向量机

支持向量机(Support Vector Machine, SVM)是在统计学习理论基础上发展起来并借助最优化方法来解决机器学习问题的新工具，它最初于 20 世纪 90 年代由 Vapnik[1]提出，近年来在其理论研究和算法实现方面都取得了突破性进展，开始成为克服“维数灾难”和“过学习”等问题的有效方法。目前，在许多领域都获得了成功的应用，如模式识别、回归分析、综合评价和经济预测等领域，逐渐成为新的研究热点。

7.1　最优分类超平面

以二维空间上的分类问题为例来说明最优超平面的思想[2]。如图7.1 所示的二维空间上的分类问题 1，讨论函数 $g(\boldsymbol{x})$ 为线性函数 $g(\boldsymbol{x})=(\boldsymbol{w}\cdot\boldsymbol{x})+b$ 的情况。此时分类问题是要寻找一条合适的直线划分整个二维平面，即确定法线方向 $\boldsymbol{w}$ 和截距 b。

能将两类样本点正确分开的直线有很多，如直线 H_1，假设它的法线方向为 $\boldsymbol{w}$，不改变法线方向，平行地向右上方或左下方推移直线 H_1，直到碰到某类训练样本点。这样就得到了两条极限直线 H_2 和 H_3，称这两条直线之间的距离为与该法线方向相应的“间隔”。如图7.2 所示，应该选取使“间隔”达到最大的法线方向。对于选定的法线方向 $\boldsymbol{w}$，有两条极限直线，选取 b 使得要找的直线为两条极限直线“中间”的那条直线。

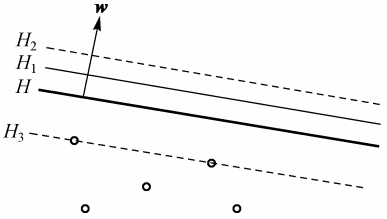


图 7.1　分类问题 1

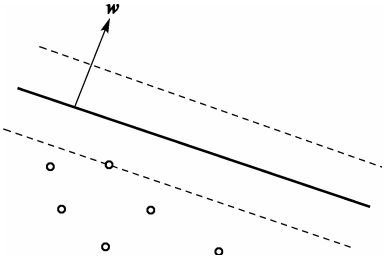


图 7.2　最大间隔

用直线方程分别来表示这三条直线，由于表示同一条直线的方程有很多，所以先将直线方程规范化，即调整 $\boldsymbol{w}$ 和 b，使得两条极端的直线 H_2 和 H_3 分别表示为

$$(\boldsymbol{w}\cdot\boldsymbol{x})+b=1 \text{ 和 } (\boldsymbol{w}\cdot\boldsymbol{x})+b=-1 \tag{7.1}$$

而中间的分隔直线为

$$(\boldsymbol{w}\cdot\boldsymbol{x})+b=0 \tag{7.2}$$

可分超平面必须满足约束条件 $y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\geq 1$，$i=1,\mathrm{L}\ ,l$，点 $\boldsymbol{x}$ 到超平面的距离是

$$d=\frac{|(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b|}{\|\boldsymbol{w}\|} \tag{7.3}$$

最优超平面是由最大化间隔 ρ 给出的。ρ 满足等式约束条件 $y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\geq 1$，$i=1,\mathrm{L}\ ,l$，间隔为

$$\rho=\min_{x_i:y_i=-1}\frac{|(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b|}{\|\boldsymbol{w}\|}+\min_{x_i:y_i=1}\frac{|(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b|}{\|\boldsymbol{w}\|}=\frac{1}{\|\boldsymbol{w}\|}\left(\min_{x_i:y_i=-1}|(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b|+\min_{x_i:y_i=1}|(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b|\right)=\frac{2}{\|\boldsymbol{w}\|} \tag{7.4}$$

故相应的两条极限直线距离——相应的“间隔”为 $2/\|\boldsymbol{w}\|$。极大化“间隔”的思想导致求解下列对变量 $\boldsymbol{w}$ 和 b 的最优化问题：

$$\Phi(\boldsymbol{w})=\max_{\boldsymbol{w},b}\frac{2}{\|\boldsymbol{w}\|} \tag{7.5}$$

即对于所有使 $y_i=-1$ 的下标 i，有 $(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b\geq 1$ (7.6)

对于所有使 $y_i=-1$ 的下标 i，有 $(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b\leq -1$ (7.7)

或

$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2 \tag{7.8}$$

$$y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\geq 1,\ \ i=1,\mathrm{L}\ ,l \tag{7.9}$$

由式(7.8)和式(7.9)的最优解 $\boldsymbol{w}^*$ 和 b^*，就可以得到要寻找的直线和决策函数 $f(x)=\mathrm{sgn}((\boldsymbol{w}^*\cdot\boldsymbol{x})+b^*)$。特别地，最优超平面方法不直接求解问题式(7.8)和式(7.9)，而是通过求解该问题的对偶问题来得到它的解。引入式(7.8)和式(7.9)的对偶式：

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) \tag{7.10}$$

$$\text{s.t.}\quad\sum_{i=1}^{l}y_i\alpha_i=0 \tag{7.11}$$

$$\alpha_i\geq 0\quad i=1,\mathrm{L}\ ,l \tag{7.12}$$

得到对偶式(7.10)~式(7.12)的解后，利用最优化理论中原始问题和对偶问题解的关系来得到原始问题式(7.8)和式(7.9)的解，从而得到决策函数。

【例 7.1】 表7.1所示为线性可分的数据点。利用最优超平面分类的结果如图7.3所示。

表 7.1 线性可分数据

x_1	x_2	y
1	1	−1
3	3	1
1	3	1
3	1	−1
2	2.5	1
3	2.5	−1
4	3	−1

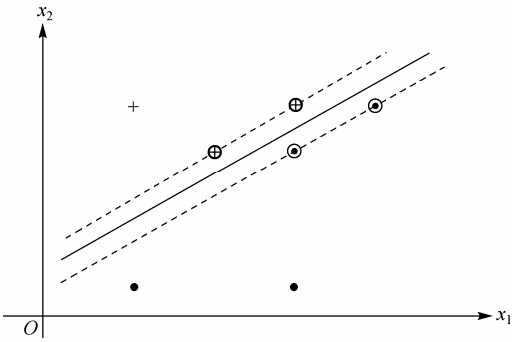


图 7.3 最优可分超平面

图7.3中的虚线表示间隔的轨迹，圆圈包围的点为支持向量。

上述方法是对二维空间的分类问题 1 导出的，事实上对于一般 n 维空间中类似的分类问题也适用，但对于图7.4所示的分类问题 2 不适用，对于图7.4所示的问题不能用直线正确地划分训练集，如果仍然用直线去划分，必然会出现错分点。因此，放宽要求，希望错分的程度尽可能小，即不要求所有训练点满足约束条件 $y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\geqslant 1$。为此对第 i 个训练点 $(\boldsymbol{x}_i, y_i)$ 引进松弛变量 $\xi_i>0$，把约束条件放松为 $y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)+\xi_i\geqslant 1$。显然 $\sum_{i=1}^{l}\xi_i$ 描述训练集被错划的程度，这样现在就有两个目标：仍希望间隔 $2/\|\boldsymbol{w}\|$ 尽可能大；同时希望错划程度 $\sum_{i=1}^{l}\xi_i$ 尽可能小。引进一个惩罚参数 C，把这两个目标综合起来，即极小化新的目标函数 $\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i$，这里 C 作为两个目标的权重，因此得到式(7.13)至式(7.15)的最优化问题：

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i \tag{7.13}$$

$$\text{s.t.}\quad y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)+\xi_i\geqslant 1,\ \ i=1,\mathrm{L}\ ,l \tag{7.14}$$

$$\xi_i>1,\ \ i=1,\mathrm{L}\ ,l \tag{7.15}$$

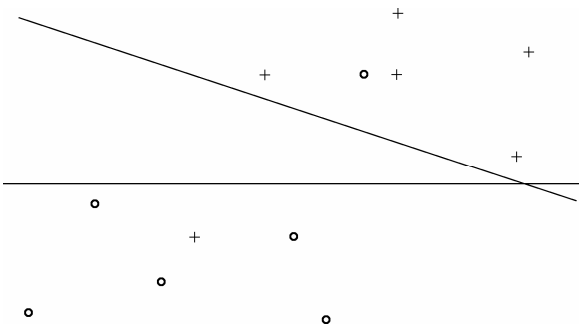


图 7.4　分类问题 2

根据该问题的最优解 $\boldsymbol{w}^*$, b^*, ξ^*来构造决策函数 $f(x)=\mathrm{sgn}((\boldsymbol{w}^*\cdot\boldsymbol{x})+b^*)$。与前述类似，引入问题式(7.13)~式(7.15)的对偶问题：

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{l}\alpha_j \tag{7.16}$$

$$\text{s.t.}\quad \sum_{i=0}^{l}y_i\alpha_i=0 \tag{7.17}$$

$$0\leqslant\alpha_i\leqslant C,\quad i=1,\mathrm{L}\ ,l \tag{7.18}$$

得到对偶问题的解后，利用最优化理论中原始问题和对偶问题解的关系来得到原始问题式(7.13)~式(7.15)的解，最后得到决策函数。

【例 7.2】 对线性可分问题中的数据加上 2 个数据点得到非线性可分数据点如表7.2所示。

表 7.2 非线性可分数据

x_1	x_2	y
1	1	−1
3	3	1
1	3	1
3	1	−1
2	2.5	1
3	2.5	−1
4	3	−1
1.5	1.5	1
1	2	−1

对于非线性可分数据 $C=1$ 时的分类结果如图7.5所示。

此时支持向量不再如图7.3一样只位于间隔内，超平面的方向和间隔的宽度都有所不同。当 $C\to\infty$ 时非线性可分问题趋于线性可分超平面，如图7.6所示。

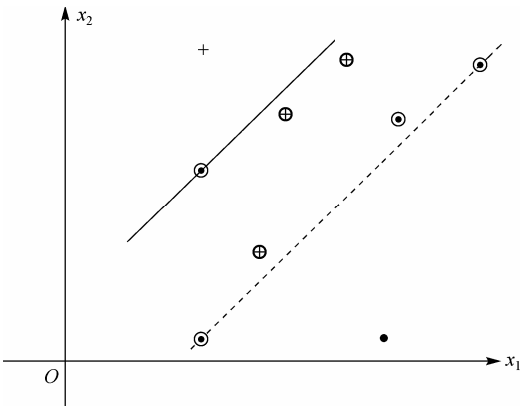


图 7.5 广义线性可分最优超平面分类结果（$C=1$）

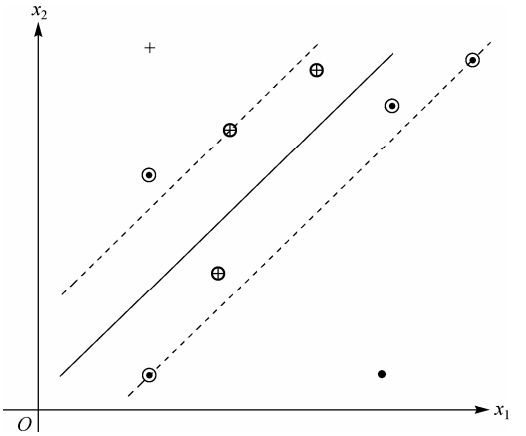


图 7.6 $C=10^5$ 时的广义最优超平面

当 $C\to 0$ 时，非线性可分解收敛于间隔最优域，如图7.7所示。此时具有最小的错分率，但是产生了最大化间隔，因此当 C 减小时，间隔宽度增加。

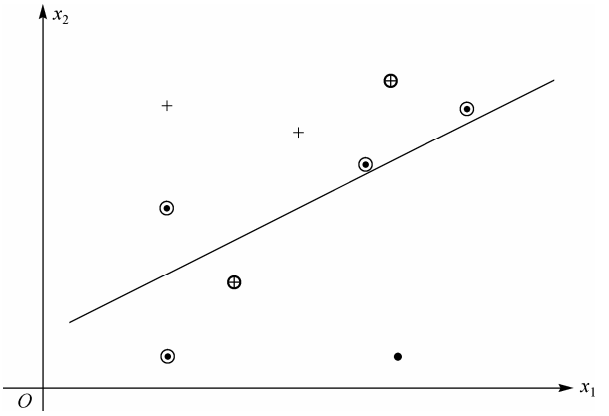


图 7.7 $C=10^{-8}$ 时的广义超平面

7.2　支持向量机的理论基础

支持向量机通过某种事先选择的非线性映射将输入向量 $\boldsymbol{x}$ 映射到高维特征空间 F，在特征空间 F 中构造最优分类超平面，如图7.8所示[1]。

本节介绍支持向量机的三种分类形式及其理论基础——统计学习理论和最优化理论，其中包括 VC 维理论、结构风险最小化原则、基于间隔的推广估计以及凸规划的 Wolfe 对偶及 KKT 条件等理论。

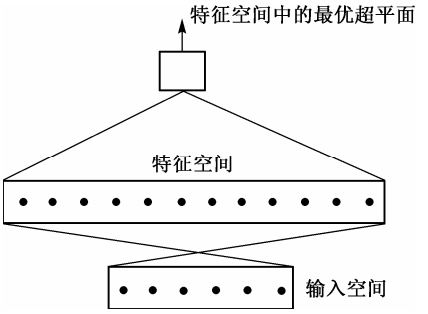


图 7.8　输入空间映射到特征空间[1]

7.2.1　支持向量机的三种分类形式

分类问题是根据给定的训练集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),\mathrm{L},(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\mathrm{L},l$，寻找决策函数

$$f(\boldsymbol{x})=\mathrm{sgn}(g(\boldsymbol{x})) \tag{7.19}$$

推断任一模式 $\boldsymbol{x}$ 相对应的 y 值。当 $g(\boldsymbol{x})=(\boldsymbol{w}\cdot\boldsymbol{x})+b$ 时为线性函数，由决策函数(7.19)确定分类准则时，称为线性分类学习问题；当 $g(\boldsymbol{x})$ 为非线性函数时，称为非线性分类学习问题。下面以输入为二维向量的问题为例，分别阐述相应的分类学习问题。

1．线性可分问题

对于图7.9所示的问题，很容易用一条直线把训练集正确地分开(即两类点分别在直线的两侧，没有错分点)，这类问题就是前述的线性可分问题。应用最大“间隔”思想，通过求解最优化问题式(7.8)~式(7.9)的对偶式(7.10)~式(7.12)，根据对偶问题的解得到原问题的解，来确定决策函数。

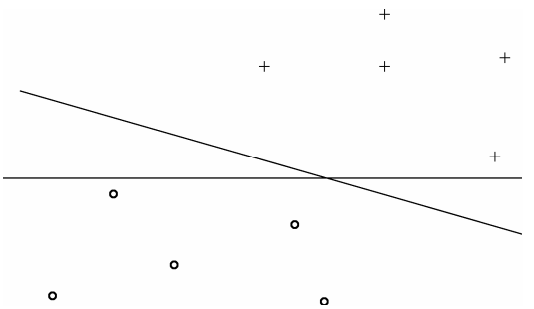


图 7.9　线性可分问题

算法的具体步骤如下。

【算法 7.1】 线性可分支持向量机

(1) 设已知训练集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),\mathrm{L},(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\mathrm{L},l$。

(2) 构造并求解最优化问题

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{l}\alpha_j \tag{7.20}$$

$$\text{s.t.}\quad\sum_{i=0}^{l}y_i\alpha_i=0 \tag{7.21}$$

$$\alpha_i \geqslant 0, \quad i=1,\mathrm{L}\ ,l \tag{7.22}$$

得最优解 $\alpha^*=(\alpha_1^*,\mathrm{L}\ ,\alpha_l^*)$。

(3) 计算 $\boldsymbol{w}^*=\sum_{i=1}^{l} y_i\alpha_i^*\boldsymbol{x}_i$；选择 α^* 的一个分量 $\alpha_j^*>0$，并据此计算 $b^*=y_i-\sum_{i=1}^{l} y_i\alpha_i^*(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)$。

(4) 构造划分超平面 $(\boldsymbol{w}^*\cdot\boldsymbol{x})+b^*=0$，由此求得决策函数 $f(x)=\mathrm{sgn}((\boldsymbol{w}^*\cdot\boldsymbol{x})+b^*)$。

图7.10为标准超平面的示意图，给出了与每个超平面距离最近的点。

关于“支持向量机”

“支持向量机”中的“机器”是一个算法。在机器学习领域，常把一些算法视为一个机器，称分类算法为分类机(或分类器)。“支持向量”是指训练集中的某些训练点的输入 $\boldsymbol{x}_i$，事实上，最优化问题式(7.22)~式(7.24)的解 α^* 的每一个分量 α_i^* 都与一个训练点相对应，算法 7.1 所构造的分类超平面，仅依赖于那些相应于 α_i^* 不为零的训练点 $(\boldsymbol{x}_i,y_i)$，而与相应于 α_i^* 为零的那些训练点无关。所以人们特别关心相应于 α_i^* 不为零的训练点 $(\boldsymbol{x}_i,y_i)$，并称这些训练点的输入 $\boldsymbol{x}_i$ 为支持向量。只有支持向量对最终求得的分类超平面的法线方向 $\boldsymbol{w}^*$ 有影响，而与非支持向量无关。故称这种方法为支持向量机。

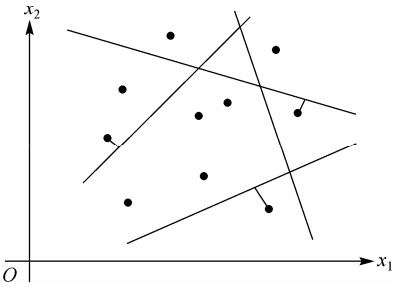


图 7.10 标准超平面

2. 近似线性可分问题

对于图7.11所示的问题，用一条直线也能大体上把训练集正确分开，这类问题称为近似线性可分问题，这时仍可以考虑使用线性分类学习机。

应用前述方法，通过求解原始问题式(7.13)~式(7.15)的对偶问题式(7.16)~式(7.18)来确定决策函数，具体算法步骤如下：

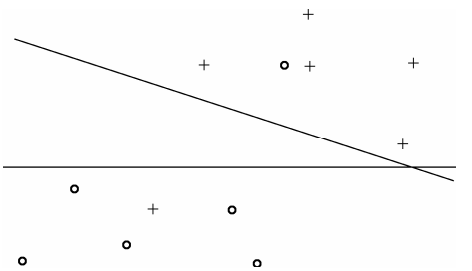


图 7.11 近似线性可分问题

【算法 7.2】 近似线性支持向量机

(1) 设已知训练集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\mathrm{L}\ ,l$。

(2) 选择适当的惩罚参数 $C>0$，构造并求解最优化问题：

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{l}\alpha_j \tag{7.23}$$

$$\text{s.t.}\quad \sum_{i=0}^{l}y_i\alpha_i=0 \tag{7.24}$$

$$\alpha_i\geqslant 0, \quad i=1,\mathrm{L}\ ,l \tag{7.25}$$

得最优解 $\alpha^*=(\alpha_1^*,\mathrm{L}\ ,\alpha_l^*)^{\mathrm{T}}$。

(3) 计算 $\boldsymbol{w}^* = \sum_{i=1}^{l} y_i \alpha_i^* \boldsymbol{x}_i$；选择 α^* 的一个分量 $0 < \alpha_j^* < C$，并据此计算 $b^* = y_i - \sum_{i=1}^{l} y_i \alpha_i^* (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$。

(4) 构造分类超平面 $(\boldsymbol{w}^* \cdot \boldsymbol{x}) + b^* = 0$，求得决策函数 $f(\boldsymbol{x}) = \mathrm{sgn}((\boldsymbol{w}^* \cdot \boldsymbol{x}) + b^*)$。

3. 线性不可分问题

对于图7.12所示的问题，如果用直线分类会产生很大的误差，这类问题称为线性不可分问题。这时就必须使用非线性分类学习机进行分类。

对于这类问题，显然不能用超平面去划分，此时可以通过一个映射，把寻找超曲面的问题转化为寻找超平面的问题。下面以图7.13所示的分类问题来说明。

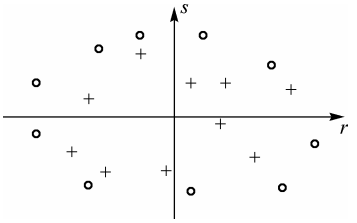


图 7.12　线性不可分问题

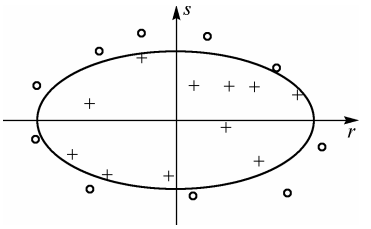


图 7.13　非线性划分

设训练集为 $\boldsymbol{T} = \{(\boldsymbol{x}_i, y_i), i = 1, \mathrm{L}\ , 20\}$，其中 $\boldsymbol{x}_i$ 是 (r, s) 平面上的点：$\boldsymbol{x}_i = (x_{i1}, x_{i2})^{\mathrm{T}}$，$y_i \in \{1, -1\}$。可以看出，比较合理的划分是 (r, s) 平面上的一个椭圆

$$w_1 r^2 + w_2 s^2 + b = 0 \tag{7.26}$$

其中 w_1, w_2 和 b 都是常数，如图 7.13 所示。

希望仍然使用线性分类方法，求出这个非线性分类椭画。考虑从 (r, s) 平面上的点 $\boldsymbol{x} = (x_1, x_2)^{\mathrm{T}}$ 到 (u_1, u_2) 平面上的点 $\boldsymbol{x} = (x_1, x_2)^{\mathrm{T}}$ 的映射 $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2, x_2^2)$：

$$\boldsymbol{\phi}: \begin{matrix} x_1 = x_1^2 \\ x_2 = x_2^2 \end{matrix} \tag{7.27}$$

它把 (r, s) 平面上的椭圆 $w_1 r^2 + w_2 s^2 + b = 0$ 映射到 (u_1, u_2) 平面上的一条直线：$w_1 u_1 + w_2 u_2 + b = 0$，如图7.14所示。

所以只要用映射式(7.27)把 (r, s) 平面上的两类训练点分别映射到 (u_1, u_2) 平面上，然后在 (u_1, u_2) 平面上使用线性学习机求出分类直线，最后把分类直线再映射回平面 (r, s)，就得到所寻找的非线性分类椭圆了。

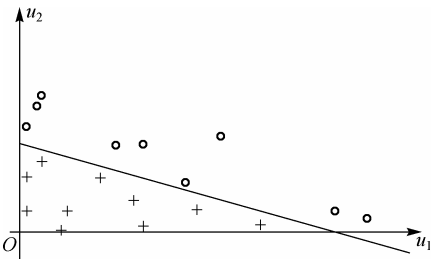


图 7.14　非线性映射

由这一简单的例子可以看出，对于线性不可分问题，引入一个非线性函数 $\boldsymbol{\Phi}$ 将输入空间映射到一个高维特征空间

$$\boldsymbol{\Phi}: \begin{matrix} X \subset R^n \to H \\ \boldsymbol{x}\ \mathrm{a}\quad \boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{x}) \end{matrix} \tag{7.28}$$

然后在特征空间中构造线性分类，此时最优化问题为

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i \tag{7.29}$$

$$\text{s.t.}\quad y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)+\xi_i\geq 1\ i=1,\mathrm{L}\ ,l \tag{7.30}$$

$$\xi_i\geq 0\ ,\quad i=1,\mathrm{L}\ ,l \tag{7.31}$$

其中 $\boldsymbol{x}_i=\boldsymbol{\Phi}(\boldsymbol{x}_i)$。类似地，引入它的对偶问题：

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_jK(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{l}\alpha_j \tag{7.32}$$

$$\text{s.t.}\quad \sum_{i=0}^{l}y_i\alpha_i=0 \tag{7.33}$$

$$0\leq\alpha_i\leq C\ ,\quad i=1,\mathrm{L}\ ,l \tag{7.34}$$

其中，

$$K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)=(\boldsymbol{\Phi}(\boldsymbol{x}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{x}_j)) \tag{7.35}$$

称为核函数，通过求解对偶问题来确定最终的决策函数，这样就得到非线性可分支持向量机(标准的支持向量机)算法。

将 2 维输入向量投影到 6 维特征空间，应用非线性 SVC 到线性非可分训练数据，产生分类图形，如图7.15($C=\infty$)所示。由于是非线性地投影到输入空间，所以间隔不再是等宽的。和图7.6相反，此时训练数据被正确分类。

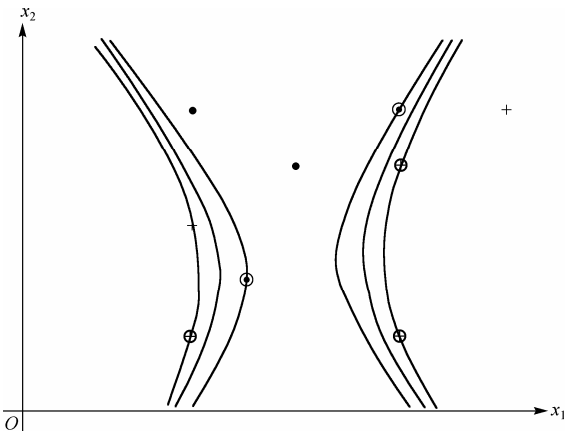


图 7.15　映射输入空间到多项式特征空间

【算法 7.3】 非线性支持向量机-标准支持向量机

(1) 已知训练集 $\boldsymbol{T}=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}i=$，$1,\mathrm{L}\ ,l$。

(2)选择核函数 K 和惩罚参数 C，构造并求解最优化问题：

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_i y_j\alpha_i\alpha_j K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{l}\alpha_j \tag{7.36}$$

$$\text{s.t.}\quad \sum_{i=0}^{l}y_i\alpha_i=0 \tag{7.37}$$

$$0\leq\alpha_i\leq C,\quad i=1,\mathrm{L}\ ,l \tag{7.38}$$

得最优解 $\alpha^*=(\alpha_1^*,\mathrm{L}\ ,\alpha_l^*)^{\mathrm{T}}$。

(3) 选择 α^* 的一个分量 $0\leq\alpha_j^*\leq C$，并据此计算 $b^*=y_j-\sum_{i=1}^{l}y_i\alpha_i^*K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)$。

(4) 构造分类超平面 $(\boldsymbol{w}^*\cdot\boldsymbol{x})+b^*=0$，由此求得决策函数

$$f(\boldsymbol{x})=\mathrm{sgn}\left(\left(\sum_{i=1}^{l}y_i\alpha_i^*K(\boldsymbol{x}_i,\boldsymbol{x})\right)+b*\right) \tag{7.39}$$

线性可分支持向量机、近似线性可分支持向量机和不可分支持向量分类机之间的关系分析如下：首先，三者都是将分类问题转化为最优化问题，分别对应三个原始最优化问题和对偶问题，由对偶问题的解来确定决策函数。

原始问题式(7.8)~式(7.9)适用于线性可分问题，原始问题式(7.13)~式(7.15)适用于近似线性可分问题，但比较两个问题会发现，如果对同一个线性可分的训练集，分别求解两个问题，一般来说可能得到不同的决策函数，但当参数 $C\to\infty$ 时，问题式(7.13)~式(7.15)就退化为问题式(7.8)~式(7.9)，这时就得到相同的决策函数。从这一意义上说，原始问题式(7.13)~式(7.15)不仅适用于求解线性不可分问题，也适用于求解线性可分问题。

对于线性不可分问题，引进从输入空间 R^n 到特征空间 H 的变换，然后在特征空间求解原始问题式(7.29)~式(7.31)。显然，如果所做的变换是线性变换，则问题式(7.29)~式(7.31)就退化为问题式(7.13)~式(7.15)，因此问题式(7.29)~式(7.31)适用的范围更广。事实上，在实际应用中所要处理的分类问题往往比较复杂，一般属于这种线性不可分问题，如果直接在原输入空间用超平面去划分效果可能不好，将输入空间映射到一个高维特征空间后，就增加了线性可分的可能性。因此，由问题式(7.29)~式(7.31)的解来确定决策函数，是最常用的也是人们主要研究的方法，即标准支持向量机。

支持向量机方法是通过先求解它的对偶问题，然后再根据对偶问题的解来确定决策函数的。由对偶问题的解得到原始问题的解来确定决策函数，也是支持向量机理论的关键因素。考查算法 7.3 中决策函数的表达式(7.39)可以发现，在给定训练集后，它仅依赖于 α^* 和由式(7.35)定义的核函数。注意到问题式(7.36)~式(7.38)的解也是由核函数确定的，因此在给定训练点后，决策函数式(7.39)仅仅依赖于核函数。即只要选定了核函数，就可以利用它求得决策函数式(7.39)。而从核函数的定义看，核函数是由所做的映射决定的，一般来讲需要知道具体的映射，然后由内积计算核函数。但由后面的讨论将会看到，由于核函数具有很好的性质，所以不需要知道具体映射可以直接选取核函数。这样，选择不同的核函数，就意味着选取了不同的映射，可以通过选择适当的核函数，使算法达到更好的效果[3-5]，这也是支持向量机的重要优点。显然，如果直接求解原始问题是不能做到这一点的。另一方面，由于映

射之后的希尔伯特空间往往维数较高，有时甚至是无穷维的，这使得原问题的求解比较困难，而对偶问题的约束条件相对比较简单，而且问题的维数只和训练点的个数有关，和空间的维数无关，因此求解对偶问题相对比较容易一些。

4．核函数的性质

核函数是在输入空间而不是在高维空间进行操作的，内积不需要在特征空间进行计算，因此解决了维数灾难问题。然而计算仍然和训练模式数量有关，提供高维空间较好的数据分布将需要很大的训练集合。核函数理论是基于再生核希尔伯特空间（Reproducing Kernel Hilbert Spaces, RKHS）理论建立的。特征空间的内积在输入空间有等价核 $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{x}'))$，提供了一定的条件支持，如果 K 是对称正定函数，它满足 Mercer 条件

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{m}^{\infty} a_m \phi_m(\boldsymbol{x}) \phi_m(\boldsymbol{x}'), \qquad a_m \geq 0 \tag{7.40}$$

$$\iint K(\boldsymbol{x}, \boldsymbol{x}') g(\boldsymbol{x}) g(\boldsymbol{x}') \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{x}' > 0, \qquad g \in L_2 \tag{7.41}$$

因而核表示了特征空间合理的内积，给出了满足 Mercer 条件的有效函数。

要计算核函数，并不需要知道具体的映射 $\boldsymbol{\Phi}$，然后由内积构造出核函数；而只需直接选取核函数即可。下面举例说明。

考虑把二维空间 $\boldsymbol{x} = (x_1, x_2)^{\mathrm{T}}$ 映射到 6 维空间 $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6)^{\mathrm{T}}$ 的变换 $\boldsymbol{\Phi}$：

$$\boldsymbol{\Phi}(\boldsymbol{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2)^{\mathrm{T}} \tag{7.42}$$

对于训练集中的任意点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$，映射之后为

$$\boldsymbol{\Phi}(\boldsymbol{x}_i) = (1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, \sqrt{2}x_{i1}x_{i2}, x_{i1}^2, x_{i2}^2)^{\mathrm{T}} \tag{7.43}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}_j) = (1, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}, \sqrt{2}x_{j1}x_{j2}, x_{j1}^2, x_{j2}^2)^{\mathrm{T}} \tag{7. 44}$$

此时，

$$\boldsymbol{\Phi}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{i2}x_{j1}x_{j2} + x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 \tag{7.45}$$

引入函数

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv ((\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + 1)^2 \tag{7.46}$$

$$\begin{aligned} K(\boldsymbol{x}_i, \boldsymbol{x}_j) &= ((\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + 1)^2 = (x_{i1}x_{j1} + x_{i2}x_{j2} + 1)^2 \\ &= x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 1 + 2x_{i1}x_{j1}x_{i2}x_{j2} + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} \end{aligned} \tag{7.47}$$

比较式(7.45)和式(7.47)，有

$$(\boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j)) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv ((\boldsymbol{x}_i \cdot \boldsymbol{x}_j) + 1)^2 \tag{7.48}$$

这一等式说明，6 维空间中的内积 $(\boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j))$，可以通过计算核函数 $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 在 2 维空间中的内积 $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 得到。

计算核函数 $K(\boldsymbol{x}, \boldsymbol{x}')$，不需要知道具体的映射，许多常见的函数都可以作为核函数。但要求核函数是正定核[6]，即

【定义 7.1】 正定核：设 X 是 R^n 中的一个子集。称定义在 $X\times X$ 上的函数 $K(\boldsymbol{x}, \boldsymbol{x}')$ 是正定核，如果存在着从 X 到某一个特征空间 H 的映射，

$$\begin{array}{l}\Phi: X \subset R^n \to H \\ \boldsymbol{x} \text{ a } \quad \boldsymbol{x}=\phi(\boldsymbol{x})\end{array} \tag{7.49}$$

使

$$K(\boldsymbol{x}, \boldsymbol{x}')=(\Phi(\boldsymbol{x})\cdot\Phi(\boldsymbol{x}')) \tag{7.50}$$

其中 $(\cdot)$ 表示 H 中的内积。

但实际上定义 7.1 的条件很难判断，因此可以用下面的等价定义：

【定义7.2】 正定核的等价定义：设 X 是 R^n 的子集，称定义在 $X\times X$ 上的对称函数 $K(\boldsymbol{x}, \boldsymbol{x}')$ 为一个正定核，如果对任意的 $\boldsymbol{x}_1, \mathrm{L}\ , \boldsymbol{x}_l \in X$，$K(\cdot,\cdot)$ 相对于 $\boldsymbol{x}_1, \mathrm{L}\ , \boldsymbol{x}_l$ 的 Gram 矩阵都是半正定的。$K(\cdot,\cdot)$ 相对于 $\boldsymbol{x}_1, \mathrm{L}\ , \boldsymbol{x}_l$ 的 Gram 矩阵是指第 i 行第 j 列元素为 $K_{ij}=K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 的 $l\times l$ 阶矩阵。

常用的核函数如下[14]。

(1) 多项式核函数

对于非线性模型多项式映射是最普遍的方法，其表达式为

$$K(\boldsymbol{x}, \boldsymbol{x}')=(\boldsymbol{x}\cdot\boldsymbol{x}')^d \tag{7.51}$$

$$K(\boldsymbol{x}, \boldsymbol{x}')=((\boldsymbol{x}\cdot\boldsymbol{x}')+1)^d \tag{7.52}$$

通常式(7.52)可以避免黑塞行列式变为 0 的问题。

(2) 高斯径向基核函数

径向基函数得到了人们的普遍关注，应用最广泛的是高斯径向基函数：

$$K(\boldsymbol{x}, \boldsymbol{x}')=\mathrm{e}^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma^2}} \tag{7.53}$$

(3) 指数径向基函数

指数径向基函数是另一类径向基函数：

$$K(\boldsymbol{x}, \boldsymbol{x}')=\mathrm{e}^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|}{2\sigma^2}} \tag{7.54}$$

当函数不连续时，指数径向基函数对于分段线性解是非常有效的。

(4) 样条核函数

由于样条核函数具有很大的灵活性，所以人们常常使用样条核函数进行建模，在 $\boldsymbol{t}_s$ 点具有 N 个节点的有限 k 阶样条核函数

$$K(\boldsymbol{x}, \boldsymbol{x}')=\sum_{r=0}^{k}\boldsymbol{x}^r\boldsymbol{x}''^r+\sum_{s=1}^{N}(\boldsymbol{x}-\boldsymbol{t}_s)_+^k(\boldsymbol{x}'-\boldsymbol{t}_s)_+^k \tag{7.55}$$

在区间[0, 1]上的无穷样条核函数定义为

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{r=0}^{k} \boldsymbol{x}^r \boldsymbol{x}''^r + \int_0^1 (\boldsymbol{x} - \boldsymbol{t}_s)_+^k (\boldsymbol{x}' - \boldsymbol{t}_s)_+^k \mathrm{d}\boldsymbol{t} \tag{7.56}$$

在 $k=1$ 的情况下 (S_1^∞)，核函数为

$$K(\boldsymbol{x}, \boldsymbol{x}') = 1 + (\boldsymbol{x}, \boldsymbol{x}') + \frac{1}{2}(\boldsymbol{x}, \boldsymbol{x}')\min(\boldsymbol{x}, \boldsymbol{x}') - \frac{1}{6}\min(\boldsymbol{x}, \boldsymbol{x}')^3 \tag{7.57}$$

其解为分段立方函数。

(5) B-样条核函数

B-样条核函数是另一种广泛使用的样条核函数，在区间[－1, 1]上定义 B-样条核函数

$$K(\boldsymbol{x}, \boldsymbol{x}') = B_{2p+1}(\boldsymbol{x}, \boldsymbol{x}') \tag{7.58}$$

其中 B_{2p+1} 是 $2p+1$ 阶 B-样条函数。

(6) 多层感知器核函数(Sigmoid 核)

具有一个隐层的多层感知器也是一种有效的核函数：

$$K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\kappa(\boldsymbol{x}, \boldsymbol{x}') + \upsilon) \tag{7.59}$$

其中 $\kappa > 0$，$\upsilon < 0$，支持向量 SV 和第一层对应，拉格朗日乘子对应权值。

(7) 傅里叶序列

傅里叶序列可以视为 $2N+1$ 特征空间的扩展，其核函数定义在区间 $[-\pi/2, \pi/2]$，

$$K(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sin\left(N + \frac{1}{2}\right)(\boldsymbol{x} - \boldsymbol{x}')}{\sin\left(\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')\right)} \tag{7.60}$$

但是这种核函数的适应性很差。

(8) 加法核函数

更复杂的核函数可以通过核函数求和完成，因为两个正定函数的和也是正定的：

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_i K_i(\boldsymbol{x}, \boldsymbol{x}') \tag{7.61}$$

(9) 张量积

多维核函数可以通过核函数的张量积获得：

$$K(\boldsymbol{x}, \boldsymbol{x}') = \prod_i K_i(\boldsymbol{x}_i, \boldsymbol{x}_i') \tag{7.62}$$

式(7.62)在构造多维样条核函数时特别有用，它可以通过一维核函数的乘积获得。

在支持向量机方法中，选择适当的核函数是一个重要的因素。除了这里介绍的几种常用的核函数外，还可以根据具体问题构造相应的核函数[7-10]。此外，还可以从数据中学习，构造核函数[11, 12]。

7.2.2 统计学习理论

前面从基于最大“间隔”思想推导出标准支持向量机方法。支持向量机方法之所以能够得到广泛的应用，是由于其深刻的理论基础——统计学习理论。本节简要介绍统计学习理论[13]。

建模的目的是为了在假设空间选择一个模型，选择的模型最接近目标空间的函数，如图7.16所示。选择模型时引起错误的原因有两点[14]：

(1)近似错误(Approximation Error)：是由假设空间小于目标空间造成的，因此函数可能位于假设空间之外。较差的模型空间选择将导致很大的近似误差，而且模型将被误匹配。

(2)估计错误(Estimation Error)：在假设空间的学习过程中选择的模型不是最优而导致的误差。

这些错误就是通常所说的误差。

根据训练集求出决策函数 $f(\boldsymbol{x})$ 是分类问题，决策函数也称为假设。在求得一个假设 $f(\boldsymbol{x})$ 后，对于新的输入 $\boldsymbol{x}$，按 $y=f(\boldsymbol{x})$ 推断 $\boldsymbol{x}$ 相应的输出 y，称为推广。对于给定的一个假设 $f(\boldsymbol{x})$，用推广能力描述其“推广”的优劣程度。

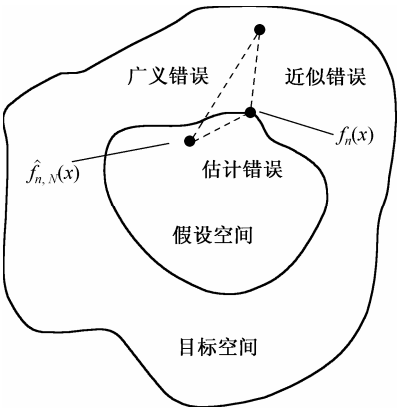


图 7.16　建模误差

【定义 7.3】 损失函数：设 $X\subset R^n$，$y=\{-1,1\}$。引进 3 元组 $(\boldsymbol{x},y,f(\boldsymbol{x}))\in X\times Y\times Y$，其中 $\boldsymbol{x}$ 是模式，y 是观察值，$f(\boldsymbol{x})$ 是假设值(或称预测值)。若映射 $c:X\times Y\times Y\to(0,\infty)$ 对任意的 $\boldsymbol{x}\in X$，$y\in Y$，都有 $c(\boldsymbol{x},y,y)=0$，则称 c 是一个损失函数。

损失函数 $c(\boldsymbol{x},y,y)$ 是当 $f(\boldsymbol{x})=y$ 时 $c(\boldsymbol{x},y,f(\boldsymbol{x}))=0$ 的函数。其含义是，当预测准确无误时，损失值为零(无损失发生)。实际上人们常常要求当预测有误差时，或者至少当误差达到一定程度时，其损失值不为零。

【定义7.4】 期望风险：设 $P(\boldsymbol{x},y)$ 为 $X\times Y$ 上的概率分布，c 为给定的损失函数，$f(\boldsymbol{x})$ 是一个假设(决策函数)，

$$f:X(X\subset R^n)\to y=\{-1,1\} \tag{7.63}$$

$f(\boldsymbol{x})$ 的期望风险是指损失函数关于概率分布 $P(\boldsymbol{x},y)$ 的积分：

$$R(f)\equiv E\left[c(\boldsymbol{x},y,f(\boldsymbol{x}))\right]=\int_{X\times Y}c(\boldsymbol{x},y,f(\boldsymbol{x}))\mathrm{d}P(\boldsymbol{x},y) \tag{7.64}$$

从期望风险的定义中可以看出，决策函数 $f(\boldsymbol{x})$ 的期望风险依赖于概率分布和损失函数，概率分布 $P(\cdot,\cdot)$ 是客观存在的，而损失函数 $c(\cdot,\cdot,\cdot)$ 是根据具体问题选定的。

分类问题就是要根据给定的损失函数，寻找使其期望风险最小的决策函数。

1．经验风险最小化

因为仅已知训练集 T，所以只能计算出 $f(\boldsymbol{x})$ 在这些样本点上的偏差，这就导致了“经验风险”的概念和经验风险最小化归纳原则[15]。在一定的条件下，可以用评价训练集上的经验风险大小的方法来评价 $f(\boldsymbol{x})$。

【定义 7.5】 经验风险：给定训练集 $T=\{(\boldsymbol{x}_1,y_1),\mathrm{L},(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\mathrm{L},l$；且给定损失函数 c。$f(\boldsymbol{x})$ 对于它们的经验风险是指

$$R_{\text{emp}}(f)=\frac{1}{l}\sum_{i=1}^{l}c(\boldsymbol{x}_i,y_i,f(\boldsymbol{x}_i)) \tag{7.65}$$

【定义 7.6】 经验风险最小化归纳原则：设任意给定训练集 T，并适当选定由假设构成的集合 $F\subset\{f:X(X\subset R^n)\to Y=\{-1,1\}\}$。经验风险最小化归纳原则就是在 F 中选择使其经验风险 $R_{\text{emp}}(f)$ 最小的假设 f。

经验风险最小化原则强调使经验风险 $R_{\text{emp}}(f)$ 达到最小。但是如果只考虑选择 f 使得 $R_{\text{emp}}(f)$ 达到最小而忽视对 F 的选择是不可行的。若选择 F 为包含所有的 $f:X\to Y$ 的假设集 $F=\{f:X\to Y\}$，则假设

$$f(\boldsymbol{x})=\begin{cases}y_i, & \boldsymbol{x}=\boldsymbol{x}_i,i=1,\text{L},m\\ 1, & \text{其他}\end{cases} \tag{7.66}$$

也在 F 中。显然，这个假设的经验风险满足

$$R_{\text{emp}}(f)=0 \tag{7.67}$$

这导致最终可能选择由式(7.66)定义的假设 f，因为它是 F 中使经验风险最小的假设。然而把这个 f 作为决策函数是不妥当的。如果使用经验风险最小化原则，应该适当缩小假设集 F，至少它不应该包含类似式(7.66)的假设，总之要对 F 进行限制。

2．VC 维

在学习算法中需要选择适当的假设集 F。实际上，这里的关键因素是假设集 F 的大小，或 F 的丰富程度，或者说 F 的“表达能力”。由 Vapnik 和 Chervonenkis 提出的 VC 维[6, 16]是对这种“表达能力”的一种描述。F 的 VC 维概念是建立在点集被 F“打散”的基础上的。图 7.17(a)说明了在平面中线性指示函数集能打散三个点，因此其 VC 维等于 3；图7.17(b)说明不能打散四个点的情况，因为不能用直线将向量 $\boldsymbol{x}_2,\boldsymbol{x}_4$ 与向量 $\boldsymbol{x}_1,\boldsymbol{x}_3$ 分开[1]。

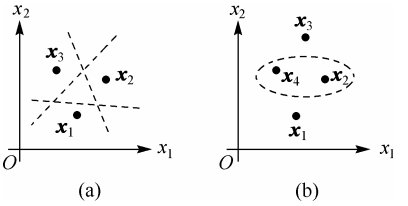


图 7.17 VC 维的举例说明

【定义 7.7】 $N(F,N_m)$：设 F 是一个假设集，即由在 $X\subset R^n$ 上取值为 – 1 或 1 的若干函数组的集合。记 $Z_m=\{\boldsymbol{x}_1,\text{L},\boldsymbol{x}_m\}$ 为 X 中的 m 个点组成的集合。考虑当 f 取遍 F 中所有可能的假设时产生的 m 维向量 $(f(\boldsymbol{x}_1),\text{L},f(\boldsymbol{x}_m))$。定义 $N(F,Z_m)$ 为上述 m 维向量中不同的向量个数。

【定义 7.8】 Z_m 被 F 打散：设 F 是一个假设集，$Z_m=\{\boldsymbol{x}_1,\text{L},\boldsymbol{x}_m\}$ 为 X 中 m 个点组成的集合。如果 $N(F,Z_m)=2^m$，则称 Z_m 被 F 打散，或 F 打散 Z_m。

【例 7.3】 设 X 为二维空间 $X=\{x_1,x_2\}$，令 F 是 X 上的线性指示函数的集合，即

$$F=\{f(\boldsymbol{x},\alpha)=\text{sgn}(\alpha_2x_2+\alpha_1x_1+\alpha_0)\} \tag{7.68}$$

令 $Z_3=(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)\subset X$，且 $\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3$ 不共线。现在说明 Z_3 被 F 打散。对 $\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3$ 分别标

上“+”标号或“o”标号，共有 $2^3 = 8$ 种标号方式。对每一种标号方式，都存在 $f \in F$，使得“+”标号和“o”标号被 $f = 0$ 分开，如图7.18所示。图中“+”表示“+”标号的点，“o”表示“o”标号的点。这表明 $N(F, Z_3) = 2^3$，即 Z_3 被 F 打散。

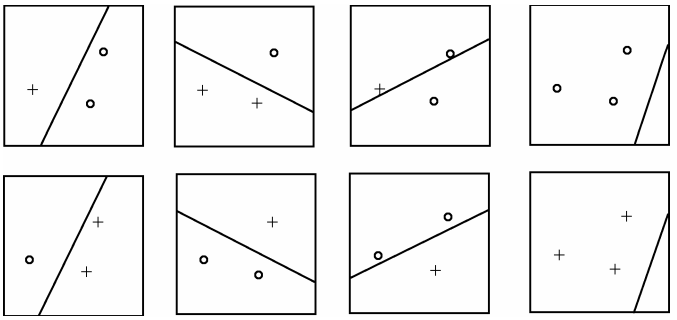


图 7.18　Z_3 被 F 打散

【定义 7.9】 增长函数：增长函数 $N(F, m)$ 定义为

$$N(F, m) = \max\{N(F, m) : Z_m \subset X\} \tag{7.69}$$

其中 $Z_m = \{\boldsymbol{x}_1, \mathrm{L}\ , \boldsymbol{x}_m\}$ 是 X 中 m 个点组成的集合，$\max\{\cdot\}$是对这些点跑遍 X 而言的。

假设集 F 能打散点的个数越多，表明 F 的“表达能力”越强。F 的 VC 维就是使 $N(F, Z_m) = 2^m$ 成立的最大 m 值。

【定义 7.10】 VC 维：设假设集 F 是一个由 X 上取值为 1 或–1 的函数值组成的集合。定义 F 的 VC 维为

$$VC_{\dim}(F) = \max\{m : N(F, m) = 2^m\} \tag{7.70}$$

当 $\left\{m : N(F, Z_m) = 2^m\right\}$ 是一个无限集合时，定义 $VC_{\dim}(F) = \infty$。

由定义 7.10 可见 F 的 VC 维就是它能打散 X 中点的最大个数。换句话说，若存在 m 个点组成的集合 Z_m 能被 F 打散，且任意 $m + 1$ 个点的集合 Z_{m+1} 不能被 F 打散，则 F 的 VC 维就是 m；若任给正整数 m，都存在 m 个点组成的集合 Z_m 能被 F 打散，则 F 的 VC 维就是∞，例如函数集合

$$F = \{f(x, \alpha) = \mathrm{sgn}(\sin(\alpha x)), \quad \alpha \in R\} \tag{7.71}$$

3．结构风险最小化原则

根据 VC 维的理论，得到期望风险在概率意义下的一个上界，如定理 7.1[15]所述。

【定理 7.1】 记 h 为 F 的 VC 维。若

$$l > h \tag{7.72}$$

$$h\left(\ln\frac{2l}{h} + 1\right) + \ln\frac{4}{\delta} \geq \frac{1}{4} \tag{7.73}$$

则对于任意的概率分布 $P(\boldsymbol{x}, y)$ 和任意的 $\delta \in (0, 1)$，F 中的任意假设 f 都可使得下列不等式至少以 $1 - \delta$ 的概率成立：

$$R(f)\leq R_{\text{emp}}(f)+\sqrt{\frac{8}{l}\left(h(\ln\frac{2l}{h}+1)+\ln\frac{4}{\delta}\right.} \tag{7.74}$$

特别地，当 $R_{\text{emp}}(f)=0$ 时，有

$$R(f)\leq \sqrt{\frac{8}{l}\left(h(\ln\frac{2l}{h}+1)+\ln\frac{4}{\delta}\right.} \tag{7.75}$$

定理 7.1 是一个定量的估计，但实际计算出的式(7.74)的右端值往往较大，常常没有多少实用价值。因此这个定理的意义主要是在定性分析上的应用：称式(7.74)中右端的第二项 $\sqrt{\frac{8}{l}\left(h(\ln\frac{2l}{h}+1)+\ln\frac{4}{\delta}\right.}$ 为置信区间，而此式右端的两项之和称为结构风险，它是期望风险 $R(f)$ 的一个上界。由式(7.74)看出，这个上界是经验风险和置信区间的和。经验风险依赖于 f 的选择，而置信区间则是 f 的 VC 维 h 的增函数。当集合 F 比较大时，可以选到适当的 f，使得 $R_{\text{emp}}(f)$ 比较小。而此时由于 F 的 VC 维比较大，也会使置信区间比较大。反之，当缩小集合 F 时，F 的 VC 维会降低，置信区间变小，而 $R_{\text{emp}}(f)$ 有可能增大。所以两者有互相矛盾的倾向。为了选出兼顾二者的假设，引入结构风险最小化原则[15]。

【定义 7.11】 结构风险最小化原则：结构风险最小化原则是寻找一个假设 f，使得式(7.74)右端所示的结构风险达到最小值。例如，适当选择一系列嵌套的假设集

$$\text{L}\ F_{n-1}\subset F_n\subset F_{n+1}\text{L} \tag{7.76}$$

在每个 F_n 中找出使经验风险最小的假设 f_n，得到一系列假设

$$\text{L}\ f_{n-1},f_n,f_{n+1},\text{L} \tag{7.77}$$

考查与 f_n 相应的结构风险随 n 的变化情况，可以发现：

(1)置信区间随着 n 的增加而增大，因为 F_n 的 VC 维是递增的：

$$\text{L}\ h_{n-1}\leq h_n\leq h_{n+1}\leq \text{L} \tag{7.78}$$

(2)经验风险随着 n 的增加而减小：

$$R_{\text{emp}}(f_{n-1})\geq R_{\text{emp}}(f_n)\geq R_{\text{emp}}(f_{n+1})\text{L} \tag{7.79}$$

因为 F_n 是嵌套的，结构风险最小化原则就是要选择适当的 n^*，使置信区间与经验风险之和达到最小，由此得到相应的假设 f_n。

图 7.19 示意性地描述了结构风险最小化原则。其中 $S_{n-1}\subset S_n\subset S_{n+1}$

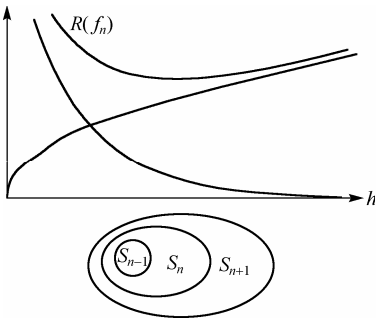


图 7.19 结构风险最小化原则

4．基于间隔的推广估计

定理7.1导出了控制期望风险的结构风险表达式(7.74)的右端。其中置信区间与假设集 F 的 VC 维有关。而当 VC 维很大甚至是无穷时，定理 7.1 就会失去意义。因为用 VC 维来描述假设集的丰富程度和表达能力，而 VC 维强烈地依赖于空间的维数。为了在维数很高的空间中可行，引入间隔的概念[6]估计学习算法的推广能力。

【定义 7.12】 间隔和几何间隔：设 G 是由在 X 上取实值的若干函数组成的集合。样本点 $(\boldsymbol{x}_i, y_i)$ 关于 $g \in G$ 的间隔定义为

$$\gamma_i = y_i g(\boldsymbol{x}_i) \tag{7.80}$$

训练集 $T = \{(\boldsymbol{x}_1, y_1), \mathrm{L}\ , (\boldsymbol{x}_l, y_l)\}$ 关于 g 的间隔定义为

$$\gamma(g) = \min\{\gamma_i, i = 1, \mathrm{L}\ , l\} \tag{7.81}$$

当 $G = \{g(\boldsymbol{x}) = (\boldsymbol{w} \cdot \boldsymbol{x}) + b\}$ 时，样本点 $(\boldsymbol{x}_i, y_i)$ 关于 $g \in G$ 的几何间隔定义为

$$\gamma_i = \frac{y_i g(\boldsymbol{x}_i)}{\|\boldsymbol{w}\|} = \frac{y_i(< \boldsymbol{w} \cdot \boldsymbol{x}_i > + b)}{\|\boldsymbol{w}\|} \tag{7.82}$$

训练集 $T = \{(\boldsymbol{x}_1, y_1), \mathrm{L}\ , (\boldsymbol{x}_l, y_l)\}$ 关于 g 的几何间隔定义为

$$\gamma_T(g) = \frac{1}{\|\boldsymbol{w}\|}\gamma(g) = \min_i \frac{y_i(< \boldsymbol{w} \cdot \boldsymbol{x}_i > + b)}{\|\boldsymbol{w}\|} \tag{7.83}$$

注意到当 $\gamma_i > 0$ 时，说明用函数 g 进行分类时样本点 $(\boldsymbol{x}_i, y_i)$ 被正确分类。当 g 为线性函数 $g(\boldsymbol{x}) = \langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle + b$ 且 $\gamma_T(g) > 0$ 时，$\gamma_T(g)$ 就是 $\boldsymbol{x}_1, \mathrm{L}\ , \boldsymbol{x}_l$ 到直线 $g = 0$ 的距离最小值，这也是称 $\gamma_T(g)$ 为几何间隔的原因。

根据几何间隔的定义，前面提到规范划分直线的几何间隔即为 $\frac{2}{\|\boldsymbol{w}\|}$，因此对于规范的划分直线，它的几何间隔大小可以用 $\|\boldsymbol{w}\|$ 来衡量。定理7.2[17]就给出了基于间隔的推广能力估计，这一定理为标准的支持向量机提供了理论基础。

【定理 7.2】 设概率分布 P 确定的在 X 上的分布 P_x 满足

$$P_x\{\boldsymbol{x} : \|\boldsymbol{x}\| \leqslant \zeta\} = 1 \tag{7.84}$$

若考虑线性决策函数 $f = \mathrm{sgn}(\langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle + b)$，则对任给 $\delta \in (0,1)$，相对于 0–1 损失函数，期望风险 $R(f)$至少以 $1-\delta$ 的概率满足

$$R(f) \leqslant \frac{2}{l}\left(\left[d_{\mathrm{eff}} \mathrm{lb}\left(\frac{8el}{d_{\mathrm{eff}}}\right)\mathrm{lb}(32l)\right] + \mathrm{lb}\left(\frac{(16 + \mathrm{lb}l)}{8}\right)\right) \tag{7.85}$$

其中

$$d_{\mathrm{eff}} = 65\left[\sqrt{2}\zeta(\|\boldsymbol{w}\|^2 + 1) + 3\sqrt{\sum_{(\boldsymbol{x}_i, y_i) \in T}(\max\{0, 1 - y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b)\})^2}\right] \tag{7.86}$$

这里 d_{eff}必须满足 $d_{\mathrm{eff}} \leqslant 2l$。

这个定理给出了期望风险 $R(f)$至少以 $1-\delta$ 概率成立的一个上界，即式(7.87)的右端。显然这个上界是 d_{eff}的单调增函数，因此在选择决策函数 $f = \mathrm{sgn}(\langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle + b)$ 时，应使 d_{eff}尽可能小。求解原始问题式(7.29)~式(7.31)就是寻找使 d_{eff}达到最小的$(\boldsymbol{w}, b)$。为此考虑原始问题式(7.29)~式(7.31)的一个变形：

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i^2 \tag{7.87}$$

$$\text{s.t.}\qquad y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle+b)\geq 1-\xi_i,\qquad i=1,\mathrm{L}\ ,l \tag{7.88}$$

$$\xi_i\geq 0,\qquad i=1,\mathrm{L}\ ,l \tag{7.89}$$

其中 $C>0$ 是一个惩罚参数。这个问题虽然与原始问题不完全一致，但其本质是相同的，只不过对应了不同的损失函数。因此只需考查问题式(7.87)~式(7.89)和 d_{eff} 的关系。事实上，式(7.86)给出的 d_{eff} 由两项组成。它们分别对应目标函数(7.87)中的第一项和第二项中的 $\sum_{i=1}^{l}\xi_i^2$。目标函数中的参数 C 则是调节以上两个因素 $\|\boldsymbol{w}\|^2$ 和 $\sum_{i=1}^{l}\xi_i^2$ 的因子。容易看出，这里的 $\sum_{i=1}^{l}\xi_i^2$ 体现了经验风险，而 $\|\boldsymbol{w}\|^2$ 则体现了表达能力。所以因子 C 实质上是对经验风险和表达能力如何匹配的一个裁决。

7.2.3 优化理论

支持向量机涉及两个凸规划问题——原始问题和对偶问题，而且由两个问题解之间的关系建立算法。因此优化理论也是支持向量机的重要理论基础。在这一部分给出关于凸规划问题的优化理论如凸规划问题解的充分必要条件及其对偶理论[18-20]。

1．KKT 条件

考虑凸约束问题：

$$\min f(\boldsymbol{x}),\quad \boldsymbol{x}\in R^n \tag{7.90}$$

$$\text{s.t.}\qquad c_i(\boldsymbol{x})\leq 0,\qquad i=1,\mathrm{L}\ ,p \tag{7.91}$$

$$c_i(\boldsymbol{x})=0,\qquad i=p+1,\mathrm{L}\ ,p+q \tag{7.92}$$

其中目标函数 $f(\boldsymbol{x})$和约束函数 $c_i(\boldsymbol{x})$，$i=1,\mathrm{L}\ ,p$ 都是凸函数，而 $c_i(\boldsymbol{x})$，$i=p+1,\mathrm{L}\ ,p+q$ 都是线性函数。

【定理 7.3】（凸约束问题的解）考虑凸约束问题式(7.90)~式(7.92)。设 D 是问题的可行域

$$D=\{\boldsymbol{x}\mid c_i(\boldsymbol{x})\leq 0,\ i=1,\mathrm{L}\ ,p;\ c_i(\boldsymbol{x})=0,\ i=p+1,\mathrm{L}\ ,p+q;\ \boldsymbol{x}\in R^n\} \tag{7.93}$$

则

(1)若问题有局部解 $\boldsymbol{x}^*$，则 $\boldsymbol{x}^*$是问题的整体解。

(2)问题的整体解组成的集合是凸集。

(3)若问题有局部解 $\boldsymbol{x}^*$，$f(\boldsymbol{x})$是 D 上的严格凸函数，则 $\boldsymbol{x}^*$是问题的唯一整体解。

【定义 7.13】 约束规格。考虑一般约束问题式(7.92)~式(7.94)的可行域

$$D=\{\boldsymbol{x}\mid c_i(\boldsymbol{x})\leq 0,\ i=1,\mathrm{L}\ ,p\ ;\ c_i(\boldsymbol{x})=0,\ i=p+1,\mathrm{L}\ ,p+q\ ;\ \boldsymbol{x}\in R^n\} \tag{7.94}$$

其中 p 个约束函数 $c_1(\boldsymbol{x}),\mathrm{L}\ , c_p(\boldsymbol{x})$ 是可微函数。引进下列两种对约束的限制性条件(约束规格)：

(1)线性条件：p 个约束函数 $c_1(\boldsymbol{x}),\mathrm{L}\ , c_p(\boldsymbol{x})$ 都是线性函数。

(2)梯度线性无关条件：梯度向量集

$$\{\nabla c_i(\overline{\boldsymbol{x}})\,|\,i\in\overline{A}\} \tag{7.95}$$

线性无关。其中 $\overline{A}$ 为 $\overline{\boldsymbol{x}}$ 处的有效集。

【定理 7.4】（凸约束问题解的必要条件）考虑凸约束问题式(7.90)~式(7.92)，其中 $f:R^n\to R$ 和 $c_i:R^n\to R\big(i=1,\mathrm{L}\ , p\big)$，都是可微凸函数，且定义 7.13 中的某一个约束规格成立，若 $\boldsymbol{x}$ 是该问题的解，则存在着 $\overline{\alpha}=(\overline{\alpha}_1,\mathrm{L}\ ,\overline{\alpha}_p)\in R^p$，$\overline{\beta}=(\overline{\beta}_{p+1},\mathrm{L}\ ,\overline{\beta}_{p+q})\in R^q$，使得 KKT 条件成立，即

$$\nabla_x L(\overline{\boldsymbol{x}},\overline{\alpha},\overline{\beta})=\nabla f(\overline{\boldsymbol{x}})+\sum_{i=1}^{p}\overline{\alpha}_i\nabla c_i(\overline{\boldsymbol{x}})+\sum_{i=p+1}^{p+q}\overline{\beta}_i\nabla c_i(\overline{\boldsymbol{x}})=0 \tag{7.96}$$

$$c_i(\overline{\boldsymbol{x}})\leq 0,\qquad i=1,\mathrm{L}\ , p \tag{7.97}$$

$$c_i(\overline{\boldsymbol{x}})=0,\qquad i=p+1,\mathrm{L}\ , p+q \tag{7.98}$$

$$\overline{\alpha}_i\geq 0,\qquad i=1,\mathrm{L}\ , p \tag{7.99}$$

$$\overline{\alpha}_i c_i(\overline{\boldsymbol{x}})=0,\qquad i=1,\mathrm{L}\ , p \tag{7.100}$$

【定理 7.5】（凸约束问题解的充分条件）考虑凸约束问题式(7.90)~式(7.92)，其中 $f:R^n\to R$ 和 $c_i:R^n\to R(i=1,\mathrm{L}\ , p)$，都是可微凸函数，若 $\overline{\boldsymbol{x}}\in R^n$ 满足 KKT 条件，即存在着 $\overline{\alpha}=(\overline{\alpha}_1,\mathrm{L}\ ,\overline{\alpha}_p)\in R^p,\overline{\beta}=\overline{\beta}_{p+1},\mathrm{L}\ ,\overline{\beta}_{p+q})\in R^q$，使得(7.96)~式(7.100)都成立，则 $\overline{\boldsymbol{x}}$ 是问题式(7.90)~式(7.92)的解。

3．Wolfe 对偶

【定义 7.14】 Wolfe 对偶问题

$$\max_{\alpha,\beta,x} L(\boldsymbol{x},\alpha,\beta) \tag{7.101}$$

$$\text{s.t.}\quad \nabla_x L(\boldsymbol{x},\alpha,\beta)=0 \tag{7.102}$$

$$\alpha\geqslant 0 \tag{7.103}$$

为凸最优化问题式(7.90)~式(7.92)的 Wolfe 对偶。其中 $L(\boldsymbol{x},\alpha)$ 为拉格朗日函数，即

$$L(\boldsymbol{x},\alpha)=f(\boldsymbol{x})+\sum_{i=1}^{p}\alpha_i c_i(\boldsymbol{x})+\sum_{i=p+1}^{p+q}\beta_i c_i(\boldsymbol{x}) \tag{7.104}$$

【定理 7.6】（凸约束问题的强对偶定理）考虑凸约束问题式(7.90)~式(7.92)，其中 $f:R^n\to R$ 和 $c_i:R^n\to R(i=1,\mathrm{L}\ , p)$ 都是可微凸函数，$c_i(x)$，$i=p+1,\mathrm{L}\ , p+q$ 是线性函数，且定义 7.13 中的某一个约束规格成立。则

(1)若原始问题式(7.90)~式(7.92)有解，则它的对偶问题也有解。

(2)若原始问题式(7.90)~式(7.92)和对偶问题分别有可行解 $\bar{\boldsymbol{x}}$ 和 $(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}})$，则这两个可行解分别为原始问题和对偶问题的最优解的充要条件是，它们相应的原始问题和对偶问题的目标函数值相等。

【定理 7.7】（凸约束问题的 Wolfe 对偶定理）考虑凸约束问题式(7.90)~式(7.92)，其中 $f: R^n \to R$ 和 $c_i: R^n \to R(i=1,\mathrm{L}\ , p)$，都是可微凸函数，$c_i(x)$，$i=p+1,\mathrm{L}\ , p+q$ 是线性函数，且定义 7.13 中的某一个约束规格成立。则

(1)若原始问题式(7.90)~式(7.92)有解，则它的 Wolfe 对偶问题也有解。

(2)若原始问题式(7.90)~式(7.92)和 Wolfe 对偶问题分别有可行解 $\bar{\boldsymbol{x}}$ 和 $(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}})$，则这两个可行解分别为原始问题和对偶问题最优解的充要条件是，它们相应的原始问题和对偶问题的目标函数值相等。

7.3 常用的几种支持向量机

7.3.1 *C*-支持向量分类机

考虑分类问题。设给定训练集

$$T=\{(\boldsymbol{x}_1, y_1),\mathrm{L}\ ,(\boldsymbol{x}_l, y_l)\}\in(X,Y)^l \tag{7.105}$$

其中 $\boldsymbol{x}_i \in X=R^n$，$y_i \in Y \in \{1,-1\}$，$i=1,\mathrm{L}\ ,l$；分类问题是要寻找函数 $g(\boldsymbol{x})$，以便用决策函数

$$f(\boldsymbol{x})=\mathrm{sgn}(g(\boldsymbol{x})) \tag{7.106}$$

来推断任一模式 $\boldsymbol{x}$ 相对应的 y 值。*C*-支持向量机是支持向量机理论中最基本的方法，为了叙述方便，将算法重写如下：

【算法 7.4】（*C*-SVC）

(1)设已知训练集 $T=\{(\boldsymbol{x}_1, y_1),\mathrm{L}\ ,(\boldsymbol{x}_l, y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i \in X=R^n$，$y_i \in Y \in \{1,-1\}$，$i=1,\mathrm{L}\ ,l$。

(2)选择核函数 $K(\boldsymbol{x},\ \boldsymbol{x}')$ 和惩罚参数 C，构造并求解最优化问题

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_i y_j \alpha_i \alpha_j K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{j=1}^{l}\alpha_j \tag{7.107}$$

$$\text{s.t.}\quad \sum_{i=0}^{l}y_i\alpha_i=0 \tag{7.108}$$

$$0\leqslant \alpha_i \leqslant C,\qquad i=1,\mathrm{L}\ ,l \tag{7.109}$$

得最优解 $\alpha^*=(\alpha_1^*,\mathrm{L}\ ,\alpha_l^*)^{\mathrm{T}}$。

(3)选择 α^* 的一个分量 $0<\alpha_j^*<C$，并据此计算 $b^*=y_j-\sum_{i=1}^{l}y_i\alpha_i^* K(\boldsymbol{x}_i,\boldsymbol{x}_j)$。

(4)求得决策函数

$$f(\boldsymbol{x})=\mathrm{sgn}\left(\sum_{i=1}^{l}y_i\alpha_i^* K(\boldsymbol{x}_i,\boldsymbol{x})+b^*\right) \tag{7.110}$$

C-SVC 算法建立在统计学习理论的基础上，由前面的讨论可知，基于统计学习理论的定理 7.2，分类问题的决策函数

$$f(\boldsymbol{x})=\operatorname{sgn}\left(\langle \boldsymbol{w}\cdot\boldsymbol{x}\rangle+b\right)$$

中参数 $\boldsymbol{w}$ 和 b 是由如下优化问题的解来确定的：

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i \tag{7.111}$$

$$\text{s.t.}\quad y_i\left(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle+b\right)+\xi_i\geqslant 1,\ \ i=1,\mathrm{L}\ ,l \tag{7.112}$$

$$\xi_i\geqslant 0,\ i=1,\mathrm{L}\ ,l \tag{7.113}$$

其中，$C>0$。但算法 7.4 并不直接求解这一问题，而是求解它的对偶问题式(7.107)~式(7.109)，然后构造决策函数，因此需要由相应的理论来保证算法的合理性。目前关于这方面的论述都是建立在 Wolfe 对偶定理的基础之上的。下面分别详细讨论算法在求解 $\boldsymbol{w}$ 和 b 时的理论依据。

1．求解 $\boldsymbol{w}$

首先，根据 Wolfe 对偶问题的定义 7.14，推导问题式(7.111)~式(7.113)的 Wolfe 对偶问题。问题式(7.111)~式(7.113)的拉格朗日函数为

$$L(\boldsymbol{w},b,\xi,\alpha,r)=\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i(y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle+b-1+\xi_i)-\sum_{i=1}^{l}r_i\xi_i \tag{7.114}$$

其中 α_i 和 r_i 为拉格朗日乘子，满足 $\alpha_i\geqslant 0$ 和 $r_i\geqslant 0$，$i=1,\mathrm{L}\ ,l$。根据 Wolfe 对偶定义 7.14，对 L 关于 $\boldsymbol{w},b,\xi$ 求极小，即

$$\nabla_w L(\boldsymbol{w},b,\xi,\alpha,r)=0 \tag{7.115}$$

$$\nabla_b L(\boldsymbol{w},b,\xi,\alpha,r)=0 \tag{7.116}$$

$$\nabla_\xi L(\boldsymbol{w},b,\xi,\alpha,r)=0 \tag{7.117}$$

得到

$$\sum_{i=1}^{l}\alpha_i y_i=0 \tag{7.118}$$

$$\boldsymbol{w}=\sum_{i=1}^{l}\alpha_i y_i\boldsymbol{x}_i \tag{7.119}$$

$$C-\alpha_i-r_i=0 \tag{7.120}$$

然后将上述极值条件代入拉格朗日函数，对 α 求极大，得到对偶问题

$$\min_{\alpha}-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_i y_j\alpha_i\alpha_j K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)+\sum_{j=1}^{l}\alpha_j \tag{7.121}$$

$$\text{s.t.} \quad \sum_{i=0}^{l} y_i \alpha_i = 0 \tag{7.122}$$

$$0 \leq \alpha_i \leq C, \qquad i = 1, \text{L}\ , l \tag{7.123}$$

上述问题等价于问题式(7.107)~式(7.109)。

根据前面的推导及 Wolfe 对偶定理 7.7，容易得到下面的结论：

【定理 7.8】 设$(\boldsymbol{w}^*, b^*, \xi^*)$是原始问题式(7.111)~式(7.113)的解，则对偶问题式(7.107)~式(7.109)必有解$\alpha^* = (\alpha_1^*, \text{L}\ , \alpha_l^*)^{\text{T}}$，使得

$$\boldsymbol{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \boldsymbol{x}_i \tag{7.124}$$

根据这一定理，得到对偶问题的解α^*后，由式(7.124)得到原问题关于$\boldsymbol{w}$的解$\boldsymbol{w}^*$，进一步决策函数为

$$f(\boldsymbol{x}) = \text{sgn}(\langle \boldsymbol{w}^* \cdot \boldsymbol{x} \rangle + b^*) = \sum_{i=1}^{l} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^*$$

注意到对偶问题式(7.107)~式(7.109)是凸规划问题，不是严格凸规划，那么其解可能不唯一，因此仅由定理 7.8 并不能保证求得对偶问题的任一解$\bar{\alpha} = \{\bar{\alpha}_1, \text{L}\ , \bar{\alpha}_l\}$之后，直接由式$\bar{\boldsymbol{w}} = \sum_{i=1}^{l} \bar{\alpha}_i y_i \boldsymbol{x}_i$得到的$\bar{\boldsymbol{w}}$就是原始问题的解，所以上述定理不能作为建立算法的基础，现有的逻辑系统是从原始问题出发，得到原始问题和对偶问题的解关系。这是现有的逻辑系统存在的缺陷。要保证算法的合理性，需要从对偶问题出发讨论如何得到原始问题的解。

2．求解 b

求解 b 时，也存在同样的问题。由前面的推导可知，α^*，$i = 1, \text{L}\ , l$分别为约束条件

$$y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) + \xi_i \geq 1, \quad i = 1, \text{L}\ , l$$

所对应的拉格朗日乘子，由原始问题的 KKT 条件可知，如果$(\boldsymbol{w}, b)$是解，则存在对偶问题的解$\alpha^* = (\alpha_1^*, \text{L}\ , \alpha_l^*)^{\text{T}}$

$$\alpha_i \left(y_i \left(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b \right) + \xi_i - 1 \right) = 0, \quad i = 1, \text{L}\ , l \tag{7.125}$$

因此，若存在α^*的分量$\alpha_j^* > 0$，则相应地

$$y_j \left(\langle \boldsymbol{w} \cdot \boldsymbol{x}_j \rangle + b \right) + \xi_i - 1 = 0 \tag{7.126}$$

进一步，由式(7.109)得，若存在α^*的分量$\alpha_j^* < C$，则

$$\xi_j = 0 \tag{7.127}$$

由式(7.126)和(7.127)可得，若存在对偶问题的解α^*的分量α_j^*满足$0 < \alpha_j^* < C$，则计算b^*的公式为

$$b^* = y_j - \left\langle \boldsymbol{w}^* \cdot \boldsymbol{x}_i \right\rangle = y_j - \sum_{i=1}^{l} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{7.128}$$

从前面推导的过程可以看出，求解 b 的公式仍然是通过考察原始问题的 KKT 条件出发而得到的。显然和求解 $\boldsymbol{w}$ 存在同样的问题。算法 7 在求解 b 的公式时有限制条件，即要求存在对偶问题解 α^* 的分量 $0 < \alpha_j^* < C$。若找不到这样的分量，如例 7.4 所示，则算法失效。另一方面，这一算法只给出 b 的唯一解，事实上原始问题关于 b 的解可能不唯一。

【例 7.4】 设训练集为

$$T = \{(\boldsymbol{x}_1, y_1), \mathrm{L}\ , (\boldsymbol{x}_6, y_6)\} = \{(0,1), (1,1), (4,1), (3,-1), (6,-1)(7,-1)\} \tag{7.129}$$

即一维输入 0, 1, 4 为正类，3, 6, 7 为负类，这是一维线性不可分的问题。则原始问题关于 b 的解不唯一。

若选取惩罚参数 $C = 1/12$，则原始最优化问题为

$$\min_{\boldsymbol{w}, b, \xi} \frac{1}{2}\boldsymbol{w}^2 + \frac{1}{12}\sum_{i=1}^{6} \xi_i \tag{7.130}$$

$$\text{s.t.} \quad y_i \left(\left\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \right\rangle + b \right) \geqslant 1 - \xi_i, \quad i = 1, \mathrm{L}\ , 6 \tag{7.131}$$

$$\xi_i \geqslant 0, \quad i = 0, \mathrm{L}\ , 6 \tag{7.132}$$

为此构造一个依赖于参数 $\rho \in [1, 4/3]$ 的向量 $(\boldsymbol{w}^*, b^*, \xi^*)$：

令

$$\boldsymbol{w}^* = \boldsymbol{w}^*(\rho) = -\frac{1}{3}, \ b^* = b(\rho) = \rho, \ \xi^* = \xi^*(\rho) = (\xi_1^*(\rho), \mathrm{L}\ , \xi_6^*)) \tag{7.133}$$

$$\xi_1^*(\rho) = 0, \ \xi_2^*(\rho) = \frac{4}{3} - \rho, \ \xi_3^*(\rho) = \frac{7}{3} - \rho, \ \xi_4^*(\rho) = \rho, \ \xi_5^*(\rho) = \rho - 1, \ \xi_6^*(\rho) = 0 \tag{7.134}$$

对任意的 $\rho \in [1, 4/3]$，$(\boldsymbol{w}^*(\rho), b^*(\rho), \xi^*(\rho))$ 都是问题的解。引进拉格朗日乘子向量

$$\alpha^* = \left(0, \frac{1}{12}, \frac{1}{12}, \frac{1}{12} \cdot \frac{1}{12}, 0\right), \quad \beta^* = \left(\frac{1}{12}, 0, 0, 0, 0, \frac{1}{12}\right) \tag{7.135}$$

容易验证拉格朗日函数

$$L(\boldsymbol{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\boldsymbol{w}^2 + \frac{1}{12}\sum_{i=1}^{6} \xi_i - \sum_{i=1}^{6} \alpha_i (y_i (\left\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \right\rangle + b - 1) - \sum_{i=1}^{6} \beta_i \xi_i \tag{7.136}$$

满足

$$\nabla_{(\boldsymbol{w}, b, \xi)} L(\boldsymbol{w}^*, b^*, \xi^*, \alpha^*, \beta^*) = 0 \tag{7.137}$$

$$\nabla_{\alpha} L(\boldsymbol{w}^*, b^*, \xi^*, \alpha^*, \beta^*) \leqslant 0 \tag{7.138}$$

$$\nabla_{\beta} L(\boldsymbol{w}^*, b^*, \xi^*, \alpha^*, \beta^*) \leqslant 0 \tag{7.139}$$

$$\sum_{i=1}^{6}\alpha_i^*(y_i(\langle \boldsymbol{w}^*\cdot\boldsymbol{x}_i\rangle+b^*)-1=0 \tag{7.140}$$

$$\sum_{i=1}^{6}\beta_i^*\xi_i^*=0 \tag{7.141}$$

因为问题式(7.130)~式(7.132)是凸优化问题，当$\rho\in[1,4/3]$时，$(\boldsymbol{w}^*(\rho),b^*(\rho),\xi^*(\rho))$都是问题的解。由此可见区间$[1,4/3]$上的点都是问题关于 b 的解。

原始问题式(7.130)~式(7.132)的对偶问题为

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{6}\sum_{j=1}^{6}\alpha_i\alpha_j y_i y_j\boldsymbol{x}_i\boldsymbol{x}_j-\sum_{i=1}^{6}\alpha_i \tag{7.142}$$

$$\text{s.t.}\quad \sum_{i=1}^{6}\alpha_i y_i=0 \tag{7.143}$$

$$0\leq\alpha_i\leq\frac{1}{12},\quad i=1,\mathrm{L}\ ,6 \tag{7.144}$$

显然，$\alpha^*=\left(0,\frac{1}{12},\frac{1}{12},\frac{1}{12},\frac{1}{12},0\right)$为对偶问题的解，且不存在分量$0<\alpha_j^*<\frac{1}{12}$。

例7.4 说明了两个问题，原始问题关于 b 的解不唯一，而且对偶问题的解α^*不存在，分量α_j^*可满足$0<\alpha_j^*<C$，显然此时不能应用算法 7.4。

【例 7.5】 利用线性核函数对 Fisher's iris 数据包含 150 个鸢尾花样本进行分类，分类结果如图7.20所示。其中图中只显示了训练样本的分类结果。

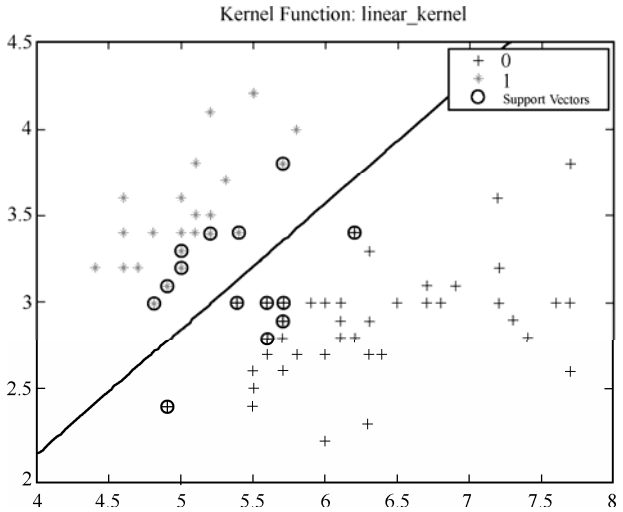


图 7.20 SVM 采用线性核函数的结果(训练样本)

既显示训练集也显示测试集的分类结果如图7.21所示。

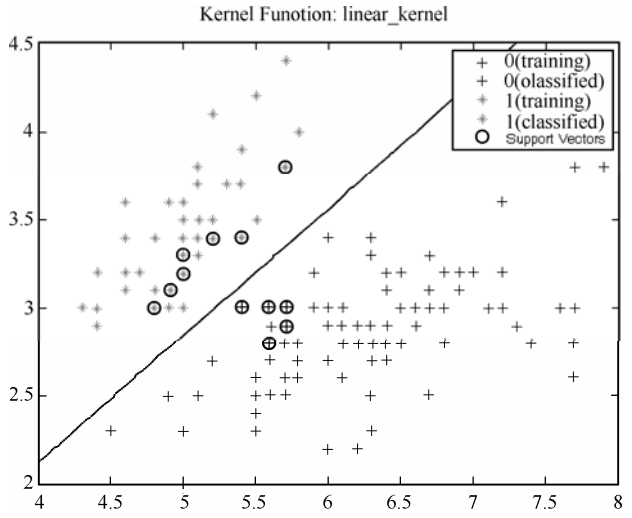


图 7.21　SVM 采用线性核函数的结果(训练和测试样本)

【例 7.6】 利用 RBF 径向基核函数训练 SVM 的得到的分类器结果如图 7.22 所示。

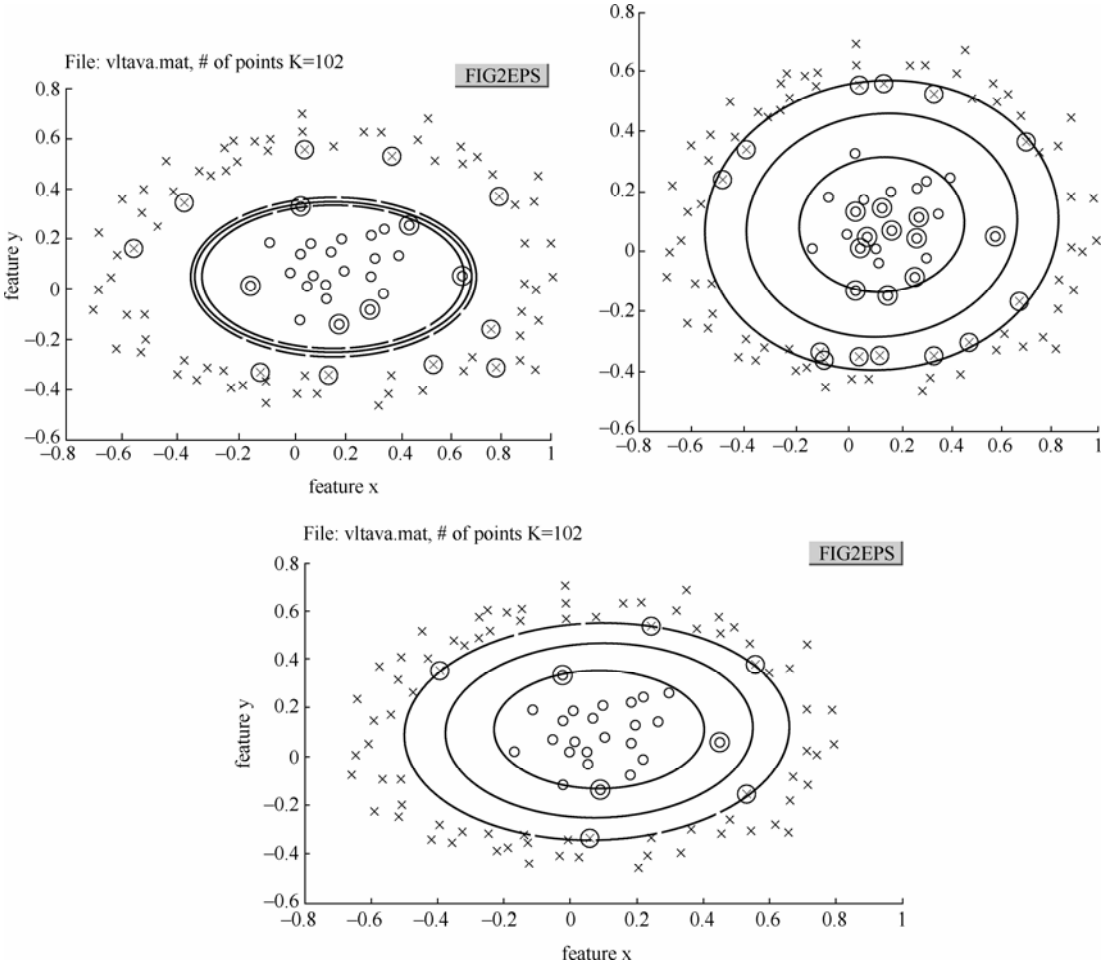
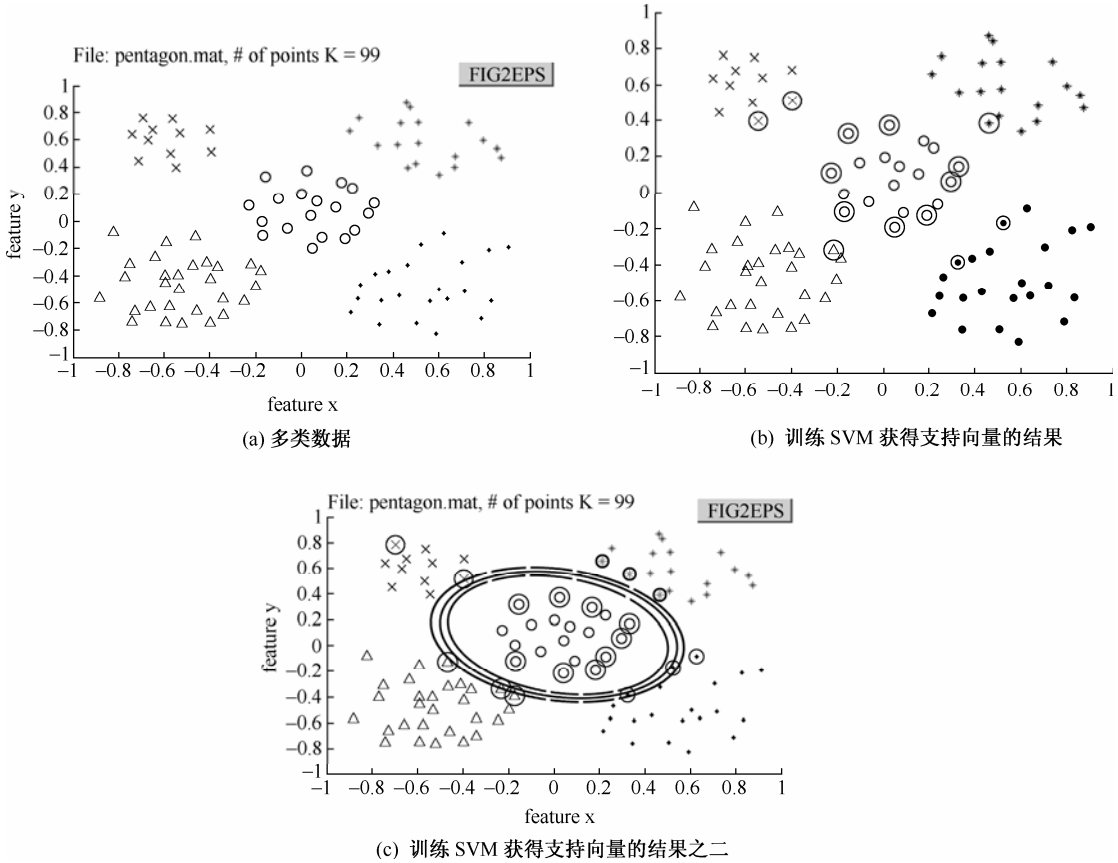


图 7.22　SVM 采用 RBF 核函数的结果

【例 7.7】 训练多分类支持向量机的结果如图7.23所示。



(a) 多类数据

(b) 训练 SVM 获得支持向量的结果

(c) 训练 SVM 获得支持向量的结果之二

图 7.23 多类 SVM 的训练结果

7.3.2 *C*-支持向量机的变形

由前面的讨论可知 *C*-SVC 算法的原始问题式(7.111)~式(7.113)关于 b 的解可能不唯一，将该问题的目标函数增加一项 $\frac{b^2}{2}$[21]，可以避免 b 的不唯一问题，此时相应的原始问题为

$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}(\|\boldsymbol{w}\|^2 + b^2) + C\sum_{i=1}^{l}\xi_i \tag{7.145}$$

$$\text{s.t.} \quad y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle + b) \geqslant 1-\xi_i, \quad i=1,\text{L}\ ,l \tag{7.146}$$

$$\xi_i \geqslant 0, \quad i=1,\text{L}\ ,l \tag{7.147}$$

求解该问题的对偶问题，然后构造决策函数，可以得到算法：

【算法 7.5】 *C*-支持向量机的变形

(1) 设已知训练集 $T=\{(\boldsymbol{x}_1,y_1),\text{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\text{L}\ ,l$。

(2) 选取适当的核函数 $K(\boldsymbol{x},\boldsymbol{x}')$ 和适当的参数 C，构造并求解最优化问题

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (K(\boldsymbol{x}_i, \boldsymbol{x}_j)+1) - \sum_{j=1}^{l}\alpha_j \tag{7.148}$$

$$\text{s.t.} \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \mathrm{L}\ , l \tag{7.149}$$

得最优解 $\alpha^* = (\alpha_1^*, \mathrm{L}\ , \alpha_l^*)^{\mathrm{T}}$。

(3) 构造决策函数 $f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b^*\right)$，其中 $b^* = \sum_{i=1}^{l} \alpha_i^* y_i$。

这一算法的优点在于对偶问题的约束简单，容易求解，而且可以证明问题式(7.145)~式(7.147)关于 $\boldsymbol{w}$ 和 b 的解都是唯一的。这就避免了 *C*-SVC 算法中计算 b 时可能不唯一的问题。但算法的理论依据仍然是和 *C*-SVC 类似，由原始问题的 Wolfe 对偶及 KKT 条件出发，得到求解 $\boldsymbol{w}$ 和 b 的公式，因此现有的逻辑系统存在问题。

数值实验表明修改后问题的解和原问题的解相差很小，但还没有严格的理论证明[21]。*C*-SVC 中的最优化问题式(7.111)~式(7.113)是基于统计学习理论的定理 7.2 而提出的，其中 $\sum_{i=1}^{l}\xi_i$ 体现了经验风险，而 $\|\boldsymbol{w}\|$ 体现了表达能力，因子 *C* 实质上是对经验风险和表达能力如何匹配的一个裁决，因此求解该问题就意味着在某种程度上极小化推广能力的上界。

7.3.3　广义支持向量机

支持向量机方法将分类问题转化为凸规划问题，在 *C*-SVC 中需要求解凸二次规划式(7.107)~式(7.109)，然后由这一问题的解 α^* 来确定决策函数。为了叙述的方便，将问题式(7.107)~式(7.109)用矩阵和向量的形式来表示：

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha} \tag{7.150}$$

$$\text{s.t.} \quad \boldsymbol{y}^{\mathrm{T}}\boldsymbol{\alpha} = \boldsymbol{0} \tag{7.151}$$

$$0 \leqslant \boldsymbol{\alpha} \leqslant C\boldsymbol{e} \tag{7.152}$$

这里 $\boldsymbol{H} = (y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{ij}$，$\boldsymbol{y} = (y_1, \mathrm{L}\ , y_l)$，$\boldsymbol{e} = (1, \mathrm{L}\ , 1)^{\mathrm{T}} \in R^l$，$0$ 表示零向量。注意到，在 *C*-SVC 中选取的核函数是正定核，因此矩阵 $\boldsymbol{H}$ 是对称半正定的，进而优化问题式(7.150)~式(7.152)是凸规划。若选取非正定核，如 Sigmoid 核，则此时优化问题式(7.150)~式(7.152)不是凸规划，而原有的 *C*-SVC 算法 7.3 是建立在凸规划的 Wolfe 对偶之上的，显然对非正定核不适用。

为了解决这一问题，Mangasarian 在文献[22]中，提出了一种广义支持向量机方法(GSVM)。GSVM 需要求解如下的优化问题：

$$\min_{\alpha, b, \xi} \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\bar{\boldsymbol{H}}\boldsymbol{\alpha} + C\boldsymbol{e}^{\mathrm{T}}\xi \tag{7.153}$$

$$\text{s.t.} \quad \boldsymbol{H}\boldsymbol{\alpha} + \boldsymbol{y}b + \xi \geqslant \boldsymbol{e} \tag{7.154}$$

$$\xi \geqslant 0 \tag{7.155}$$

和 *C*-SVC 类似，引入它的对偶问题

$$\min_{r} \frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{H}\bar{\boldsymbol{H}}^{-1}\boldsymbol{H}\boldsymbol{r}-\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r} \tag{7.156}$$

$$\text{s.t.} \quad \boldsymbol{y}^{\mathrm{T}}\boldsymbol{r}=0 \tag{7.157}$$

$$0\leqslant \boldsymbol{r}\leqslant C\boldsymbol{e} \tag{7.158}$$

通过求解对偶问题来得到原始问题的解。其具体算法步骤如下：

【算法 7.6】 GSVM

(1) 设已知训练集 $T=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\mathrm{L}\ ,l$。

(2) 选择适当的正数 C，核函数 $K(\boldsymbol{x},\boldsymbol{x}')$ 以及对称正定阵 $\bar{H}$，构造并求解最优化问题式(7.145)~式(7.147)，得到最优解 $\boldsymbol{r}^*=(r_1^*,\mathrm{L}\ ,r_l^*)$

(3) 构造决策函数

$$f(\boldsymbol{x})=\sum_{i=1}^{l}\alpha_i^*K(\boldsymbol{x}_i,\boldsymbol{x})+b^* \tag{7.159}$$

其中，

$$\boldsymbol{\alpha}^*=\bar{\boldsymbol{H}}^{-1}\boldsymbol{H}\boldsymbol{r}^*$$

b^*按照式(7.160)计算，选择 r^* 位于开区间 $(0,C)$ 中的分量 r_j^*，令

$$b^*=y_j-\sum_{i=1}^{l}y_i\alpha_i^*K(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{7.160}$$

注意到，这时不论核是否为正定核，矩阵 $\boldsymbol{H}\bar{\boldsymbol{H}}^{-1}\boldsymbol{H}$ 都是对称半正定的，即算法中的优化问题式(7.156)~式(7.158)是一个凸二次规划。因此，这一算法适用于任意的核函数。

算法在求解 $\boldsymbol{w}$ 和 b 时存在和 C-SVC 类似问题，这里不再详细说明。如果令问题式(7.153)~式(7.155)目标函数中的正定矩阵 $\bar{\boldsymbol{H}}$ 为 $\bar{\boldsymbol{H}}=\boldsymbol{H}$，则可以证明 $\boldsymbol{\alpha}=\boldsymbol{r}$，此时对偶问题式(7.156)~式(7.158)就等价于 C-SVC 中的优化问题式(7.150)~式(7.152)[23]。此时算法 7.6 就退化为算法 7.2。但首先要求 $\boldsymbol{H}$ 是对称正定矩阵，事实上，C-SVC 算法的优化问题式(7.150)~式(7.152)中，$\boldsymbol{H}$ 是对称半正定矩阵，因此从这个意义上来说，GSVM 并不能完全包含标准的 C-SVC 算法。

7.3.4 v-支持向量机

设给定训练集

$$T=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l \tag{7.161}$$

其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y\in\{1,-1\}$，$i=1,\mathrm{L}\ ,l$。

在 C-支持向量机中，最大化间隔和最小化训练错误是两个相互矛盾的目标。其中常数 C 起着调和这两个目标的作用。定性地讲，C 值有着明确的含义：选取大的 C 值，意味着

更强调最小化训练错误。但定量地讲，C 值本身并没有确切的意义，所以 C 值的选取比较困难。为此，Scholkopf 等提出了一个改进的方法——v-支持向量分类机（v-SVC）[24]，它用参数 v 代替参数 C，而参数 v 有一些直观上的意义，容易选取，这样就避免了 C-SVC 中选取 C 时遇到的困难。

v-支持向量分类机的原始最优化问题为

$$\min_{\boldsymbol{w},b,\xi,\rho} \tau(\boldsymbol{w},\xi,\rho)=\frac{1}{2}\|\boldsymbol{w}\|^2-\upsilon\rho+\frac{1}{l}\sum_{i=1}^{l}\xi_i \tag{7.162}$$

$$\text{s.t.}\quad y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle+b)\geqslant \rho-\xi_i \tag{7.163}$$

$$\xi_i\geqslant 0,\quad i=1,\mathrm{L}\ ,l\qquad \rho\geqslant 0 \tag{7.164}$$

其中$\xi=(\xi_1,\mathrm{L}\ ,\xi_l)^{\mathrm{T}}$与已支持向量分类机的原始问题式（7.111）~式（7.113）比较，这里不含参数 C，而是换成了参数 v，这是需要实际选定的参数。另外还多了一个变量ρ。注意到当$\xi=0$的时候，约束条件（7.163）意味着两类点以 $\dfrac{2\rho}{\|\boldsymbol{w}\|}$ 的间隔被分开。

和前面的推导类似，可以得到其对偶问题为

$$\min_{\alpha}\quad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_jK(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{7.165}$$

$$\text{s.t.}\quad \sum_{i=1}^{l}y_i\alpha_i=0 \tag{7.166}$$

$$0\leqslant \alpha_i\leqslant \frac{1}{l},\qquad i=1,\mathrm{L}\ ,l \tag{7.167}$$

$$\sum_{i=1}^{l}\alpha_i\geqslant \boldsymbol{v} \tag{7.168}$$

v-支持向量机同样是通过对偶问题的解来确定决策函数的，具体步骤如下：

【算法 7.7】 v-SVC

（1）设已知训练集 $T=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y=\{1,-1\}$，$i=1,\mathrm{L}\ ,l$。

（2）选取适当的参数 v 和核函数 $K(\boldsymbol{x},\boldsymbol{x}')$，构造并求解最优化问题式（7.165）~式（7.168），得最优解 $\boldsymbol{\alpha}^*=(\alpha_i^*,\mathrm{L}\ ,\alpha_l^*)^{\mathrm{T}}$。

（3）选取 $j\in S_+=\{i\mid\alpha_i^*\in(0,1/l),y_i=1\}$，$k\in S_-=\{i\mid\alpha_i^*\in(0,1/l),y_i=-1\}$，计算

$$b^*=-\frac{1}{2}\sum_{i=1}^{l}\alpha_i^*y_i(K(\boldsymbol{x}_i,\boldsymbol{x}_j)+K(\boldsymbol{x}_i,\boldsymbol{x}_k)) \tag{7.169}$$

（4）构造决策函数

$$f(\boldsymbol{x})=\mathrm{sgn}\left(\sum_{i=1}^{l}\alpha_i^*y_iK(\boldsymbol{x},\boldsymbol{x}_i)+b^*\right)$$

下面的定理解释了 v 的意义。

【定理 7.9】 设给定由 l 个样本点组成的训练集 T［见式(7.161)］，并用算法 7.7(v-SVC)进行分类。若所得到的 $\rho^*>0$，则

(1)若间隔错误样本点的个数为 p，则 $v\geq p/l$，即 v 是间隔错误样本的个数与总样本点数之比的上界。

(2)若支持向量的个数为 q，则 $v\leq q/l$，即 v 是支持向量的个数与总样本点数之比的下界。

这里间隔错误样本点是指被超平面错分的点，支持向量是指对应于对偶问题的解 α^* 的非零分量的样本点。

可以证明，在一定条件下，当样本点个数 $l\to\infty$时，v 以概率 1 渐近于支持向量个数和样本点个数之比。由此可见，定理 7.9 和上述结论为 v 值的选取提供了依据。

算法 7.9 存在和 C-SVC 类似的问题，它是从原始问题出发来得到求解 $\boldsymbol{w}$ 和 b 的公式的，而且在求解 b 时，算法要求对偶问题的解 α^* 存在两个分量 α_j^* 和 α_k^* 分别满足 $\alpha_j^*\in(0,1/l)$, $y_j=1$ 和 $\alpha_k^*\in(0,1/l)$, $y_k=-1$，然后由公式(7.171)来计算 b。显然，若对偶问题的解中不存在这样的分量，则算法就无法执行。换句话说，算法只适用于集合 $S_+\neq\varnothing$ 和 $S_-\neq\varnothing$ 的情况，其中 $S_+=\{i\,|\,\alpha_i^*\in(0,1/l),\ y_i=1\}$, $S_-=\{i\,|\,\alpha_i^*\in(0,1/l),\ y_i=-1\}$；如果集合 S_+和 S_-其中一个为空集时，上述算法不可行。

7.4 支持向量回归机

支持向量机最初是解决分类问题的方法，Vapnik[23]首次将支持向量机方法应用于回归问题，提出支持向量回归机 ε-SVR。在 ε-SVR 算法中，需要事先给定参数 ε 的值，在实际应用中，有时很难选择合适的 ε，这种情形与 C-SVC 中的需要事先选定 C 的情形相似。因此，相应于 v-SVC，Scholkopf[24]提出能够自动计算 ε 的 v-支持向量回归机(v-SVR)。下面，首先引入回归问题，并介绍线性回归和非线性回归，然后分别介绍 ε-SVR 和 v-SVR 算法，分析算法中存在的问题。

7.4.1 回归问题

首先讨论一个最简单的回归问题：考虑两个量 x 和 y 的关系，设已测得若干个 $\boldsymbol{x}$ 值和其相对应的 y 值

$$T=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l \tag{7.170}$$

其中 $\boldsymbol{x}_i\in X=R^n$, $y_i\in Y=R$, $i=1,\mathrm{L}\ ,l$。

根据这 l 对值，推断 y 对 $\boldsymbol{x}$ 的依赖关系 $y=f(\boldsymbol{x})$。从几何图像上看，把这 l 个点标在 $(\boldsymbol{x},y)$ 平面上，图7.24 问题就变为寻求一条曲线 $y=f(\boldsymbol{x})$ (图中 $\boldsymbol{x}$ 为一维变量)。将上面的问题推广，用数学的语言来描述如下。

回归问题：设给定训练集

$$T=\{(\boldsymbol{x}_1,y_1),\mathrm{L}\ ,(\boldsymbol{x}_l,y_l)\}\in(X,Y)^l$$

其中 $\boldsymbol{x}_i\in X=R^n$, $y_i\in Y=R$, $i=1,\mathrm{L}\ ,l$；假定训练集是按 $X\times Y$ 上的某个概率分布 $P(\boldsymbol{x},y)$ 选

取的独立同分布的样本点，又设给定损失函数 $C(\boldsymbol{x}, y, f)$。试寻求一个函数 $f(\boldsymbol{x})$，使得期望风险

$$R(f)=\int c(\boldsymbol{x}, y, f)\mathrm{d}P(\boldsymbol{x}, y) \tag{7.171}$$

达到极小。其中概率分布 $P(\boldsymbol{x}, y)$ 是未知的，已知的仅仅是训练集。回归问题数学提法与分类问题数学提法相似，主要不同之处在于变量 y 的取值。在分类问题中，变量 y 仅取 -1 和 1 两个值，即 $y_i \in Y \in \{1,-1\}$。但在回归问题中，变量 y 可取任意实数值，即 $y \in Y = R$。

从上述回归问题的数学提法可以看出，需要选择适当的损失函数。图7.25 是几种可能的损失函数。

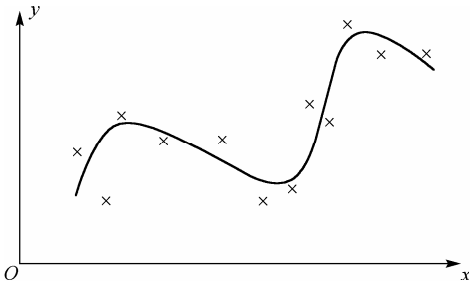


图 7.24　回归问题

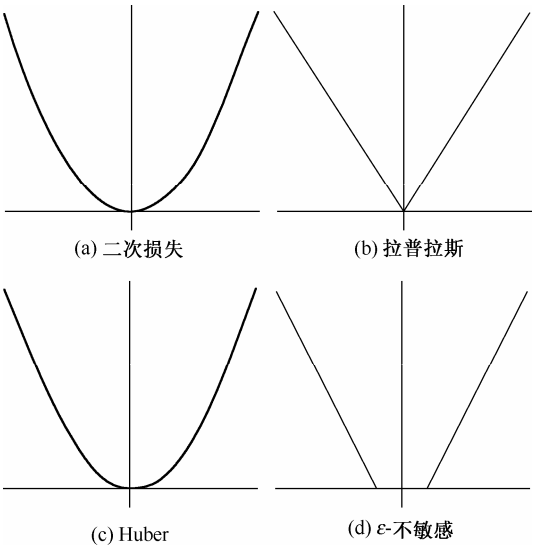


图 7.25　几种损失函数

图 7.25(a)对应传统的最小平方误差准则。图 7.25(b)是拉普拉斯损失函数，和图(a)的二次损失函数比较对外部信息不是很敏感。Huber 提出的图7.25(c)损失函数具有最优性，即当数据分布未知时，此损失函数是鲁棒的。这三种损失函数不能产生较少的支持向量。为了解决这个问题，Vapnik 提出了图7.25(d)所示的 ε-不敏感损失函数，它与 Huber 损失函数近似，但能获得很少的支持向量数，ε-不敏感损失函数也是回归估计中最常用的一种损失函数。

7.4.2　线性回归

考虑数据集合(7.170)，

$$T=\{(\boldsymbol{x}_1, y_1),\mathrm{L}\ ,(\boldsymbol{x}_l, y_l)\} \in (X, Y)^l$$

其中 $\boldsymbol{x}_i \in X = R^n$，$y_i \in Y = R$，$i = 1,\mathrm{L}\ ,l$。

线性函数

$$f(\boldsymbol{x})=(\boldsymbol{w}\cdot\boldsymbol{x})+b \tag{7.172}$$

最优回归函数由最小化函数(7.173)函数获得：

$$\phi(\boldsymbol{w},\xi)=\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_i(\xi_i^-+\xi_i^+) \tag{7.173}$$

其中 C 是预先给定值，ξ^- 和 ξ^+ 分别是约束系统输出的上下界，是松弛变量。

1．ε-不敏感损失函数

ε-不敏感损失函数如图7.25(d)所示：

$$L_\varepsilon(y)=\begin{cases}0, & |f(\boldsymbol{x})-y|<\varepsilon\\ |f(\boldsymbol{x})-y|-\varepsilon, & \text{其他}\end{cases} \tag{7.174}$$

其解由式(7.175)给出：

$$\max_{\alpha,\alpha^*}W(\boldsymbol{\alpha},\boldsymbol{\alpha}^*)=\max_{\alpha,\alpha^*}-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)+\sum_{i=1}^{l}\alpha_i(y_i-\varepsilon)-\alpha_i^*(y_i-\varepsilon) \tag{7.175}$$

或

$$\boldsymbol{\alpha},\boldsymbol{\alpha}^*=\arg\min_{\alpha,\alpha^*}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)y_i+\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)\varepsilon \tag{7.176}$$

约束项为

$$0\leq\alpha_i,\alpha_i^*\leq C,\qquad i=1,\mathrm{L}\ ,l \tag{7.177}$$

$$\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0$$

解具有约束项(7.177)的方程(7.175)即可确定拉格朗日乘子 $\boldsymbol{\alpha},\boldsymbol{\alpha}^*$，回归函数由式(7.174)给出，其中

$$\bar{\boldsymbol{w}}=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\boldsymbol{x}_i\ ,\quad \bar{b}=-\frac{1}{2}(\bar{\boldsymbol{w}}\cdot(\boldsymbol{x}_r+\boldsymbol{x}_s)) \tag{7.178}$$

通过解方程(7.179)， Karush-Kuhn-Tucker(KKT)条件被满足：

$$\bar{\alpha}_i\bar{\alpha}_i^*=0\ ,\quad i=1,\mathrm{L}\ ,l \tag{7.179}$$

因此支持向量是大于 0 的拉格朗日乘子之一。当 $\varepsilon=0$，得到 L_1 损失函数，且最优问题得到简化：

$$\min_{\beta}\frac{1}{2}\sum_{j=1}^{l}\beta_i\beta_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{i=1}^{l}\beta_i y_i \tag{7.180}$$

约束条件为

$$\begin{aligned} &-C \leqslant \beta_i \leqslant C, \qquad i=1,\mathrm{L}\ ,l \\ &\sum_{i=1}^{l}\beta_i = 0 \end{aligned} \tag{7.181}$$

回归函数仍然由式(7.174)给出，其中

$$\bar{\boldsymbol{w}} = \sum_{i=1}^{l}\beta_i \boldsymbol{x}_i\ , \quad \bar{b} = -\frac{1}{2}(\bar{\boldsymbol{w}}\cdot(\boldsymbol{x}_r + \boldsymbol{x}_s)) \tag{7.182}$$

2．二次损失函数

二次损失函数如图7.25(a)所示，其函数形式为

$$L_{\text{quad}}(f(\boldsymbol{x})-y) = (f(\boldsymbol{x})-y)^2 \tag{7.183}$$

其解为

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} W(\alpha,\alpha^*) = \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} &-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) + \\ &\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)y_i - \frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2+(\alpha_i^*)^2) \end{aligned} \tag{7.184}$$

相应的最优解可以通过用 KKT 条件简化，式(7.179)隐含着 $\beta_i^* = |\beta_i|$，则最优解为

$$\min_{\beta}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i\beta_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) - \sum_{i=1}^{l}\beta_i y_i + \frac{1}{2C}\sum_{i=1}^{l}\beta_i^2 \tag{7.185}$$

约束条件为

$$\sum_{i=1}^{l}\beta_i = 0 \tag{7.186}$$

回归函数由式(7.174)和式(7.183)给出。

3．Huber 损失函数

Huber 损失函数如图 7.25(c)所示，其表达式为

$$L_{\text{huber}}(f(\boldsymbol{x})-y) = \begin{cases} \dfrac{1}{2}(f(\boldsymbol{x})-y)^2, & |f(\boldsymbol{x})-y| < \mu \\ \mu|f(\boldsymbol{x})-y| - \dfrac{\mu^2}{2}, & \text{其他} \end{cases} \tag{7.187}$$

其解为

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} W(\alpha,\alpha^*) = \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} &-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) + \\ &\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)y_i - \frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2+(\alpha_i^*)^2)\mu \end{aligned} \tag{7.188}$$

最优解为

$$\min_{\beta}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i\beta_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)-\sum_{i=1}^{l}\beta_i y_i+\frac{1}{2C}\sum_{i=1}^{l}\beta_i^2\mu \tag{7.189}$$

满足约束条件

$$\begin{aligned}&-C\leq\beta_i\leq C, \qquad i=1,\mathrm{L}\ ,l\\&\sum_{i=1}^{l}\beta_i=0\end{aligned} \tag{7.190}$$

【例7.8】 表 7.3 所示为数据集，采用拉普拉斯损失函数的支持向量回归解如图 7.26 所示。

表 7.3 线性回归用数据

x	y	x	y
1.0	−1.6	10.2	6.8
3.0	−1.8	11.0	10.0
4.0	−1.0	11.5	10. 0
5.6	1.2	12.7	10.0
7.8	2.2		

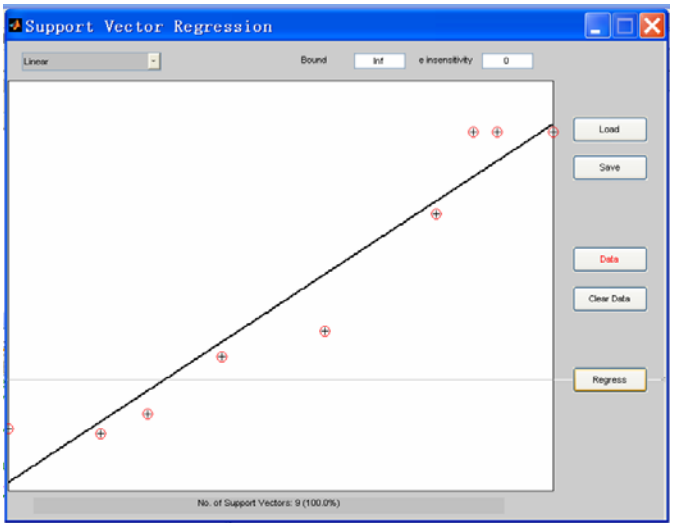


图 7.26 线性回归解

7.4.3 非线性回归

和分类问题类似，非线性建模通常需要大量建模数据。因此采用和非线性SVC相同的方法，将数据映射到高维特征空间，在高维特征空间实现线性回归。利用核方法克服了维数灾难。非线性回归解通常使用图7.26(d)所示的 ε-不敏感损失函数，由式(7.191)给出

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*}W(\alpha,\alpha^*)=\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*}\sum_{i=1}^{l}\alpha_i^*(y_i-\varepsilon)-\alpha_i(y_i+\varepsilon)-\\\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)\end{aligned} \tag{7.191}$$

具有约束

$$\begin{aligned} &0 \leq \alpha_i, \alpha_i^* \leq C, \qquad i = 1,\mathrm{L}\ ,l \\ &\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \end{aligned} \tag{7.192}$$

解具有约束(7.192)的方程(7.191)就可以确定拉格朗日乘子α_i, α_i *，回归函数由式(7.193)确定：

$$f(\boldsymbol{x}) = \sum_{SVs}(\bar{\alpha}_i - \alpha_i^*)K(\boldsymbol{x}_i, \boldsymbol{x}) + \bar{b} \tag{7.193}$$

其中，

$$(\boldsymbol{w} \cdot \boldsymbol{x}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\bar{b} = -\frac{1}{2}\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(K(\boldsymbol{x}_i, \boldsymbol{x}_r) + K(\boldsymbol{x}_i, \boldsymbol{x}_s))$$

对于其他损失也通过替换核函数点积的方法得到最优化准则。ε-不敏感损失函数不像二次和 Huber 等损失函数，其中的所有数据点都将是支持向量，其 SV 解是稀疏的，二次损失函数将产生尖峰回归或零阶调整，其中调整参数是$\lambda = \dfrac{1}{2C}$。举例说明一些非线性 SVR 解，将各种核函数用于建模表 7.3 的回归数据，采用ε-不敏感损失函数(ε= 0.5)。图7.26是采用 2 阶多项式核的 SVR 解，SV 用圆圈标示，虚线描述了ε-不敏感区域的解范围，如果所有的数据点位于这个区域，将具有零误差——即无损失；图7.27说明在ε-不敏感区域没有 SV。

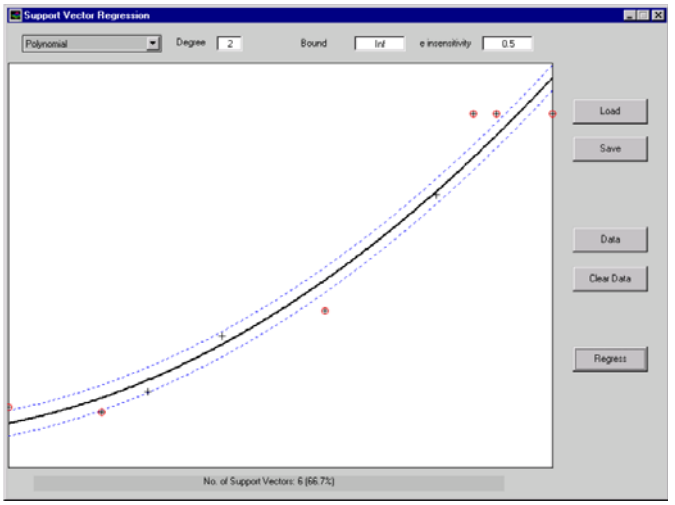


图 7.27　多项式回归示例

图 7.28 是采用径向基函数的SV 解，其中$\sigma = 1.0$，在此例中，损失函数具有零误差，而且验证了所有数据点位于或在ε-不敏感区域。

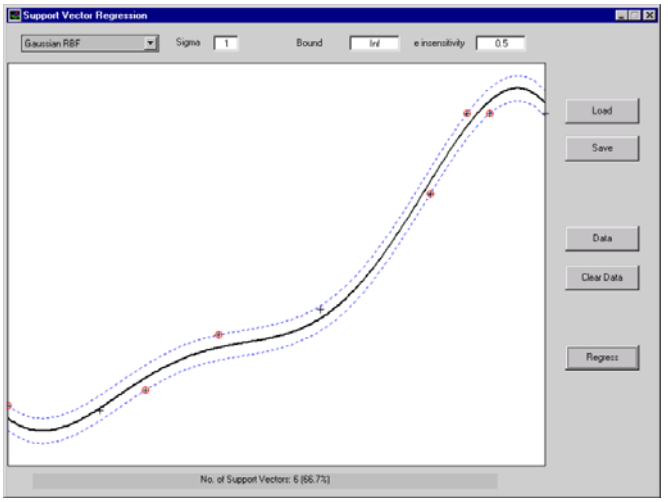


图 7.28 径向基函数回归

7.4.4 ε-支持向量回归机

下面介绍使用 ε-不敏感损失函数的支持向量回归机。和 ε-支持向量回归机相对应的原始最优化问题为

$$\min_{\boldsymbol{w}\in R^n,\,\xi^{(*)},\,b\in R}\tau(\boldsymbol{w},\xi^{(*)})=\frac{1}{2}\|\boldsymbol{w}\|^2+C\cdot\frac{1}{l}\sum_{i=1}^{l}(\xi_i+\xi_i^*) \tag{7.194}$$

$$\text{s.t.}\quad((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)-y_i\leq\varepsilon+\xi_i,\qquad i=1,2,\mathrm{L}\ ,l \tag{7.195}$$

$$y_i-((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\leq\varepsilon+\xi_i^*,\qquad i=1,2,\mathrm{L}\ ,l \tag{7.196}$$

$$\xi_i^{(*)}\geq 0,\qquad i=1,2,\mathrm{L}\ ,l \tag{7.197}$$

其中(*)表示向量有*号和无*号两种情况的简单记号。例如 $\xi_i^{(*)}\geq 0$ 意味着 $\xi_i\geq 0$ 和 $\xi_i^*\geq 0$，$\boldsymbol{\xi}^{(*)}$ 表示向量 $(\xi_1,\xi_1^*,\mathrm{L}\ ,\xi_l,\xi_l^*)^{\mathrm{T}}$ 。和 C-SVC 类似，引入问题式(7.194)~式(7.197)的对偶问题：

$$\min_{\boldsymbol{\alpha}^{(*)}\in R^{2l}}\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(\boldsymbol{x}_i,\boldsymbol{x}_j)+\varepsilon\sum_{i=1}^{l}(\alpha_i^*+\alpha_i)-\sum_{i=1}^{l}y_i(\alpha_i^*-\alpha_i) \tag{7.198}$$

$$\text{s.t.}\quad\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0 \tag{7.199}$$

$$0\leq\alpha_i,\alpha_i^*\leq\frac{C}{l},\qquad i=1,2,\mathrm{L}\ ,l \tag{7.200}$$

其中 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 是核函数。对偶问题式(7.198)~式(7.200)对输入 $\boldsymbol{x}_i(i=1,2,\mathrm{L}\ ,l)$ 的依赖关系仅仅体现在核函数 $(\boldsymbol{x}_i,\boldsymbol{x}_j)$ $(i,j=1,\mathrm{L}\ ,l)$ 上。因而其解 α^* 及最终的决策函数也仅仅依赖于核函数。

具体的算法步骤如下：

【算法 7.8】　ε-支持向量回归机 ε-SVR

(1) 设已知训练集 $T=\{(\boldsymbol{x}_1, y_1),\mathrm{L}\ ,(\boldsymbol{x}_l, y_l)\}\in(X,Y)^l$，其中 $\boldsymbol{x}_i\in X=R^n$，$y_i\in Y=R$，$i=1,\mathrm{L}\ ,l$。

(2) 选择适当的正数 ε 和 C，选择适当的核 $K(\boldsymbol{x},\boldsymbol{x}')$。

(3) 构造并求解最优化问题式(7.198)~式(7.200)，得到最优解 $\overline{\boldsymbol{\alpha}}=(\overline{\alpha}_1,\overline{\alpha}_1^*,\mathrm{L}\ ,\overline{\alpha}_l,\overline{\alpha}_l^*)^{\mathrm{T}}$。

(4) 构造决策函数

$$f(\boldsymbol{x})=\sum_{i=1}^{l}(\overline{\alpha}_i^*-\overline{\alpha}_i)K(\boldsymbol{x}_i,\boldsymbol{x})+\overline{b} \tag{7.201}$$

其中 $\overline{b}$ 按照式(7.202)计算；选择位于开区间 $(0,C/l)$ 中的 $\overline{\alpha}_j$ 或 $\overline{\alpha}_k^*$。若选择的是 $\overline{\alpha}_j$，则

$$\overline{b}=y_j-\sum_{i=1}^{l}(\overline{\alpha}_i^*-\overline{\alpha}_i)K(\boldsymbol{x}_i,\boldsymbol{x}_j)+\varepsilon \tag{7.202}$$

若选择的是 $\overline{\alpha}_k^*$，则

$$\overline{b}=y_k-\sum_{i=1}^{l}(\overline{\alpha}_i^*-\overline{\alpha}_i)K(\boldsymbol{x}_i,\boldsymbol{x}_k)-\varepsilon \tag{7.203}$$

显然，上述算法表明，得到对偶问题的解 $\overline{\boldsymbol{\alpha}}^{(*)}$ 后，需要先选取位于开区间 $(0,C/l)$ 中的 $\overline{\alpha}_j$ 或 $\overline{\alpha}_k^*$，才能确定决策函数中的参数 b，这就意味着要求对偶问题式(7.198)~式(7.200)的解，$\overline{\boldsymbol{\alpha}}^{(*)}$ 中至少存在某个分量在开区间 $(0,C/l)$ 内，否则算法将无法执行。另一方面，此时只给出 b 的唯一解情况。

7.4.5　v-支持向量回归机

在 ε-支持向量回归机中，需要事先确定 ε-不敏感损失函数中的参数 ε。然而在某些情况下选择合适的 ε 并不是一件容易的事情。这种情形与 C-SVC 中的需要事先选定 C 的情形相似。在那里，引进了 v-SVC。相应地，本节引进能够自动计算 ε 的 v-支持向量回归机(v-SVR)作为 ε-SVR 的一种变形。

在 ε-支持向量回归机(ε-SVR)中，出发点是选定 ε 和 C，求解最优化问题(7.194)~(7.195)。与此不同，这里的出发点是选定另外一个参数 $v(v\geqslant 0)$ 和 C，即把最优化问题修改

$$\min_{\boldsymbol{w}\in H,\,\xi^{(*)}\in R^{2l},\,\varepsilon,\,b\in R}\frac{1}{2}\|\boldsymbol{w}\|^2+C\cdot\left(v\varepsilon+\frac{1}{l}\sum_{i=1}^{l}(\xi_i+\xi_i^*)\right) \tag{7.204}$$

$$\text{s.t.}\quad((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)-y_i\leqslant\varepsilon+\xi_i \tag{7.205}$$

$$y_i-((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b)\leqslant\varepsilon+\xi_i^* \tag{7.206}$$

$$\xi_i^*\geqslant 0,\quad \varepsilon\geqslant 0 \tag{7.207}$$

其中 $\boldsymbol{\xi}^{(*)}=(\xi_1,\xi_1^*,\mathrm{L}\ ,\xi_l,\xi_l^*)^{\mathrm{T}}$。注意与原来的原始问题式(7.194)~式(7.197)不同，这里的 ε 是作为优化问题的变量出现的，其值将作为解的一部分给出。

原始问题式(7.204)~式(7.207)的对偶问题是

$$\max_{\boldsymbol{\alpha}^{(*)}\in R^l} W(\boldsymbol{\alpha}^{(*)}) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i - \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{7.208}$$

$$\text{s.t.} \quad \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{7.209}$$

$$\alpha_i^{(*)} \in [0, C/l], \qquad i = 1, \text{L}\ , l \tag{7.210}$$

$$\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \leq C \cdot v \tag{7.211}$$

其中 $v \geq 0$，$C > 0$ 是常数，根据对偶问题的解 $\boldsymbol{\alpha}^{(*)}$ 确定决策函数。

【算法 7.9】 v-支持向量回归机

(1) 设已知训练集 $T = \{(\boldsymbol{x}_1, y_1), \text{L}\ , (\boldsymbol{x}_l, y_l)\} \in (X \times Y)^l$，其中 $\boldsymbol{x}_i \in X = R^n$，$y_i \in Y = R$，$i = 1, \text{L}\ , l$。

(2) 选择适当的正数 v 和 C，选择适当的核 $K(\boldsymbol{x}, \boldsymbol{x}')$。

(3) 构造并求解最优化问题，得到最优解 $\bar{\boldsymbol{\alpha}} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \text{L}\ , \bar{\alpha}_l, \bar{\alpha}_l^*)^{\text{T}}$。

(4) 构造决策函数

$$f(\boldsymbol{x}) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)K(\boldsymbol{x}_i, \boldsymbol{x}) + b^* \tag{7.212}$$

其中 b^* 按照式(7.213)计算；选择 $\bar{\boldsymbol{\alpha}}^{(*)}$ 位于开区间 $(0, C/l)$ 中的 $\bar{\alpha}_j$ 或 $\bar{\alpha}_k^*$。令

$$b^* = \frac{1}{2}[y_j + y_k - (\sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(\boldsymbol{x}_i, \boldsymbol{x}_k))] \tag{7.213}$$

如果还需计算 ε^*，可以使用与式(7.213)对应的公式

$$\varepsilon^* = \sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(\boldsymbol{x}_i, \boldsymbol{x}_k) + b^* - y_j \tag{7.214}$$

或

$$\varepsilon^* = y_k - \sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(\boldsymbol{x}_i, \boldsymbol{x}_k) - b^* \tag{7.215}$$

显然，算法中求解决策函数中的 b^* 时，需要对偶问题的解 $\bar{\boldsymbol{\alpha}}^{(*)}$ 中至少有 2 个分量 $\bar{\alpha}_j$ 和 $\bar{\alpha}_k^*$ 均位于开区间 $(0, C/l)$ 中。

下面解释 v 的含义。

【定理 7.10】 设已知训练集 $T = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \text{L}\ , (\boldsymbol{x}_l, \boldsymbol{y}_l)\} \in (X, Y)^l$，其中 $\boldsymbol{x}_i \in X = R^n$，$\boldsymbol{y}_i \in Y = R$，$i = 1, \text{L}\ , l$；并用 v-支持向量回归机进行回归，若所得到的 ε^* 值非零，则

(1)若记错误样本的个数为 q，则 $v \geq q/l$，即 v 是错误样本的个数所占总样本点数的份额的上界。

(2)若记支持向量的个数为 p，则 $v \leq p/l$，即 v 是支持向量的个数所占总样本点数的份额的下界。

另外，在一定条件下，还可以证明，当训练集 T 中的样本点个数 $l \to \infty$ 时，v 以 1 的概率渐近于支持向量个数与样本点个数之比，也渐近于错误样本点个数与样本点个数之比。

由此可见大体上可以用 $v(0 \leqslant v \leqslant 1)$ 来控制支持向量个数或错误样本点个数。这就为 v 值的选取提供了一个依据。

习题 7

7.1　Fisher 准则方法与支持向量机提出的最佳准则是不一致的，它们是否有各自适用的范围？

7.2　异或问题(XOR)是最简单的一个无法直接对特征采用线性判别函数来解决的问题。对于空间中的点 $\boldsymbol{x}_1 = (1, 1)^{\mathrm{T}}$, $\boldsymbol{x}_2 = (-1, -1)^{\mathrm{T}}$, $\boldsymbol{x}_3 = (1, -1)^{\mathrm{T}}$ 和 $\boldsymbol{x}_4 = (-1, 1)^{\mathrm{T}}$。设计解决 XOR 问题的 SVM。

7.3　以习题 7.2 为基础考虑另外 4 个特征，除了上面 4 个特征点之外的其他 $\binom{4}{2} - 1 = 5$ 对特征组合，作出样本和判别函数 $g = \pm 1$ 对应的直线。在所作图中，这些间隔是否一样？请给出解释。

7.4　通过修改感知器算法，写出实现“支持向量机”(SVM)学习算法的伪码程序。对当前最难分的样本的操作，给出详细的数学表达式。解释为什么在训练的后半部，权向量的更新只需用到支持向量。

7.5　只考虑支持向量机和分属两类的训练样本：

ω_1: $(1, 1)^{\mathrm{T}}$　　$(2, 2)^{\mathrm{T}}$　　$(2, 0)^{\mathrm{T}}$

ω_2: $(0, 0)^{\mathrm{T}}$　　$(1, 0)^{\mathrm{T}}$　　$(0, 1)^{\mathrm{T}}$

在图中作出这 6 个训练点，构造具有最优超平面和最优间隔的权向量，并指出哪些是支持向量。

参考文献

[1] V. Vapnik. *The Nature of statistical learning theory*[M]. Springer, N.Y., 1995. 张学工译. 统计学习理论的本质. 北京：清华大学出版社, 2000.

[2] B. Scholkopf, C. J. C. Butgcs and A. J. Smola. *Advances in kernel methods-support vector learning*[J]. MIT Press, Cambridge, MA, 1999.

[3] D. MacKay. *Introduction to Gaussian processs*. In Neural Networks and Machine Learning(NATO Asi Serics); Ed. Chris Bishop, 1999.

[4] D.Haussller. *Convolution kernels on discrete structures*. Technical Report UCSC-CRIr99-lO, University of California in Santa Cruz.Computcr Science Department, July 1999.

[5] T. Evgeniou, M. Pontil, and T. Poggio. *A unified framework for regularization networks and support vector machines*. Technical Report CBCL Paper #171/AI Memo#1654, Massachusetts Institute of Technology, 1999.

[6] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines*. Cambridge University Press, Cambridge, UK, 2000. 李国正，王猛，曾华军译. 支持向量机导论. 北京：电子工业出版社，2004.

[7] A. Scholkopf, A. J Smola, K. R. Miiller. *Kernel principal component analysis*. In: Advances in Kernel Methods Suport Vector Learning. MIT Press, 1999:327-352.

[8] C. Watkins. *Dynamic alignment kernels*. Technical Report, Royal Holloway, University of Lodon, Computer Science Department, January 1999.

[9] D. Mackay. *Introduction to Gaussian processes*. In: Neural Networks and Machine Learning（NATO Asi Series）, 1999.

[10] T. Evgeniou, M. Pontil and T. Poggio. *A unified framework for regularization networks and support vector machines*. Technical Report, Massachusetts Institute of Technology, 1999.

[11] T. S. Jaskkola and D. Haussler. *Exploiting generative models in discriminative classifiers*. In: advances in Neural Information Processing Systems, 11. MIT Press, 1998.

[12] N. Cristianini, J. SHawwe-Taylor and C.Campbell. *Dynamically adapting kernels in support vector machines*. In: Advances in Neural Information Processing Systems, 11. MIT Press, 1998.

[13] V. Vapnik. *Statistical learning theory*[M]. New York: John Wiley & Sons, 1998.

[14] Steve R. Gunn, Support Vector Machines for Classification and Regression[R], 1998.

[15] B. Scholkopf and A. *Smola. Learning with kernels*. MIT Press, Cambridge, MA, 2002.

[16] Martin Anthony and Norman Biggs. *Computational learning theory*. Cambridge University Press, 1992.

[17] Ralf Herbrich. *Learning kernel classifiers theory and Algorithms*. The MIT Press, 2002.

[18] S. G. Nash and A. Sofer. *Linear and nonlinear programming*. George Mason University, 1996.

[19] R. Fletcher. *Constrained Optimization*. New York:Wiley, 1981.

[20] O.L. Mangasarian. *Nonlinear Programming*. SIAM, University of Wisconsin, 1994.

[21] O. L. Mangasarian and David R. Musicant. *Successive Overrelaxation for Support Vector Machines*. IEEE Transactions on Neural Networks, 10:1032-1037, 1999.

[22] O. L. Mangasarian. *Generalized Support Vector Machines*. In A.Smola, P.Bartlett, B.Scholkopf, and D_Schuurmans, editors, Advances in Large Margin Classifiers, Pages 135-146, Cambridge, MA, 2000. MIT Press.

[23] V. Vapnik, S. Golowich and A. Smola. *Support vector method for function approximation, regression estimation, and signal processing*. In M. Mozer, M. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9, pages 281-287, Cambridge, MA, MIT Press, 1997.

[24] B. Scholkopf, A. Smola, R. C. Williamson et. al. *New Support vector algorithms*. Neural Computation, 2000, 12（5）: 1207-1245.

[25] 张春华. 支持向量机中最优化问题的研究[D]. 中国农业大学, 2004.

[26] Steve R. Gum. *Support Vector machines for classification and Regression*[R]. Facalty of Engineering and Applied Science Department of Electronics and Computer Science of University of Southaprpton, 1998.10.

第 8 章　核函数方法及应用

核函数方法(Kernel Function Methods, KFM)是一类新的机器学习算法，它与统计学习理论(Statistical Learning, SL)和以此为基础的支持向量机(Support Vector Machines, SVM)的研究及发展密不可分。核函数方法的许多性质是在支持向量机的研究中被不断发现并得以推广应用的。在支持向量机中被广泛采用的特征变换方法是核函数方法。在实际应用中，这两种方法常常是一起使用的，首先通过非线性变换手段将训练样本变换到新的特征空间中，在该空间中再使用广义最优分类面对样本进行划分。

在支持向量机的求解计算过程中只需要计算样本间的内积，并不需要知道样本在空间中的具体坐标值。最后得到的超平面，也只涉及支持向量和特征空间中向量间的内积。即只要知道了任何希尔伯特空间中的内积定义，就可以利用支持向量机的求解算法，计算出该空间中的最优(广义)超平面。这一事实虽然简单，但是直到 1992 年才被 Boser, Guyon 和 Vapnik 等人发现[1]。

核函数方法正是基于这个事实，通过定义变换后空间中的内积，实现人们所需要的某种非线性变换，即

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{y}) \rangle \tag{8.1}$$

称 $\boldsymbol{\phi}: \boldsymbol{x} \to \boldsymbol{\phi}(\boldsymbol{x})$ 为核函数 $k(\boldsymbol{x}, \boldsymbol{y})$ 导出的特征变换。$\boldsymbol{x}$ 为输入空间，$\boldsymbol{\phi}(\boldsymbol{x})$ 为特征空间。$k(\boldsymbol{x}, \boldsymbol{y})$ 作为定义在某个希尔伯特空间上的内积，它首先是实对称的。但是一个对称的二元函数并非一定对应着某个希尔伯特空间上的内积，它还要满足 Mercer 条件——Mercer 定理。

【Mercer 定理】在 L_2 范数下对称函数 $k(\boldsymbol{x}, \boldsymbol{y})$ 能以正的系数 $a_k > 0$ 展开成

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{\infty} a_k \psi_k(\boldsymbol{x}) \psi_k(\boldsymbol{y}) \tag{8.2}$$

即 $k(\boldsymbol{x}, \boldsymbol{y})$ 描述了某个特征空间中的一个内积的充分必要条件是，对使得 $\int g^2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} < \infty$ 的所有函数 $g \neq 0$，条件

$$\iint k(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} > 0 \tag{8.3}$$

成立。条件式(8.3)就是 Mercer 条件。

给定某个核函数 $k(\boldsymbol{x}, \boldsymbol{x})$，就定义了一个相应的重投影核希尔伯特空间(Reproducing Kernel Hilbert Space, RKHS)。该空间中的基本元素是如式(8.4)所示的连续函数：

$$H = \left\{ f(\boldsymbol{x}) : f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) \right\} \tag{8.4}$$

可以证明，式(8.4)所定义的函数空间是一个希尔伯特空间。与常用的 L_2 空间相比，它是由一些更加光滑的函数构成的。在这个空间中，两个元素(函数)的内积定义如下：

$$\langle f, g\rangle = \sum_i \sum_j \alpha_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{8.5}$$

重投影，是指该空间中的内积具有如下性质：

(1) $\langle k(\cdot, \boldsymbol{x}), f\rangle = f(\boldsymbol{x})$

(2) $\langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}')\rangle = k(\boldsymbol{x}, \boldsymbol{x}')$

由一个核函数可以导出某个相应的非线性变换。即同一个核函数可以对应很多非线性变换，下面的变换是其中之一：

$$\Phi : \boldsymbol{x} \to k(\cdot, \boldsymbol{x}) \tag{8.6}$$

该变换是将输入空间中的一个元素映射到核函数导出的 RKHS 中的一个元素(连续函数)。于是根据 RKHS 的重投影性质，有

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})\rangle = \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{y})\rangle = k(\boldsymbol{x}, \boldsymbol{y}) \tag{8.7}$$

核函数方法的实质是通过定义特征变换后样本在特征空间中的内积来实现一种特征变换。它关心的是结果，而不是实现结果所采用的具体方式。

支持向量机通过引入核函数，有效地解决了模式分类中的线性不可分问题。而核函数的应用领域不仅限于支持向量机这一领域。事实上，它正逐步成为一种重要的、将非线性问题线性化的普适方法。比如 Scholkopf[2]将核函数方法用于主成分分析(PCA)，提出了基于核的主成分分析方法(KPCA)，从而将原本用于线性相关分析的 PCA 方法扩展到了非线性相关分析的领域。F. R. Bach 等人将核函数方法用于独立成分分析(ICA)，使得用于分解独立信号线性叠加的 ICA 方法也可以用于独立信号的非线性混迭[3]。总之，解决线性问题时的许多技术手段，都可以尝试通过核函数方法扩展到非线性领域。

8.1 核函数的可分性条件[4]

核函数是利用支持向量机解决线性不可分问题时引入的一种非线性变换手段。基本思想是通过非线性变换，使样本在变换之后的特征空间中变得线性可分。然后利用线性可分时构造最优超平面的方法，在特征空间中实现最优超平面的求解。

核函数的可分性，是指对给定的训练样本，核函数导出的特征变换能否将这些样本在特征空间中线性分开的能力。目前，实际中经常使用的核函数有三种类型：多项式核函数、RBF 核函数及多层感知器核函数。但事实上，当人们在使用这些核函数前，往往并不知道样本是否真的在特征空间是线性可分的。不过许多经验说明，如果选择高斯 RBF 核函数，那么只要选择合适的参数，样本几乎总能被线性分开。

8.1.1 输入空间中样本点线性可分的判别条件

判断样本点在输入空间中是否是线性可分的，通常要先确定对样本点进行分类的分类面参数的搜索算法，然后针对所给的具体样本点进行迭代搜索。如果样本点线性不可分，则要么迭代算法不会终结，即算法不收敛，如求解线性感知器时通常采用的梯度下降算法；要么不论样本线性可分与否，算法都将得到一个分类面，该分类面将使得两类样本中被错分的个数最少，如解线性不等式组的共轭梯度法和搜索算法[4, 34]。

1．线性可分性的定义

假设已知一组容量为 n 的样本集，分别来自两个不同的类。则这个集合可以记为 $T\{(\boldsymbol{x}^k, y^k), k=1,\mathrm{L}\ ,n\}$，其中 $\boldsymbol{x}^k \in R^{\mathrm{m}}$，$y^k \in (-1,1)$。如果有一个线性分类器能够把每个样本正确分类，即如果存在 $\boldsymbol{w}_i, i=1,\mathrm{L}\ ,m$，使得

$$\sum_{i=1}^{m}(\boldsymbol{w}_i \boldsymbol{x}_i^k + b)y^k > 0, \qquad k=1,\mathrm{L}\ ,n \tag{8.8}$$

则称这组样本集为线性可分的；否则称为线性不可分的。反过来，如果样本集是线性可分的，则必然存在一个权向量 $\boldsymbol{w}=(w_0, w_1,\mathrm{L}\ ,w_m)$，使式(8.8)成立。

若记

$$s(k)=(x_0^k, x_1^k,\mathrm{L}\ ,x_m^k)^{\mathrm{T}}, \qquad x_0^k=1 \tag{8.9}$$

$$c^k = s(k)y^k, \quad k=1,\mathrm{L}\ ,n \tag{8.10}$$

$$\boldsymbol{c}=(c^1, c^2,\mathrm{L}\ ,c^n) \tag{8.11}$$

则 T 线性可分的充要条件是，存在某向量 $\boldsymbol{w}$，使得

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{w} > 0 \tag{8.12}$$

2．可分性判别的充要条件

【定理 8.1】 [35] T 线性不可分，当且仅当坐标原点 O 属于 $\{c^i\}_{i=1,\mathrm{L}\ ,n}$ 的凸包。

【推论 8.1】 样本 T 线性不可分的充要条件是下面的方程组有非负解：

$$\boldsymbol{A}\cdot\boldsymbol{X}=\boldsymbol{b} \tag{8.13}$$

其中 $\boldsymbol{A}=\begin{bmatrix}1 & \mathrm{L} & 1\\ c^1 & \mathrm{L} & c^1\end{bmatrix}$，$\boldsymbol{b}=(1,0,\mathrm{L}\ ,0)^{\mathrm{T}}$。

证明：根据二择一定理，式(8.13)存在非负解与存在向量 $\boldsymbol{v}$，使得

$$\boldsymbol{A}\boldsymbol{v} \geq 0, \quad \boldsymbol{b}^{\mathrm{T}}\boldsymbol{v} < 0 \tag{8.14}$$

这两者必居其一。若存在 $\boldsymbol{w}$，使得 $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{w}>0$，则令 $\boldsymbol{v}=(-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{w}, \boldsymbol{w}^{\mathrm{T}})^{\mathrm{T}}$。于是 $\boldsymbol{v}$ 即可满足式(8.14)，表明如果样本点 T 线性可分，则式(8.13)无非负解。反之，若式(8.12)不成立，即样本点 T 线性不可分，则式(8.14)也无法成立。因此式(8.13)必有非负解。

注意推论 8.1 只是将定理 8.1 用代数方程的形式重新进行了表示，并没有实质性的新内容。但是这种形式上的改变带来的直接好处是，可以利用线性代数中的二择一定理，很简洁地得到证明。

8.1.2　特征空间中样本点线性可分的判别条件

特征空间是指经过特征变换后样本点所在的空间。对于传统的特征变换而言，判别样本点的可分性与样本是在特征空间还是在输入空间并没有本质的不同。因为对任意的一个输入

空间中的样本点 $\boldsymbol{x}$，都可以通过特征变换的显式表达式 $\boldsymbol{\Phi}(\cdot)$ 得到特征空间中所对应的镜像点坐标 $\boldsymbol{\Phi}(\boldsymbol{x})$。因此判断样本点在特征空间中的可分性完全可以利用上一节的结论。但是如果一个特征变换是由核函数导出的，问题就有些棘手了。因为核函数只能给出空间中各点之间的内积，相当于只是知道各点之间的距离和夹角，而并不知道各点具体的坐标值。

1．一个充要条件

首先，根据判别样本点在输入空间中线性可分性的充要条件，推出基于核函数导出的特征变换下特征空间中样本点线性可分的一个充要条件。

设核函数 $k(\cdot,\cdot)$ 导出的非线性映射为 φ: $\boldsymbol{x}\rightarrow\varphi(\boldsymbol{x})$。则在特征空间 $\varphi(\boldsymbol{x})$ 中，样本 T 线性不可分的充要条件仍为式(8.13)具有非负解。但此时

$$\boldsymbol{A}=\begin{bmatrix} 1 & \mathrm{L} & 1 \\ y_1 & \mathrm{L} & y_1 \\ y_1\boldsymbol{\phi}(\boldsymbol{x}_1) & \mathrm{L} & y_n\boldsymbol{\phi}_n(\boldsymbol{x}_n) \end{bmatrix}$$

以下如无特别说明，$\boldsymbol{A}$ 的定义均为此式所示。相应地有

$$\boldsymbol{C}=\begin{bmatrix} y_1 & \mathrm{L} & y_1 \\ y_1\boldsymbol{\phi}(\boldsymbol{x}_1) & \mathrm{L} & y_n\boldsymbol{\phi}_n(\boldsymbol{x}_n) \end{bmatrix}$$

进一步，令

$$\boldsymbol{H}^0=\boldsymbol{C}^{\mathrm{T}}\cdot\boldsymbol{C}=(h_{ij}^0)_{n\times n},\quad h_{ij}^0=k(\boldsymbol{x}_i,\boldsymbol{x}_j)\cdot y_i\cdot y_j+y_i\cdot y_j \tag{8.15}$$

$$\boldsymbol{H}^1=\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{A}=(h_{ij}^1)_{n\times n},\quad h_{ij}^1=k(\boldsymbol{x}_i,\boldsymbol{x}_j)\cdot y_i\cdot y_j+y_i\cdot y_j+1 \tag{8.16}$$

其中，$i,j=1,\mathrm{L}\ ,n$。

【引理 8.1】 式 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{X}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 有解的充要条件是式 $\boldsymbol{AX}=\boldsymbol{b}$ 有解。

证明： 显然方程 $\boldsymbol{AX}=\boldsymbol{b}$ 的任何一个解 $\boldsymbol{x}_0$，必然也是方程 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{X}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 的解。可证其逆命题：由方程 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{X}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 的任意一个解 $\boldsymbol{x}_0$，可以构造出方程 $\boldsymbol{AX}=\boldsymbol{b}$ 的一个解。

首先，$(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{X}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 的任意一个解 $\boldsymbol{x}_0$，可以表示为 $\boldsymbol{x}_0=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{+}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$。这里 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{+}$ 表示矩阵 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$ 的广义逆。由于方程 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{X}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 未必只有唯一解，因此这里采用了广义逆。注意到 $\boldsymbol{A}$ 为 $(m+2)\times n$ 阶矩阵，其中 m 是非线性特征映射 φ 作用下的样本所在特征空间的维数，n 为样本点个数。若 $(m+2)\leqslant n$，将 $\boldsymbol{A}$ 进行奇异值分解可得

$$\boldsymbol{A}=\boldsymbol{S}_{(m+2)\times(m+2)}\begin{bmatrix} \lambda_1 & & & \\ & \mathrm{O} & & \\ & & \lambda_r & \\ & & & 0 \end{bmatrix}\boldsymbol{U}_{n\times n}$$

于是

$$\begin{aligned}\boldsymbol{A}\boldsymbol{x}_0 &= \boldsymbol{A}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}) + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \\ &= \boldsymbol{S}\begin{bmatrix}\lambda_1 & & & \\ & \mathrm{O} & & \\ & & \lambda_r & \\ & & & 0\end{bmatrix}\boldsymbol{U}\cdot\boldsymbol{U}^{\mathrm{T}}\begin{bmatrix}\lambda_1^{-2} & & & \\ & \mathrm{O} & & \\ & & \lambda_r^{-2} & \\ & & & 0\end{bmatrix}\boldsymbol{U}\cdot\boldsymbol{U}^{\mathrm{T}}\begin{bmatrix}\lambda_1 & & & \\ & \mathrm{O} & & \\ & & \lambda_r & \\ & & & 0\end{bmatrix}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{b} \\ &= \boldsymbol{S}\begin{bmatrix}I & \\ & 0\end{bmatrix}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{b}\end{aligned}$$

记 $\boldsymbol{S} = (\boldsymbol{\alpha}_{(m+2)r} \quad \boldsymbol{\beta}_{(m+2)(m+2-r)})$，则

$$\boldsymbol{A}\boldsymbol{x}_0 = (\boldsymbol{\alpha} \quad \boldsymbol{\beta})\cdot\begin{bmatrix}I_{r\times r} & \\ & 0\end{bmatrix}(\boldsymbol{\alpha} \quad \boldsymbol{\beta})^{\mathrm{T}}\cdot\boldsymbol{b} = \begin{bmatrix}I_{r\times r} & \\ & 0\end{bmatrix}\cdot\boldsymbol{b} = (b_1, \mathrm{L}\ , b_r, 0, \mathrm{L}\ , 0)^{\mathrm{T}}$$

根据 $\boldsymbol{A}$ 的表达式，存在 $r \geqslant 1$。注意到 $\boldsymbol{b} = (1, 0, \mathrm{L}\ , 0)^{\mathrm{T}}$，命题成立。当 $(m+2) > n$ 时，证明完全类似。

【定理 8.2】 样本点 T 在核函数导出的特征映射下线性不可分的充要条件是式(8.17)存在非负解：

$$\boldsymbol{H}^{\mathrm{I}}\boldsymbol{X} = = (1, \mathrm{L}\ , 1)^{\mathrm{T}} \tag{8.17}$$

证明：直接将 $\boldsymbol{A}, \boldsymbol{b}$ 的表达式代入引理 8.1 即可。

定理 8.2 表明通过分析式(8.17)的解的分布，可以判断样本点在特征空间中是否线性可分：如果式(8.17)有且仅有一个解，则样本的可分性取决于这个解向量中是否含有负分量；如果方程组无解，则样本点必然线性可分。如果式(8.17)有无穷多解，此时判断方程是否含有非负解则会相当麻烦。目前人们尚未找到有效的判别方法。事实上，训练支持向量机的过程就是在这无穷多解中选择一个最优解的过程。所以这个定理的价值将更多地体现在理论分析中。对实际给定的样本点进行线性可分性的判断时，使用这个定理并不方便。

2．一个实用的充分条件

鉴于上面给出的充要条件使用起来不太方便，本小节将给出一个使用起来稍微方便一些的充分条件。基本思想是，既然样本点是否线性可分取决于式(8.17)是否存在非负解，那么当式(8.17)无解时(自然也就不会有非负解)，样本点线性可分。

首先注意到如下事实：

(1)对任意矩阵 $\boldsymbol{A}$，有 $\mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A}^{\mathrm{T}}) = \mathrm{rank}(\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A}\cdot\boldsymbol{A}^{\mathrm{T}})$。

(2)如果矩阵 $\boldsymbol{C}$ 列满秩，亦即 $\mathrm{rank}(\boldsymbol{C}) = n$，则必存在 $\boldsymbol{W}$，使得 $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{W} > 0$。即此时样本点线性可分。

由 $\boldsymbol{C}$ 列满秩可知如下方程组必存在解：

$$\boldsymbol{C}^{\mathrm{T}}\cdot\boldsymbol{W} = \begin{bmatrix}1 \\ \mathrm{M} \\ 1\end{bmatrix}_{n\times 1}$$

因此存在 $\boldsymbol{W}$，使得 $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{W} > 0$。

【定理 8.3】 当 $\mathrm{rank}(\boldsymbol{H}^0)=\mathrm{rank}(\boldsymbol{H}^1)$ 时，样本点线性可分。

证明： 记 $\tilde{\boldsymbol{A}}=(\boldsymbol{A}\ \ \boldsymbol{b})$，则式(8.13)无解的充要条件是

$$\mathrm{rank}(\boldsymbol{A})\neq\mathrm{rank}(\tilde{\boldsymbol{A}})$$

因为 $\boldsymbol{A}\sim\begin{bmatrix}1\\ \boldsymbol{C}\end{bmatrix}$，$\tilde{\boldsymbol{A}}\sim\begin{bmatrix}0 & 1\\ \boldsymbol{C} & 0\end{bmatrix}$（~表示两个矩阵相似），从而有

$$\mathrm{rank}(\tilde{\boldsymbol{A}})=\mathrm{rank}(\boldsymbol{C})+1$$

又 $\mathrm{rank}(\boldsymbol{A})=\mathrm{rank}(\boldsymbol{C})$ 或 $\mathrm{rank}(\boldsymbol{C})+1$，因此式(8.13)无解的充要条件即为

$$\mathrm{rank}(\boldsymbol{A})=\mathrm{rank}(\boldsymbol{C})$$

注意到

$$\mathrm{rank}(\boldsymbol{H}^0)=\mathrm{rank}(\boldsymbol{C}^{\mathrm{T}}\cdot\boldsymbol{C})=\mathrm{rank}(\boldsymbol{C}),\quad \mathrm{rank}(\boldsymbol{H}^1)=\mathrm{rank}(\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{A})=\mathrm{rank}(\boldsymbol{A})$$

于是当 $\mathrm{rank}(\boldsymbol{H}^0)=\mathrm{rank}(\boldsymbol{H}^1)$ 时，式(8.13)无解。又根据引理8.1可知式(8.17)此时也无解，因此此时样本点线性可分。

【例 8.1】 证明 XOR 问题在输入空间线性不可分，选用二阶多项式核函数 $k(\boldsymbol{x},\boldsymbol{y})=(1+\boldsymbol{x}\cdot\boldsymbol{y})^2$ 后，在此核函数导出的非线性映射下，XOR 问题线性可分。

证明：设所给四个样本分别为

$$\boldsymbol{x}=\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\\ 1 & 1\end{bmatrix},\qquad \boldsymbol{y}=(1,1,-1,-1)^{\mathrm{T}}$$

如果不做任何特征变换，则相应的核函数为 $k(\boldsymbol{x},\boldsymbol{y})=\boldsymbol{x}\cdot\boldsymbol{y}$。于是

$$\boldsymbol{H}^0=\begin{bmatrix}2 & 1 & -1 & -2\\ 1 & 2 & -1 & -2\\ -1 & -1 & 1 & 1\\ -2 & -2 & 1 & 3\end{bmatrix},\qquad \boldsymbol{H}^1=\begin{bmatrix}3 & 2 & 0 & -1\\ 2 & 3 & 0 & -1\\ 0 & 0 & 2 & 2\\ -1 & -1 & 2 & 4\end{bmatrix}$$

由于 $\mathrm{rank}(\boldsymbol{H}^0)=3$，$\mathrm{rank}(\boldsymbol{H}^1)=4$，故式 $\boldsymbol{H}^1\boldsymbol{x}=(1\ 1\ 1\ 1)^{\mathrm{T}}$ 存在唯一解，解之得 $\boldsymbol{x}=(0.25\ \ 0.25\ \ 0.25\ \ 0.25)^{\mathrm{T}}$，该解的各个分量均为正，因此样本线性不可分。

如果选用二阶多项式核函数，则

$$\boldsymbol{H}^0=\begin{bmatrix}5 & 2 & -2 & -5\\ 2 & 5 & -2 & -5\\ -2 & -2 & 2 & 2\\ -5 & -5 & 2 & 10\end{bmatrix},\qquad \boldsymbol{H}^1=\begin{bmatrix}6 & 3 & -1 & -4\\ 3 & 6 & -1 & -4\\ -1 & -1 & 3 & 3\\ -4 & -4 & 3 & 11\end{bmatrix}$$

由于 $\mathrm{rank}(\boldsymbol{H}^0)=\mathrm{rank}(\boldsymbol{H}^1)=4$，所以样本点在此特征映射下线性可分。图8.1是二阶多项式核函数作用下所形成的分界面[42]。

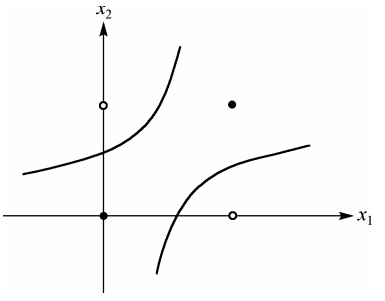


图 8.1 二阶多项式核函数求解 XOR 问题

8.2　核函数的参数确定

构造出一个具有良好性能的支持向量机，模型选择是关键。模型选择是指如何针对所给的训练样本，确定合适的核函数。模型选择包括两部分工作：一是核函数类型的选择，二是确定核函数类型后相关参数的选择。目前，核函数类型基本还是凭经验选定。选定核函数后，再进行相关参数的确定。

核函数的选择是支持向量机理论研究的一个核心问题。研究的目标是能够实现针对具体应用背景及给定的样本集合构造合适的核函数。但是目前，还没有一种针对具体问题构造出合适的核函数的有效方法。在实际使用中被广泛使用的仍然是下面的三种核函数，其中又尤其以 RBF 核函数使用得最广：

(1) 径向基(Radial Basis Function, RBF)核函数

$$K(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{\frac{-\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}} \tag{8.18}$$

(2) 多项式核函数

$$K(\boldsymbol{x}, \boldsymbol{y}) = (1 + \boldsymbol{x} \cdot \boldsymbol{y})^p, \qquad p = 1, 2, \mathrm{L} \tag{8.19}$$

(3) Sigmoid 核函数

$$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(b(\boldsymbol{x} \cdot \boldsymbol{y}) - c) \tag{8.20}$$

其中 b 和 c 为常数。

模型选择的基本过程是：首先确定一个有效评估支持向量机期望(真实)风险的指标，然后找到该评估指标与核函数之间的关系，最后通过解参数优化问题得到一个在所给评估指标下的最佳核函数。

如何尽可能准确地评估支持向量机的期望(真实)风险，是统计学习理论所要解决的核心问题之一。

经典支持向量机实际上是利用两类样本间的间隔作为 SVM 泛化性能的指标。但是这个指标存在严重的问题。首先是它的精确性。间隔其实是对风险估计的一个非常粗糙的上界，这使得 SVM 的性能与间隔的大小不成严格的正比关系。另外，当引入核函数后，不同的核函数会导出不同的特征空间。由于不同空间下的间隔值没有可比性，所以仅凭间隔值的大小并不能对选择哪一种核函数做出有效的判断。这就要求人们寻找到关于风险的更精确估计。Vapnik 等人首先提出利用 R^2 / M^2 作为 SVM 性能的估计[7,36]，其中 R^2 为特征空间中包含所有训练样本的最小球的半径。注意 R^2 / M^2 没有了物理量纲，因此该指标用于衡量不同核函数导出的特征空间中 SVM 的性能比单纯使用间隔要更合理一些。

目前用于模型选择的风险估计指标的理论基础是 A. Luntz 和 V. Brailovsky 关于泛化误差估计的定理[7]：

【定理 8.5】　$EP_{\mathrm{err}}^{n-1} = \dfrac{1}{n} E(L(x_1, y_1, \mathrm{L}\ , x_n, y_n))$

其中 EP_{err}^{n-1} 是采用 $n-1$ 个样本来训练分类器所得到的错误率的期望值。$L(x_1, y_1, \mathrm{L}\ , x_n, y_n)$ 是针对 n 个训练样本，每次选择其中的 $n-1$ 个进行分类器的训练，然后用该分类器判断剩下的一个样本，将这一过程重复 n 次后，得到总的错误个数。这种估计方法被称为留一法(Leave-One-Out, LOO)。

定理 8.5 表明，采用留一法得到的错误率估计是分类器真实错误率的一个无偏估计。但是仅根据这个定理对 SVM 进行性能估计是不大可行的。因为用留一法进行估计就必须根据样本构造出 n 个支持向量机。当训练样本特别多时，计算量会特别大。因此在实际问题中，人们往往采用 k-遍交叉验证的办法对 LOO 过程进行近似估计。k-遍交叉验证是指将训练样本划分成 k 个互不相交的子集 $S_1, S_2, \mathrm{L}\ , S_k$。每个子集的元素个数大致相等。训练和测试进行 k 次。在第 i 次，S_i 用做测试集，其余的子集都用于训练分类器。也就是说，第一次迭代的分类法在子集 $S_2, \mathrm{L}\ , S_k$ 上训练，而用 S_1 做测试；第二次，分类器在子集 $S_1, S_3, \mathrm{L}\ , S_k$ 上训练，而在 S_2 上测试；依此类推。错误率估计就是 k 次错误分类数的总和除以初始训练样本的总数。可见留一法也可以视为 k-遍交叉验证的极端情形：$k = n$。

虽然 k-遍交叉验证相比留一法，计算量已经减少了不少，而且目前一些支持向量机软件如 Libsvm 也确实是利用该方法确定参数的，但是人们对这种近乎蒙特卡罗式的方法始终不满意，所以进一步寻找错误率估计的更有效方法一直没有停止过。从目前已经取得一些结果的解决思路来看，基本上都是试图通过其他手段估计出 $L(x_1, y_1, \mathrm{L}\ , x_n, y_n)$ 的一个尽可能小的上界。

假设 $\psi(\cdot)$ 是一个阶梯函数，f^0 为利用所有训练样本得到的判别函数，f^i 为去掉第 i 个样本后得到的判别函数，则

$$L(x_1, y_1, \mathrm{L}\ , x_n, y_n) = \sum_{i=1}^{n} \psi(-y_i f^i(x_i)) = \sum_{i=1}^{n} \psi(-y_i f^0(x_i) + y_i (f^0(x_i) - f^i(x_i))) \tag{8.21}$$

其中，$\psi(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{其他} \end{cases}$ 称为阶梯函数。

设 U_i 为 $y_i(f^0(x_i) - f^i(x_i))$ 的一个上界，对于硬分类而言(即样本线性可分)，又有 $y_i f^0(x_i) \geqslant 1$，从而 $L(x_1, y_1, \mathrm{L}\ , x_n, y_n)$ 的上界估计为

$$L(x_1, y_1, \mathrm{L}\ , x_n, y_n) \leqslant \sum_{i=1}^{n} \psi(U_i - 1) \tag{8.22}$$

式(8.22)可以表示目前多数的风险估计。

8.3 核函数的构造方法

8.3.1 基于特征变换的核函数构造

核函数是作为一种非线性映射的隐式表达方法而提出的。这种隐式表达方法给分析映射的性质带来了很多困难。反之，在已知非线性映射的情况下，构造与之对应的核函数，则非常容易。正如核函数所要表达的含义：

$$k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{y}) \tag{8.23}$$

因此任何一种特征提取时所构造的非线性变换 $\boldsymbol{\phi}(\cdot)$，都可以通过式(8.23)来实现相应的核函数的构造。有下面的结论：

【定理 8.6】 任何形如式(8.23)的对称函数 $k(\boldsymbol{x}, \boldsymbol{y})$，均满足 Mercer 条件，即对使得

$$\int g^2(u)\mathrm{d}u < \infty \tag{8.24}$$

的所有 $g \neq 0$，条件

$$\iint k(u,v)g(u)g(v)\mathrm{d}u\mathrm{d}v \geq 0 \tag{8.25}$$

成立。此即表明 $k(\boldsymbol{x}, \boldsymbol{y})$ 为核函数。

证明：

$$\iint k(u,v)g(u)g(v)\mathrm{d}u\mathrm{d}v = \iint \phi(u)\phi(v)g(u)g(v)\mathrm{d}u\mathrm{d}v = \iint (\phi(u)g(u))\cdot(\phi(v)g(v))\mathrm{d}u\mathrm{d}v$$

$$= \left(\int \phi(u)g(u)\mathrm{d}u\right)\left(\int \phi(v)g(v)\mathrm{d}v\right) = \left(\int \phi(u)g(u)\mathrm{d}u\right)^2 \geq 0$$

用这种方法构造核函数，存在的最大的问题是：核函数性能的好坏，直接取决于特征变换的好坏。但是对很多实际问题，找到合适的特征变换往往很难。它要求人们对事物的本质了解得非常清楚。如果揭示事物本质特征的变换被找到，不论是做分类也好，做函数逼近也好，应该都不会存在什么问题。

事实上，寻找特征变换是比构造核函数更加困难的事情。所以目前用特征变换构造核函数，主要是为了在变换后的空间中使用支持向量机。但是，并不是说用这种方法构造核函数一点价值也没有。通过这种方法，能够将传统的特征变换纳入到核函数的框架下进行统一讨论，以便人们进一步看清楚问题的本质。下面用特征变换方法分析采用支持向量机解决两类判别问题时，用主成分分析(PCA)进行特征选择的作用。

设输入的 n 个训练样本为 $\boldsymbol{x}_i, i = 1, \mathrm{L}\ , n$。如果首先进行主成分分析，将得到这 n 个训练样本在 r 个主分量上重投影的坐标变换矩阵 $\boldsymbol{A}_{n\times r}$，且 $\boldsymbol{A}_{n\times r}$ 具有如下性质：

$$\boldsymbol{A}\cdot\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{I}_{n\times n} \tag{8.26}$$

如果选择形如 $k(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}^{\mathrm{T}}\cdot\boldsymbol{y})$ 的核函数，则进行主成分分析之后的核函数为

$$k(\boldsymbol{x}, \boldsymbol{y}) = f((\boldsymbol{A}^{\mathrm{T}}\cdot\boldsymbol{x})^{\mathrm{T}}\cdot(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y})) = f(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\cdot\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}) = f(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})$$

可见，如果选用这种类型的核函数，那么采用 KL 变换的方法提取训练样本的主成分无助于分类精度的提高。这和人们以往的经验是有出入的。当样本的特征过多时，要进行降维处理以尽可能减少特征维数。而在众多的降维方法中，主成分分析又是运用得最为普遍的一种方法。现在看到，如果使用这种类型的核函数构造 SVM，那么首先提取主成分再送入 SVM 进行训练的方法就显得有些多余。

注意到形如 $k(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{y})$ 和的核函数并不在少数，如多项式核函数和 Sigmoid 核函数均属于此种类型。所以本节的结论虽然简单，但是很有借鉴意义。它说明了将特征变换纳入到核函数框架下讨论问题的好处。

8.3.2　利用Mercer核函数的性质组合核函数

利用 Mercer 核函数的性质构造核函数，就是利用核函数集合在某些运算下闭合的性质，组合现有的一些核函数而构造出新的核函数。

如果 $k_1(\boldsymbol{x}, \boldsymbol{y})$ 和 $k_2(\boldsymbol{x}, \boldsymbol{y})$ 是满足条件式(8.3)的核函数，即 Mercer 核函数，则下面这些核函数也是 Mercer 核函数[8]：

(1) $k(\boldsymbol{x}, \boldsymbol{y}) = ak_1(\boldsymbol{x}, \boldsymbol{y}) = bk_2(\boldsymbol{x}, \boldsymbol{y}),\ \forall a, b \in R^+$。

(2) $k = k_1(\boldsymbol{x}, \boldsymbol{y}) \cdot k_2(\boldsymbol{x}, \boldsymbol{y})$。

(3) $k(\boldsymbol{x}, \boldsymbol{y}) = k_1(\phi(\boldsymbol{x}), \phi(\boldsymbol{y}))$，即先进行初步的特征变换，再用核函数作用，最后得到的整个变换仍是一个核函数。

(4) $k(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x})f(\boldsymbol{y})$。

(5) $k(\boldsymbol{x}, \boldsymbol{y}) = k_1(\boldsymbol{x}, \boldsymbol{y}) + k_2(\boldsymbol{x}, \boldsymbol{y})$。

(6) $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{y}$。

其中 $f(\cdot)$ 是 $X \subseteq R^n$ 空间的实函数，$\boldsymbol{B}$ 是 $n \times n$ 对称正半定矩阵。

8.3.3 借助其他领域知识构造核函数

核函数反映了样本在特征空间中的相似程度。所以其他领域中计算样本间相似程度(或距离)的方法，都可以相应地改造成核函数。一方面，这种方法为核函数的构造提供了广阔的素材；另一方面，通过这种方法，也使很多看似不太相关的技术手段被统一到支持向量机的框架下。只有在一个统一的框架下，才更便于了解各个方法的本质特征，分析和比较各个方法之间的优劣。因此这方面的研究工作显得特别有意义。

(1)用协方差函数定义核函数

协方差函数可以视为样本间相似程度的度量。所以如果采用样本的协方差函数来定义核函数，通过适当修改支持向量机中对经验风险误差的计算，支持向量机将等价于高斯过程(一种用于函数逼近的方法)[9]。

在地质统计中，也要经常用到协方差函数。但是由于地质统计中有一些实际困难，比如采样的样本数目比较有限，同时分布还不太均匀等，使得估计样本的协方差比较困难。对此，地质统计学中提出了一些稳定性的假设，然后用估计变异函数的方法来代替对协方差函数的估计。而在很多时候，地质统计学中遇到的困难，同时也是机器学习的研究中需要面对的。很自然地，可以借鉴地质统计学中估计变异函数的方法，来实现针对给定样本的核函数构造。采用这种核函数的支持向量机将等价于地质统计学中的克里金(Kriging)方法[10] (这一点首先由 V. Vapnik[7]提出)。克里金方法所得到的解具有均方误差最小的性质[11]，因此当支持向量机采用变异函数作为核函数时，它也将具有这样一个良好的性质。

(2)用距离函数定义核函数

距离函数可以视为样本间的相异性度量。因此很多跟距离有关的定义，都可以借鉴过来用于核函数的定义。距离的概念是广泛的。它不仅包括两个样本间的距离函数，还包括某个泛函空间中范数的定义。支持向量机本质上也是一种正则化方法[12]。支持向量机中核函数的选择，对应着正则化方法中正则化算子(正则化算子是在一个赋泛线性空间上定义的泛函)的选择。反之，在正则化方法中选择一个正则化算子，也可以用核函数的形式表现出来。

8.4 几种核方法

8.4.1 KPCA的基本思想

对一个给定的非线性映射 $\boldsymbol{\Phi}$，将输入空间 R^n 映射成为特征空间 F：

$$\begin{aligned} \boldsymbol{\Phi}: R^n &\to F \\ \boldsymbol{x} &\to \boldsymbol{\Phi}(\boldsymbol{x}) \end{aligned} \tag{8.27}$$

相应地，一个在 R^n 空间的模式映射成为特征空间中具有更高维数的模式，这时，在 R^n 空间中线性不可分的模式，在映射后的特征空间中可能变得线性可分，或是比在 R^n 空间中更容易分类。KPCA[13]就是一种在特征空间中进行 PCA 的方法。

一般地，很难得到最佳的非线性变换 $\boldsymbol{\Phi}$，同时高维空间甚至是无穷维空间的内积也是难于计算的。但是核方法的出现解决了这个难题，使得显性的非线性变换中不再需要，同时也大大简化了内积的计算。首先将输入样本投影到一个比原来样本空间维数更高的特征空间 F 中，然后计算各阶统计矩，一般来说这一工作量较大。我们现在寻求仅通过对训练样本 $\boldsymbol{x}$ 的点积运算 $(\boldsymbol{\Phi}(\boldsymbol{x}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{x}_j))$ 进行计算，其中 $\boldsymbol{\Phi}$ 是一个非线性映射。这一目的可以通过利用 Mercer 核函数实现[8]，这里 Mercer 核函数 $k(\boldsymbol{x}_i, \boldsymbol{y}_j)$ 计算在空间 F 中的点积，即 $k(\boldsymbol{x}_i, \boldsymbol{y}_j)=(\boldsymbol{\Phi}(\boldsymbol{x}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{x}_j))$。这一思想首先被应用在支持向量机(SVM)中并产生了一种较通用的机器学习算法。

令 $\{\boldsymbol{x}_i\}$ 是输入空间 X 中的 M 个向量，X_j 代表 X 的一个子集，且 $X=\bigcup_{j=1}^{k} X_j$，其中 k 为类别数。假设空间 X 通过下面一个非线性映射 φ 变换到一个希尔伯特空间 F：

$$\begin{aligned} \varphi: X &\to F \\ \boldsymbol{x} &\to \varphi(\boldsymbol{x}) \end{aligned} \tag{8.28}$$

为计算公式简便起见，假设所有的数据已经中心化。这样在特征空间 F 中的协方差矩阵为

$$\boldsymbol{C}=\frac{1}{M}\sum_{j=1}^{M}\varphi(\boldsymbol{x}_j)\varphi(\boldsymbol{x}_j)^{\mathrm{T}} \tag{8.29}$$

令 $M\times M$ 阶矩阵 $\boldsymbol{K}$ 由元素 $k_{ij}=\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x}_j)$ 构成，则 Kernel PCA 可通过解特征方程

$$M\lambda\alpha=K\alpha \tag{8.30}$$

求得。同样，可以得到空间 F 中的类间散布矩阵为

$$\boldsymbol{B}=\frac{1}{M}\sum_{j=1}^{k} n_j\bar{\boldsymbol{\varphi}}_j\bar{\boldsymbol{\varphi}}_j^{\mathrm{T}} \tag{8.31}$$

其中，$\bar{\boldsymbol{\varphi}}_j$ 是第 j 类的均值向量，$\bar{\boldsymbol{\varphi}}_j=\frac{1}{n_j}\sum_{i=1}^{n_j}\boldsymbol{\varphi}(x_{ji})$，$x_{ji}$ 是第 j 类的第 i 个元素。同时，空间 F 中的总体散布矩阵

$$\boldsymbol{V}=\frac{1}{M}\sum_{j=1}^{k}\sum_{i=1}^{n_j}\boldsymbol{\varphi}(x_{ji})\boldsymbol{\varphi}(x_{ji})^{\mathrm{T}}$$

由散布矩阵，在空间 F 中的 Fisher 判别函数可定义为

$$J(\boldsymbol{\varphi})=\frac{\boldsymbol{\varphi}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\varphi}}{\boldsymbol{\varphi}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\varphi}}$$

由 Fisher 最优判别分析可知，第一个最优判别向量的求解可以归结为如下广义特征值问题：

$$\lambda \boldsymbol{V}\boldsymbol{v} = \boldsymbol{B}\boldsymbol{v}$$

而对上述问题的求解已有多种方法。KPCA 的基本思想和推导过程可以用图8.2表示出来。

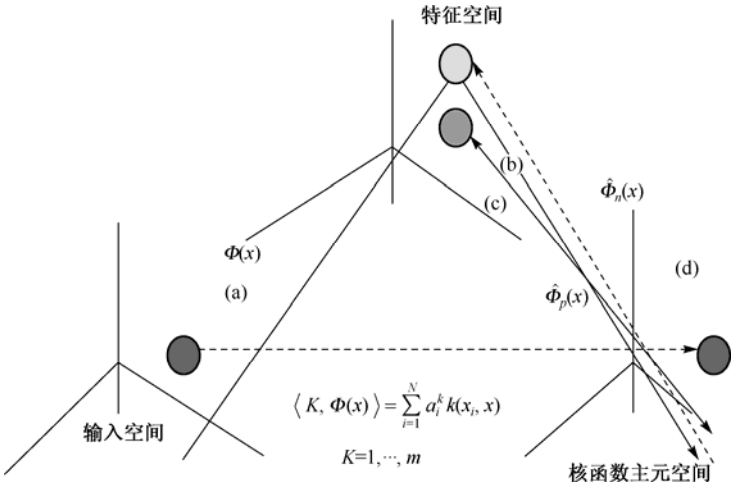


图 8.2 KPCA 示意图

因为特征空间 F 相对于输入空间是非线性的，因此在输入空间投影到主本征向量上的向量就成为非线性。对于 KPCA 重要的是，实际上并没有向 F 空间进行映射，而是在输入空间进行了核函数 k 的计算。

实际上在特征空间 F 中进行的是标准 PCA 计算，所以 KPCA 的性质和 PCA 是一致的。

和线性 PCA 不同，KPCA 方法允许主成分的数量超过输入空间的维数。设观察值 M 大于输入空间的维数 N，则线性 PCA 对 $M\times M$ 点积矩阵进行变换时，至多能找到 N 个非零本征值，等于 $N\times N$ 协方差矩阵的非零本征值；相反，KPCA 可以找到高达 M 个非零本征值，这说明基于 $N\times N$ 协方差矩阵实现 KPCA 是不可能的。

多项式核函数 $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y}+1)^d$ 在 $d=5$ 时，对于 256 维输入空间会产生 10^{10} 维空间。在这样的空间里计算主成分是不可能的。实际上，① KPCA 并不是在整个 F 空间计算本征向量，只是对 F 空间的样本 $\boldsymbol{x}_i$ 的子空间进行计算。② KPCA 不需要直接计算 F 空间向量的点积，而是在输入空间计算核函数；KPCA 和 PCA 计算具有 l 个观察值的 $l\times l$ 点积矩阵的计算量是相同的。估计核函数和点积计算相比，不会使计算复杂度改变；如果 k 很容易计算，如多项式核函数，计算复杂度的改变可以忽略。③ KPCA 虽然比 PCA 增加了计算复杂度，但在设计分类器时会得到补偿。在 KPCA 方法中提取主成分特征后，可以采用线性支持向量机(Support Vector Machine, SVM)构造决策边界；线性 SVM 比非线性 SVM 的速度快很多。这是因为对于 $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y})$，SVM 的决策函数

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \lambda_i k(\boldsymbol{x}_i \boldsymbol{x}_j) + b\right) \tag{8.32}$$

可以表示为权向量表达式

$$f(\boldsymbol{x}) = \operatorname{sgn}\big((\boldsymbol{x}\cdot\boldsymbol{w}) + b\big) \tag{8.33}$$

其中 $\boldsymbol{w}=\sum_{i=1}^{l}\lambda_i\boldsymbol{x}_i$。这样在分类时速度可以很快，从而补偿了主成分特征提取阶段的速度，通过控制主成分特征的数量或减小集合参数 N 使准确率和速度在整个分类器设计过程得到折中。

对上述 KPCA 的描述，以 ORL 数据库中第 30 人为例，将其输入向量经过核空间映射后的向量以图 8.3 表示出来，其中“×”符号表示为输入向量，而“+”符号为经过核空间映射后重构的可以用来作为识别的标准向量。

采用多项式核函数 $d=2$、分别采用 5 个训练样本和 5 个测试样本、取不同主成分个数时的识别率如表8.1 所示。图8.4 为根据表 8.1 得到的识别率。

表 8.1　$d=2$、训练样本和测试样本分别为 5 时的识别率

主成分个数	10	20	30	40	50	60	70	80	90	100
识别率(%)	57	79	87	915	94	945	965	97	985	99
主成分个数	110	120	130	140	150	160	170	180	190	200
识别率(%)	99	99	99	99	99	99	0.99	0.99	0.99	0.99

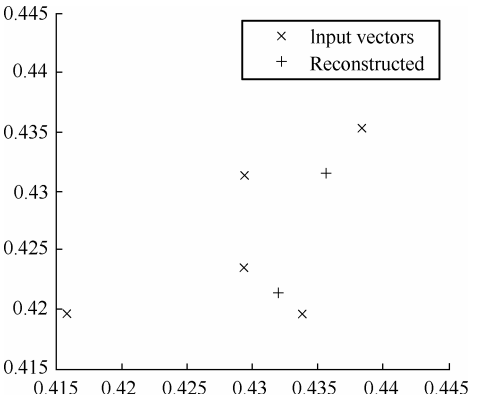


图 8.3　输入向量和经过 KPCA 重构向量

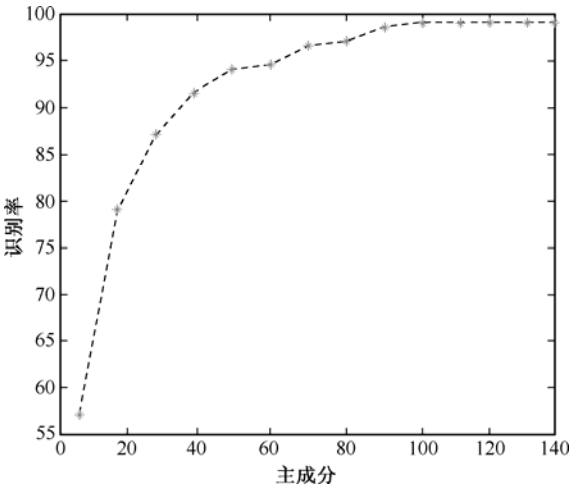


图 8.4　$d=2$、训练样本和识别样本分别是 5 的识别率

8.4.2　基于类内散布的最优kernel PCA展开方法

运用核方法，可以将基于类内散布的最优 PCA 展开方法推广成为基于类内散布的最优 kernel PCA 展开方法。

基于类内散布的最优kernel PCA展开方法旨在抽取非线性变换下包含最佳判别信息的向量集。方法的过程如下：

1．运用 KPCA 的方法将原始样本空间映射到 m 维的特征空间，$m = M-1$, M 为原始空间训练样本的个数。

2．在特征空间 R^m 中，求得类内散布矩阵 $\boldsymbol{S}_w$，计算 $\boldsymbol{S}_w$ 的特征值 $\lambda_1 \leq \mathrm{L} \leq \lambda_m$ 和对应的特征向量 $\xi_1,\mathrm{L}\ ,\xi_m$。选取前 d 个特征向量，令 $V=(\xi_1,\mathrm{L}\ ,\xi_d)$。

如果在求解最佳判别向量集的过程中，仅考虑由总体散布矩阵 $\boldsymbol{S}_t$ 的非零特征值构成的

特征子空间，不但不会影响最佳判别向量集的求解，而且会减少运算量。KFD 可以视为特征空间中的 FDA，同样先进行 KPCA 运算，再求取特征空间中类内散布矩阵从小到大特征值所对应的特征向量是一种行之有效的方法。相比于其他 KFD 方法，基于类内散布的最优 kernel PCA 展开方法不仅大大减少了运算量，而且避免了由于特征空间中的中心化给 KFD 带来的问题，使得更容易抓住 KFD 的本质，简化了 KFD。同时，基于类内散布的最优 kernel PCA 展开方法不但未求取类间散布矩阵，而且考虑了类内散布矩阵的核对求取最佳判别向量集的作用。这是其他 KFD 方法所没有考虑到的。

8.4.3 融合先验类别信息的非线性主元分析算法

Kernel 提供了一种处理非线性问题的算法。将 $\boldsymbol{x}$ 投影到非常高维的空间 F，使其分散性更好，在高维 F 空间 Kernel 运算代替了点积运算，主要有空间维数压缩和优化算法两种形式。空间维数压缩包括如 KPCA, KPLS, KFD 等变换及变体形式。虽然要向非常高的空间进行映射，求取特征向量等，但这些运算都可以通过 Kernel 函数的运算直接进行求解，从而使这个映射空间变成完全透明的。并且能够提取比变量数更多的主成分。

优化算法：一般情况下优化问题都含有系数与基函数的点乘运算，即 $\boldsymbol{w}\cdot\varphi(\boldsymbol{x})$ 可以转化为 Kernel 函数的形式，这样就变成以 Kernel 函数表达的优化目标。例如支持向量机就是其中的一种典型应用。

1．PKPCA 算法[15]

主成分分析(PCA)是从高维数据提取其结构的有效方式，它舍弃了相关性不强的信息。PKPCA (Priori Kernel Principal Component Analysis) 是综合了 PCA 对总体方差和 Fisher 判据对类间差、类内差的分析基础上，将类别信息融入 PCA 算法中的一种更易于分类的新算法。PKPCA 首先引入类别信息到 PCA 中，对高维空间进行映射，在高维空间重构样本库。然后进行 Kernel 变换，得到提取主元的表达式。

(1) 先验类别信息的融入

考虑 k 个样本集 $X_1,\mathrm{L}\ ,X_k$，每个样本集中有 N_i, $i=1,\mathrm{L}\ ,k$ 个样本，且 $\sum N_i=N$。相应的均值和协方差阵分别为 $\boldsymbol{m}_1,\mathrm{L}\ ,\boldsymbol{m}_k\in R^q$ (其中 q 为变量数) 和 $\boldsymbol{V}_1,\mathrm{L}\ ,\boldsymbol{V}_k\in R^{q\times q}$。则各样本集的类间差 $\boldsymbol{S}_m$ 及类内差 $\boldsymbol{S}_w$ 分别为

$$\begin{cases}\boldsymbol{S}_m=\sum\limits_{i=1}^{k}\left(\boldsymbol{m}_i-\dfrac{1}{k}\sum\limits_{j=1}^{k}\boldsymbol{m}_j\right)\\ \boldsymbol{S}_w=\sum\limits_{i=1}^{k}\boldsymbol{V}_i\end{cases}\tag{8.34}$$

PCA 不考虑类的差别，将样本总体作为一类，只考虑总体方差

$$\boldsymbol{S}=\sum_{i=1}^{N}\left(\boldsymbol{x}_i-\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{x}_j\right)^2\tag{8.35}$$

由于 PCA 的优化目标只具有总体性，而不包含类别信息，因而在分类时，提取的主元有可能是盲目的。如果考虑样本中各类方差有不同的重要性，则赋予不同的类权值 β_i；同时若

还考虑样本总方差和类间差，也相应地赋予不同的权值系数 γ，则此时分类的目标将是最小化如下的类内方差：

$$\boldsymbol{S}_e = \sum_{i=1}^{k} \beta_i \boldsymbol{V}_i \tag{8.36}$$

并且同时最大化如下加权的类间差和总体方差：

$$\boldsymbol{S}_b = (1-\gamma)\boldsymbol{S} + \gamma\tau\boldsymbol{S}_m \tag{8.37}$$

式中，$\tau = N/k$，以使得式(8.35)右边两项在同一量级上。

这样就将先验的类别信息蕴含于 PCA 中，提取的主成分会更倾向于分类的目的。如果考虑向一维空间的投影 $y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$，则式(8.36)和式(8.37)所共同表述的优化目标即为

$$\arg\max_{w\neq 0} \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_e\boldsymbol{w}} = \lambda_1 \tag{8.38}$$

式中 $\boldsymbol{S}_b$，$\boldsymbol{S}_e$ 均为非负定阵，对式(8.38)的求解实际上是求解如下方程的特征值和特征向量：

$$\boldsymbol{S}_b\boldsymbol{w} = \lambda\boldsymbol{S}_e\boldsymbol{w} \tag{8.39}$$

记方程的非零特征根为 $\lambda_1 \geq \mathrm{L} \geq \lambda_p$，最大特征根 λ_1 就是式(8.39)的解，而特征向量就是一维的变换 $\boldsymbol{w}$，即第一主成分。

(2)样本库重构

为使样本分散性更好，将样本向量映射到另一更高维的空间 $\boldsymbol{\Phi}: R^N \to F$，在 F 空间，假设 $\boldsymbol{w}$ 为 $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ 的线性组合，即

$$\boldsymbol{w} = \boldsymbol{\Phi}(X)_N\boldsymbol{\alpha} \tag{8.40}$$

式中 $\boldsymbol{\Phi}(X)_N = (\boldsymbol{\Phi}(\boldsymbol{x}_1), \mathrm{L}\ , \boldsymbol{\Phi}(\boldsymbol{x}_N))$，$\boldsymbol{\alpha} = (\alpha_1, \mathrm{L}\ , \alpha_N)^{\mathrm{T}}$。由式(8.40)可以看出 $\boldsymbol{\alpha}$ 维数与样本数相同，如果样本量较大，会造成最终的主成分维数非常多，使计算复杂度增加。在 F 空间，还会出现某样本 $\boldsymbol{\Phi}(\boldsymbol{x})$ 可以近似由其他样本线性表示的情况，造成后面的 K 阵非常奇异。出于这两个目的的考虑，要对初始样本进行重构，建立新的用来建模的稀疏样本库。重构的基本思想是，逐个引入样本，并判断这个样本是否可以由样本库的样本线性表示，如果否则引入，反之则不引之。为此引入如下定理：

【定理8.9】 $K_n = (\boldsymbol{\Phi}(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{x}_j))$，$i,j = 1,\mathrm{L}\ ,n$，$\boldsymbol{K}_n = \begin{bmatrix} K_{n-1} & \vec{K}_n \\ \vec{K}_n^{\mathrm{T}} & \hat{K}_n \end{bmatrix}$，$|K_{n-1}| \neq 0$，$|K_{n-1}| \neq 0$，$\hat{K}_n = \boldsymbol{\Phi}(\boldsymbol{x}_n)^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{x}_n)$ 为标量，其他矩阵具有相应的维数，$\delta = \hat{K}_n - \vec{K}_n^{\mathrm{T}}K_{n-1}^{-1}\vec{K}_n$，如果 $\delta = 0$，则 $\boldsymbol{\Phi}(\boldsymbol{x}_n)$ 可以由 $\boldsymbol{\Phi}(\boldsymbol{x}_i)$，$i = 1,\ \mathrm{L}\ ,n-1$ 表示。

证明：因为初等变换不改变矩阵的秩，所以对 K_n 做变换

$$\begin{bmatrix} I_{n-1} & 0 \\ -\vec{K}_n^{\mathrm{T}}K_{n-1}^{-1} & 1 \end{bmatrix} \begin{bmatrix} K_{n-1} & \vec{K}_n \\ \vec{K}_n^{\mathrm{T}} & \hat{K}_n \end{bmatrix} \begin{bmatrix} I_{n-1} & -K_{n-1}^{-1}\vec{K}_n \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} K_{n-1} & 0 \\ 0 & \hat{K}_n - \vec{K}_n^{\mathrm{T}}K_{n-1}^{-1}\vec{K}_n \end{bmatrix} = \boldsymbol{A}$$

很明显，如果 $\delta = 0$，则 $|\boldsymbol{A}| = |K_n| = 0$，因而 $\mathrm{rank}(K_n) < n$，令 $\boldsymbol{C} = (\boldsymbol{\Phi}(\boldsymbol{x}_1), \mathrm{L}\ , \boldsymbol{\Phi}(\boldsymbol{x}_{n-1}),\ \boldsymbol{\Phi}(\boldsymbol{x}_n))$，因为 $K_n = \boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}$，$\mathrm{rank}(K_n) = \mathrm{rank}(\boldsymbol{C}) < n$，且 $|K_{n-1}| \neq 0$，所以 $\boldsymbol{\Phi}(\boldsymbol{x}_n)$ 可以用 $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ 线性表示。证毕。

以任意一个样本开始作为初始样本库，逐个引入样本，计算 δ。根据定理 8.9，如果 $\delta=0$，则说明这个新样本可以由样本库的样本线性表示，不引入，否则引入。对于样本数量较大的情况，可以采用判据 $\delta\geq\varepsilon$（ε 为较小的数），通过 ε 调整重构样本库的数量，这样就可以减少 $\boldsymbol{\alpha}$ 的维数，得到一个稀疏的样本库，达到降低特征向量减少计算复杂度的目的。

(3)进行 Kernel 转换，求取特征方程

在高维 F 空间，将各类样本的总体方差(S)、样本类内总离散度矩阵($\boldsymbol{S}_m$)、类间离散度矩阵($\boldsymbol{S}_w$)写成矩阵形式，得到 F 空间 $\boldsymbol{S}_e,\boldsymbol{S}_b$ 的矩阵表达形式：

$$\boldsymbol{S}_e=\sum_{i=1}^{k}\beta_i\boldsymbol{\Phi}(\boldsymbol{X})_{N_i}\left(I_{N_i}-\frac{1}{N_i}1_{N_i\times N_i}\right)\boldsymbol{\Phi}(\boldsymbol{X})_{N_i}^{\mathrm{T}} \tag{8.41}$$

$$\boldsymbol{S}_b=(1-\gamma)\sum_{i=1}^{N}\boldsymbol{\Phi}(\boldsymbol{X})_N\left(I_N-\frac{1}{N}1_{N\times N}\right)\boldsymbol{\Phi}(\boldsymbol{X})_N^{\mathrm{T}}+\gamma\tau\boldsymbol{\Phi}(M)\left(I_k-\frac{1}{k}1_{k\times k}\right)\boldsymbol{\Phi}(\boldsymbol{M})^{\mathrm{T}} \tag{8.42}$$

式中，$\boldsymbol{\Phi}(M)=(\boldsymbol{\Phi}(m_1),\mathrm{L}\ ,\boldsymbol{\Phi}(m_k))$ 为 k 个类均值点的映射矩阵；$\boldsymbol{\Phi}(m_i)=\frac{1}{N_i}\boldsymbol{\Phi}(x)_{N_i}1_{N_i}$，$1_{N_i}$ 表示 N_i 个 1 组成的列向量，$1_{k\times k}$ 是各元素为 1 的 $k\times k$ 维矩阵，$\boldsymbol{\Phi}(\boldsymbol{X})_{N_i}=(\boldsymbol{\Phi}(x_1^{N_i}),\mathrm{L}\ ,\boldsymbol{\Phi}(x_{N_i}^{N_i}))$，$x_i^{N_i}$ 为 N_i 类的第 i 个样本。

将式(8.40)~式(8.42)代入式(8.39)，并左乘以 $\boldsymbol{\Phi}(\boldsymbol{X})_N^{\mathrm{T}}$，且进行核变换，得到特征方程

$$\boldsymbol{A}\boldsymbol{\alpha}=\lambda\boldsymbol{B}\boldsymbol{\alpha} \tag{8.43}$$

式中 $\boldsymbol{A}=(1-\gamma)\boldsymbol{K}\left(I_N-\frac{1}{N}1_{N\times N}\right)\boldsymbol{K}^{\mathrm{T}}+\gamma\tau\boldsymbol{T}\left(I_k-\frac{1}{k}1_{k\times k}\right)\boldsymbol{T}^{\mathrm{T}}$，$\boldsymbol{B}=\sum_{i=1}^{k}\beta_i\boldsymbol{K}_{N_i}\left(I_{N_i}-\frac{1}{N_i}1_{N_i\times N_i}\right)\boldsymbol{K}_{N_i}^{\mathrm{T}}$，$\boldsymbol{T}=(T_1,\mathrm{L}\ ,T_k)$，$\boldsymbol{T}_i=\frac{1}{N_i}K_{N_i}1_{N_i}$ 为 N 维列向量，$\boldsymbol{K}_{N_i}=(K(x_i,x_j^{N_j}))$，$i=1,\mathrm{L}\ ,N,j=1,\mathrm{L}\ ,N_i$ 为 $N\times N_i$ 维矩阵，$\boldsymbol{K}=(K(x_i,x_j))$，$i=1,\mathrm{L}\ ,N$ 为 $N\times N$ 维矩阵，$\boldsymbol{K}=(K_{N1},\mathrm{L}\ ,K_{Nk})$。

这样就将式(8.43)转化为求解下式的特征向量：

$$\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{\alpha}=\lambda\boldsymbol{\alpha} \tag{8.44}$$

式(8.44)中如果矩阵 $\boldsymbol{B}$ 是奇异的，则引入一个小变量将其修改为 $\boldsymbol{B}+\mu\boldsymbol{I}$。

(4)分界阈值点和 LS-SVM 分类器

任意样本 $\boldsymbol{x}$ 在第 n 个主成分上的投影为

$$h(\boldsymbol{x})^{(n)}=\boldsymbol{\Phi}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{w}^{(n)}=\boldsymbol{\Phi}(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{\Phi}(\boldsymbol{x}_1),\mathrm{L}\ ,\boldsymbol{\Phi}(\boldsymbol{x}_N))\boldsymbol{\alpha}^{(n)}=\sum_{i=1}^{N}\alpha_i^{(n)}K(\boldsymbol{x}_i,\boldsymbol{x}) \tag{8.45}$$

利用 PKPCA 分析会发现，对于前 $k-1$ 个主成分的投影，其在三维空间中呈现出围绕各类中心的峰或谷的形式，而对其他主成分则没有。

因此前 $k-1$ 个主成分对分类起很大的作用。对于两类问题，如果通过第一个主成分进行分类，则常用两种分界阈值点：

$$y_1=(\boldsymbol{w}^{(1)})^{\mathrm{T}}\frac{\boldsymbol{\Phi}(\boldsymbol{m}_1)+\boldsymbol{\Phi}(\boldsymbol{m}_2)}{2}=\frac{1}{2}(\boldsymbol{\alpha}^{(1)})^{\mathrm{T}}\left(\frac{1}{N_1}K_{N1}1_{N1}+\frac{1}{N_2}K_{N2}1_{N2}\right) \tag{8.46}$$

$$y_2 = (\boldsymbol{w}^{(1)})^{\mathrm{T}} \frac{N_1\boldsymbol{\Phi}(\boldsymbol{m}_1) + N_2\boldsymbol{\Phi}(\boldsymbol{m}_2)}{2} = \frac{1}{N_1+N_2}(\boldsymbol{\alpha}^{(1)})^{\mathrm{T}}\left(\frac{1}{N_1}K_{N1}1_{N1} + \frac{1}{N_2}K_{N2}1_{N2}\right) \tag{8.47}$$

当提取足够的主成分后，可以采用其他分类器进行分类。如采用 SVM 的一种 LS-SVM (Least Square Support Vector Machine) (Gestel T. V, Suykens J. A. K. et al. 2002) 作为 PKPCA 的输出分类器。

8.4.4　PKPCA与KPCA和KFD的关系

PKPCA 和 KPCA 及 KFD 都是利用 Kernel 变换，提取主成分来进行分析。三者之间既有区别也有联系[15]。

(1) 当取式 (8.39) $S_e = 1$，考虑 $\gamma \to 0$，此时 $\boldsymbol{S}_b \to \boldsymbol{S}$，就得到 KPCA 的特征方程

$$\lambda\boldsymbol{\alpha} = \left(I_N - \frac{1}{N}1_{N\times N}\right)K\boldsymbol{\alpha} \tag{8.48}$$

取 $\gamma = 0.1$，代表 (近似) KPCA 的情况，$\gamma = 0.9$ 表示 PKPCA，来对其分类能力进行比较。

(2) 当 $\gamma \to 1$，$\boldsymbol{S}_b \to \boldsymbol{S}_m$，则为 KFD 形式

$$\lambda\sum_{i=1}^{k} K_{N_i}\left(I_{N_i} - \frac{1}{N_i}1_{N_i\times N_i}\right)\boldsymbol{K}_{N_i}^{\mathrm{T}}\boldsymbol{\alpha} = \boldsymbol{A}'\boldsymbol{\alpha} \tag{8.49}$$

此时，$\boldsymbol{A}' = \boldsymbol{T}\left(I_k - \frac{1}{k}1_{k\times k}\right)\boldsymbol{T}^{\mathrm{T}}$，注意到 $\mathrm{tank}(\boldsymbol{A}') \leq k-1$，即 KFD 只能求得 (类别数 – 1) 个特征向量，这对于变量维数较高时的压缩是极大的，而式 (8.43) 中 $\boldsymbol{A}$ 的另一项即总体方差的加入使提取的主成分更多。

此外，如果 $\beta_i = 1$，而 $\beta_j = 0$, $j = 1,\mathrm{L}\ ,k, j \neq i$，则

$$\lambda\boldsymbol{K}_{N_i}\left(I_{N_i} - \frac{1}{N_i}1_{N_i}\right)\boldsymbol{K}_{N_i}^{\mathrm{T}}\boldsymbol{\alpha} = \boldsymbol{A}\boldsymbol{\alpha} \tag{8.50}$$

表示只考虑某类样本的样本内差，从而提高这类样本的特征识别能力这样特别适合特定样本集的判别，如身份认证等场合。

习题 8

8.1　令 k_1 和 k_2 是 $X\times X$ 空间的核函数，$X \subseteq R^n$, $a \in R^+$，$f(\cdot)$ 是 X 空间的实值函数，$\varphi: X \to R^N$ 是 $R^N \times R^N$ 空间具有核函数 k_3 的映射。$\boldsymbol{B}$ 是 $n\times n$ 的对称半正定矩阵。证明下列函数是核函数：

(1) $k(\boldsymbol{x},\boldsymbol{z}) = k_1(\boldsymbol{x},\boldsymbol{z}) + k_2(\boldsymbol{x},\boldsymbol{z})$

(2) $k(\boldsymbol{x},\boldsymbol{z}) = ak_1(\boldsymbol{x},\boldsymbol{z})$

(3) $k(\boldsymbol{x},\boldsymbol{z}) = k_1(\boldsymbol{x},\boldsymbol{z})k_2(\boldsymbol{x},\boldsymbol{z})$

(4) $k(\boldsymbol{x},\boldsymbol{z}) = f(\boldsymbol{x})f(\boldsymbol{z})$

(5) $k(\boldsymbol{x},\boldsymbol{z}) = k_3(\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{z}))$

(6) $k(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{x}'\boldsymbol{B}\boldsymbol{z}$

8.2 令 $k_1(\boldsymbol{x}, \boldsymbol{z})$ 是 $X \times X$ 空间的核函数，其中 $\boldsymbol{x}, \boldsymbol{z} \in X$，$p(x)$ 是具有正系数的多项式，则下面的函数也是核函数：

(1) $k(\boldsymbol{x}, \boldsymbol{z}) = p(k_1(\boldsymbol{x}, \boldsymbol{z}))$

(2) $k(\boldsymbol{x}, \boldsymbol{z}) = \exp(k_1(\boldsymbol{x}, \boldsymbol{z}))$

(3) $k(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\| \boldsymbol{x} - \boldsymbol{z} \|^2 / (2\sigma^2))$

参考文献

[1] Boser B, Guyon, Vapnik V N. *A training algorithm for optimal margin classifiers*. Fifth Annual Workshop on Computational Learning Theory, Pittsburgh ACM, 1992: 144-152.

[2] Scholkopf B. *Support Vector Learning* [D]. Berlin University, 1997.

[3] Francis R. Bach, Michael I. Jordan. *Kernel Independent Component Analysis*[R]. University of California, Berkeley, CA 94720, USA 2001.11.

[4] 吴涛. 核函数的性质、方法及其在障碍检测中的应用[D], 国防科技技术大学, 2003.

[5] 边肇祺, 张学工. 模式识别(第二版), 北京: 清华大学出版社, 1999.

[6] Vapnik V. N. *The Nature of Statistical Learning Theory*, NY: Springer-Verlag, 1995. 张学工译. 统计学习理论的本质. 北京: 清华大学出版社, 2000.9.

[7] A. Luntz, V Brailovsky. *On estimation of characters obtained in statistical procedure of recognition*. Technicheskaya Kibernetica, Russia, 3, 1969.

[8] John Shawe-Taylor, Nello Cristianini. *Kernel Methods for Pattern Analysis* [M]. 北京: 机械工业出版社, 2005.1(影印版).

[9] M. Seeger. *Bayes method for support vector machine and Gaussian processes*. 1999.

[10] 阎辉, 张学工, 马云潜, 李衍达. 基于变异函数的径向机核函数的参数估计[J]，自动化学报，28(3)；450-455, 2002.

[11] 王政权. 地统计学及在生态学中的应用[M]. 北京: 科学出版社, 1999.

[12] A. J. Smola, B. Scholkopf, K. R. Muller. *The Connection between regularization operators and support vector kernels*. Neural Networks 11: 637-649, 1998.

[13] Tao Wu, Han-gen He. *Interpolation of scattered data and classifying in SVM*[C]. International Conference of Neural Information Processing, Shanghai, 2001.11.

[14] M. Kass, A. Witkin, D. Terzopoules. Snakes. *Active contour models*[C], Proceedings of First International Conference on Computer Vision, 1987: 259-269.

[15] 解应春. 基于 Kernel 学习机的建模与分类的应用算法研究[D], 浙江大学，2003.

第 9 章　模糊模式识别

在我们的日常生活中，常常会遇到一些“模糊性”现象。例如，“秃头悖论”问题：一位已经谢顶的老教授与他的学生争论他是否为秃头问题。

教授问：我是秃头吗？

学生答：您的头顶上已经没有多少头发，确实应该说是。

教授问：你秀发稠密，绝对不算秃头；我问你，如果你头上脱落了一根头发之后，能说变成了秃头了吗？

学生答：我减少一根头发之后，当然不会变成秃头。

教授说：好了，总结我们的讨论，得出下面的命题：“如果一个人不是秃头，那么他减少一根头发仍不是秃头”，你说对吗？

学生答：对！

教授说：我年轻时也和你一样是一头秀发，当时没有人说我秃头，后来随着年龄的增大，头发一根根减少到今天的样子。但每掉一根头发，根据我们刚才的命题，我都不应该称为秃头，这样经过有限次头发的减少，用这一命题有限次，结论是：“我今天仍不是秃头”。

这个悖论反映了精确与模糊之间的矛盾，对于模糊的事物，比如秃与不秃，没有绝对的界限。虽然在微小的量变之中已经蕴含着质的差别，然而这种差别绝对不能仅用“是”与“非”这两个字来刻画。因此，数学归纳法这种只适用精确的方法，不能直接搬到模糊现象中来用。

1965 年，美国控制论专家扎德(L. A. Zadeh)把模糊性和数学统一起来，提出了模糊集理论，随着数学界和工程界研究的广泛和深入，理论成果和应用成果不断出现，创建了一门新学科——模糊数学。模糊集理论是对一类客观事物和性质更合理的抽象和描述，是传统集合理论的必然推广。

传统的模式识别是一种硬划分，它把每个待识别的对象严格地划分到某类中，类别之间的界限是分明的。而实际上大多数对象并没有严格的属性，它们的形态和类属存在着中介性，适合进行软划分。模糊集理论的提出为这种软划分提供了有力的分析工具，人们开始用模糊方法处理聚类问题，并称为模糊模式识别(或模糊聚类分析)。由于模糊聚类得到了样本属于各个类别的不确定性程度，建立了样本对类别的不确定性描述，因此更能客观地反映现实世界。

9.1　模糊数学的基本理论

9.1.1　模糊集合

1．隶属函数

与模糊集合相对应，这里将传统经典集合论中的集合称为经典集合或普通集合。假设在普通集合中 x 为论域 $\boldsymbol{U}$ 中的一个元素，A 为 $\boldsymbol{U}$ 的一个子集，即 $A \subset \boldsymbol{U}$，那么 x 与 A 的关系可以用特征函数 $C_A(x)$ 来描述：

$$C_A(x)=\begin{cases}1, & x\in A\\ 0, & x\notin A\end{cases} \tag{9.1}$$

即 x 要么属于 A，要么不属于 A，两者必取其一，非此即彼。由此可见，集合中的特征函数与二值逻辑相对应，即只能取 $\{0,1\}$ 两个值，从而不能表达现实中存在的“亦此亦彼”的模糊现象。因此，有必要将式(9.1)所示的特征函数推广到闭区间[0, 1]，这就形成了模糊数学中的隶属函数 $\mu_{\tilde{A}}(x)$。

【定义 9.1】 设在论域 $\boldsymbol{U}$ 上给定的一个映射为

$$\mu_{\tilde{A}}:\boldsymbol{U}\to[0,1] \tag{9.2}$$

$$x\to\mu_{\tilde{A}}(x) \tag{9.3}$$

则称 $\tilde{A}$ 为 $\boldsymbol{U}$ 上的模糊(Fuzzy)集合，$\mu_{\tilde{A}}(x)$ 称为 $\tilde{A}$ 的隶属函数(Membership Function)，或称为 x 对 $\tilde{A}$ 的隶属度。隶属度越大，表示 x 隶属 $\tilde{A}$ 的程度越高。例如，$\mu_{\tilde{A}}(x)=1$，表示 x 完全属于 $\tilde{A}$；$\mu_{\tilde{A}}(x)=0$，表示 x 不属于 $\tilde{A}$；$0<\mu_{\tilde{A}}(x)<1$，表示 x 属于 $\tilde{A}$ 的程度介于“属于”和“不属于”之间，即是模糊的。

特别地，当 $\mu_{\tilde{A}}(x)=\{0,1\}$ 时，$\tilde{A}(x)$ 蜕化为一个普通集合的特征函数，$\tilde{A}$ 也成为一个普通集合。由此可见，普通集合是模糊集合的特殊情况：

$$\tilde{A}=\{x\in\boldsymbol{U}\mid\mu_{\tilde{A}}(x)=1\} \tag{9.4}$$

隶属函数的表示方法大致有三种。一般情况下，序偶表示为

$$\tilde{A}=\{(\mu_{\tilde{A}}(x),x)\mid x\in\boldsymbol{U}\} \tag{9.5}$$

如果 $\boldsymbol{U}$ 是有限集合或可数集，那么求和形式(也称 Zadeh 表示法)表示为

$$\tilde{A}=\sum_i\mu_{\tilde{A}}(x_i)/x_i \tag{9.6}$$

如果模糊集合中各元素的顺序已确定，那么模糊集合 $\tilde{A}$ 对应的向量形式表示为

$$\tilde{A}=(\mu_{\tilde{A}}(x_1),\mu_{\tilde{A}}(x_2),\mathrm{L}\ ,\mu_{\tilde{A}}(x_n)) \tag{9.7}$$

如果 $\boldsymbol{U}$ 是无限集，则可表示为

$$\tilde{A}=\int_U\mu_{\tilde{A}}(x)/x \tag{9.8}$$

其中 $\sum_i$ 和 $\int_U$ 不是求和与积分，它们表示附有隶属度的各元素的并，是各个元素与隶属函数对应关系的一个总括。

【例 9.1】 设有论域 $\boldsymbol{U}=\{a,b,c,d\}$，$\tilde{A}$ 是 $\boldsymbol{U}$ 上“圆形”的一个模糊子集，如图9.1所示，对 $\boldsymbol{U}$ 中的每一个元素指定一个它对 $\tilde{A}$ 的隶属度(对圆形的隶属程度)，分别为

$\mu_{\tilde{A}}(a)=1$，$\mu_{\tilde{A}}(b)=0.8$，$\mu_{\tilde{A}}(c)=0.5$，$\mu_{\tilde{A}}(d)=0.3$

那么，模糊集合 $\tilde{A}$ 可表示为

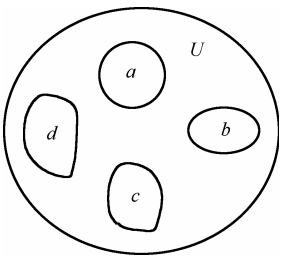


图 9.1 模糊集合

求和表示法：$\tilde{A} = \frac{1}{a}+\frac{0.8}{b}+\frac{0.5}{c}+\frac{0.3}{d}$

序偶表示法：$\tilde{A} = \{(1, a), (0.8, b), (0.5, c), (0.3, d)\}$

向量表示法：$\tilde{A} = (1, 0.8, 0.5, 0.3)$

2．模糊集合的运算

【定义 9.2】 设 $\tilde{A}, \tilde{B}$ 为论域 $\boldsymbol{U}$ 中的模糊集合，规定模糊集合之间的包含为 $\tilde{A} \subseteq \tilde{B}$、相等为 $\tilde{A} = \tilde{B}$、并为 $\tilde{A} \cup \tilde{B}$、交为 $\tilde{A} \cap \tilde{B}$、余为 $\tilde{A}^C$，相应的运算如下：

包含　$\tilde{A} \subset \tilde{B} \Leftrightarrow (\forall x \in \boldsymbol{U})(\mu_{\tilde{A}}(x) \geqslant \mu_{\tilde{B}}(x))$

相等　$\tilde{A} = \tilde{B} \Leftrightarrow (\forall x \in \boldsymbol{U})(\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x))$

并　$\tilde{C} = \tilde{A} \cup \tilde{B} \Leftrightarrow (\forall x \in \boldsymbol{U})\ (\mu_{\tilde{C}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x)) = \max\left[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\right]$

交　$\tilde{D} = \tilde{A} \cap \tilde{B} \Leftrightarrow (\forall x \in \boldsymbol{U})(\mu_{\tilde{D}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x)) = \min\left[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\right]$

余　$\tilde{E} = A^C \Leftrightarrow (\forall x \in \boldsymbol{U})(\mu_{\tilde{E}}(x) = 1 - \mu_{\tilde{A}}(x))$

其中，符号“$\vee$”表示取最大值，“$\wedge$”表示取最小值。

3．λ 截集

截集的概念描述了模糊集合与普通集合之间的转换关系。

【定义 9.3】 设 $\tilde{A}$ 为论域 $\boldsymbol{U}$ 中的模糊集合，对任意 $\lambda \in [0,1]$，集合

$$A_\lambda = \left\{ x \middle| x \in \boldsymbol{U}, \mu_{\tilde{A}}(x) \geq \lambda \right\},$$

则普通集合 A_λ 被称为集合 $\tilde{A}$ 的 λ 截集，λ 称为阈值或置信水平。

由定义可知，集合 $\tilde{A}$ 为模糊集，A_λ 为普通集，通过阈值实现了模糊集到普通集的转换。下面举一个例子对这种集合的转换予以说明。

【例 9.2】 有 5 个病人 x_1, x_2, x_3, x_4, x_5，体温分别为 38.9°C, 37.0°C, 37.2°C, 39.2°C, 38.1°C，护士统计时的记录如下[1]：

37.0°C 以上者有 5 人：x_1, x_2, x_3, x_4, x_5

37.5°C 以上者有 3 人：x_1, x_4, x_5

39.0°C 以上者有 1 人：x_4

在考虑有多少病人发烧时，医生就可能根据不同经验得出不同的结论。如果认为发烧的温度界限是 37.0°C，则有 5 人发烧；如果温度界限是 37.5°C，则只有 3 人发烧。

由于发烧属于模糊概念，所以用模糊数学来描述更为合适。根据医生的经验，可将各个温度段用发烧的隶属度表示：

$T \geqslant$ 39.0°C，隶属度等于 1.0

38.5°C $\leqslant T <$ 39.0°C，隶属度等于 0.9

38.0°C $\leqslant T <$ 38.5°C，隶属度等于 0.7

37.0°C $\leqslant T <$ 38.0°C，隶属度等于 0.4

$T <$ 37.0°C，隶属度等于 0.0

用模糊集合 $\tilde{A}$ 表示“发烧病人”，有

$$\tilde{A} = \{(0.9, x_1), (0.4, x_2), (0.4, x_3), (1.0, x_4), (0.7, x_5)\}$$

这样，就可以方便地对病人进行分类。如果将隶属度在 0.9 以上的病人认为是发高烧，并进行特护处理，那么这些病人可表示为 $A_{0.9} = \{x_1, x_4\}$。

9.1.2 模糊关系

设 $\boldsymbol{X}, \boldsymbol{Y}$ 是两个论域，则 $\boldsymbol{X} \times \boldsymbol{Y} = \{(x, y) | x \in \boldsymbol{X}, y \in \boldsymbol{Y}\}$ 为 $\boldsymbol{X}$ 与 $\boldsymbol{Y}$ 的笛卡儿乘积(也称直积)。笛卡儿积 $\boldsymbol{X} \times \boldsymbol{Y}$ 是由两个集合间元素无约束地搭配成的序偶 (x, y) 的全体所构成的集合。

序偶中两个元素的排列是有序的，对于 $\boldsymbol{X} \times \boldsymbol{Y}$ 中的元素必须是 (x, y), $x \in \boldsymbol{X}, y \in \boldsymbol{Y}$；也就是说，$(x, y)$ 与 (y, x) 是不同的序偶。

例如，$\boldsymbol{X} = \{\alpha, \beta\}, \boldsymbol{Y} = \{1, 2\}$，则 $\boldsymbol{X} \times \boldsymbol{Y}$ 与 $\boldsymbol{Y} \times \boldsymbol{X}$ 分别为

$$\boldsymbol{X} \times \boldsymbol{Y} = \{(\alpha, 1), (\alpha, 2), (\beta, 2), (\beta, 2)\}$$

$$\boldsymbol{Y} \times \boldsymbol{X} = \{(1, \alpha), (1, \beta), (2, \alpha), (2, \beta)\}$$

1. 模糊关系的定义和表示

【定义 9.4】 设 $\boldsymbol{X}, \boldsymbol{Y}$ 是两个论域，笛卡儿积 $\boldsymbol{X} \times \boldsymbol{Y}$ 上的一个模糊子集 $\tilde{R}$ 称为从 $\boldsymbol{X}$ 到 $\boldsymbol{Y}$ 的一个模糊关系，记为 $X \xrightarrow{\tilde{R}} Y$。$\tilde{R}$ 的隶属函数 $\mu_{\tilde{R}}(x, y)$ 表示了 X 中元素 x 与 Y 中元素 y 具有关系的程度。

特别地，当 $\boldsymbol{X} = \boldsymbol{Y}$ 时，称 $\tilde{R}$ 为论域 $\boldsymbol{X}$ 上的二元模糊关系。

$\boldsymbol{X} \times \boldsymbol{Y}$ 上的全体模糊关系记为 $F(\boldsymbol{X} \times \boldsymbol{Y})$。

当论域 $\boldsymbol{X} = \{x_1, x_2, \cdots, x_m\}$, $\boldsymbol{Y} = \{y_1, y_2, \cdots, y_n\}$ 都是有限离散论域时，模糊关系可用矩阵 $\boldsymbol{R} = (r_{ij})_{m \times n}$ 来表示，其中 $r_{ij} = \mu_{\tilde{R}}(x_i, y_j)$，$0 \leqslant r_{ij} \leqslant 1 (1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n)$。矩阵 $\boldsymbol{R}$ 称为模糊矩阵。

特别地，当 $r_{ij} \in \{0, 1\} (1 \leqslant i, j \leqslant n)$ 时，模糊矩阵 $\boldsymbol{R}$ 转化为布尔矩阵。

【例 9.3】 在医学上通常用公式“身高(cm)–100=标准体重(kg)”来描述成年男性的体重与身高的关系，通常认为超过标准体重 10%者为偏重，超过 20%者为肥胖，低于 10%者为偏瘦，低于 20%者为消瘦。事实上，对一般健康人，采用这个公式衡量时常会有些错误，但这不能说明他们不正常。用模糊关系更能客观地反映出身高与体重的关系，如表9.1所示。

表 9.1 成年男性身高与标准体重的模糊关系

$\mu_{\tilde{R}}(x, y)$	50 kg	60 kg	70 kg	80 kg	90 kg
150 cm	1	0.8	0.2	0.1	0
160 cm	0.8	1	0.8	0.2	0.1
170 cm	0.2	0.8	1	0.8	0.2
180 cm	0.1	0.2	0.8	1	0.8
190 cm	0	0.1	0.2	0.8	1

表 9.1 的模糊矩阵表示为

$$\boldsymbol{R}=\begin{bmatrix}1 & 0.8 & 0.2 & 0.1 & 0\\ 0.8 & 1 & 0.8 & 0.2 & 0.1\\ 0.2 & 0.8 & 1 & 0.8 & 0.2\\ 0.1 & 0.2 & 0.8 & 1 & 0.8\\ 0 & 0.1 & 0.2 & 0.8 & 1\end{bmatrix}$$

2．模糊关系的合成

若已知 $\boldsymbol{X}$ 到 $\boldsymbol{Y}$ 的模糊关系 $\tilde{R}$，$\boldsymbol{Y}$ 到 $\boldsymbol{Z}$ 的模糊关系 $\tilde{S}$；欲通过 $\tilde{R}$ 与 $\tilde{S}$ 求 $\boldsymbol{X}$ 到 $\boldsymbol{Z}$ 的模糊关系，可以运用关系合成来解决。

【定义 9.5】 设 $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ 是三个论域，模糊关系 $\tilde{R}\in F(\boldsymbol{X}\times\boldsymbol{Y})$，$\tilde{S}\in F(\boldsymbol{Y}\times\boldsymbol{Z})$，由 $\tilde{R}$ 与 $\tilde{S}$ 合成一个新的模糊关系 $\tilde{R}\circ\tilde{S}\in F(\boldsymbol{X}\times\boldsymbol{Z})$，$\tilde{R}\circ\tilde{S}$ 是 $\boldsymbol{X}$ 到 $\boldsymbol{Z}$ 的模糊关系，它的隶属函数为

$$\mu_{\tilde{R}\circ\tilde{S}}(x,z)=\bigvee_{y\in\boldsymbol{Y}}(\mu_{\tilde{R}}(x,y)\wedge\mu_{\tilde{S}}(y,z))$$

模糊关系与自身的运算又称为幂运算，即

$$\tilde{R}^{\theta}=\tilde{R}\circ\tilde{R}$$

$$\tilde{R}^{\theta}=\tilde{R}^{\theta-1}\circ\tilde{R}$$

如果 $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ 都是有限论域，$\tilde{R}$ 和 $\tilde{S}$ 对应的模糊矩阵分别为 $\boldsymbol{R}=(r_{ij})_{m\times n}$ 和 $\boldsymbol{S}=(s_{ij})_{n\times 1}$，那么 $\tilde{R}$ 对 $\tilde{S}$ 的模糊关系的合成 $\tilde{Q}=\tilde{R}\circ\tilde{S}$，其模糊矩阵 $\boldsymbol{Q}=(q_{ij})_{m\times 1}$，其中 $q_{ij}=\bigvee_{k=1}^{n}(r_{ik}\wedge s_{kj})$。也就是说，模糊关系的合成对应模糊矩阵的乘积。

【例 9.4】 已知模糊矩阵 $\boldsymbol{R}=\begin{bmatrix}0.3 & 0.7 & 0.2\\ 1 & 0 & 0.4\\ 0 & 0.5 & 1\\ 0.6 & 0.7 & 0.8\end{bmatrix}_{4\times 3}$ 和 $\boldsymbol{S}=\begin{bmatrix}0.1 & 0.9\\ 0.9 & 0.1\\ 0.6 & 0.4\end{bmatrix}_{3\times 2}$，求 $\boldsymbol{R}$ 对 $\boldsymbol{S}$ 的合成矩阵（也称 $\boldsymbol{R}$ 对 $\boldsymbol{S}$ 的模糊矩阵的乘积）。

解： $\boldsymbol{Q}=\boldsymbol{R}\circ\boldsymbol{S}$

$q_{11}=(0.3\wedge 0.1)\vee(0.7\wedge 0.8)\vee(0.2\wedge 0.6)=0.1\vee 0.7\vee 0.2=0.7$

$q_{12}=(0.3\wedge 0.9)\vee(0.7\wedge 0.1)\vee(0.2\wedge 0.4)=0.3\vee 0.1\vee 0.2=0.3$

$q_{21}=(1\wedge 0.1)\vee(0\wedge 0.8)\vee(0.4\wedge 0.6)=0.1\vee 0\vee 0.4=0.4$

$q_{22}=$ ⊥

类似地，可计算得

$$\boldsymbol{Q}=\begin{bmatrix}0.7 & 0.3\\ 0.4 & 0.9\\ 0.6 & 0.4\\ 0.7 & 0.6\end{bmatrix}_{4\times 2}$$

模糊矩阵的合成运算不满足交换律，即 $\boldsymbol{R}\circ\boldsymbol{S}\neq\boldsymbol{S}\circ\boldsymbol{R}$。

3．模糊等价关系和模糊相似关系

【定义 9.6】 自反性：设 $\tilde{R}$ 是 $\boldsymbol{U}$ 中的模糊关系，若对 $\forall x\in\boldsymbol{U}$，都存在 $\mu_{\tilde{R}}(x,x)=1$，则称 $\tilde{R}$ 满足自反性。

【定义 9.7】 对称性：设 $\tilde{R}$ 是 $\boldsymbol{U}\times\boldsymbol{U}$ 中的模糊关系，当且仅当对 $\forall(x, y)\in \boldsymbol{U}$，都存在 $\mu_{\tilde{R}}(x, y)=\mu_{\tilde{R}}(y, x)$，则称 $\tilde{R}$ 满足对称性。

【定义 9.8】 传递性：设 $\tilde{R}$ 是 $\boldsymbol{U}\times\boldsymbol{U}$ 中的模糊关系，对 $\forall(x, y), (y, z), (x, z)\in \boldsymbol{U}\times\boldsymbol{U}$，都存在 $\vee_{y}(\mu_{\tilde{R}}(x, y)\wedge\mu_{\tilde{R}}(y, z))\leqslant \mu_{\tilde{R}}(x, z)$，则称 $\tilde{R}$ 满足传递性。

【定义 9.9】 模糊等价关系：设 $\tilde{R}$ 是 $\boldsymbol{U}$ 中的模糊关系，若 $\tilde{R}$ 同时具有自反性、对称性、传递性，则称 $\tilde{R}$ 是一个模糊等价关系，$\tilde{R}$ 对应的矩阵为模糊等价矩阵。

【定义 9.10】 模糊相似关系：设 $\tilde{R}$ 是 $\boldsymbol{U}$ 中的模糊关系，若 $\tilde{R}$ 同时具有自反性和对称性，则称 $\tilde{R}$ 为模糊相似关系，$\tilde{R}$ 对应的矩阵为模糊相似矩阵。

【定理 9.1】 $\tilde{R}\in F(\boldsymbol{U}\times\boldsymbol{U})$ 是 $\boldsymbol{U}$ 上的模糊等价关系的充要条件是，对 $\forall\lambda\in[0,1]$，R_λ 都是 $\boldsymbol{U}$ 上的普通等价关系。或者说，设 $\boldsymbol{R}$ 是 $n\times n$ 阶模糊等价矩阵，当且仅当 $\forall\lambda\in[0,1]$ 时，R_λ 都是等价的布尔矩阵。

由定理 9.1 可知，若 $\tilde{R}$ 为模糊等价关系，则对于给定的 $\lambda\in[0,1]$ 便可得到相应的普通等价关系 R_λ，这就意味着得到了一个 λ 水平的分类。

【定理 9.2】 若 $0\leqslant\lambda\leqslant\mu\leqslant1$，则截矩阵 $\boldsymbol{R}_\mu$ 所分出的每一类必是截矩阵 $\boldsymbol{R}_\lambda$ 所分出的某一类的子类，或称 $\boldsymbol{R}_\mu$ 的分类法是 $\boldsymbol{R}_\lambda$ 分类法的加细。

通常根据规定的类别数来选择合适的 λ 进行分类；或将 λ 从 1 逐渐降至 0，其决定的分类由细变粗，逐步归并，形成一个分级聚类树。

【例 9.5】 设论域 $\boldsymbol{X}=\{x_1, x_2, x_3, x_4, x_5\}$，给定 $\boldsymbol{X}$ 上一个模糊关系 $\tilde{R}$，其模糊矩阵为

$$\boldsymbol{R}=\begin{bmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{bmatrix}$$

要求按不同的 λ 分类。

解：矩阵显然具有自反性和对称性，而

$$\boldsymbol{R}\circ\boldsymbol{R}=\begin{bmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{bmatrix}=\boldsymbol{R}$$

所以，$\boldsymbol{R}$ 具有传递性，故 $\boldsymbol{R}$ 是模糊等价矩阵。

令 λ 由 1 降至 0，写出 $\boldsymbol{R}_\lambda$，按 $\boldsymbol{R}_\lambda$ 分类。

(1) 当 $\lambda=1$ 时，$\boldsymbol{R}_1=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}$。此时分为 5 类，即 $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}$，这是最细分类。

(2) 当$\lambda = 0.8$ 时，$\boldsymbol{R}_{0.8} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$。此时分为 4 类，即$\{x_1, x_3\}, \{x_2\}, \{x_4\}, \{x_5\}$。

(3) 当$\lambda = 0.6$ 时，$\boldsymbol{R}_{0.6} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$。此时分为 3 类，即$\{x_1, x_3\}, \{x_2\}, \{x_4, x_5\}$。

(4) 当$\lambda = 0.5$ 时，$\boldsymbol{R}_{0.5} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$。此时分为 2 类，即$\{x_1, x_3, x_4, x_5\}, \{x_2\}$。

(5) 当$\lambda = 0.4$ 时，$\boldsymbol{R}_{0.4} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$。此时 5 个元素分为 1 类，即$\{x_1, x_2, x_3, x_4, x_5\}$，它是最粗分类。

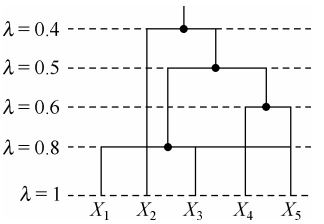


图 9.2　分级聚类树

于是可得出分级聚类树，也称动态聚类图，如图9.2所示。

9.1.3　模糊集合的度量

1．模糊度定义

对于模糊集合而言，其隶属函数的确定方法各种各样，常常带有主观性。对同一论域上的模糊集，不同的人或者用不同的判断标准，所得到的各元素的隶属度也不尽相同。而不同的隶属度结构又导致了该模糊集呈现不同的模糊性。为了建立合理地测度模糊集合模糊性的函数，首先给出模糊度的定义。

【定义 9.11】 设论域 $\boldsymbol{U}$ 上任一个模糊子集 $\tilde{A}$，为度量其模糊性大小，定义

$$D: \tilde{A} \to [0, 1]$$

为 $\tilde{A}$ 的模糊度 $D(\tilde{A})$，它应满足：

(a) 当且仅当 $\mu_{\tilde{A}}(x_i)$ 只取 0 或 1 时，

$$D(\tilde{A}) = 0$$

$x_i \in \boldsymbol{U}$，$\mu_{\tilde{A}}(x_i)$ 是 x_i 对 $\tilde{A}$ 的隶属度。若 $\mu_{\tilde{A}}(x_i) = 0$，则说明 x_i 完全不隶属于 $\tilde{A}$，没有模糊性；若 $\mu_{\tilde{A}}(x_i) = 1$，则说明 x_i 完全隶属于 $\tilde{A}$，也没有模糊性。

(b) 当 $\mu_{\tilde{A}}(x_i) = 0.5$ 时，$D(\tilde{A})$ 应取最大值，即

$$D(\tilde{A}) = 1$$

也就是说，$\mu_{\tilde{A}}(x_i)$越靠近 1 或 0，模糊性就越小；$\mu_{\tilde{A}}(x_i)$越远离 1 或 0，模糊性就越大，最大模糊性发生在 $\mu_{\tilde{A}}(x_i) = 0.5$，此处 $\mu_{\tilde{A}}(x_i)$离 1 或 0 同样远。

(c)对任意 $x_i \in \boldsymbol{U}$，设 $\boldsymbol{U}$ 上有两个模糊子集 $\tilde{A}$和 $\tilde{B}$，若

$$\mu_{\tilde{A}}(x_i) \geqslant \mu_{\tilde{B}}(x_i) \geqslant 0.5$$

或

$$\mu_{\tilde{A}}(x_i) \leqslant \mu_{\tilde{B}}(x_i) \leqslant 0.5$$

则

$$D(\tilde{B}) \geqslant D(\tilde{A})$$

也就是说，越靠近 0.5 就越模糊。

(d)$D(\tilde{A}) = D(\tilde{A}^C)$，其中 $\tilde{A}^C$ 是 $\tilde{A}$的补集，$\tilde{A}^C$ 定义为

$$\mu_{\tilde{A}^C}(x_i) = 1 - \mu_{\tilde{A}}(x_i)$$

这说明 $\tilde{A}$和它的补集 $\tilde{A}^C$ 具有同等的模糊性。

2．模糊度的计算

模糊度刻画了一个模糊集合的整体模糊程度。设 $\boldsymbol{U} = \{x_1, x_2, \cdots, x_n\}$，下面给出几个模糊度的计算公式。

(1)距离模糊度

设 $A_{0.5}$ 是 $\tilde{A}$的 $\lambda = 0.5$ 的截集，有

$$d_p(\tilde{A}) = \frac{2}{n^{1/p}}\left(\sum_{i=1}^{n}\left|\mu_{\tilde{A}}(x_i) - \mu_{A_{0.5}}(x_i)\right|^p\right)^{1/p} \tag{9.9}$$

则 $d_p(\tilde{A})$是 $\tilde{A}$的模糊度，又称为明可夫斯基(Minkowski)模糊度。

当 $p = 1$ 时，d_1 称海明(Hamming)模糊度，即

$$d_1(\tilde{A}) = \frac{2}{n}\left(\sum_{i=1}^{n}\left|\mu_{\tilde{A}}(x_i) - \mu_{A_{0.5}}(x_i)\right|\right) \tag{9.10}$$

当 $p = 2$ 时，d_2 称欧几里得(Euclidean)模糊度，即

$$d_2(\tilde{A}) = \frac{2}{\sqrt{n}}\sqrt{\sum_{i=1}^{n}\left|\mu_{\tilde{A}}(x_i) - \mu_{A_{0.5}}(x_i)\right|^2} \tag{9.11}$$

【例 9.6】 计算模糊集合 $\tilde{A}$和 $\tilde{B}$的海明模糊度和欧几里得模糊度，其中，

$$\tilde{A} = \frac{0.8}{a} + \frac{0.9}{b} + \frac{0.1}{c} + \frac{0.8}{d}$$

$$\tilde{B} = \frac{0.3}{a} + \frac{0}{b} + \frac{0.3}{c} + \frac{0}{d}$$

解：因为

$$A_{0.5}=\frac{1}{a}+\frac{1}{b}+\frac{0}{c}+\frac{1}{d}$$

$$B_{0.5}=\frac{0}{a}+\frac{0}{b}+\frac{0}{c}+\frac{0}{d}$$

则海明模糊度为

$$d_1(\tilde{A})=\frac{2}{4}(|0.8-1|+|0.9-1|+|0.1-0|+|0.8-1|)=0.3$$

$$d_1(\tilde{B})=\frac{2}{4}(|0.3-0|+|0-0|+|0.3-0|+|0-0|)=0.3$$

欧几里得模糊度为

$$d_2(\tilde{A})=\frac{2}{\sqrt{4}}\sqrt{(0.8-1)^2+(0.9-1)^2+(0.1-0)^2+(0.8-1)^2}=0.316$$

$$d_2(\tilde{B})=\frac{2}{\sqrt{4}}\sqrt{(0.3-0)^2+(0-0)^2+(0.3-0)^2+(0-1)^2}=0.424$$

可见，按照海明模糊度计算，模糊集 $\tilde{A}$ 和 $\tilde{B}$ 的模糊度一样，而按照欧几里得模糊度计算，$d_2(\tilde{A})<d_2(\tilde{B})$。$d_1$ 采用线性运算虽然方便，但不能区分 $\tilde{A}$ 和 $\tilde{B}$ 的模糊度的大小，其误差较大。d_2 采用非线性运算，虽然较前者麻烦，但比较准确。

(2)熵模糊度

如果令 $H(\tilde{A})=-\sum_{i=1}^{n}\mu_{\tilde{A}}(x_i)\ln\mu_{\tilde{A}}(x_i)$，则熵模糊度的定义为

$$\begin{aligned}d_E(\tilde{A})&=\frac{1}{n\ln 2}\left[H(\tilde{A})+H(\tilde{A}^c)\right]\\&=\frac{1}{n\ln 2}\sum_{i=1}^{n}\left\{-\mu_{\tilde{A}}(x_i)\ln\mu_{\tilde{A}}(x_i)-[1-\mu_{\tilde{A}}(x_i)]\ln[1-\mu_{\tilde{A}}(x_i)]\right\}\end{aligned}\tag{9.12}$$

显然，各元素的隶属度越接近 0.5，$d_E(\tilde{A})$ 越大；如果每个 x_i 的隶属度 $\mu_{\tilde{A}}(x_i)=0.5$，则 $d_E(\tilde{A})$ 最大，$d_E(\tilde{A})=1$。

(3) 贴近度

由于用距离这个概念刻画一个模糊集的模糊度往往不会十分理想，而且需要比较麻烦的计算，为了弥补距离度量模糊集的模糊性的缺点，汪培庄等人提出了贴近度的概念。

贴近度用来衡量两个模糊集合之间的相近程度，贴近度越大，则表明这两者越接近。

【定义 9.5】 令 $\tilde{A},\tilde{B},\tilde{C}$ 为论域 $\boldsymbol{U}$ 中的模糊集合，若映射

$$N:\boldsymbol{U}\times\boldsymbol{U}\to[0,1]$$

满足条件：

- $N(\tilde{A},\tilde{B})=N(\tilde{B},\tilde{A})$；
- $N(\tilde{A},\tilde{A})=1,\ N(\boldsymbol{U},\varnothing)=0$；

● 若 $\tilde{A} \subseteq \tilde{B} \subseteq \tilde{C}$，则 $N(\tilde{A},\tilde{C}) \leqslant N(\tilde{A},\tilde{B}) \wedge N(\tilde{B},\tilde{C})$。

则称 $N(\tilde{A},\tilde{B})$ 为在 $\boldsymbol{U}$ 上的 $\tilde{A}$ 与 $\tilde{B}$ 的贴近度，N 称为在 $\boldsymbol{U}$ 上的贴近函数。

贴近度的这个定义是原则性的概念，其具体规则视实际需要而定。下面介绍几种常见的计算贴近度的方法。

(a) 格贴近度

若 $\boldsymbol{U} = \{u_1, u_2, \cdots, u_n\}$，则

$$\tilde{A} \circ \tilde{B} = \bigvee_{u_i \in \boldsymbol{U}} (\tilde{A}(u_i) \wedge \tilde{B}(u_i))$$

$$\tilde{A} \odot \tilde{B} = \bigwedge_{u_i \in \boldsymbol{U}} (\tilde{A}(u_i) \vee \tilde{B}(u_i))$$

分别称为 $\tilde{A}$ 与 $\tilde{B}$ 的内积和外积。

当 $\boldsymbol{U} = [a, b]$ 时，则

$$\tilde{A} \circ \tilde{B} = \bigvee_{u \in \boldsymbol{U}} (\tilde{A}(u) \wedge \tilde{B}(u))$$

$$\tilde{A} \odot \tilde{B} = \bigwedge_{u \in \boldsymbol{U}} (\tilde{A}(u) \vee \tilde{B}(u))$$

因此，格贴近度定义为

$$N(\tilde{A},\tilde{B}) = (\tilde{A} \circ \tilde{B}) \wedge (1 - \tilde{A} \odot \tilde{B})$$

【例 9.7】 设 $\boldsymbol{U} = \{a, b, c, d, e, f\}$，

$$\tilde{A} = \frac{0.6}{a} + \frac{0.8}{b} + \frac{1}{c} + \frac{0.8}{d} + \frac{0.6}{e} + \frac{0.4}{f}$$

$$\tilde{B} = \frac{0.4}{a} + \frac{0.6}{b} + \frac{0.8}{c} + \frac{1}{d} + \frac{0.8}{e} + \frac{0.6}{f}$$

试求 $\tilde{A}$ 与 $\tilde{B}$ 的格贴近度 $N(\tilde{A},\tilde{B})$。

解： $\tilde{A}$ 与 $\tilde{B}$ 的内积为

$$\begin{aligned}\tilde{A} \circ \tilde{B} &= (0.6 \wedge 0.4) \vee (0.8 \wedge 0.6) \vee (1 \wedge 0.8) \vee (0.8 \wedge 1) \vee (0.6 \wedge 0.8) \vee (0.4 \wedge 0.6) \\ &= 0.4 \vee 0.6 \vee 0.8 \vee 0.8 \vee 0.6 \vee 0.4 \\ &= 0.8\end{aligned}$$

$\tilde{A}$ 与 $\tilde{B}$ 的外积为

$$\begin{aligned}\tilde{A} \odot \tilde{B} &= (0.6 \vee 0.4) \wedge (0.8 \vee 0.6) \wedge (1 \vee 0.8) \wedge (0.8 \vee 1) \wedge (0.6 \vee 0.8) \wedge (0.4 \vee 0.6) \\ &= 0.6 \wedge 0.8 \wedge 1 \wedge 1 \wedge 0.8 \wedge 0.6 \\ &= 0.6\end{aligned}$$

$\tilde{A}$ 与 $\tilde{B}$ 的格贴近度为

$$N(\tilde{A},\tilde{B}) = (\tilde{A} \circ \tilde{B}) \wedge (1 - \tilde{A} \odot \tilde{B}) = 0.8 \wedge (1 - 0.6) = 0.4$$

(b) 海明贴近度

若 $\boldsymbol{U} = \{u_1, u_2, \cdots, u_n\}$，则

$$N_H(\tilde{A},\tilde{B})=1-\frac{1}{n}\sum_{i=1}^{n}\left|\tilde{A}(u_i)-\tilde{B}(u_i)\right|$$

当 $\boldsymbol{U}=[a,b]$时，则

$$N_H(\tilde{A},\tilde{B})=1-\frac{1}{b-a}\int_a^b\left|\tilde{A}(u)-\tilde{B}(u)\right|\mathrm{d}u$$

(c) 欧几里得贴近度

若 $\boldsymbol{U}=\{u_1,u_2,\cdots,u_n\}$，则

$$N_E(\tilde{A},\tilde{B})=1-\frac{1}{\sqrt{n}}\sqrt{\sum_{i=1}^{n}\left[\tilde{A}(u_i)-\tilde{B}(u_i)\right]^2}\ ;$$

当 $\boldsymbol{U}=[a,b]$时，则

$$N_E(\tilde{A},\tilde{B})=1-\frac{1}{\sqrt{b-a}}\sqrt{\int_a^b\left[\tilde{A}(u)-\tilde{B}(u)\right]^2\mathrm{d}u}$$

9.2　模糊模式识别的基本方法

模糊模式识别大致有两种方法：一是直接方法，按“最大隶属原则”进行归类，主要应用于个体的识别；二是间接方法，按“择近原则”进行归类，一般应用于群体模型的识别。

9.2.1　最大隶属原则

最大隶属原则是在模糊集合的基础上进行的个体模式识别。该原则简单直观，适用范围很广。

最大隶属原则

设 $\tilde{A}_1,\tilde{A}_2,\cdots,\tilde{A}_m$ 是论域 $\boldsymbol{U}$ 中的 m 个已知模糊集合，x 是 $\boldsymbol{U}$ 中的一个元素，若

$$\mu_{\tilde{A}_i}(x)=\max\left\{\mu_{\tilde{A}_1}(x),\mu_{\tilde{A}_2}(x),\cdots,\mu_{\tilde{A}_m}(x)\right\} \tag{9.13}$$

则认为 x 相对地隶属于 $\tilde{A}_i$。

【例 9.8】 考虑人的年龄问题，人的岁数作为论域 $\boldsymbol{U}=(0,120]$，单位是“岁”；可分为青年、中年、老年三类，分别对应三个模糊子集 $\tilde{A}_1$，$\tilde{A}_2$，$\tilde{A}_3$，其隶属函数如下：

$$\text{青年：}\ \mu_{\tilde{A}_1}(x)=\begin{cases}1, & 0<x\le 20\\ 1-2\left(\dfrac{x-20}{20}\right)^2, & 20<x\le 30\\ 2\left(\dfrac{x-40}{20}\right)^2 & 30<x\le 40\\ 0, & 40<x\end{cases}$$

$$\text{中年：}\ \mu_{\tilde{A}_2}(x)=\begin{cases}0, & 0<x\le 20\\ 2\left(\dfrac{x-20}{20}\right)^2, & 20<x\le 30\\ 1-2\left(\dfrac{x-40}{20}\right)^2, & 30<x\le 40\\ 1, & 40<x\le 50\\ 1-2\left(\dfrac{x-50}{20}\right)^2, & 50<x\le 60\\ 2\left(\dfrac{x-70}{20}\right)^2, & 60<x\le 70\\ 0, & 70<x\end{cases}$$

$$\text{老年：}\ \mu_{\tilde{A}_3}(x)=\begin{cases}0, & 0<x\le 50\\ 2\left(\dfrac{x-50}{20}\right)^2, & 50<x\le 60\\ 1-2\left(\dfrac{x-70}{20}\right)^2, & 60<x\le 70\\ 1, & 70<x\end{cases}$$

现有 22 岁、33 岁、65 岁各一人，问应该分别属于哪一类？

解：将 $x=22$ 代入三个隶属函数，有

$$\mu_{\tilde{A}_1}(22)=0.98,\ \ \mu_{\tilde{A}_2}(22)=0.02,\ \ \mu_{\tilde{A}_3}(22)=0$$

$$\max\{\mu_{\tilde{A}_1}(22),\ \mu_{\tilde{A}_2}(22),\ \mu_{\tilde{A}_3}(22)\}=\{0.98,0.02,0\}=0.98$$

所以 22 岁的人应属于青年人 $\tilde{A}_1$。

当 $x=33$ 时，$\mu_{\tilde{A}_1}(33)=0.25,\ \mu_{\tilde{A}_2}(33)=0.75,\ \mu_{\tilde{A}_3}(33)=0$，可见 30 岁的人可以认为属于中年人 $\tilde{A}_2$。

当 $x=65$ 时，$\mu_{\tilde{A}_1}(65)=0,\ \mu_{\tilde{A}_2}(65)=0.13,\ \mu_{\tilde{A}_3}(65)=0.87$，可见 65 岁的人属于老年人 $\tilde{A}_3$。

9.2.2 择近原则

当识别的对象不是特定模式，而是论域 $\boldsymbol{U}$ 中的一个模糊集合时，识别问题就变成了求模糊集合之间接近程度的问题。模糊集合之间的相似程度可以根据实际情况，选择使用距离模糊度或者贴近度。

择近原则

设 $\tilde{A}_i(i=1,2,\mathrm{L}\ ,n)$ 是论域 $\boldsymbol{U}$ 上的 n 个已知类别的模糊子集，待识别对象 $\tilde{B}$ 也是 $\boldsymbol{U}$ 上的模糊子集，若有 k 使得

$$N(\tilde{A}_k,\tilde{B})=\max_{1\le j\le n}\left[N(\tilde{A}_j,\tilde{B})\right] \tag{9.14}$$

则认为 $\tilde{B}$ 与 $\tilde{A}_k$ 最接近，$\tilde{B}$ 归入 $\tilde{A}_k$ 模式类。

【例 9.9】 反映茶叶质量的论域 $\boldsymbol{U}$ = {条索，色泽，净度，汤色，香气，滋味}，$\boldsymbol{U}$ 上的模糊子集有茶叶等级标准样品五种：

$$\tilde{A}_1 = (0.5, 0.4, 0.3, 0.6, 0.5, 0.4)$$

$$\tilde{A}_2 = (0.3, 0.2, 0.2, 0.1, 0.2, 0.2)$$

$$\tilde{A}_3 = (0.2, 0.2, 0.2, 0.1, 0.1, 0.2)$$

$$\tilde{A}_4 = (0, 0.1, 0.2, 0.1, 0.1, 0.1)$$

$$\tilde{A}_5 = (0, 0.1, 0.1, 0.1, 0.1, 0.1)$$

以及待识别的茶 $\tilde{A}_x = (0.4, 0.2, 0.1, 0.4, 0.5, 0.6)$，确定待识别茶叶的型号。

解 1：根据格贴近度计算公式 $N(\tilde{A}, \tilde{B}) = (\tilde{A} \circ \tilde{B}) \wedge (1 - \tilde{A} \odot \tilde{B})$，有

$$\tilde{A}_1 \cdot \tilde{A}_x = (0.5 \wedge 0.4) \vee (0.4 \wedge 0.2) \vee (0.3 \wedge 0.1) \vee (0.6 \wedge 0.4) \vee (0.5 \wedge 0.5) \vee (0.4 \wedge 0.6)$$
$$= 0.4 \vee 0.2 \vee 0.1 \vee 0.4 \vee 0.5 \vee 0.4 = 0.5$$

$$\tilde{A}_1 \oplus \tilde{A}_x = (0.5 \vee 0.4) \wedge (0.4 \vee 0.2) \wedge (0.3 \vee 0.1) \wedge (0.6 \vee 0.4) \wedge (0.5 \vee 0.5) \wedge (0.4 \vee 0.6)$$
$$= 0.5 \wedge 0.4 \wedge 0.3 \wedge 0.6 \wedge 0.5 \wedge 0.6 = 0.3$$

$$N(\tilde{A}_1, \tilde{A}_x) = 0.5 \wedge (1 - 0.3) = 0.5$$

$$\tilde{A}_2 \cdot \tilde{A}_x = (0.3 \wedge 0.4) \vee (0.2 \wedge 0.2) \vee (0.2 \wedge 0.1) \vee (0.1 \wedge 0.4) \vee (0.2 \wedge 0.5) \vee (0.2 \wedge 0.6)$$
$$= 0.3 \vee 0.2 \vee 0.1 \vee 0.1 \vee 0.2 \vee 0.2 = 0.3$$

$$\tilde{A}_2 \oplus \tilde{A}_x = (0.3 \vee 0.4) \wedge (0.2 \vee 0.2) \wedge (0.2 \vee 0.1) \wedge (0.1 \vee 0.4) \wedge (0.2 \vee 0.5) \wedge (0.2 \vee 0.6)$$
$$= 0.4 \wedge 0.2 \wedge 0.2 \wedge 0.4 \wedge 0.5 \wedge 0.6 = 0.2$$

$$N(\tilde{A}_2, \tilde{A}_x) = 0.3 \wedge (1 - 0.2) = 0.3$$

同理有

$$N(\tilde{A}_3, \tilde{A}_x) = 0.2 \wedge (1 - 0.2)] = 0.2$$

$$N(\tilde{A}_4, \tilde{A}_x) = 0.1 \wedge (1 - 0.2)] = 0.1$$

$$N(\tilde{A}_5, \tilde{A}_x) = 0.1 \wedge (1 - 0.1)] = 0.1$$

由择近原则，将茶叶 $\tilde{A}_x$ 确定为 $\tilde{A}_1$ 等级。

解 2：根据海明贴近度计算公式 $N_H(\tilde{A}, \tilde{B}) = 1 - \frac{1}{n}\sum_{i=1}^{n}\left|\tilde{A}(u_i) - \tilde{B}(u_i)\right|$，有

$$N_H(\tilde{A}_1, \tilde{A}_x)$$
$$= 1 - [|0.5 - 0.4| + |0.4 - 0.2| + |0.3 - 0.1| + |0.6 - 0.4| + |0.5 - 0.5| + |0.4 - 0.6|]/6$$
$$= 0.85$$

$$N_H(\tilde{A}_2, \tilde{A}_x)$$
$$= 1 - [|0.3 - 0.4| + |0.2 - 0.2| + |0.2 - 0.1| + |0.6 - 0.1| + |0.5 - 0.2| + |0.4 - 0.2|]/6$$
$$= 0.8$$

$$N_H(\tilde{A}_3, \tilde{A}_x)$$
$$= 1 - [|0.2 - 0.4| + |0.2 - 0.2| + |0.2 - 0.1| + |0.1 - 0.4| + |0.1 - 0.5| + |0.2 - 0.6|]/6$$
$$= 0.77$$

$$N_H(\tilde{A}_4, \tilde{A}_x)$$
$$= 1 - [|0.0 - 0.4| + |0.1 - 0.2| + |0.2 - 0.1| + |0.1 - 0.4| + |0.1 - 0.5| + |0.1 - 0.6|]/6$$
$$= 0.7$$

$$N_H(\tilde{A}_5, \tilde{A}_x)$$
$$= 1 - [|0.0 - 0.4| + |0.1 - 0.2| + |0.1 - 0.1| + |0.1 - 0.4| + |0.1 - 0.5| + |0.1 - 0.6|]/6$$
$$= 0.72$$

由择近原则，将茶叶 $\tilde{A}_x$ 确定为 $\tilde{A}_1$ 等级。

9.3 模糊聚类分析方法

在第 5 章中已经讨论了聚类分析方法，它是根据模式之间的相似程度和规定的聚类准则，按照最近邻原则对未知类别的模式进行分类。在模糊聚类分析方法中，通常根据分类对象之间的模糊相似程度来衡量互相的异同程度，进而实现模糊分类。

模糊聚类的实质是根据一定的客观要求对模糊模式进行分类；模式识别是将待识别的模式与模板进行比较，从而得到所属的类别。也就是说，模糊聚类事先不知道具体的分类类别，而模糊识别是在已知分类类别(模板)的情况下进行的，其中模糊聚类的分类结果为模糊识别提供模板。

9.3.1 基于模糊等价矩阵的聚类分析

在 9.1.2 节中介绍了模糊关系，当模糊关系是等价关系时，可以用模糊等价矩阵的截矩阵直接进行模式分类。

基于模糊等价矩阵的模糊聚类分析步骤如下：

- 数据标准化；
- 建立模糊相似关系；
- 在模糊相似关系基础上建立模糊等价关系；
- 根据参数λ的值进行模糊聚类。

1．数据标准化

在实际应用中，由于所获得的分类对象的数据比较复杂，往往不是[0,1]区间的数，因此需要将原始数据标准化。首先计算每一维特征的均值和方差：

$$\overline{x}'_k = \frac{1}{n}\sum_{i=1}^{n} x_{ik}, \qquad S_k^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{ik} - \overline{x}'_k)^2$$

然后将各个数据标准化，

$$x'_{ik} = \frac{x_{ik} - \overline{x}'_k}{S_k}$$

如果得到的标准化数据不在[0, 1]闭区间，再采用如下的极值标准化公式将数据压缩到闭区间[0, 1]：

$$x''_{ik} = \frac{x'_{ik} - x'_{k\min}}{x'_{k\max} - x'_{k\min}}$$

2．建立模糊相似关系

由于模糊等价关系在模糊聚类分析中起重要作用，为了建立分类对象的模糊等价关系，需要计算各个分类对象之间的相似性统计量，建立分类对象集合 X 上的模糊相似关系 $R_S = [r_{ij}]_{n\times n}, 0 \leqslant r_{ij} \leqslant 1 (i, j = 1, 2, \cdots, n)$，$r_{ij}$ 表示分类对象 x_i 与 x_j 的相似程度。计算 r_{ij} 的常用方法主要有以下几种。

(1)夹角余弦法

$$r_{ij} = \frac{\sum_{k=1}^{m} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{m} x_{ik}^2 \sum_{k=1}^{m} x_{jk}^2}}$$

其中 x_{ik}, x_{jk} 分别表示 x_i, x_j 的第 k 维特征，$k = 1, 2, \cdots, m$。

(2)数量积法

$$r_{ij} = \begin{cases} 1, & i = j \\ \dfrac{1}{M} \sum_{k=1}^{m} x_{ik} x_{jk}, & i \neq j \end{cases}$$

其中 M 是一个适当选择的正数，满足

$$M \leqslant \max_{i,j} \{\sum_{k=1}^{m} x_{ik} x_{jk}\}$$

(3)相关系数法

$$r_{ij} = \frac{\sum_{k=1}^{m} (x_{ik} - \overline{x}_i)(x_{jk} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{m} (x_{ik} - \overline{x}_i)^2 \sum_{k=1}^{m} (x_{jk} - \overline{x}_j)^2}}$$

其中 $\overline{x}_i = \dfrac{1}{m} \sum_{k=1}^{m} x_{ik}$，$\overline{x}_j = \dfrac{1}{m} \sum_{k=1}^{m} x_{jk}$。

(4)最大最小法

$$r_{ij} = \frac{\sum_{k=1}^{m} \min(x_{ik}, x_{jk})}{\sum_{k=1}^{m} \max(x_{ik}, x_{jk})}$$

(5)算数平均最小法

$$r_{ij}=\frac{\sum_{k=1}^{m}\min(x_{ik},x_{jk})}{\frac{1}{2}\sum_{k=1}^{m}(x_{ik}+x_{jk})}$$

(6)几何平均最小法

$$r_{ij}=\frac{\sum_{k=1}^{m}\min(x_{ik},x_{jk})}{\sum_{k=1}^{m}\sqrt{x_{ik}x_{jk}}}$$

(7)绝对值指数法

$$r_{ij}=\mathrm{e}^{-\sum_{k=1}^{m}|x_{ik}-x_{jk}|}$$

(8)绝对值倒数法

$$r_{ij}=\begin{cases}1, & i=j\\ \dfrac{M}{\sum_{k=1}^{m}|x_{ik}-x_{jk}|}, & i\neq j\end{cases}$$

其中 M 是一个适当选取的数，使得 $0\leqslant r_{ij}\leqslant 1$。

(9)绝对值减数法

$$r_{ij}=1-c\cdot\sum_{k=1}^{m}|x_{ik}-x_{jk}|$$

其中 $c>0$ 为常数，其值可以根据实际要求选定，使得 $r_{ij}\in[0,1]$。

3．用传递闭包法建立模糊等价关系

如果模糊矩阵 $\boldsymbol{R}$ 只是一个模糊相似矩阵，那么它不一定具有传递性，即 $\boldsymbol{R}$ 不一定是模糊等价矩阵。为了进行分类，需要将 $\boldsymbol{R}$ 改造成模糊等价矩阵。

【定理 9.3】 设 $\boldsymbol{R}\in\mu_{n\times n}$ 是模糊相似矩阵，则存在一个最小自然数 $k(k\leqslant n)$，使得传递闭包 $\boldsymbol{t}(\boldsymbol{R})=\boldsymbol{R}^k$，对于一切大于 k 的自然数 l，恒有 $\boldsymbol{R}^l=\boldsymbol{R}^k$。此时，$\boldsymbol{t}(\boldsymbol{R})$ 为模糊等价矩阵。

定理 9.3 表明，通过求传递闭包 $\boldsymbol{t}(\boldsymbol{R})$，可将模糊相似矩阵改造成为模糊等价矩阵。求传递闭包的简捷方法是平方法，即从模糊相似矩阵出发，依次求平方：

$$\boldsymbol{R}\to\boldsymbol{R}^2\to\boldsymbol{R}^4\to\cdots\to\boldsymbol{R}^{2^i}\to\cdots$$

当第一次出现 $\boldsymbol{R}^k\circ\boldsymbol{R}^k=\boldsymbol{R}^k$ 时，表明 $\boldsymbol{R}^k$ 具有传递性，$\boldsymbol{R}^k$ 就是所求的传递闭包 $\boldsymbol{t}(\boldsymbol{R})$。

【例 9.10】 设

$$\boldsymbol{R}=\begin{bmatrix}1 & 0.2 & 0.8 & 0.5 & 0.3\\ 0.2 & 1 & 0.1 & 0.2 & 0.4\\ 0.8 & 0.1 & 1 & 0.3 & 0.1\\ 0.5 & 0.2 & 0.3 & 1 & 0.6\\ 0.3 & 0.4 & 0.1 & 0.6 & 1\end{bmatrix}$$

求 $\boldsymbol{R}$ 的传递闭包 $\boldsymbol{t}(\boldsymbol{R})$。

解：矩阵显然具有自反性和对称性，$\boldsymbol{R}$ 是模糊相似矩阵，由

$$\boldsymbol{R}^2=\boldsymbol{R}\circ\boldsymbol{R}=\begin{bmatrix}1 & 0.3 & 0.8 & 0.5 & 0.5\\ 0.3 & 1 & 0.2 & 0.4 & 0.4\\ 0.8 & 0.2 & 1 & 0.5 & 0.3\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.3 & 0.6 & 1\end{bmatrix}$$

$$\boldsymbol{R}^4=\boldsymbol{R}^2\circ\boldsymbol{R}^2=\begin{bmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{bmatrix}$$

$$\boldsymbol{R}^8=\boldsymbol{R}^4\circ\boldsymbol{R}^4=\boldsymbol{R}^4$$

于是 $\boldsymbol{t}(\boldsymbol{R})=\boldsymbol{R}^4$。

4．模糊等价关系的截矩阵分类法

在模糊等价矩阵的基础上，根据参数λ的值，求出其λ截矩阵，并进行分类。

【例 9.11】 某环保单位要对 5 个地方的环境进行分类，其环境质量的好坏是由污染物在空气、水、土壤、植被中含量的超限度标定的，如表 9.2 所示。求$\lambda=0.6$ 时的分类结果。

表 9.2　5 个地方的环境污染数据

地方	空气	水	土壤	植被
A	5	5	3	2
B	2	3	4	5
C	5	5	2	3
D	1	5	3	1
E	2	4	5	1

解：用不同的数学公式将得到不同的模糊相似矩阵，这里选用绝对值减数法计算得出模糊相似矩阵，$r_{ij}=1-c\cdot\sum_{k=1}^{m}|x_{ik}-x_{jk}|$，其中 $c=0.1, m=4$。

$$\begin{aligned}r_{11}&=1\\ r_{12}&=1-0.1\times\{|x_{11}-x_{21}|+|x_{12}-x_{22}|+|x_{13}-x_{23}|+|x_{14}-x_{24}|\}\\ &=1-0.1\times\{|5-2|+|5-3|+|3-4|+|2-5|\}\\ &=0.1\end{aligned}$$

依次计算得到模糊相似矩阵为

$$\boldsymbol{R}=\begin{bmatrix}1 & 0.1 & 0.8 & 0.5 & 0.3\\ 0.1 & 1 & 0.1 & 0.2 & 0.4\\ 0.8 & 0.1 & 1 & 0.3 & 0.1\\ 0.5 & 0.2 & 0.3 & 1 & 0.6\\ 0.3 & 0.4 & 0.1 & 0.6 & 1\end{bmatrix}$$

$\boldsymbol{R}$ 仅是一个模糊相似矩阵，要将其改造成模糊等价关系，需求出其传递闭包 $\boldsymbol{t}(\boldsymbol{R})$。

$$\boldsymbol{R}^2=\boldsymbol{R}\circ\boldsymbol{R}=\begin{bmatrix}1 & 0.3 & 0.8 & 0.5 & 0.5\\ 0.3 & 1 & 0.2 & 0.4 & 0.4\\ 0.8 & 0.2 & 1 & 0.5 & 0.3\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.3 & 0.6 & 1\end{bmatrix}$$

$$\boldsymbol{R}^4=\boldsymbol{R}^2\circ\boldsymbol{R}^2=\begin{bmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{bmatrix}$$

$$\boldsymbol{R}^8=\boldsymbol{R}^4\circ\boldsymbol{R}^4=\boldsymbol{R}^4$$

于是 $\boldsymbol{t}(\boldsymbol{R})=\boldsymbol{R}^4$，$\boldsymbol{R}^4$ 为模糊等价矩阵。当$\lambda=0.6$ 时，可分为 3 类：$\{A, C\}, \{B\}, \{D, E\}$。

基于模糊等价矩阵的聚类分析，通常需要求出模糊相似矩阵的传递闭包；当矩阵阶数较高时，求模糊等价矩阵的计算量很大。

9.3.2 模糊*C*均值聚类算法

模糊 C 均值(Fuzzy C-Means, FCM)聚类算法是由 K 均值聚类算法派生而来的。在 K 均值聚类算法中的每一步迭代过程，每一个样本都被认为是完全属于某一类或完全不属于某一类。J. C. Bezdek[2]利用模糊集合的概念提出了模糊分类，被分类的每一个样本被认为以不同的隶属度属于某一类。

给定样本集 $\boldsymbol{X}=\{x_1, x_2, \cdots, x_N\}$有 N 个待分类的特征向量，每个特征向量有 k 个特征，欲将这 N 个样本分成 C 类(ω_i, $i=1, 2, \cdots, C$)，其聚类中心为 $V=\{v_1, v_2, \cdots, v_C\}$, $2\leqslant C\leqslant N$，分类结果用模糊分类矩阵 $\boldsymbol{U}=(\mu_{ij})_{C\times N}$ 表示，其中 μ_{ij} 表示 x_j ($j=1, 2, \cdots, N$)属于 ω_i ($i=1, 2, \cdots, C$)类的程度。因此，$\boldsymbol{U}$ 也称为隶属度矩阵或 $\boldsymbol{X}$ 的 C 模糊划分矩阵，μ_{ij} 应该满足

① $\mu_{ij}\in[0, 1]$;

② $0<\sum_{j=1}^{N}\mu_{ij}<N$，$\forall i$，即每个 ω_i 不等于空集$\varnothing$或全集 $\boldsymbol{X}$;

③ $\sum_{i=1}^{C}\mu_{ij}=1$，$\forall j$，每一个模式 x_j 属于各类的总和为1。

FCM 算法在迭代寻优过程中，不断更新各类的中心及隶属度矩阵各元素的值，直到逼近下列准则函数(也称为目标函数)的最小值：

$$J_{\mathrm{m}}(\boldsymbol{U}, \boldsymbol{V}) = \sum_{j=1}^{N}\sum_{i=1}^{C}\mu_{ij}^{m}d_{ij}^{2} \tag{9.15}$$

式中，模糊性加权指数 $m(m > 1)$是用来控制聚类结果模糊程度的常数，通常取 $1 < m \leqslant 5$；模式 x_j 到 ω_i 类中心向量的距离平方 $d_{ij}^2 = (x_j - v_i)^{\mathrm{T}}\boldsymbol{A}(x_j - v_i)$，$\boldsymbol{A}$ 为对称正定矩阵；当 $\boldsymbol{A}$ 取单位矩阵 $\boldsymbol{I}$ 时，d_{ij} 为欧氏距离，即 $d_{ij}^2 = \left\|x_j - v_i\right\|^2$。

式(9.15)的约束条件为

$$\sum_{i=1}^{C}\mu_{ij} = 1, \qquad \forall j \tag{9.16}$$

运用拉格朗日乘数法，可得无约束的准则函数

$$F = \sum_{j=1}^{N}\sum_{i=1}^{C}\mu_{ij}^{m}d_{ij}^{2} - \sum_{j=1}^{N}\lambda_j\left(\sum_{i=1}^{C}\mu_{ij} - 1\right) \tag{9.17}$$

式(9.17)取极小值的必要条件是

$$\frac{\partial F}{\partial \mu_{ij}} = m\mu_{ij}^{m-1}d_{ij}^{2} - \lambda_j = 0 \tag{9.18}$$

$$\frac{\partial F}{\partial \lambda_j} = -\left(\sum_{i=1}^{c}\mu_{ij} - 1\right) = 0 \tag{9.19}$$

由式(9.18)可得

$$\mu_{ij} = \left(\frac{\lambda_j}{md_{ij}^2}\right)^{\frac{1}{m-1}} = \left(\frac{\lambda_j}{m}\right)^{\frac{1}{m-1}}\frac{1}{(d_{ij}^2)^{\frac{1}{m-1}}} \tag{9.20}$$

将式(9.20)代入式(9.16)，可得

$$\sum_{i=1}^{C}\mu_{ij} = \left(\frac{\lambda_j}{m}\right)^{\frac{1}{m-1}}\sum_{i=1}^{C}\frac{1}{(d_{ij}^2)^{\frac{1}{m-1}}} = 1$$

从而有

$$\left(\frac{\lambda_j}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{i=1}^{C}\frac{1}{(d_{ij}^2)^{\frac{1}{m-1}}}} = \frac{1}{\sum_{k=1}^{C}\frac{1}{(d_{kj}^2)^{\frac{1}{m-1}}}} \tag{9.21}$$

将式(9.21)代入(9.20)，得

$$\mu_{ij}=\frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \tag{9.22}$$

考虑到对于一些 j，d_{ij} 可能为 0，定义集合 I_j 和 I_j'：

$$I_j=\{i|1\leqslant i\leqslant C, d_{ij}=0\},\quad I_j'=\{1,2,\cdots,C\}-I_j$$

如果 $I_j=\varnothing$，则

$$\mu_{ij}=\frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}}$$

如果 $I_j\neq\varnothing$，则令 $\mu_{ij}=0$，$\forall i\in I_j'$，并使 $\sum_{i\in I_j}\mu_{ij}=1$。

用类似的方法可以获得 $J_m(\boldsymbol{U},\boldsymbol{V})$ 为最小时 v_i 的值。令

$$\frac{\partial J_m(\boldsymbol{U},\boldsymbol{V})}{\partial v_i}=0$$

可得

$$\sum_{j=1}^{N}\mu_{ij}^m\frac{\partial}{\partial v_i}[(x_j-v_i)^{\mathrm{T}}\boldsymbol{A}(x_j-v_i)]=0$$

$$\sum_{j=1}^{N}\mu_{ij}^m[-2\boldsymbol{A}(x_j-v_i)]=0$$

由此可得

$$v_i=\frac{\sum_{j=1}^{N}\mu_{ij}^m x_j}{\sum_{j=1}^{N}\mu_{ij}^m} \tag{9.23}$$

FCM 算法的步骤如下：

(1) 已知样本集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_N\}$，确定类别数 $C(2\leqslant C\leqslant N)$、模糊性加权指数 m、矩阵 $\boldsymbol{A}$ 和一个适当小的迭代停止阈值 ε；

(2) 设置初始模糊分类矩阵 $\boldsymbol{U}^{(s)}$，令迭代次数 $s=0$；

(3) 计算 $\boldsymbol{U}^{(s)}$ 时的聚类中心 $v_i^{(s)}$：

$$v_i^{(s)}=\frac{\sum_{j=1}^{N}\mu_{ij}^m x_j}{\sum_{j=1}^{N}\mu_{ij}^m},\quad i=1,2,\cdots,C$$

(4) 按下面方法将 $\boldsymbol{U}^{(s)}$ 更新为 $\boldsymbol{U}^{(s+1)}$：

(a) 计算 I_j 和 I_j'，其中 $j = 1, 2, \cdots, N$

$I_j = \{i | 1 \leqslant i \leqslant C, d_{ij} = 0\}$

$I_j' = \{1, 2, \cdots, C\} - I_j$

(b) 计算 x_j 的新隶属度

如果 $I_j = \varnothing$，则

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}}$$

否则，若 $I_j \neq \varnothing$，令 $\mu_{ij} = 0$，$\forall i \in I_j'$，并使 $\sum_{i \in I_j} \mu_{ij} = 1$。

(5) 以一个适当的矩阵范数比较 $\boldsymbol{U}^{(s)}$ 和 $\boldsymbol{U}^{(s+1)}$，如果 $\| \boldsymbol{U}^{(s)} - \boldsymbol{U}^{(s+1)} \| < \varepsilon$，则停止；否则，$s = s + 1$，返到第 (3) 步。

在上述 FCM 算法中，模式类用一点代表，点到模式类的距离采用加权欧氏距离。

该算法也有另一种形式，即初始化聚类中心，计算模糊分类矩阵，然后更新聚类中心，直到满足停止准则为止。

FCM 算法能从任意给定初始点开始沿一个迭代子序列收敛到目标函数 $J_m(\boldsymbol{U}, \boldsymbol{V})$ 的局部极小点或鞍点。

FCM 算法得到的最优分类矩阵 $\boldsymbol{U}$ 是模糊矩阵，对应的分类也是模糊分类，要得到样本集 $\boldsymbol{X} = \{x_1, x_2, \cdots, x_N\}$ 的硬分类，可用如下方法：

- x_j $(j = 1, 2, \cdots, N)$ 与哪一个聚类中心最接近，就将它归到哪一类；
- x_j $(j = 1, 2, \cdots, N)$ 对哪一个类的隶属度最大，就将它归到哪一类，这实际上就是最大隶属原则。

【例 9.12】 如图9.3所示，设有 10 个二维样本分别是 $x_1 = [0, 0]^{\mathrm{T}}$, $x_2 = [0, 1]^{\mathrm{T}}$, $x_3 = [1, 0]^{\mathrm{T}}$, $x_4 = [1, 1]^{\mathrm{T}}$, $x_5 = [5, 2]^{\mathrm{T}}$, $x_6 = [5, 3]^{\mathrm{T}}$, $x_7 = [6, 2]^{\mathrm{T}}$, $x_8 = [3, 5]^{\mathrm{T}}$, $x_9 = [3, 6]^{\mathrm{T}}$, $x_{10} = [4, 5]^{\mathrm{T}}$。

取模糊性加权指数 $m = 2$、欧氏距离，利用 FCM 算法将数据样本集分为三类。

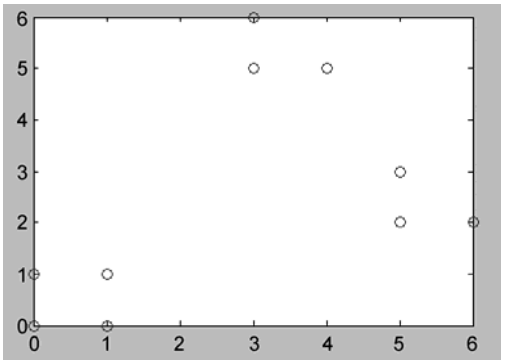


图 9.3　例 9.12 的 10 个数据样本

解：

(1) 根据题意有：样本数 $N = 10$，类别数 $C = 3$、模糊性加权指数 $m = 2$、矩阵 $\boldsymbol{A}$ 为单位矩阵 $\boldsymbol{I}$，迭代停止阈值 $\varepsilon = \mathrm{e}^{-5}$。

(2) 用随机函数设置初始模糊分类矩阵 $\boldsymbol{U}^{(0)}$：

$$\boldsymbol{U}^{(0)} = \begin{bmatrix} 0.4256 & 0.3940 & 0.4150 & 0.1883 & 0.6590 & 0.4487 & 0.6676 & 0.1964 & 0.1665 & 0.3063 \\ 0.0232 & 0.2863 & 0.2190 & 0.7767 & 0.1099 & 0.3786 & 0.0242 & 0.3941 & 0.4364 & 0.3361 \\ 0.5512 & 0.3197 & 0.3660 & 0.0350 & 0.2311 & 0.1728 & 0.3082 & 0.4094 & 0.3971 & 0.3576 \end{bmatrix}$$

(3) 计算 $\boldsymbol{U}^{(0)}$ 时的聚类中心 $v_i^{(0)}$，其聚类中心 $\boldsymbol{v}^{(0)}$ 为

$$\boldsymbol{v}^{(0)} = \begin{bmatrix} 3.7154 & 1.8947 \\ 2.1662 & 2.6881 \\ 2.2249 & 2.4844 \end{bmatrix}$$

(4) 计算新隶属度 $\boldsymbol{U}^{(1)}$ 和聚类中心 $\boldsymbol{v}^{(1)}$：

$$\boldsymbol{U}^{(1)} = \begin{bmatrix} 0.2486 & 0.2009 & 0.2699 & 0.1942 & 0.7119 & 0.5835 & 0.5862 & 0.2411 & 0.2612 & 0.3182 \\ 0.3627 & 0.3890 & 0.3446 & 0.3771 & 0.1391 & 0.2062 & 0.2021 & 0.4054 & 0.3889 & 0.3554 \\ 0.3887 & 0.4101 & 0.3856 & 0.4286 & 0.1490 & 0.2103 & 0.2117 & 0.3534 & 0.3500 & 0.3264 \end{bmatrix}$$

$$\boldsymbol{v}^{(1)} = \begin{bmatrix} 4.4095 & 2.4562 \\ 2.0829 & 2.6662 \\ 1.8820 & 2.2481 \end{bmatrix}$$

(5) 重复上述过程，共进行 9 次迭代，满足收敛条件，最后得到的隶属度矩阵和聚类中心为

$$\boldsymbol{U} = \begin{bmatrix} 0.0142 & 0.0160 & 0.0200 & 0.0236 & 0.9763 & 0.9155 & 0.9556 & 0.0161 & 0.0276 & 0.0593 \\ 0.0122 & 0.0162 & 0.0143 & 0.0201 & 0.0147 & 0.0644 & 0.0284 & 0.9762 & 0.9580 & 0.9243 \\ 0.9736 & 0.9677 & 0.9656 & 0.9563 & 0.0091 & 0.0201 & 0.0160 & 0.0077 & 0.0144 & 0.0164 \end{bmatrix}$$

$$\boldsymbol{v} = \begin{bmatrix} 5.3326 & 2.3138 \\ 3.3160 & 5.3303 \\ 0.4964 & 0.4972 \end{bmatrix}$$

根据隶属度矩阵可知，10 个样本的分类结果为 $\{x_1, x_2, x_3, x_4\}$ 属于第三类，$\{x_5, x_6, x_7\}$ 属于第一类，$\{x_8, x_9, x_{10}\}$ 属于第二类，如图9.4所示。

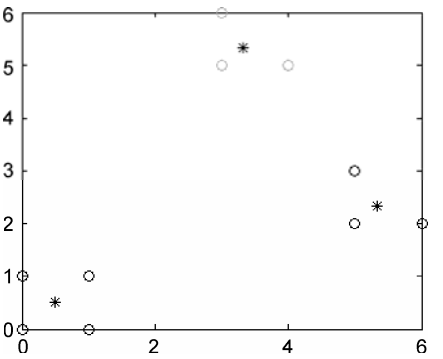


图 9.4 例 9.12 的分类结果

9.3.3 模糊聚类的有效性

模糊聚类的有效性指标有：划分指标、划分熵、比例指数[3]、分离指标、平均划分密度等，其中常用的有划分指标和划分熵。

1．划分指标

【定义 9.12】 对于给定的聚类中心 C 和隶属度矩阵 $\boldsymbol{U}$，划分指标定义为[5]

$$F(\boldsymbol{U}, C) = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}^2 \tag{9.24}$$

其中，N 为待分析的样本数据的分数。

$F(\boldsymbol{U}, C)$ 具有如下性质：

- $1/C \leqslant F(\boldsymbol{U}, C) \leqslant 1$；
- 当 $\boldsymbol{U}$ 是硬划分时，$F(\boldsymbol{U}, C) = 1$；
- 当 $\mu_{ij} = 1/C$（$i = 1, \cdots, C; j = 1, \cdots, N$）时，$F(\boldsymbol{U}, C) = 1/C$。

对于不同的 C 和 $\boldsymbol{U}$，若存在 $(\boldsymbol{U}^*, C^*)$ 使 $F(\boldsymbol{U}, C)$ 最大，则 $(\boldsymbol{U}^*, C^*)$ 为最佳有效性聚类，C^* 为最好的分类数。

2．划分熵

【定义 9.13】 对于给定的聚类中心 C 和隶属度矩阵 $\boldsymbol{U}$，划分熵定义为[5]

$$H(\boldsymbol{U}, C) = -\frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}\cdot\ln(\mu_{ij}), \qquad \mu_{ij} \neq 0 \tag{9.25}$$

其中，当 $\mu_{ij} = 0$ 时，令 $\mu_{ij}\ln(\mu_{ij}) = 0$。

$H(\boldsymbol{U}, C)$ 具有如下性质：

- $0 \leqslant H(\boldsymbol{U}, C) \leqslant \ln(C)$；
- 当 $\boldsymbol{U}$ 是硬划分时，$H(\boldsymbol{U}, C) = 0$；
- 当 $\mu_{ij} = 1/C$（$i = 1, \cdots, C; j = 1, \cdots, N$）时，$H(\boldsymbol{U}, C) = \ln(C)$。

对于不同的 C 和 $\boldsymbol{U}$，若存在 $(\boldsymbol{U}^*, C^*)$ 使 $H(\boldsymbol{U}, C)$ 最小，则 $(\boldsymbol{U}^*, C^*)$ 为最佳有效性聚类，C^* 为最好的分类数。

3．可能性划分系数

【定义 9.14】 对于给定的聚类数 C 和隶属度矩阵 $\boldsymbol{U}$，可能性划分系数定义为[4]

$$P(\boldsymbol{U}, C) = \frac{1}{C}\sum_{i=1}^{C}\left(\sum_{j=1}^{N}\mu_{ij}^2 \Big/ \sum_{j=1}^{N}\mu_{ij}\right), \qquad \mu_{ij} \neq 0 \tag{9.26}$$

$P(\boldsymbol{U}, C)$ 具有如下性质：

- $0 \leqslant P(\boldsymbol{U}, C) \leqslant 1$；
- 当 $\boldsymbol{U}$ 是硬划分时，$P(\boldsymbol{U}, C) = 1$；
- 当 $\mu_{ij} = 1/C$（$i = 1, \cdots, C; j = 1, \cdots, N$）时，$P(\boldsymbol{U}, C) = 1/C$。

对于不同的 C 和 $\boldsymbol{U}$，若存在 $(\boldsymbol{U}^*, C^*)$ 使 $P(\boldsymbol{U}, C)$ 最大，则 $(\boldsymbol{U}^*, C^*)$ 为最佳有效性聚类，C^* 为最好的分类数。

尽管上述指标可以用来确定最优类别数，但有时得到的结果会相互矛盾，建议将其给出的结果作为参考。

【例 9.13】 如图9.5所示，设有 68 个二维样本：

$x_1=[0,0]^{\mathrm{T}}$, $x_2=[2,0]^{\mathrm{T}}$, $x_3=[0,2]^{\mathrm{T}}$, $x_4=[2,2]^{\mathrm{T}}$, $x_5=[3,1]^{\mathrm{T}}$, $x_6=[1,1]^{\mathrm{T}}$
$x_7=[2,1]^{\mathrm{T}}$, $x_8=[0,8]^{\mathrm{T}}$, $x_9=[2,10]^{\mathrm{T}}$, $x_{10}=[0,10]^{\mathrm{T}}$, $x_{11}=[1,9]^{\mathrm{T}}$, $x_{12}=[0,11]^{\mathrm{T}}$
$x_{13}=[1,10]^{\mathrm{T}}$, $x_{14}=[0,9]^{\mathrm{T}}$, $x_{15}=[2,9]^{\mathrm{T}}$, $x_{16}=[5,6]^{\mathrm{T}}$, $x_{17}=[6,5]^{\mathrm{T}}$, $x_{18}=[6,4]^{\mathrm{T}}$
$x_{19}=[7,6]^{\mathrm{T}}$, $x_{20}=[7,4]^{\mathrm{T}}$, $x_{21}=[8,5]^{\mathrm{T}}$, $x_{22}=[6,6]^{\mathrm{T}}$, $x_{23}=[7,5]^{\mathrm{T}}$, $x_{24}=[8,6]^{\mathrm{T}}$
$x_{25}=[21,1]^{\mathrm{T}}$, $x_{26}=[20,3]^{\mathrm{T}}$, $x_{27}=[22,3^{\mathrm{T}}$, $x_{28}=[21,2]^{\mathrm{T}}$, $x_{29}=[21,3]^{\mathrm{T}}$, $x_{30}=[20,2]^{\mathrm{T}}$
$x_{31}=[22,2]^{\mathrm{T}}$, $x_{32}=[20,1]^{\mathrm{T}}$, $x_{33}=[25,0]^{\mathrm{T}}$, $x_{34}=[29,0]^{\mathrm{T}}$, $x_{35}=[27,1]^{\mathrm{T}}$, $x_{36}=[26,2]^{\mathrm{T}}$
$x_{37}=[28,0]^{\mathrm{T}}$, $x_{38}=[27,0]^{\mathrm{T}}$, $x_{39}=[27,2]^{\mathrm{T}}$, $x_{40}=[26,1]^{\mathrm{T}}$, $x_{41}=[28,1]$, $x_{42}=[27,6]^{\mathrm{T}}$
$x_{43}=[29,6]^{\mathrm{T}}$, $x_{44}=[28,8]^{\mathrm{T}}$, $x_{45}=[30,7]^{\mathrm{T}}$, $x_{46}=[30,9]^{\mathrm{T}}$, $x_{47}=[30,8]^{\mathrm{T}}$, $x_{48}=[27,8]^{\mathrm{T}}$
$x_{49}=[27,7]^{\mathrm{T}}$, $x_{50}=[28,6]^{\mathrm{T}}$, $x_{51}=[28,7]^{\mathrm{T}}$, $x_{52}=[29,7]^{\mathrm{T}}$, $x_{53}=[10,16]^{\mathrm{T}}$, $x_{54}=[10,18]^{\mathrm{T}}$
$x_{55}=[12,17]^{\mathrm{T}}$, $x_{56}=[12,18]^{\mathrm{T}}$, $x_{57}=[11,17]^{\mathrm{T}}$, $x_{58}=[10,17^{\mathrm{T}}$, $x_{59}=[11,16]^{\mathrm{T}}$, $x_{60}=[18,14]^{\mathrm{T}}$
$x_{61}=[19,15]^{\mathrm{T}}$, $x_{62}=[20,14]^{\mathrm{T}}$, $x_{63}=[20,16]^{\mathrm{T}}$, $x_{64}=[21,15]^{\mathrm{T}}$, $x_{65}=[22,16]^{\mathrm{T}}$,$x_{66}=[22,15]$
$x_{67}=[19,14]^{\mathrm{T}}$, $x_{68}=[20,15]$

取模糊性加权指数 $m=2$、欧氏距离，利用 FCM 算法将数据样本集分类，画出类别数($2\leqslant C\leqslant 10$)与目标函数的关系曲线。分别给出类别数是 3 和 8 时的聚类结果。用划分指标和划分熵综合确定最佳类别数。

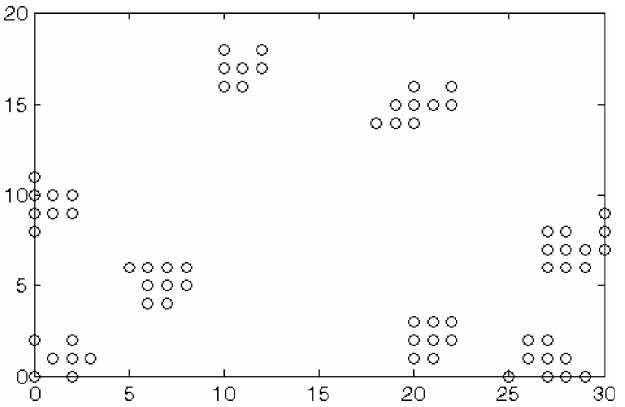


图 9.5 例 9.13 的 68 个数据样本

解：

根据题意有：样本数 $N=68$，模糊性加权指数 $m=2$，矩阵 $\boldsymbol{A}$ 为单位矩阵 $\boldsymbol{I}$，迭代停止阈值 $\varepsilon=\mathrm{e}^{-5}$。

当类别数 $C=3$ 时，分类结果如图9.6所示，此时的聚类中心如图 9.6 中的“*”所示，具体值是：

第一类：16.6274　15.3507
第二类：3.3001　5.4493
第三类：26.0663　3.6764

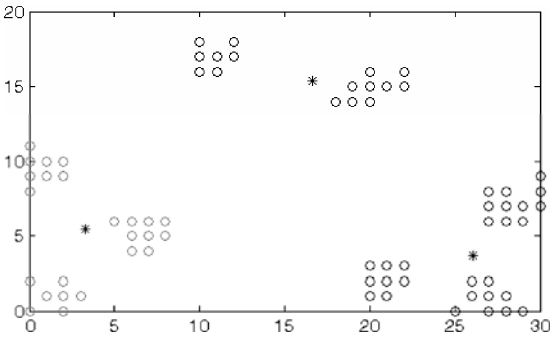


图 9.6　类别数为 3 时的分类结果

当类别数 $C = 8$ 时，分类结果如图9.7所示，此时的聚类中心如图9.7中的“*”所示，具体值是：

第一类：0.7560　9.5039

第二类：27.0785　0.8174

第三类：10.8593　16.9840

第四类：28.4646　7.1651

第五类：20.8939　2.1195

第六类：20.1277　14.8958

第七类：1.4548　1.0005

第八类：6.7036　5.2082

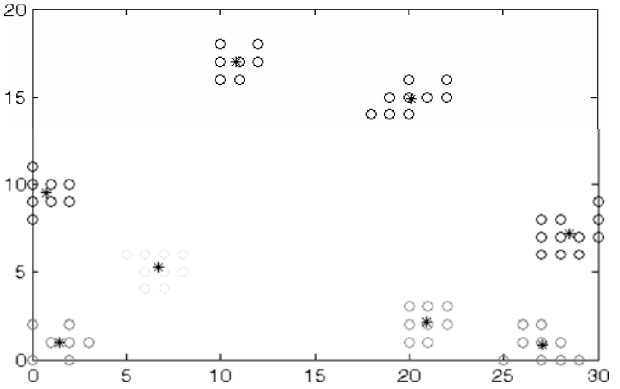


图 9.7　类别数为 8 时的分类结果

不同类别数与目标函数的关系如图9.8所示，具体值如表9.3所示。

表 9.3　类别数与目标函数的关系

类别数 C	目标函数 J_m
2	2613.797045
3	1224.631683
4	865.582439
5	577.250844
6	340.007437
7	215.076553
8	103.803073
9	91.282748
10	81.130884

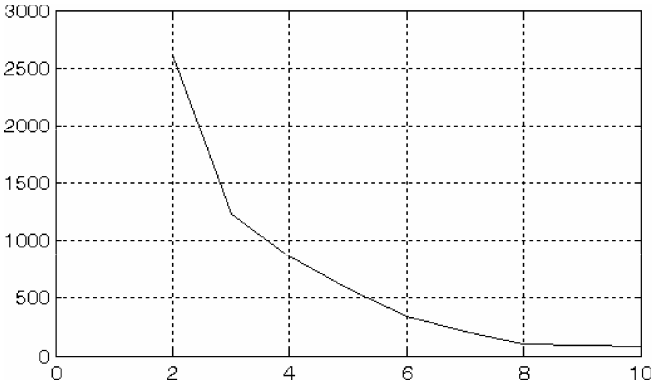


图 9.8 类别数 C 与准则函数之间的关系曲线

用式(9.24)~式(9.26)分别求出不同类别数时的划分指标、划分熵和可能性划分系数，具体如表 9.4 所示。

表 9.4 类别数与聚类有效性指标的关系

类别数 C	划分指标	可能性划分系数	划分熵	归一化划分熵 $H(\boldsymbol{U}, C)/\ln C$
2	0.831929	0.831211	0.287828	0.415248
3	0.791866	0.780326	0.416527	0.379139
4	0.725698	0.714829	0.553782	0.399469
5	0.709301	0.708499	0.595195	0.369815
6	0.752329	0.758127	0.537395	0.299926
7	0.806631	0.817127	0.453683	0.233147
8	0.859648	0.860852	0.360096	0.17317
9	0.833599	0.821979	0.411308	0.187194
10	0.800701	0.780219	0.475876	0.20667

根据不同类别数 C 对应的划分指标和可能性划分系数，可知在 $C=8$ 时 $F(\boldsymbol{U}, C)$ 和 $P(\boldsymbol{U}, C)$ 取得最大值，此时的类别数为最佳分类数。

根据不同类别数 C 对应的划分熵，可知在 $C=2$ 时 $H(\boldsymbol{U}, C)$ 取得最小值；在 $C=3$ 时，$H(\boldsymbol{U}, C)$ 取得第二个最小值。由于划分熵的值与 C 相关，当 C 取值小时，$H(\boldsymbol{U}, C)$ 也小。为了消除类别数 C 对划分熵的影响，可对划分熵做归一化处理，即 $H(\boldsymbol{U}, C)/\ln C$，所得结果见表 9.4。此时，最小值对应的类别数是 $C=8$。

习题 9

9.1 计算模糊集合 $\tilde{A}$ 的海明模糊度、欧几里得模糊度和熵模糊度，其中

$$\tilde{A} = \frac{0.9}{a}+\frac{0.7}{b}+\frac{0.5}{c}+\frac{0.3}{d}+\frac{0.2}{e}$$

9.2 设 $\boldsymbol{U}=\{a, b, c, d, e, f\}$，$\tilde{A}=\frac{0.6}{a}+\frac{0.8}{b}+\frac{1}{c}+\frac{0.8}{d}+\frac{0.6}{e}+\frac{0.2}{f}$，$\tilde{B}=\frac{0.4}{a}+\frac{0.6}{b}+\frac{0.5}{c}+\frac{1}{d}+\frac{0.8}{e}+\frac{0.3}{f}$，试分别计算格贴近度 $N(\tilde{A}, \tilde{B})$、海明贴近度 $N_H(\tilde{A}, \tilde{B})$ 和欧几里得贴近度 $N_E(\tilde{A}, \tilde{B})$。

9.3　设论域 $\boldsymbol{U}=\{a, b, c, d, e, f\}$，其 $\boldsymbol{U}$ 上的模糊子集有 6 个：

$$\tilde{A}_1=(1, 0.8, 0.5, 0.4, 0, 0.1),$$

$$\tilde{A}_2=(0.5, 0.1, 0.8, 1, 0.6, 0),$$

$$\tilde{A}_3=(0, 1, 0.2, 0.7, 0.5, 0.8),$$

$$\tilde{A}_4=(0.4, 0, 1, 0.9, 0.6, 0.5),$$

$$\tilde{A}_5=(0.8, 0.2, 0, 0.5, 1, 0.7),$$

$$\tilde{A}_6=(0.5, 0.7, 0.8, 0, 0.5, 1),$$

且待识别的模糊子集是 $\tilde{A}_x=(0.7, 0.2, 0.1, 0.4, 1, 0.8)$。采用格贴近度确定待识别的模糊子集与 $\tilde{A}_1\sim\tilde{A}_6$ 中哪个最相近。

9.4　设论域 $\boldsymbol{X}=\{x_1, x_2, x_3, x_4, x_5\}$，给定 $\boldsymbol{X}$ 上一个模糊关系 $\tilde{R}$，其模糊矩阵为

$$\boldsymbol{R}=\begin{bmatrix} 1 & 0.8 & 0.8 & 0.2 & 0.8 \\ 0.8 & 1 & 0.85 & 0.2 & 0.85 \\ 0.8 & 0.85 & 1 & 0.2 & 0.9 \\ 0.2 & 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.85 & 0.9 & 0.2 & 1 \end{bmatrix}$$

判断 $\boldsymbol{R}$ 是模糊相似矩阵还是模糊等价矩阵；按不同的λ分类并给出分级聚类树。

9.5　已知 12 个二维样本 $\boldsymbol{X}=\{\boldsymbol{x}_i, i=1, 2, \cdots, 12\}$，其中 $\boldsymbol{x}_1=[0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2=[0, 1]^{\mathrm{T}}$, $\boldsymbol{x}_3=[1, 0]^{\mathrm{T}}$, $\boldsymbol{x}_4=[0.5, 4]^{\mathrm{T}}$, $\boldsymbol{x}_5=[1, 3]^{\mathrm{T}}$, $\boldsymbol{x}_6=[1, 5]^{\mathrm{T}}$, $\boldsymbol{x}_7=[1.5, 4.5]^{\mathrm{T}}$, $\boldsymbol{x}_8=[6, 4]^{\mathrm{T}}$, $\boldsymbol{x}_9=[6.5, 5]^{\mathrm{T}}$, $\boldsymbol{x}_{10}=[7, 4]^{\mathrm{T}}$, $\boldsymbol{x}_{11}=[7.5, 7]^{\mathrm{T}}$, $\boldsymbol{x}_{12}=[8, 7]^{\mathrm{T}}$。试用 FCM 算法进行分类，其中模糊性加权指数 $m=2$，用欧氏距离，类别分别为 2 和 4。

9.6　已知 29 个二维样本，具体是 $\boldsymbol{x}_1=[0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2=[0, 1]^{\mathrm{T}}$, $\boldsymbol{x}_3=[0, 2]^{\mathrm{T}}$, $\boldsymbol{x}_4=[0, 3]^{\mathrm{T}}$, $\boldsymbol{x}_5=[1, 0]^{\mathrm{T}}$, $\boldsymbol{x}_6=[1, 1]^{\mathrm{T}}$, $\boldsymbol{x}_7=[1, 2]^{\mathrm{T}}$, $\boldsymbol{x}_8=[1, 3]^{\mathrm{T}}$, $\boldsymbol{x}_9=[2, 0]^{\mathrm{T}}$, $\boldsymbol{x}_{10}=[2, 1]^{\mathrm{T}}$, $\boldsymbol{x}_{11}=[2, 2]^{\mathrm{T}}$, $\boldsymbol{x}_{12}=[2, 3]^{\mathrm{T}}$, $\boldsymbol{x}_{13}=[3, 0]^{\mathrm{T}}$, $\boldsymbol{x}_{14}=[3, 1]^{\mathrm{T}}$, $\boldsymbol{x}_{15}=[3, 2]^{\mathrm{T}}$, $\boldsymbol{x}_{16}=[3, 3]^{\mathrm{T}}$, $\boldsymbol{x}_{17}=[1, 6]^{\mathrm{T}}$, $\boldsymbol{x}_{18}=[1, 7]^{\mathrm{T}}$, $\boldsymbol{x}_{19}=[1, 8]^{\mathrm{T}}$, $\boldsymbol{x}_{20}=[2, 6]^{\mathrm{T}}$, $\boldsymbol{x}_{21}=[2, 7]^{\mathrm{T}}$, $\boldsymbol{x}_{22}=[2, 8]^{\mathrm{T}}$, $\boldsymbol{x}_{23}=[3, 6]^{\mathrm{T}}$, $\boldsymbol{x}_{24}=[3, 7]^{\mathrm{T}}$, $\boldsymbol{x}_{25}=[3, 8]^{\mathrm{T}}$, $\boldsymbol{x}_{26}=[7, 2]^{\mathrm{T}}$, $\boldsymbol{x}_{27}=[7, 3]^{\mathrm{T}}$, $\boldsymbol{x}_{28}=[8, 2]^{\mathrm{T}}$, $\boldsymbol{x}_{29}=[8, 3]^{\mathrm{T}}$。取模糊性加权指数 $m=2$、欧氏距离，试用 FCM 算法将数据样本集分为三类。

9.7　用 FCM 对鸢尾属植物(Iris)样本数据进行分类，其中模糊性加权指数 $m=2$，用欧氏距离，类别分别为 3。

参考文献

[1] 齐敏, 李大健, 郝重阳. 模式识别导论, 北京：清华大学出版社，2009.

[2] Bezdek J C. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.

[3] M.P.Windlham. *Cluster Validity for the Fuzzy C-means Clustering Algorithms*. IEEE Trans. on Pattern Analysis and Machine Intelligence. 11:357-363, 1982.

[4] 高新波. 模糊聚类分析及其应用. 西安：西安电子科技大学出版社，2004.

[5] Witola Pedrycz. 于福生译. 基于知识的聚类，北京：北京师范大学出版社，2008.

[6] 肖健华. 智能模式识别方法. 广州：华南理工大学出版社，2006.
[7] 张德丰. MATLAB 模糊系统设计, 北京：国防工业出版社，2009.
[8] 杨纶标, 高应仪. 模糊数学原理及应用. 广州：华南理工大学出版社，2005.
[9] 梁保松, 曹殿立. 模糊数学及其应用. 北京：科学出版社，2007.
[10] 吴士力. 通俗模糊数学与程序设计. 北京：中国水利水电出版社，2008.

第 10 章　模式识别应用

10.1　车牌识别

车牌识别(License Plate Recognition，LPR)可以应用于高速公路电子收费站、出入控制、公路流量监控、停车场车辆管理、公路稽查、监控违章车辆的电子警察等需要车牌认证的场合。

车牌识别过程如图 10.1 所示，其中各个步骤的主要功能如下。

(1)车牌预处理。将输入的车牌彩色图像进行灰度化处理，然后对图像进行灰度拉伸，以增强图像对比度，用中值滤波给图像去噪，最后检测图像边缘。

(2)车牌定位。在一幅含有复杂背景的车牌图像中找到车牌区域，车牌定位可分成两个步骤进行：车牌粗定位和精定位。

(3)字符分割。对得到的车牌图像进行字符的分割和归一化，将分割出的字符输入到后续的识别模块进行识别。

(4)字符识别。在对车辆牌照字符准确分割的基础上，对车牌上的汉字、英文字母、数字进行有效识别的过程，判断识别的结果是否正确。

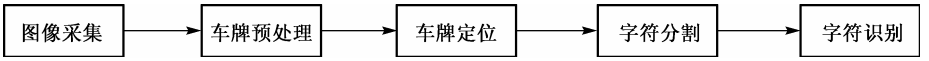


图 10.1　车牌识别系统原理流程图

10.1.1　车牌图像预处理

1．彩色图像的灰度化

数字图像可分为彩色图像和灰度图像。在 RGB 模型中，如果 $R = G = B$，则彩色变为灰度，其中 $R = G = B$ 的值称为灰度值，用 g 来表示。由彩色转换为灰度的过程称为灰度化处理。一般 CCD 摄像头得到的包含车辆的图像是 24 位真彩色图，需要转换成灰度图，便于后续的快速图像处理，另一方面也是对多种颜色车辆牌照进行统一。R, G, B 的取值范围是 0～255，所以灰度的级别是 256 级。灰度化的处理方法主要有如下三种[1]：

(a)最大值法：使 g 的值等于三值中的最大的一个，即

$$g = \max(R, G, B) \tag{10.1}$$

(b)平均值法：使 g 的值等于三值和的平均值，即

$$g = \frac{R + G + B}{3} \tag{10.2}$$

(c)加权平均值法：根据重要性或其他指标，给 R, G, B 赋予不同的权值，并使 g 等于它们的值的加权平均值，即

$$g=\frac{W_R R+W_G G+W_B B}{3} \tag{10.3}$$

式中 W_R，W_G，W_B 分别为 R, G, B 的权值。

通过对三种灰度化处理方法的试验验证，建议选取加权平均值法。由于人眼对绿色的敏感度最高，对红色的敏感度次之，对蓝色的敏感度最低，所以取 $W_R=0.9$，$W_G=1.77$，$W_B=0.33$，即

$$g=0.3R+0.59G+0.11B \tag{10.4}$$

彩色图像的灰度化处理如图 10.2 所示，用式(10.4)转换的灰度图能比较好地反映原图像的亮度信息。



(a) 车辆的彩色图像



(b) 灰度化图像

图 10.2 车辆图像的灰度化

2. 灰度拉伸

车牌图像定位的一个难点是，抓拍图像受环境因素影响较大，尤其是当外界光照条件过强或过弱时，容易使得整幅图像偏亮或偏暗，这种情况称为低对比度[2-3]。为了提高对比度，把感兴趣的灰度范围拉开，使得该范围内的像素，亮的越亮，暗的越暗，就要对图像进行灰度拉伸，使图像上的边缘更加凸显，这样牌照区域的笔画特征就会更加明显，更有益于下一步的处理[4]。

灰度拉伸变换原理图如图10.3所示，函数表达式为

$$f(x)=\begin{cases}\dfrac{d}{a}x, & x<a\\ \dfrac{d-c}{b-a}(x-a)+c, & a\leqslant x\leqslant b\\ \dfrac{255-d}{255-b}(x-b)+d, & x>b\end{cases} \tag{10.5}$$

式(10.5)中(a, c)和(b, d)是图 10.3 中的两个转折点的坐标。

设图像为 $m\times n$ 像素，其直方图为 $h(i)$，a 取满足 $\sum_{i=0}^{a}h(i)\geqslant\frac{mn}{10}$ 的最小整数，b 取满足 $\sum_{i=0}^{a}h(i)\leqslant\frac{9}{10}mn$ 的最大整数，c 和 d 分别可以在程序中动态设定，也可以根据经验自行设定。在图10.4中取 $c=\frac{a}{0.9}$，$d=\frac{3b}{2}$。

3. 滤波处理

滤波是对图像进行一系列区域处理，以达到去除干扰、突出图像特征信息的目的。滤波在频域范围内主要有低通滤波和高通滤波。

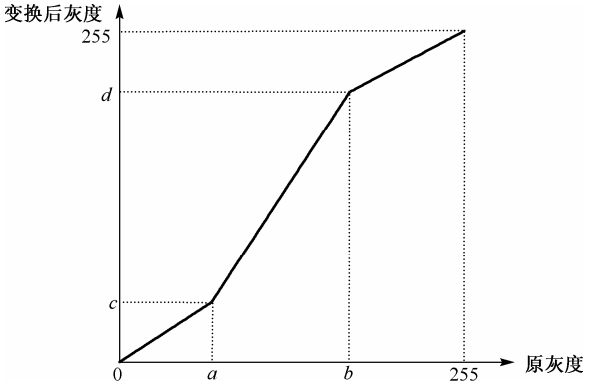


图 10.3　线性变换示意图



图 10.4　灰度拉伸后的车辆图像

低通滤波的基本思想是保留图像的低频成分，减少图像的高频成分，它可以降低图像中的视觉噪声，同时也削弱了图像的边缘信息[5]，使得图像中那些不明显的低频成分更容易识别。利用低通滤波来进行去噪，对由于拍摄环境、车牌污渍等形成的噪声进行去噪，提高系统的准确度。

高通滤波则增强图像的高频成分，减少低频成分，相对于高频成分，低频成分被削弱[6]。一般图像的边缘是图像的高频成分，所以高通滤波使图像锐化，在视觉上变得更清晰。

为了去除车辆图片背景噪声中的一些孤立噪声，一般采用中值滤波技术。中值滤波是一种非线性的信号处理方法，与其对应的中值滤波方法是一种非线性的图像平滑方法。中值滤波在一定的条件下可以克服线性滤波器如最小均方滤波、均值滤波等带来的图像细节模糊，而且对滤除脉冲干扰及图像扫描噪声最为有效。在实际运算过程中，由于不需要计算图像的统计特征，因此速度快、易实现；但对于一些细节点多，特别是点、线、分叉、尖顶较多的图像，则不宜采用中值滤波(如指纹图像宜采用 Gabor 滤波增强技术)。

中值滤波一般采用一个含有奇数点的滑动窗口，将窗口中灰度值的中值来替代指定点(一般是窗口的中心)的灰度值。对于奇元素，中值是指按大小排序后，中间的数值；对于偶数元素，指排序后中间两个元素灰度值的平均值。这种方法既能消除噪声，又能保持图像的细节[6]，其算法步骤如下：

(a)将模板在图中漫游，并将模板中心与图中某个像素位置重叠；

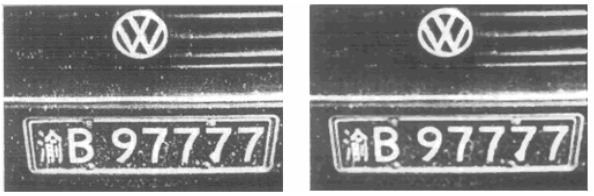
(b)读取模板下各对应像素的灰度值；

(c)将这些灰度值从小到大排成一列；

(d)找出排列在中间的 1 个；

(e)将这个中间值赋给对应模板中心位置的像素。

图10.5选用3×3的十字形窗口进行滤波，图像经过中值滤波后基本保留了图像的有用信息，去除了干扰。



(a) 原图　　(b) 中值滤波结果

图 10.5　中值滤波

4．边缘检测

两个具有不同灰度值的相邻区域之间总存在边缘，边缘就是灰度值不连续的结果，是图像分割、纹理特征提取和形状特征提取等图像分析的基础。这种灰度的不连续可以利用求导数的方法检测到：一阶导数可以用于检测图像中的一个点是否是边缘的点，二阶导数的符号可以用于判断一个边缘像素是在边缘亮的一边还是暗的一边。

为了对有意义的边缘点进行分类，与这个点相联系的灰度级必须比在这一点的背景上的变换更有效，可通过门限方法来决定一个值是否有效。所以，如果一个点的二维一阶导数比给定的门限大，就定义图像中的该点是一个边缘点。一组这样的依据事先定义好的、与连接准则相连的边缘点就定义为一条边缘。

边缘检测是检测图像局部显著变化的最基本运算[7]。在一维情况下，阶跃边缘同图像的一阶导数局部峰值有关。梯度是函数变化的一种度量，而一幅图像可以视为图像强度连续函数的取样点阵列。因此，图像灰度值的显著变化可用梯度的离散函数来检测。

二维函数 $f(x, y)$ 的梯度算子对应为一阶导数，定义为

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{10.6}$$

其幅度和方向角是

$$\nabla f = \mathrm{mag}(\nabla f) = \left[G_x^{\,2} + G_y^{\,2} \right]^{\frac{1}{2}} \tag{10.7}$$

$$\alpha(x, y) = \arctan\left(\frac{G_y}{G_x} \right) \tag{10.8}$$

在实际中，常用小区域模板卷积来近似计算上述公式，对 G_x 和 G_y 各用一个模板，所以需要 2 个模板组合起来构成一个梯度算子。常用的梯度算子有[8]：

(a) Roberts 算子。这是一种利用局部差分算子寻找边缘的算子，由下列公式给出：

$$G[i, j] = f[i, j] - f[i+1, j+1] + f[i+1, j] - f[i, j+1] \tag{10.9}$$

用卷积模板表示，上式变为

$$G[i, j] = G_x + G_y \tag{10.10}$$

其中 G_x 和 G_y 由表 10.1 所示的模板计算。

表 10.1　Roberts 算子

(a) X 方向算子

1	0
0	–1

(b) Y 方向算子

0	1
–1	0

图 10.6 是用 Roberts 算子对图 10.4 进行边缘检测的结果。Roberts 算子对具有陡峭的低噪声图像效果较好。

(b) Sobel 算子。采用3×3邻域可以避免在像素之间内插点计算梯度，是一种梯度幅值：

$$g=\left[G_x^{\ 2}+G_y^{\ 2}\right]^{\frac{1}{2}} \tag{10.11}$$

其中的偏导数用下式计算：

$$G_x=(a_2+ca_3+a_4)-(a_0+ca_7+a_6) \tag{10.12}$$

$$G_y=(a_0+ca_1+a_2)-(a_6+ca_5+a_4) \tag{10.13}$$

式中常数 $c=2$。

Sobel 算子的两个卷积计算核如表 10.2 所示，图像中的每个点都用这两个核做卷积，第一个核对通常的垂直边缘响应最大，第二个核对水平边缘响应最大。两个卷积的最大值作为该点的输出值。

表 10.2　Sobel 算子

1	2	1
0	0	0
–1	–2	–1

1	0	–1
2	0	–2
–1	0	–1

图10.7是用 Sobel 算子对图10.4进行边缘检测的结果。Sobel 算子对灰度渐变和噪声较多的图像处理得较好。

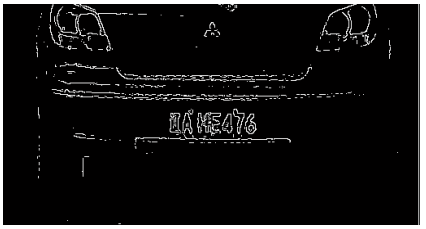


图 10.6　Roberts 算子边缘检测



图 10.7　Sobel 算子边缘检测

(c) Prewitt 算子。Prewitt 算子和 Sobel 算子的方程完全一样，只是常量 c 取 1，Prewitt 算子的两个卷积计算核如表10.3所示。

表 10.3　Prewitt 算子

1	2	1
0	0	0
–1	–2	–1

1	0	–1
2	0	–2
–1	0	–1

图10.8是用 Prewitt 算子对图10.4进行边缘检测的结果。

通过对几种边缘检测算法的大量实验验证，建议选取 Prewitt 算子进行车牌图像边缘提取。



图 10.8　Prewitt 算子边缘检测

10.1.2　车牌定位

1．车牌粗定位

为了在一幅含有复杂背景的图像中找到车牌区域，可采用窗口扫描、模板匹配的方法。一般的模板匹配是，拿已知的模板与原图像中同样大小的区域去对照[9]。首先，模板的左上角点

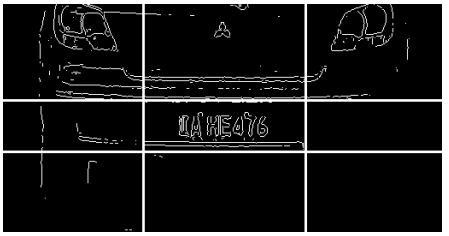
和图像的左上角点是重合的，拿模板和原图像中同样大小的一块区域去对比，然后平移到下一个像素，依次进行同样的操作。所有的位置都对比完后，差别最小的区域即为所求区域。

根据车牌区域白色点和跳变点均匀丰富的特性，用矩形匹配法来搜索车牌大致位置。在求得车牌大致位置的基础上，根据条件，搜索上下边缘；再根据长宽比信息，搜索车牌左右边缘，确定车牌准确位置。

用一个与估计的车牌大小相等的矩形，或者比估计的车牌稍大一些的矩形，遍历整个边缘二值图，落在该矩形内白色点最多的位置就是车牌区域的大致位置。

如果采集到的车辆图片大小为 768×512，可选用 160×60 像素的矩形模板，其搜索步长可定为 5。矩形模板的大小可以根据不同应用情况和场合进行调整。

在获得了车牌的大致范围［如图10.9(a)所示］并进行粗定位后，可以搜索车牌的上下边缘和左右边缘。



(a) 车牌图像粗定位



(b) 粗定位彩色图像



(c) 粗定位灰度图像

图 10.9 粗定位

搜索车牌上下边缘的算法描述是，在车牌大致位置的基础上，向上、向下按以下条件进行搜索：

(1) 搜索的范围不超过某一个阈值；

(2) 若某一行全部白色点的总数与车牌大致位置所在区域里平均单行全部白点总数相差不大，则认为该行是车牌所在行。

根据我国机动车牌号(GA36-92)标准，汽车车牌的标准尺寸为 440 mm×140 mm。根据这个特性，在确定了上下边缘之后，就可以按照长宽比为 44:14 来搜索车牌的左右边缘。

搜索左右边缘的算法描述是：以上下边缘差为高、以 44:14 为长宽比的矩形做模板，从左到右遍历由上下边缘所确定的候选区域，其车牌区域内的白色点应该是所有遍历情况中最多的。

在搜索定位到目标车牌的上下、左右边缘后，通常不能直接剪切；为了减少误切和信息丢失，按照经验，车牌边界值分别向左右上下扩展一定值，再进行图片剪切处理，以确保车牌有效区域的完整，如图10.9所示。因为后期还有针对车牌图像的切割最小范围的算法，这里多点冗余图像信息对最终字符识别的影响不大。

2．车牌图像的二值化

二值化又称为阈值化[10]，其目的就是要找出一个合适的阈值，将待研究的区域划分为前景和背景两部分。假设一幅灰度车牌图像大小为 $M\times N$，$f(x,y)$ 表示位于图像中第 x 行、第 y 列的像素的灰度值；其中，$0\leqslant x\leqslant M$，$0\leqslant y\leqslant N$，x 和 y 都是整数。二值化处理可以用下式表示：

$$f(x,y)=\begin{cases}255, & f(x,y)\geqslant T\\ 0, & f(x,y)<T\end{cases} \tag{10.14}$$

其中，T 为给定的阈值。

经过二值化处理后，字符的前景和背景就由黑白两种颜色分开，选择不同的阈值会得到不同的二值化效果。

目前，用于车牌的二值化算法有以下几种：

- 纹理分析法[11]。首先对图像进行纹理分析，确定车牌图像中字符像素点的灰度相对于牌照底色灰度的高或低，同时取得字符灰度与牌照底色的灰度的近似分布，最后采用模式识别中的最大最小准则获得灰度分割阈值。
- 灰度直方图波峰波谷法[12]。灰度直方图是灰度级的函数，描述的是图像中具有该灰度级的像素的个数，其横坐标是灰度级，纵坐标是该灰度出现的频率(像素的个数)。在车牌的灰度直方图中，找到位于两个波峰之间出现频率最小的灰度级，用这个灰度作为阈值进行二值处理。
- Ostu 方法[13]。其思想是，方差是灰度分布均匀性的一种度量，方差越大，说明构成图像的两部分的差别就越大，目标和背景的错分会导致这种差别变小。因此，使类间方差最大的分割意味着错分概率最小，找到这个使类间方差最大的灰度值，就是二值化的阈值。
- 彩色分析法[14]。该方法的输入图像是彩色的车牌，它把车牌的背景颜色分为黄、白、黑、天蓝、浅蓝、深蓝六类，然后分别统计这几种颜色的数目，确定具有最大数目的颜色值，这样也就找到了背景颜色，根据不同的背景颜色，选择相应的阈值，颜色越深阈值越大。该方法受光照的影响非常大，并且各种颜色的归类不是非常准确。
- 统计方法[15]。该方法应用的是图像的统计特征和先验信息。

(1)大律法

全局动态二值化是从整个灰度图像的像素分布出发，寻求一个最佳的阈值，其中经典算法是大律法(Ostu)算法。Ostu 算法是在判别最小二乘法的基础上推导出来的，其基本思想是：取一个阈值 k，将图像像素按灰度大小分为大于等于 k 和小于 k 两类，然后求出两类像素的平均值方差 σ_B^2 (类间方差)和两个类的各自的均方差 σ_A^2 (类内方差)，找出使两个方差比 σ_B^2/σ_A^2 最大的阈值 k，该阈值即为二值化图像的最佳阈值。这种方法不论图像的直方图有无明显的双峰，都能得到较为满意的效果，因此这种方法是阈值自动选取的较优方法。

设原始灰度图像灰度级为 L，灰度级为 i 的像素点数为 n_i，则图像的全部像素数为

$$N=n_1+n_2+\mathrm{L}\ +n_{L-1} \tag{10.15}$$

用阈值 t 把灰度级划分为两类：$C_0=(0,1,2,\text{L},t)$ 和 $C_1=(t+1,t+2,\text{L},L-1)$。因此，$C_0$ 和 C_1 类的出现概率及均值分别由下列各式给出：

$$w_0=P_r(C_0)=\sum_{i=0}^{t}p_i=w(t) \tag{10.16}$$

$$w_1=P_r(C_1)=\sum_{i=t+1}^{L-1}p_i=1-w(t) \tag{10.17}$$

$$u_0=\sum_{i=0}^{t}ip_i\Big/w_0=\frac{u(t)}{w(t)} \tag{10.18}$$

$$u_1=\sum_{i=t+1}^{L-1}ip_i\Big/w_1=\frac{u_r-u(t)}{1-w(t)} \tag{10.19}$$

其中 $u(t)=\sum_{i=0}^{t}ip_i$，$u_r=\sum_{i=0}^{L-1}ip_i$。可以得出，对任何 t 值，下式都能成立：

$$w_0+u_0+w_1+u_1=u_r,\qquad w_0+w_1=1 \tag{10.20}$$

C_0 和 C_1 类的方差可由下式求得：

$$\sigma_0{}^2=\sum_{i=0}^{t}\frac{(i-u_0)^2p_i}{w_0},\qquad \sigma_1{}^2=\sum_{i=t+1}^{L-1}\frac{(i-u_1)^2p_i}{w_1} \tag{10.21}$$

$$\sigma_A{}^2=w_0\sigma_0{}^2+w_1\sigma_1{}^2 \tag{10.22}$$

$$\sigma_B{}^2=w_0(u_0-u_r)^2+w_1(u_1-u_r)^2=w_0w_1(u_1-u_0)^2 \tag{10.23}$$

其中 $\sigma_A{}^2$ 是类内方差，$\sigma_B{}^2$ 是类间方差。

Ostu 算法描述如下：

(1) 求出图像中最大的灰度值 $G_{\max}$；

(2) 令 $k=1$；

(3) 求出大于 k 和小于 k 的这两类像素总数和像素的灰度平均值；

(4) 计算类间方差 $\sigma_B{}^2$ 和类内方差 $\sigma_A{}^2$；

(5) 令 $k=k+1$，循环 (3)~(5) 步，直到 $k=G_{\max}$ 时，循环结束；

(6) 找出最大的 $\sigma_B{}^2/\sigma_A{}^2$ 的值，其对应的值 k 即为所求阈值。

Ostu 算法基于图像像素的灰度值分类，按照使类间方差与类内方差比最大的原则获得阈值，使目标和背景之间方差最大，即找出使两个方差比 $\sigma_B{}^2/\sigma_A{}^2$ 最大的阈值 k。

用 Ostu 算法对图10.9(c)进行二值化的结果如图10.10所示。



图 10.10 Ostu 算法的二值化图像

(2) 迭代阈值法

迭代阈值法也称为基于灰度直方图的全局最佳平均阈值法，是一种基于逼近思想的方法[16]。首先将图像灰度取值范围的中值作为初始阈值(设共有 L 个灰度级)，然后按照式(10.24)迭代，直到 $T_{i+1}=T_i$ 或者相差很小时，结束迭代，并取此时的 T_i 分割阈值：

$$T_{i+1}=\frac{1}{2}\left\{\frac{\sum_{k=0}^{T_i}h_k k}{\sum_{k=0}^{T_i}h_k}+\frac{\sum_{k=T_i+1}^{L-1_i}h_k k}{\sum_{k=T_i+1}^{L-1_i}h_k}\right\} \tag{10.24}$$

其中 h_k 是灰度 k 出现的频数。

在图像目标区域和背景区域存在明显差异的情况下，这种算法效果十分理想，从路径规划的角度上是最优阈值，因为迭代阈值不需要知道先验知识。在计算中不必计算 h_k，直接通过如下步骤就可以求出迭代：

(a)求出图像的最大灰度值和最小灰度值，分别记为 $Z_{\max}$和$Z_{\min}$，计算初始阈值。

(b)根据阈值 T_i 将图像分割为前景(目标)和背景，分别求出两者的平均灰度值 Z_o+Z_b。

(c)求出新阈值(Z_o+Z_b)/2。

(d)若 $T_{i+1}=T_i$，则所得即为阈值，否则转(b)。

用迭代阈值法对图 10.9(c)进行二值化的结果如图 10.11 所示，迭代阈值法的二值化图像基本保留了车牌区域特征，而去掉了大量背景干扰，使后步的运算更为简捷。但是也可能会去掉车牌图像的一些细节，需要后期进行形态学处理。



图 10.11　迭代阈值法的二值化图像

3．车牌图像的倾斜校正

我国标准车牌为长方形，而在实际应用中，由于摄像机拍摄距离、角度或车牌本身挂歪等原因，经常遇到倾斜牌照。当牌照的倾斜角度大于 20°时，就会影响后续的牌照分割准确度和字符识别率，所以有必要考虑倾斜校正这个问题。目前的车牌倾斜校正方法主要有：

- 通过模板匹配寻找牌照区域的四个顶点，再通过双线性空间变换重建矩形车牌区域[17]。
- 通过求取车牌字符区域的局部极小和局部极大特征点，再进行投影确定车牌的倾斜角[18]。
- 通过求取车牌上各字符连通域的中心点，然后拟合为直线来确定车牌的倾斜角。
- Hough 变换法。通过 Hough 变换求取车牌的边框，确定车牌的倾斜角或者提取牌照边框的参数，求解牌照区域四个顶点的坐标，通过双线性空间变换对畸变图像进行校正。

目前，常用的方法是 Hough 变换。Hough 变换在待测直线具有小的扰动和断裂，甚至是虚线时，具有很强的检测能力[19]。具体实现时用到了 Hough 变换中一个比较简单的情形，任意的一个线性关系式 $y=xu+v$，它表达了 xy 平面上的点和 uv 平面上的直线的一种对应关系，对于 xy 平面上的任意一条直线 $y=ax+b$，取该直线上的任意两点坐标 (x_1,y_1) 和 (x_2,y_2)，这两点在 uv 平面中所对应直线的交点可以通过解如下方程组得到：

$$\begin{cases}y_1=ax_1+b\\ y_2=ax_2+b\end{cases} \tag{10.25}$$

由式(10.25)可以推出两平面上的交点为 $u=a,v=b$，它与 xy 平面上直线的选取无关。获取倾斜角度的步骤如下：

(1)在 xy 坐标系下，对粗定位后的车牌图像上的目标点扫描，求出它们在 uv 坐标系下的直线，并且把它们标记出来。

(2)在 uv 坐标系中，把这些直线经过的点都标记出来。

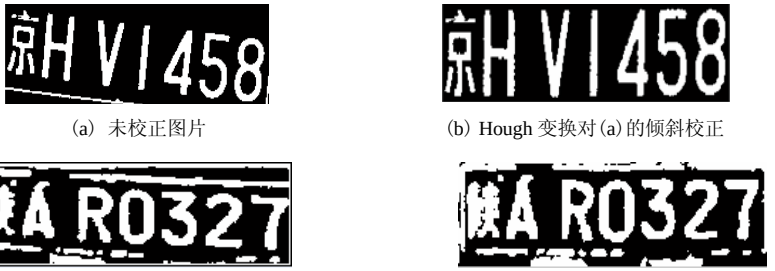
(3)在 uv 坐标系下，统计各个点被标记的次数，将被标记次数最多的点取出来。

(4)由 Hough 变换可知，在 uv 坐标系下的点 (a, b) 在 xy 坐标系下所对应的直线 $y = ax + b$ 的斜率为 α，那么这些直线的倾斜角即为车牌的倾斜角。

(5) 设倾斜角为 α，根据下式进行计算：

$$\begin{cases} x = (i - i_0)\cos\alpha - (j - j_0)\sin\alpha + i_0 \\ y = (i - i_0)\sin\alpha - (j - j_0)\cos\alpha + j_0 \end{cases} \tag{10.26}$$

式(10.26)实质是对原图像进行旋转操作，(x, y) 是原图像点的坐标。实验结果如图10.12所示。



(a) 未校正图片　(b) Hough 变换对(a)的倾斜校正

(c) 未校正图片　(d) Hough 变换对(c)的倾斜校正

图 10.12　车牌校正

4．精定位

经过粗定位、二值化处理后的车牌图像，其车牌区域具有以下三个基本特征：

- 在一个不大的区域内密集包含有多个字符；
- 车牌字符与车牌底色形成强烈对比；
- 车牌区域大小相对固定，即车牌区域长度和宽度成固定比例。

因此，车牌区域所在行相邻像素之间从 0 到 1 和从 1 到 0 的变化会很频繁，变化总数远远大于其他区域。

精确定位车牌的上下边界。由上至下统计每行相邻像素之间的灰度变化次数，当某行的变化次数首次大于某一临界值时，则假定该行为待搜索车牌的最高行；然后继续向下搜索，当某行的变化次数首次小于某一临界值时，则假定该行为待搜索车牌的最底行。具体步骤如下：

(1)逐行对车牌图像进行扫描，统计每行中像素的灰度值从 0 到 1 和从 1 到 0 变化的次数 n，黑白跳变小于某阈值的行即被视为背景；

(2)若某行连续白线长度大于某阈值，则该白线被认为是背景；

(3)若单行白色点总数大于某阈值，则该行被认为是背景；

(4)做完以上处理后，在车牌高度的上面 1/2 处向上搜索第一条全为黑的线，则认为该黑线以上为背景；在下面 1/2 处向下搜索第一条全为黑的线，则认为该黑线以下为背景；如图10.13所示。



图 10.13　经过第(1)~(4)步处理后得到的图像

(5)对步骤(4)的“原始图像”进行开运算得到图像背景，用原始图像与背景图像做减法运算，生成图10.14所示图像；



图 10.14　经过第(5)步处理后得到的图像

(6)由于图10.14中还有一些很小的面积未被滤除，使用面积滤除法进行滤波处理。所谓面积滤除法，就是找出图像中的所有连通域，并计算每一个连通域的面积，把面积小于一定阈值的连通域清除掉(阈值设定为 50)；面积滤除法的滤波结果如图10.15所示。



图 10.15　经过第(6)步处理后得到的图像

(7)沿着车牌图像的两边竖直方向扫描可去除左右边框干扰，即 $f'(i)-f'(i-1)<S'$，即为边缘部分，赋值为 0，其中 $f'(i)$ 表示第 i 列黑色像素点的总和；

(8)定义一个 $2\times M$ 矩阵 Judge1，其中 M 为图像行数，统计每列中像素的灰度值从 0 到 1 和从 1 到 0 变化的次数 n，第一行用来记录 n 的值，第二行用于对 n 进行校验，置初值为 0；

(9)如果 n 大于阈值(阈值取 8)，则将矩阵 Judge1 的第二行对应列中该位置的元素置 1，然后继续向下行扫描，重复以上判断；

(10)当全部扫描完成后，矩阵 Judge1 就记录下了每行的 0、1 变化次数，然后对 Judge1 的第二行从前向后进行观察，如果发现连续存在 10 个“1”，就可以认为最前面的“1”所在的行就是车牌的上边界；同理，对 Judge1 从后往前进行观察，可以确定车牌的下边界。

(11)定义一个 $2\times N$ 矩阵 Judge2，其中 N 为图像列数，对每列进行扫描，统计每列中像素的灰度值从 0 到 1 和从 1 到 0 变化的次数 n，第一列用来记录 n 的值，第二列用于对 n 进行校验，置初值为 0；

(12)如果 n 大于阈值(阈值取 6)，则将矩阵 Judge2 的第二列对应列中该位置的元素置 1，然后继续向下列扫描，重复以上判断；

(13)当全部扫描完成后，矩阵 Judge2 就记录下了每列的 0、1 变化次数，然后对 Judge2 的第二列从左向右进行观察，如果发现连续存在 10 个“1”，就可以认为最前面的“1”所在的列就是车牌的左边界；同理，对 Judge2 从右往左进行观察，可以确定车牌的右边界。

经过第(7)~(13)步处理后的结果如图10.16所示。



图 10.16 精定位图像

10.1.3 字符分割

在经过 10.1.2 节的车牌定位处理之后，得到的牌照图像已经基本上可以满足字符分割的需要。由于牌照有其自身的规格要求，字符之间有一定的间隔，而这个间隔是可以想到的分割依据。

根据车牌区域的特征，由于左边第一个字符为汉字字符，汉字字符的结构较复杂，一个汉字字符的中间可能有间隙，这给分割带来一定的干扰。因此，采用基于车牌区域先验知识和投影方法，从右向左对车牌进行扫描。

(1) 定义变量 p 代表车牌中的七个字符，$p = 0, 1, \cdots, 6$，$L(p)$ 和 $R(p)$ 分别为第 p 个字符的左右边界，$h(i)$ 表示第 i 列垂直投影后像素值为 1 的点的个数。

(2) 设定阈值 T，如果 $h(i)$ 大于 T，那么认为第 i 列为可能的字符区域或者比较大的干扰；如果 $h(i)$ 介于 0 和 T 之间，那么认为第 i 列为一些小的干扰噪声；如果 $h(i)$ 等于 0，则认为是字符之间的间隔区域。事实上，由于不可能完全消除噪声，间隔区域也存在一部分干扰噪声，使得 $h(i)$ 不可能等于 0，因此只要非常接近 0，就认为是字符之间的间隔区域。

(3) 设定字符投影宽度的阈值为 m，右边框的位置阈值为 n，i 初始值为图像宽度减 1。

(4) 对图像的 $h(i)$ 从右向左进行检测，直到遇到 $h(i)$ 大于 T 为止，令 $R = i$。

(5) 然后继续向左检测，直至 $h(i)$ 小于等于 T；同时统计扫描过的列数，记为 lie，用 L 记下最后一个 $h(i)$ 大于 T 的位置。

(6) 如果 lie 小于 m，同时 L 大于 n，那么认为是干扰信息，转到下一步继续执行；否则认为是字符。L 和 R 分别为当前字符的左右边界，令 $p = p - 1$，继续执行下一步，以确定下一个字符的位置。

(7) 返回至步骤(5)继续执行，直至 i 等于 0 为止，即找到所有的字符。

车牌的垂直投影如图10.17所示，图10.16的字符分割结果如图10.18所示。

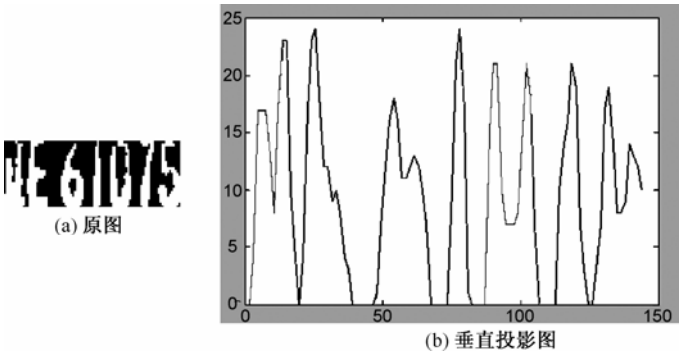


图 10.17 车牌的垂直投影图

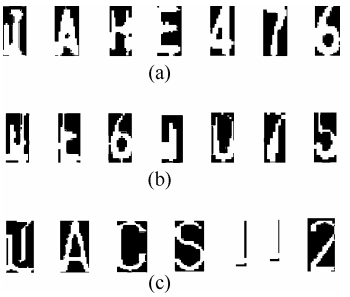


图 10.18 图 10.16 的字符分割结果

10.1.4　字符识别

字符识别是在车牌字符准确分割的基础上，对车牌上的汉字、英文字母、数字进行识别的过程。车牌上的字符属于印刷体，但由于摄像机的性能、车牌的整洁度、光照条件、拍摄时的倾斜角度及车辆运动等因素的影响，使牌照中的字符可能出现比较严重的模糊、歪斜、缺损或污迹等干扰，给字符识别带来困难。

目前用于车牌字符识别中的算法主要有基于模板匹配的方法、基于特征匹配的方法及基于人工神经网络的方法[21]。

(1) 基于模板匹配的方法。模板匹配方法是一种经典的模式识别方法，首先对待识别字符进行二值化，并将其大小归一化为字符数据库中模板的大小，然后与所有的模板进行匹配，最后选最佳匹配作为结果。但是字符大小、方向、字体的变化及噪声都将严重地影响模板匹配的正确率。在实际应用中，为提高正确率，往往必须使用多个模板进行匹配，而处理时间则随着模板个数的增加而增加。基于关键点的模板匹配算法对传统的模板匹配算法做出改进，此算法先对待识别字符进行关键点提取，即对字符进行拓扑分析以得到边符边缘的关键点，然后对关键点去噪，再确定字符的分类。使用关键点进行模板匹配有效地减少了模板中像素点的个数，只利用字符的关键点进行模板匹配，既提高了识别速度，又具有较高的识别率。

(2) 基于特征匹配的的方法。这种字符识别方法是提取字符的相关特征，然后利用这些特征来进行字符匹配，选择最接近的匹配结果。基于特征匹配的算法效率比模板匹配算法好，但是特征的正确提取比较困难。

(3) 基于神经网络的字符识别方法。车牌上的字符从车牌图像上分割下来以后，或多或少地与原来的字符存在着一定差异，由于神经网络的容错性较强，所以可以使用神经网络进行字符识别。目前，常用的神经网络主要有 Ko-honen、Hopfield 网络和 BP 网络等；由于神经网络自身的复杂性，选择哪一种类型的网络最好并没有一定的答案，主要是根据网络的分类样本类型和数量来选。这里选择了广泛使用的 BP 网络。

1．字符归一化

考虑到算法复杂性和实时性的要求，选用邻近插值法进行字符归一化，将字符归一化为 40×20 点阵。对图10.18(a)进行位置归一化和大小归一化后，得到的车牌字符如图10.19 所示。



图 10.19　对图 10.18(a)归一化后的字符

2．字符特征提取

特征提取能影响到整个系统识别的成功与否。由于粗网格特征能反映字符的整体形状分布，针对车牌字符的特点，选取字符的粗网格特征作为识别字符的特征。

粗网格特征属于统计特征中的局部特征，又称局部灰度特征。它通过把字符分成 $M\times N$ 个网格，统计每个网格中目标像素(白色像素)的数量，而每个网格各自反映字符的某一部分特征，在识别阶段，将所有网格特征值组合在一起形成一个 $M\times N$ 的粗网格特征向量。这里

将归一化后字符点阵的每个像素点作为一个网格，直接输入到神经网络分类器中。图 10.20 是数字 4 的粗网格特征。

3．BP神经网络参数的确定

根据老式牌照字符的特点，第 1 位字符是汉字，第 2 位字符是大写英文字母，第 3 位字符是大写英文字母或数字，第 4～7 位字符是阿拉伯数字。在识别过程中，由于 0 与 D、Q，8 与 B，4 与 A 不容易正确识别，为了提高系统识别率，根据字符所处位置的不同构造 4 个神经网络，即汉字网络、字母网络、字母数字网络、数字网络，如图10.21所示。

```
000000000001111111000
000000000011111111000
000000000011111111000
000000000011111111000
000000000011111111000
000000000011111111000
000000000011111111000
000000011111111111000
000000011111111111000
000000011111111111000
000000111111111111000
000000111111111111000
000000111111111111000
000001111111111111000
000001111111111111000
000001111111111111000
000011111111111111000
000011111111111111000
000011111101111111000
000011111101111111000
001111111101111111000
001111111101111111000
001111111000111111000
011111111000111111000
111111111111111111110
111111111111111111111
111111111111111111111
111111111111111111111
111111111111111111110
111111111111111111110
000111111111111111110
000000000001111111000
000000000001111111000
000000000001111111000
000000000001111111000
000000000001111111000
000000000000001111000
```

图 10.20　数字 4 的粗网格特征

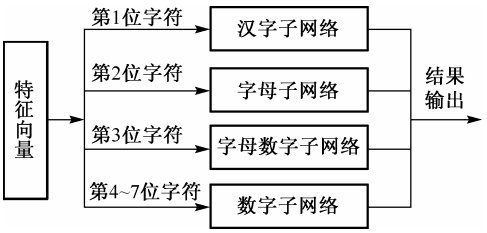


图 10.21 字符识别过程框图

BP 神经网络采用误差反向传播算法对网络权值进行训练的多层前向网络[22]。

(1) 输入层神经元个数。输入层神经元个数根据待识别字符所提取的粗网格特征的维数大小确定。由于归一化为 40×20 点阵，以每个像素点为一个网格，输入层神经元个数取 800。

(2)输出层神经元个数。各个网络输出层神经元个数分别为：汉字网络是 51；字母网络是 25；字母数字网络是 34；数字网络是 10。以最简单的数字网络为例，其输出如表 10.4 所示。

表 10.4　数字网络的期望输出

训练字符 / 输出神经元	0	1	2	3	4	5	6	7	8	9
输出神经元 1	1	0	0	0	0	0	0	0	0	0
输出神经元 2	0	1	0	0	0	0	0	0	0	0
输出神经元 3	0	0	1	0	0	0	0	0	0	0
输出神经元 4	0	0	0	1	0	0	0	0	0	0
输出神经元 5	0	0	0	0	1	0	0	0	0	0
输出神经原 6	0	0	0	0	0	1	0	0	0	0
输出神经元 7	0	0	0	0	0	0	1	0	0	0
输出神经元 8	0	0	0	0	0	0	0	1	0	0
输出神经元 9	0	0	0	0	0	0	0	0	1	0
输出神经元 10	0	0	0	0	0	0	0	0	0	1

(3)隐含层层数的选择。可采用下式确定各个网络隐层神经元的数目，具体如表 10.5 所示：

$$n_1 = \sqrt{m \times (n+1) + 1} \tag{10.27}$$

式中，n_1 表示隐含层神经元个数，n 为输出层的单元数，m 为输入层的单元数。

表 10.5　隐含层神经元个数

	输入层神经元个数	输出层神经元个数	隐含层神经元个数
数字网络	800	10	94
字母网络	800	24	142
字母数字网络	800	34	168
汉字网络	800	51	204

(5)激活函数的选择。神经元的激励函数是 S 型函数，即 $f(x) = \dfrac{1}{1+\mathrm{e}^{-x}}$。

(6)初始权值的选取。为了保证每个神经元的权重都能够在 S 型函数变化最大的地方进行调节，一般初始权值取–1 和 l 之间的随机数。

(7)学习速率和动量因子的选择。在实际应用中，结合车牌字符识别系统的小类别分类特点，通常根据对误差准则函数 E_{av} 和各类别网络实际输出值的观察，来调整学习率 η 和动量因子 α。在开始训练时，先设置学习率参数 $\eta = 0.1$，动量因子 $\alpha = 0.9$；经过数次迭代后，如果观察到 E_{av} 忽大忽小且网络实际输出值也呈现与之相同的趋势，说明 η 取得过大，则应该减小学习率参数。如果训练过程中，观察到几次迭代调整后，E_{av} 和网络实际输出值的变化不大，则可能 η 过小或网络已趋于收敛，即训练已处于误差曲面的平坦区，此时可适当增大 η 的取值。当网络的训练确实已处于误差曲面的平坦区，而学习速度又太小时，可适当调整动量因子 α 的值。

本实验系统中，各个神经网络的参数配置如表10.6所示。

4．BP网络的训练

根据不同的训练样本集，确定初始化网络参数：最大迭代次数、学习率、动量因子、激

活函数等。选择训练样本，输入训练样本特征。确定网络结构，输入各层的网络节点。调用 BP 算法训练网络。保存训练成功的网络权值。

表 10.6 神经网络的网络结构和参数配置

参数配置 / 网络结构名称	输入层节点数	隐含层节点数	输出层节点数	学习率参数	动量因子	误差目标值	迭代次数	待识别字符
数字神经网络	800	94	10	0.01	0.05	0.0001	852	0-9
字母神经网络	800	142	24	0.01	0.05	0.0001	1332	除 I 以外的 25 个大写字母
字母数字神经网络	800	168	34	0.01	0.05	0.0001	537	0-9，A-Z，除 I 和 O 之外的大写字母
汉字网络	800	204	51	0.01	0.05	0.0001	134	省份等 51 个汉字

以数字网络为例，训练样本的实际输出如表10.7所示。

表 10.7 数字网络训练样本实际输出值

训练数字 / 输出神经元	0	1	2	3	4	5	6	7	8	9
神经元 1	0.9984	0.0002	0.0002	0.0002	0.0000	0.0010	0.0004	0.0001	0.0000	0.0001
神经元 2	0.0005	0.9989	0.0007	0.0007	0.0009	0.0005	0.0005	0.0012	0.0000	0.0001
神经元 3	0.0001	0.0000	0.9989	0.0010	0.0003	0.0000	0.0000	0.0008	0.0003	0.0009
神经元 4	0.0008	0.0001	0.0007	0.9985	0.0010	0.0000	0.0003	0.0007	0.0005	0.0001
神经元 5	0.0000	0.0005	0.0007	0.0006	0.9979	0.0008	0.0001	0.0004	0.0000	0.0001
神经元 6	0.0005	0.0005	0.0000	0.0000	0.0008	0.9987	0.0004	0.0000	0.0007	0.0009
神经元 7	0.0003	0.0003	0.0006	0.0003	0.0012	0.0000	0.9992	0.0004	0.0009	0.0005
神经元 8	0.0004	0.0005	0.0004	0.0009	0.0009	0.0003	0.0000	0.9985	0.0002	0.0005
神经元 9	0.0003	0.0000	0.0009	0.0013	0.0000	0.0010	0.0007	0.0013	0.9974	0.0006
神经元 10	0.0008	0.0001	0.0003	0.0000	0.0000	0.0007	0.0002	0.0003	0.0004	0.9986

对比表10.4和表10.7，可以看出神经网络的训练样本实际输出值和期望输出值基本趋于一致。

5．识别结果

用 MATLAB 对字符进行识别。在实验中，测试数据选取实际摄取的 200 幅图片，各类字符共有 1400 个。对每类数据随机抽取 50%作为训练样本，剩余的样本作为测试数据。

车牌图像经过区域定位裁剪、二值化、倾斜校正、去除边框、字符分割及归一化处理后，得到字符的 40×20 像素点阵，作为神经网络输入。由于实际条件所限，汉字仅采集到 7 类样本，分别为京、辽、鲁、冀、浙、湘、皖；英文字母集中缺少 5 类样本，分别为 I、R、V、W、X；数字样本齐全。

以数字神经网络为例，此时结构参数中的误差目标值为 0.0001，学习率参数为 0.05，则训练完成所需的迭代次数为 593 次，训练好的神经网络就可以使用了。对于训练样本识别率能够达到 100%，对于非训练样本和识别样本的识别率也非常高。数字网络误差性能曲线如图10.22所示。

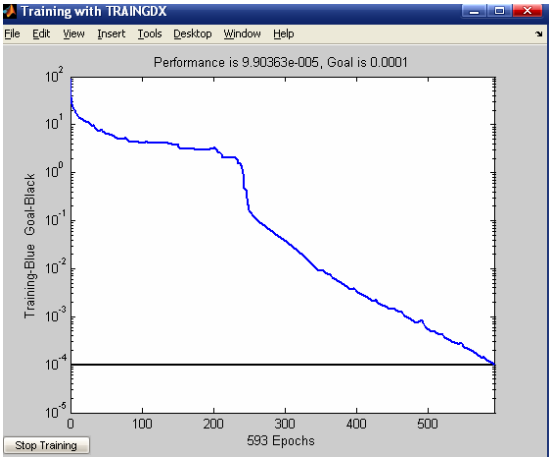


图 10.22　数字网络误差性能曲线

用所有训练样本和识别样本作为测试集时，在数字网络中错误的样本主要集中在“1”上；字母神经网络中错误的样本主要集中在相似字母如“O”和“D”上；字母数字神经网络中错误的样本主要是由相似字符“B”和“8”导致的。要想减少错误识别率，应对字母数字进行细分类。

部分车牌识别样本如图10.23所示，综合识别率如表10.8所示。

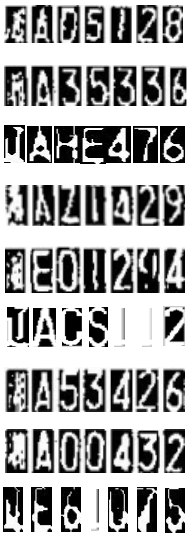


图 10.23　部分车牌样本识别结果

表 10.8　综合识别率

网 络 类 型	样　本　数	误　识　数	识　别　率
数字网络	500	35	93.0%
字母网络	400	40	90.0%
字母数字网络	600	34	91.0%
汉字网络	100	15	85.0%

10.2 语音识别

语音识别(Speech Recognition)是机器通过识别和理解过程，将人类的语音信号转变为相应的文本或命令的技术。语音识别技术是一门多学科交叉技术，涉及语音学、发声和听觉机理、心理学、数字信号处理、信息论和概率论、模式识别、人工智能及计算机科学等。语音识别应用范围相当广泛，可应用于各行各业，如语音拨号、翻译系统、智能玩具、智能家居、汽车导航、工业控制、信息查询、军事监听等。

语音识别有以下几种分类方法[23]。

(1)按发音方式分类

- 孤立词识别(Isolated Word Recognition)。用户在对系统说话时，相邻的词汇之间的发音要有明显停顿，在这种发音方式下，词汇之间语音信号的声学特征基本不受下文的影响，词汇在语音信号中的起始点和结束点的检测比较容易，系统实现难度较低。
- 连接词识别(Connected Word Recognition)。对中小规模词汇表中的若干词条，以慢速连读的方式连续说出，一般指 0~9 十个数字连接而成的多位数字识别，并包含少量的操作指令等。
- 连续语音识别(Continuous Speech Recognition)。说话人以日常自然的方式讲述并进行识别。

(2)按词汇量大小分类

- 小词汇量(词汇量小于 100)。
- 中等词汇量(词汇量在 100 到 1000 之间)。
- 大词汇量(词汇量大于 1000)。
- 无限词汇量(全音节识别)。

一般情况下，需要识别的词汇量越多，词汇之间就越容易混淆，系统实现就越困难，系统的识别率也会降低。

(3)按照说话人的限定范围分类

- 特定人语音识别(Speaker Dependent)。特定人语音识别是指识别系统只针对特征的某个用户进行识别工作的方式。
- 非特定人语音识别(Speaker Independent)。非特定人语音识别是指识别系统可以针对任何人工作。非特定人语音识别需要针对不同人建立模型，实现起来难度较大，但是通用性好，应用更广。

10.2.1 语音识别研究的发展与现状

语音识别研究从 20 世纪 50 年代开始到现在，已经历半个多世纪的蓬勃发展，在这期间获得了巨大的进展。

20 世纪 50 年代，研究人员大都致力于探索声学-语音学的基本概念。世界上最早的语音识别系统 Audrey[24]是在 1952 年由美国贝尔实验室开发的，该系统实现了孤立人英语数字语

音识别系统，方法主要是度量每个数字的元音音段的共振峰。1959 年，英国伦敦大学学院的研究人员尝试用谱分析和模板匹配方法构建了一种音素识别器，用以识别 4 个元音和 9 个辅音，第一次使用了统计语法的概念[25]。

20 世纪 60 年代以后，各种语音识别的研究才开始展开。RCA 实验室的研究成果解决了语音在时间标尺上的不均匀问题。在能够可靠检测出语音事件的始末点的基础上，发展了一套时间归整的基本方法，显著降低了识别匹配评分的变化程度[26]。1968 年，前苏联的研究人员 Vintsyuk 首次将动态规划算法（Dynamic Programming，DP）[27]应用于语音分析，使用 DP 来对齐两个不同长度的语音音段。卡耐基·梅隆大学（Carnegie Mellon University，CMU）的研究人员 Reddy 用动态跟踪音素的方法进行了连续语音识别[28]的开创性工作。

20 世纪 70 年代，语音识别技术开始快速发展。日本学者 Itakura 提出了基于 DP 技术的动态时间规整算法（Dynamic Time Warping, DTW）[29]，该算法是将时间规整和距离测度计算结合起来的一种非线性规整技术。DTW 搭配基于线性预测编码（Linear Prediction Coding, LPC）[30]的谱系数提取，使得孤立词识别的效率大大提高，线性预测技术在语音识别领域从此得到广泛应用。Helms 首次将向量量化（Vector Quantization, VQ）[31]用于说话人识别，该方法把每个人的训练数据通过标准的聚类过程生成码本，识别时将测试输入向量按此码本进行编码，以量化产生的失真度作为输出结果的判决条件。VQ 方法不需要对时间进行对齐，简化了系统复杂度；该方法识别精度高，且判断速度快。在此期间比较有代表性的语音识别系统有：CMU 的 Hearsay-II（提出了使用并行异步过程来模拟语音识别系统中不同的知识源）、IBM 的大词汇量自动语音听写系统和贝尔实验室用于通信的与说话人无关的语音识别系统。

20 世纪 80 年代，语音识别技术的研究进一步深入，连接词和大词汇量连续语音识别成为研究热点，语音识别开始由简单的基于模板的方法向统计建模框架转变。贝尔实验室的 Rabiner 等科学家把原本晦涩的隐式马尔可夫模型（Hidden Markov Model, HMM）[32]理论工程化，从而使更多的研究者了解和使用该模型。HMM 成为大词汇量连续语音识别系统的基础。人工神经网络（Artificial Neural Network, ANN）[33]在 20 世纪 50 年代被提出，其具有非线性性、自适应性和鲁棒性等特性。ANN 在 20 世纪 80 年代被引入语音识别领域，其独特的优点及其较强的分类能力和输入-输出映射能力使其在语音识别研究中备受关注。在此期间，比较具有代表性的语音识别系统是 CMU 于 1988 开发的 SPHINX 系统，该系统使用了隐式马尔可夫模型 HMM 和向量量化技术 VQ。SPHINX 系统是世界上第一个非特定人大词汇量连续语音识别系统，其能够识别包含 997 个词汇的 4200 个连续语句，在语言复杂度为 60 且环境匹配时，识别率可以达到 94.7%，经过多次改进，其识别率可达到 95.8%。

20 世纪 90 年代，随着信号特征的提取和优化技术、声学模型的细化、自然语言理解领域中语音模型的建立和解码搜索算法技术的不断成熟，出现了比较成功的大词汇量连续语音识别系统。在此期间，系统的鲁棒性越来越被人们所重视，各种试图提高语音识别系统在测试不匹配条件下（其中导致不匹配的原因包括背景噪声、不同人说话风格、麦克风、传输信道和房间回响等）性能的技术被提出，如最大似然线性回归（Maximum Likelihood Linear Regression, MLLR）准则、最大后验（Maximum A-Posteriori, MAP）准则[34]、结构化 MAP[35]、模型分解[36]、并行模型合并（Parallel Model Composition, PMC）[37]等。在此期间，比较有代表性的语音识别系统有：IBM 对 ViaVoice 系列、Microsoft 的 Whisper 系统、CMU 的 SPHINX-II 系统等。

IBM 的 ViaVoice 语音识别系统带有一个 32 000 词的基本词汇表，可扩展到 65 000 个词，

还包括办公常用词条，具有“纠错机制”，其平均识别率可以达到 95%。ViaVoice 为用户提供了非常个性化的设计，根据每个人的不同发音习惯做了相应的设定。运行在 Windows 下的 ViaVoice 支持 Microsoft Office 2003，可以为不同要求的用户提供了精确的语音识别技术。ViaVoice 使用便利，特别适合于起草文稿、撰写文章、准备教案。

Nuance（原名 Scansoft）是著名的语音和图像解决方案提供商。在语音技术市场，有超过 80%的语音识别采用的是 Nuance 的识别引擎技术，其名下有超过 1000 个专利技术，公司研发的语音产品可以支持超过 50 种语言，在全球拥有超过 20 亿用户。该公司提供人性化、高效率的电话口语或语言辨识功能。其英文语音产品 Dragon Naturally Speaking 在法律和医院临床记录领域占据很大市场。Dragon Naturally Speaking 的速度为手动输入字符速度的 3 倍，而且准确率高达 99%。

我国的语音识别研究工作最早开始于中国科学院声学所。20 世纪 50 年代后期，中科院声学所用频谱分析的方法研究了汉语 10 个元音的语音识别，到 20 世纪 70 年代后期，构建了基于模板匹配的孤立词语音识别系统。在 20 世纪 80 年代后期，主持研究了“八五”期间中科院人机语音对话研究项目。在此期间，国内大专院校和研究所相继开始了语音识别研究。中科院声学所和自动化所、北京大学、清华大学等研究机构在中国的语音识别研究的方向和内容等方面起了积极的催化和引导作用。比较具有代表性的系统有：1991 年 12 月四川大学计算机中心在微机上实现了一个主题受限的特点人连续英语-汉语语音翻译演示系统；1992 年，由清华大学电子工程系和中国电子器件公司合作研制成功的 THED-919 特定人语音识别与理解实时系统；2002 年，中科院自动化所及其所属模式科技（Pattek）公司推出面向不同计算平台和应用的“天语”中文语音系列产品——PattekASR，结束了中文语音识别产品自 1998 年以来一直由国外公司垄断的历史。

科大讯飞作为中国最大的智能语音技术提供商，在智能语音技术领域有着长期的研究积累，并在中文语音合成、语音识别、口语评测等多项技术上拥有国际领先的成果。其产品 InterReco 是高效、稳定、易用的电话语音识别系统，提供最好的中文语音识别效果，支持识别国际标准协议。

模识科技公司依托中国科学院自动化所语音识别技术领域 20 年的技术积累，在多语言、大词汇量、非特定人、连续语音识别等核心技术方向代表了世界先进水平。Pattek ASR3.0/SK 是专门面向电信语音交互服务应用设计开发的语音识别系统，由于使用了全新的识别引擎，全面提升了使用中的稳定交互性能，通过简单集成，可以让使用者通过电话方便地使用语音与系统交互，搜索信息和获取服务。

现阶段语音识别的主要研究方向是如何提高语音识别系统的鲁棒性和如何建立新模型来提高语音识别的性能。

10.2.2 语音识别方法简介

比较成功的语音识别方法有模板匹配法、随机模型法和人工神经网络法等，下面将分别介绍这几种语音识别方法。

1．模板匹配法

模板匹配法（Template matching method）一般用于特定人、小词汇量或者语音识别系统。

在训练阶段，用户将词汇表中的词每个读一遍，将其特征向量作为模板存入模板库。在识别阶段，将输入语音的特征向量序列依次与模板库中的每个模板进行相似度比较，并将相似度最高者作为识别结果输出。因为语音信号输出有较大的随机性，不同人说同一句话中的同一音具有不同的时间长度，即使是同一人在不同时刻说同一句话中的同一音，也不可能具有完全相同的时间长度，因此在语音识别时需要时间伸缩处理。Helms 首次将向量量化(Vector Quantization, VQ)用于说话人识别，该方法把每个人的训练数据通过标准的聚类过程生成码本，识别时将测试输入向量按此码本进行编码，以量化产生的失真度作为输出结果的判决条件。VQ 方法不需要对时间进行对齐，简化了系统复杂度；该方法识别精度高，且判断速度快。日本学者板仓(Itakura)将动态规划算法(DP)的概念应用于解决孤立词识别时说话速度不均匀的问题，提出了著名的动态时间规整法(DTW)。DTW 是一个典型的最优化问题，它使用满足一定条件的时间规整函数将测试语音模板的时间轴非线性地映射到参考模型的时间轴上。当词汇量较小且各个词不容易混淆时，DTW 方法取得了很大的成功。

2．随机模型法

随机模型法(Stochastic model method)用于非特定人、大词汇量、连续语音识别系统。随机模型法的突出代表是基于概率运算的隐式马尔可夫模型(HMM)方法，该方法最初由 Baum 提出，Rabiner 等人对其在语音识别领域的应用进行了广泛深入的研究。HMM 的出现使得自然语音识别系统取得了实质性的突破。

HMM 对语音信号的时间序列建立统计模型，将其视为数学上的双重随机过程：一个是使用具有有限状态数的马尔可夫链来模拟语音信号统计特性变化的隐含随机过程，另一个是与马尔可夫链每个状态相关的观察序列的随机过程。前者通过后者表现，但前者的具体参数是不可见的。人说话的过程就是一个双重随机过程，HMM 合理地模拟了人的发音过程，表现了语音信号的整体非平稳性和局部平稳性，是一种比较合理的语音模型。

HMM 在训练阶段对每个可观察到的符号序列进行建模，即 HMM 将每个参考模板用一个数学模型来表示，测试样本代入所有的参考模型中，具有最大概率的模型所代表的语音即为识别结果。

3．人工神经网络法

人工神经网络(ANN)在某种程度上模拟了生物的感知特性，是一种分布式并行处理结构的网络模型，这种网络具有自组织、自学习、输入概括和输入信息特征提取等能力。ANN 本质上是一个自适应非线性动力学系统，具有自适应性、并行性、鲁棒性、容错性和学习特性。目前拥有语音识别的神经网络主要有多层感知器神经网络、镜像基函数神经网络、自组织映射网络和概率神经网络等。

4．混合模型

HMM 经典模型只考虑了系统处于当前时刻的状态，因此使用经典 HMM 模型识别时，会错误地拒绝一部分正确的样本，有一定的误识率。1998 年，A. Ganapathiraju 等人首次提出 HMM/SVM 混合模型[38]；2006 年，S. R. Quchani 和 K. Rahbar 在将 HMM/SVM 模型应用于离散单词的语音识别时，取得了较高的识别效果和识别效率[39]。由于 HMM 适合于处理连续信

号，其结果反映了同类样本的相似度，而 SVM 的输出结果则体现了异类样本间的差异，因此 SVM 适合于分类问题，基于此，如果结合这两个模型的特点，建立一个用于分类的 SVM 和 HMM 的混合模型，就能综合二者的优点，取得较好的识别结果。

HMM 的突出优点是对动态时间序列的建模能力，是一种基于时序累积概率的动态信息处理方法。其缺点是仅考虑了特征的类内变化，而忽略了类间重叠性，因此导致对一些易混淆语音难以识别。ANN 的突出优点是其分类决策能力和对不确定信息的描述能力，其缺点是对时间序列的处理能力尚不尽人意。因此将 HMM 模型的动态建模能力和 ANN 的模式分类能力有机地结合起来就可以互相取长补短，弥补彼此的不足。将 HMM 和 ANN 结合起来用于语音识别曾是 20 世纪 90 年代以来语音识别领域的一个研究热点[40]。

10.2.3 DHMM语音识别系统

语音识别系统主要包括训练和识别两个阶段，无论是训练还是识别，都需要对语音信号进行预处理，并提取其特征参数。训练过程使用语音信号的特征参数对模型进行训练，反复修改模型参数，从而获得参数最优的模型。识别过程使用已经训练出的模型对待识别语音进行识别，根据判决条件，输出识别结果。离散隐马尔可夫模型(Discrete Hidden Markov Model, DHMM)语音识别系统框图如图10.24所示。

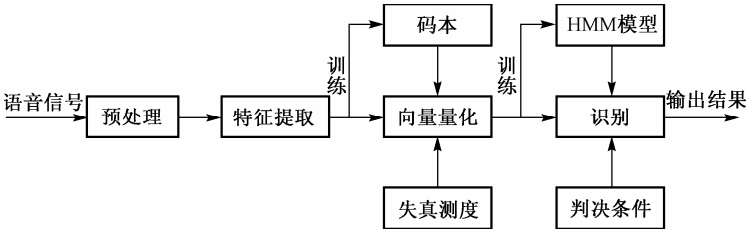


图 10.24 DHMM 语音识别系统框图

在本节的实验中，语音样本使用自录制语音，录制环境为实验室安静环境，采样频率为 22.05 kHz，16 位单声道。由 5 个人分别录制 0-9 的十个数字，每人每个数字读 20 遍，保存为 20 个语音 WAVE 文件，每个数字有 100 个样本，总样本为 1000 个语音样本。

向量量化训练数据是随机选择的 500 个样本的特征参数；每个数字的 HMM 模型训练，是从 100 个样本中随机选取 80 个样本；剩下的 20 个语音样本作为测试语音。

1．预处理

语音信号的预处理包括采样、去除噪音、预加重、分帧、加窗、端点检测等。

(1)预加重

由于语音信号的平均功率受声门激励和口鼻辐射的影响，语音信号从嘴唇发出后，高频端大约在 800 以上有 6 dB/倍频的衰减。因此，在对语音信号处理之前需要对语音信号的高频部分加以提升。预加重的目的是提升高频部分，使信号的频谱变得平坦，以便于进行频谱分析或声道参数分析：

$$y(n) = x(n) - \mu x(n-1) \tag{10.28}$$

式中，$x(n)$为原始信号序列，$y(n)$为预加重后的信号序列，μ 为预加重系数，μ 值接近于 1。在本节中，μ 值选取 0.94。

(2)分帧与加窗

语音信号是一种典型的非平稳过程，但由于语音的形成过程是与发声器官的运动密切相关的，这种物理过程比起声音振动速度来说缓慢得多，因此语音信号可假定为短时(10~30 ms)平稳的。在这样的时间段内，语音信号的频谱特性和语音特征参数可以近似地视为不变的，这样就可以采用平稳信号的处理方法来处理语音信号了。

加窗就是用长度有限的窗函数 $w(m)$截取一段语音信号；窗在语音信号上滑动，将语音信号分成长度为 10~30 ms 的帧的操作称为分帧。为了保证信号提取效果，帧之间都有重叠，重叠的部分为窗长的 1/3 或 1/2。常用的窗函数有矩形窗、汉明窗和汉宁窗，其定义分别为：

(a)矩形窗

$$w(n)=\begin{cases}1, & 0\leq n\leq L-1\\ 0, & \text{其他}\end{cases} \tag{10.29}$$

(b)汉明窗

$$w(n)=\begin{cases}0.54-0.46\cos(2\pi n/(L-1)), & 0\leq n\leq L-1\\ 0, & \text{其他}\end{cases} \tag{10.30}$$

(c)汉宁窗

$$w(n)=\begin{cases}0.5[1-\cos(2\pi n/L)], & 0\leq n\leq L-1\\ 0, & \text{其他}\end{cases} \tag{10.31}$$

其中 L 是窗长，这些窗函数都有低通特性。

在本节中，语音信号采样频率为 22.05 kHz，取 512 个采样点作为一帧，即每帧 23.22 ms；帧之间重叠 212 个采样点，即 9.61 ms；使用汉明窗对语音信号进行加窗处理。

(3)端点检测

在提取特征参数之前需要进行端点检测，这样就可以只处理真正的语音信号数据，从而减少计算量和处理时间。端点检测实质上就是区分语音和噪声。

短时能量指一帧语音信号能量之和，第 n 帧的短时能量用 E_n 表示，其计算公式如下：

$$E_n=\sum_{m=-\infty}^{\infty}[x(m)w(n-m)]^2 \tag{10.32}$$

式中，$x(m)$为语音序列，$w(n-m)$为窗函数。短时能量反映了语音振幅或能量随时间缓慢变化的规律。

由于短时能量对于高电平信号过于敏感，在 CPU 字长有限的情况下容易产生溢出，对于这种情况，可以采用另外一种度量语音幅度的参数短时幅度，短时幅度定义如下：

$$M_n=\sum_{m=-\infty}^{\infty}|x(m)w(n-m)| \tag{10.33}$$

短时过零率 z_n 定义如下：

$$z_n = \sum_{m=-\infty}^{\infty} |\mathrm{sgn}[x(m)] - \mathrm{sgn}[x(m-1)]|\, w(n-m)| \tag{10.34}$$

式中，sgn[] 是符号函数，定义为

$$\mathrm{sgn}\, x = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \tag{10.35}$$

基于短时能量和短时过零率的端点检测方法(双门限比较法)，是一种常用且行之有效的语音端点检测方法。该方法是一种两级判决法，首先使用短时能量做第一次判别，然后使用短时过零率做第二次判别。在使用短时能量做第一次判别时，为了不至于将能量的局部下降点错误当做起止点，常常采用双门限比较的方法，如图10.25所示。

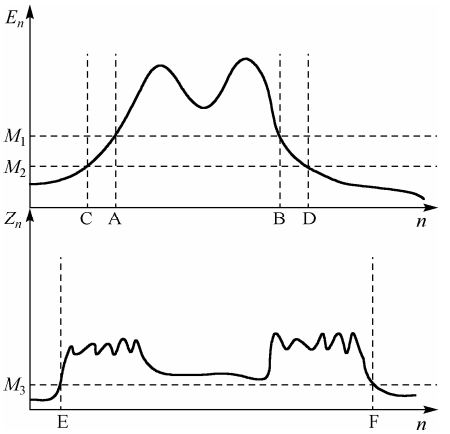


图 10.25　双门限比较方法的端点检测

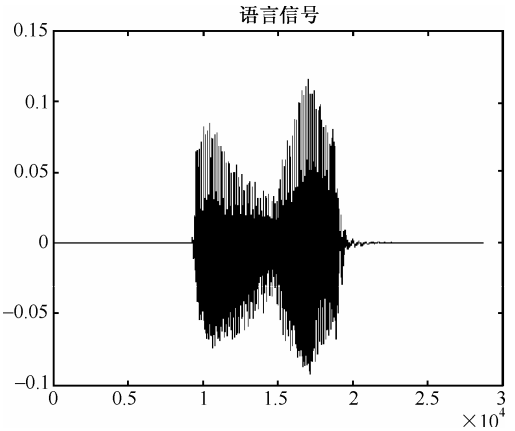
首先根据短时能量 E_n 的轮廓选取一个较高门限 M_1，在大多数情况下语音帧的短时能量都在该门限之上，由此可以粗判为语音起止点位于 AB 段之外；然后根据背景噪声确定一个较低的门限 M_2，分别从 A 向左、从 B 向右搜索，找到短时能量包络与门限 M_2 相交的两个点 C 和 D，于是 CD 段就是双门限方法根据短时能量判断的语音段。第二级判断以短时过零率 Z_n 为标准，从 C 点向左和从 D 点向右搜索，找到短时过零率第一次低于某个门限 M_3 的两点 E 和 F，这就是语音段的起止点。注意，门限 M_3 由背景噪声的过零率 Z_r 确定，一般 M_3 取 Z_r 的 3~5 倍。门限 M_2 和 M_3 都是由背景噪声特性所确定的。因此，在进行起止点判决前，通常要采集若干帧背景噪声，并计算其短时能量和短时过零率，作为选择 M_2 和 M_3 的依据。

图10.26(a)给出了数字‘5’的语音信号，用 MATLAB 函数 wavread()读取语音数据，该函数的第二个参数使用“double”，其返回值在–1.0 至 1.0 之间。用短时幅度和短时过零率处理 wavread()的返回值，得到图10.26(b)~(c)所示的波形，对其进行端点检测；图10.26(b)是短时幅度，其低阈值为 1.45，高阈值为 2.90，分别检测到 A、B 点和 C、D 点；图10.26(c)是短时过零率，其阈值为 5，检测到的语音帧大为 39 帧。

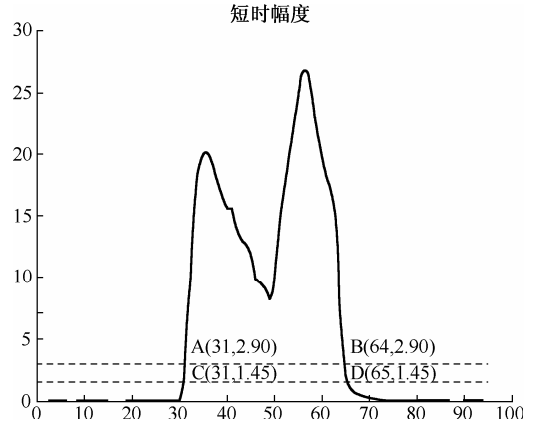
2．特征提取

语音信号的特征参数选择及提取对于语音识别系统至关重要，通常使用两类特征参数：时域特征参数和频域特征参数。常用的时域特征参数有：短时能量或短时幅度、短时过零率。频域特征参数有：线性预测系数(Linear Predictive Coefficients, LPC)、LPC 倒谱系数(LPCC)、线谱对参数(LSP)、共振峰频率、短时频谱、 Mel 频率倒谱系数(Mel Frequency Cepstrum Coefficient, MFCC)；由于 MFCC 能很好地反映人耳的听觉特征，其性能和鲁棒性是频域参数中最好的。

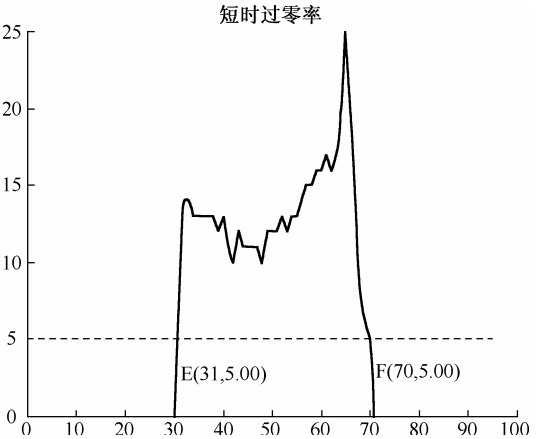
MFCC [41-43]是由 Davies 和 Mermelstein 提出的，该参数利用了听觉原理和倒谱的解相关性，从人耳对不同频率声音的敏感程度的角度，反映了语音短时幅度谱的特征。



(a) 语音信号



(b) 短时幅度检测端点情况



(c) 短时过零率检测端点情况

图 10.26　端点检测示意图

实验发现人耳对不同频率的语音具有不同的感知能力：频率在 1 kHz 以下，人耳的感知能力与频率成线性关系；频率在 1 kHz 以上，人耳的感知能力与频率成对数关系。为了模拟这种人耳的感知特性，人们提出了 Mel 频率的概念，Mel 频率大体上对应于实际频率的对数分布关系。Mel 频率与实际频率的关系如图10.27所示，变换公式如下：

$$f_{\text{mel}} = 2595\lg\left(1+\frac{f_{\text{Hz}}}{700}\right) \tag{10.36}$$

或

$$f_{\text{mel}} = 1127\ln\left(1+\frac{f_{\text{Hz}}}{700}\right) \tag{10.37}$$

式中，f_{Hz} 为实际频率，单位为 Hz，f_{mel} 为 Mel 频率。

MFCC 的计算过程如下：

(a) 原始语音信号 $s(n)$ 通过预处理和端点检测后，得到语音段各帧。每帧的时域信号是 $x(n)$，一帧内的采样数通常取 2 的整数次幂。

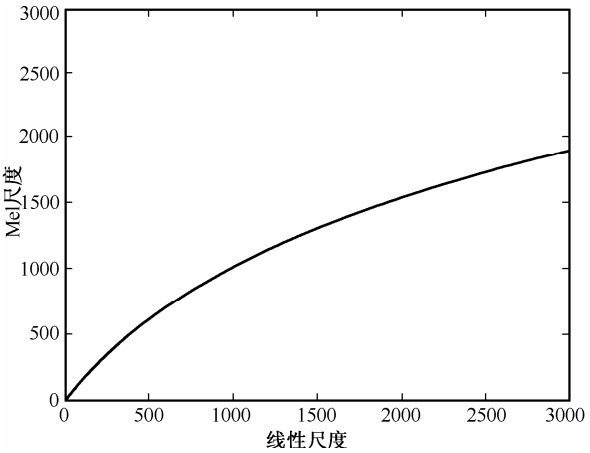


图 10.27 线性频率与 Mel 频率之间的关系

(b)对语音信号 $x(n)$进行离散傅里叶变换，计算各段的线性频谱。离散傅里叶变换的长度为 N。首先计算 $x(n)$的长度 L_x，如果 $L_x < N$，将时域信号 $x(n)$后补若干 0，形成长度为 N 的序列。经过 N 点的离散快速傅里叶变换后，得到线性频谱 $X(k)$：

$$X(k)=\sum_{n=0}^{N-1}x(n)\mathrm{e}^{-\mathrm{j}2\pi nk/N},\quad 0\leq n,k\leq N-1 \tag{10.38}$$

(c)计算线性频谱 $X(k)$的短时能量谱 $P(k)$：

$$P(k)=|X(k)|^2,\quad 0\leq k\leq N-1 \tag{10.39}$$

(d)构造 Mel 滤波器组。根据式(10.36)或式(10.37)将线性频率转换为 Mel 频率。如果 Mel 滤波器组含有 M 个滤波器，则将 Mel 频率范围分成 M 段(通常将 Mel 频率范围平均分成 M 段)。根据式(10.40)计算各滤波器的中心频率，并根据式(10.41)计算各滤波器的相应权值。

中心频率 $f(m)$的计算方法如下：

$$f(m)=\frac{N}{f_s}B^{-1}\left(B(f_l)+m\frac{B(f_h)-B(f_l)}{M+1}\right) \tag{10.40}$$

其中 f_h 和 f_l 是滤波器组的最高频率和最低频率，f_s 是采样频率，N 是傅里叶变换长度。B^{-1} 是函数 B 的反函数。B 的计算公式如下：

$$B=2595\lg\left(1+\frac{f_{\mathrm{Hz}}}{700}\right)\qquad 或\qquad B=1127\ln\left(1+\frac{f_{\mathrm{Hz}}}{700}\right)$$

各个带通滤波器的传递函数如下：

$$H_m(k)=\begin{cases}0, & k<f(m-1),k>f(m+1)\\ \dfrac{k-f(m-1)}{f(m)-f(m-1)}, & f(m-1)\leq k\leq f(m)\\ \dfrac{f(m+1)-k}{f(m+1)-f(m)}, & f(m)<k\leq f(m+1)\end{cases}\qquad (0\leq m<M) \tag{10.41}$$

图10.28是 Mel 滤波器组的示意图。

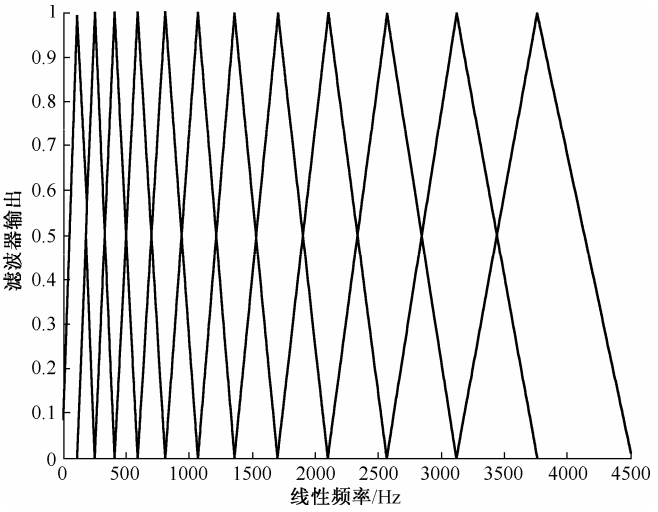


图 10.28　Mel 滤波器组示意图

(a) 计算短时能量谱 $P(w)$ 通过 Mel 滤波器组的输出。计算公式如下：

$$T(m)=\ln\left(\sum_{k=0}^{N-1}|X(k)|^2 H_m(k)\right),\quad 0\leqslant m<M \tag{10.42}$$

(b) 将上述计算结果 $T(m)$ 进行离散余弦变换(DCT)得到 K 阶 Mel 频率倒谱系数 $M_c(K \leqslant M)$：

$$M_c(k)=\sum_{m=0}^{M-1}T(m)\cos\left(\frac{\pi k(m-0.5)}{M}\right),\quad 1\leqslant k\leqslant M \tag{10.43}$$

对图10.26(a)所示的数字“5”语音信号进行特征提取，得到的 MFCC 特征参数如表10.9所示。

表 10.9　数字“5”语音信号的 MFCC 特征参数序列

帧　号	MFCC 分量(1～24)							
1	49.2845	24.1761	−5.94354	13.7403	1.04046	−3.064	4.3169	−5.84143
	−0.637768	0.720412	−5.36405	1.04782	−1.64941	−2.40694	1.6365	−3.22561
	−0.465437	0.525874	−3.39509	0.0910967	−1.12718	−1.35716	0.844071	−2.40174
2	63.2973	30.7343	−8.79031	17.2497	2.2555	−4.19648	5.06889	−6.84774
	−0.745846	0.593332	−7.62218	0.3815	−1.46636	−2.51687	1.48257	−4.00453
	−0.663853	0.200528	−3.5374	1.05188	−1.12746	−1.72205	0.285497	−3.63027
3	64.2312	30.1177	−9.83805	18.6273	4.03989	−2.80073	5.69395	−7.12232
	−0.670572	0.264183	−8.35747	−0.135845	−2.58152	−3.15326	1.75968	−3.76312
	0.128884	0.864422	−3.74336	0.597336	−2.22341	−2.68238	0.569877	−3.07577
M	M							
37	49.8562	18.7659	−13.1827	12.3872	−1.03306	−4.17648	6.03823	−5.23118
	0.472649	1.73417	−5.16673	1.92204	−1.50952	−1.73854	2.90749	−3.35864
	−0.484031	0.824165	−2.33449	1.78053	−1.41993	−1.57759	1.78145	−2.22526

（续表）

帧　号	MFCC 分量(1～24)

38	47.8706	15.8658	−14.8319	11.8206	−1.69918	−3.8952	6.24089	−5.19631
	1.71176	2.61857	−4.81079	1.98037	−1.99561	−0.966938	3.72524	−3.78546
	−0.407402	1.07444	−1.88482	2.34941	−1.77127	−1.2509	2.15547	−2.93977
39	46.5238	15.332	−15.1666	11.6024	−0.0193229	−3.51788	5.0945	−6.24648
	0.677644	3.33857	−3.88191	1.87483	−1.51426	−1.28551	3.21809	−3.69839
	−1.20426	1.12751	−1.79728	1.31501	−1.37507	−0.880922	2.03636	−2.05656

3. 向量量化

向量量化是 20 世纪 80 年代发展起来的一种信源编码技术，它是根据香农信息理论提出的。向量量化就是将向量作为一个整体进行量化，该过程利用了向量各分量之间的关联性，抑制了信号量化过程中的信号冗余。向量量化有着广泛的应用，如语音压缩、图像压缩、波形编码、线性预测编码及语音识别等。

(1) 向量量化过程

设 N 维向量空间的一个向量

$$\boldsymbol{x}=(x_1,x_2,\mathrm{L}\ ,x_N) \tag{10.44}$$

对向量 $\boldsymbol{x}$ 进行向量量化，就是将 $\boldsymbol{x}$ 的 N 个元素作为一个整体(向量)进行量化，即向量量化是将一个 N 维随机向量映射为另一个 N 维向量 $\boldsymbol{y}$ 的过程，记为

$$q[\boldsymbol{x}]=\boldsymbol{y} \tag{10.45}$$

其中 $\boldsymbol{y}$ 是向量集合 Y 中的一个向量，Y 称为码本(Codebook)。Y 由 L 个向量组成，即

$$Y=\{\boldsymbol{y}_i,1\leq\ i\leq\ L\} \tag{10.46}$$

L 是码本的大小，向量 $\boldsymbol{y}_i$ 称为码矢(Code Vector)。码矢是 N 维向量，记为

$$\boldsymbol{y}_i=(y_{i1},y_{i2},\mathrm{L}\ ,y_{iN}),\quad 1\leq\ i\leq\ L \tag{10.47}$$

所以，对向量 $\boldsymbol{x}$ 进行向量量化的过程，就是在码本 Y 中寻找一个与向量 $\boldsymbol{x}$ 最接近的码矢 $\boldsymbol{y}_i$ 的过程。这里所说的最接近，是按照某个失真测度标准来衡量的。

在进行向量量化之前，需要设计出码本。为了设计含有 L 个码矢的码本 Y，需要将 N 维向量空间 X 分成 L 个不同区域，如图10.29所示，这些区域称为胞腔，用 C_i 表示，则

$$\begin{cases} X=\overset{L}{\underset{i=1}{\mathrm{U}}}C_i \\ C_i\ \mathrm{I}\ C_j=\varnothing,\quad i\neq j \end{cases} \tag{10.48}$$

同时，使每个胞腔 C_i 与一个 N 维向量 $\boldsymbol{y}_i$ 相联系，并使 $\boldsymbol{y}_i$ 成为胞腔 C_i 的代表。因此，L 个向量 $\boldsymbol{y}_i$ 的集合就构成了码本 Y。码本设计过程又称为训练过程。

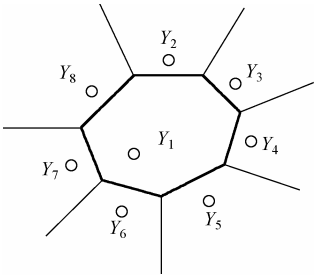


图 10.29 二维向量空间划分成 8 个胞腔

码本设计完成之后，就可以对随机向量 $\boldsymbol{x}$ 进行向量量化。如果向量 $\boldsymbol{x}$ 落在胞腔 C_i 中，那么向量量化器就将代表该胞腔的向量 $\boldsymbol{y}_i$ 作为 $\boldsymbol{x}$ 的量化结果，即

$$q(\boldsymbol{x})=\boldsymbol{y}_i,\quad \boldsymbol{x}\in C_i \tag{10.49}$$

(2) 码本设计

(a) 码本设计准则

码本设计必须遵循以下两个原则。

(a-1) 最近邻选择原则，定义如下：

$$q(\boldsymbol{x})=\boldsymbol{y}_i\text{，当且仅当 }d(\boldsymbol{x},\boldsymbol{y}_i)\leq d(\boldsymbol{x},\boldsymbol{y}_j) \tag{10.50}$$

其中 $j\neq i,1\leq j\leq N$。也就是说，只有 $\boldsymbol{x}$ 与 $\boldsymbol{y}_i$ 的失真小于或等于与其他任意码矢 $\boldsymbol{y}_j$ 之间的失真时，才认为 $\boldsymbol{x}$ 属于 C_i 且被量化为 $\boldsymbol{y}_i$。

(a-2) 平均失真最小原则：在选择码矢时使用平均失真最小原则，即胞腔 C_i 所对应的码矢 $\boldsymbol{y}_i$ 应该是使下式所表示的平均失真最小的向量 $\boldsymbol{y}$：

$$D_i=E[d(\boldsymbol{x},\boldsymbol{y})\,|\,\boldsymbol{x}\in C_i]=\int_{\boldsymbol{x}\in C_i}d(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{10.51}$$

向量 $\boldsymbol{y}$ 称为胞腔 C_i 的形心。

在实际应用中，假设有 M_i 个向量落入胞腔 C_i 中，则该胞腔的平均失真为

$$D_i=\frac{1}{M_i}\sum_{\boldsymbol{x}\in C_i}d(\boldsymbol{x},\boldsymbol{y}_i) \tag{10.52}$$

式中，$\boldsymbol{y}_i$ 是胞腔 C_i 的待定码矢。根据 (a-2) 原则，应该选择使 D_i 最小的向量 $\boldsymbol{y}_i$ 来作为该胞腔的码矢。胞腔的形心如何计算取决于失真的定义，当采用均方误差或者加权均方误差作为失真度定义时，胞腔的形心为

$$\boldsymbol{y}_i=\frac{1}{M_i}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x} \tag{10.53}$$

即将 C_i 中 M_i 个训练向量的均值作为 C_i 的码矢。

(b) 码本设计算法：LBG 算法

码本设计的常用算法是 LBG 算法，在模式识别中称为 K 均值算法。LBG 算法就是将训练向量分成 L 类，并且满足码本设计的两个原则。

设 m 是迭代次数，$C_i(m)$ 是 m 次迭代中的第 i 类，$\boldsymbol{y}_i(m)$ 是 m 次迭代中第 i 类的形心。则 LBG 算法步骤如下：

(b-1) 初始化：令 $m=0$，并建立初始码本 $\{\boldsymbol{y}_i(0),1\leq i\leq L\}$。

(b-2) 分类：根据最近邻原则将训练向量 $\boldsymbol{X}$ 划分成 L 个类，并用 $C_i(m)$ 表示。

(b-3) 码矢更新：$m=m+1$，计算训练向量各类的形心，并用这些形心替换原来的码矢，即

$$\boldsymbol{y}_i(m)=\mathrm{cent}[C_i(m)],\quad 1\leq i\leq L \tag{10.54}$$

(b-4) 结束条件判断：如果 m 次迭代的平均失真 $D(m)$ 相对于第 $m-1$ 次迭代的平均失真 $D(m-1)$ 的减小量低于阈值，则停止迭代计算；否则，返回 (b-2)：

$$D(m)=\sum_{i=0}^{L}\sum_{\boldsymbol{x}\in C_i(m)}d(\boldsymbol{x},\boldsymbol{y}_i(m)) \tag{10.55}$$

上述算法通常能够得到局部最佳码本。如果选择几个不同的初始码本，多次进行上述迭代过程，最后可以得到近似全局最佳的码本。

为了实现 10 个数字的孤立词 DHMM 语音识别系统，向量量化训练使用 LBG 算法，训

练数据来源于 500 个语音文件中提取的 15 306 个特征向量，训练结果为含有 256 个码矢的码本，如表 10.10 所示，循环条件中的阈值为 0.000 001。

表 10.9 中数字“5”的 MFCC 特征参数序列，经过向量量化后获得的数字“5”的观察值序列(码矢在码本中的编号)如下：27, 191, 191, 191, 191, 191, 157, 157, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 159, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 191, 159, 239, 247, 247, 81。

表 10.10 向量量化码本

码矢编号	码矢分量(1～24)							
1	76.9146	36.3035	−15.6655	15.8843	1.47398	−8.76697	1.37115	−9.67831
	−1.90852	2.67331	−4.15734	1.59389	−2.79427	−2.40444	2.28085	−4.49277
	−0.370621	0.234945	−3.84819	1.98633	−1.22298	−1.97189	1.47649	−3.32464
2	13.6091	−1.69124	−8.16869	7.19817	−1.7025	−1.41749	4.03665	−2.566
	1.71141	1.41128	−1.99465	2.85983	−0.54567	−0.765719	1.90077	−1.94721
	0.51816	0.678874	−1.41529	1.40237	−1.09407	−0.810647	1.23558	−1.5031
3	59.1209	24.0092	−14.6157	13.6267	−1.09519	−6.15088	5.23414	−5.88843
	1.35471	3.41086	−4.95205	2.41255	−1.07248	−2.34342	2.9718	−2.58866
	−0.214467	0.18577	−3.31743	1.43274	−1.49921	−1.95852	1.27793	−2.65727
4	−90.2425	−36.7969	27.9988	−12.5371	1.7827	7.97351	−8.2174	7.08382
	0	−2.79196	6.48369	−3.32672	1.86729	3.36089	−3.1299	4.35786
	0	−0.757768	4.28987	−1.93189	2.05574	2.36963	−1.75959	3.77945
M	M							
253	35.1228	13.8148	−10.1507	4.87047	−4.50738	−4.02335	7.0701	0.933026
	3.11674	2.13139	−4.4073	−0.0392122	−0.200173	0.939635	3.74643	−1.19161
	−0.712575	−0.653297	−2.36289	0.996815	−0.616406	−0.909371	0.844282	−1.61779
254	81.6565	38.0721	−18.6267	14.5062	0.565334	−10.4026	0.463836	−9.33237
	−0.8698	2.75822	−4.39909	2.24271	−2.67483	−2.90839	2.17245	−4.21029
	0.105913	0.310392	−3.88469	2.19611	−1.33808	−2.04017	1.37524	−3.63941
255	69.5435	33.9617	−12.5753	9.64786	−8.22122	−9.16351	7.71658	−4.5569
	0.27837	2.40701	−6.80407	− 0.483525	−2.18997	−0.835835	4.05467	−3.00143
	−0.130606	0.514104	−3.34494	1.74148	−1.65772	−2.43737	1.50519	−2.63321
256	95.6946	42.4533	−29.4584	7.56639	−4.2838	−11.2273	7.51033	−3.46727
	1.08026	2.82159	−6.20973	2.20521	−1.29644	−2.77167	2.61258	−4.03647
	−0.722276	0.327589	−3.74611	1.33043	−2.12307	−2.52175	1.03962	−3.23073

4．隐马尔可夫模型

(1)隐马尔可夫模型的基本概念

(a) 马尔可夫链

马尔可夫链是马尔可夫随机过程的特殊情况，即马尔可夫链是状态和时间参数都离散的马尔可夫过程。图10.30给出一个含有 5 个状态的马尔可夫链，其中 S_1~S_5 是状态，a_{ij} 是转移概率(i, j =1, 2, 3, 4, 5)。

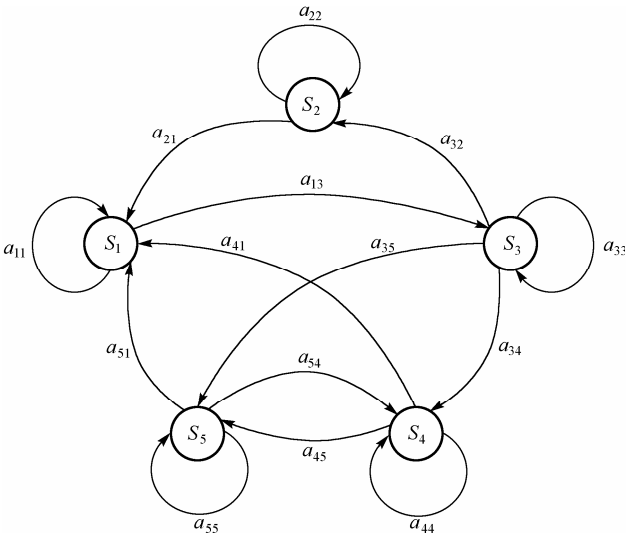


图 10.30　含有 5 个状态的马尔可夫链

通常，马尔可夫链共有 N 个状态，分别用 $S_1, S_2, \cdots, S_N$ 表示。随机序列 x_t，在 m 时刻所处的状态记为 q_m，在 $m + k$ 时刻所处的状态 q_{m+k} 的概率，只与它在 m 时刻所处的状态 q_m 有关，而与 m 时刻以前所处的状态无关，即

$$P(x_{m+k} = q_{m+k} \mid x_m = q_m, x_{m-1} = q_k, \cdots, x_1 = q_t) = P(x_{m+k} = q_{m+k} \mid x_m = q_m) \tag{10.56}$$

式中，$q_1, q_2, \cdots, q_m, q_{m+k} \in (S_1, S_2, \cdots, S_N)$。则称 x_t 是马尔可夫链，并且称

$$P_{ij}(m, m+k) = P(q_{m+k} = S_j \mid q_m = S_i) \qquad 1 \leqslant i, j \leqslant N, m, k \text{ 为正整数} \tag{10.57}$$

为 k 步转移概率，当 $P_{ij}(m, m+k)$ 与 m 无关时，称这个马尔可夫链为齐次马尔可夫链，此时

$$P_{ij}(m, m+k) = P_{ij}(k) \tag{10.58}$$

当 k = 1 时，$P_{ij}(1)$ 称为一步转移概率，简称为转移概率，记为 a_{ij}。所有转移概率 a_{ij} ($1 \leqslant i, j \leqslant N$) 可以构成一个转移概率矩阵，即

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \qquad 0 \leqslant a_{ij} \leqslant 1, \sum_{j=1}^{N} a_{ij} = 1 \tag{10.59}$$

由于 k 步转移概率 $P_{ij}(k)$ 可由转移概率 a_{ij} 得到，因此描述马尔可夫链的最重要参数就是转移概率矩阵 $\boldsymbol{A}$。但转移概率矩阵 $\boldsymbol{A}$ 不能决定初始分布，即由 $\boldsymbol{A}$ 求不出 $q_1 = S_i$ 的概率，因此，完全描述马尔可夫链，除矩阵 $\boldsymbol{A}$ 之外，还必须引进初始概率向量 $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_N)$，其中

$$\pi_i = P(q_1 = S_i),\quad 1 \leq i \leq N \tag{10.60}$$

显然有

$$0 \leq \pi_i \leq 1,\quad \sum_{i=1}^{N} \pi_i = 1 \tag{10.61}$$

上述马尔可夫链可以称为可观察马尔可夫链，因为系统的输出就是系统的状态。在实际中，马尔可夫链的每个状态可以对应于一个可观察到的物理事件。例如，天气预报可视为一个简单的含有 3 个状态的马尔可夫模型，3 个状态分别是雨雪(S_1)、多云(S_2)、晴(S_3)，描述该马尔可夫模型的参数如下：

初始概率向量：

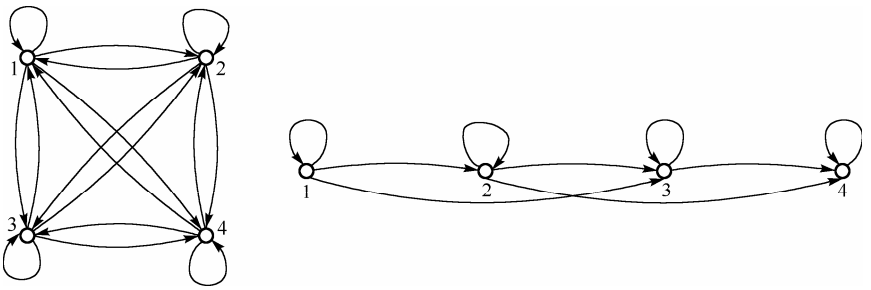
$$\boldsymbol{\pi} = (1, 0, 0)$$

转移概率矩阵：

$$\boldsymbol{A} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.0 & 0.1 & 0.8 \end{bmatrix}$$

(b) 马尔可夫链形状

马尔可夫链由$\boldsymbol{\pi}$、$\boldsymbol{A}$ 组成，通常根据实际应用，选择由$\boldsymbol{\pi}$、$\boldsymbol{A}$ 决定的马尔可夫链形状。两种典型的马尔可夫链如图10.31所示。



(a) 各态历经马尔可夫链　　(b) 从左到右马尔可夫链

图 10.31　两种典型的马尔可夫链(状态数 $N = 4$)

图 10.31(a)所示是各态历经的马尔可夫链，其特点是：模型中的每个状态都可以由模型中的其他任何状态转移到；模型状态转移概率矩阵中的每个元素都为正数，即 $a_{ij} > 0$。

图10.31(b)所示是从左到右的马尔可夫链，其特点是：随着时间的增加，状态序号也增加或者保持在原状态；起始状态为序号最小的状态，终止状态为序号最大的状态。描述图10.31(b)的参数如下：

初始状态为

$$\boldsymbol{\pi} = (1, 0, 0, 0)$$

状态转移概率为

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2)隐马尔可夫模型的定义

由于实际问题比马尔可夫链模型所描述的更加复杂，观察到的事件并不是与状态一一对应的，而是与一组概率分布相联系，这样的模型称为隐马尔可夫模型(Hidden Markov Model, HMM)。HMM 是一个双重随机过程：一个随机过程是马尔可夫链，描述状态的转移；另一个随机过程描述状态与观测值之间的统计对应关系。这样，站在观察者的角度，就只能直接看到观测值，而不能直接看到状态；通过另外一个随机过程去感知状态的存在及其特性。

下面通过球和缸的实验来说明 HMM 的概念。设有 N 个缸，每个缸装有多种颜色的球。球的颜色用一组概率分布表示，如图10.32所示。实验操作如下：根据某个初始化分布描述，随机地选择 N 个缸中的一个，例如第 i 个缸，再根据这个缸中彩色球颜色的概率分布，随机地选择一个球，记下球的颜色，假设记为 O_1，再把球放回缸中；又根据缸的转移概率分布，随机地选择下一个缸，重复以上操作。操作完成后，可以得到一个描述球的颜色的序列 $O_1, O_2,$ L ，由于这是观察到的事件，所以称为观测值序列。但是，缸之间的转移以及每次选择的缸被隐藏起来了，并不能直接观察到，而且每个缸中选取球的颜色并不是与缸一一对应，而是由该缸中球的颜色的概率分布随机决定的。此外，每次选择哪个缸，则由一组转移概率所决定。

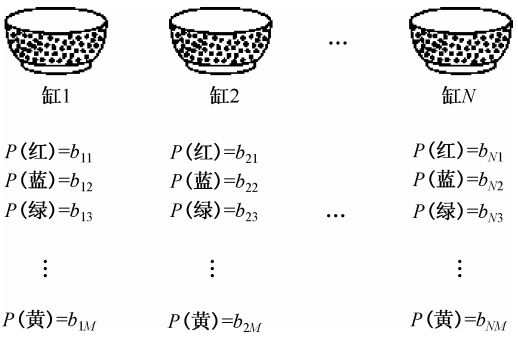


图 10.32　球和缸实验

有了前面讨论的马尔可夫链及球和缸的实验，就可以给出 HMM 的定义，或者说，一个 HMM 可以由下列参数描述：

- N：马尔可夫链状态数目。N 个状态为 $S_1, S_2, \cdots, S_N$, t 时刻马尔可夫链所处的状态为 q_t，$q_t \in (S_1, S_2, \mathrm{L}\ , S_N)$。在球和缸实验中缸就相当于状态。
- M：每个状态对应的可能的观测值数目。M 个观测值为 $V_1, V_2, \cdots, V_M$，t 时刻观察到的观察值为 O_t，其中 $O_t \in (V_1, V_2, \mathrm{L}\ , V_M)$。在球和缸实验中所选球的颜色就是观测值。
- $\boldsymbol{\pi}$：初始状态概率向量。$\boldsymbol{\pi} = (\pi_1, \pi_2, \mathrm{L}\ , \pi_N)$，其中

$$\pi_i = P(q_1 = S_i),\quad 1 \leq i \leq N$$

在球和缸实验中，π_i 指开始时选取第 i 个缸的概率。

- $\boldsymbol{A}$：状态转移概率矩阵。$\boldsymbol{A}=(a_{ij})_{N\times N}$，其中

$$a_{ij}=P(q_{t+1}=S_j \mid q_t=S_i),\quad 1\leq i,j\leq N$$

在球和缸实验中，a_{ij}指在当前选取的缸是第 i 个缸的状态下，选择的下一个缸是第 j 个缸的概率。

- $\boldsymbol{B}$：观察值概率矩阵。$\boldsymbol{B}=(b_{jk})_{N\times M}$，其中

$$b_{jk}=P(O_t=V_k \mid q_t=S_j),\quad 1\leq j\leq N, 1\leq k\leq M$$

在球和缸实验中，b_{jk}指在选择第 j 个缸的状态下选择颜色 k 的概率。

一个 HMM 可记为

$$\lambda=(N,M,\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B}) \tag{10.62}$$

或简记为

$$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B}) \tag{10.63}$$

从上述内容可看出经典 HMM 做了两个重要假设：

- 状态转移假设：从 t 时刻的状态向 t +1 时刻的状态的转移概率只与 t 时刻的状态有关。
- 输出值假设：在 t 时刻的输出观察值的概率，只取决于当前时刻 t 所处的状态。

一个 HMM 系统从 t = 1 时刻开始运行到 t = T 时刻结束，可得到 T 个观察值，从而构成观察值向量$\boldsymbol{O}=(O_1,O_2,\mathrm{L}\ ,O_T)$，称为观察值向量序列。对于 HMM 来说，每次系统运行所产生的状态转移序列$q=(q_1,q_2,\mathrm{L}\ ,q_T)$是不可见的，观察者只能观察到观察向量序列 $\boldsymbol{O}$。

(3)隐马尔可夫模型的三个基本问题及解决方法

为了将 HMM 应用于实际，必须要解决 HMM 的三个基本问题。这三个基本问题分别是：

(P1)识别问题：已知观察值序列$O=O_1,O_2,\mathrm{L}\ ,O_T$和模型$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$，如何计算由模型 λ 产生观察值序列 O 的概率 $P(O|\lambda)$。

(P2)解码问题：已知观察值序列$O=O_1,O_2,\mathrm{L}\ ,O_T$和模型$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$，如何获得由模型 λ 产生观察值序列 O 的概率最大时对应的状态序列$q=q_1,q_2,\mathrm{L}\ ,q_T$。

(P3)训练问题：如何根据系统给定的观察值序列$O=O_1,O_2,\mathrm{L}\ ,O_T$来调整模型$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$的相关参数，使得概率 $P(O|\lambda)$最大。

问题(P1)是识别问题，即观察序列概率计算问题，在给定模型 O 和观察值序列 λ 的情况下，计算由模型 λ 产生观察值序列 O 的概率，用来评价模型 λ 与观察值序列 O 的匹配程度。问题(P2)是解码问题，即最优状态序列搜索问题，目的在于在已知模型 λ 和观察值序列 O 的情况下，获得由模型 λ 产生观察值序列 O 的“最佳”状态序列。问题(P3)是训练问题，即模型参数估计问题。

(a)观察序列概率计算问题——前向-后向算法

前向-后向算法用来计算在给定观察值序列$O=O_1,O_2,\mathrm{L}\ ,O_T$和模型$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$时，由模型模型 λ 产生观察值序列 O 的概率 $P(O|\lambda)$。

(a1)前向算法

定义前向变量为

$$\alpha_t(i)=P(O_1,O_2,\text{L}\ ,O_t,q_t=S_i\mid\lambda),\qquad 1\leq\ t\leq\ T \tag{10.64}$$

初始化

$$\alpha_1(i)=\pi_i b_i(O_1),\qquad 1\leq\ i\leq\ N \tag{10.65}$$

递推

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1}),\quad 1\leq\ t\leq\ T-1,1\leq\ j\leq\ N \tag{10.66}$$

终结

$$P(O\mid\lambda)=\sum_{i=1}^{N}\alpha_T(i) \tag{10.67}$$

其中，$b_j(O_{t+1})=b_{jk}\mid O_{t+1}=V_k$。

(a2)后向算法

与前向算法类似，定义后向变量为

$$\beta_t(i)=P(O_{t+1},O_{t+2},\text{L}\ ,O_T\mid q_t=S_i,\lambda),\quad 1\leq\ t\leq\ T-1 \tag{10.68}$$

初始化

$$\beta_T(i)=1,\quad 1\leq\ i\leq\ N \tag{10.69}$$

递推

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\beta_{t+1}(j)\qquad 1\leq\ t\leq\ T-1,1\leq\ i\leq\ N \tag{10.70}$$

终结

$$P(O\mid\lambda)=\sum_{i=1}^{N}\pi_i b_i(O_1)\beta_1(i) \tag{10.71}$$

前向变量和后向变量计算出后，考察整体概率，即整个观察值序列 O 的概率：

$$P(O\mid\lambda)=\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)=\sum_{i=1}^{N}\alpha_T(i)=\sum_{i=1}^{N}\alpha_1(i)\beta_1(i),\quad 1\leq\ t\leq\ T-1 \tag{10.72}$$

另一种形式为

$$P(O\mid\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j),\quad 1\leq\ t\leq\ T-1 \tag{10.73}$$

实际计算中，首先计算出对于每个时刻 t 和每个状态 i 的前向变量和后向变量，然后使用上述公式，计算出模型 λ 产生观察值序列 O 的概率。这两个公式称为全概率公式。

(b)最优状态序列搜索问题——Viterbi 算法

Viterbi 算法解决了给定观察值序列 $O=O_1,O_2,\text{L}\ ,O_T$ 和模型 $\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$，确定由模型 λ 产生观察值序列 O 的最优状态序列 $q=q_1,q_2,\text{L}\ ,q_T$ 的问题。

Viterbi 算法可以描述如下。

定义 $\delta_t(i)$ 为时刻 t 时沿一条路径 $q_1, q_2, \mathrm{L}\ , q_t$（且 $q_t = S_i$）产生出观察值序列 O 的最大概率，即

$$\delta_t(i) = \max_{q_1, q_2, \mathrm{L}\ , q_{t-1}} P(q_1, q_2, \mathrm{L}\ , q_{t-1}, q_t = S_i, O_1, O_2, \mathrm{L}\ , O_t \mid \lambda) \tag{10.74}$$

那么，求取最优状态序列 q 的过程为

(b1)初始化

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \tag{10.75}$$

$$\phi_1(i) = 0, \quad 1 \leq i \leq N \tag{10.76}$$

(b2)递推

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij} \right] b_j(O_t), \quad 2 \leq t \leq T, 1 \leq j \leq N \tag{10.77}$$

$$\phi_t(j) = \arg\max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij} \right], \quad 2 \leq t \leq T, 1 \leq j \leq N \tag{10.78}$$

(b3)终结

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{10.79}$$

$$q_T^* = \arg\max_{1 \leq i \leq N} [\delta_T(i)] \tag{10.80}$$

(b4)状态序列求取

$$q_t^* = \phi_{t+1}(q_{t+1}^*), \quad 1 \leq t \leq T-1 \tag{10.81}$$

在实际应用中，通常用对数形式的 Viterbi 算法，这样将避免进行大量的乘法计算，减少了计算量，同时还可以保证很高的动态范围，而不会由于过多的连乘而导致溢出问题。

(c)模型参数估计问题——Baum-Welch 算法

Baum-Welch 算法实际上用于解决模型参数估计问题，即 HMM 训练问题，或者说，给定一个观察值序列 $O = O_1, O_2, \mathrm{L}\ , O_T$，该算法能确定一个模型 $\lambda = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$，使 $P(O \mid \lambda)$ 最大。

由式(10.64)和式(10.68)定义的前向变量和后向变量可知

$$P(O \mid \lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad 1 \leq t \leq T-1 \tag{10.82}$$

这里，求取 λ 使 $P(O \mid \lambda)$ 最大是一个泛函数极值问题。但是由于给定的训练序列有限，所以不存在一个最佳的方法来估计 λ。在这种情况下，Baum-Welch 算法采用递归的思想，使 $P(O \mid \lambda)$ 局部最大，最后得到模型参数 $\lambda = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$。

定义 $\xi_t(i, j)$ 为给定训练序列 O 和模型 λ 时，时刻 t 时马尔可夫链处于 S_i 状态和时刻 t+1 时处于 S_j 状态的概率，即

$$\xi_t(i, j) = P(O, q_t = S_i, q_{t+1} = S_j \mid \lambda) \tag{10.83}$$

可以推导出

$$\xi_t(i,j)=\left[\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)\right]\Big/P(O|\lambda) \tag{10.84}$$

那么，时刻 t 时马尔可夫链处于 S_i 状态的概率为

$$\xi_t(i)=P(O,q_t=\theta_i|\lambda)=\sum_{j=1}^{N}\xi_t(i,j)=\alpha_t(i)\beta_t(i)/P(O|\lambda) \tag{10.85}$$

因此，$\sum_{t=1}^{T-1}\xi_t(i)$ 表示从 S_i 状态转移出去的次数的期望值，而 $\sum_{t=1}^{T-1}\xi_t(i,j)$ 表示从 S_i 状态转移到 S_j 状态的次数的期望值。由此，导出了 Baum-Welch 算法中著名的重估公式：

$$\overline{\pi_i}=\text{在时刻}t=1\text{时处于状态}S_i\text{的期望值}=\xi_1(i) \tag{10.86}$$

$$\overline{a_{ij}}=\frac{\text{从状态}S_i\text{转移到状态}S_j\text{的期望值}}{\text{从状态}S_i\text{转移出去的期望值}}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\xi_t(i)} \tag{10.87}$$

$$\overline{b_j}(k)=\frac{\text{处于状态}S_j\text{且观察值是}V_k\text{的期望值}}{\text{处于状态}S_j\text{的期望值}}=\frac{\sum_{t=1\text{且}O_t=V_k}^{T}\xi_t(j)}{\sum_{t=1}^{T}\xi_t(j)} \tag{10.88}$$

HMM 参数 $\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$ 的求取过程为：根据观察值序列 O 和选取的初始模型 $\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$，由重估公式(10.86)、(10.87)和(10.88)，求得一组新参数 $\overline{\pi}_i$，$\overline{a}_{ij}$ 和 $\overline{b}_{ij}$，亦即得到了一个新的模型 $\overline{\lambda}=(\overline{\boldsymbol{\pi}},\overline{\boldsymbol{A}},\overline{\boldsymbol{B}})$。可以证明，$P(O|\overline{\lambda})>P(O|\lambda)$，重复这个过程，逐步改进模型参数，直到 $P(O|\overline{\lambda})$ 收敛，即不再明显增大，此时的 $\overline{\lambda}$ 即为所求之模型。

(4) 多观察值序列训练

设 L 个观察值序列用于训练模型，则重估公式为

$$\overline{a}_{ij}=\frac{\sum_{l=1}^{L}\text{第}l\text{个序列从状态}i\text{转移到状态}j\text{的次数}}{\sum_{l=1}^{L}\text{第}l\text{个序列状态}i\text{的状态数目}}=\frac{\sum_{l=1}^{L}trans\_counts(i,j,l)}{\sum_{l=1}^{L}state\_counts(i,l)} \tag{10.89}$$

$$\overline{b}_{jk}=\frac{\sum_{l=1}^{L}\text{第}l\text{个序列位于状态}j\text{且观察值是}k\text{的次数}}{\sum_{l=1}^{L}\text{第}l\text{个序列位于状态}j\text{的次数}}=\frac{\sum_{l=1}^{L}vect\_counts(k,j,l)}{\sum_{l=1}^{L}state\_counts(j,l)} \tag{10.90}$$

注意，

$$state\_count(j,l)|_{j=i}=state\_counts(i,l) \tag{10.91}$$

因此，有

$$\bar{a}_{ij}=\sum_{l=1}^{L}R_{il}\cdot\frac{trans\_counts(i,j,l)}{state\_counts(i,l)}=\sum_{l=1}^{L}R_{il}\cdot\bar{a}_{ijl} \tag{10.92}$$

$$\bar{b}_{jk}=\sum_{l=1}^{L}R_{jl}\cdot\frac{vect\_counts(k,j,l)}{state\_counts(j,l)}=\sum_{l=1}^{L}R_{jl}\cdot\bar{b}_{jkl} \tag{10.93}$$

其中，

$$R_{jl}=\frac{state\_counts(j,l)}{\sum_{l'=1}^{L}state\_counts(j,l')} \tag{10.94}$$

分析式(10.92)和式(10.93)可知，用 L 个观察值序列进行训练获取 HMM 参数时，可以分别用每个训练序列获取相应的 HMM 参数，然后再合并，而合并的权值取决于每个状态的数目。因此，可认为是状态数目描述了 HMM 的相对可靠程度。同理，当需要合并 K 个 HMM 时，对任意状态 j 的权值可由式(10.94)确定。

因为这种估计权值的方法是由 Baum-Welch 算法的重估公式推导出的，所以是局部最佳的，而且每个状态对应一个权值，这样就使得合并生成的模型更好。

(5)基于驻留状态的隐马尔可夫模型

经典 HMM 的马尔可夫链由初始状态矩阵 $\boldsymbol{\pi}$ 和状态转移矩阵 $\boldsymbol{A}$ 来描述，状态 S_i 产生 d 个观察值的概率为

$$p_i(d)=(a_{ii})^d(1-a_{ii}) \tag{10.95}$$

该概率值描述了状态 S_i 的驻留时间。$p_i(d)$ 呈指数分布，当 $d=0$ 时取得最大值。这与语音段物理意义不符，因为在语音处理中使用 HMM 时，状态与语音单位相对应，而这些语音单位具有稳定的分布。很多研究人员针对经典 HMM 的这个缺陷提出了各自的改进方法，基本思想是对描述马尔可夫链的参数进行修正，如增加一项描述状态驻留时间参数。

一种方法称为非参数方法，该方法令 $a_{ii}=0$，同时增加状态驻留时间概率分布 $p_i(d)$，$d=1,2,\text{L},D$，其中 D 为所有状态的最长可能驻留时间。这样，HMM 产生观察值序列的过程为：由 π_i 选择初始状态 q_i，根据 $p_{q_1}(d_1)$ 确定驻留状态时间 d_1，产生 d_1 个观察值 $O_1, O_2,\text{L}, O_{d1}$，其概率为 $\prod_{t=1}^{d_1}b_{q_1}(O_t)$，然后根据 $a_{q_1q_2}$ 选择下一个状态 q_2；重复这个过程直到整个观察值序列 O 生成完毕。

由于改变了描述马尔可夫链的参数，前向-后向算法所用变量都需重新定义[44]：

➢ 前向变量

$$\hat{\alpha}_t(i)=P\{O_1,O_2,\text{L},O_t,q_t=S_i\}=\sum_{d=1}^{D}\alpha_t(i,d)p_i(d) \tag{10.96}$$

$$\hat{\alpha}_t(i)=P\{O_1,O_2,\text{L},O_t,q_t\neq S_i,q_{t+1}=S_i\}=\sum_{j=1,j\neq i}^{N}\hat{\alpha}_t(j)a_{ji} \tag{10.97}$$

其中，

$$\alpha_t(i,d)=\begin{cases}\alpha_{t-1}(i)b_i(O_t), & d=1\\ \alpha_{t-1}(i,d-1)b_i(O_t), & 2\le d\le D\end{cases} \tag{10.98}$$

初始值

$$\alpha_1(i,d)=\begin{cases}\pi_i b_i(O_1), & d=1\\ 0, & 2\le d\le D\end{cases} \tag{10.99}$$

➢ 后向变量

$$\hat{\beta}_t(i)=P\{O_t,O_{t+1},\cdots,O_T,q_t=S_i\}=\sum_{d=1}^{D}\beta_t(i,d)p_i(d) \tag{10.100}$$

$$\hat{\beta}_t(i)=P\{O_t,O_{t+1},\cdots,O_T,q_{t-1}=S_i,q_t\neq S_i\}=\sum_{j=1,j\neq i}^{N}a_{ij}\hat{\beta}_t(j) \tag{10.101}$$

其中，

$$\beta_t(i,d)=\begin{cases}b_i(O_t)\hat{\beta}_{t+1}(i), & d=1\\ b_i(O_t)\beta_{t+1}(i,d-1), & 2\le d\le D\end{cases} \tag{10.102}$$

初始值

$$\beta_T(i,d)=\begin{cases}b_i(O_T), & d=1\\ 0, & 2\le d\le D\end{cases} \tag{10.103}$$

由模型 λ 产生观察值序列 O 的概率 $P(O|\lambda)$：

$$P(O|\lambda)=\sum_{i=1}^{N}\hat{\alpha}_T(i) \tag{10.104}$$

重估参数

$$\overline{\pi}_i=\frac{v_1(i)}{\sum_{j=1}^{N}v_1(j)} \tag{10.105}$$

$$\overline{a}_{ij}=\frac{\sum_{t=1}^{T-1}\mu_t(i,j)}{\sum_{j=1}^{N}\sum_{t=1}^{T-1}\mu_t(i,j)} \tag{10.106}$$

$$\overline{p}_i(d)=\frac{\sum_{t=1}^{T}\omega(t,i,d)}{\sum_{d=1}^{D}\sum_{t=1}^{T}\omega(t,i,d)} \tag{10.107}$$

$$\overline{b}_i(k)=\frac{\sum_{t=1\text{且}q_t=k}^{T}v_t(i)}{\sum_{t=1}^{T}v_t(i)} \tag{10.108}$$

其中，

$$\omega(t,i,d)=\hat{\alpha}_{t-1}(i)\beta_t(i,d)p_i(d) \tag{10.109}$$

$$\mu_t(i,j)=\hat{\alpha}_t(i)a_{ij}\hat{\beta}_{t+1}(j) \tag{10.110}$$

$$v_t(i)=\begin{cases}\hat{\alpha}_T(i), & t=T\\ v_{t+1}(i)+\sum\limits_{j=1,j\neq i}^{N}(\mu_t(i,j)-\mu_t(j,i)), & 1\leq t<T\end{cases} \tag{10.111}$$

增加了 $p_i(d)$ 参数的 HMM，比经典 HMM 具有更好的性能，这是以增大计算量和存储空间为代价的。

基于驻留状态的隐马尔可夫模型训练的参数选择为：状态数为 7、观察值数目为 256（与向量量化码本大小相同）、状态驻留最大时间为 25。

根据汉语语音信号的特性，选择从左到右的马尔可夫链，状态转移允许跳转的状态数为 1，如图10.33所示。

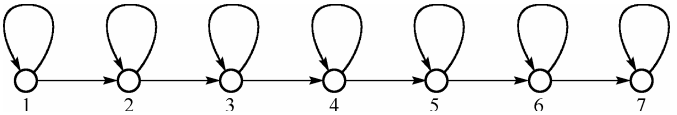


图 10.33　7 状态从左到右马尔可夫链

图10.33的马尔可夫链的初始状态概率矩阵 $\boldsymbol{\pi}$ 、状态转移概率矩阵 $\boldsymbol{A}$ 的初始值如下：

$$\boldsymbol{\pi}=(1,0,0,0,0,0,0)$$

$$\boldsymbol{A}=\begin{bmatrix}a_{11}&a_{12}&0&0&0&0&0\\0&a_{22}&a_{23}&0&0&0&0\\0&0&a_{33}&a_{34}&0&0&0\\0&0&0&a_{44}&a_{45}&0&0\\0&0&0&0&a_{55}&a_{56}&0\\0&0&0&0&0&a_{66}&a_{67}\\0&0&0&0&0&0&1\end{bmatrix}$$

在 HMM 训练过程中，使用式(10.111)进行参数重估时，会出现估计出的概率为负值的情况（与实际物理意义不符），所以将式(10.111)修改为

$$v_t(i)=\sum_{\tau=1}^{t-1}\hat{\alpha}_\tau(i)\sum_{d=t-\tau}^{\min(D,T-\tau)}p_i(d)\beta_{\tau+1}(i,d) \tag{10.112}$$

基于驻留状态的 HMM 使用均值法进行参数初始化，初始化后参数如表 10.11 所示。循环条件中，阈值为 0.000 001。

用 10.2.3.3 节中数字“5”的单观察值序列训练的 HMM 参数如表 10.12 所示。

表 10.11　HMM 初始化参数：初始状态概率和状态转移概率

初始状态概率	1	0	0	0	0	0	0
状态转移概率	0.5	0.5	0	0	0	0	0
	0	0.5	0.5	0	0	0	0
	0	0	0.5	0.5	0	0	0
	0	0	0	0.5	0.5	0	0
	0	0	0	0	0.5	0.5	0
	0	0	0	0	0	0.5	0.5
	0	0	0	0	0	0	1

表 10.12　数字“5”单观察值序列训练的 HMM 参数

状　　态	1	2	3	4	5	6	7
状态转移概率	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	1	0	0
	0	0	0	0	0	1	0
	0	0	0	0	0	0.026277	0.973723
	0	0	0	0	0	0	1

状态	观察值概率							
1	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.07141	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.142821	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.785515	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001

（续表）

状态	观察值概率							
M	M							
7	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.999745	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001

状态	驻留状态概率							
1	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.999976	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
2	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.999976	0.000001	0.000001	0.000001	0.000001
	0.000001							

（续表）

3	0.999976	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
4	0.999976	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
5	0.000001	0.999976	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
6	0.999976	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
7	0.999976	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							

用数字“5”的多观察值序列(表 10.13)训练的 HMM 参数如表 10.14 所示。

表 10.13　数字“5”的观察值序列

观察值序列长度	观察值序列									
32	27	191	191	191	157	191	191	191	191	191
	191	191	191	191	191	191	191	191	191	157
	157	157	157	1	1	1	250	163	231	247
	81	119								
32	27	191	191	191	191	191	191	191	191	191
	191	191	191	191	191	191	191	191	191	191
	157	157	157	157	157	157	1	1	174	163
	175	167								
39	27	191	191	191	191	191	157	157	191	191
	191	191	191	191	191	191	191	191	159	191
	191	191	191	191	191	191	191	191	191	191
	191	191	191	191	159	239	247	247	81	
38	213	223	191	157	191	157	157	157	157	157
	157	157	191	191	191	191	191	191	157	157
	157	157	157	157	157	157	157	157	1	1
	1	1	1	163	163	3	231	201		
37	27	159	157	157	157	157	157	157	157	157
	157	191	191	191	191	191	191	191	191	191
	191	191	157	157	157	1	1	1	1	1
	1	221	250	163	60	231	231			

表 10.14 数字“5”多观察序列训练的 HMM 参数

初始状态概率	1	0	0	0	0	0	0
状态转移概率	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	1	0	0
	0	0	0	0	0	1	0
	0	0	0	0	0	0	1
	0	0	0	0	0	0	1

状态	观察值概率							
1	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.044353	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.231702	0.000001	0.008952	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.692469	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.011137	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.011137	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001

（续表）

状态	观察值概率							
M	M							
7	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.052291	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.947455	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001

状态	驻留状态概率							
1	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.246928	0.000001	0.000001
	0.000001	0.200498	0.16884	0.16884	0.000001	0.000001	0.000001	0.000001
	0.214874							

（续表）

状态	驻留状态概率							
2	0.214874	0.000001	0.000001	0.16884	0.000001	0.16884	0.000001	0.000001
	0.000001	0.200498	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.246928	0.000001	0.000001	0.000001	0.000001
	0.000001							
3	0.246928	0.16884	0.000001	0.16884	0.41537	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
4	0.415768	0.584209	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
5	0.538176	0.461801	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
6	0.916205	0.083772	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							
7	0.999976	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
	0.000001							

5．孤立词语音识别

语音识别过程如图10.24所示，输入的语音信号经过预处理、特征提取和向量量化后，得到的观察值序列，采用 Viterbi 算法与已经训练的非特定人的 10 个数字语音的 HMM 模型进行匹配，找出概率最大的 HMM 模型，所代表的数字模型就是识别结果。

对 10 个数字的 200 个识别样本进行识别，正确识别样本为 191，正确识别率为 95.5%。

参考文献

[1] 朱志刚译. 数字图像处理, 北京：电子工业出版社, 1998, 143-154.

[2] 章为川. 基于神经网络的车牌识别系统的研究与设计[D], 西南交通大学, 2006.

[3] Wenjing Jia, Huaifeng Zhang, Xiangjian He. Region-based license plate detection[J], Journal of Network and Computer Applications, 2007, 30(4): 1324-1333.

[4] 陶军. 车牌识别技术研究与实现[D], 南京理工大学, 2004.

[5] 冈萨雷斯. 数字图像处理[M], 北京：电子工业出版社, 2003, 70-72.

[6] 王慧琴. 数字图像处理[M], 北京：北京邮电大学出版社, 2003, 104-106.

[7] 杨超. 汽车牌照自动识别技术的研究[D], 燕山大学, 2007.

[8] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen. *"Automatic License Plate Recognition" Intelligent Transportation Systems*[J], IEEE Transactions, 2004, 5(1): 42-53.

[9] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen. *Morphology-based License Plate Detection from Complex Scenes*[A], Proceedings of the 16 th International Conference on Pattern Recognition[C], 2002, 3(2): 176.

[10] 方凯. 车牌图像识别应用技术研究[D], 河北工业大学, 2007.

[11] 李晨. 车牌识别技术的研究及其在智能交通系统中的应用[D], 西北工业大学, 2006.

[12] 黄新. 汽车牌照自动识别系统中字符的分割和识别[D], 南京航空航天大学, 2002.

[13] 江炜亮. 车牌图像自动定位与识别算法的研究[D], 国防科学技术大学, 2003.

[14] 孙兴征. 车牌识别系统中的牌照定位分割技术研究[D], 重庆大学, 2004.

[15] 李文举, 梁德群, 崔连延等. 一种新的车牌倾斜校正方法[J]. 信息与控制, 2004, 33(2): 231-235.

[16] Xianchao Zhang, Xinyue Liu, He Jiang. *A Hybrid Approach to License Plate Segmentation under Complex Conditions*[J], Proceedings of the Third International Conference on Natural Computation, 2007, 68-73.

[17] Cemil Oz and Fikret Ercal. *A Practical License Plate Recognition System for Real-Time Environments*[J], Lecture Notes in Computer Science, chapter Computational Intelligence and Bioinspired Systems, 2005, 3512: 881-888.

[18] 郝永杰, 刘文耀, 路烁. 畸变汽车牌照图像的空间校正[J]. 西南交通大学学报. 2002, 37(4):106-109.

[19] Wen C Y, Yu C C. Hun Z D. *A 3-D transformation to improve the legibility of license plate numbers*[J]. Journal of Forensic Sciences, 2002, 47(3): 578-585.

[20] 王爱玲, 叶明生, 邓秋香. MATLAB R2007 图像处理技术与应用[M], 北京：电子工业出版社, 2008, 195-204.

[21] Wing W. Y. Ng, Andres Dorado, Daniel S. Yeung, Witold Pedrycz, Ebroul Izquierdo. *Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error*[J], Pattern Recognition, 2007, 40(1): 19-32.

[22] 周亮. 基于神经网络的车牌识别算法研究[M], 北京：人民邮电出版社, 2003, 50-53.

[23] Yoshizawa, Shingo, Miyanaga, Yoshikazu. *A High-Speed HMM VLSI Module with Block Parallel Processing*. Electronics and Communications in Japan, Part III: Fundamental Electronic Science. 2004. 87(5): 12-23P.

[24] Dacis K H, Biddulph R, Balashek S. *Automatic Recognition of Spoken Digits*. The Journal of the Acoustical Society of America, 1952, 24(6):637-642.

[25] Fry D B, Denes P. *Theoretical Aspects of Mechanical Speech Recognition. The Design and Operation of the Mechanical Speech Recognizer at University College London*. Journal British Institution of Radio Engineering, 1959, 19(4):211-229.

[26] T.B.Martin, A.L.Nelson, H.J.Zadel. *Speech recognition by feature abstraction techniques*, Tech report AL-TDR-64-176, Air Force Avionics Lab, 1964.

[27] Vintsyuk T K. *Speech Discrimination by Dynamic Programming*. Kibernetika, 1968, 4(1):81-88.

[28] Reddy D R. *Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave*. Technical report, Stanford Univ., 1966.

[29] Sakoe H, Chiba S. *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*. IEEE Trans. On Acoustics, Speech and Signal Processing, 1978, 26(1):43-49.

[30] Makhoul J I. *Linear Prediction: A Tutorial Review*. Proc. IEEE, 1975, 63(4):561-580.

[31] F K Soong, A Rosenberg, L Rabiner, et al. *A Vector Quantization Approach to Speaker Recognition*. ICASSP,1985.

[32] Rabiner L R, Juang B H. *An introduction to hidden Markov models*[J]. ASSP Magazine, IEEE, 1986, 3(1): 4-16.

[33] Lippmann R P. *Review of neural networks for speech recognition*[J]. Neural Computation, 1990, 1(1): 1-38.

[34] Gauvain J L, Lee C. *Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains*. IEEE Trans. on Speech and Audio Processing, 1994, 2(2): 291-298.

[35] Shinoda K, Lee C H. *A Structural Bayes Approach to Speaker Adaption*. IEEE Trans. on Speech and Audio Processing, 2001, 9(3): 276-287.

[36] Varga A P, Moore R K. *Hidden Markov Model Decomposition of Speech and Noise*. Proceeding of ICASSP, 1990, 845-848.

[37] Gales M J F, Young S J. *Parallel Model Combination for Speech Recognition in Noise*. Technical report, University of Cambridge: Department of Engineering, 1993.

[38] Ganapathiraju A, Hamarker J, Picone J. *Application of Support Vector Machines to Speech Recognition*. IEEE Trans on Signal Processing, 2004, 52(8):2348-2355.

[39] S R Quchani, K Rahbar. *Local Orthogonal Discriminate Bases to Hybrid SVM/Self-adaptive HMM Classifier for Discrete Word Speech Recognition*. IEEE International Symposium on Signal Processing and Information Technology, 2006, 6:370-373.

[40] Gas B, Zarader J.L, Chavy C. *Discriminant neural predictive coding applied to phoneme recognition*. Neurocomputing. 2004, 56(4):141-146.

[41] 王炳锡, 屈丹, 彭煊. 实用语音识别基础. 北京：国防工业出版社, 2005, 26-213.

[42] 易克初, 田斌, 付强. 语音信号处理. 北京：国防工业出版社, 2003, 51-233.

[43] 姚天任. 数字语音处理. 武汉：华中科技大学出版社, 1992, 87-356.

[44] Stephen Winters-Hilt, Zuliang Jiang. *A Hidden Markov Model With Binned Duration Algorithm*[J]. IEEE Transactions on Signal Processing, 2010, 58(2): 948-952.

[45] 谢锦辉. 隐马尔可夫模型(HMM)及其在语音处理中的应用. 武汉：华中理工大学出版社，1995.

[46] 杨行峻, 迟惠生. 语音信号数字处理. 北京：电子工业出版社，1995.

[47] 赵力. 语音信号处理(第 2 版). 北京：机械工业出版社, 2009.

[48] 蔡莲红, 黄德智, 蔡锐. 现代语音技术基础与应用. 北京：清华大学出版社, 2003.

[49] 吴朝晖, 杨莹春. 说话人识别模型与方法. 北京：清华大学出版社, 2009.

[50] 张雄伟, 陈亮, 杨吉斌. 现代语音处理技术及应用. 北京：机械工业出版社, 2003.

[51] 韩纪庆, 张磊, 郑铁然. 语音信号处理. 北京：清华大学出版社, 2004.

附录A　鸢尾属植物样本数据(Iris Data)

鸢尾属植物样本数据，也称为 Iris 数据，是模式识别文献中最著名的数据集之一，其创建者是 R. A. Fisher。Fisher 的文章是模式识别领域的经典文献。该数据集共 150 个样本，有 3 个类，每个类有 50 个样本属于一种类型的鸢尾属植物。3 个类分别是山鸢尾、变色鸢尾、维吉尼亚鸢尾，其中山鸢尾与变色鸢尾、维吉尼亚鸢尾是线性可分的，变色鸢尾与维吉尼亚鸢尾是线性不可分的。

山鸢尾(Setosa)					变色鸢尾(Versicolor)					维吉尼亚鸢尾(Verginica)				
序号	花瓣宽	花瓣长	萼片宽	萼片长	序号	花瓣宽	花瓣长	萼片宽	萼片长	序号	花瓣宽	花瓣长	萼片宽	萼片长
1	0.2	1.4	3.5	5.1	51	1.4	4.7	3.2	7	101	2.5	6	3.3	6.3
2	0.2	1.4	3	4.9	52	1.5	4.5	3.2	6.4	102	1.9	5.1	2.7	5.8
3	0.2	1.3	3.2	4.7	53	1.5	4.9	3.1	6.9	103	2.1	5.9	3	7.1
4	0.2	1.5	3.1	4.6	54	1.3	4	2.3	5.5	104	1.8	5.6	2.9	6.3
5	0.2	1.4	3.6	5	55	1.5	4.6	2.8	6.5	105	2.2	5.8	3	6.5
6	0.4	1.7	3.9	5.4	56	1.3	4.5	2.8	5.7	106	2.1	6.6	3	7.6
7	0.3	1.4	3.4	4.6	57	1.6	4.7	3.3	6.3	107	1.7	4.5	2.5	4.9
8	0.2	1.5	3.4	5	58	1	3.3	2.4	4.9	108	1.8	6.3	2.9	7.3
9	0.2	1.4	2.9	4.4	59	1.3	4.6	2.9	6.6	109	1.8	5.8	2.5	6.7
10	0.1	1.5	3.1	4.9	60	1.4	3.9	2.7	5.2	110	2.5	6.1	3.6	7.2
11	0.2	1.5	3.7	5.4	61	1	3.5	2	5	111	2	5.1	3.2	6.5
12	0.2	1.6	3.4	4.8	62	1.5	4.2	3	5.9	112	1.9	5.3	2.7	6.4
13	0.1	1.4	3	4.8	63	1	4	2.2	6	113	2.1	5.5	3	6.8
14	0.1	1.1	3	4.3	64	1.4	4.7	2.9	6.1	114	2	5	2.5	5.7
15	0.2	1.2	4	5.8	65	1.3	3.6	2.9	5.6	115	2.4	5.1	2.8	5.8
16	0.4	1.5	4.4	5.7	66	1.4	4.4	3.1	6.7	116	2.3	5.3	3.2	6.4
17	0.4	1.3	3.9	5.4	67	1.5	4.5	3	5.6	117	1.8	5.5	3	6.5
18	0.3	1.4	3.5	5.1	68	1	4.1	2.7	5.8	118	2.2	6.7	3.8	7.7
19	0.3	1.7	3.8	5.7	69	1.5	4.5	2.2	6.2	119	2.3	6.9	2.6	7.7
20	0.3	1.5	3.8	5.1	70	1.1	3.9	2.5	5.6	120	1.5	5	2.2	6
21	0.2	1.7	3.4	5.4	71	1.8	4.8	3.2	5.9	121	2.3	5.7	3.2	6.9
22	0.4	1.5	3.7	5.1	72	1.3	4	2.8	6.1	122	2	4.9	2.8	5.6
23	0.2	1	3.6	4.6	73	1.5	4.9	2.5	6.3	123	2	6.7	2.8	7.7
24	0.5	1.7	3.3	5.1	74	1.2	4.7	2.8	6.1	124	1.8	4.9	2.7	6.3
25	0.2	1.9	3.4	4.8	75	1.3	4.3	2.9	6.4	125	2.1	5.7	3.3	6.7
26	0.2	1.6	3	5	76	1.4	4.4	3	6.6	126	1.8	6	3.2	7.2
27	0.4	1.6	3.4	5	77	1.4	4.8	2.8	6.8	127	1.8	4.8	2.8	6.2
28	0.2	1.5	3.5	5.2	78	1.7	5	3	6.7	128	1.8	4.9	3	6.1

（续表）

山鸢尾（Setosa）					变色鸢尾（Versicolor）					维吉尼亚鸢尾（Verginica）				
序号	花瓣宽	花瓣长	萼片宽	萼片长	序号	花瓣宽	花瓣长	萼片宽	萼片长	序号	花瓣宽	花瓣长	萼片宽	萼片长
29	0.2	1.4	3.4	5.2	79	1.5	4.5	2.9	6	129	2.1	5.6	2.8	6.4
30	0.2	1.6	3.2	4.7	80	1	3.5	2.6	5.7	130	1.6	5.8	3	7.2
31	0.2	1.6	3.1	4.8	81	1.1	3.8	2.4	5.5	131	1.9	6.1	2.8	7.4
32	0.4	1.5	3.4	5.4	82	1	3.7	2.4	5.5	132	2	6.4	3.8	7.9
33	0.1	1.5	4.1	5.2	83	1.2	3.9	2.7	5.8	133	2.2	5.6	2.8	6.4
34	0.2	1.4	4.2	5.5	84	1.6	5.1	2.7	6	134	1.5	5.1	2.8	6.3
35	0.2	1.5	3.1	4.9	85	1.5	4.5	3	5.4	135	1.4	5.6	2.6	6.1
36	0.2	1.2	3.2	5	86	1.6	4.5	3.4	6	136	2.3	6.1	3	7.7
37	0.2	1.3	3.5	5.5	87	1.5	4.7	3.1	6.7	137	2.4	5.6	3.4	6.3
38	0.1	1.4	3.6	4.9	88	1.3	4.4	2.3	6.3	138	1.8	5.5	3.1	6.4
39	0.2	1.3	3	4.4	89	1.3	4.1	3	5.6	139	1.8	4.8	3	6
40	0.2	1.5	3.4	5.1	90	1.3	4	2.5	5.5	140	2.1	5.4	3.1	6.9
41	0.3	1.3	3.5	5	91	1.2	4.4	2.6	5.5	141	2.4	5.6	3.1	6.7
42	0.3	1.3	2.3	4.5	92	1.4	4.6	3	6.1	142	2.3	5.1	3.1	6.9
43	0.2	1.3	3.2	4.4	93	1.2	4	2.6	5.8	143	1.9	5.1	2.7	5.8
44	0.6	1.6	3.5	5	94	1	3.3	2.3	5	144	2.3	5.9	3.2	6.8
45	0.4	1.9	3.8	5.1	95	1.3	4.2	2.7	5.6	145	2.5	5.7	3.3	6.7
46	0.3	1.4	3	4.8	96	1.2	4.2	3	5.7	146	2.3	5.2	3	6.7
47	0.2	1.6	3.8	5.1	97	1.3	4.2	2.9	5.7	147	1.9	5	2.5	6.3
48	0.2	1.4	3.2	4.6	98	1.3	4.3	2.9	6.2	148	2	5.2	3	6.5
49	0.2	1.5	3.7	5.3	99	1.1	3	2.5	5.1	149	2.3	5.4	3.4	6.2
50	0.2	1.4	3.3	5	100	1.3	4.1	2.8	5.7	150	1.8	5.1	3	5.9

附录B　习 题 解 答

习题 2

2.1　两类问题的最小错误率贝叶斯决策的内容是什么？

答：如果对待分类模式的特征得到一个观察值 $\boldsymbol{x}$，则利用公式

$$p(\omega_i|\boldsymbol{x})=\frac{p(\boldsymbol{x}|\omega_i)p(\omega_i)}{p(\boldsymbol{x})}=\frac{p(\boldsymbol{x}|\omega_i)p(\omega_i)}{\sum_{i=1}^{2}p(\boldsymbol{x}|\omega_i)p(\omega_i)}$$

可计算出结果：

$$\text{如果 } p(\omega_1|\boldsymbol{x})>p(\omega_2|\boldsymbol{x})\text{，那么 } \boldsymbol{x}\in\omega_1\text{，否则 } \boldsymbol{x}\in\omega_2$$

2.2　两类问题的最小风险贝叶斯决策的内容是什么？

答：对于两类问题最小风险贝叶斯决策，动作 a_1 相当于决策“真正的状态为 ω_1”，而动作 a_2 相当于决策“真正的状态为 ω_2”。$\lambda_{ij}=\lambda(a_1|\omega_j)$ 表示当真正状态为 ω_j 而把 ω_i 误做真正状态时所受的损失。将多类条件风险公式按照两类问题展开可以得到

$$R(a_i|\boldsymbol{x})=E\{\lambda_{ij}\}=\sum_{j=1}^{c}\lambda_{ij}p(\omega_j|\boldsymbol{x})$$

$$R(a_1|\boldsymbol{x})=\lambda_{11}p(\omega_1|\boldsymbol{x})+\lambda_{12}p(\omega_2|\boldsymbol{x})$$

$$R(a_2|\boldsymbol{x})=\lambda_{21}p(\omega_1|\boldsymbol{x})+\lambda_{22}p(\omega_2|\boldsymbol{x})$$

这时，最小风险的贝叶斯决策法则为：如果 $R(a_1|\boldsymbol{x})<R(a_2|\boldsymbol{x})$，那么 $\boldsymbol{x}\in\omega_1$，即判定 ω_1 为真正的状态；否则 $\boldsymbol{x}\in\omega_2$，即判定 ω_2 为真正的状态。

2.3　简述密度函数的参数估计与非参数估计方法的主要差别。

答：概率密度函数的参数估计首先需要假定概率密度函数的函数形式，而具体的概率密度函数由一组参数决定，最后利用已知的训练样本集合估计出具体的分布参数；非参数估计无需对分布的形式做出假定，而是直接利用训练样本集合对概率密度函数做出估计。

2.4　简述最大似然估计与贝叶斯估计的基本思想及主要差别。

答：最大似然估计和贝叶斯估计都属于参数估计方法，即假定了密度函数的形式之后，需要估计分布的参数。最大似然估计将参数视为确定而未知的向量，通过建立似然函数或对数似然函数，然后求解似然函数的最优解，来确定最有可能产生训练样本集合作为参数的最大似然估计。与此不同，贝叶斯估计将未知参数视为一个随机向量，并且其分布形式已知，满足一定的先验概率分布，然后利用训练样本集合估计出参数向量的分布，而在识别时则需要考虑所有可能参数产生待识别样本的平均值。

2.5 设在一维特征空间中两类样本服从正态分布，$\sigma_1=\sigma_2=1$，$\mu_1=0$，$\mu_2=3$，两类先验概率之比为 $p(\omega_1)/p(\omega_2)=\mathrm{e}$。试求基于最小错误率贝叶斯决策原则的决策分界面的 x 值。

答：按基于最小错误率的贝叶斯决策，则分界面上的点服从

$$p(x|\omega_1)p(\omega_1)=p(x|\omega_2)p(\omega_2)$$

所以有

$$p(x|\omega_1)\cdot\mathrm{e}=p(x|\omega_2)$$

$$\exp(-x^2/2)\cdot\mathrm{e}=\exp(-(x-3)^2/2)$$

$$-x^2/2+1=-(x-3)^2/2$$

$$x=11/6$$

2.6 设有两类正态分布的样本集，第一类均值 $\boldsymbol{\mu}_1=(2,0)^{\mathrm{T}}$，方差 $\boldsymbol{\Sigma}_1=\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}$，第二类均值 $\boldsymbol{\mu}_2=(2,2)^{\mathrm{T}}$，方差 $\boldsymbol{\Sigma}_2=\begin{bmatrix}1 & -1/2\\ -1/2 & 1\end{bmatrix}$，先验概率 $p(\omega_1)=p(\omega_2)$。按最小错误率贝叶斯决策求两类的分界面。

答：由于 $|\boldsymbol{\Sigma}_1|=|\boldsymbol{\Sigma}_2|$，故先验概率相等。基于最小错误率的贝叶斯决策规则，在两类决策面分界面上的样本 $\boldsymbol{X}=(x_1,x_2)^{\mathrm{T}}$ 应满足

$$(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_1)=(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_2) \tag{1}$$

对式(1)进行分解有

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{X}-2\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{X}+\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{X}-2\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{X}+\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2$$

得

$$\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\Sigma}_2^{-1})\boldsymbol{X}-2(\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1})\boldsymbol{X}+\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_2=0$$

由已知条件可计算出 $\boldsymbol{\Sigma}_1^{-1}=\begin{bmatrix}4/3 & -2/3\\ -2/3 & 4/3\end{bmatrix}$ 和 $\boldsymbol{\Sigma}_2^{-1}=\begin{bmatrix}4/3 & 2/3\\ 2/3 & 4/3\end{bmatrix}$，将已知条件代入上式并化简计算得

$$x_1x_2-4x_2-x_1+4=0$$

即 $(x_1-4)(x_2-1)=0$，所以分解决策面由两根直线组成，一根为 $x_1=4$，另一根为 $x_2=1$。

2.7 已知某一正态分布二维随机变量的协方差矩阵为 $\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}$，均值向量为零向量。试求其 Mahalanobis 距离为 1 的点的轨迹。

解：$\boldsymbol{x}$ 到 $\boldsymbol{\mu}$ 的 Mahalanobis 距离的平方为 $\gamma^2=(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$，设 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$，Mahalanobis 距离为 1 的点的轨迹满足 $\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}^{-1}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=1$，整理得 $x_1^2-x_1x_2+x_2^2=1$。

2.8 对两类问题，若损失函数 $\lambda_{11}=\lambda_{22}=0,\lambda_{12}\neq 0,\lambda_{21}\neq 0$，试求基于最小风险贝叶斯决策分界面处的两类错误率与 $\lambda_{12},\lambda_{21}$ 的关系。

答：由于在基于最小风险贝叶斯决策分界面处满足 $R(\omega_1 \mid \boldsymbol{X}) = R(\omega_2 \mid \boldsymbol{X})$，且

$$R(\omega_1 \mid \boldsymbol{X}) = \lambda_{11} P(\omega_1 \mid \boldsymbol{X}) + \lambda_{12} P(\omega_2 \mid \boldsymbol{X})$$

$$R(\omega_2 \mid \boldsymbol{X}) = \lambda_{22} P(\omega_2 \mid \boldsymbol{X}) + \lambda_{21} P(\omega_1 \mid \boldsymbol{X})$$

故在分界面处应有

$$\lambda_{12} P(\omega_2 \mid \boldsymbol{X}) = \lambda_{21} P(\omega_1 \mid \boldsymbol{X})$$

而在两类问题中，$P(\mathrm{e})|_{x\in\omega_1} = P(\omega_2 \mid \boldsymbol{X})$，故有 $\dfrac{P(\mathrm{e})|_{x\in\omega_1}}{P(\mathrm{e})|_{x\in\omega_2}} = \dfrac{\lambda_{21}}{\lambda_{12}}$。

2.9 设一个二维空间中的两类样本服从正态分布，其参数分别为 $\boldsymbol{\mu}_1 = (1,0)^{\mathrm{T}}$，$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$，$\boldsymbol{\mu}_2 = (-1,0)^{\mathrm{T}}$，$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$，先验概率 $p(\omega_1) = p(\omega_2)$。试证明其基于最小错误率的贝叶斯决策分界面为圆，并求其方程。

证明：先验概率相等条件下，基于最小错误率贝叶斯决策的分界面上两类条件概率密度函数相等，因此有

$$-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_1) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_1| = -\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_2) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_2|$$

由已知条件可计算出 $\boldsymbol{\Sigma}_1^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$，$\boldsymbol{\Sigma}_2^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} = \dfrac{1}{2}\boldsymbol{\Sigma}_1^{-1}$，代入上式并整理得

$$(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_1) + \ln|\boldsymbol{\Sigma}_1| = \frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_2) + \ln|\boldsymbol{\Sigma}_2|$$

设 $\boldsymbol{X} = (x_1, x_2)^{\mathrm{T}}$，则将已知条件代入得

$$(x_1-1)^2 + x_2^2 = \frac{1}{2}(x_1+1)^2 + \frac{1}{2}x_2^2 + \ln 4$$

化简为 $(x_1-3)^2 + x_2^2 = 8 + 2\ln 4$，它是一个圆的方程。

2.10 将上题推广到一般情况，若 $\boldsymbol{\Sigma}_1 = \sigma^2 \boldsymbol{I}$，$\boldsymbol{\Sigma}_2 = k\boldsymbol{\Sigma}_1$，试判断先验概率相等条件下，基于最小错误率的贝叶斯决策面是否是超球面。

解：先验概率相等条件下，基于最小错误率贝叶斯决策的分界面上两类条件概率密度函数相等，因此有

$$-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_1) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_1| = -\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_2) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_2|$$

代入已知条件，得

$$\sigma^2(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{\mu}_1) + \ln|\boldsymbol{\Sigma}_1| = k\sigma^2(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{\mu}_2) + \ln|\boldsymbol{\Sigma}_2|$$

由已知条件可以计算出 $\boldsymbol{\Sigma}_1^{-1} = \dfrac{1}{\sigma^2}\boldsymbol{I}$，$\boldsymbol{\Sigma}_2^{-1} = \dfrac{1}{k\sigma^2}\boldsymbol{I}$，则

$$-\frac{1}{\sigma^2}(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{I}(\boldsymbol{X}-\boldsymbol{\mu}_1) + \ln|\boldsymbol{\Sigma}_1| = \frac{1}{k\sigma^2}(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{I}(\boldsymbol{X}-\boldsymbol{\mu}_2) + \ln|\boldsymbol{\Sigma}_2|$$

$$(\boldsymbol{X}-\boldsymbol{\mu}_1)^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{\mu}_1)-\frac{1}{k}(\boldsymbol{X}-\boldsymbol{\mu}_2)^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{\mu}_2)=\sigma^2\ln k$$

$$k\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}-k\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\mu}_1-k\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{X}+k\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\mu}_1-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\mu}_2+\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{X}-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\mu}_2=k\sigma^2\ln k$$

$$(k-1)\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\mu}_2-k\boldsymbol{\mu}_1)+(\boldsymbol{\mu}_2^{\mathrm{T}}-k\boldsymbol{\mu}_1^{\mathrm{T}})\boldsymbol{X}+(k\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\mu}_2)=k\sigma^2\ln k$$

$$X^{\mathrm{T}}X+X^{\mathrm{T}}\frac{(\boldsymbol{\mu}_2-k\boldsymbol{\mu}_1)}{(k-1)}+\frac{(\boldsymbol{\mu}_2^{\mathrm{T}}-k\boldsymbol{\mu}_1^{\mathrm{T}})}{(k-1)}X+\frac{(k\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\mu}_2)}{(k-1)}=\frac{k\sigma^2\ln k}{(k-1)}$$

$$\left(X+\frac{(\boldsymbol{\mu}_2-k\boldsymbol{\mu}_1)}{(k-1)}\right)^{\mathrm{T}}\left(X+\frac{(\boldsymbol{\mu}_2-k\boldsymbol{\mu}_1)}{(k-1)}\right)=\frac{k\sigma^2\ln k}{(k-1)}+\frac{k(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^{\mathrm{T}}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)}{(k-1)^2}$$

设 $\frac{k\sigma^2\ln k}{(k-1)}+\frac{k(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^{\mathrm{T}}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)}{(k-1)^2}=\gamma^2$，则

$$\left(\boldsymbol{X}+\frac{(\boldsymbol{\mu}_2-k\boldsymbol{\mu}_1)}{(k-1)}\right)^{\mathrm{T}}\left(\boldsymbol{X}+\frac{(\boldsymbol{\mu}_2-k\boldsymbol{\mu}_1)}{(k-1)}\right)=\gamma^2$$

这是一个超球面方程，所以在 $\boldsymbol{\Sigma}_1=\sigma^2\boldsymbol{I}$，$\boldsymbol{\Sigma}_2=k\boldsymbol{\Sigma}_1$ 且先验概率相等的条件下，基于最小错误率的贝叶斯决策面是超球面。

习题 3

3.1　设五维空间的线性方程为 $55x_1+68x_2+32x_3+16x_4+26x_5+10=0$，试求出其权向量与样本向量点积表达式 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0=0$ 中的 $\boldsymbol{w}$，$\boldsymbol{x}$，以及相应的增广权向量和增广特征向量。

解：样本向量 $\boldsymbol{x}=(x_1,x_2,x_3,x_4,x_5)^{\mathrm{T}}$，权向量 $\boldsymbol{w}=(55,68,32,16,26)^{\mathrm{T}}$，$w_0=10$；增广特征向量 $\boldsymbol{y}=(x_1,x_2,x_3,x_4,x_5,1)^{\mathrm{T}}$；增广权向量 $\boldsymbol{a}=(55,68,32,16,26,10)^{\mathrm{T}}$。

3.2　给出一组三类问题的判别函数如下：

$$g_1(\boldsymbol{x})=-x_1,g_2(\boldsymbol{x})=x_1+x_2-1,g_3(\boldsymbol{x})=x_1-x_2-1$$

① 假设每一模式类与其他模式类之间可用单个判别平面分隔；

② 每两类模式之间都可分别用判别平面分隔开，且

$$g_{12}(\boldsymbol{x})=g_1(\boldsymbol{x})，\ g_{13}(\boldsymbol{x})=g_2(\boldsymbol{x})，\ g_{23}(\boldsymbol{x})=g_3(\boldsymbol{x})$$

对于以上两种情况，分别求出每类的判别边界和区域。

解：① 根据 3.1 节介绍的第一种情况，此时有 $c=3$ 个判别函数，其具有下面的性质：

$$g_i(x)=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}\begin{cases}>0, & \boldsymbol{x}\in\omega_i,i=1,2,\mathrm{L}\ ,c\\ \le 0 & \boldsymbol{x}\in\omega_i\end{cases}$$

三个判别边界分别为 $g_1(\boldsymbol{x})=-x_1=0\to x_2$ 轴，$g_2(\boldsymbol{x})=x_1+x_2-1=0$，$g_3(\boldsymbol{x})=x_1-x_2-1=0$，则判别区域如题图3.2.1 所示。

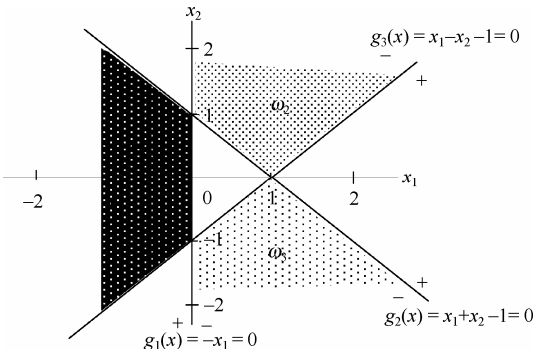
② 由 3.1 节介绍的第二种情况，对 c 类别，有 $\frac{c(c-1)}{2}$ 个判别函数，且 $g_{ij}(\boldsymbol{x})=\boldsymbol{w}_{ij}^{\mathrm{T}}\boldsymbol{x}$ $\begin{cases}>0 & \boldsymbol{x}\in\omega_i\\ <0 & \boldsymbol{x}\in\omega_j\end{cases}$，$g_{ij}(\boldsymbol{x})=g_{ji}(\boldsymbol{x})$。除了关心 $g_{ij}(\boldsymbol{x})$ 的正负之外，还要考虑其他类域的判别函数

才能做出正确的判决。所以这种情况的判别规则是：如果 $g_{ij}(\boldsymbol{x})>0, \forall j\neq i, j=1,2,\mathrm{L}\ ,c$，则 $\boldsymbol{x}\in\omega_i$。此时，

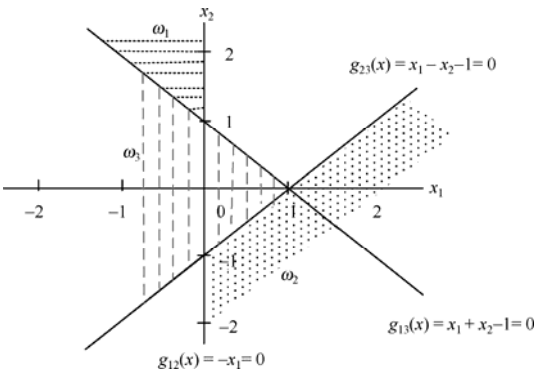
$$g_{12}(\boldsymbol{x})=g_1(\boldsymbol{x})=-x_1, g_{13}(\boldsymbol{x})=g_2(\boldsymbol{x})=x_1+x_2-1, g_{23}(\boldsymbol{x})=g_3(\boldsymbol{x})=x_1-x_2-1$$

$$g_{21}(\boldsymbol{x})=-g_1(\boldsymbol{x})=x_1, g_{31}(\boldsymbol{x})=-g_2(\boldsymbol{x})=-x_1-x_2+1, g_{32}(\boldsymbol{x})=-g_3(\boldsymbol{x})=-x_1+x_2+1$$

其判别区域如图3.2.2所示。



题图 3.2.1



题图 3.2.2

3.3　设两类样本的类内离散矩阵分别为 $\boldsymbol{S}_1=\begin{bmatrix}1 & 1/2\\ 1/2 & 1\end{bmatrix}, \boldsymbol{S}_2=\begin{bmatrix}1 & -1/2\\ -1/2 & 1\end{bmatrix}$，各类样本均值分别为 $\boldsymbol{\mu}_1=(2,0)^{\mathrm{T}}$ 和 $\boldsymbol{\mu}_2=(2,2)^{\mathrm{T}}$，试用 Fisher 准则求其决策面方程。

答：

$$\boldsymbol{S}_w=\boldsymbol{S}_1+\boldsymbol{S}_2=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$$

$$\boldsymbol{w}^*=\boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)=\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}\begin{bmatrix}0\\ -2\end{bmatrix}=\begin{bmatrix}0\\ -1\end{bmatrix}$$

根据 3.6 节介绍的阈值选取方法，取 y_0 为

$$y_0=\frac{\not{p}_1\not{0}+\not{p}_2\not{0}}{2}=\boldsymbol{w}^{*\mathrm{T}}\frac{\boldsymbol{\mu}_1+\boldsymbol{\mu}_2}{2}=\begin{bmatrix}0 & -1\end{bmatrix}\begin{bmatrix}2\\ 1\end{bmatrix}=-1$$

则 fisher 准则最佳决策面方程为 $\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}=y_0$，将求得的数据代入得 $x_2=1$。

习题 4

4.1　已知两个数据集分别为 $\omega_1:(0,0,1),(1,1,1),(1,0,1),(1,0,0)$ 和 $\omega_2:(0,0,0),(1,1,0),(0,1,0),(0,1,1)$。

(1) 将该 8 个数据作为一个数据集对其进行 K-L 变换；

(2) 求这两个数据集的类内离散矩阵，并以此作为其产生矩阵进行 K-L 变换。

答：(1)先求这 8 个点的均值向量，得 $\boldsymbol{\mu}=\frac{1}{2}\begin{bmatrix}1\\1\\1\end{bmatrix}$，再计算协方差矩阵为 $\boldsymbol{\Sigma}=\frac{1}{4}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$。由于它已是一个对角矩阵，且主对角线元素值相等，因此无需进一步做 K-L 变换，原坐标系的基已经是 K-L 变换的基，并且任何一组正交基都可作为其 K-L 变换的基。

(2)分别求两组数据的均值及类内离散矩阵得

$$\boldsymbol{\mu}_1=\frac{1}{4}(3,1,3)^{\mathrm{T}},\ \boldsymbol{\mu}_2=\frac{1}{4}(1,3,1)^{\mathrm{T}}$$

已知 $\boldsymbol{S}_i=\sum_{x\in\omega_i}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}},i=1,2$，代入已知数据，得

$$\boldsymbol{S}_1=\boldsymbol{S}_2=\frac{1}{16}\begin{bmatrix}12&4&-4\\4&12&4\\-4&4&12\end{bmatrix}=\frac{1}{4}\begin{bmatrix}3&-1&1\\1&3&1\\-1&1&3\end{bmatrix},\ \boldsymbol{S}_w=\frac{1}{2}\begin{bmatrix}3&1&-1\\1&3&1\\-1&1&3\end{bmatrix}$$

对 $\boldsymbol{S}_w$ 进行特征值分解 $\begin{vmatrix}\frac{3}{2}-\lambda&\frac{1}{2}&-\frac{1}{2}\\\frac{1}{2}&\frac{3}{2}-\lambda&\frac{1}{2}\\-\frac{1}{2}&\frac{1}{2}&\frac{3}{2}-\lambda\end{vmatrix}=0$，得其特征值为 1/2, 2, 2，对应于 $\lambda=1/2$，特征向量 $\boldsymbol{\varphi}_1=(-0.5774\quad 0.5774\quad -0.5774)^{\mathrm{T}}$，而对应于 $\lambda=2$ 的特征向量没有唯一解。所以取 $\boldsymbol{\varphi}_1=(-0.5774\quad 0.5774\quad -0.5774)^{\mathrm{T}}$ 作为变换矩阵 $\boldsymbol{\Phi}$，由 $\boldsymbol{y}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}$ 将原样本变换为新的样本：$-0.5774,\ -0.5774,\ -0.5774*2,\ -0.5774, 0, 0, 0.5774, 0$。

4.2 已知一组数据的协方差矩阵为 $\begin{bmatrix}1&1/2\\1/2&1\end{bmatrix}$，试问

(1)协方差矩阵中各元素的含义是什么？

(2)K-L 变换的最佳准则是什么？

(3)为什么说经 K-L 变换后，消除了各分量之间的相关性？

答：协方差矩阵为 $\begin{bmatrix}1&1/2\\1/2&1\end{bmatrix}$，则

(1)对角元素是各分量的方差，非对角元素是各分量之间的协方差。

(2)K-L 变换的最佳准则为：对一组数据按一组正交基分解，在只取相同数量分量的条件下，以均方误差计算截尾误差最小。

(3)在经 K-L 变换后，协方差矩阵成为对角矩阵，因而各主分量间的相关消除。

4.3 求如下序列的离散傅里叶变换。

(1) $x_1(n)=\delta(n-3)$

(2) $x_2(n)=\frac{1}{2}\delta(n+1)+\delta(n)+\frac{1}{2}\delta(n-1)$

(3) $x_3(n)=\begin{cases}\left(\dfrac{1}{2}\right)^n & n=0,2,4,\mathrm{L}\\ 0 & \text{其他}\end{cases}$

解：

(1) $X_1(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=-\infty}^{\infty}\delta(n-3)\mathrm{e}^{-\mathrm{j}\omega n}=\mathrm{e}^{-3\mathrm{j}\omega}$

(2) $X_2(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=-\infty}^{\infty}x_2(n)\mathrm{e}^{-\mathrm{j}\omega n}=\frac{1}{2}\mathrm{e}^{\mathrm{j}\omega}+1+\frac{1}{2}\mathrm{e}^{-\mathrm{j}\omega}=1+\frac{1}{2}(\mathrm{e}^{\mathrm{j}\omega}+\mathrm{e}^{-\mathrm{j}\omega})=1+\cos(\omega)$

(3) $X_3(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=-\infty}^{\infty}x_3(n)\mathrm{e}^{-\mathrm{j}n\omega}=\sum_{n=0,2,4,\mathrm{L}}\left(\frac{1}{2}\right)^n\mathrm{e}^{-\mathrm{j}n\omega}$

所以

$$X_3(\mathrm{e}^{\mathrm{j}\omega})=\sum_{n=0}^{\infty}\left(\frac{1}{2}\right)^{2n}\mathrm{e}^{-2\mathrm{j}n\omega}=\sum_{n=0}^{\infty}\left(\frac{1}{4}\ \mathrm{e}^{-2\mathrm{j}\omega}\right)^n=\frac{1}{1-\frac{1}{4}\mathrm{e}^{-\mathrm{j}2\omega}}$$

4.4　离散傅里叶变换的性质及在图像处理中的应用是什么？

答：离散傅里叶变换的性质：分离性、平移性、周期性、共轭对称性、旋转不变性、分配性和比例性。

离散傅里叶变换在图像处理中的应用有：它是图像处理中的一个最基本的数学工具，利用这个工具可以对图像进行频谱分析，进行滤波、降噪等处理，例如可以用低通滤波器滤掉图像中的高频噪声等。

4.5　小波变换有哪些特点？

答：小波变换的特点：能量集中；易于控制各子带噪声；具有与人视觉系统相吻合的对数特征。例如在图像压缩中，由于能量集中，所以压缩比高，且在传输中抗干扰能力强。

4.6　取一幅实际的图像，对其进行傅里叶变换，观察变换后的图像，从图像上来分析两者之间的关系。

答：MATLAB 程序如下。

```
clc;
clear all
f = imread('Fig8.23.jpg');
figure, imshow(f);

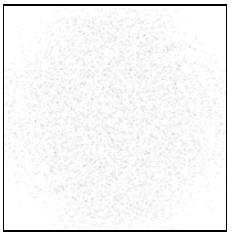
g = im2double(f);
R = fft2(g);
figure, imshow(abs(R));

S = fftshift(fft2(f));
figure, imshow(log(abs(S)), []);
```

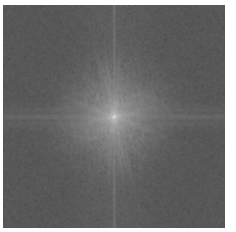
在图中可以看到图像的低频能量都集中在中心部分，而高频能量集中在四周，这样就便于以后对图像频谱进行各种处理。



(a) 原始图像



(b) 傅里叶变换谱



(c) 傅里叶变换中心谱

习题 5

5.1 简述监督分类方法和无监督分类方法的区别。

答：监督分类方法和无监督分类方法的区别主要如下。

(1) 监督分类方法有训练样本集，在训练样本集中给出不同类别的训练样本，用这些训练样本可以找出区分不同类样本的方法，从而在特征空间中划定决策域。

(2) 监督分类方法由训练阶段和测试阶段组成。训练阶段利用训练集中的训练样本进行分类器设计，确定分类器参数；测试阶段将待识别样本输入，根据分类的决策规则，确定待识别样本的所属类别。

(3) 无监督分类方法可用来分析数据的内在规律，它没有训练样本，如聚类分析、主分量分析、数据拟合等方法都是无监督分类方法。

5.2 证明欧几里得距离满足三角不等式。

证明：

$$\begin{aligned} d_2(x,y)+d_2(y,z) &= \left(\sum_{i=1}^{1}|x_i-y_i|^2\right)^{1/2}+\left(\sum_{i=1}^{1}|y_i-z_i|^2\right)^{1/2} \\ &\geq \left(\sum_{i=1}^{1}|x_i-y_i+y_i-z_i|^2\right)^{1/2} \\ &= d_2(x,y) \end{aligned}$$

5.3 证明总离散度矩阵等于类内离散度矩阵与类间离散度矩阵之和，即 $\boldsymbol{S}_T=\boldsymbol{S}_W+\boldsymbol{S}_B$。

证明：根据定义可知

$$\text{类内离散度矩阵 } \boldsymbol{S}_W=\sum_{i=1}^{K}\boldsymbol{S}_i\text{，其中 } \boldsymbol{S}_i=\sum_{\boldsymbol{x}\in C_i}(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^{\mathrm{T}}$$

$$\text{类间离散度矩阵 } \boldsymbol{S}_B=\sum_{i=1}^{K}n_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}$$

$$\text{总离散度矩阵 } \boldsymbol{S}_T=\sum_{\boldsymbol{x}\in\boldsymbol{X}}(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}$$

其中 K 是聚类类别数目，$\boldsymbol{C}_i$ 是第 i 类聚类中心域的样本集合，n_i 是 $\boldsymbol{C}_i$ 中的样本数；$\boldsymbol{m}_i$ 是第 i 类的样本均值向量 $\boldsymbol{m}_i=\dfrac{1}{n_i}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\boldsymbol{x}$；总平均向量 $\boldsymbol{m}=\dfrac{1}{N}\sum_{i=1}^{K}n_i\boldsymbol{m}_i$。

可以证明

$$\begin{aligned}&\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m}_i)^{\mathrm{T}}\\&=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{m}_i\boldsymbol{x}^{\mathrm{T}}-\boldsymbol{m}\boldsymbol{x}^{\mathrm{T}}-\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}}+\boldsymbol{m}\boldsymbol{m}_i^{\mathrm{T}})\\&=\sum_{i=1}^{K}(\boldsymbol{m}_i n_i\boldsymbol{m}_i^{\mathrm{T}}-\boldsymbol{m}n_i\boldsymbol{m}_i^{\mathrm{T}}-n_i\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}}+n_i\boldsymbol{m}\boldsymbol{m}_i^{\mathrm{T}})\\&=0\end{aligned}$$

同理可得

$$\begin{aligned}&\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}\\&=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{x}\boldsymbol{m}_i^{\mathrm{T}}-\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}}-\boldsymbol{x}\boldsymbol{m}^{\mathrm{T}}+\boldsymbol{m}_i\boldsymbol{m}^{\mathrm{T}})\\&=\sum_{i=1}^{K}(n_i\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}}-n_i\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}}-n_i\boldsymbol{m}_i\boldsymbol{m}^{\mathrm{T}}+n_i\boldsymbol{m}_i\boldsymbol{m}^{\mathrm{T}})\\&=0\end{aligned}$$

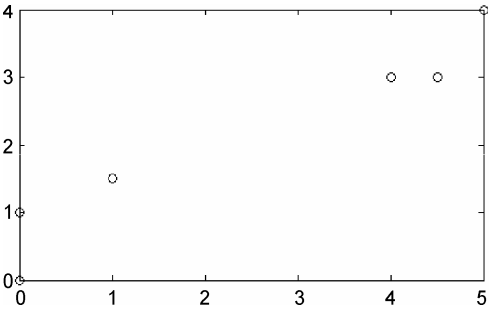
$$\begin{aligned}\boldsymbol{S}_T&=\sum_{\boldsymbol{x}\in\boldsymbol{X}}(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\\&=\sum_{i=1}^{K}\Big(\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\Big)\\&=\sum_{i=1}^{K}\left(\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{x}-\boldsymbol{m}_i+\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m}_i+\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}\right)\\&=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}[(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^{\mathrm{T}}+(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m}_i)^{\mathrm{T}}+(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}+(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}]\\&=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}+\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}=\boldsymbol{S}_W+\sum_{i=1}^{K}n_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}\\&=\boldsymbol{S}_{\mathrm{w}}+\boldsymbol{S}_{\mathrm{B}}\end{aligned}$$

5.4　已知有 6 个二维样本 $\boldsymbol{X}=\{\boldsymbol{x}_i,i=1,2,\mathrm{L}\ ,6\}$，其中 $\boldsymbol{x}_1=[0,0]^{\mathrm{T}}$，$\boldsymbol{x}_2=[0,1]^{\mathrm{T}}$，$\boldsymbol{x}_3=[1,1.5]^{\mathrm{T}}$，$\boldsymbol{x}_4=[4,3]^{\mathrm{T}}$，$\boldsymbol{x}_5=[4.5,3]^{\mathrm{T}}$，$\boldsymbol{x}_6=[5,4]^{\mathrm{T}}$。试按照最短距离进行层次聚类。

解：已知的 6 个二维样本如题图5.4.1 所示。

用最短距离进行层次聚类的 MATLAB 源程序如下：

```
x = [0 0;0 1;1 1.5;4 3;4.5 3;5 4]
y = pdist(x);          %计算 x 的欧氏距离，可以改为其他距离，查参数文档
z = zf_linkage(y)      %聚类，改变参数可以选择聚类方法
h = dendrogram(z)      %画出聚类树图
```



题图 5.4.1

(1)初始将每个样本视为一类，得

$$G_1^{(0)}=\{\boldsymbol{x}_1\},\ G_2^{(0)}=\{\boldsymbol{x}_2\},G_3^{(0)}=\{\boldsymbol{x}_3\},G_4^{(0)}=\{\boldsymbol{x}_4\},G_5^{(0)}=\{\boldsymbol{x}_5\},G_6^{(0)}=\{\boldsymbol{x}_6\}$$

计算各类间的欧氏距离，得到距离矩阵 $\boldsymbol{D}^{(0)}$：

$\boldsymbol{D}^{(0)}$	$G_1{}^{(0)}$	$G_2{}^{(0)}$	$G_3{}^{(0)}$	$G_4{}^{(0)}$	$G_5{}^{(0)}$	$G_6{}^{(0)}$
$G_1{}^{(0)}$	0	1.0000	1.8028	5.0000	5.4083	6.4031
$G_2{}^{(0)}$	1.0000	0	1.1180	4.4721	4.9244	5.8310
$G_3{}^{(0)}$	1.8028	1.1180	0	3.3541	3.8079	4.7170
$G_4{}^{(0)}$	5.0000	4.4721	3.3541	0	0.5000	1.4142
$G_5{}^{(0)}$	5.4083	4.9244	3.8079	0.5000	0	1.1180
$G_6{}^{(0)}$	6.4031	5.8310	4.7170	1.4142	1.1180	0

(2)将最短距离 0.5 对应的类 $G_4{}^{(0)}$和 $G_5{}^{(0)}$合并为一类，得到新的分类

$$G_{45}{}^{(1)}=\{G_4{}^{(0)},G_5{}^{(0)}\}$$

$$G_2{}^{(1)}=\{G_2{}^{(0)}\},G_3{}^{(1)}=\{G_3{}^{(0)}\},G_4{}^{(1)}=\{G_4{}^{(0)}\},G_5{}^{(1)}=\{G_4{}^{(0)}\},G_6{}^{(1)}=\{G_6{}^{(0)}\}$$

计算各类间的欧氏距离，得到距离矩阵 $\boldsymbol{D}^{(1)}$：

$\boldsymbol{D}^{(1)}$	$G_1{}^{(1)}$	$G_2{}^{(1)}$	$G_3{}^{(1)}$	$G_{45}{}^{(1)}$	$G_6{}^{(1)}$
$G_1{}^{(1)}$	0	1.0000	1.8028	5.0000	6.4031
$G_2{}^{(1)}$	1.0000	0	1.1180	4.4721	5.8310
$G_3{}^{(1)}$	1.8028	1.1180	0	3.3541	4.7170
$G_{45}{}^{(1)}$	5.0000	4.4721	3.3541	0	1.1180
$G_6{}^{(1)}$	6.4031	5.8310	4.7170	1.1180	0

(3)将最短距离 1.0 对应的类 $G_1{}^{(1)}$和 $G_2{}^{(1)}$合并为一类，得到距离矩阵 $\boldsymbol{D}^{(2)}$：

$\boldsymbol{D}^{(2)}$	$G_{12}{}^{(2)}$	$G_3{}^{(2)}$	$G_{45}{}^{(2)}$	$G_6{}^{(2)}$
$G_{12}{}^{(2)}$	0	1.1180	4.4721	5.8310
$G_3{}^{(2)}$	1.1180	0	3.3541	4.7170
$G_{45}{}^{(2)}$	4.4721	3.3541	0	1.1180
$G_6{}^{(2)}$	5.8310	4.7170	1.1180	0

(4)将最短距离 1.1180 对应的类 $G_{12}{}^{(2)}$和 $G_3{}^{(2)}$合并为一类，得到距离矩阵 $\boldsymbol{D}^{(3)}$：

$\boldsymbol{D}^{(3)}$	$G_{123}^{(3)}$	$G_{45}^{(3)}$	$G_6^{(3)}$
$G_{123}^{(3)}$	0	3.3541	4.7170
$G_{45}^{(3)}$	3.3541	0	1.1180
$G_6^{(3)}$	4.7170	1.1180	0

(5)将最短距离 1.1180 对应的类 $G_{45}^{(3)}$和 $G_6^{(3)}$合并为一类，得到距离矩阵 $\boldsymbol{D}^{(4)}$：

$\boldsymbol{D}^{(4)}$	$G_{123}^{(4)}$	$G_{456}^{(4)}$
$G_{123}^{(4)}$	0	3.3541
$G_{456}^{(4)}$	3.3541	0

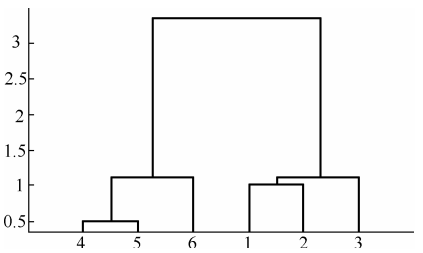
设定一个距离阈值为 $D_T=2$，$\boldsymbol{D}^{(4)}$中的最小元素为 3.3541，超过给定阈值，则聚类结束。结果为

$$G_1=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3\},\ G_2=\{\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6\}$$

如果无阈值条件，继续聚类，最终全部样本归为一类。

上述层次聚类过程可用题图5.4.2的分类树表示，右边的数据为类间的最短距离。

5.5　已知 13 个二维样本为 $\boldsymbol{X}=\{\boldsymbol{x}_i, i=1,2,\mathrm{L}\ ,13\}$，其中 $\boldsymbol{x}_1=[0,0]^{\mathrm{T}}$，$\boldsymbol{x}_2=[0,1]^{\mathrm{T}}$，$\boldsymbol{x}_3=[1,0]^{\mathrm{T}}$，$\boldsymbol{x}_4=[0.5,4]^{\mathrm{T}}$，$\boldsymbol{x}_5=\ [1,3]^{\mathrm{T}}$，$\boldsymbol{x}_6=[1,5]^{\mathrm{T}}$，$\boldsymbol{x}_7=[1.5,4.5]^{\mathrm{T}}$，$\boldsymbol{x}_8=[6,4]^{\mathrm{T}}$，$\boldsymbol{x}_9=[6.5,5]^{\mathrm{T}}$，$\boldsymbol{x}_{10}=[7,4]^{\mathrm{T}}$，$\boldsymbol{x}_{11}=[7.5,7]^{\mathrm{T}}$，$\boldsymbol{x}_{12}=[8,6]^{\mathrm{T}}$，$\boldsymbol{x}_{13}=[8,7]^{\mathrm{T}}$。用 K 均值算法进行分类，分别取 $K=2$ 和 $K=4$。



题图 5.4.2

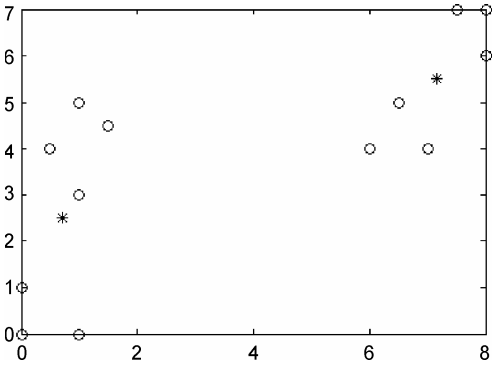
解：通过调用 MATLAB 的 K 均值函数 kmeans，对 X 样本进行分类，其源程序如下：

```
x = [0,0;0,1;1,0;0.5,4;1,3;1,5;1.5,4.5;6,4;6.5,5;7,4;7.5,7;8,6;8,7];
[idx2, c2] = kmeans(x, 2);
plot(x(:,1), x(:,2),'bo', c2(:,1), c2(:,2),'r*');
c2
idx2'
[idx4,c4] = kmeans(x,4);
figure(2);
plot(x(:,1), x(:,2),'bo',c4(:,1),c4(:,2),'r*');
c4
idx4'
```

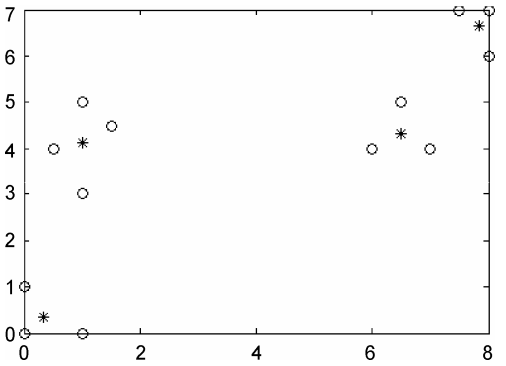
当 $K=2$ 时，分类结果是 $\boldsymbol{x}_1$~$\boldsymbol{x}_7$ 为第一类，$\boldsymbol{x}_8$~$\boldsymbol{x}_{13}$ 为第二类，如题图 5.5.1 所示；聚类中心分别是[0.7143, 2.5000], [7.1667, 5.5000]。

当 $K=4$ 时，分类结果是 $\boldsymbol{x}_1$~$\boldsymbol{x}_3$ 为第四类，$\boldsymbol{x}_4$~$\boldsymbol{x}_7$ 为第二类，$\boldsymbol{x}_8$~$\boldsymbol{x}_{10}$ 为第三类，$\boldsymbol{x}_{11}$~$\boldsymbol{x}_{13}$ 为第一类，如题图5.5.2所示；聚类中心分别是[7.8333, 6.6667], [1.0000, 4.1250], [6.5000, 4.3333], [0.3333, 0.3333]。

5.6　用 K 均值算法对鸢尾属植物(Iris)样本数据进行分类，取 $K=3$。



题图 5.5.1



题图 5.5.2

解：使用附录 A 提供的鸢尾属植物(Iris)样本数据，将其保存为 Iris.xls 文件，其中表单 Sheet1 更名为的 data，具体如下：

Petal_width	Petal_length	Sepal_width	Sepal_length
0.2	1.4	3.5	5.1
0.2	1.4	3	4.9
L	L	L	L
1.8	5.1	3	5.9

在表单 data 中，第 1 行为文本信息，它是对第 2 行到第 151 行数据信息的说明。

用 K 均值算法对鸢尾属植物(Iris)样本数据进行分类的源程序如下：

```
[x,headertext] = xlsread('iris.xls','data');    %读取 iris.xls 中表单为
                                                data 的数据，存入 x 数组
[idx3,c] = kmeans(x, 3, 'display', 'iter');     %调用 MATLAB 的 K 均值函数
                                                kmeans，
xlswrite('iris_kmeans_result.xls', idx3, 'sheet1');  %结果保存在 iris_
                                                kmeans_result.xls 的 sheet1
xlswrite('iris_kmeans_result.xls', c, 'sheet2');  %聚类中心保存在 sheet2
```

当 $K=3$ 时，分类结果是 $\boldsymbol{x}_1$~$\boldsymbol{x}_{50}$ 为第三类；$\boldsymbol{x}_{51}$~$\boldsymbol{x}_{100}$ 为第一类，其中 $\boldsymbol{x}_{53}$ 和 $\boldsymbol{x}_{78}$ 被错分为第二类；$\boldsymbol{x}_{100}$~$\boldsymbol{x}_{150}$ 为第二类，其中 $\boldsymbol{x}_{102}$, $\boldsymbol{x}_{107}$, $\boldsymbol{x}_{114}$, $\boldsymbol{x}_{115}$, $\boldsymbol{x}_{120}$, $\boldsymbol{x}_{122}$, $\boldsymbol{x}_{124}$, $\boldsymbol{x}_{127}$, $\boldsymbol{x}_{128}$, $\boldsymbol{x}_{134}$, $\boldsymbol{x}_{139}$, $\boldsymbol{x}_{143}$, $\boldsymbol{x}_{147}$, $\boldsymbol{x}_{150}$ 共 14 个被错分为第一类；聚类中心是

第一类	1.433871	4.393548	2.748387	5.901613
第二类	2.071053	5.742105	3.073684	6.85
第三类	0.246	1.462	3.428	5.006

5.7 对习题 5.5 中的样本集 X，用 ISODATA 算法进行聚类分析。

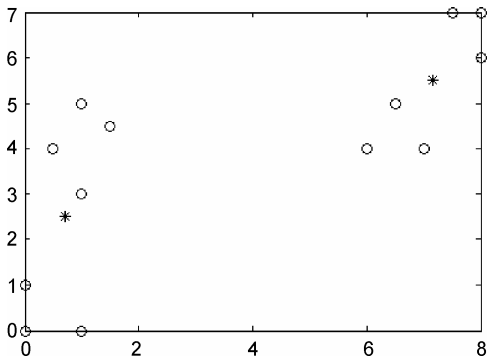
解：用 ISODATA 算法，对 13 个样本集 X 进行分类，其源程序如下：

```
x = [0, 0;0, 1;1, 0;0.5, 4;1, 3;1, 5;1.5, 4.5;6, 4;6.5, 5;7, 4;7.5, 7;8,
6;8, 7];                    %样本集
leastNumber = 1;            %最小聚类中的样本数，少于此数将不能作为独立的样本
stdvar = 0.7; %样本距离分布的标准差
```

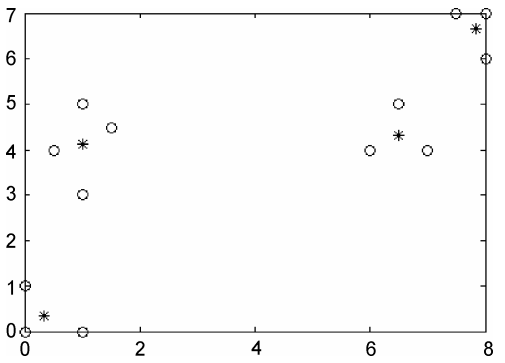
```
minmumDistance = 1;      %两聚类中心的最小距离，小于此数将两个聚类合并
maximumCluster = 2;      %一次运算中可以合并的聚类中心的最大对数
opCount = 80;            %迭代运算次数
n = 13;                  %样本集个数
number = 2;              %样本集的数据维数
clusterNumber = 2;       %聚类中心数
figure(1)
[s, c2, clusterCount] = ISODATA(x, n, number, clusterNumber, leastNumber,
stdvar, minmumDistance, maximumCluster, opCount)
plot(x(:, 1), x(:, 2), 'bo', c2(:, 1), c2(:, 2), 'r*');
figure(2)
stdvar = 0.2;            %样本距离分布的标准差
clusterNumber = 4;       %聚类中心数
[s, c4, clusterCount] = ISODATA(x, n, number, clusterNumber, leastNumber,
stdvar, minmumDistance, maximumCluster, opCount)
plot(x(:, 1), x(:, 2), 'bo', c4(:, 1), c4(:, 2), 'r*');
```

当 $K=2$ 时，分类结果是 $\boldsymbol{x}_1$~$\boldsymbol{x}_7$ 为第一类，$\boldsymbol{x}_8$~$\boldsymbol{x}_{13}$ 为第二类，如题图5.7.1所示；聚类中心分别是[0.7143, 2.5000]和[7.1667, 5.5000]。

当 $K=4$ 时，分类结果是 $\boldsymbol{x}_1$~$\boldsymbol{x}_3$ 为第三类，$\boldsymbol{x}_4$~$\boldsymbol{x}_7$ 为第一类，$\boldsymbol{x}_8$~$\boldsymbol{x}_{10}$ 为第二类，$\boldsymbol{x}_{11}$~$\boldsymbol{x}_{13}$ 为第四类，如题图5.7.2所示；聚类中心分别是[1.0000, 4.1250], [6.5000, 4.3333], [0.3333, 0.3333], [7.8333, 6.6667]。



题图 5.7.1



题图 5.7.2

5.8　用ISODATA 算法对鸢尾属植物(Iris)样本数据进行聚类分析，并与习题 5.6 的结果进行对比分析。

解：用 ISODATA 算法对鸢尾属植物(Iris)样本数据进行分类的源程序如下：

```
[x, headertext] = xlsread('iris.xls', 'data');
clusterNumber = 3;       %预期聚类中心数目
leastNumber = 1;         %最小聚类中的样本数，少于此数将不能作为独立的样本
stdvar = 0.35;           %样本距离分布的标准差
minmumDistance = 0.5;    %两聚类中心的最小距离，小于此数将两个聚类合并
opCount = 30;            %迭代运算次数
n = 150;                 %样本个数
```

```
number = 4;                %样本维数
maximumCluster = 2;        %一次运算中可以合并的聚类中心的最大对数
[s, clusterCenter, clusterCount] = ISODATA(x, n, number, clusterNumber,
leastNumber, stdvar, minmumDistance, maximumCluster, opCount)
xlswrite('iris_isodata_result.xls', s', 'sheet1');
xlswrite('iris_isodata_result.xls', clusterCenter, 'sheet2');
```

当 $K=3$ 时，分类结果是 $\boldsymbol{x}_1$~$\boldsymbol{x}_{50}$ 为第一类；$\boldsymbol{x}_{51}$~$\boldsymbol{x}_{100}$ 为第二类，其中 $\boldsymbol{x}_{53}$ 和 $\boldsymbol{x}_{78}$ 被错分为第三类；$\boldsymbol{x}_{100}$~$\boldsymbol{x}_{150}$ 为第三类，其中 $\boldsymbol{x}_{102}$, $\boldsymbol{x}_{107}$, $\boldsymbol{x}_{114}$, $\boldsymbol{x}_{115}$, $\boldsymbol{x}_{120}$, $\boldsymbol{x}_{122}$, $\boldsymbol{x}_{124}$, $\boldsymbol{x}_{127}$, $\boldsymbol{x}_{128}$, $\boldsymbol{x}_{134}$, $\boldsymbol{x}_{139}$, $\boldsymbol{x}_{143}$, $\boldsymbol{x}_{147}$, $\boldsymbol{x}_{150}$ 共 14 个被错分为第二类；聚类中心是

第一类	0.246	1.462	3.428	5.006
第二类	1.433871	4.393548	2.748387	5.901613
第三类	2.071053	5.742105	3.073684	6.85

对鸢尾属植物(Iris)样本数据进行聚类分析，用 K 均值算法和 ISODATA 算法得到的结果完全一致。

习题 6

6.1 人工神经元网络也可用来进行模式识别，它与统计模式识别原理上是否相同？

解：人工神经元网络中使用的数据也是向量形式，它实现模式识别的原理也是在特征空间中划分决策域，因此在原理上可以属于统计模式识别，但是在方法上与传统的统计模式识别有明显的不同。

6.2 人工神经元网络用做模式识别所能做的，是否用传统模式识别方法都可以做？

解：从实现分类这一功能看，多层感知器与传统模式识别方法中的分段线性分类器的原理是一样的，只是用网络形式表示后，可以运用人工神经元网络一整套分析与研究的方法。由于人工神经元网络的多种模型使人工神经元网络有广泛的应用，因而有一套系统的研究与分析方法。使用人工神经元网络有其优点。

6.3 证明：如果隐单元的激活函数是线性的，那么三层网络等价于二层网络。

解：假设三层线性网络有一个输入向量 $\boldsymbol{x}$、一个隐层单元 $\boldsymbol{y}$ 和一个输出向量 $\boldsymbol{z}$。因为是线性系统，对于矩阵 $\boldsymbol{W}_1$ 和 $\boldsymbol{W}_2$，$\boldsymbol{y}=\boldsymbol{W}_1\boldsymbol{x}$，$\boldsymbol{z}=\boldsymbol{W}_2\boldsymbol{y}$，则输出 $\boldsymbol{z}=\boldsymbol{W}_2\boldsymbol{y}=\boldsymbol{W}_2\boldsymbol{W}_1\boldsymbol{x}=\boldsymbol{W}_3\boldsymbol{x}$，其中 $\boldsymbol{W}_3=\boldsymbol{W}_1\boldsymbol{W}_2$，则和具有连接矩阵 $\boldsymbol{W}_3$ 的两层网络等价。

6.4 利用题 6.3 的结论解释为什么具有线性隐单元的三层网络不能解决某个非线性可分问题，如 XOR 问题。

解：非线性可分问题不能用具有线性隐单元的三层网络求解。假设非线性可分问题可以通过具有隐层单元的三层网络求解，则它也可以通过二层网络求解，很明显这是线性可分问题，但是题中假设是非线性可分的，因此条件和假设是矛盾的，故具有线性隐单元的三层网络不能解决非线性可分问题。

6.5 考虑具有 d 个输入单元、n 个隐单元、c 个输出单元及偏置的一个标准三层 BP 网络。(a)网络中有多少权值？(b)考虑权值对称。证明：如果将每一个权值的符号反向，则网络功能不变。(c)考虑隐单元的对称交换。隐单元上没有标记，因此它们可以相互交换(沿着

对应权值)而使网络功能不受影响。证明该等价标记数——对称交换因子为 $n!2^n$，并在 $n = 10$ 的情况下估计该因子的值。

解：(a)网络权值的数量是 $dn + (n + 1)c$，其中第一项是输入到隐层的权值数量，第二项是隐层到输出的权值，包括一个偏置单元。

(b)输出节点 k 由下式给出：

$$z_k = \sum_{j=1}^{n} f\left(w_{kj} f\left(\sum_{i=1}^{d} w_{ji} x_i + w_{j0} \right) + w_{k0} \right)$$

假设激活函数 $f(\cdot)$ 是奇函数，且为反对称的，给出

$$f\left(\sum_{j=1}^{d} -w_{ji} x_i \right) \leftrightarrow -f\left(\sum_{j=1}^{d} w_{ji} x_i \right)$$

则有

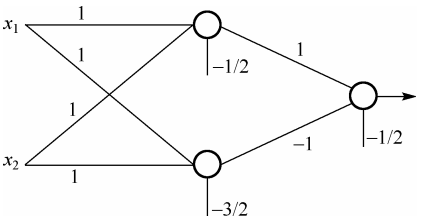
$$(-w_{kj})\left[-f\left(\sum_{j=1}^{d} w_{ji} x_i \right)\right] = (w_{kj}) f\left(\sum_{j=1}^{d} w_{ji} x_i \right)$$

原始输出是不变的。

(c)隐层单元可以沿着对应权值相互交换而使网络功能不受影响。可以构造的集合 n 的子集的数量是 2^n。因为可以构造对于每个子集 n 的不同权值序列，因此总的对称隐层单元数是 $n!2^n$，对于 $n = 10$，其因子数量为 $10! \times 2^{10} = 3\,628\,800 \times 1024 = 3\,715\,891\,200$。

6.6　设计一个 2 层的感知器网络，以实现 A XOR B。

解：由 2 层感知器需要经过两个步骤：由 $g_1(\boldsymbol{x}) = x_1 + x_2 - \frac{1}{2} = 0$ 和 $g_2(\boldsymbol{x}) = x_1 + x_2 - \frac{3}{2} = 0$ 完成输入向量 $\boldsymbol{x}$ 到 $\boldsymbol{y}$ 的映射；由 $g(\boldsymbol{y}) = y_1 + y_2 - \frac{1}{2} = 0$ 实现将两类分开的目的。感知器网络如题图6.6所示。



题图 6.6

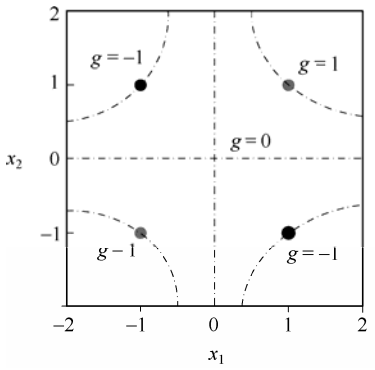
习题 7

7.1　Fisher 准则方法与支持向量机提出的最佳准则是不一致的，它们是否有各自适用的范围？

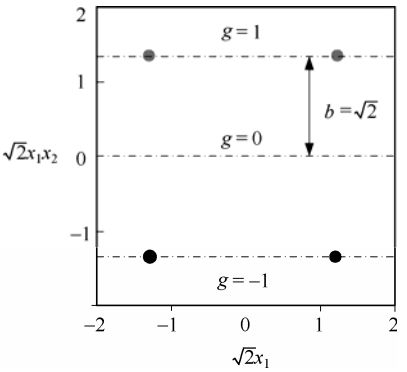
答：对于模式识别，最佳是相对于某一个准则而言的。因此有时不便于对不同的分类器设计方法进行比较。Fisher 准则已有很长历史，也经常得到应用。支持向量机是在对泛化误差研究的基础上提出的，实际中也有好的评价，是近年来比较推崇的一种方法。但对于一个具体问题来说，还不能肯定地说哪一种方法好。

7.2　异或问题(XOR)是最简单的一个无法直接对特征采用线性判别函数来解决的问题。对于空间中的点 $\boldsymbol{x}_1 = (1, 1)^{\mathrm{T}}, \boldsymbol{x}_2 = (-1, -1)^{\mathrm{T}}, \boldsymbol{x}_3 = (1, -1)^{\mathrm{T}}$ 和 $\boldsymbol{x}_4 = (-1, 1)^{\mathrm{T}}$，设计解决 XOR 问题的 SVM。

解：题图7.2(a)为 XOR 问题在原始 x_1x_2 空间，$\boldsymbol{x}_1 = (1, 1)^{\mathrm{T}}, \boldsymbol{x}_2 = (-1, -1)^{\mathrm{T}}$ 属于 ω_1 类，$\boldsymbol{x}_3 = (1, -1)^{\mathrm{T}}$ 和 $\boldsymbol{x}_4 = (-1, 1)^{\mathrm{T}}$ 属于 ω_2 类。在原始二维空间，不可能将其采用线性判别函数分开。



题图 7.2(a)



题图 7.2(b)

通过 SVM 的方法，将上述 4 个点映射到高维空间，在高维空间使其线性可分。有许多映射函数可以完成这一目的，这里采用最简单且展开不超过二次的映射：$1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2$，其中 $\sqrt{2}$ 是为了规范化。在映射空间中，可以找到最佳超平面 $g(x_1, x_2) = x_1x_2 = 0$，且裕量为 $b = \sqrt{2}$。其二维投影空间如题图 7.2(b)所示。通过支持向量的超平面是 $\sqrt{2}x_1x_2 = \pm 1$，它对应于原始特征空间中的双曲线 $x_1x_2 = \pm 1$。

极大化 $L(\alpha) = \sum_{k=1}^{n}\alpha_i - \frac{1}{2}\sum \alpha_k\alpha_j z_k z_j \boldsymbol{y}_j^t \boldsymbol{y}_k$，约束为 $\sum_{k=1}^{n} z_k\alpha_k = 0$，$\alpha_k \geqslant 0$，$k = 1, \text{L}\ , n$，利用对称性取 $\alpha_1 = \alpha_3$，$\alpha_2 = \alpha_4$，利用梯度下降法可得 $\alpha_k^* = \frac{1}{8}$，$k = 1, 2, 3, 4$。4 个样本均为支持向量。

7.3 以习题 7.2 为基础考虑另外 4 个特征，除了上面 4 个特征点之外的其他 $\binom{4}{2} - 1 = 5$ 对特征组合，作出样本和判别函数 $g = \pm 1$ 对应的直线。在所作的图中，这些间隔是否一样？请给出解释。

解：各点的映射为

$$\omega_1 : (1, \sqrt{2}, \sqrt{2}, \sqrt{2}, 1, 1)\ \ (1, -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 1, 1)$$

$$\omega_2 : (1, \sqrt{2}, -\sqrt{2}, -\sqrt{2}, 1, 1)\ \ (1, -\sqrt{2}, \sqrt{2}, -\sqrt{2}, 1, 1)$$

如题图7.3(a)，(b)，(c)，(d)所示。

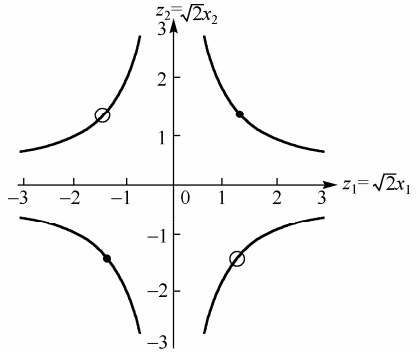
图中的间隔不相同，因为实际的间隔是在 R^6 中的最优超平面，它们投影到低维空间不必保持间隔。

7.4 通过修改感知器算法，写出实现“支持向量机”(SVM)学习算法的伪码程序。对当前最难分的样本的操作，给出详细的数学表达式。解释为什么在训练的后半部，权向量的更新只需用到支持向量。

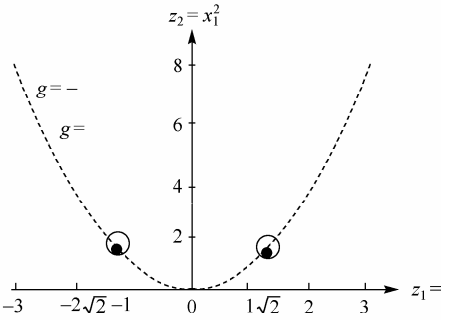
解：支持向量机算法的伪码程序如下。

```
begin initialize a;w1←∞;w2←∞;b←∞
   i←0
     do i←i + 1
     if z1 = -1 and a'yizi<w1, then w1←a'yizi; kw1←k
```

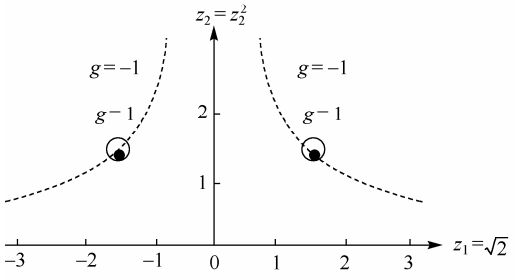
```
        if z1 = 1 and a′yizi<w2, then w2←a′yizi; kw2←k
        until i = n
        a←a + yw2-yw1
        a0←a′(ykw1 + ykw2)/2
        oldb←b;b←a′ykw1/||a||
        until |b-oldb|<ε
    return a0, a
end
```



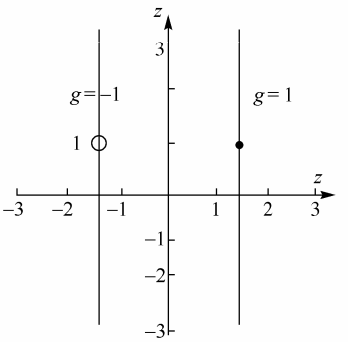
图(a)



图(b)



图(c)



图(d)

题图 7.3

算法从每类中选出错分样本，并修改超平面使其移向错分模式的中心，一旦超平面将类分开，所有的更新将只涉及支持向量。

7.5　只考虑支持向量机和分属两类的训练样本：

$$\omega_1:\ (1,1)^{\mathrm{T}} \qquad (2,2)^{\mathrm{T}} \qquad (2,0)^{\mathrm{T}}$$

$$\omega_2:\ (0,0)^{\mathrm{T}} \qquad (1,0)^{\mathrm{T}} \qquad (0,1)^{\mathrm{T}}$$

在图中作出这 6 个训练点，构造具有最优超平面和最优间隔的权向量，并指出哪些是支持向量。

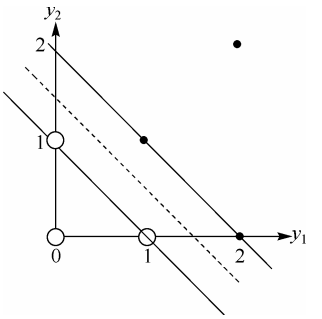
解：考虑用于分类的支持向量。

给出下列两类中的 6 个点且 $z_1=z_2=z_3=-1, z_4=z_5=z_6=+1$：

$$\omega_1:\ \boldsymbol{x}_1=\begin{pmatrix}1\\1\end{pmatrix} \quad \boldsymbol{x}_2=\begin{pmatrix}2\\2\end{pmatrix} \quad \boldsymbol{x}_3=\begin{pmatrix}1\\1\end{pmatrix}$$

$$\omega_2:\quad \boldsymbol{x}_4=\begin{pmatrix}0\\0\end{pmatrix}\qquad \boldsymbol{x}_5=\begin{pmatrix}1\\0\end{pmatrix}\qquad \boldsymbol{x}_6=\begin{pmatrix}0\\1\end{pmatrix}$$

最优超平面方程为 $y_1+y_2=3/2$，或$(3/2\ -1\ -1)^{\mathrm{T}}(1\quad y_1\quad y_2)=0$，将$(3/2\ -1\ -1)$乘以 2，得权向量$(3\ -2\ -2)^{\mathrm{T}}$。最优间隔是模式到最优超平面的最短距离，其值为 $\sqrt{2}/4$。支持向量是间隔上的点，因此支持向量为$\{(1,1)^{\mathrm{T}}(2,0)^{\mathrm{T}}(1,0)^{\mathrm{T}}(0,1)^{\mathrm{T}}\}$。



题图 7.5

习题 8

8.1 令 k_1 和 k_2 是 $X\times X$ 空间的核函数，$X\subseteq R^n$，$a\in R^+$，$f(\cdot)$ 是 X 空间的实值函数，φ: $X\to R^N$ 是 $R^N\times R^N$ 空间具有核函数 k_3 的映射。$\boldsymbol{B}$ 是 $n\times n$ 的对称半正定矩阵。证明下列函数是核函数。

(1) $k(\boldsymbol{x},\boldsymbol{z})=k_1(\boldsymbol{x},\boldsymbol{z})+k_2(\boldsymbol{x},\boldsymbol{z})$

(2) $k(\boldsymbol{x},\boldsymbol{z})=ak_1(\boldsymbol{x},\boldsymbol{z})$

(3) $k(\boldsymbol{x},\boldsymbol{z})=k_1(\boldsymbol{x},\boldsymbol{z})k_2(\boldsymbol{x},\boldsymbol{z})$

(4) $k(\boldsymbol{x},\boldsymbol{z})=f(\boldsymbol{x})f(\boldsymbol{z})$

(5) $k(\boldsymbol{x},\boldsymbol{z})=k_3(\varphi(\boldsymbol{x})\varphi(\boldsymbol{z}))$

(6) $k(\boldsymbol{x},\boldsymbol{z})=\boldsymbol{x}'B\boldsymbol{z}$

证明：令 S 是有限点集$\{\boldsymbol{x}_1,\ \cdots,\ \boldsymbol{x}_l\}$，并且令 $\boldsymbol{K}_1$ 和 $\boldsymbol{K}_2$ 是通过约束 k_1 和 k_2 在这些点上得到的相应核矩阵。$\boldsymbol{\alpha}\in R^l$ 是任意向量。如果对于所有的 $\boldsymbol{\alpha}$，都有 $\boldsymbol{\alpha}'\boldsymbol{K}\boldsymbol{\alpha}\geqslant 0$，则 $\boldsymbol{K}$ 是半正定矩阵。

(1)因为 $\boldsymbol{\alpha}'(\boldsymbol{K}_1+\boldsymbol{K}_2)\boldsymbol{\alpha}=\boldsymbol{\alpha}'\boldsymbol{K}_1\boldsymbol{\alpha}+\boldsymbol{\alpha}'\boldsymbol{K}_2\boldsymbol{\alpha}\geq 0$，因此 $\boldsymbol{K}_1+\boldsymbol{K}_2$ 是半正定的，所以 k_1+k_2 是核函数。特征向量是相应向量的串联，$\varphi(\boldsymbol{x})=[\varphi_1(\boldsymbol{x}),\varphi_2(\boldsymbol{x})]$，因此

$$\begin{aligned}k(\boldsymbol{x},\boldsymbol{z})&=<\varphi(\boldsymbol{x}),\varphi(\boldsymbol{z})>=<[\varphi_1(\boldsymbol{x}),\varphi_2(\boldsymbol{x})],[\varphi_1(\boldsymbol{z}),\varphi_2(\boldsymbol{z})]>\\&=<\varphi_1(\boldsymbol{x}),\varphi_1(\boldsymbol{z})>+<\varphi_2(\boldsymbol{x}),\varphi_2(\boldsymbol{z})>\\&=k_1(\boldsymbol{x},\boldsymbol{z})+k_2(\boldsymbol{x},\boldsymbol{z})\end{aligned}$$

(2)类似地，$\boldsymbol{\alpha}'a\boldsymbol{K}\boldsymbol{\alpha}=a\boldsymbol{\alpha}'\boldsymbol{K}\boldsymbol{\alpha}\geqslant 0$，因此 ak_1 是核函数。

(3)令 $\boldsymbol{K}=\boldsymbol{K}_1\otimes\boldsymbol{K}_2$ 是 $\boldsymbol{K}_1$ 和 $\boldsymbol{K}_2$ 的张量积，两个半正定矩阵的张量积也是半正定的，因为张量积的本征值是两个半正定矩阵本征值的积。相应于 k_1k_2 函数的矩阵是 Schur 积 $\boldsymbol{H}$。对于任意 $\boldsymbol{\alpha}\in R^l$，$\boldsymbol{\alpha}_1\in R^{l^2}$，有 $\boldsymbol{\alpha}'\boldsymbol{H}\boldsymbol{\alpha}=\boldsymbol{\alpha}_1'\boldsymbol{K}\boldsymbol{\alpha}_1\geqslant 0$，因此 $\boldsymbol{H}$ 是半正定阵，因此 k_1k_2 是核函数。

对于 $k(\boldsymbol{x},\boldsymbol{z})=k_1(\boldsymbol{x},\boldsymbol{z})k_2(\boldsymbol{x},\boldsymbol{z})$ 的 Hadamard 构造，相应的特征是所有特征对的乘积，这样，第(i,j)对特征就由下式给出：

$$\varphi(\boldsymbol{x})_{ij}=\varphi_1(\boldsymbol{x})_i\varphi_2(\boldsymbol{x})_j\ ,\quad i=1,\ \cdots,N_1;\quad j=1,\ \cdots,N_2$$

其中 N_i 是相应于 φ_i, $i=1,2$ 的特征空间特征维数，则其内积由下式给出：

$$\begin{aligned}k(\boldsymbol{x},\boldsymbol{z})&=<\varphi(\boldsymbol{x}),\varphi(\boldsymbol{z})>=\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\varphi(\boldsymbol{x})_{ij}\varphi(\boldsymbol{z})_{ij}\\&=\sum_{i=1}^{N_1}\varphi_1(\boldsymbol{x})_i\varphi_1(\boldsymbol{z})_i\sum_{j=1}^{N_2}\varphi_2(\boldsymbol{x})_j\varphi(\boldsymbol{z})_j\\&=k_1(\boldsymbol{x},\boldsymbol{z})k_2(\boldsymbol{x},\boldsymbol{z})\end{aligned}$$

(4) 对于一维的特征映射 φ: $\boldsymbol{x}$ a $f(\boldsymbol{x}) \in R$， $k(\boldsymbol{x}, \boldsymbol{z})$ 是相应的核函数。

(5) 因为 k_3 是核函数，通过约束 k_3 到点 $\varphi(\boldsymbol{x}_1)$，…，$\varphi(\boldsymbol{x}_l)$ 得到的矩阵是半正定矩阵。因此 $k(\boldsymbol{x}, \boldsymbol{z}) = k_3(\varphi(\boldsymbol{x})\varphi(\boldsymbol{z}))$ 是核函数。

(6) 假设 $\boldsymbol{V}$ 为正交矩阵，$\boldsymbol{B} = \boldsymbol{V}'\boldsymbol{\Lambda}\boldsymbol{V}$，其中 $\boldsymbol{\Lambda}$ 是包含非负本征值的对角矩阵。令 $\sqrt{\boldsymbol{\Lambda}}$ 是具有本征值平方根的对角矩阵且有 $\boldsymbol{A} = \sqrt{\boldsymbol{\Lambda}}\boldsymbol{V}$，因此有

$$k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}'\boldsymbol{B}\boldsymbol{z} = \boldsymbol{x}'\boldsymbol{V}'\boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{z} = \boldsymbol{x}'\boldsymbol{A}'\boldsymbol{A}\boldsymbol{z} = <\boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{z}>$$

是利用线性特征映射 $\boldsymbol{A}$ 的内积。

8.2　令 $k_1(\boldsymbol{x}, \boldsymbol{z})$ 是 $X \times X$ 空间的核函数，其中 $\boldsymbol{x}, \boldsymbol{z} \in X$，$p(\boldsymbol{x})$ 是具有正系数的多项式，则下面的函数也是核函数：

(1) $k(\boldsymbol{x}, \boldsymbol{z}) = p(k_1(\boldsymbol{x}, \boldsymbol{z}))$

(2) $k(\boldsymbol{x}, \boldsymbol{z}) = \exp(k_1(\boldsymbol{x}, \boldsymbol{z}))$

(3) $k(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\| \boldsymbol{x} - \boldsymbol{z} \|^2 / (2\sigma^2))$

证明：(1) 如果 $f(\cdot)$ 是常数，对于多项式结果，从上题中的 (1)~(4) 说明覆盖的是常数项。

(2) 指数函数任意逼近具有正系数的多项式，因此是一个有限的核函数。因为有限的半正定特性是接近有限点，因此该函数依然是核函数。

(3) 由 (2) 有对于 $\sigma \in R^+$， $\exp(< \boldsymbol{x} - \boldsymbol{z} > / \sigma^2)$ 是核函数。规范化核函数可得核函数

$$\frac{\exp(< \boldsymbol{x}, \boldsymbol{z} > / \sigma^2)}{\sqrt{\exp(\| \boldsymbol{x} \|^2 / \sigma^2)\exp(\| \boldsymbol{z} \|^2 / \sigma^2)}} = \exp\left(\frac{< \boldsymbol{x}, \boldsymbol{z} >}{\sigma^2} - \frac{< \boldsymbol{x}, \boldsymbol{x} >}{2\sigma^2} - \frac{< \boldsymbol{z}, \boldsymbol{z} >}{2\sigma^2} \right) = \exp\left(-\frac{\| \boldsymbol{x} - \boldsymbol{z} \|^2}{2\sigma^2} \right)$$

习题 9

9.1　计算模糊集合 $\tilde{A}$ 的海明模糊度、欧几里得模糊度和熵模糊度，其中

$$\tilde{A} = \frac{0.9}{a} + \frac{0.7}{b} + \frac{0.5}{c} + \frac{0.3}{d} + \frac{0.2}{e}$$

解：因为

$$A_{0.5} = \frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{0}{d} + \frac{0}{e}$$

则海明模糊度为

$$d_1(\tilde{A}) = \frac{2}{5}(|0.9-1| + |0.7-1| + |0.5-1| + |0.3-0| + |0.2-0|) = 0.56$$

欧几里得模糊度为

$$d_2(\tilde{A}) = \frac{2}{\sqrt{5}}\sqrt{(0.9-1)^2 + (0.7-1)^2 + (0.5-1)^2 + (0.3-0)^2 + (0.2-0)^2} = 0.62$$

熵模糊度为

$$d_E(\tilde{A}) = \frac{1}{n\ln 2}\sum_{i=1}^{n}\{-\mu_{\tilde{A}}(x_i)\ln\mu_{\tilde{A}}(x_i) - [1-\mu_{\tilde{A}}(x_i)]\ln[1-\mu_{\tilde{A}}(x_i)]\}$$

$$= \frac{1}{5\ln 2}[(-0.9\ln 0.9 - 0.1\ln 0.1) + (-0.7\ln 0.7 - 0.3\ln 0.3) + (-0.5\ln 0.5 - 0.5\ln 0.5)$$

$$+(-0.3\ln 0.3 - 0.7\ln 0.7) + (-0.2\ln 0.2 - 0.8\ln 0.8)]$$

$$= (0.095+0.23+0.25+0.361+0.347+0.347+0.361+0.25+0.322+0.179)/(5\ln 2)$$

$$= 0.79$$

9.2 设 $\boldsymbol{U} = \{a, b, c, d, e, f\}$，$\tilde{A} = \frac{0.6}{a} + \frac{0.8}{b} + \frac{1}{c} + \frac{0.8}{d} + \frac{0.6}{e} + \frac{0.2}{f}$，$\tilde{B} = \frac{0.4}{a} + \frac{0.6}{b} + \frac{0.5}{c} + \frac{1}{d} + \frac{0.8}{e} + \frac{0.3}{f}$，试分别计算格贴近度 $N(\tilde{A}, \tilde{B})$、海明贴近度 $N_H(\tilde{A}, \tilde{B})$ 和欧几里得贴近度 $N_E(\tilde{A}, \tilde{B})$。

解：$\tilde{A}$与 $\tilde{B}$的内积为

$$\tilde{A} \circ \tilde{B} = (0.6 \wedge 0.4) \vee (0.8 \wedge 0.6) \vee (1 \wedge 0.5) \vee (0.8 \wedge 1) \vee (0.6 \wedge 0.8) \vee (0.2 \wedge 0.3)$$

$$= 0.4 \vee 0.6 \vee 0.5 \vee 0.8 \vee 0.6 \vee 0.2 = 0.8$$

$\tilde{A}$与 $\tilde{B}$的外积为

$$\tilde{A} \oplus \tilde{B} = (0.6 \vee 0.4) \wedge (0.8 \vee 0.6) \wedge (1 \vee 0.5) \wedge (0.8 \vee 1) \wedge (0.6 \vee 0.8) \wedge (0.2 \vee 0.3)$$

$$= 0.6 \wedge 0.8 \wedge 1 \wedge 1 \wedge 0.8 \wedge 0.3 = 0.3$$

$\tilde{A}$与 $\tilde{B}$的格贴近度为

$$N(\tilde{A}, \tilde{B}) = (\tilde{A} \circ \tilde{B}) \wedge (1 - \tilde{A} \oplus \tilde{B}) = 0.8 \wedge (1 - 0.3) = 0.7$$

$\tilde{A}$与 $\tilde{B}$的海明贴近度为

$$N_H(\tilde{A}, \tilde{B}) = 1 - \frac{1}{n}\sum_{i=1}^{n} \left| \tilde{A}(u_i) - \tilde{B}(u_i) \right|$$

$$= 1(|0.6 0.4| + |0.8 0.6| + |1 0.5| + |0.8 1| + |0.6 0.8| + |0.2 0.3|) / 6$$

$$= 0.77$$

$\tilde{A}$与 $\tilde{B}$的欧几里得贴近度为

$$N_E(\tilde{A}, \tilde{B}) = 1 - \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^{n} \left[\tilde{A}(u_i) - \tilde{B}(u_i) \right]^2}$$

$$= 1 - \sqrt{(0.6-0.4)^2 + (0.8-0.6)^2 + (1-0.5)^2 + (0.8-1)^2 + (0.6-0.8)^2 + (0.2-0.3)^2} / \sqrt{6}$$

$$= 0.74$$

9.3 设论域 $\boldsymbol{U} = \{a, b, c, d, e, f\}$，$\boldsymbol{U}$ 上的模糊子集有 6 个：

$$\tilde{A}_1 = (1, 0.8, 0.5, 0.4, 0, 0.1),$$

$$\tilde{A}_2 = (0.5, 0.1, 0.8, 1, 0.6, 0),$$

$$\tilde{A}_3 = (0, 1, 0.2, 0.7, 0.5, 0.8),$$

$$\tilde{A}_4 = (0.4, 0, 1, 0.9, 0.6, 0.5),$$

$$\tilde{A}_5 = (0.8, 0.2, 0, 0.5, 1, 0.7),$$

$$\tilde{A}_6 = (0.5, 0.7, 0.8, 0, 0.5, 1),$$

且待识别的模糊子集是 $\tilde{A}_x$ = (0.7, 0.2, 0.1, 0.4, 1, 0.8)。采用格贴近度确定待识别的模糊子集与 $\tilde{A}_1$~ $\tilde{A}_6$ 中哪个最相近。

解：根据格贴近度计算公式 $N(\tilde{A},\tilde{B}) = (\tilde{A}\circ\tilde{B}) \wedge (1 - \tilde{A}\odot\tilde{B})$，有

$$\tilde{A}_1\circ\tilde{A}_x = (1\wedge0.7)\vee(0.8\wedge0.2)\vee(0.5\wedge0.1)\vee(0.4\wedge0.4)\vee(0\wedge1)\vee(0.1\wedge0.8)$$

$$= 0.7\vee0.2\vee0.1\vee0.4\vee0\vee0.1 = 0.7$$

$$\tilde{A}_1\odot\tilde{A}_x = (1\vee0.7)\wedge(0.8\vee0.2)\wedge(0.5\vee0.1)\wedge(0.4\wedge0.4)\wedge(0\vee1)\wedge(0.1\vee0.8)$$

$$= 1\wedge0.8\wedge0.5\wedge0.4\wedge1\wedge0.8 = 0.4$$

$$N(\tilde{A}_1,\ \tilde{A}_x) = 0.7\wedge(1-0.4) = 0.6$$

同理有

$$N(\tilde{A}_2,\ \tilde{A}_x) = 0.6\wedge(1-0.2) = 0.6$$

$$N(\tilde{A}_3,\ \tilde{A}_x) = 0.8\wedge(1-0.2) = 0.8$$

$$N(\tilde{A}_4,\ \tilde{A}_x) = 0.6\wedge(1-0.2) = 0.6$$

$$N(\tilde{A}_5,\ \tilde{A}_x) = 1\wedge(1-0.1) = 0.9$$

$$N(\tilde{A}_6,\ \tilde{A}_x) = 0.8\wedge(1-0.4) = 0.6$$

可见，$\tilde{A}_x$ 与 $\tilde{A}_5$ 最相近。

9.4　设论域 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}$，给定 $\boldsymbol{X}$ 上一个模糊关系 $\tilde{R}$，其模糊矩阵为

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0.8 & 0.8 & 0.2 & 0.8 \\ 0.8 & 1 & 0.85 & 0.2 & 0.85 \\ 0.8 & 0.85 & 1 & 0.2 & 0.9 \\ 0.2 & 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.85 & 0.9 & 0.2 & 1 \end{bmatrix}$$

判断 $\boldsymbol{R}$ 是模糊相似矩阵还是模糊等价矩阵；按不同的λ分类并给出分级聚类树。

解：矩阵显然具有自反性和对称性，而

$$\boldsymbol{R}\circ\boldsymbol{R} = \begin{bmatrix} 1 & 0.8 & 0.8 & 0.2 & 0.8 \\ 0.8 & 1 & 0.85 & 0.2 & 0.85 \\ 0.8 & 0.85 & 1 & 0.2 & 0.9 \\ 0.2 & 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.85 & 0.9 & 0.2 & 1 \end{bmatrix} = \boldsymbol{R}$$

所以，$\boldsymbol{R}$ 具有传递性，故 $\boldsymbol{R}$ 是模糊等价矩阵。

令λ由 1 降至 0，写出 $\boldsymbol{R}_\lambda$，按 $\boldsymbol{R}_\lambda$分类。

(1) 当$\lambda = 1$ 时，$\boldsymbol{R}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$。此时分为 5 类，即$\{\boldsymbol{x}_1\}, \{\boldsymbol{x}_2\}, \{\boldsymbol{x}_3\}, \{\boldsymbol{x}_4\}, \{\boldsymbol{x}_5\}$，这是最细分类。

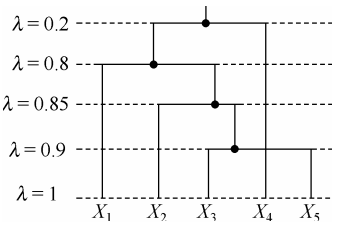
(2) 当$\lambda = 0.9$ 时，$\boldsymbol{R}_{0.9} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$。此时分为 4 类，即$\{\boldsymbol{x}_1\}, \{\boldsymbol{x}_2\}, \{\boldsymbol{x}_3, \boldsymbol{x}_5\}, \{\boldsymbol{x}_4\}$。

(3) 当$\lambda = 0.85$ 时，$\boldsymbol{R}_{0.85} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$。此时分为 3 类，即$\{\boldsymbol{x}_1\}, \{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_5\}, \{\boldsymbol{x}_4\}$。

(4) 当$\lambda = 0.8$ 时，$\boldsymbol{R}_{0.8} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$。此时分为 2 类，即$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_5\}, \{\boldsymbol{x}_4\}$。

(5) 当$\lambda = 0.2$ 时，$\boldsymbol{R}_{0.2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$。此时 5 个元素分为 1 类，即$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}$，它是最粗分类。

于是可得出分级聚类树，也称动态聚类图，如题图9.4所示。



题图 9.4

9.5 已知 12 个二维样本 $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, 2, \cdots, 12\}$，其中 $\boldsymbol{x}_1 = [0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2 = [0, 1]^{\mathrm{T}}$, $\boldsymbol{x}_3 = [1, 0]^{\mathrm{T}}$, $\boldsymbol{x}_4 = [0.5, 4]^{\mathrm{T}}$, $\boldsymbol{x}_5 = [1, 3]^{\mathrm{T}}$, $\boldsymbol{x}_6 = [1, 5]^{\mathrm{T}}$, $\boldsymbol{x}_7 = [1.5, 4.5]^{\mathrm{T}}$, $\boldsymbol{x}_8 = [6, 4]^{\mathrm{T}}$, $\boldsymbol{x}_9 = [6.5, 5]^{\mathrm{T}}$, $\boldsymbol{x}_{10} = [7, 4]^{\mathrm{T}}$, $\boldsymbol{x}_{11} = [7.5, 7]^{\mathrm{T}}$, $\boldsymbol{x}_{12} = [8, 7]^{\mathrm{T}}$。试用 FCM 算法进行分类，其中模糊性加权指数 $m = 2$，用欧氏距离，类别分别为 2 和 4。

解：(1) 根据题意有样本数 $N = 12$，当类别数 $C = 2$ 时，模糊性加权指数 $m = 2$，矩阵 $\boldsymbol{B}$ 为单位矩阵 $\boldsymbol{I}$，迭代停止阈值 $\varepsilon = \mathrm{e}^{-5}$。

(2) 用随机函数设置初始模糊分类矩阵 $\boldsymbol{U}^{(0)}$：

$$\boldsymbol{U}^{(0)} = \begin{bmatrix} 0.4735 & 0.1221 & 0.8664 & 0.3374 & 0.4981 & 0.1397 & 0.6635 & 0.8494 & 0.3153 & 0.4523 & 0.9484 & 0.4762 \\ 0.5265 & 0.8779 & 0.1336 & 0.6626 & 0.5019 & 0.8603 & 0.3365 & 0.1506 & 0.6847 & 0.5477 & 0.0516 & 0.5238 \end{bmatrix}$$

(3) 计算 $\boldsymbol{U}^{(0)}$ 时的聚类中心 $v_i^{(0)}$，其聚类中心 $\boldsymbol{v}^{(0)}$ 为

$$\boldsymbol{v}^{(0)} = \begin{bmatrix} 4.2144 & 3.8803 \\ 2.4187 & 3.5521 \end{bmatrix}$$

(4) 计算新隶属度 $\boldsymbol{U}^{(1)}$ 和聚类中心 $\boldsymbol{v}^{(1)}$：

$\boldsymbol{U}^{(1)}=$

$$\begin{bmatrix}0.3601 & 0.3218 & 0.3656 & 0.2194 & 0.1726 & 0.2618 & 0.1835 & 0.8027 & 0.7433 & 0.7316 & 0.6475 & 0.6414\\ 0.6399 & 0.6782 & 0.6344 & 0.7806 & 0.8274 & 0.7382 & 0.8165 & 0.1973 & 0.2567 & 0.2684 & 0.3525 & 0.3586\end{bmatrix}$$

$$\boldsymbol{v}^{(1)}=\begin{bmatrix}5.7709 & 4.5583\\ 1.4413 & 3.1420\end{bmatrix}$$

⑸ 重复上述过程，共进行 8 次迭代，满足收敛条件，最后得到的隶属度矩阵和聚类中心为

$\boldsymbol{U}=$

$$\begin{bmatrix}0.0755 & 0.0355 & 0.0841 & 0.0562 & 0.0107 & 0.1618 & 0.1423 & 0.9164 & 0.9918 & 0.9560 & 0.9585 & 0.9521\\ 0.9245 & 0.9645 & 0.9159 & 0.9438 & 0.9893 & 0.8382 & 0.8577 & 0.0836 & 0.0082 & 0.0440 & 0.0415 & 0.0479\end{bmatrix}$$

$$\boldsymbol{v}=\begin{bmatrix}6.9251 & 5.3922\\ 0.6992 & 2.4085\end{bmatrix}$$

根据隶属度矩阵可知，12 个样本的分类结果为$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6,\boldsymbol{x}_7\}$属于第一类，$\{\boldsymbol{x}_8,\boldsymbol{x}_9,\boldsymbol{x}_{10},\boldsymbol{x}_{11},\boldsymbol{x}_{12}\}$属于第二类，如题图9.5.1 所示。

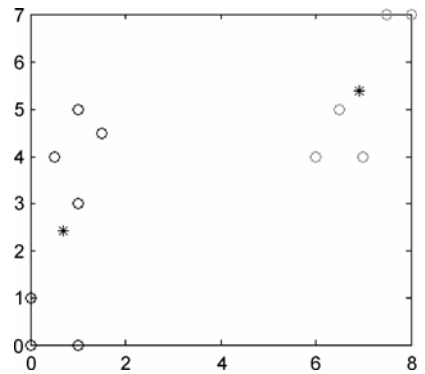
同理，当类别数 $C=4$ 时，FCM 的聚类结果为

$\boldsymbol{U}=$

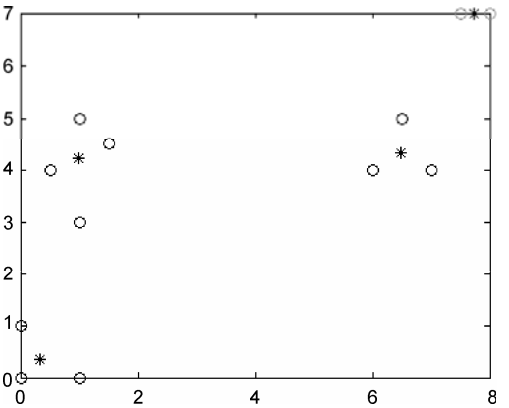
$$\begin{bmatrix}0.0038 & 0.0096 & 0.0110 & 0.0079 & 0.0364 & 0.0190 & 0.0128 & 0.9529 & 0.9006 & 0.9499 & 0.0071 & 0.0069\\ 0.0021 & 0.0053 & 0.0057 & 0.0046 & 0.0189 & 0.0118 & 0.0071 & 0.0270 & 0.0782 & 0.0351 & 0.9911 & 0.9913\\ 0.0122 & 0.0449 & 0.0301 & 0.9662 & 0.7894 & 0.9427 & 0.9628 & 0.0129 & 0.0139 & 0.0092 & 0.0012 & 0.0012\\ 0.9819 & 0.9403 & 0.9532 & 0.0213 & 0.1552 & 0.0264 & 0.0172 & 0.0071 & 0.0072 & 0.0058 & 0.0006 & 0.0006\end{bmatrix}$$

$$\boldsymbol{v}=\begin{bmatrix}6.4944 & 4.3086\\ 7.7429 & 6.9896\\ 0.9991 & 4.2130\\ 0.3360 & 0.3463\end{bmatrix}$$

由隶属度矩阵可知，12 个样本的分类结果为$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3\}$属于第四类，$\{\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6,\boldsymbol{x}_7\}$属于第三类，$\{\boldsymbol{x}_8,\boldsymbol{x}_9,\boldsymbol{x}_{10}\}$属于第一类，$\{\boldsymbol{x}_{11},\boldsymbol{x}_{12}\}$属于第二类，如题图9.5.2 所示。



题图 9.5.1



题图 9.5.2

9.6 已知 29 个二维样本，具体是 $\boldsymbol{x}_1=[0,0]^{\mathrm{T}}$, $\boldsymbol{x}_2=[0,1]^{\mathrm{T}}$, $\boldsymbol{x}_3=[0,2]^{\mathrm{T}}$, $\boldsymbol{x}_4=[0,3]^{\mathrm{T}}$, $\boldsymbol{x}_5=[1,0]^{\mathrm{T}}$, $\boldsymbol{x}_6=[1,1]^{\mathrm{T}}$, $\boldsymbol{x}_7=[1,2]^{\mathrm{T}}$, $\boldsymbol{x}_8=[1,3]^{\mathrm{T}}$, $\boldsymbol{x}_9=[2,0]^{\mathrm{T}}$, $\boldsymbol{x}_{10}=[2,1]^{\mathrm{T}}$, $\boldsymbol{x}_{11}=[2,2]^{\mathrm{T}}$, $\boldsymbol{x}_{12}=[2,3]^{\mathrm{T}}$, $\boldsymbol{x}_{13}=[3,0]^{\mathrm{T}}$, $\boldsymbol{x}_{14}=[3,1]^{\mathrm{T}}$, $\boldsymbol{x}_{15}=[3,2]^{\mathrm{T}}$, $\boldsymbol{x}_{16}=[3,3]^{\mathrm{T}}$, $\boldsymbol{x}_{17}=[1,6]^{\mathrm{T}}$, $\boldsymbol{x}_{18}=[1,7]^{\mathrm{T}}$, $\boldsymbol{x}_{19}=[1,8]^{\mathrm{T}}$, $\boldsymbol{x}_{20}=[2,6]^{\mathrm{T}}$, $\boldsymbol{x}_{21}=[2,7]^{\mathrm{T}}$, $\boldsymbol{x}_{22}=[2,8]^{\mathrm{T}}$, $\boldsymbol{x}_{23}=[3,6]^{\mathrm{T}}$, $\boldsymbol{x}_{24}=[3,7]^{\mathrm{T}}$, $\boldsymbol{x}_{25}=[3,8]^{\mathrm{T}}$, $\boldsymbol{x}_{26}=[7,2]^{\mathrm{T}}$, $\boldsymbol{x}_{27}=[7,3]^{\mathrm{T}}$, $\boldsymbol{x}_{28}=[8,2]^{\mathrm{T}}$, $\boldsymbol{x}_{29}=[8,3]^{\mathrm{T}}$。取模糊性加权指数 $m=2$、欧氏距离，试用 FCM 算法将数据样本集分为三类。

解：根据题意有样本数 $N=29$，类别数 $C=3$，模糊性加权指数 $m=2$，矩阵 $\boldsymbol{A}$ 为单位矩阵 $\boldsymbol{I}$，迭代停止阈值 $\varepsilon=\mathrm{e}^{-5}$。

用 FCM 算法经过 11 次迭代，得到的聚类结果

$\boldsymbol{U}=$

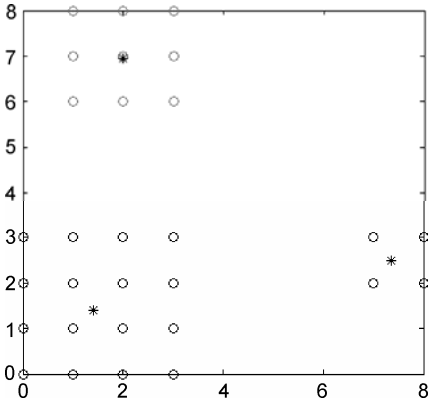
$$\begin{bmatrix}
0.0576 & 0.0352 & 0.0386 & 0.0636 & 0.0421 & 0.0077 & 0.0127 & 0.0544 & 0.0591 & 0.0156 \\
0.0668 & 0.0506 & 0.0742 & 0.1779 & 0.0399 & 0.0091 & 0.0203 & 0.1343 & 0.0429 & 0.0137 \\
0.8756 & 0.9142 & 0.8872 & 0.7585 & 0.9180 & 0.9832 & 0.9670 & 0.8113 & 0.8980 & 0.9706
\end{bmatrix}$$

$$\begin{bmatrix}
0.0228 & 0.0776 & 0.1396 & 0.1046 & 0.1177 & 0.1673 & 0.0314 & 0.0153 & 0.0274 & 0.0203 \\
0.0271 & 0.1449 & 0.0715 & 0.0611 & 0.0891 & 0.1948 & 0.8910 & 0.9552 & 0.9282 & 0.9408 \\
0.9500 & 0.7776 & 0.7888 & 0.8343 & 0.7931 & 0.6380 & 0.0777 & 0.0295 & 0.0444 & 0.0389
\end{bmatrix}$$

$$\begin{bmatrix}
0.0001 & 0.0181 & 0.0535 & 0.0248 & 0.0399 & 0.9812 & 0.9791 & 0.9752 & 0.9732 \\
0.9998 & 0.9574 & 0.8756 & 0.9463 & 0.9173 & 0.0073 & 0.0095 & 0.0104 & 0.0126 \\
0.0001 & 0.0244 & 0.0709 & 0.0289 & 0.0428 & 0.0115 & 0.0114 & 0.0144 & 0.0142
\end{bmatrix}$$

$$\boldsymbol{v}=\begin{bmatrix}
7.3621 & 2.4880 \\
1.9856 & 6.9419 \\
1.4185 & 1.4010
\end{bmatrix}$$

由隶属度矩阵可知，29 个样本的分类结果为{$\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}, \boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15}, \boldsymbol{x}_{16}$}属于第三类，{$\boldsymbol{x}_{17}, \boldsymbol{x}_{18}, \boldsymbol{x}_{19}, \boldsymbol{x}_{20}, \boldsymbol{x}_{21}, \boldsymbol{x}_{22}, \boldsymbol{x}_{23}, \boldsymbol{x}_{24}, \boldsymbol{x}_{25}$, }属于第二类，{$\boldsymbol{x}_{26}$, $\boldsymbol{x}_{27}$, $\boldsymbol{x}_{28}$, $\boldsymbol{x}_{29}$}属于第一类，如题图9.6所示。



题图 9.6

9.7 用 FCM 对鸢尾属植物(Iris)样本数据进行分类，其中模糊性加权指数 $m=2$，用欧

氏距离，类别分别为 3。

解：附录 A 给出了鸢尾属植物(Iris)样本数据，共 150 个样本，每个样本有 4 个特征向量；150 个样本被分为 3 类：1~50, 51~100, 101~150。

可以调用 MATLAB 提供的 FCM 程序，源程序如下：

```
[data, headertext] = xlsread('iris.xls', 'data');  % load iris data
[center, U, obj_fcn] = fcm(data, 3);
   xlswrite('iris_result.xls', U', 'sheet1');      %save the grade of
   membership
xlswrite('iris_result.xls', center', 'sheet2');    %save cluster center
   xlswrite('iris_result.xls', obj_fcn, 'sheet3'); %save objective
function
```

运行上述程序后，共进行 25 次迭代，FCM 对鸢尾属植物(Iris)样本数据的聚类结果保存在 iris_result.xls 文件中，相应的各次迭代的准则函数在表单 sheet1 中，表单 sheet2 中给出的聚类中心是：

第一类：1.397156, 4.363643, 2.760993, 5.888721

第二类：2.053424, 5.646464, 3.052308, 6.774756

第三类：0.253539, 1.482799, 3.414099, 5.003965

观察 iris_result.xls 文件的表单 sheet1 中给出的隶属度矩阵，其中 1~50 样本都属于第三类，无错误分类样本；52, 54~77, 79~100 样本都属于第一类，51, 53, 78 共 3 个样本被错分为第二类；102, 107, 114, 122, 124, 127, 128, 134, 139, 143, 147, 150 共 12 个样本被错分为第一类，而其余的从 101~150 的 38 个样本被正确地分类到第二类。

本次聚类分析的正确率为(150–3–12)/150 = 90%。