第一章 绪 论

1.1 模式识别的概念

人有这样一种能力：听到走廊里的脚步声，就知道谁来了；在人群中掠过一个人的背影，就能认出这个人是谁；留言条上的字，一看就知道是谁写的，而且尽管写得龙飞凤舞，还是能认出写的是什么意思。这种能力就是人的识别能力。

随着计算机科学的发展和计算机应用的普及，迫切希望计算机也能听懂我们说的话，看懂我们写的字，……。这种强烈愿望和不断探索实践，促使模式识别这门学科得以形成和发展。

什么是模式和模式识别呢？按照广义的定义，模式是一些供模仿用的、完美无缺的标本。模式识别就是识别出特定客体所模仿的标本。根据客体的性质，将客体分成两种类型：抽象的客体和具体的客体。论点、思想、信仰、……，是非物质的客体，对它们的研究主要属于哲学、政治学的范畴；声音、图象、文字、……，是具体的客体，它们通过对感官的刺激而被识别。研究人类对具体客体的识别机理是沿着两个方向进行的。一个方向是研究人类对客体识别能力的生物学机理，这属于生物学、生理学及心理学的范畴；另一方向是研究用计算机模拟人的识别能力，提出识别具体客体的基本理论与实用技术，这就是模式识别这一学科的研究内容。根据模式识别的研究内容，我们对模式和模式识别作如下狭义的定义：**模式是对感兴趣的客体的定量的或结构的描述；模式类是具有某些共同特性的模式的集合。模式识别是研究一些自动技术，依靠这些技术，计算机自动地（或者人进行少量干涉）把待识模式分到各自的**

模式类中去。

1.2 模式识别的方法

从上一节中模式的定义可以看出，描述模式有两种方法：定量描述和结构性描述。定量描述就是用一组数据来描述模式。比如，判断某细胞是正常细胞还是癌变细胞，我们可以抓住两个特征，一个是细胞的圆形度 x_1，一个是细胞的形心偏差度 x_2。因为，正常细胞比较圆，即圆形度大；正常细胞的细胞核中心偏离细胞中心小，即形心偏差度小；而癌变细胞的圆形度小，形心偏差度大，这两个特征能比较有效地区分正常细胞与癌变细胞，所以，我们可以用这两个特征来描述细胞。为了便于数学处理，我们把这些数据特征组成向量，称为特征向量。比如，把上述描述细胞的两个特征（圆形度 x_1 和形心偏差度 x_2）组成描述细胞的特征向量：$\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$，其中 T 是转置符号。另一种描述模式的方法是结构性描述，即用一组基元来描述模式。比如，我们可以用一组基元来描述图 1－1 中的图形。

这个图形由四个基本元素（称为基元）组成。两个圆弧段分别用符号 a 和 c 表示，直线段用符号 b 表示。同样，为了便于用形式语言处理，我们把这些符号组成符号串：$\boldsymbol{x}=abcb$，这四个基元不仅分别表达了图形中四个线段的局部特征，而且这些基元之间的连接关系从结构上描述了这个图形。

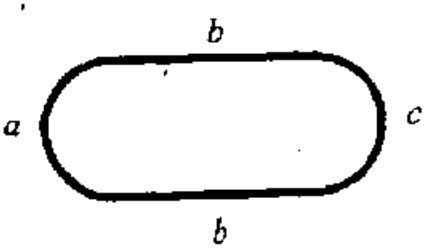


图 1－1

相应于两种模式描述方法，有两种基本的模式识别方法：统计模式识别方法和结构（句法）模式识别方法。在统计模式识别方法中，用特征向量描述模式；在结构模式识别方法中，用符号串（树）来描述模式。本书只讨论统计模式识别方法。基于统计识别法的

模式识别系统主要由五部分组成：数据获取、预处理、特征抽取、分类器设计和分类器。具体见图 1－2。

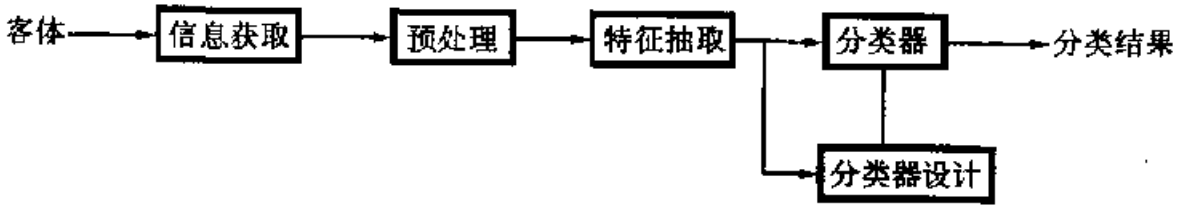


图 1－2

下面对这五个部分作些说明。

1. 数据获取

为了使计算机能够对客体进行分类识别，必须将客体用计算机所能接受的形式表示，通常从客体获得的信息有下列三种类型：

①二维图像，如文字、指纹、照片等；

②一维波形，如语音、机械振动波、心电图等；

③物理参量和逻辑值，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象或一维波形，这就是信息获取过程。

2. 预处理

预处理的目的是去除噪声，加强有用的信息，并对种种因素造成的退化现象进行复原。

3. 特征抽取

由信息获取部分获得的原始数据量一般是相当大的。为了有效地实现分类识别，要对原始数据进行选择或变换，得到最能反映分类本质的特征，构成特征向量。这就是特征抽取的过程。

4. 分类器设计

为了把待识模式分配到各自的模式类中去，必须设计出一套分类判别规则。基本作法是：用一定数量的样本（称为训练样本集），确定出一套分类判别规则，使得按这套分类判别规则对待识

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。这就是分类器设计的过程。

5. 分类器

分类器按已确定的分类判别规则对待识模式进行分类判别，输出分类结果。

模式识别系统中的第一和第二部分是数字信号处理和图象处理等课程的研究课题，本书只讨论第三、第四和第五部分的理论和方法。

模式识别是60年代发展起来的一门学科。目前，模式识别技术已在语音识别、文字识别、图象识别等领域得到成功的应用。但是由于问题的复杂性，离人们的期望还有一段距离。因此，模式识别仍然是一门发展中的新兴学科，新的理论和方法不断出现，同时，与其它学科相互结合和相互渗透，不断推动模式识别向前发展。模糊数学和神经网络引入模式识别就是典型的例子。由于客体的特征常常带有某些模糊，因此，1965年扎德提出模糊集合的概念后不久，模糊数学就深入到模式识别的许多环节。80年代掀起的人工神经网络研究热潮很快就波及到模式识别，出现了模糊模式识别和神经网络模式识别的提法。本书不把模糊模式识别和神经网络模式识别从统计模式识别中分离出来，而是以人工神经网络为主干线(或称为平台)将现有的模式识别方法融汇贯通起来，培育一个独特的模式识别体系结构，让读者从中掌握到模式识别的精髓，为以后从事模式识别研究打下坚实的基础。

第二章　贝叶斯分类器

模式识别的分类问题就是根据待识客体的特征向量值及其它约束条件将其分到某个类别中去。贝叶斯决策理论是处理模式分类问题的基本理论之一。本章要讨论的贝叶斯分类器在统计模式识别中被称为最优分类器。采用贝叶斯分类器必须满足下列两个先决条件：

①要决策分类的类别数是一定的；

②各类别总体的概率分布是已知的。

在条件①中，假设要研究的分类问题有 c 个类别，各类别状态用 ω_i 来表示，$i=1,2,\cdots,c$。在条件②中，假设待识客体的特征向量值 $\boldsymbol{x}$ 所对应的状态后验概率 $P(\omega_i/\boldsymbol{x})$ 是已知的；或者，对应于各个类别 ω_i 出现的先验概率 $P(\omega_i)$ 和类条件概率密度函数 $p(\boldsymbol{x}/\omega_i)$ 是已知的。

2.1　最小错误率贝叶斯决策

我们从一个实际例子讲起。假如我们要进行细胞识别，识别某细胞属正常类 ω_1 还是属癌变类 ω_2？假设待识细胞已作过预处理，抽取出二个特征：圆形度 x_1 和形心偏差度 x_2，组成特征向量 $\boldsymbol{x}$（或称为模式 $\boldsymbol{x}$）。现在要识别模式 $\boldsymbol{x}$ 属正常类 ω_1 还是属癌变类 ω_2，根据什么规则来识别？一个直观的想法是：如果模式 $\boldsymbol{x}$ 属正常类 ω_1 的概率大于模式 $\boldsymbol{x}$ 属癌变类 ω_2 的概率，则决策模式 $\boldsymbol{x}$ 属正常类 ω_1；反之，如果模式 $\boldsymbol{x}$ 属正常类 ω_1 的概率小于模式 $\boldsymbol{x}$ 属癌变类 ω_2 的概率，则决策模式 $\boldsymbol{x}$ 属癌变类 ω_2。这句话用数学语言可表示为：

若　　$$P(\omega_1/\boldsymbol{x}) \gtrless P(\omega_2/\boldsymbol{x})$$

则 $$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

其中，条件概率 $P(\omega_i/x)$称为状态的后验概率。

利用贝叶斯公式

$$P(\omega_i/x)=\frac{p(x/\omega_i)P(\omega_i)}{p(x)}$$

上面的决策规则可改写为：

若 $$\frac{p(x/\omega_1)P(\omega_1)}{p(x)}\gtrless\frac{p(x/\omega_2)P(\omega_2)}{p(x)}$$

则 $$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

其中，$p(x)>0$，将不等式两边的分母消去，决策规则可改写为：

若 $$p(x/\omega_1)P(\omega_1)\gtrless p(x/\omega_2)P(\omega_2)$$

则 $$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

其中，$p(x/\omega_1)$是正常类下模式 x 的类条件概率密度，$p(x/\omega_2)$是癌变类下模式 x 的类条件概率密度。

这样，最小错误率贝叶斯决策有两种形式，一种是后验概率形式：

若 $$P(\omega_1/x)\gtrless P(\omega_2/x)$$

则 $$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

另一种是类条件概率密度形式：

若 $$p(x/\omega_1)P(\omega_1)\gtrless p(x/\omega_2)P(\omega_2)$$

则 $$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

将二类情况推广到 c 类情况，最小错误率贝叶斯决策规则为：

(1)后验概率形式

若 $$P(\omega_i/x)>P(\omega_j/x),\quad j=1,2,\cdots,c;\quad j\neq i$$

则 $$x\in\omega_i \tag{2.1}$$

(2)类条件概率密度形式

若

$$p(\boldsymbol{x}/\omega_i)P(\omega_i)>p(\boldsymbol{x}/\omega_j)P(\omega_j),\qquad j=1,2,\cdots,c;\ j\neq i$$

则

$$\boldsymbol{x}\in\omega_i \tag{2.2}$$

例 2.1 有一家医院为了研究癌症的诊断，对一大批人作了一次普查，给每人打了试验针，然后进行统计，得到如下统计数字：

①这批人中，每 1 000 人有 5 个癌症病人；

②这批人中，每 100 个正常人有 1 人对试验的反应为阳性；

③这批人中，每 100 个癌症病人有 95 人对试验的反应为阳性。

假如正常人用 ω_1 类表示，癌症病人用 ω_2 类表示。以试验结果作为特征，特征值为阳或阴。根据统计数字，得到如下概率：

$P(\omega_1)=0.995$，$P(\omega_2)=0.005$，$p(\text{阳}/\omega_1)=0.01$，

$p(\text{阴}/\omega_1)=0.99$，$p(\text{阳}/\omega_2)=0.95$，$p(\text{阴}/\omega_2)=0.05$。

通过普查统计，该医院可开展癌症诊断。现在王某，试验结果为阳性，诊断结果是什么？

因为

$$p(\boldsymbol{x}/\omega_1)P(\omega_1)=p(\text{阳}/\omega_1)P(\omega_1)=0.01\times0.995=0.009\ 95$$

$$p(\boldsymbol{x}/\omega_2)P(\omega_2)=p(\text{阳}/\omega_2)P(\omega_2)=0.95\times0.005=0.004\ 75$$

故

$$p(\boldsymbol{x}/\omega_1)P(\omega_1)>p(\boldsymbol{x}/\omega_2)P(\omega_2)$$

所以 $\boldsymbol{x}\in\omega_1$，即王某属正常人。

上面我们介绍了贝叶斯决策规则。应用贝叶斯决策规则对模式 $\boldsymbol{x}$ 进行分类的分类器称为贝叶斯分类器 。

对于 c 类分类问题，按照决策规则可以把特征向量空间（或称模式空间）分成 c 个决策域。我们将划分决策域的边界称为决策边界，在数学上用解析形式可以表示成决策边界方程。用于表达决策规则的某些函数称为判别函数。判别函数与决策边界方程是密切相关的，而且它们都由相应的决策规则所确定。

对于 c 类分类问题，通常定义 c 个判别函数 $d_i(x)$，$i=1,2,\cdots,c$. 对照两种形式的最小错误率贝叶斯决策规则，判别函数显然可定义为：

①$d_i(x)=P(\omega_i/x)$，　$i=1,2,\cdots,c$；

②$d_i(x)=p(x/\omega_i)P(\omega_i)$，　$i=1,2,\cdots,c$.

这样，决策规则可写为：

若　$d_i(x)>d_j(x)$，　$j=1,2,\cdots,c$；$j\neq i$

则　$x\in\omega_i$

确定了判别函数，决策边界也就确定下来了。相邻的两个决策域在决策边界上其判别函数值是相等的。如果决策域 R_i 与 R_j 是相邻的，则分割这两个决策域的决策边界方程应满足：

$$d_i(x)=d_j(x)$$

一般地说，模式 x 为一维时，决策边界为一分界点；x 为二维时，决策边界为一曲线；x 为三维时，决策边界为一曲面；x 为 n 维 ($n>3$)时，决策边界为一超曲面。

分类器可看成是由硬件或软件组成的“机器”，贝叶斯分类器的结构见图 2-1。

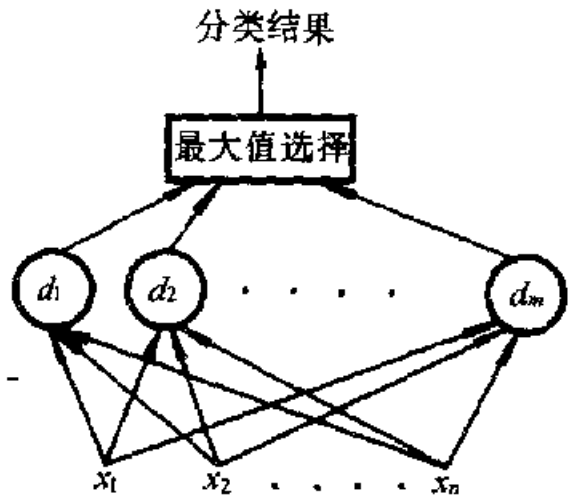


图 2-1

分类器先计算出 c 个判别函数 $d_i(x)$值，再从中选出对应于判

别函数值为最大的类别作为分类结果。

2.2 最小风险贝叶斯决策

在例 2.1 中，王某的试验结果为阳性，根据最小错误率贝叶斯决策，判他属正常人，那么他属正常人的概率是不是 100%呢？我们可计算出试验结果为阳性的条件下他属正常人的概率：

$$P(\omega_1/\text{阳})=\frac{p(\text{阳}/\omega_1)P(\omega_1)}{p(\text{阳}/\omega_1)P(\omega_1)+p(\text{阳}/\omega_2)P(\omega_2)}$$

$$=\frac{0.01\times 0.995}{0.01\times 0.995+0.95\times 0.005}\approx 67.7\%$$

该人属正常人的概率为 67.7%，换句话说，他属癌症病人的概率为 32.2%。

从这里可以看出，尽管采用了最小错误率贝叶斯决策，但仍然可能将正常人错判为癌症病人，也可能将癌症病人错判为正常人。这些错判都会带来一定的损失。将正常人错判为癌症病人，会给他带来短期的精神负担，造成一定的损失，这个损失比较小。如果把癌症病人错判为正常人，致使患者失去挽救的机会，这个损失就大了。这两种不同的错判所造成损失的程度是有显著差别的。所以，在决策时还要考虑到各种错判所造成的不同损失，提出了最小风险贝叶斯决策。

风险是什么？条件风险定义为：**将模式 x 判属某类所造成的损失的条件数学期望**。

仍以细胞识别为例。假定：

模式 x 本属正常类而判属正常类所造成的损失为 L_{11}；

模式 x 本属癌变类而判属正常类所造成的损失为 L_{21}；

模式 x 本属正常类而判属癌变类所造成的损失为 L_{12}；

模式 x 本属癌变类而判属癌变类所造成的损失为 L_{22}.

根据条件风险的定义，将模式 x 判属正常类 ω_1 的条件风险为将模式 x 判属 ω_1 类所造成的损失的条件数学期望：

$$r_1(x)=L_{11}P(\omega_1/x)+L_{21}P(\omega_2/x)$$

同理，将模式 x 判属癌变类 ω_2 的条件风险为：

$$r_2(x)=L_{12}P(\omega_1/x)+L_{22}P(\omega_2/x)$$

我们可以根据条件风险的大小来决策。如果将 x 判属 ω_1 类的条件风险 $r_1(x)$ 小于判属 ω_2 类的条件风险 $r_2(x)$，则决策 x 属 ω_1 类；反之，如果将 x 判属 ω_1 类的条件风险 $r_1(x)$ 大于判属 ω_2 类的风险 $r_2(x)$，则决策 x 属 ω_2 类。即

若

$$L_{11}P(\omega_1/x)+L_{21}P(\omega_2/x)\lessgtr L_{12}P(\omega_1/x)+L_{22}P(\omega_2/x)$$

则

$$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

利用贝叶斯公式，上面的决策规则改写为：

若

$$L_{11}\frac{p(x/\omega_1)P(\omega_1)}{p(x)}+L_{21}\frac{p(x/\omega_2)P(\omega_2)}{p(x)}\lessgtr L_{12}\frac{p(x/\omega_1)P(\omega_1)}{p(x)}+L_{22}\frac{p(x/\omega_2)/P(\omega_2)}{p(x)}$$

消去 $p(x)$，简化为：

若

$$L_{11}p(x/\omega_1)P(\omega_1)+L_{21}p(x/\omega_2)P(\omega_2)\lessgtr L_{12}p(x/\omega_1)P(\omega_1)+L_{22}p(x/\omega_2)P(\omega_2)$$

则

$$x\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}\tag{2.3}$$

将两类情况推广到 c 类情况：

①后验概率形式

$$r_i(x)=\sum_{k=1}^{c}L_{ki}P(\omega_k/x)\tag{2.4}$$

②类条件概率密度形式

$$r_i(x)=\sum_{k=1}^{c}L_{ki}p(x/\omega_k)P(\omega_k)\tag{2.5}$$

决策规则为：

若 $$r_i(\boldsymbol{x})<r_j(\boldsymbol{x}),\quad j=1,2,\cdots,c;j\neq i$$

则 $$\boldsymbol{x}\in\omega_i \tag{2.6}$$

在这种决策中，我们对每一个 $\boldsymbol{x}$ 计算出全部类别的条件风险值 $r_1(\boldsymbol{x}),r_2(\boldsymbol{x}),\cdots,r_c(\boldsymbol{x})$，并且将 $\boldsymbol{x}$ 分到具有最小条件风险值的那一类，因此对所有的 $\boldsymbol{x}$ 作出决策时，其期望风险也必然最小。这样的决策就被称为最小风险贝叶斯决策。

例 2.2 在例 2.1 条件的基础上，令 $L_{11}=0,L_{21}=3,L_{12}=1,L_{22}=0$，按最小风险贝叶斯决策为王某诊断。

计算条件风险：

$$r_1(\boldsymbol{x})=L_{11}p(\boldsymbol{x}/\omega_1)P(\omega_1)+L_{21}p(\boldsymbol{x}/\omega_2)P(\omega_2)=0.013\ 25$$

$$r_2(\boldsymbol{x})=L_{12}p(\boldsymbol{x}/\omega_1)P(\omega_1)+L_{22}p(\boldsymbol{x}/\omega_2)P(\omega_2)=0.009\ 95$$

由于 $r_1(\boldsymbol{x})>r_2(\boldsymbol{x})$，所以 $\boldsymbol{x}\in\omega_2$，即王某属癌症病人。

将本例与例 2.1 相对比，分类结果正好相反。这是因为最小风险贝叶斯决策多考虑了一个因素，即损失。而且两种错判所造成的损失相差悬殊，导致不同的分类结果。从这个例子可以看出，采用最小风险贝叶斯决策，各种损失的确定很关键。一定要客观地分析错判所造成的严重程度，确定恰当的损失值。

上面我们介绍了两种贝叶斯决策。下面简单讨论一下这两种决策规则之间的关系。以两类问题为例加以分析。按照式(2.2)，最小错误率贝叶斯决策规则可写为：

若 $$\frac{p(\boldsymbol{x}/\omega_1)}{p(\boldsymbol{x}/\omega_2)}\gtrless\frac{P(\omega_2)}{P(\omega_1)}$$

则 $$\boldsymbol{x}\in\begin{matrix}\omega_1\\ \omega_2\end{matrix} \tag{2.7}$$

其中，$l(\boldsymbol{x})=\dfrac{p(\boldsymbol{x}/\omega_1)}{p(\boldsymbol{x}/\omega_2)}$ 在统计学中称为似然比，不等号右边的值称为似然比阈值。

按照式(2.3)，并假定错误决策总是比正确决策所造成的损失要大，即 $L_{12}>L_{11},L_{21}>L_{22}$，最小风险贝叶斯决策规则为：

若
$$\frac{p(\boldsymbol{x}/\omega_1)}{p(\boldsymbol{x}/\omega_2)} \gtrless \frac{(L_{21}-L_{22})P(\omega_2)}{(L_{12}-L_{11})P(\omega_1)}$$

则
$$\boldsymbol{x} \in \begin{matrix}\omega_1 \\ \omega_2\end{matrix} \tag{2.8}$$

比较式(2.7)与式(2.8)可知，这两种决策规则都采用似然比决策，区别仅在于取不同的似然比阈值。最小错误率贝叶斯决策的似然比阈值为 $P(\omega_2)/P(\omega_1)$，而最小风险贝叶斯决策的似然比阈值为 $(L_{21}-L_{22})P(\omega_2)/(L_{12}-L_{11})P(\omega_1)$。如果正确决策的损失均取为零，错误决策的损失均相等，即 $L_{11}=L_{22}=0$，$L_{12}=L_{21}$，则这两种决策是等价的，换句话说，最小错误率贝叶斯决策是最小风险贝叶斯决策的特例。

2.3 贝叶斯分类器的错误率

在分类器设计出来后，通常总是以错误率评价其性能。特别是当同一个分类问题设计出几种不同的分类方案时，通常总是以错误率作为方案比较的标准。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。

所谓错误率是指平均错误率，以 $P(e)$ 来表示，其定义为

$$P(e)=\int_{-\infty}^{\infty} P(e,\boldsymbol{x})\mathrm{d}\boldsymbol{x}=\int_{-\infty}^{\infty} P(e/\boldsymbol{x})P(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.9}$$

对于两类问题，整个模式空间划分为两个决策域 R_1 和 R_2。假设我们采用最小错误率贝叶斯决策。在决策域 R_1 中，$P(\omega_1/\boldsymbol{x})>(\omega_2/\boldsymbol{x})$，决策为 ω_1，显然在作出决策 ω_1 时，$\boldsymbol{x}$ 的条件错误概率为 $P(\omega_2/\boldsymbol{x})$。反之，则应为 $P(\omega_1/\boldsymbol{x})$。可表示为

$$P(e/\boldsymbol{x})=\begin{cases}P(\omega_2/\boldsymbol{x}), & \text{当 } P(\omega_1/\boldsymbol{x})>P(\omega_2/\boldsymbol{x}) \\ P(\omega_1/\boldsymbol{x}), & \text{当 } P(\omega_2/\boldsymbol{x})>P(\omega_1/\boldsymbol{x})\end{cases}$$

这样就有

$$P(e)=\int_{R_1} P(\omega_2/\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}+\int_{R_2} P(\omega_1/\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$$= \int_{R_1} p(x/\omega_2)P(\omega_2)\mathrm{d}x + \int_{R_2} p(x/\omega_1)P(\omega_1)\mathrm{d}x$$

$$= P(\omega_2)\int_{R_1} p(x/\omega_2)\mathrm{d}x + P(\omega_1)\int_{R_2} p(x/\omega_1)\mathrm{d}x$$

$$= P(\omega_2)P_2(e) + P(\omega_1)P_1(e) \qquad (2.10)$$

图 2－2 给出了一维情况的例子。图中 t 是决策边界，斜线面积为 $P(\omega_2)P_2(e)$，纹线面积为 $P(\omega_1)P_1(e)$，两者之和为 $P(e)$.

决策规则(2.1)实际上是对每个 x 都使 $P(e/x)$取小者，这就使式(2.9)的积分也必然达到最小，即使平均错误率 $P(e)$达到最小。这就说明了为什么这种决策称为最小错误率贝叶斯决策。

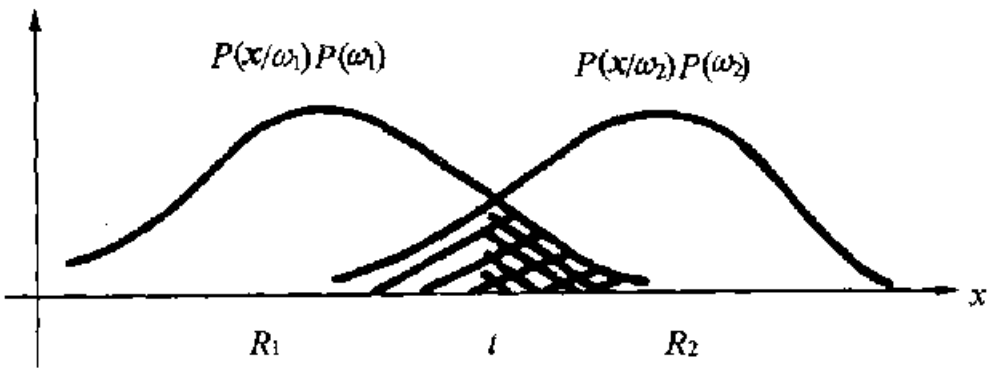


图 2－2

从式(2.9)可以看出，当 x 是多维向量时，实际上要进行多重积分的计算。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解析表达式又较复杂时，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计算错误率。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。

一、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

$$p(\boldsymbol{x}/\omega_1)=\frac{1}{(2\pi)^{\frac{n}{2}}|C|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_1)^{\mathrm{T}}C^{-1}(\boldsymbol{x}-\boldsymbol{m}_1)\right]$$

$$p(\boldsymbol{x}/\omega_2)=\frac{1}{(2\pi)^{\frac{n}{2}}|C|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_2)^{\mathrm{T}}C^{-1}(\boldsymbol{x}-\boldsymbol{m}_2)\right]$$

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

最小错误率贝叶斯决策规则为：

若 $$p(\boldsymbol{x}/\omega_1)P(\omega_1)\gtrless p(\boldsymbol{x}/\omega_2)P(\omega_2)$$

则 $$\boldsymbol{x}\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

这个决策规则可改写为：

若 $$\frac{p(\boldsymbol{x}/\omega_1)}{p(\boldsymbol{x}/\omega_2)}\gtrless\frac{P(\omega_2)}{P(\omega_1)}$$

则 $$\boldsymbol{x}\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

两边取负对数，进一步改写为：

若 $$-\ln p(\boldsymbol{x}/\omega_1)+\ln p(\boldsymbol{x}/\omega_2)\lessgtr\ln\frac{P(\omega_1)}{P(\omega_2)}$$

则 $$\boldsymbol{x}\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

令 $h(\boldsymbol{x})=-\ln p(\boldsymbol{x}/\omega_1)+\ln p(\boldsymbol{x}/\omega_2)$，称 $h(\boldsymbol{x})$ 为负对数似然比，$t=\ln\frac{P(\omega_1)}{P(\omega_2)}$

则决策规则简化为：

若 $$h(\boldsymbol{x})\lessgtr t$$

则 $$\boldsymbol{x}\in\begin{matrix}\omega_1\\ \omega_2\end{matrix}$$

$h(\boldsymbol{x})$ 是 $\boldsymbol{x}$ 的函数，$\boldsymbol{x}$ 是随机向量，因此 $h(\boldsymbol{x})$ 是随机变量。我们记它的分布密度函数为 $p(h/\omega_i)$。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

$$P_2(e)=\int_R p(\boldsymbol{x}/\omega_2)\mathrm{d}\boldsymbol{x}=\int_{-\infty}^{t}p(h/\omega_2)\mathrm{d}h$$

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

从上面两式看到，只要知道 $h(x)$密度函数的解析形式就可以算出 $P_1(e)$和 $P_2(e)$。根据我们所讨论的情况，$h(x)$可写为

$$\begin{aligned}h(x)&=-\ln p(x/\omega_1)+\ln p(x/\omega_2)\\&=-\left[-\frac{1}{2}(x-m_1)^{\mathrm{T}}C^{-1}(x-m_1)-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln|C|\right]\\&\quad+\left[-\frac{1}{2}(x-m_2)^{\mathrm{T}}C^{-1}(x-m_2)-\frac{n}{2}\ln 2\pi-\frac{1}{2}\ln|C|\right]\\&=(m_2-m_1)^{\mathrm{T}}C^{-1}x+\frac{1}{2}(m_1^{\mathrm{T}}C^{-1}m_1-m_2^{\mathrm{T}}C^{-1}m_2)\end{aligned}$$

从上式看出，$h(x)$是 x 的线性函数。x 是正态分布的随机向量，根据概率论的知识可知，$h(x)$服从一维正态分布，即 $p(h/\omega_1)$为 $N(\eta_1,\sigma_1)$，$p(h/\omega_1)$为 $N(\eta_2,\sigma_2)$.

$$\begin{aligned}\eta_1&=\mathrm{E}[h(x)/\omega_1]=(m_2-m_1)^{\mathrm{T}}C^{-1}m_1\\&\quad+\frac{1}{2}(m_1^{\mathrm{T}}C^{-1}m_1-m_2^{\mathrm{T}}C^{-1}m_2)\\&=-\frac{1}{2}(m_1-m_2)^{\mathrm{T}}C^{-1}(m_1-m_2)\end{aligned}$$

令 $\eta=\frac{1}{2}(m_1-m_2)^{\mathrm{T}}C^{-1}(m_1-m_2)$，则有

$\eta_1=-\eta$

$\sigma_1^2=\mathrm{E}[(h(x)-\eta_1)^2/\omega_1]=(m_1-m_2)^{\mathrm{T}}C^{-1}(m_1-m_2)=2\eta=\sigma^2$

$\eta_2=\mathrm{E}[h(x)/\omega_2]=\frac{1}{2}(m_1-m_2)^{\mathrm{T}}C^{-1}(m_1-m_2)=\eta$

$\sigma_2^2=\mathrm{E}[(h(x)-\eta_2)^2/\omega_2]=(m_1-m_2)^{\mathrm{T}}C^{-1}(m_1-m_2)=2\eta=\sigma^2$ 因此，可以利用 $p(h/\omega_1)$和 $p(h/\omega_2)$计算出 $P_1(e)$和 $P_2(e)$

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

$$=\int_{\left(\frac{t+\eta}{\sigma}\right)}^{\infty}(2\pi)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\xi^2\right)d\xi \tag{2.11}$$

$$P_2(e)=\int_{-\infty}^{t}p(h/\omega_2)dh$$

$$=\int_{-\infty}^{t}\frac{1}{(2\pi)^{\frac{1}{2}}\sigma}\exp\left[-\frac{1}{2}\left(\frac{h-\eta}{\sigma}\right)^2\right]dh$$

$$=\int_{-\infty}^{\left(\frac{t-\eta}{\sigma}\right)}(2\pi)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\xi^2\right)d\xi \tag{2.12}$$

式(2.11)和(2.12)的计算可以查标准正态 $N(0,1)$ 的累积分布函数表而得出。图 2-3 示出 $h(x)$ 的概率密度函数，阴影部分相当于最小错误率贝叶斯决策的错误率。

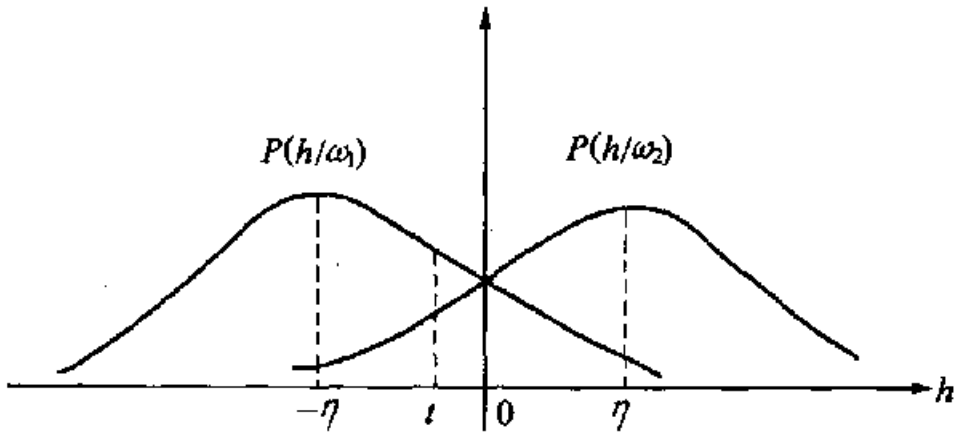


图 2-3

二、错误率的估计

如果先验概率 $P(\omega_i)$ 未知，我们可简单地随机抽取 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

$$\hat{\varepsilon}=\frac{t}{N}$$

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。这样，上式的估计是不是最好

的？回答是肯定的。下面作一简单说明。

设 ε 是真实的错误率。在给定 ε 后，τ 的密度函数为二项分布

$$P(\tau/\varepsilon) = C_N^{\tau}\varepsilon^{\tau}(1-\varepsilon)^{N-\tau}$$

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

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

解此方程得

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

这说明 τ/N 是错误率 ε 的最大似然估计 $\hat{\varepsilon}$.

如果先验概率 $P(\omega_i)$ 已知，对于两类情况，可分别从类别 ω_1 和 ω_2 总体中抽取 $N_1 = P(\omega_1)N$ 和 $N_2 = P(\omega_2)N$ 个样本，对给定的分类器作分类检验。假设 N_1 个样本被错分了 τ_1 个，N_2 个样本被错分了 τ_2 个，因 τ_1 与 τ_2 是相互独立的，故 τ_1 和 τ_2 的联合概率为

$$P(\tau_1,\tau_2) = P(\tau_1)P(\tau_2) = \prod_{i=1}^{2} C_{N_i}^{\tau_i}\varepsilon_i^{\tau_i}(1-\varepsilon_i)^{N_i-\tau_i}$$

其中 ε_i 为 ω_i 类的真实错误率。

同样可求得 ε_i 的最大似然估计为

$$\hat{\varepsilon}_i = \frac{\tau_i}{N_i}, \quad i = 1,2$$

而总的错误率估计为

$$\hat{\varepsilon}' = P(\omega_1)\hat{\varepsilon}_1 + P(\omega_2)\hat{\varepsilon}_2 = \sum_{i=1}^{2} P(\omega_i)\frac{\tau_i}{N_i}$$

通过上面的讨论可以看出，这些估计是在最大似然估计意义上最好的估计。

2.4 聂曼-皮尔逊决策

采用最小错误率贝叶斯决策需要知道先验概率 $P(\omega_i)$，但有时 $P(\omega_i)$ 难以确定。采用最小风险贝叶斯决策需要确定恰当的损

失值，这也并非易事。在两类问题决策中，有时要求 $P_2(e)$ 不得大于某个常数，即取 $P_2(e)=\varepsilon_0$，ε_0 是一个很小的常数，在这个条件下再要求 $P_1(e)$ 尽可能小。在这种情况下，聂曼－皮尔逊决策提供了一种决策方案。

这种决策可看成是在 $P_2(e)=\varepsilon_0$ 条件下，求 $P_1(e)$ 极小值的条件极值问题。可采用拉格朗日乘子法求解条件极值问题，按拉格朗日乘子法建立数学模型为

$$\gamma = P_1(e) + \mu(P_2(e) - \varepsilon_0) \tag{2.13}$$

其中 μ 是拉格朗日乘子，目的是求 γ 的极小值。

从式(2.10)可知

$$P_1(e)=\int_{R_2} p(\boldsymbol{x}/\omega_1)\mathrm{d}\boldsymbol{x} \tag{2.14}$$

$$P_2(e) = \int_{R_1} p(\boldsymbol{x}/\omega_2)\mathrm{d}\boldsymbol{x} = \varepsilon_0 \tag{2.15}$$

根据类条件概率密度的性质，有

$$\int_{R_2} p(\boldsymbol{x}/\omega_1)\mathrm{d}\boldsymbol{x} = 1 - \int_{R_1} p(\boldsymbol{x}/\omega_1)\mathrm{d}\boldsymbol{x} \tag{2.16}$$

将式(2.14)、(2.15)、(2.16)代入式(2.13)，得

$$\gamma = (1 - \mu\varepsilon_0) + \int_{R1} [\mu p(\boldsymbol{x}/\omega_2) - p(\boldsymbol{x}/\omega_1)]\mathrm{d}\boldsymbol{x} \tag{2.17}$$

式中只有 R_1 是变量，要使 γ 最小，积分项应取负值，即在 R_1 区域内应使

$$\mu p(\boldsymbol{x}/\omega_2) < p(\boldsymbol{x}/\omega_1)$$

此时决策规则可写为

若 $$\frac{p(\boldsymbol{x}/\omega_1)}{p(\boldsymbol{x}/\omega_2)} > \mu$$

则 $$\boldsymbol{x} \in \omega_1 \tag{2.18}$$

同理可求得

$$\gamma = (1 - \varepsilon_0)\mu + \int_{R_2} [p(\boldsymbol{x}/\omega_1) - \mu p(\boldsymbol{x}/\omega_2)]\mathrm{d}\boldsymbol{x}$$

即确定 R_2 区域应有

$$p(x/\omega_1) < \mu p(x/\omega_2)$$

决策规则为：

若 $$\frac{p(x/\omega_1)}{p(x/\omega_2)} < \mu$$

则 $$x \in \omega_2 \tag{2.19}$$

将式(2.18)与(2.19)综合起来得到：

若 $$\frac{p(x/\omega_1)}{p(x/\omega_2)} \gtrless \mu$$

则 $$x \in \begin{matrix}\omega_1 \\ \omega_2\end{matrix} \tag{2.20}$$

这种决策称为聂曼-皮尔逊决策。将式(2.20)与式(2.7)、(2.8)对比，可以看出聂曼-皮尔逊决策与前面介绍的两种贝叶斯规则都是以似然比为基础的，所不同的只是聂曼-皮尔逊决策用的阈值为拉格朗日乘子 μ.

如果要采用聂曼-皮尔逊决策，须确定 μ 值。将式(2.17)对 x 求导，并令

$$\frac{\partial \gamma}{\partial x} = 0$$

得

$$\mu = \frac{p(x/\omega_1)}{p(x/\omega_2)} \tag{2.21}$$

满足(2.21)式的最佳 μ 及满足(2.15)式的已知条件就能使 γ 极小。故 μ 是(2.21)和(2.15)方程的解。

例 2.3 一个两类问题，模式分布为二维正态，其分布参数 $m_1 = (-1,0)^T$，$m_2 = (1,0)^T$，$C_1 = C_2 = I$.

假定 $P_2(e) = \epsilon_0 = 0.046$，求聂曼-皮尔逊决策阈值。

解

$$p(x/\omega_1) = \frac{1}{2\pi}\exp\left[-\frac{1}{2}(x-m_1)^T(x-m_1)\right]$$

$$= \frac{1}{2\pi}\exp\left[-\frac{1}{2}((x_1+1)^2+x_2^2)\right]$$

$$p(\boldsymbol{x}/\omega_2)=\frac{1}{2\pi}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_2)^T(\boldsymbol{x}-\boldsymbol{m}_2)\right]$$

$$=\frac{1}{2\pi}\exp\left[-\frac{1}{2}((x_1-1)^2+x_2^2)\right]$$

所以

$$\frac{p(\boldsymbol{x}/\omega_1)}{p(\boldsymbol{x}/\omega_2)}=\exp(-2x_1)$$

故得决策边界为

$$\mu=\exp(-2x_1)$$

即

$$x_1=-\frac{1}{2}\ln\mu$$

可见，这个决策边界是一条直线，这条直线将模式空间分割为两个决策域 R_1 和 R_2，如图 2－4 所示。

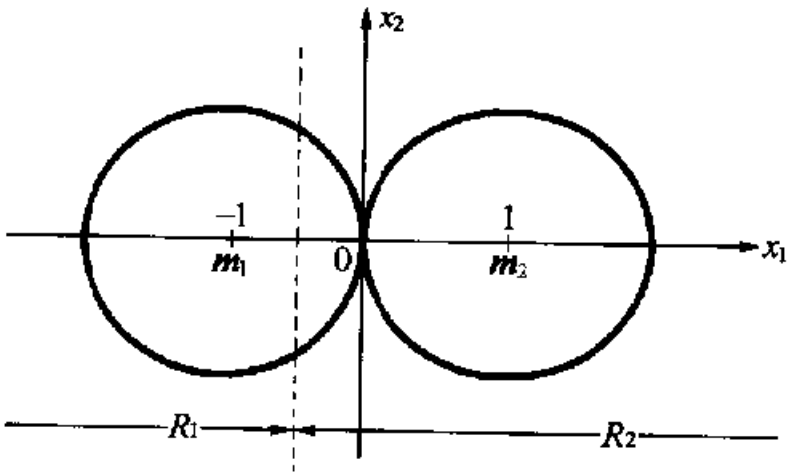


图 2－4

知道了决策域 R_1 就可计算错误率 $P_2(e)$：

$$P_2(e)=\int_{R_1}p(\boldsymbol{x}/\omega_2)\mathrm{d}\boldsymbol{x}$$

$$=\int_{-\infty}^{-\frac{1}{2}\ln\mu}\int_{-\infty}^{\infty}\frac{1}{2\pi}\exp\left[-\frac{(x_1-1)^2+x_2^2}{2}\right]\mathrm{d}x_1\mathrm{d}x_2$$

$$=\int_{-\infty}^{-\frac{1}{2}\ln\mu}\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{(x_1-1)^2}{2}\right]\mathrm{d}x_1$$

在不同的 μ 值时，可求得 $P_2(e)$ 的值，见表 2－1。

表 2-1

μ	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$
$P_2(e)$	0.046	0.089	0.015 9	0.258	0.378

对于给定的 ϵ_2 值，查表可找到相应的 μ 值。如本例，$P_2(e)=\epsilon_0=0.046$，查表可得 $\mu=4$。

2.5 均值向量和协方差矩阵的估计

前面几节分析了贝叶斯分类器，构造这些分类器都需要知道类条件概率密度 $p(\boldsymbol{x}/\omega_i)$，如果已知概率密度的形式，不知道该分布的参数，那么只要估计出来这些参数就行了，这是参数估计的问题。例如已知模式服从正态分布，那么只需要估计出均值向量和协方差矩阵。

设模式 $\boldsymbol{x}$ 的概率密度为 $p(\boldsymbol{x})$，则其均值向量定义为

$$\boldsymbol{m}=\mathrm{E}[\boldsymbol{x}]=\int_{\boldsymbol{x}}\boldsymbol{x}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

式中 $\boldsymbol{x}=(x_1,x_2\cdots,x_n)^T$，$\boldsymbol{m}=(m_1,m_2,\cdots,m_n)^T$。

如果用样本的平均值作为均值向量的近似值，则均值向量的估计量为

$$\hat{\boldsymbol{m}}=\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{x}^j$$

式中 N 为样本数目，$\boldsymbol{x}^j$ 是第 j 个样本。

协方差矩阵为

$$C=\begin{pmatrix}c_{11} & c_{12} & \cdots & c_{1n}\\ c_{21} & c_{22} & \cdots & c_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ c_{n1} & c_{n2} & \cdots & c_{nn}\end{pmatrix}$$

其中元素 c_{ik} 定义为

$$c_{lk}=\mathrm{E}[(x_l-m_l)(x_k-m_k)^T]$$
$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x_l-m_l)(x_k-m_k)p(x_l,x_k)\mathrm{d}x_l\mathrm{d}x_k$$

式中，x_l,x_k 和 m_l,m_k 分别为 $\boldsymbol{x}$ 和 $\boldsymbol{m}$ 的第 l、k 个分量。协方差矩阵写成向量形式为

$$C=\mathrm{E}[(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^T]$$
$$=\mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T-2\boldsymbol{x}\boldsymbol{m}^T+\boldsymbol{m}\boldsymbol{m}^T]$$
$$=\mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T]-\boldsymbol{m}\boldsymbol{m}^T$$

协方差矩阵的估计量为

$$\hat{C}\approx\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{x}^j(\boldsymbol{x}^j)^T-\boldsymbol{m}\boldsymbol{m}^T$$

下面介绍均值向量和协方差矩阵的估计量的迭代运算。

假设我们已经计算了 N 个样本的均值向量估计量。如果再加一个样本，则其新的估计量 $\hat{\boldsymbol{m}}(N+1)$ 为

$$\hat{\boldsymbol{m}}(N+1)=\frac{1}{N+1}\sum_{j=1}^{N+1}\boldsymbol{x}^j=\frac{1}{N+1}[\sum_{j=1}^{N}\boldsymbol{x}^j+\boldsymbol{x}^{N+1}]$$
$$=\frac{1}{N+1}[N\hat{\boldsymbol{m}}(N)+\boldsymbol{x}^{N+1}]$$

式中 $\hat{\boldsymbol{m}}(N)$ 为由 N 个样本算得的估计量。估计过程从 $\boldsymbol{m}(1)=\boldsymbol{x}^1$ 开始。

协方差矩阵估计量的迭代运算与上述类似。假设我们已经计算了 N 个样本的协方差矩阵估计量：

$$\hat{C}(N)=\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{x}^j(\boldsymbol{x}^j)^T-\hat{\boldsymbol{m}}(N)\hat{\boldsymbol{m}}^T(N)$$

增加一个样本，则

$$\hat{C}(N+1)=\frac{1}{N+1}\sum_{j=1}^{N+1}\boldsymbol{x}^j(\boldsymbol{x}^j)^T-\hat{\boldsymbol{m}}(N+1)\hat{\boldsymbol{m}}^T(N+1)$$
$$=\frac{1}{N+1}[\sum_{j=1}^{N}\boldsymbol{x}^j(\boldsymbol{x}^j)^T+\boldsymbol{x}^{N+1}(\boldsymbol{x}^{N+1})^T]$$
$$-\hat{\boldsymbol{m}}(N+1)\hat{\boldsymbol{m}}^T(N+1)$$

$$=\frac{1}{N+1}[\hat{N}\hat{C}(N)+N\hat{\boldsymbol{m}}(N)\boldsymbol{m}^T(N)+\boldsymbol{x}^{N+1}(\boldsymbol{x}^{N+1})^T]$$

$$-\frac{1}{(N+1)^2}[N\hat{\boldsymbol{m}}(N)+\boldsymbol{x}^{N+1}][N\hat{\boldsymbol{m}}(N)+\boldsymbol{x}^{N+1}]^T$$

估计过程从 $\hat{C}(1)=0$ 和 $\hat{\boldsymbol{m}}(1)=\boldsymbol{x}^1$ 开始。

2.6 概率密度的函数逼近

如果概率密度的形式不知道，那就必须直接估计概率密度。

令 $\hat{p}(\boldsymbol{x})$表示对 $p(\boldsymbol{x})$的估计。假如我们将 $\hat{p}(\boldsymbol{x})$用级数形式展开

$$\hat{p}(\boldsymbol{x})=\sum_{j=1}^{m}c_j\varphi_j(\boldsymbol{x}) \tag{2.22}$$

式中，c_j 是待定系数，$\{\varphi_j(\boldsymbol{x})\}$是选定的基函数集，它应是正交函数。付里叶级数展开就是式(2.22)的一个特例，其基函数为正弦和余弦函数系。

我们希望在均方误差函数最小的情况下得到估计 $\hat{p}(\boldsymbol{x})$。均方误差函数定义为

$$R=\int_x u(\boldsymbol{x})[p(\boldsymbol{x})-\hat{p}(\boldsymbol{x})]^2\mathrm{d}\boldsymbol{x} \tag{2.23}$$

式中 $u(\boldsymbol{x})$是权函数。将式(2.22)代入式(2.23)得

$$R=\int_x u(\boldsymbol{x})[p(\boldsymbol{x})-\sum_{j=1}^{m}c_j(\boldsymbol{x})]^2\mathrm{d}\boldsymbol{x}$$

希望确定系数 c_j 使得均方误差函数 R 最小。因此，必须使得

$$\frac{\partial R}{\partial c_k}=0,\qquad k=1,2,\cdots,m$$

将 R 偏微分后得

$$\sum_{j=1}^{m}c_j\int_x u(\boldsymbol{x})\varphi_j(\boldsymbol{x})\varphi_k(\boldsymbol{x})\mathrm{d}\boldsymbol{x}=\int_x u(\boldsymbol{x})\varphi_k(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

式中右边是函数 $u(\boldsymbol{x})\varphi_k(\boldsymbol{x})$的数学期望，可用样本均值来近似，得

$$\sum_{j=1}^{m} c_j \int_x u(x)\varphi_j(x)\varphi_k(x)\mathrm{d}x = \frac{1}{N}\sum_{i=1}^{N} u(x^i)\varphi_k(x^i) \quad (2.24)$$

所选的基函数$\{\varphi_k(x)\}$对权函数$u(x)$条件下正交，正交定义为

$$\int_x u(x)\varphi_j(x)\varphi_k(x)\mathrm{d}x = \begin{cases} A_k, & \text{若 } j = k \\ 0, & \text{若 } j \neq k \end{cases}$$

将上式代入式(2.24)，可求得系数c_k

$$c_k = \frac{1}{NA_k}\sum_{i=1}^{N} u(x^i)\varphi_k(x^i), \quad k = 1,2,\cdots,m$$

如果基函数$\{\varphi_k(x)\}$正交归一，则对于全部k，有$A_k=1$. 由于$u(x^i)$与k无关，因而常被略掉，这样得

$$c_k = \frac{1}{N}\sum_{i=1}^{N} \varphi_k(x^i), \quad k = 1,2,\cdots,m$$

一旦系数c_k求得后，就可得到估计概率密度$\hat{p}(x)$.

为了方便，将上式c_k写成迭代式。用$c_k(N)$表示N个样本求得的系数。如果再增加一个样本，c_k就写成

$$\begin{aligned} c_k(N+1) &= \frac{1}{N+1}\sum_{i=1}^{N+1} \varphi_k(x^i) \\ &= \frac{1}{N+1}[Nc_k(N) + \varphi_k(x^{N+1})] \end{aligned}$$

式中，$c_k(1)=\varphi_k(x^1)$. 利用迭代形式可简化c_k的计算。

应用上面介绍的函数逼近来估计概率密度，有二点需要考虑。

第一点，基函数的选择很重要。基函数必须是线性独立的。正交性是线性独立的一种特殊情况。基函数的线性独立区域必须覆盖模式空间。如果有$p(x)$的先验知识，有助于选择更合适的基函数。然而，常常没有$p(x)$的先验知识，都只好试探性地选择满足基函数基本要求的容易实现的函数作为基函数。

第二点，项数m的确定。由于我们并不知道$p(x)$，就不能用直接比较的方法来评价$\hat{p}(x)$对$p(x)$的逼近程度，因此可以用训练样本来试验，如果用$\hat{p}(x)$设计的分类器性能不佳，则可增加项数，然后观察分类器性能是否有所改善，如果再增加项数，分类器

性能提高很小，或者项数多到不能接受的程度，项数就不要再增加了。

例 2.4　已知训练样本为

$$\omega_1:\quad \boldsymbol{x}^{11}=\begin{pmatrix}3\\4\end{pmatrix},\boldsymbol{x}^{12}=\begin{pmatrix}4\\4\end{pmatrix},\boldsymbol{x}^{13}=\begin{pmatrix}4\\3\end{pmatrix},\boldsymbol{x}^{14}=\begin{pmatrix}3\\3\end{pmatrix},$$

$$\boldsymbol{x}^{15}=\begin{pmatrix}3\\2\end{pmatrix},\boldsymbol{x}^{16}=\begin{pmatrix}2\\3\end{pmatrix},\boldsymbol{x}^{17}=\begin{pmatrix}2\\2\end{pmatrix}$$

$$\omega_2:\quad \boldsymbol{x}^{21}=\begin{pmatrix}-3\\-2\end{pmatrix},\boldsymbol{x}^{22}=\begin{pmatrix}-2\\-3\end{pmatrix},\boldsymbol{x}^{23}=\begin{pmatrix}-3\\-3\end{pmatrix},$$

$$\boldsymbol{x}^{24}=\begin{pmatrix}-4\\-3\end{pmatrix},\boldsymbol{x}^{25}=\begin{pmatrix}-3\\-4\end{pmatrix},\boldsymbol{x}^{26}=\begin{pmatrix}-3\\-5\end{pmatrix}$$

试通过训练样本直接估计类条件概率密度，并由此设计贝叶斯分类器。

解　我们用 4 项基函数的多项式来逼近类条件概率密度

$$\hat{p}(\boldsymbol{x}/\omega_1)=\sum_{k=1}^{4}c_{1k}\varphi_k(\boldsymbol{x})$$

$$\hat{p}(\boldsymbol{x}/\omega_2)=\sum_{k=1}^{4}c_{2k}\varphi_k(\boldsymbol{x})$$

所选取的基函数必须在模式的定义域内正交。考虑到埃尔米特函数的正交域为$(-\infty,\infty)$，用起来比较方便，所以我们选用埃尔米特函数作为基函数。该函数一维形式的前几项为

$$H_0(x)=1,\qquad H_1(x)=2x$$

$$H_2(x)=4x^2-2,\qquad H_3(x)=8x^3-12x$$

该函数的四个最低阶的二维正交集为

$$\varphi_1(\boldsymbol{x})=\varphi_1(x_1,x_2)=H_0(x_1)H_0(x_2)=1$$

$$\varphi_2(\boldsymbol{x})=\varphi_2(x_1,x_2)=H_1(x_1)H_0(x_2)=2x_1$$

$$\varphi_3(\boldsymbol{x})=\varphi_3(x_1,x_2)=H_0(x_1)H_1(x_2)=2x_2$$

$$\varphi_4(\boldsymbol{x})=\varphi_4(x_1,x_2)=H_1(x_1)H_2(x_2)=4x_1x_2$$

对于 ω_1 类

$$c_{1k}=\frac{1}{N_1}\sum_{j=1}^{N_1}\varphi_k(\boldsymbol{x}^{1j}),\quad k=1,2,3,4$$

$$c_{11}=\frac{1}{7}[\varphi_1(\boldsymbol{x}^{11})+\varphi_1(\boldsymbol{x}^{12})+\cdots+\varphi_1(\boldsymbol{x}^{17})]=1$$

$$c_{12}=\frac{1}{7}[\varphi_2(x^{11})+\varphi_2(x^{12})+\cdots+\varphi_2(x^{17})]$$

$$=\frac{1}{7}[6+8+8+6+6+4+4]=6$$

同理可得

$$c_{13}=\frac{1}{7}[\varphi_3(x^{11})+\varphi_3(x^{12})+\cdots+\varphi_3(x^{17})]=6$$

$$c_{14}=\frac{1}{7}[\varphi_4(x^{11})+\varphi_4(x^{12})+\cdots+\varphi_4(x^{17})]=37.1$$

同理可求得 $c_{21}=1$，$c_{22}=-6$，$c_{23}=-6.7$，$c_{24}=40$.

因此

$$\hat{p}(x/\omega_1)=c_{11}\varphi_1(x)+c_{12}\varphi_2(x)+c_{13}\varphi_3(x)+c_{14}\varphi_4(x)$$
$$=1+12x_1+12x_2+148.4x_1x_2$$

$$\hat{p}(x/\omega_2)=c_{21}\varphi_1(x)+c_{22}\varphi_2(x)+c_{23}\varphi_3(x)+c_{24}\varphi_4(x)$$
$$=1-12x_1-13.4x_2+160x_1x_2$$

本例题的判别函数为

$$d_1(x)=\hat{p}(x/\omega_1)P(\omega_1)$$
$$d_2(x)=\hat{p}(x/\omega_2)P(\omega_2)$$

设 $P(\omega_1)=P(\omega_2)=\frac{1}{2}$，得

$$d_1(x)=0.5+6x_1+6x_2+74.2x_1x_2$$
$$d_2(x)=0.5-6x_1-6.7x_2+80x_1x_2$$

显然这是一个非线性分类器。如果不采用非线性项 $\varphi_4(x)$，那么将得到的判别函数是线性函数，分类器就成了线性分类器。按照先易后难的原则，先用线性函数逼近，只有当分类效果不好时，才逐步增加非线性项以提高逼近程度。

2.7 正态分布模式的贝叶斯分类器

模式可能服从各种各样的分布，为什么要单列一节专门讨论正态分布的贝叶斯分类器呢？有两个目的。正态分布是常见的分

布。由于正态密度函数易于分析，也常常把一些分布近似为正态分布，所以再重点讨论一下正态分布的贝叶斯分类器。另外，通过分析正态分布中的一些特定情况，把贝叶斯分类器与后面要讨论的线性分类器、非线性分类器等联系起来，分析它们之间的内在联系。

假设模式 $\boldsymbol{x}$ 服从正态分布，其类条件概率密度函数为

$$p(\boldsymbol{x}/\omega_i)=\frac{1}{(2\pi)^{\frac{n}{2}}|C_i|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T C_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right] \tag{2.25}$$

式中，n 为 $\boldsymbol{x}$ 的维数，$\boldsymbol{m}_i$ 和 C_i 分别为 ω_i 类 $\boldsymbol{x}$ 的期望向量和协方差矩阵，其定义为

$$\boldsymbol{m}_i=\mathrm{E}[\boldsymbol{x}/\omega_i]$$

$$C_i=\mathrm{E}[(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^T/\omega_i]$$

协方差矩阵 C_i 是对称的正定矩阵，其对角线上元素 c_{kk} 是 $\boldsymbol{x}$ 第 k 个元素的方差，非对角线上元素 c_{jk} 是 $\boldsymbol{x}$ 的第 j 和第 k 个元素的协方差。$|C_i|$ 是矩阵 C_i 的行列式。

ω_i 类的最小错误率贝叶斯判别函数为

$$d_i(\boldsymbol{x})=p(\boldsymbol{x}/\omega_i)P(\omega_i)$$

对于正态密度函数，取自然对数的形式更为方便，即

$$d_i(\boldsymbol{x})=\ln[p(\boldsymbol{x}/\omega_i)]+\ln P(\omega_i)$$

因为对数是单调递增函数，取对数后分类性能不变。将式(2.25)代入上式，得

$$d_i(\boldsymbol{x})=-\frac{n}{2}\ln(2\pi)-\frac{1}{2}\ln|C_i|-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T C_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)+\ln P(\omega_i)$$

去掉与 i 值无关的项并不影响分类判别，故可简化为

$$d_i(\boldsymbol{x})=-\frac{1}{2}\ln|C_i|-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T C_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)+\ln P(\omega_i) \tag{2.26}$$

这就是正态分布模式的最小错误率贝叶斯判别函数。上式表明，$d_i(\boldsymbol{x})$是超二次曲面。所以对于正态分布模式的贝叶斯分类器，两个模式类别之间用一个二次决策边界分开，就可以达到最优的分类效果。

下面分几种特殊情况作进一步讨论。

(1)各类协方差矩阵相等，即 $C_i=C$

由式(2.26)可得

$$d_i(\boldsymbol{x}) = -\frac{1}{2}\ln|C| - \frac{1}{2}\boldsymbol{x}^T C^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T C^{-1}\boldsymbol{m}_i + \frac{1}{2}\boldsymbol{m}_i^T C^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{m}_i^T C^{-1}\boldsymbol{m}_i + \ln P(\omega_i)$$

因 C 为对称矩阵，并去掉与 i 无关的项，上式可简化为

$$d_i(\boldsymbol{x}) = \boldsymbol{m}_i^T C^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{m}_i^T C^{-1}\boldsymbol{m}_i + \ln P(\omega_i) \qquad (2.27)$$

上式表明，$d_i(\boldsymbol{x})$是线性判别函数，决策边界一定是超平面，由$d_i(\boldsymbol{x})$构成的分类器成了线性分类器。这里，尽管 $d_i(\boldsymbol{x})$是简单的线性函数，分类效果却是最优的。

(2)各类协方差矩阵相等，而且各类先验概率也相等，即 $C_i=C$，$P(\omega_i)=P(\omega)$

由式(2.26)得

$$d_i(\boldsymbol{x}) = -\frac{1}{2}\ln|C| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T C^{-1}(\boldsymbol{x}-\boldsymbol{m}_i) + \ln P(\omega)$$

去掉与 i 无关的项，判别函数可简化为

$$d_i(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{m}_i)^T C^{-1}(\boldsymbol{x}-\boldsymbol{m}_i) \qquad (2.28)$$

式中$(\boldsymbol{x}-\boldsymbol{m}_i)^T C^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)$称为马氏距离。这时决策规则变为：如果 $\boldsymbol{x}$ 到期望向量 $\boldsymbol{m}_i$ 的马氏距离最小，则将 $\boldsymbol{x}$ 分到 ω_i 类去。

(3)各类协方差矩阵都为 $\sigma^2\,\mathrm{I}$，而且各类先验概率相等，即 $P(\omega_i)=P(\omega)$.

由式(2.28)可得

$$d_i(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{m}_i)^T \frac{1}{\sigma^2} I (\boldsymbol{x}-\boldsymbol{m}_i)$$

进一步简化为

$$d_i(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{m}_i)^T(\boldsymbol{x}-\boldsymbol{m}_i) = \| \boldsymbol{x}-\boldsymbol{m}_i \|^2 \quad (2.29)$$

式中 $\| \boldsymbol{x}-\boldsymbol{m}_i \|$ 为 $\boldsymbol{x}$ 到 ω_i 类的均值向量 $\boldsymbol{m}_i$ 的欧氏距离。这时决策规则变为：如果 $\boldsymbol{x}$ 到期望向量 $\boldsymbol{m}_i$ 的欧氏距离最小，则将 $\boldsymbol{x}$ 分到 ω_i 类去。这种分类器称为最小距离分类器。

在这里，按照最小距离分类就能达到最优的分类效果，令人兴奋不已。别忘了只有在上面所假定的条件下才有最优效果。不过，式(2.29)判别函数为我们设计最小距离分类器提供了启示。

第三章　线性分类器

第二章我们讨论了贝叶斯分类器。贝叶斯判别函数中的类条件概率密度是利用样本估计的，估计出来的类条件概率密度函数可能是线性函数，也可能是各种各样的非线性函数。这种设计判别函数的思路，在用样本估计之前，是不知道判别函数是线性函数还是别的什么函数的。我们可以换一种设计判别函数的思路，设计判别函数时，首先确定判别函数为某种函数，比如为线性函数，然后利用样本集估计判别函数中的未知参数。如何估计这些未知参数，应针对不同的实际情况，提出不同的设计要求，使得所设计的分类器尽可能好地满足这些要求。当然，由于所提要求不同，结果也将各异，这说明上述"尽可能好"是相对于所提要求而言的。这种设计要求，往往用某个特定的函数来表达，称之为准则函数。"尽可能好"的结果相应于准则函数取最优值。这样，分类器设计问题就转化为求准则函数极值的问题了，因此就可以利用最优化技术解决模式识别问题。

实际上，设计贝叶斯分类器时，我们已经采用了准则函数，所用的准则是错误率或风险。贝叶斯分类器的错误率或风险是最小的，所以通常称之为最优分类器，而在其它准则函数下得到的分类器则称为是"次优"的。这里的"次优"，只是相对于错误率或风险而言的，而对所提的准则函数来讲，则是最好。虽然采用线性判别函数导致的错误率或风险可能比贝叶斯分类器大，但是，线性函数是最简单的函数，计算量小，容易实现。因此，线性判别函数是统计模式识别的基本方法之一。

3.1 线性判别函数的基本概念

假设模式向量 $\boldsymbol{x}$ 是 n 维的，线性判别函数的一般形式为

$$\begin{aligned} d(\boldsymbol{x}) &= w_1x_1 + w_2x_2 + \cdots + w_nx_n + w_{n+1} \\ &= \boldsymbol{w}_0^T\boldsymbol{x} + w_{n+1} \\ &= \boldsymbol{w}^T\boldsymbol{X} \end{aligned}$$

式中 $\boldsymbol{w}_0 = (w_1, w_2, \cdots w_n)^T$，称为权向量；

w_{n+1} 称为阈值；

$\boldsymbol{w} = (w_1, w_2, \cdots w_n, w_{n+1})^T$ 称为增广权向量；

$\boldsymbol{X} = (x_1, x_2 \cdots x_n, 1)^T$ 称为增广模式向量。

在两类情况下，可以仅定义一个判别函数：

$$d(\boldsymbol{x}) = d_1(\boldsymbol{x}) - d_2(\boldsymbol{x})$$

决策规则为：

若 $$d(\boldsymbol{x}) \gtrless 0$$

则 $$\boldsymbol{x} \in \begin{matrix} \omega_1 \\ \omega_2 \end{matrix}$$

决策边界方程为 $d(\boldsymbol{x}) = 0$，当 $d(\boldsymbol{x})$ 为线性函数时，这个决策边界便是超平面。这个超平面将模式空间分割成两个决策域。超平面的方向由权向量 $\boldsymbol{w}_0$ 确定，它的位置由阈值 w_{n+1} 确定。

在多类情况下，比如 c 类，定义多少个判别函数，通常有三种方案。

方案一：用线性判别函数将属于 ω_i 类的模式与其余不属于 ω_i 类的模式分开，c 类问题要有 c 个判别函数：$d_i(\boldsymbol{x}), i = 1, 2, \cdots c$. 决策规则为：

若 $$d_i(\boldsymbol{x}) > 0, d_j(\boldsymbol{x}) < 0, \quad j \neq i$$

则 $$\boldsymbol{x} \in \omega_i$$

采用这种方案，模式空间中可能存在不确定区域，如图 3－1 (a) 中的阴影区域。不确定区域中的模式无法确定其类别。

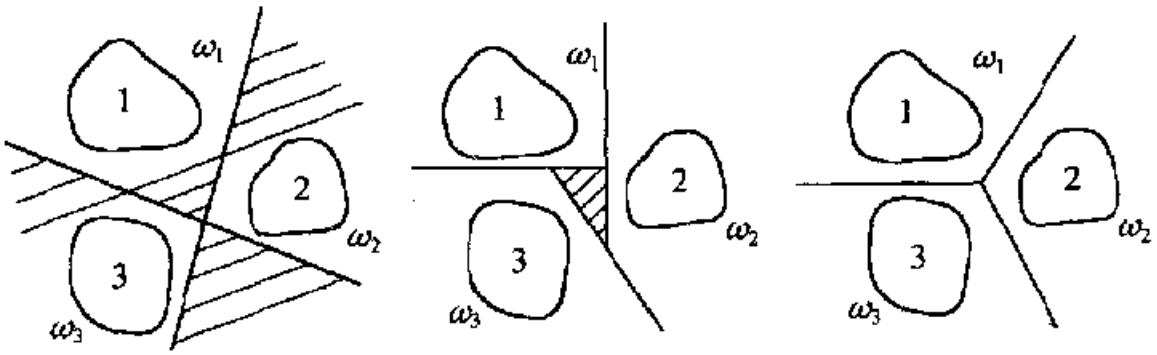


图 3-1

方案二：用线性判别函数将属于 ω_i 类的模式与 ω_j 类的模式分开，c 类问题要有 $c(c-1)/2$ 个判别函数：$d_{ij}(\boldsymbol{x})$，　$i,j=1,2,\cdots c$；$i\neq j$. 决策规则为：

若　　$d_{ij}(\boldsymbol{x})>0$，　$j=1,2,\cdots,c$；$j\neq i$

则　　$\boldsymbol{x}\in\omega_i$

采用这种方案，模式空间存在不确定区域。如图 3-1(*b*)中的阴影区域。不确定区域中的模式无法确定其类别。

方案三：c 类问题定义 c 个判别函数：$d_i(x)$，$i=1,2,\cdots c$. 决策规则为：

若　　$d_i(x)>d_j(x)$，　$j=1,2,\cdots c$；$j\neq i$

则　　$\boldsymbol{x}\in\omega_i$

采用这种方案，模式空间不存在不确定区域。如图 3-1(*c*)所示。处理多类问题时通常采用这种方案。

下面解释一个基本概念：线性可分性。

假设已知一组容量为 N 的样本集，如果有一个线性分类器能把每个样本正确分类，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，则必然存在一个线性分类器能把每个样本正确分类。

3.2 最小距离分类器

最小距离分类器在上一章就提到了。如果模式 x 服从正态分布，各类模式的协方差矩阵均为 $\sigma^2 I$，而且各类先验概率相等，若要对模式 x 分类，只要计算 x 到各类均值 m_i 的欧氏距离平方 $\| x - m_i \|^2$，然后把 x 分到距离最小的那类去即可。这一分类方法，虽然是在十分特殊的条件下推出来的，但却给了我们一个重要的启示，这就是可以把均值作为各类的代表，用距离作为判别函数进行分类。

设有 c 类已知类别的模式(样本)。从每类中找出一个标准样本，或者样本均值：$m_1, m_2, \cdots m_c$，定义判别函数为

$$d_i(x) = \| x - m_i \|^2, \quad i = 1, 2, \cdots, c \tag{3.1}$$

按最小距离分类原理，决策规则为：

若 $$d_i(x) < d_j(x), \quad j = 1, 2, \cdots c; j \neq i$$

则 $$x \in \omega_i$$

式(3.1)可改写为

$$\begin{aligned} d_i(x) &= (x - m_i)^T (x - m_i) \\ &= x^T x - 2 m_i^T x + m_i^T m_i \end{aligned}$$

将上式中与 i 无关的项去掉，上式可改写为

$$d_i(x) = m_i^T x - \frac{1}{2} m_i^T m_i \tag{3.2}$$

这样，决策规则为：

若 $$d_i(x) > d_j(x), \quad j = 1, 2, \cdots, c; j \neq i$$

则 $$x \in \omega_i$$

从式(3.2)可以看出，这个判别函数是线性判别函数，可写成

$$d_i(x) = w_i^T X \tag{3.3}$$

式中

$$w_i = \begin{bmatrix} m_i \\ -\frac{1}{2} m_i^T m_i \end{bmatrix}$$

$$X = \begin{bmatrix} x \\ 1 \end{bmatrix}$$

最小距离分类器结构见图3－2。

图中中间那层节点计算增广模式向量 X 与增广权向量 w_i 的点积，计算结果送到最大值选择器，最大值选择器输出分类结果。

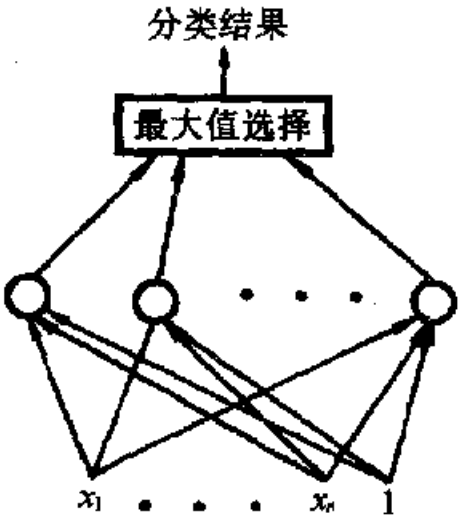


图 3－2

最小距离分类器构造简便，标准样本直接作为分类器的权向量。但是分类效果常常不理想。分类效果不好的原因恰恰就是权向量仅仅利用了标准样本的分类信息，而其它样本的信息没有利用。

3.3 感知器准则函数

这里，对分类器的设计要求用感知器准则函数表达。问题是如何利用训练样本集，“学习”得出使感知器准则函数达到极小值的判别函数中的权向量。

假设有一个两类问题，训练样本集 $S = \{X^1, X^2, \cdots, X^N\}$ 是线性可分的。目的是求出判别函数 $d(x) = w^T X$ 中的权向量 w。

为了便于下面的推导方便起见，我们对样本进行规范化处理：若

$$X^i \in \omega_1$$

则令 $$X^{(i)}=X^i$$

若 $$X^i\in\omega_2$$

则令 $$X^{(i)}=-X^i$$

称 $X^{(i)}$ 为规范化样本。规范化后，

若 $X^i\in\omega_1$ 且 w 将 X^i 正确分类，则 $w^TX^{(i)}>0$

且 w 将 X^i 错误分类，则 $w^TX^{(i)}<0$

若 $X^i\in\omega_2$ 且 w 将 X^i 正确分类，则 $w^TX^i<0, w^TX^{(i)}>0$

且 w 将 X^i 错误分类，则 $w^TX^i>0, w^TX^{(i)}<0$

这样，不管 X^i 属于哪一类，正确分类的话，$w^TX^{(i)}>0$；错误分类的话，$w^TX^{(i)}<0$。

感知器准则函数定义为

$$J(w)=\sum_{X^{(i)}\in S_e}(-w^TX^{(i)}) \tag{3.4}$$

式中，S_e 是被权向量 w 错分类的样本集合。当 X^i 被错分类时，就有 $w^TX^{(i)}<0$，即 $-w^TX^{(i)}>0$。因此，式(3.4)中的 $J(w)$ 总是大于等于零，而且仅当不存在错分类样本，即 S_e 为空集时，$J(w)$ 才为 0. 这时准则函数达到极小值，这时的 w 就是我们要寻找的权向量 w^*. 这一准则函数是 50 年代由 Rosenblatt 提出来，企图用于脑模型感知器上的，所以一般称为感知器准则函数。

有了准则函数 $J(w)$，下一步便是采用某种算法求解使 $J(w)$ 达到极小值时的权向量 w^*，这里我们采用梯度下降法。首先将准则函数 $J(w)$ 对 w 求梯度，这是一个纯量函数对向量求导的问题，不难求出

$$\nabla J(w)=\frac{\partial J(w)}{\partial w}=\sum_{X^{(i)}\in S_e}(-X^{(i)})$$

可以看出，梯度 $\nabla J(w)$ 是一个向量，其方向是 $J(w)$ 增长最快的方向。显然，负梯度方向则是 $J(w)$ 减小最快的方向。它启发我们在求准则函数极小值时，沿负梯度方向走能最快地到达极小点。

梯度下降法的基本步骤是，首先任意给定初始权向量 $w(1)$，

然后从 $w(1)$ 出发，沿负梯度方向（$-\nabla J(w(1))$）走一步，步长为 $c(1)$，得到下一次的权向量 $w(2)$

$$w(2) = w(1) - c(1)\nabla J(w(1))$$

依次类推，从 $w(k)$ 出发，沿负梯度方向（$-\nabla J(w(k))$）走一步，步长为 $c(k)$，得到下一次的权向量 $w(k+1)$

$$w(k+1) = w(k) + c(k)\nabla J(w(k)) \tag{3.5}$$

上式建立了一种迭代算法，从 $w(1)$ 出发，反复使用上式，就可得到序列

$$w(1), w(2), \cdots, w(k), w(k+1), \cdots$$

可以证明，在一定的限制条件下，它将收敛于使 $J(w)$ 达到极小的权向量 w^*。式(3.5)就是梯度下降法的迭代公式。在算法中，步长 $c(k)$ 的选择很重要。若 $c(k)$ 选得太小，算法收敛慢；若 $c(k)$ 选得太大，会使修正过分，甚至引起发散。理论上可以计算出最佳步长 $c^*(k)$，然而 $c^*(k)$ 的计算量太大，通常根据具体情况人为选定步长 $c(k)$。

对于感知器准则函数来讲，梯度下降法的迭代公式为

$$w(k+1) = w(k) + c(k)\sum_{x^{(i)}\in S_r} X^{(i)} \tag{3.6}$$

感知器准则函数梯度下降法可归纳成如下算法步骤：

①给定初始权向量 $w(1)$ 和步长 $c(k)$；

②找出被权向量 $w(k)$ 错分类的样本，转第③步；如果无错分类样本，算法结束；

③按下面的迭代公式求新的权向量 $w(k+1)$：

$$w(k+1) = w(k) + c(k)\sum_{x^{(i)}\in S_r} X^{(i)}$$

然后转第②步。

从迭代公式(3.6)可以看出，这种迭代法是把样本集中所有被错分类的样本同时找出来，然后按迭代公式(3.6)一次性修正权向量。下面我们将这种迭代法变更一下。把样本集看作一个不断重

复出现的序列而逐个加以考虑。对于任意权向量 $w(k)$，如果它把某个样本错分了，就对 $w(k)$作一次修正。这种方法称为单样本修正法。

我们构造这样一个准则函数

$$J(w)=\frac{1}{2}(|w^TX^{(k)}|-w^TX^{(k)})$$

当 $X^{(k)}$被 w 错分类时，$w^TX^{(k)}<0$，所以 $J(w)=(-w^TX^{(k)}-w^TX^{(k)})/2=-w^TX^{(k)}>0$；当 $X^{(k)}$被 w 正确分类时，$w^TX^{(k)}>0$，所以，$J(w)=(w^TX^{(k)}-w^TX^{(k)})/2=0$，这时准则函数取得极小值。

同样，我们采用梯度下降法求解使准则函数 $J(w)$达到极小值时的权向量 w^*。

准则函数 $J(w)$的梯度为

$$\nabla J(w)=\frac{\partial J(w)}{\partial w}=\begin{cases}0, & \text{若 } w^TX^{(k)}>0\\ -X^{(k)}, & \text{若 } w^TX^{(k)}<0\end{cases}$$

这样，迭代公式为

$$w(k+1)=\begin{cases}w(k), & \text{若 } w^T(k)X^{(k)}>0\\ w(k)+c(k)X^{(k)}, & \text{若 } w^T(k)X^{(k)}<0\end{cases}$$

如果不规范化样本，则迭代公式为

$$w(k-1)=\begin{cases}w(k), & \text{若 } w^T(k)X^k>0 \quad \text{且 } x^k\in\omega_1\\ & \text{或 } w^T(k)X^k<0 \quad \text{且 } X^k\in\omega_2\\ w(k)+c(k)X^k, & \text{若 } w^T(k)<0 \quad \text{且 } X^k\in\omega_1\\ w(k)-c(k)X^k, & \text{若 } w^T(k)>0 \quad \text{且 } X^k\in\omega_2\end{cases}$$

(3.7)

在上述算法中需要判断 $w^T(k)X^k$ 是大于零，还是小于零。这种判断工作可由符号函数 sgn 来完成，或者称为由阀值单元来实现。符号函数 sgn 定义为

$$\mathrm{sgn}(w^T(k)X^k)=\begin{cases}1, & \text{若 } w^T(k)X^k>0\\ -1, & \text{若 } w^T(k)X^k<0\end{cases}$$

这样，迭代公式(3.7)可改写为

$$w(k+1)=w(k)+\frac{1}{2}c(k)(d^k-\mathrm{sgn}(w^T(k)X^k))X^k \tag{3.8}$$

式中，d^k 是 X^k 的类别标志，定义如下

$$d^k=\begin{cases}1, & \text{若 } X^k\in\omega_1\\ -1, & \text{若 } X^k\in\omega_2\end{cases}$$

如果记 $o^k=\mathrm{sgn}(w^T(k)X^k)$，则迭代公式(3.8)改写为

$$w(k+1)=w(k)+\frac{1}{2}c(k)(d^k-o^k)X^k \tag{3.9}$$

单样本修正法的算法步骤如下：

已知训练样本集 $S=\{X^1,d^1;X^2,d^2;\cdots;X^N,d^N\}$，其中 d^i 是 X^i 的类别标志。

①给定初始权向量 $w(1)$ 和步长 $c(k)$，$k\leftarrow 1$，$e\leftarrow 0$，k 是步序，e 是累积误差；

②输入样本 X^k，计算 o^k：

$$o^k=\mathrm{sgn}(w^T(k)X^k)$$

③修正权向量：

$$w(k+1)=w(k)+\frac{1}{2}c(k)(d^k-o^k)X^k$$

④误差累积：

$$e\leftarrow e+\frac{1}{2}(d^k-o^k)^2$$

如果 $k<N$，则 $k\leftarrow k+1$，并转第②步；

如果 $k=N$，且 $e>0$，则 $e\leftarrow 0$，$k\leftarrow 1$，转第②步；

如果 $k=N$，且 $e=0$，训练结束，得到权向量 w.

图 3－3 给出了线性分类器的结构。

图中中间那个节点用于求取 w 与 X 的点积 w^TX. 我们把函数 f 称为激励函数。本算法中，函数 f 取为 sgn 函数。

该分类器有两个运行状态：训练(学习)状态和识别状态。

训练状态按照单样本修正法的算法步骤运行。大致过程是，首

先初始化权向量，然后输入一样本 X^k，作点积运算，得到 w^TX^k 的值。接着作符号运算，若 $w^TX^k>0$，则 $o^k=1$；若 $w^TX^k<0$，则 $o^k=-1$，然后按式(3.9)修正权向量。再输入下一个样本，……，直至训练结束。

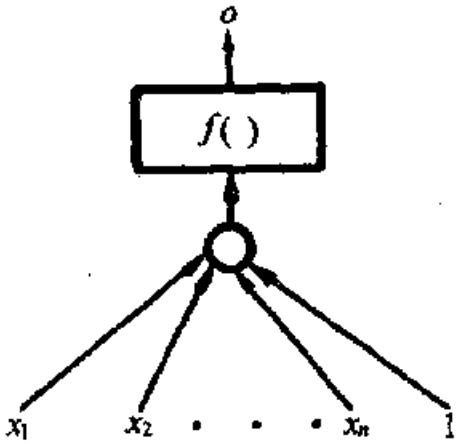


图 3-3

训练状态结束后可转入识别状态。设待识模式为 X^i，送入分类器，经运算得到 w^TX^i 的值，经符号运算得输出 o^i，若 $o^i=1$. 则将 X^i 分到 ω_1 类去；若 $o^i=-1$，则将 X^i 分别 ω_2 类去。

可以证明，采用梯度下降法对于线性可分的样本集，经过有限步修正，一定能找到一个使准则函数达到极小值的权向量 w^*，即算法在有限步内收敛。其收敛速度取决于初始权向量 $w(1)$和步长 $c(k)$。我们以二维模式为例，用图示的方法来说明这种算法的收敛性。

为了观察方便起见，假定决策边界为 $w^TX=w_1x_1+w_2x_2$，这样，决策边界为一条穿过原点的直线，见图 3-4。

图中，l 是一条由权向量 w^* 构成的决策边界，正是我们希望找到的决策边界。假如第 k 步迭代时，已得到的权向量为 $w(k)$，由 $w(k)$构成的决策边界为 $l(k)$。$l(k)$将模式空间分割成两个决策域 $R_1(k)$和 $R_2(k)$，$w(k)$是 $l(k)$的法向量，指向 $R_1(k)$。此时，若训练样本 $X^k\in\omega_1$，而且 X^k 落在 $R_1(k)$中，那么必然有 $w^T(k)X^k>0$，分类正确，权向量不需修正，$w(k+1)=w(k)$。反之，若 $X^k\in\omega_1$，但是 X^k 落在 $R_2(k)$中，则 $w^T(k)X^k<0$，分类错误，权向量应该修正，$w(k+1)=w(k)+c(k)X^k$。如图所示，向量 $w(k)$与向量 $c(k)X^k$ 之和为向量 $w(k+1)$。由 $w(k+1)$构成的决策边界 $l(k+1)$比 $l(k)$更接近于决策边界 l。因此，只要步长 $c(k)$选得合适，那么每迭代一步，所

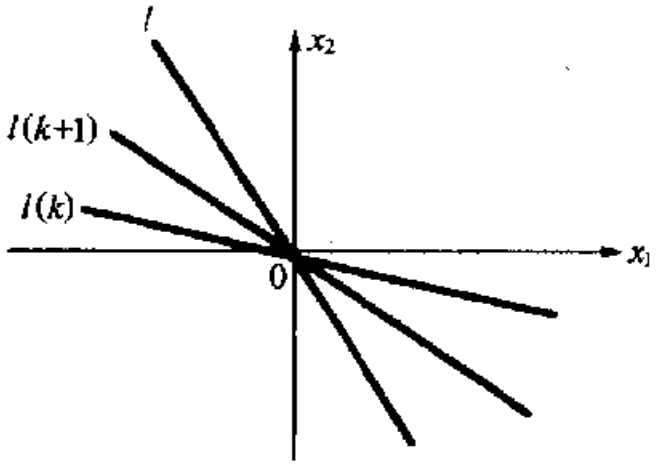


图 3－4

得到的决策边界向决策边界 l 迈进一步。经过有限步迭代后，所得到的决策边界就是决策边界 l.

另外，从图中可以看出，如果 $l(k)$ 很接近于 l 时，步长必须选得非常小，否则，加上修正量 $c(k)\boldsymbol{X}^k$ 后，修正就过头了。但是，如果各步的步长都选得非常小，收敛速度将变得十分缓慢。为了解决这个矛盾，通常采用变步长方案，在训练初始阶段，步长大一些，然后逐渐减小，比如取 $c(k)=c(1)/k$.

3.4 平方误差准则函数

感知器准则函数及其梯度下降算法只适用于线性可分情况，对于线性不可分情况，迭代过程不会终止，即算法不收敛。但在实际问题中往往事先无法知道样本集是否线性可分。因此，我们希望找到一种既适用于线性可分情况，又适用于线性不可分情况的算法。对于线性可分问题，采用这种算法一定能找到将全部样本正确分类的权向量 $\boldsymbol{w}^*$；对于线性不可分问题，能得到一个使误差平方和极小的权向量 $\boldsymbol{w}^*$。这种准则函数称为平方误差准则函数。

假设训练样本集为 $\{\boldsymbol{X}^1,\boldsymbol{X}^2,\cdots,\boldsymbol{X}^N\}$，同样为了公式推导方便

起见，将样本作规范化处理，规范化后的样本集为$\{\boldsymbol{X}^{(1)},\boldsymbol{X}^{(2)},\cdots,\boldsymbol{X}^{(N)}\}$。

假定我们找到了最佳的权向量$\boldsymbol{w}^*$，并由此$\boldsymbol{w}^*$构成判别函数。若样本为$\boldsymbol{X}^{(i)}$，则判别函数值为$(\boldsymbol{w}^*)^T\boldsymbol{X}^{(i)}=d^i$，称$d^i$为样本$\boldsymbol{X}^{(i)}$的希望输出值。

如果权向量$\boldsymbol{w}$不是最佳的权向量$\boldsymbol{w}^*$，对于样本$\boldsymbol{X}^{(i)}$，判别函数值为$\boldsymbol{w}^T\boldsymbol{X}^{(i)}$。我们定义误差

$$e_i = d^i - \boldsymbol{w}^T\boldsymbol{X}^{(i)}$$

现在我们把误差平方和作为平方误差准则函数

$$J(\boldsymbol{w})=\frac{1}{2}\sum_{i=1}^{N}e_i^2=\frac{1}{2}\sum_{i=1}^{N}(d^i-\boldsymbol{w}^T\boldsymbol{X}^{(i)})^2$$

这个准则函数也可以用矩阵形式表示

$$J(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{d}-X\boldsymbol{w}\|^2$$

其中，$\boldsymbol{d}=(d^1,d^2,\cdots,d^N)^T$，$X=(\boldsymbol{X}^{(1)},\boldsymbol{X}^{(2)},\cdots,\boldsymbol{X}^{(N)})^T$.

有了准则函数$J(\boldsymbol{w})$，下一步便是采用何种算法求解使$J(\boldsymbol{w})$取极小值的权向量$\boldsymbol{w}^*$.

一、伪逆法

这是一种解析法，首先求$J(\boldsymbol{w})$的梯度

$$\nabla J(\boldsymbol{w}) = \sum_{i=1}^{N}(\boldsymbol{w}^T\boldsymbol{X}^{(i)} - d^i)\boldsymbol{X}^{(i)} = X^T(X\boldsymbol{w} - \boldsymbol{d}) \quad (3.10)$$

令$\nabla J(\boldsymbol{w})=0$，得

$$X^TX\boldsymbol{w} = X^T\boldsymbol{d}$$

式中　X^TX是方阵，而且一般是非奇异的，因此可得唯一解

$$\boldsymbol{w}^* = (X^TX)^{-1}X^T\boldsymbol{d} = X^{\#}\boldsymbol{d}$$

式中，$X^{\#}=(X^TX)^{-1}X^T$，称为X的伪逆。

例　已知样本为

$$\omega_1:\ \boldsymbol{x}^1 = (0,0)^T,\qquad \boldsymbol{x}^2 = (0,1)^T$$

$$\omega_2:\ \boldsymbol{x}^3 = (1,0)^T,\qquad \boldsymbol{x}^4 = (1,1)^T$$

样本规范化为

$\boldsymbol{X}^{(1)}=(0,0,1)^T,\boldsymbol{X}^{(2)}=(0,1,1)^T,\boldsymbol{X}^{(3)}=(-1,0,-1)^T,$

$\boldsymbol{X}^{(4)}=(-1,-1,-1)^T$

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

$$X^{\#}=(X^TX)^{-1}X^T=\frac{1}{2}\begin{bmatrix}-1 & -1 & -1 & -1\\ -1 & 1 & 1 & -1\\ \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2}\end{bmatrix}$$

若取 $$\boldsymbol{d}=(1,1,1,1)^T$$

则 $$\boldsymbol{w}=X^{\#}\boldsymbol{d}=(-2,0,1)^T$$

故 $$d(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{X}=(-2,0,1)(x_1,x_2,1)^T=-2x_1+1$$

二、梯度下降法

用伪逆法可以求得

$$\boldsymbol{w}^*=X^{\#}\boldsymbol{d}$$

其中需要计算伪逆 $X^{\#}=(X^TX)^{-1}X^T$. 计算 $X^{\#}$ 可能遇到两个问题，其一是要求 X^TX 非奇异，其二是求 $X^{\#}$ 时计算量大，同时还可能引入较大的计算误差。实际上往往不用这种解析法。为此，我们可以采用梯度下降法来求解。

由式(3.10)可知，$J(\boldsymbol{w})$的梯度为

$$\nabla J(\boldsymbol{w})=X^T(X\boldsymbol{w}-\boldsymbol{d})$$

这样，梯度下降法的迭代公式为

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)-c(k)X^T(X\boldsymbol{w}(k)-\boldsymbol{d})$$

可以证明，如果选择步长 $c(k)=c(1)/k$，$c(1)$是任意正常数，则用该算法得到的权向量序列收敛于使

$$\nabla J(w) = X^T(Xw - d) = 0$$

的权向量 w^*.

无论矩阵(X^TX)是否奇异，这个算法总能得到权向量 w^*，而且该算法只计算 $n\times n$ 的方阵 X^TX，比计算 $n\times N$ 阵 $X^{\#}$计算量要小得多。

三、Widrow—Hoff 算法

上面介绍的迭代法是成批修正法，即，求出所有样本的误差$(Xw(k)-d)$后，才对权向量修正一次。也可以逐个样本加以考虑。

定义准则函数为

$$J(w) = \frac{1}{8\|X^{(k)}\|^2}[d^k - w^TX^{(k)} + |d^k - w^TX^{(k)}|]^2$$

可以求出 $J(w)$的梯度为

$$\nabla J(w) = \begin{cases} 0, & d^k - w^TX^{(k)} < 0 \\ -(d^k - w^TX^{(k)})\dfrac{X^{(k)}}{\|X^{(k)}\|^2}, & d^k - w^TX^{(k)} > 0 \end{cases}$$

这样，迭代公式为

$$w(k+1) = \begin{cases} w(k), & d^k - w^T(k)X^{(k)} < 0 \\ w(k) + \dfrac{c(k)}{\|X^{(k)}\|^2}(d^k - w^T(k)X^{(k)})X^{(k)}, & \\ & d^k - w^T(k)X^{(k)} > 0 \end{cases}$$

步长 $c(k)$一般是人为确定的，干脆把 $\|X^{(k)}\|^2$ 放入步长中，上述迭代公式可改写为

$$w(k+1) = \begin{cases} w(k), & d^k - w^T(k)X^{(k)} < 0 \\ w(k) + c(k)(d^k - w^T(k)X^{(k)})X^{(k)}, & \\ & d^k - w^T(k)X^{(k)} > 0 \end{cases}$$

一般选择 $c(k)=c(1)/k$，以保证算法收敛。

四、单样本修正法

不管是伪逆法还是上面介绍的迭代法，都存在一个致命的问

题：如何确定 X^k 的希望输出值 d^k？我们知道，d^k 是由最佳权 向量 w^* 确定的：$d^k=(w^*)^TX^k$，而最佳权向量 w^* 是未知的，正是我们要寻找的。既然 w^* 是未知的，X^k 的希望输出值从何而来？只好任意假定。可想而知，凭任意假定的 d^k，采用上述算法就不可能求得最佳的权向量 w^*。这个问题就成了：有了最佳权向量 w^*，才能有各个样本的希望输出值；而有了各个样本的希望输出值才能应用上述算法求出最佳权向量。这就形成了一个死扣(如图 3－5)。

为了切断这个死扣，我们假定各样本的希望输出值是已知的。在样本的希望输出值已知的条件下，采用某种迭代算法求出权向量 w^*，这就是我们下面要介绍的单样本修正法。

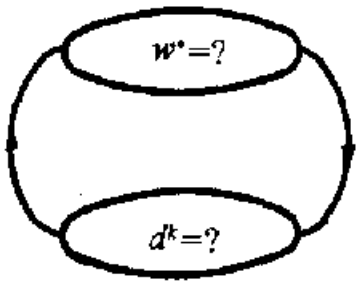


图 3－5

现在，我们不把 $(w^*)^TX$ 作为判别函数，而是把 $(w^*)^TX$ 的某种函数 $f((w^*)^TX)$ 作为判别函数，这样，样本 X^k 的希望输出值为 $d^k=f((w^*)^TX^k)$。这种函数 $f((w^*)^TX^k)$ 的值只需根据 X^k 的类别就能确定：

若 $X^k\in\omega_1$

则 $f((w^*)^TX^k)=1$

若 $X^k\in\omega_2$

则 $f((w^*)^TX^k)=-1$

这样，根据 X^k 的类别就能确定 X^k 的希望输出值：

若 $X^k\in\omega_1$

则 $d^k=1$

若 $X^k\in\omega_2$

则 $d^k=-1$

如果用非最佳的权向量 w 构成判别函数 $f(w^TX)$，那么，对于 X^k 的判别函数值 $f(w^TX^k)=o^k$ 与 X^k 的希望输出值 d^k 之间会存在

·定误差

$$e^k = d^k - f(\boldsymbol{w}^T\boldsymbol{X}^k)$$

现在我们定义准则函数为

$$J(\boldsymbol{w}) = \frac{1}{2}[d^k - f(\boldsymbol{w}^T\boldsymbol{X}^k)]^2 = \frac{1}{2}[d^k - o^k]^2$$

我们仍用梯度下降法求出使得 $J(\boldsymbol{w})$ 达到极小值时的权向量，$J(\boldsymbol{w})$ 的梯度为

$$\nabla J(\boldsymbol{w} = -(d^k - o^k)f'(\boldsymbol{w}^T\boldsymbol{X}^k)\boldsymbol{X}^k$$

若记 $\boldsymbol{w}^T\boldsymbol{X}^k \triangleq y^k$，则上式改写为

$$\nabla J(\boldsymbol{w}) = -(d^k - o^k)f'(y^k)\boldsymbol{X}^k$$

式中 $f'(y^k)$ 是函数 f 对 y^k 的一阶导数。

接下来的问题是函数 f 可取什么样的函数。函数 f 应满足两个基本条件：

①函数 f 的值域为[−1,1]；

②函数 f 必须可导。

满足这两个基本条件的函数很多，常用的一种函数为双极型 S 形函数

$$f(y) = \frac{2}{1 + \exp(-y)} - 1 = \frac{1 - \exp(-y)}{1 + \exp(-y)}$$

这个函数的图形见图 3-6。

显然，这个函数满足条件：

①当 $y \to \infty$ 时，$f(y) = 1$；

当 $y \to -\infty$ 时，$f(y) = -1$。

②函数 $f(y)$ 在 $(-\infty, \infty)$ 范围内可导。

函数 f 的一阶导数为

$$f'(y^k) = \frac{1}{2}[1 - (\frac{1 - \exp(-y^k)}{1 + \exp(-y^k)})^2]$$

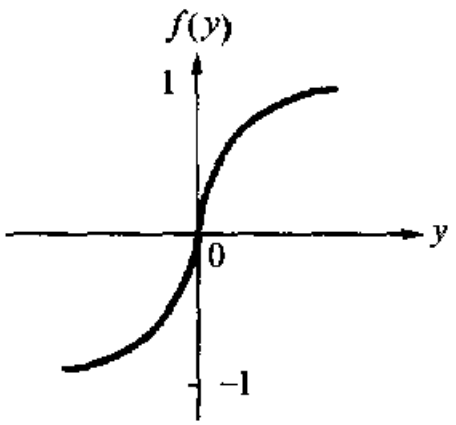


图 3-6

$$=\frac{1}{2}[1-(f(y^k))^2]$$

$$=\frac{1}{2}(1-(o^k)^2)$$

式中 $o^k=f(y^k)$.

这样，$J(w)$的梯度为

$$\nabla J(w)=-\frac{1}{2}(d^k-o^k)(1-(o^k)^2)X^k$$

由此，单样本修正法的迭代公式为

$$w(k+1)=w(k)+\frac{1}{2}c(k)(d^k-o^k)(1-(o^k)^2)X^k \tag{3.11}$$

单样本修正法的线性分类器结构与感知器准则函数单样本修正法的分类器结构基本相同，两者的训练算法步骤也基本相同。差别主要有两点：

①感知器准则函数单样本修正法中的激励函数取为 sgn 函数，而平方误差准则函数单样本修正法中的激励函数取为双极型 S 型函数。

②前者的迭代公式为式(3.9)：

$$w(k+1)=w(k)+\frac{1}{2}c(k)(d^k-o^k)X^k$$

后者的迭代公式为式(3.11)。

表面上看，这两种算法差别不大，然而在线性不可分的情况下，感知器准则函数单样本修正法不收敛，而平方误差准则函数单样本修正法肯定收敛，能得到一个使平方误差最小的权向量 w^*. 当然，平方误差准则函数单样本修正法也有其缺点，在迭代公式中多了一个因子$[1-(o^k)^2]$，该因子是小于 1 的正数。这样，修正量比较小，相对来讲，收敛速度就比较慢了。

3.5 多类模式的线性分类器

设有 c 类模式。设计 c 个线性判别函数：

$$d_1(X)=w_1^TX, d_2(X)=w_2^TX, \cdots, d_c(X)=w_c^TX.$$

决策规则为：

若 $$d_i(X)>d_j(X), \quad j=1,2,\cdots,c; j\neq i$$

则 $$X\in\omega_i$$

我们将感知器准则函数单样本修正法推广到 c 类情况。算法如下：

设初始权向量为 $w_1(1), w_2(1), \cdots w_c(1)$.

如果在训练过程的第 k 次迭代时，所用样本为 X^k，且 $X^k\in\omega_i$，分别计算出 $d_1(X^k), d_2(X^k), \cdots, d_c(X^k)$。然后按下列规则修正权向量：

(1)若 $d_i(X^k)>d_j(X^k), j=1,2,\cdots c; j\neq i$，则权向量不修正，即

$$w_j(k+1)=w_j(k), \quad j=1,2,\cdots,c$$

(2)若其中第 l 个权向量使得 $d_i(X^k)\leqslant d_l(X^k)$，则权向量作如下修正

$$w_i(k+1)=w_i(k)+c(k)X^k$$

$$w_l(k+1)=w_l(k)-c(k)X^k$$

$$w_j(k+1)=w_j(k), \quad j=1,2,\cdots,c; j\neq i, j\neq l \qquad (3.12)$$

其中 $c(k)$ 为步长。可以证明，只要样本集是线性可分的，则该算法经有限次迭代后收敛。

在上述算法中，需要作大小比较运算：$w_i^TX^k>w_j^TX^k$？现在我们设法将比较运算改成加减运算。为此，设 X^k 相对各类别的类别标志为 $d_1^k, d_2^k, \cdots, d_c^k$。

我们对类别标志作如下规定：

若 $X^k\in\omega_i$，则 $d_i^k=1, d_j^k=-1, j=1,2,\cdots,c; j\neq i$. 不难想像，这里所谓的类别标志 d_i^k 就是 X^k 相对于第 l 类的希望输出值。

这样，权向量的修正规则改写为

$$w_i(k+1)=w_i(k)+\frac{1}{2}c(k)[d_i^k-\mathrm{sgn}(w_i^T(k)X^k]X^k$$

$$= \boldsymbol{w}_i(k) + \frac{1}{2}c(k)(d_i^k - o_i^k)\boldsymbol{X}^k \qquad (3.13)$$

其中，$o_i^k = \mathrm{sgn}(\boldsymbol{w}_i^T(k)\boldsymbol{X}^k)$，符号函数规定如下

$$\mathrm{sgn}(\boldsymbol{w}^T\boldsymbol{X}) = \begin{cases} 1, & \text{若 } \boldsymbol{w}^T\boldsymbol{X} > 0 \\ -1, & \text{若 } \boldsymbol{w}^T\boldsymbol{X} < 0 \end{cases}$$

不难看出式(3.13)与式(3.9)是一致的。

c 类线性分类器结构见图 3-7。

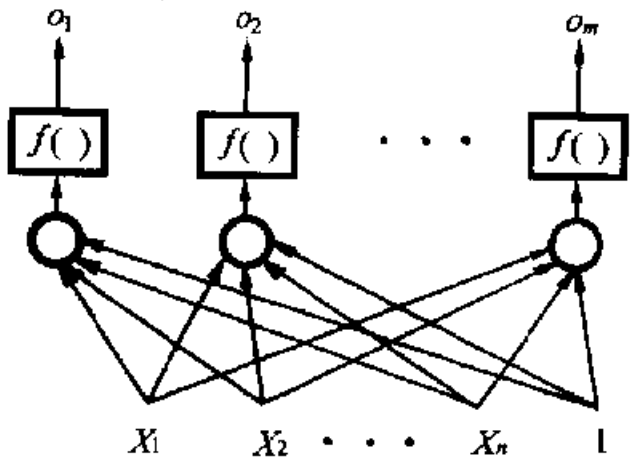


图 3-7

图中，中间那层节点作点积运算，激励函数 f 为 sgn 函数。

该分类器在训练状态时，按迭代公式(3.13)修正权向量。识别状态时，将待识模式 $\boldsymbol{X}$ 送入分类器输入端，若输出端 o_i 为+1，则将 $\boldsymbol{X}$ 判属 ω_i 类。

同样可将平方误差准则函数单样本修正法推广到 c 类情况。权向量修正公式为

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \frac{1}{2}c(k)(d_i^k - o_i^k)(1 - (o_i^k)^2)\boldsymbol{X}^k$$

分类器的结构见图 3-7，激励函数 f 取为双极型 S 型函数。

我们可以把点积运算与激励函数运算合在一起看作一个运算单元，那么这个单元可以看作是一个神经元。整个多类线性分类器可以看作由神经元组成的网络，即神经网络。如果把输入节点不算

作一层的话，那么线性分类器就是单层的前向神经网络。其实，两类的线性分类器也可看作神经网络，只不过是单个神经元的神经网络罢了。

3.6 人工神经网络概述

计算机处理视觉、听觉等信息显得非常笨拙和无能为力，而人脑处理这些信息却轻松自如。因此，人们一直在寻求构作更加逼近人脑功能的新一代计算机模型，于是出现了人工神经网络。

人工神经网络是由大量简单的基本元件——神经元相互连接而成的自适应非线性动态系统。每个神经元的结构和功能比较简单，而大量神经元组合产生的系统行为却非常复杂。与目前广泛应用的计算机相比较，人工神经网络在构成原理和功能特点等方面更加接近人脑。它不是按给定的程序一步一步地执行运算，而是能够自身适应环境、总结规则，完成某种运算、识别或过程控制。因而有人称人工神经网络为非程序化的自适应信息处理机。

人工神经网络的应用主要在以下三个方面：信号处理与模式识别，知识处理工程或专家系统，运动过程控制。这些应用的共同特点是：难以用算法来描述待处理的问题，然而存在大量的范例可供学习。实际上，自然界提供给我们的信息处理问题可分为两大类型：结构性问题和非结构性问题。前者如数值计算或方程求解，这正是冯·诺依曼计算机善于解决的问题。后者如语音或图形识别，人们无法把自己的认识翻译成严密的机器指令，因而难以用冯·诺依曼机获得满意的效果。人工神经网络在处理非结构性问题方面显示了突出的优点。由于人工神经网络为解决这些非结构性问题提供了一条颇具潜力的新途径，因而引起人们的巨大兴趣。

一、生物神经元和人工神经元

人工神经元的研究起源于脑神经元学说。19 世纪末，在生物、

生理学领域，Waldeger 等人创建了神经元学说。人们开始认识到，复杂的神经系统由数目众多的神经元组合而成。大脑皮层神经元的数目在 10^{10}～10^{11}量级，每立方毫米内约有数万个。神经元的类型有多种，它的基本结构如图 3－8 所示。

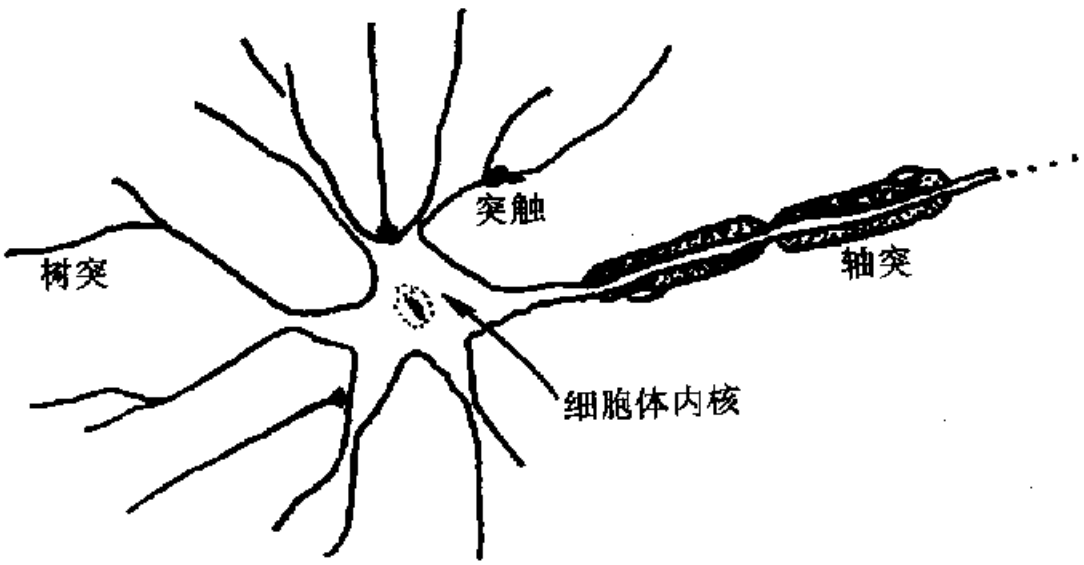


图 3－8

神经元基本上由三部分组成：细胞体、树突和轴突。细胞体是信息处理的地方。树突是由细胞体延长出来的，逐渐分支变细，呈现树状形式。树突的全长各部位都可与其它神经元的轴突末稍相互连系，形成所谓“突触”。在突触处两神经元并未连通，它只是发生信息传递功能的结合部。联系界面之间隙约为$(15\sim50)\times10^{-9}$ m，每个神经元的突触数目不等，最高可达 10^5 个。各神经元之间的连接强度有所不同，并且都可调整，基于这一特性，人脑具有存储信息的功能。轴突可以看作为信息传输管道，轴突的末端分裂成许多末稍，由这些末稍把信息传输到其它神经元。

神经元具有两种工作状态：兴奋状态和抑制状态。当传入的神经冲动，使细胞膜电位升高到阈值(约为 40mv)时，细胞进入兴奋状态，产生神经冲动，由轴突输出。相反，若传入的神经冲动，使细

胞膜电位下降到低于阈值时，细胞进入抑制状态，没有神经冲动输出。

人工神经元是对生物神经元的某种模仿、简化和抽象，其结构模型如图 3－9 所示。

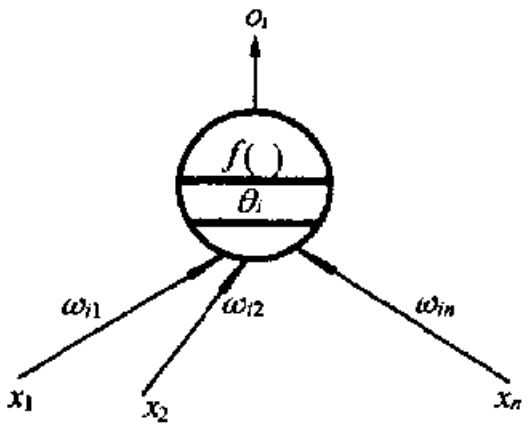


图 3－9

对于第 i 个神经元，接受多个其它神经元的输入信号 x_j，各突触强度用实数 w_{ij} 来表示，称为权系数，这是第 j 个神经元对第 i 个神经元作用的加权值。利用某种运算把输入信号的作用结合起来，称为“净输入”，用 Net_i 表示。净输入表达式有多种类型，其中最简单的一种形式是线性加权求和，即 $Net_i = \sum w_{ij}x_j$. 这样，人工神经元的数学表示式为

$$o_i = f(\sum_{j=1}^{n} w_{ij}x_j - \theta_i) \tag{3.14}$$

式中，θ_i 为阈值，f 为激励函数。当净输入 Net_i 超过阈值 θ_i 时，$o_i \to 1$，反之，$o_i \to -1$（或定义为 0）。为了数学上处理方便起见，可将 $-\theta_i$ 看作 $w_{i,n+1}$，这样，式(3.14)可写为 $o_i = f(\sum_{j=1}^{n+1} w_{ij}x_j) = f(y_i)$

式中，$y_i = \sum_{j=1}^{n+1} w_{ij}x_j$.

激励函数 f 是单调非降函数。有三种形式的函数常被用作激

励函数：

(1)线性函数

$$f(y) = y$$

(2)阶跃函数

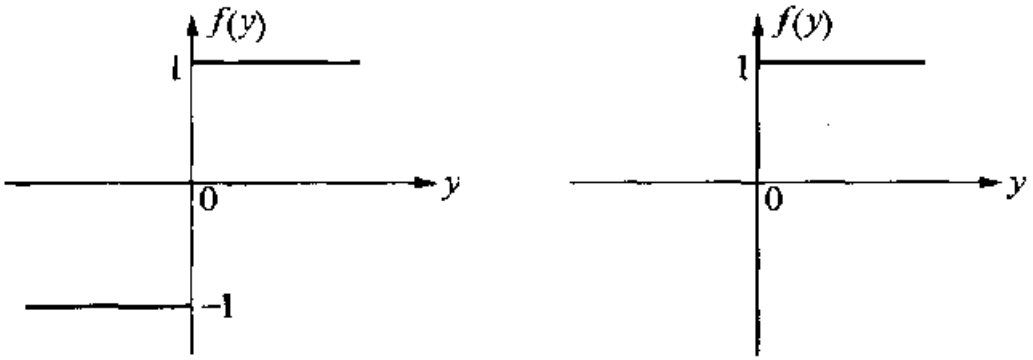
双极型阶跃函数：

$$f(y) = \begin{cases} 1, & y \geqslant 0 \\ -1, & y < 0 \end{cases}$$

单极型阶跃函数：

$$f(y) = \begin{cases} 1, & y \geqslant 0 \\ 0, & y < 0 \end{cases}$$

相应的函数图形如图 3－10 所示。



双级型　　单级型

图 3－10

感知器准则算法的线性分类器中就使用了双极型阶跃函数作为激励函数。

(3)Sigmoid 函数(简称 S 形函数)

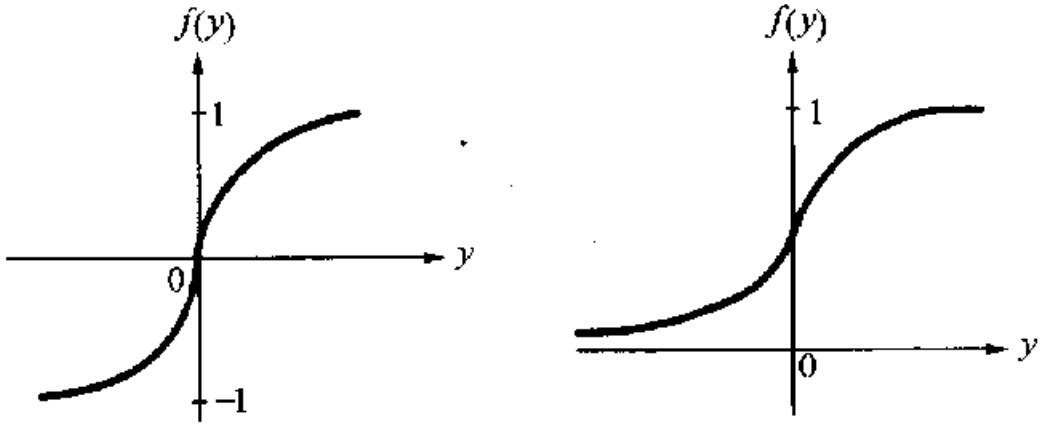
双极型 S 形函数：

$$f(y) = \frac{2}{1+\exp(-\lambda y)} - 1$$

单极型 S 形函数：

$$f(y) = \frac{1}{1+\exp(-\lambda y)}$$

式中，参数 λ 影响函数形状的陡度。S 形函数的图形如图 3－11 所



双级型　　单级型

图 3-11

示。

平方误差准则算法的线性分类器中就使用了双极型 S 形函数作为激励函数。

二、神经元之间的连接形式

神经网络是一个复杂的互连系统，神经元之间的互连形式对网络的性质和功能会产生重要影响，典型的互连形式为两种。

(1)前向网络(前馈网络)

前馈网络的结构如图 3-12 所示。网络可划分为若干层，各层依次排列，上一层的神经元只接受下一层神经元输出的信号，各神经元之间没有反馈。输入节点没有计算功能，只是为了标明输入值。其它各层的神经元都有计算功能。每个神经元有多个输入，但只有一个输出。

输入层和输出层称为可见层，而其它中间层则称为隐含层，隐含层中的神经元称为隐节点。

(2)反馈网络

典型的反馈网络如图 3-13 所示。每个节点都表示一个神经元。每个神经元既接受其它各节点的反馈输入又接受外加输入，也都直接向外部输出。在有些反馈网络中，还包括自环反馈。

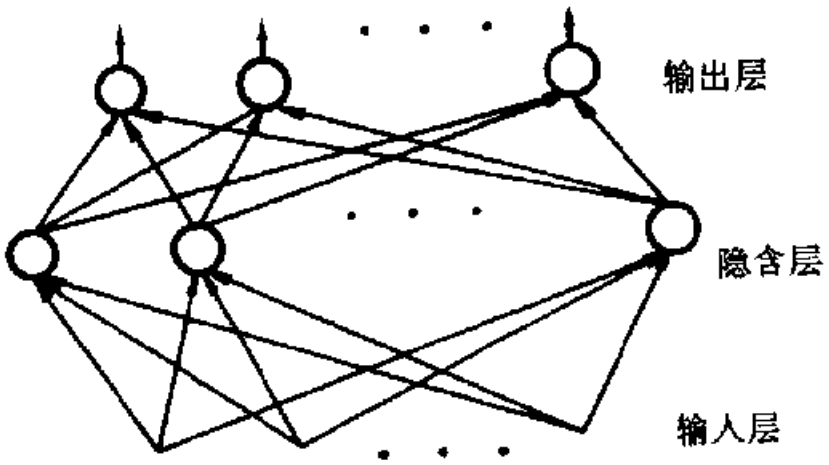


图 3－12

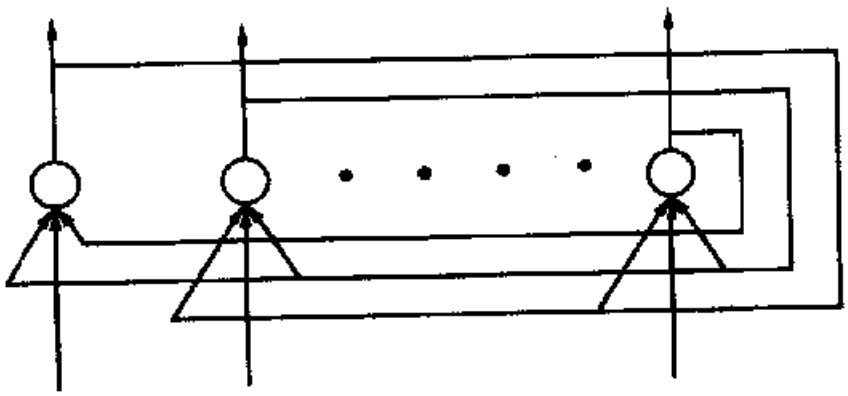


图 3－13

第四章　非线性分类器

如果样本集线性可分，那么用线性分类器就能很好地完成分类任务。但是，如果样本集线性不可分，那么采用线性分类器，虽然能在某种准则函数下达到最好的分类效果，但分类错误往往偏大。因此，如果样本集线性不可分时，采用非线性分类器往往是明智的选择。

按判别函数的形式，非线性判别函数一般可分为分段线性判别函数、二次型判别函数和其它非线性判别函数。

分段线性判别函数是一种特殊的非线性判别函数，它确定的决策边界是由若干段超平面组成的，能逼近各种形状的超曲面，具有很强的适应能力，而与一般超曲面相比，超平面是简单的。因此，分段线性分类器是一种常用的非性线分类器。本章要介绍的近邻法分类器和前向多层神经网络均属分段线性分类器。

4.1　近　邻　法

3.1 节曾讨论了一种最简单而直观的分类方法：最小距离分类法。这种方法在每类样本中只选择了一个标准样本作为分类的依据，其它样本的信息没有利用，所以分类效果往往不理想。近邻法采用一种极端方法，将全部样本都作为标准样本。这种决策方法称为近邻法。

一、最近邻法

最近邻法也是按距离来分类的，其分类思想很简单：如果待识模式 x 与样本 x^k 之间的距离最小，而 $x^k \in \omega_i$，则决策 x 属 ω_i 类。

将 x 与 ω_i 类之间的距离定义为

$$D_i(x) = \min_{x^k \in \omega_i} \{ \| x - x^k \|^2 \}$$

则最近邻法决策规则为：

若 $$D_i(x) < D_j(x), \quad j=1,2,\cdots,c; j \neq i$$

则 $$x \in \omega_i$$

类似于 3.1 节中的推导，可得最近邻法判别函数为

$$d_i(x) = \max_{x^k \in \omega_i} \{ (x^k)^T x - \frac{1}{2} \| x^k \|^2 \}$$

显然，上式是分段线性判别函数。决策规则改写为

若 $$d_i(x) > d_j(x), \quad j=1,2,\cdots,c; j \neq i$$

则 $$x \in \omega_i$$

可以证明，最近邻法的错误率 P 与贝叶斯错误率 P^* 之间存在下述关系

$$P^* \leqslant P \leqslant P^* (2 - \frac{c}{c-1} P^*)$$

式中 c 为类别数。若设

$$P(\omega_m / x) = \max_i \{ P(\omega_i / x) \}$$

则 $$P^* = \int [1 - P(\omega_m / x)] p(x) \mathrm{d}x$$

二、k -近邻法

最近邻法的一个合理的推广是 k -近邻法。顾名思义，这个方法就是取待识模式 x 的 k 个近邻，如果 k 个近邻中多数属 ω_i 类，就决策 x 属 ω_i 类。

设有 N 个样本，其中 N_1 个样本属 ω_1 类，N_2 个样本属 ω_2 类，……，N_c 个样本属 ω_c 类。在 x 的 k 个近邻中，k_1 样本属 ω_1 类，k_2 个样本属 ω_2 类，……，k_c 个样本属 ω_c 类。我们可以定义判别函数为

$$d_i(x) = k_i$$

决策规则为：

若 $$d_i(\boldsymbol{x})>d_j(\boldsymbol{x}),\quad j=1,2,\cdots,c;j\neq i$$

则 $$\boldsymbol{x}\in\omega_i$$

经理论分析，k-近邻法的错误率 P 与贝叶斯错误率有如下关系

$$P^*\leqslant P\leqslant C_k(P^*)\leqslant C_{k-1}(P^*)\leqslant\cdots\leqslant C_1(P^*)$$

$$\leqslant P^*\left(2-\frac{c}{c-1}P^*\right)$$

其中 $C_i(P^*)$ 是 P^* 的一种函数。

上面我们讨论了两种近邻法。近邻法直观简单，而且分类效果相当不差：

$$P^*\leqslant P\leqslant P^*\left(2-\frac{c}{c-1}P^*\right)$$

由于 P^* 一般很小，$P^*<<1$，可将上式右边括号中的第二项略去，则

$$P^*\leqslant P\leqslant 2P^*$$

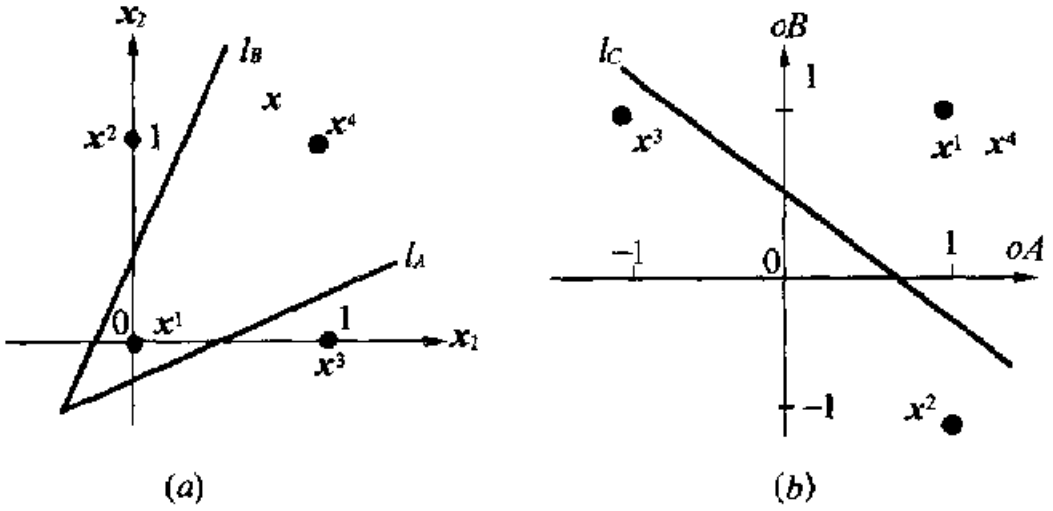
可见，近邻法错误率的上界仅为贝叶斯错误率的两倍。所以，近邻法颇为诱人，直至现在仍然是模式识别最重要的方法之一。但是，近邻法存在明显的缺点：存储量大，计算量大。采用近邻法，需要将所有的样本存入计算机中，每次分类时都要计算待识模式与全部样本之间的距离，然后进行比较。因此存储量和计算量都很大。为了解决存储量大的问题，出现了剪辑近邻法等。一些快速算法相继提出，以解决计算量大的问题。在此我们不作讨论了。

4.2 前向多层神经网络

4.2.1 引言

异或问题是一个典型的线性不可分问题。设有四个样本：$\boldsymbol{x}^1=$

$(0,0)^T$，$x^2=(0,1)^T$，$x^3=(1,0)^T$，$x_4=(1,1)^T$。其中，x^1、x^4 属 ω_1 类，x^2、x^3 属 ω_2 类。四个样本的分布情况如图 4－1(a)所示。



(a)　　(b)

图 4－1

显然，这四个样本不可能用一个线性决策边界将两类样本正确分开。然而，可用由两段线性边界组成的分段线性决策边界将两类样本正确分开。在图 4－1(a)中，线性边界 l_A 满足下列方程

$$-x_1+2x_2+\frac{1}{2}=0$$

这个线性边界 l_A 所完成的线性分类功能可用一个神经元来实现，见图 4－2 中的神经元 A。同样，线性边界 l_B 满足下列方程

$$2x_1+x_2+\frac{1}{2}=0$$

同样，这个线性边界所完成的线性分类功能也用一个神经元来实现，见图 4－2 中的神经元 B。

我们将神经元 A 和 B 的输出 o_A 和 o_B 构成一个空间，姑且称为映象空间，如图 4－1(b) 所示。

从图 4－1(b)可以看出，在映象空间中，这四个样本的映射点是线性可分的。图 4－1(b)中的线性边界 l_C 的方程为

$$o_A+o_B-\frac{1}{2}=0$$

这个线性边界所完成的线性分类功能用神经元 C 来完成，见图 4－2。

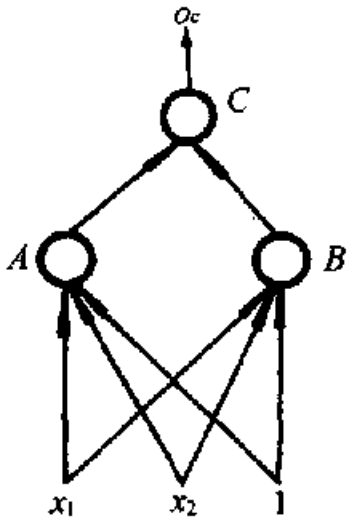


图 4－2

这样，我们就构造了一个两层的前向神经网络，这个网络解决了线性不可分的异或问题。这个例子给了我们一个极大的鼓舞：多层前向网络能解决线性不可分问题。

下面的问题是如何确定网络结构和权系数。在二维、三维情况下，我们可根据分布情况，人为地画出线性边界，根据线性边界的方程可确定网络结构和权系数，到高维，这种办法就行不通了。因此需要一种有效的学习算法。利用样本，按照一种有效的的学习算法，确定网络结构和权系数。

在上面的分析中，我们没有提及神经元的激励函数采用什么形式的函数。这意味着阶跃函数和 S 形函数均可。遗憾的是，采用阶跃函数的多层前向网络至今还没有一种有效的学习算法。采用 S 形函数的学习算法也在 80 年代中期才被提出来。

4.2.2 BP 算法

1986 年 Rumelhart 等人提出了前向多层网络的反向传播(Back Propagation)学习算法，简称 BP 算法。

前向多层网络可用一有向无环路图表示，如图 4－3 所示。网络中，输入节点数等于模式的维数，即特征个数。输出节点数一般取为类别数。隐层数和隐节点数我们后面讨论。

BP 算法的基本思想是，根据样本的希望输出与实际输出之间的平方误差，利用梯度下降法，从输出层开始，逐层修正权系数。

每个修正周期分两个阶段：前向传播阶段和反向传播阶段。

前向传播阶段：

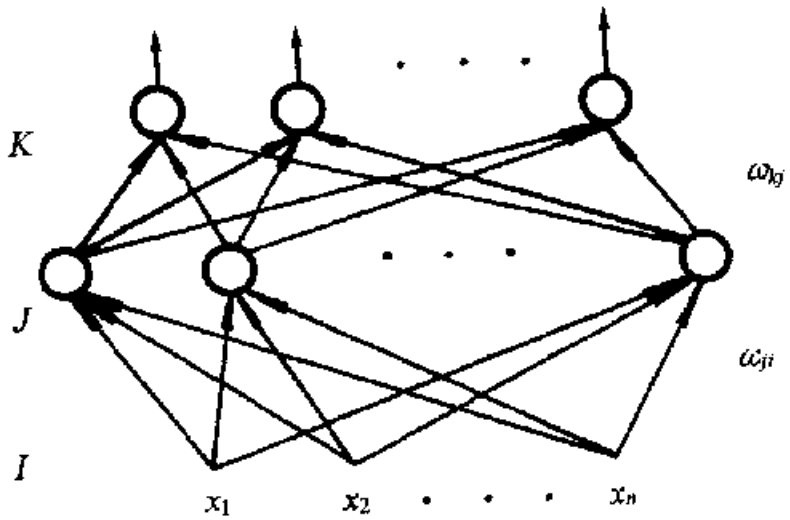


图 4－3

输入样本　　$x=(x_1,x_2,\cdots x_n)^T$

第 J 层的节点的输入为

$$y_j=\sum_{i=1}^{n+1}w_{ji}o_i,\quad x_{n+1}=1;o_i=x_i \tag{4.1}$$

节点的输出为

$$o_j=f(y_j),\quad j=1,2,\cdots,h;h\text{ 为第 }J\text{ 层的节点数} \tag{4.2}$$

第 K 层的节点的输入为

$$y_k=\sum_{j=1}^{h+1}w_{kj}o_j,\quad o_{h+1}=1 \tag{4.3}$$

节点的输出为

$$o_k=f(y_k),\quad k=1,2,\cdots,c;c\text{ 为类别数} \tag{4.4}$$

反向传播阶段：

假设输入样本 $x^p=(x_1,x_2,\cdots,x_n)^T$。该样本的希望输出为 $d^p=(d_1^p,d_2^p,\cdots,d_c^p)$。该样本经前向传播，网络输出层的实际输出为 $o^p=(o_1^p,o_2^p,\cdots,o_c^p)$。定义平方误差 E_p 为

$$E_p=\frac{1}{2}\sum_{k=1}^{c}(d_k^p-o_k^p)^2 \tag{4.5}$$

而对于全部 N 个学习样本，系统的均方误差为

$$E=\frac{1}{2N}\sum_{p=1}^{N}\sum_{k=1}^{c}(d_k^p-o_k^p)^2 \tag{4.6}$$

BP 算法以平方误差函数作为准则函数，采用梯度下降法求使准则函数达到极小值的权系数。按理说，应以式(4.6)作为准则函数，采用成批样本迭代法，但这与生物神经网络的学习过程不一样。因此，我们还是以式(4.5)作准则函数，采用单样本修正法。为了书写简单起见，将上标 p 去掉，式(4.5)改写为

$$E=\frac{1}{2}\sum_{k=1}^{c}(d_k-o_k)^2 \tag{4.7}$$

由于有误差，说明网络权系数不合适，应该修正：

$$w_{kj}\leftarrow w_{kj}+\Delta w_{kj}$$

$$\Delta w_{kj}=-\eta\frac{\partial E}{\partial w_{kj}}=-\eta\frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial w_{kj}} \tag{4.8}$$

式中，η 是步长。E 与 w_{kj} 没有直接联系，中间隔了二层：o_k 和 y_k. 因此，E 对 w_{kj} 的偏导要借助于中间量 y_k.

由式(4.3)可得

$$\frac{\partial y_k}{\partial w_{kj}}=\frac{\partial}{\partial w_{kj}}\sum_{j=1}^{h+1}w_{kj}o_j=o_j$$

将上式代入式(4.8)得

$$\begin{aligned}\Delta w_{kj}&=\eta(-\frac{\partial E}{\partial y_k})o_j\\&=\eta\delta_k o_j\end{aligned} \tag{4.9}$$

式中

$$\delta_k=-\frac{\partial E}{\partial y_k}=-\frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial y_k} \tag{4.10}$$

由式(4.7)可得

$$\frac{\partial E}{\partial o_k}=-(d_k-o_k) \tag{4.11}$$

由式(4.4)可得

$$\frac{\partial o_k}{\partial y_k}=f'(y_k) \tag{4.12}$$

将式(4.11)和(4.12)代入式(4.10)得

$$\delta_k = (d_k - o_k) f'(y_k) \tag{4.13}$$

隐含层的权系数也应该修正：

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

$$\begin{aligned}\Delta w_{ji} &= -\eta \frac{\partial E}{\partial w_{ji}} \\ &= -\eta \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial w_{ji}} \\ &= \eta \left(-\frac{\partial E}{\partial y_j}\right) \frac{\partial y_j}{\partial w_{ji}} \\ &= \eta \delta_j \frac{\partial y_j}{\partial w_{ji}}\end{aligned} \tag{4.14}$$

由式(4.1)得

$$\frac{\partial y_j}{\partial w_{ji}} = o_i \tag{4.15}$$

而

$$\delta_j = -\frac{\partial E}{\partial y_j} = -\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial y_j} \tag{4.16}$$

由式(4.2)得

$$\frac{\partial o_j}{\partial y_j} = f'(y_j) \tag{4.17}$$

由式(4.7)可得

$$\begin{aligned}-\frac{\partial E}{\partial o_j} &= -\frac{\partial}{\partial o_j}\left(\frac{1}{2}\sum_{k=1}^{c}(d_k - o_k)^2\right) \\ &= \sum_{k=1}^{c}(d_k - o_k)\frac{\partial o_k}{\partial y_k}\frac{\partial y_k}{\partial o_j} \\ &= \sum_{k=1}^{c}(d_k - o_k) f'(y_k) w_{kj} \\ &= \sum_{k=1}^{c}\delta_k w_{kj}\end{aligned} \tag{4.18}$$

将式(4.17)和(4.18)代入式(4.16)得

$$\delta_j = f'(y_j)\sum_{k=1}^{c}\delta_k w_{kj} \tag{4.19}$$

如果采用单极型 S 形函数作为激励函数，则

$$\begin{aligned} f'(y_k) &= \frac{\exp(-y_k)}{[1+\exp(-y_k)]^2} \\ &= \frac{1}{1+\exp(-y_k)}\left[1-\frac{1}{1+\exp(-y_k)}\right] \\ &= f(y_k)(1-f(y_k)) \\ &= o_k(1-o_k) \end{aligned}$$

将上式代入式(4.13)得

$$\delta_k = (d_k - o_k)o_k(1 - o_k) \tag{4.20}$$

同理可得

$$f'(y_j) = o_j(1 - o_j)$$

将上式代入式(4.19)得

$$\delta_j = o_j(1 - o_j)\sum_{k=1}^{c}\delta_k w_{kj} \tag{4.21}$$

这样，输出层节点的权系数修正公式为

$$\delta_k = (d_k - o_k)o_k(1-o_k)$$

$$w_{kj} \leftarrow w_{kj} + \eta\delta_k o_j, \quad k = 1,2,\cdots,c; j = 1,2,\cdots,h+1$$

隐含层节点的权系数修正公式为

$$\delta_j = o_j(1-o_j)\sum_{k=1}^{c}\delta_k w_{kj}$$

$$w_{ji} \leftarrow w_{ji} + \eta\delta_j o_i, \quad j = 1,2,\cdots,h; i = 1,2,\cdots,n+1$$

从上式可见，修正隐含层节点的权系数 w_{ji} 时，需要上一层算出的 δ_k 和上一层修正后的权系数 w_{kj}。因此，这是一种由输出层向输入层逐层反推的学习算法，称之为“逆推”学习算法。

BP 算法步骤如下：

给定训练样本集 $\{\boldsymbol{x}^1, \boldsymbol{d}^1; \boldsymbol{x}^2, \boldsymbol{d}^2; \cdots; \boldsymbol{x}^N, \boldsymbol{d}^N\}$。

(1)用随机数初始化所有的权系数；选定步长 $\eta > 0$；

$l \leftarrow 1$，l 用于计迭代次数，最大的迭次数取为 L；

$p \leftarrow 1$，p 用于计样本数；

$e \leftarrow 0$，e 用于累计误差，允许误差取为 Emax.

(2)输入一样本 $\boldsymbol{x}^p$ 及其希望输出 $\boldsymbol{d}^p$。

$o_i \leftarrow x_i, \quad i=1,2,\cdots,n$

$d \leftarrow d^p$

先计算隐含层 J 各节点的输出：

$$y_j=\sum_{i=1}^{n+1} w_{ji}o_i, \quad o_{n+1}=1$$

$$o_j=f(y_j), \quad j=1,2,\cdots,h$$

再计算输出层 K 各节点的输出：

$$y_k=\sum_{j=1}^{h+1} w_{kj}o_j, \quad o_{h+1}=1$$

$$o_k=f(y_k), \quad k=1,2,\cdots,c$$

(3)累计误差

$$e \leftarrow e+\sum_{k=1}^{c}(d_k-o_k)^2$$

(4)修正权系数

从输出层开始，先修正 w_{kj}

$$\delta_k=(d_k-o_k)o_k(1-o_k)$$

$$w_{kj} \leftarrow w_{kj}+\eta\delta_k o_j, \quad o_{h+1}=1, k=1,2,\cdots c; j=1,2,\cdots,h+1$$

再修正 w_{ji}

$$\delta_j=o_j(1-o_j)\sum_{k=1}^{c}\delta_k w_{kj}$$

$$w_{ji} \leftarrow w_{ji}+\eta\delta_j o_i, \quad o_{n+1}=1, j=1,2,\cdots c; i=1,2,\cdots,n+1$$

(5)如果 $p<N$，则 $p \leftarrow p+1$，转第(2)步；否则转第(6)步。

(6)$l \leftarrow l+1$. 若 $l>L$，迭代结束，否则：

如果 $e<$Emax，迭代结束；

如果 $e \geqslant$Emax，则 $e \leftarrow 0$，$p \leftarrow 1$，转第(2)步，进入下一轮迭代。

前向多层网络有两种运行状态，学习状态和识别状态。学习状态时，采用上述BP算法进行网络学习。识别状态时，将待识模式 $\boldsymbol{x}$ 送入网络输入层，网络进行计算，先求隐含层输出，再求输出层输

出 $o_1, o_2, \cdots o_c$。

若　　$o_k \approx 1,\ o_l \approx 0,\ l=1,2,\cdots,c;\ l \neq k$

则决策　　$x \in \omega_k$

4.2.3　影响 BP 算法若干因素的讨论

在实际应用中，如何使用 BP 算法，如何确定前向网络的结构，还有许多问题需要讨论。

(1)样本希望输出值的设定

由于在 BP 算法中，用 S 形函数作为神经元的激励函数，因此神经元的最大输出趋近于 1，但不能达到 1；神经元的最小输出趋近于 0，但不能达到 0。这样，在设定样本希望输出分量 d_i 时，不宜设定为 1 或 0，设定为 0.9 或 0.1 较为适宜。

(2)权系数的初始化问题

已经证明，如果将所有的权系数初始化为相同的值，那么在学习过程中，它们将始终保持相同。因此，应该用随机数设定各权系数。

(3)步长 η 的选择

步长 η 的选择是一个至关重要的问题，BP 算法的有效性和收敛性在很大程度上取决于步长 η 值。我们知道，在学习的初始阶段 η 选得大一些可以使学习速度加快，但是在临近最佳点时 η 必须相当小，否则权系数将产生振荡而不能收敛。可以采用变步长的方法，令步长 η 随着学习的进展而逐步减小，这样可以收到很好的效果。另一种权系数修正策略是采用所谓“惯性”修正法，即在权系数修正量中加上一项“惯性”项

$$\Delta w_{ij}(l+1) = \eta \hat{\delta}_j o_i + \alpha \Delta w_{ji}(l)$$

其中 l 表示第 l 次迭代；α 称为惯性系数，α 值越大则权系数修正的惯性越大，即每一次权系数的修正是与前一次权系数的修正更加密切相关。适当的 α 值有益于抑制振荡，但学习时间会延长，一般

α 值选在 0.9 左右。

(4)局部最小问题

从数学上看，*BP* 算法是一个非线性优化算法，不可避免地遇到局部最小问题。

BP 算法采用平方误差准则函数 E，以梯度下降法，希望得到使 E 达到最小的权系数。对于前向多层网络，E 相对于各权系数的变化具有非常复杂的“曲面”，其中有很多极小点，除了一个是“全局最小点”外，其它都是“局部最小点”。显然，只有全局最小点所对应的权系数才是最佳的。但是，如果权系数初值设置不当，就有可能在学习结束时 E 落在局部最小点上。

(5)前向网络结构问题

前向网络的结构由输入节点数、输出节点数、隐含层层数和隐节点数确定。

①输入节点数一般等于模式 x 的维数，即特征个数。

②输出节点数 k 一般取为类别数 c，即 $k=c$. 如果将输出进行编码的话，那么 k 可取 $\log_2 c$ 到 c 之间的任一整数。从表面上看，取 $k<c$ 简化了网络结构，但这一简化对确定网络层数的影响和对网络学习的影响尚有待研究。

③隐含层层数和隐节数。

Lippman[1987]认为，不需要更复杂的网络，即使在模式空间中，各样本分布在相互犬牙交错的复杂区域内，只有两层隐含层的前向网络就能构成所需要的任意复杂的判别函数。

例如，模式分布如图 4-4(*a*)所示。

ω_1 类的模式分布在区域 R_1 或 R_2 内，ω_2 类的模式分布在这两个区域之外。这两类之间有六条线性边界，每条边对应一个线性判别函数，每个线性判别函数可用一个神经元来实现。因此需要六个神经元，这六个神经元构成第一隐含层。这样，第一隐含层的节点数等于线性决策边界数。

R_1 的三条边组合起来构成区域 R_1，我们可用一个神经元完成

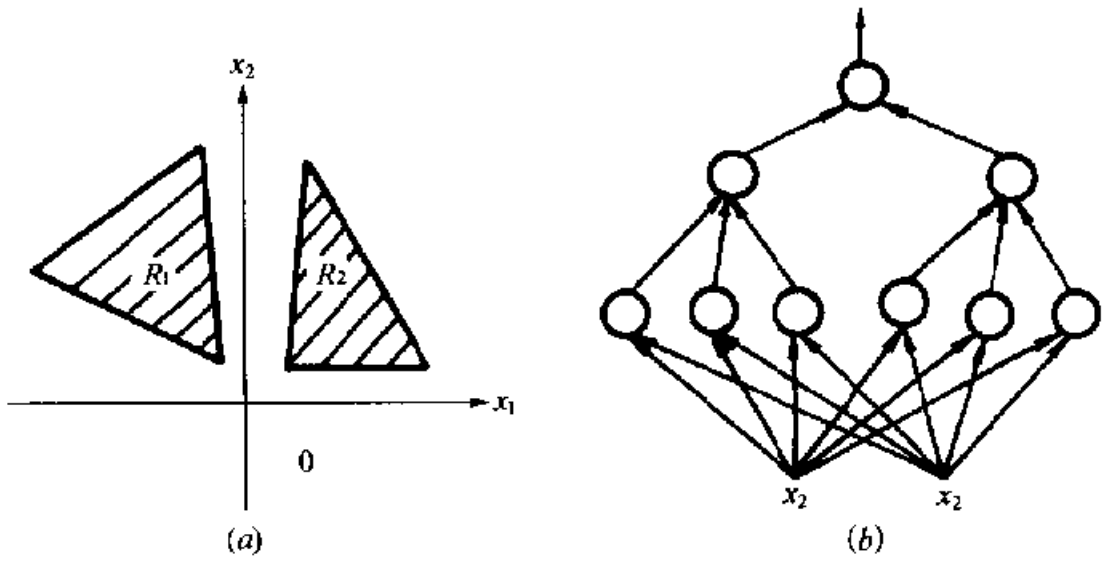


图 4－4

这个组合作用；R_2 的三条边组合成区域 R_2 的作用也用一个神经元完成。这两个神经元构成第二隐含层。因此，第二隐含层的的节点数等于区域数。

R_1 区域中的模式或 R_2 区域中的模式都是 ω_1 类的模式。所以我们还应该把 R_1 区域和 R_2 区域“或”起来得到 ω_1 类的决策区域。可用一个神经元完成将这两个区域“或”起来的作用，这个神经元就是输出节点。整个网络结构见图 4－4(b)。

从上面分析可以看出，如果模式分布复杂，区域形状复杂，需要构成这区域的超平面就多，那么第一隐含层的节点数就要多。

第二隐含层节点的作用相当于对第一隐含层节点所形成的超平面作“与”运算，构成区域。因此，第二隐含层的节点数对应于区域的个数。

输出层节点的作用是将属于同类的几个区域“或”起来，因此，输出节点数一般取为类别数。

对隐节数的确定，我们再作些讨论。为了逼近训练样本所构成的区域，似乎隐节点应选得足够多，甚至可以说，隐节点越多，逼近得越精确。那么从这个角度看，隐节点越多越好。但是从另一个角

度看，隐节点越多，不一定越好。我们训练网络的目的是利用训练好的网络来识别未知类别的模式，对未知类别模式的识别能力，称为泛化能力。由于待识模式并不是训练样本，用训练样本训练得很好的网络并不一定对待识模式识别得好，关键在于隐节点数。隐节点数越多越精确地逼近训练样本所构成的区域，产生了所谓过拟合现象，也就是没有能把不是训练样本但确属这一类的模式圈到这个区域里去，导致误识，降低了泛化能力。因此，隐节点数不能过多。但隐节点数也不能过少，过少的话，不能逼近训练样本所构成的区域，产生所谓欠拟合现象，达不到逼近精度，因而学习过程不能收敛。

上面从原理上解释了隐含层层数和隐节点数的选择原则。实际应用中，隐含层取一层或两层。由于我们并不知道模式分布的复杂性，因此，隐节点数通常要通过试验来确定恰当的隐节数。

图 4－5 给出了单层、双层、三层前向网络对异或问题和一种模式分布所采用的决策域形状，以及这三种网络所能确定的决策域的一般形状。

4.2.4 网络学习的技巧

用 BP 算法训练网络时，会遇到不收敛、训练速度慢、泛化性能差等问题。下面介绍的技巧在 BP 网的学习中有一定效果。

①重新给网络的权值初始化。有时由于网络的权系数的初始值选得不合适，BP 算法将无法获得满意的结果，此时不妨重新初始化权值，让网络重新学习。

②给权值加以扰动。在学习中途，给权值加以扰动，有可能使网络脱离目前局部最小点的陷井，但仍然能保持网络学习已获得的结果。如果知道网络权值的分布范围，例在－5 与 5 之间，则加上 10%左右的扰动，即在权值上加－0.5 到 0.5 之间的随机数。

③在网络的学习样本中适当加些噪声。这是一种加快学习速度和提高网络抗噪声能力的行之有效的办法。在学习样本中加些

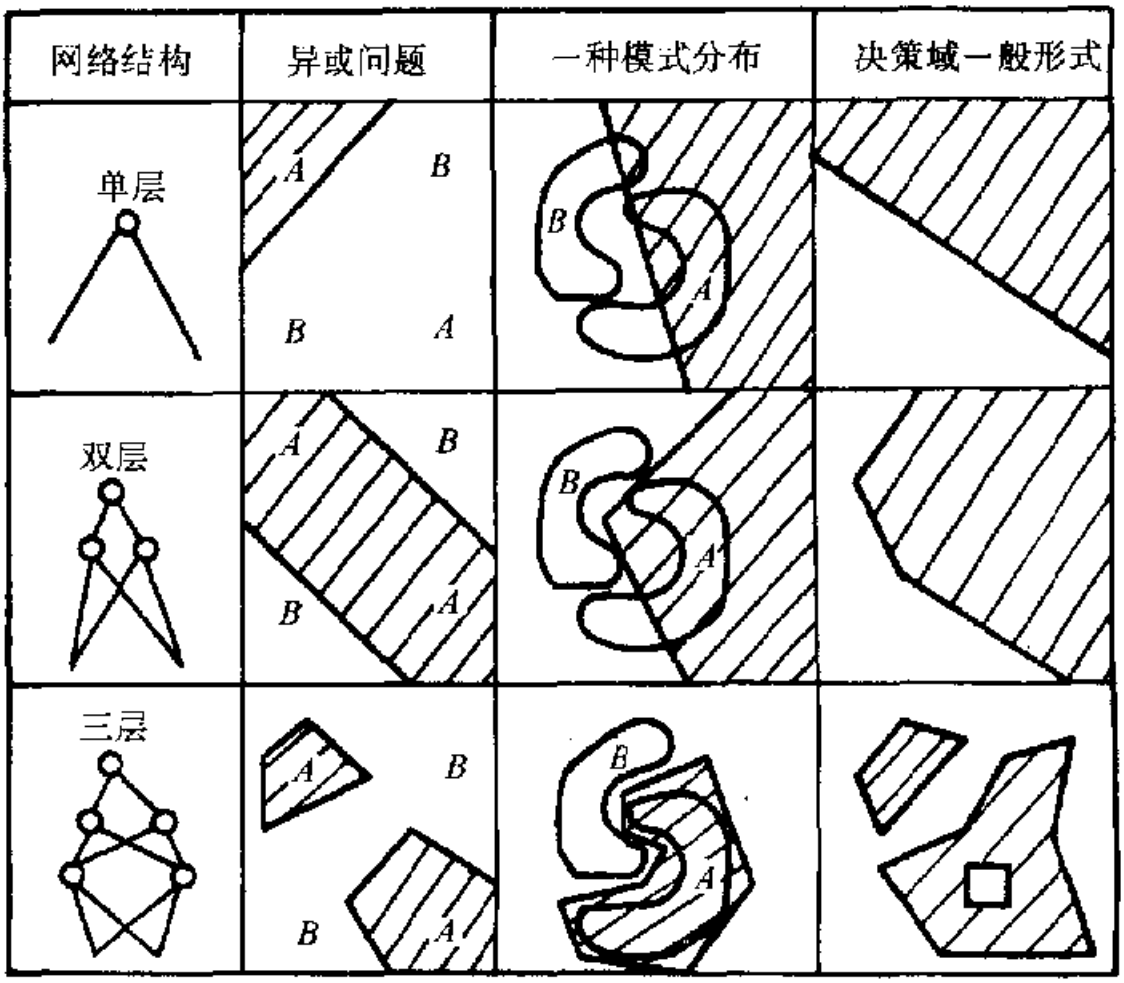


图 4－5

噪声，可避免网络依靠死记的办法来学习，因为所有的样本都会不同（虽然有些样本相似）。

④学习可有允许误差。当网络的输出与样本之间的差小于给的允许误差范围，则对此样本网络不再修正其权值。采用对网络学习宽容的做法，可加快网络学习速度。另外还可以在开始时允许误差取大些，然后逐渐减小。

⑤选择合适的网络规模。网络的层数尽量保持在三层，能不用

多层，就不要给网络加层。因为在BP算法中，误差是通过输出层向输入层反向传播的。层数愈多，反向传播误差在靠近输入层时就愈不可靠，这样用不可靠的误差来修正权值，其效果是可以想象的。另外隐含层的节点数也不要选得太多。太多的话，影响泛化能力，也会使网络的学习时间太长。

4.3 最优化算法

由梯度下降法构成的BP算法虽然为网络的学习提供了简单有力的方法，但它存在着局部极小问题。网络的权系数通过沿局部改善的方向一小步一小步地进行修正，力图达到使准则函数最小化的全局解。因此网络权系数的初始值对网络的学习有较大的影响。此外，即使权系数的初始值能够选在全局解附近，当解的周围是平坦的话，那么权系数的修正量会变得很小，因此学习的收敛速度变得很慢，迭代次数急剧增多。为了避免类似的问题，有一些最优化算法，如模拟退火算法和遗传算法可用来帮助寻找全局解。

4.3.1 模拟退火算法

模拟退火算法(Simulated Annealing，简称SA算法)是模拟加热熔化的金属的退火过程，来寻找全局最优解的有效方法之一。在金属退火处理过程中，往往先将它加温熔化，使其中的粒子可自由运动。然后逐渐降低温度，使粒子形成低能态的晶体。如果温度下降得足够慢，则金属一定会形成最低能量的基态，这一过程和金属的初始状态无关。

设 $S=\{s_1,s_2,\cdots,s_N\}$ 为所有可能的状态所构成的集合，$f:s\to R$ 为非负的代价函数。函数寻优的问题可表述如下：

寻找 $$s^*\in S$$

使 $$f(s^*)=\min_{\forall s_i\in S} f(s_i)$$

解决上述组合优化问题的SA算法过程如下：

第一步　随机选取一个初始化状态 s_i 作为当前状态 s，即令 $s=s_i$，并取一个较高的初始温度 T。

第二步　按一定的方式产生一个当前状态 s 的邻域 $N(s)$，从邻域 $N(s)$ 中随机地选取一个状态作为候选的新状态 s'，且计算

$$\Delta f = f(s') - f(s)$$

第三步　若 $\Delta f<0$，把新状态 s' 作为当前状态，即令 $s=s'$，然后转到第四步；

若 $\Delta f\geqslant 0$，则以概率 $\exp(-\Delta f/T)$ 接受新状态 s'，即令 $s=s'$。具体做法：用均匀分布的概率来产生 0 与 1 之间的随机数 ξ，若 $\xi<\exp(-\Delta f/T)$，则令 $s=s'$；否则 s 保持不变。

第四步　按一定方式降温，例如：$T\leftarrow\alpha T$，其中 α 在 0.8 与 0.9999 之间。

第五步　检查退火过程是否结束。是，转向第六步；否则转向第二步。

第六步　以当前状态 s 作为最优解输出，结束。

检查退火过程是否结束，可以采用：

①当 T 小于某一阈值；

②在最后的 L 步过程中，f 的值变化不超过预先设置的阈值 ε。

模拟退火算法是否能达到代价函数 f 的最小值，取决于初始温度是否足够高和温度是否下降得充分慢。但是温度下降太慢，会使退火过程增长。因此恰当地确定这些参数，对 SA 算法的性能有很大影响。

SA 算法是一种通用的随机搜索算法，可用于解决众多的优化问题。在对问题的领域知识甚少的情况下，采用 SA 算法最合适。因为 SA 算法不像其它确定型启发式搜索算法那样，需要依赖于问题的领域知识来提高算法的性能。但从另一个角度看，如果已知有关待解问题的一些知识话，SA 算法却无法充分利用它们，这时 SA 算法的优点就成了缺点。

4.3.2 遗传算法

生物进化包括两个基本过程：自然选择和有性生殖。第一个过程决定了群体中的哪些成员能够生存下来并传宗接代；第二个过程保证其后代的基因混合和基因重组。遗体算法(Genetic Algorithms，简称GA)是J. H. Holland根据这一生物进化模型提出的一种优化算法。

GA的算法过程如下：

设解空间为 $S=\{s_1,s_2,\cdots,s_n\}$。

第一步　在解空间中取一群点，作为遗传开始的第一代。每个点用一个二进制的数字串(基因串)表示，其优劣程度用一适应度函数(Fitness Function)来衡量。适应度大，表明那个点好，容易在遗传中生存下去。

第二步　对当前一代的每个数字串，根据由其适应度决定的概率复制到配对池中。好的数字串以高的概率被复制下来，劣的数字串被淘汰掉。

第三步　将配对池中的数字串任意配对，并对每一对数字串进行交叉操作，产生新的子孙(数字串)。

第四步　对新的数字串的某一位进行变异，这样就产生了新一代。

重复第二到第四步，直至终止。终止条件为数字串的平均适应度和最优数字串的适应度趋于收敛。从最后一代中便得到全局最优解或近似最优解。

显然，GA的最大特点在于其演算简单，只是复制数字串，交换部分数字串和改变数字串中的某一位。即只有三种演算：复制(Reproduction)、交叉(Crossover)和变异(Mutation)。

复制就是根据适应度决定的概率简单地照抄同样的数字串。交叉是指将一对数字串，按指定的地方进行交换而产生一对新的数字串，交叉的位置可由产生的随机数来决定。变异是指改变一数

字串的某一位，即改 1 为 0，或改 0 为 1。

将遗传算法用于解决一个优化问题通常要解决以下四个问题：(1)如何将问题的解编码到数字串中；(2)如何构造一个有效的适应度函数；(3)遗传演算的设计；(4)确定控制遗传演算的概率。这四个方面直接关系到遗传算法的最优解的获得。下面我们用一简单的例子讨论如何在计算机上实现 GA 的基本演算。

假设适应度函数为 $f(x)=x^2, x\in\{0,1,\cdots,31\}$。另外，有三个函数：random、flip 和 rnd，其中 random 产生在[0,1]范围内服从均匀分布的随机数；flip 根据给定的概率产生 0 或 1；rnd 产生在给定区间内服从均匀分布的整数。GA 的算法过程实现如下.

(1)编码　用有限长的数字串表示解。例如，二进制数字串 10011 的解码为

$$10011 = 1\cdot 2^2 + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 19$$

因此 5 位的二进制数字串可表示的范围为 0(00000)～31(11111)。

(2)设定初始的一群数字串。假定给定的每代数字串的个数为 4，则可调用 flip 函数 20 次，产生 4 个数字串，并对每个数字串解码，求出 $f(x)$，其结果如表 4－1 所示。

表 4－1

数字串编号	数字串	x	f_i	$f_i/\sum f_i$	被复制次数
1	01101	13	169	0.14	1
2	11000	24	576	0.49	2
3	01000	8	64	0.06	0
4	10011	19	361	0.31	1
		和	1 170	1.0	4
		平均	293	0.25	1
		最大值	576	0.49	2

(3)复制：

(3.a)$r=\text{random}\cdot\sum f_i$；

(3.b)若$\sum_{i=1}^{k-1} f_i < r$，$\sum_{i=1}^{k} f_i \geq r$，则复制第 k 个数字串。

重复(3.a)和(3.b)，直到复制成 4 个数字串。在上面的例中，第 1 个数字串被复制一次，第 2 个被复制二次，第 3 个被淘汰，第 4 个被复制一次。

(4)交叉　首先将复制的数字串任意配对。这里第 1 和第 2 个，第 3 和第 4 个数字串配成两对。然后分别对每对进行交叉操作。在进行交叉前，先决定是否要交叉，这可由给定的交叉概率 P_c，用 flip 函数来决定，若 flip 的值取 1，则表示进行交叉，否则不进行交叉，即数字串保持不变。例如 $P_c=0.6$，若 fliP 取值为 1，表明进行交叉。接下来决定交叉的位置，这可由 rnd 函数来决定。结果是第 1 和第 2 个数字串在第 4 位，第 3 和第 4 个数字串在第 2 位分别进行交叉，如表 4－2 所示。

表 4－2

数字串编号	复制的数字串	配对号码	卓接位置	新的数字串	x	f_i
1	01101	2	4	01100	12	144
2	11000	1	4	11001	25	625
3	11000	4	2	11011	27	729
4	10011	3	2	10000	16	256
					和	1 754
					平均	439
					最大值	739

(5)变异　首先根据给定的变异概率 P_m，用 flip 函数决定是否进行变异。一般来讲，给定的 P_m 都很小，例如 $P_m=0.03$。如果在一对数字串中要进行变异时，首先用 rnd 函数决定变异的位置，然后将变异位置上的 1 改 0，或 0 改为 1。这里取 $P_m=0.03$，发现不需要进行变异。

(6)由以上的(3)到(5)产生了新一代数字串后，对新一代进行

评价，即对每个数字串解码，并求出适应度。在上面的例子中，发现新的一代适应度的平均值从 293 升到 439，最大值从 576 升到 729，即新一代比上一代要改善了许多(参见表 4－1 和表 4－2)。

(7)重复上述第(3)步到第(6)步，直到规定的次数为止。然后在最后一代中找出最佳解。

遗传算法之所以成功地用于解决优化问题，是基于以下几个方面的原因：

①GA 在解空间中不只是局限于一点，而是同时处理一群点，这样可以避免陷于局部解。

②GA 不直接和解空间中的解打交道，而是处理代表解的数字串。

③GA 在寻优过程中，不需要适应度函数的微分，只需要适应度函数的值。

④GA 的寻优规则是随机性的而不是确定性的。

最后需要指出，GA 常常出现“过早收敛”问题，虽然在短时间内能找到接近全局最优解的近似解，但是它不能绝对保证收敛到全局最优解。为解决此问题，通常是先用 GA 搜索近似最优解，然后把它作为梯度下降法的初始值，这样就能获得很好的结果。

4.4 遗传 BP 算法

在 4.2 节中，我们讨论了 BP 算法，它是目前网络训练中最常用的学习方法之一。但是 BP 算法存在两个突出的弱点：收敛速度慢和可能收敛到局部极小点。这是由于在 BP 算法中，网络权系数依赖于准则函数的一阶导数信息进行修正，当求解空间存在多个局部极小点时，一旦随机产生的网络初始权系数不合适，网络会陷入局部极小点而无法逸出。为了克服这个问题，其它一些最优化方法继续被用来对 BP 算法加以改进。其中，将遗传算法与 BP 算法相结合而得到的遗传 BP 算法(GA－BP)成为一种有效的网络学

习方法。

遗传算法采用并行搜索，因此搜索效率高。GA 本质上属于随机寻优过程，不存在局部收敛问题。这就是说，GA 具有快速收敛于全局最优解的能力，这一能力恰好弥补 BP 算法容易局部收敛的缺陷。同时，与 BP 算法结合，也解决了单独利用 GA 往往只能在短时间内寻找到接近全局最优解的近优解这一问题。

GA－BP 算法训练网络分两步走：首先，用遗传算法得到最佳适应度的基因串（即一组网络权系数）；然后用这组权系数初始化网络，再利用 BP 算法训练网络。

实践证明，遗传 BP 算法是一种有效的网络学习算法。

4.5 高阶神经网络

近邻法和 BP 网的判别函数是分段线性判别函数。分段线性判别函数只是非线性判别函数中的一种。本节要介绍的高阶神经网络所采用的判别函数是广义线性判别函数。

我们先看一个具体例子，如图 4－6 所示。

设有一维模式空间 x，我们所希望的划分是，如果 $x>b$ 或 $x>a$，则 x 属于 ω_1 类；如果 $b<x<a$，则 x 属于 ω_2 类。显然，没有任何一个线性判别函数能解决上述划分问题。但是，从图 4－6 中可以看出，如果建立一个二次判别函数。

$$d(x) = (x - a)(x - b) = ab - (a + b)x + x^2$$

则可以很好地解决上述分类问题，决策规则是：

若 $d(x)>0$，则决策 $x\in\omega_1$；

若 $d(x)<0$，则决策 $x\in\omega_2$.

二次判别函数可写成如下一般形式

$$d(x) = c_0 + c_1x + c_2x^2$$

如果适当选择 $x\rightarrow x^*$ 的映射，则可把二次判别函数转化为 x^* 的线性函数

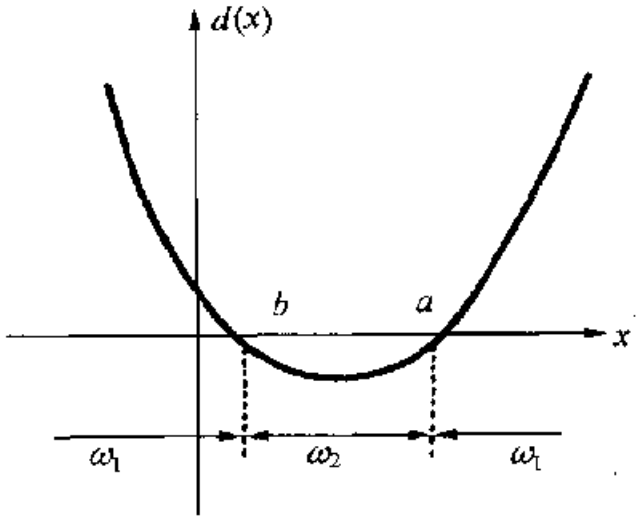


图 4－6

$$d(x) = \boldsymbol{w}^T\boldsymbol{x}^* = \sum_{i=1}^{3} w_i x_i^*$$

式中

$$\boldsymbol{x}^* = \begin{pmatrix} x_1^* \\ x_2^* \\ x_3^* \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}, \boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}$$

一般来说，对于任意高次判别函数 $d(x)$（这里的 $d(x)$ 可看作对任意判别函数作级数展开，然后取其截尾部分的逼近），都可以通过适当的变换，化为广义线性判别函数来处理。$\boldsymbol{w}^T\boldsymbol{x}^*$ 不是 $\boldsymbol{x}$ 的线性函数，但却是 $\boldsymbol{x}^*$ 的线性函数。$\boldsymbol{w}^T\boldsymbol{x}^* = 0$ 在 $\boldsymbol{x}^*$ 空间确定了一个通过原点的超平面。这样，我们就可以利用线性判别函数的简单性来解决复杂的问题。

下面我们进一步讨论 n 维模式情况。设有一样本集 $\{\boldsymbol{x}^j\}$，在模式空间 $\boldsymbol{x}$ 中线性不可分，而在 $\boldsymbol{x}^*$ 空间中线性可分。这里 $\boldsymbol{x}^*$ 的各个分量是 $\boldsymbol{x}$ 的单值函数，它的维数高于 $\boldsymbol{x}$ 的维数 n，若取

$$\boldsymbol{x}^* = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x}), 1)^T, k > n$$

则非线性判别函数为

$$d(\boldsymbol{x}) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + \cdots + w_k f_k(\boldsymbol{x}) + w_{k+1}$$

上式可写成

$$d(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}^* \tag{4.22}$$

这里，$\boldsymbol{w} = (w_1, w_2 \cdots, w_k, w_{k+1})^T$，而 $d(\boldsymbol{x}^*) = d(\boldsymbol{x})$。式(4.22)表明，非线性判别函数已被变换成线性的了。

对于 n 维向量 $\boldsymbol{x}$，如果取 $f_i(\boldsymbol{x})$ 为二次多项式函数，则判别函数为

$$d(\boldsymbol{x}) = \sum_{j=1}^{n} w_{jj} x_j^2 + \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} w_{jk} x_j x_k + \sum_{j=1}^{n} w_j x_j + w_{n+1}$$

其中，平方项有 n 项，二次项有 $n(n-1)/2$ 项，一次项为 n 项，其总共项数为

$$n + \frac{n(n-1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}$$

显然，$\boldsymbol{x}^*$ 的维数要比 $\boldsymbol{x}$ 高，$\boldsymbol{w}$ 分量的数目与 $\boldsymbol{x}^*$ 的维数相应。

由于在 $\boldsymbol{x}^*$ 空间中，判别函数 $d(\boldsymbol{x}^*)$ 是线性判别函数，所以可用单层神经网络来实现。图 4－7 给出了二阶神经网络结构。

输入样本 $\boldsymbol{x}^*$ 为

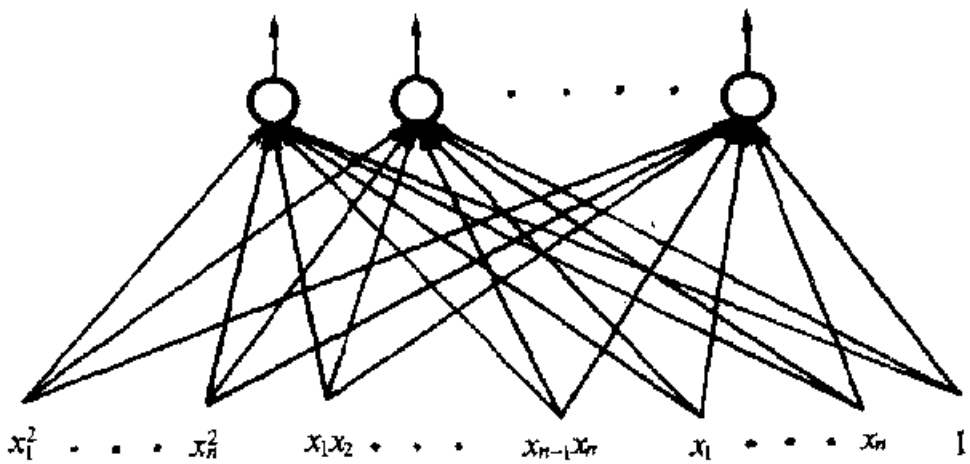


图 4－7

$$x^* = (x_1^2, \cdots, x_n^2, x_1x_2, \cdots, x_{n-1}x_n, x_1, \cdots, x_n, 1)^T$$

第 j 个输出节点的权向量为

$$w_j = (w_{j1}, w_{j2}, \cdots, w_{j(n+1)(n+2)/2})^T$$

第 j 个输出节点的输出为

$$o_j = f(w_j^T x^*)$$

f 为激励函数。

高阶神经网络的权值学习，通常也采用梯度下降法，修正公式为

$$w_j(k+1) = w_j(k) + \eta(d_j - o_j)x^*$$

式中 d_j 为样本 x^* 的希望输出，o_j 为样本 x^* 的实际输出，η 为步长。

与 BP 网相比，高阶神经网络具有以下特点：

①由于网络的输出与输入的高阶相关函数相对应，因此容易实现平移不变性、旋转不变性、比例不变性模式识别。

②由于网络不存在隐含层，因此可以获得较快的训练速度，且不容易出现局部最小问题，也避免了难以解决的隐节点个数选择问题。

③高阶神经网络的明显不足是，随着阶数的增加，权系数的个数将以几何幂速度增长，陷入所谓“维数灾难”。

第五章　聚类分析

前面讨论的分类器在学习状态时，利用已知类别的样本进行学习，学习的结果确定了分类器的参数。比如，前向多层网络，利用已知类别的样本，按 BP 算法学习，学习结束时，确定了所有的权系数。由于利用的样本是已知类别的，所以我们把这种学习称为有监督学习。在一些实际应用中，往往没有已知类别的样本可供利用，甚至将提供的样本应分成几类都不知道。例如，卫星遥感照片上各象点的分类问题，我们不知道各象点属于哪一类的，甚至不知道应将照片上的象点分几类。本章要讨论的内容就是将未知类别的样本集划分成若干子集(类)，划分的直接成果是完成了样本的分类，可能的间接成果是确定了分类器的参数。由于所用样本是没有类别标志的，所以通常称为无监督学习。

按照物以类聚的思想，对未知类别的样本集根据样本之间的相似程度分类，相似的归为一类，不相似的归为另一类，故这种分类称为聚类分析。

采用聚类分析，首先要解决两个问题：什么叫两个样本相似？相似到什么程度归为一类？因此下面先讨论模式相似性的测度和聚类准则。

5.1　模式相似性测度和聚类准则

一、相似性测度

为了能将样本集划分成不同类别，必须定义一种相似性的测度来度量同一类样本间的类似性和不属于同一类样本间的差异

性。

(1) 欧氏距离(简称距离)

设 x^i 和 x^j 为两个样本,其欧氏距离定义为

$$D = \| x^i - x^j \|$$

显然,样本 x^i 与 x^j 间的距离越小,则越相似。这与习惯用的距离概念一致。使用时要注意模式各特征的量纲。量纲不同,聚类结果可能不同。为了克服这个缺点,常使特征数据标准化,使它与量纲标尺没有关系。

(2) 街坊距离

计算欧氏距离时,需要求两个样本的各分量差的平方和并开方,为了减小计算量,可采用街坊距离:

$$D_1(x^i, x^j) = \sum_k |x_k^i - x_k^j|$$

(3) 角度相似性函数

角度相似性函数定义为

$$S(x^i, x^j) = \frac{(x^i)^T x^j}{\| x^i \| \cdot \| x^j \|}$$

$S(x^i, x^j)$ 是 x^i 的单位向量 $x^i / \| x^i \|$ 与 x^j 的单位向量 $x^j / \| x^j \|$ 之间的点积,也就是样本 x^i 与 x^j 间夹角的余弦。

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,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,当本属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。

相似性测度的共同点,都涉及把两个相比较的向量 x^i 和 x^j 的分量组合起来,但怎样组合并无普遍有效的方法,对于具体的模式分类,需视情况作适当的选择或将几种测度联合使用。

二、聚类准则

为了评价聚类结果的好坏，必须定义准则函数。有了模式相似性测度和准则函数后，聚类就变成了使准则函数取极值的优化问题了。

常用的准则函数是误差平方和准则。

若 N_i 是第 i 类 R_i 中的样本数目，$\boldsymbol{m}_i$ 是这些样本的均值，即

$$\boldsymbol{m}_i = \frac{1}{N_i}\sum_{\boldsymbol{x}\in R_i}\boldsymbol{x}$$

把 R_i 中的各样本 $\boldsymbol{x}$ 与均值 $\boldsymbol{m}_i$ 间的误差平方和对所有类相加后为

$$J = \sum_{i=1}^{c}\sum_{\boldsymbol{x}\in R_i} \| \boldsymbol{x} - \boldsymbol{m}_i \|^2$$

J 是误差平方和聚类准则，它度量了用 c 个聚类中心 $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_c$ 代表 c 个样本子集 $R_1, R_2, \cdots, R_c$ 时所产生的总的误差平方和。对于不同的聚类，J 的值当然是不同的，使 J 极小的聚类是误差平方和准则下的最优结果。这种类型的聚类通常称为最小方差划分。

5.2 分级聚类法

聚类分析的三要素为：相似性测度、聚类准则和聚类算法。选定相似性测度和聚类准则后，下面的问题是用什么算法找出使准则函数取极值的最好聚类结果。有两种聚类算法，即非迭代的分级聚类算法和迭代的动态聚类算法。本节讨论分级聚类算法。

聚类分析是把 N 个没有类别标志的样本分成若干类，在极端情况下，N 个样本分成 N 类，每一个样本自成一类；另一个极端，N 个样本分在一类中。因此我们可以把问题看成是将 N 个样本划分成 c 类的划分序列。第一级划分是把样本集分成 N 个类，每个样本自成一类；第二级划分是将样本集分成 $N-1$ 个类等等，直到第 N 级划分时，把样本仅分成一个类。分级聚类就是这样一种划

分序列。这种划分序列具有如下的性质：在某一级划分时归入同一类的样本，在后面的划分时，它们永远属于同一类。生物分类就是分级聚类的一个例子。先是把许多个体集合成种，然后种集合成类，类集合成族等。分级聚类法已经渗透到了科学技术领域里的各种分类活动中。

分级聚类法可以表示成一棵树。图 5－1 是一棵具有 6 个样本的分类树。第 1 级划分时，6 个样本分成 6 类。第 2 级划分时，x^5 和 x^6 分在同一类，在以后各级划分时，它们始终属于同一类。如果把类间距离也在图上标出，那么可以帮助我们确定合理的聚类类数。如图所示，第 4 级划分的聚类结果是比较合适的，此时的类间距离为 40。若进入第 5 级划分后，类间距离大大增大，说明此时的聚类是不合适的。

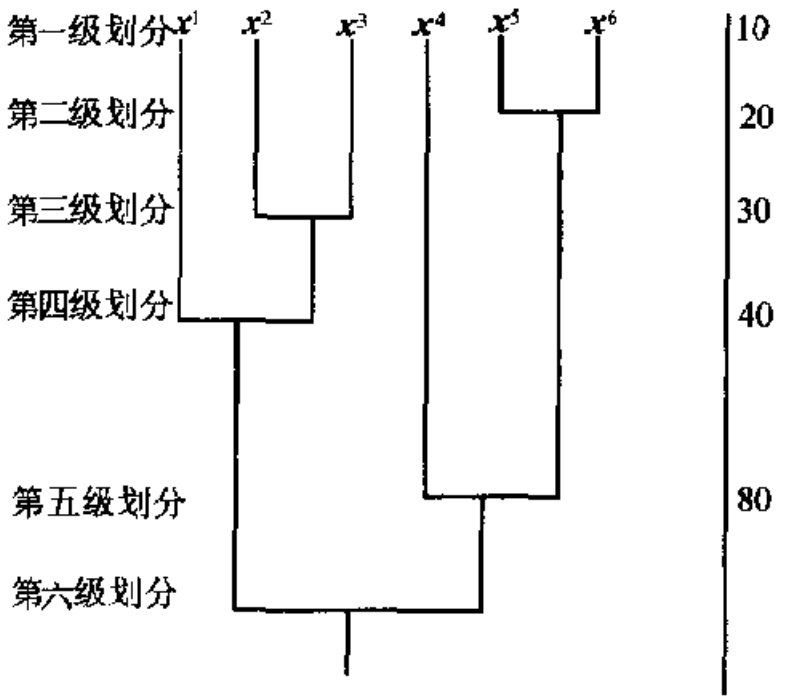


图 5－1

上面的讨论中涉及到类间距离。最常用的类间距离定义有以下几种：

(1) 最近距离

定义两个聚类 R_i 与 R_j 之间的距离 $\triangle(R_i,R_j)$ 为 R_i 类中所有样本与 R_j 中所有样本间的最小距离，即

$$\triangle(R_i,R_j)=\min_{\substack{x\in R_i\\ y\in R_j}}\{D(x,y)\}$$

(2) 最远距离

$$\triangle(R_i,R_j)=\max_{\substack{x\in R_i\\ y\in R_j}}\{D(x,y)\}$$

(3) 均值距离

$$\triangle(R_i,R_j)=D(\boldsymbol{m}_i,\boldsymbol{m}_j)$$

其中，$D()$ 可以是任何一种模式相似性测度。$\boldsymbol{m}_i$ 和 $\boldsymbol{m}_j$ 分别是 R_i 和 R_j 的均值向量。

由于定义类间距离的方法不同，得到的聚类结果可能不一致。实际问题中常用几种不同的方法进行计算，比较聚类结果，选择一个比较切合实际的聚类。

已知样本集为 $\{\boldsymbol{x}^1,\boldsymbol{x}^2,\cdots,\boldsymbol{x}^N\}$，下面我们给出分级聚类算法。

① 初始时，设置 $R_j=\{\boldsymbol{x}^j\}$，$\forall\ j\in I$，　$I=\{1,2,\cdots,N\}$；

② 在集合 $\{R_j\mid j\in I\}$ 中找到一对满足条件

$$\triangle(R_i,R_k)=\min_{\forall\ j,l\in I}\{\triangle(R_j,R_l)\}$$

的聚类 R_i 和 R_k；

③ 把 R_i 并入 R_k，去掉 R_i；

④ 把 i 从指标集 I 中删掉，若 I 的基数等于 c 时，则算法终目；否则转第②步。

上述算法中，一旦在某一级划分时，I 中的基数达到预定的类别数 c，算法终止。也可用类间距离超过预定的阈值作为算法终止的条件。

5.3 c-均值算法

动态聚类法是一种普遍采用的方法，其聚类过程如图 5-2 所示。

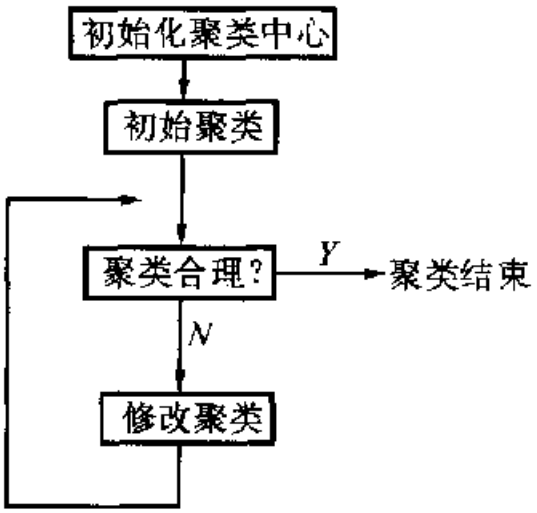


图 5-2

聚类过程中，首先选取一批有代表性的样本作为初始聚类中心，然后将样本作初始分类，接下来根据聚类准则，判断聚类是否合理，不合理就修改聚类，直至合理为止。

初始聚类中心的选择方法有以下几种：

① 根据问题的性质，凭经验从样本集中找出 c 个比较合适的样本作为初始聚类中心。

② 用前 c 个样本作为初始聚类中心。

③ 将全部样本随机地分成 c 类，计算每类的样本均值，将样本均值作为初始聚类中心。

以上这些选择办法都是带有启发性的，我们还可以设想出其它一些办法。需要注意的是，初始聚类中心的选择会影响聚类结果。

选定初始聚类中心后，可以有不同的初始分类方法：

① 按就近的原则将样本归入各聚类中心所代表的类中。

② 取一样本，将其归入与其最近的聚类中心的那一类中，并立即计算该类的样本均值，以此样本均值代替原来的聚类中心，作为新的聚类中心，然后再取下一个样本，如法炮制，直至所有的样本都归到相应的类中为止。

还可以设想出其它一些初始分类方法。初始分类后，要判断分类是否合理，不合理则修改分类。这里就涉及到聚类准则和修改分类的方法，因此出现了各种聚类算法。

本节讨论在误差平方和准则基础上的 c -均值算法。

误差平方和准则函数为

$$J=\sum_{i=1}^{c}\sum_{x^j\in R_i}\| x^j-w_i\|^2$$

式中 w_i 是类 R_i 的聚类中心。上面的准则函数也可写成

$$J=\sum_{i=1}^{c}\sum_{j=1}^{N}d_{ji}\| x^j-w_i\|^2$$

其中，N 为样本数；

$$d_{ji}=\begin{cases}1 & ,若\ x^j\in R_i\\ 0 & ,若\ x^j\notin R_i\end{cases}$$

c -均值算法可得到使误差平方和准则取得极小值时的聚类结果。

下面给出 c -均值算法：

已知样本集为 $\{x^1,x^2,\cdots,x^N\}$。

① 给定类别数 c 和允许误差 Emax，$k\leftarrow 1$；

② 初始化聚类中心 $w_i(k)$，$i=1,2,\cdots,c$；

③ 按下式修正 d_{ji}，　$i=1,2,\cdots,c;j=1,2,\cdots,N$

$$d_{ji}=\begin{cases}1,若\| x^j-w_i(k)\|^2=\min\limits_{l}\{\| x^j-w_l(k)\|^2\}\\ 0,其它\end{cases}$$

④ 按下式修正聚类中心 w_i，　$i=1,2,\cdots,c$

$$\boldsymbol{w}_i(k+1)=\sum_{j=1}^{N}d_{ji}\boldsymbol{x}^j/\sum_{j=1}^{N}d_{ji}$$

⑤ 计算误差：

$$e=\sum_{i=1}^{c}\|\boldsymbol{w}_i(k+1)-\boldsymbol{W}_i(k)\|^2$$

如果 $e<\mathrm{Emax}$，则算法结束；

否则 $k\leftarrow k+1$，转第③步。

聚类结果为：若 $d_{ji}=1$，则 $\boldsymbol{x}^j\in\omega_i$。

我们对这个 c -均值算法作些解释。d_{ji}是将第 j 个样本归入第 i 类的标志。$d_{ji}=1$，表示第 j 个标本归入第 i 类；$d_{ji}=0$，表示第 j 个样本不归入第 i 类。

在算法第③步中，修正 d_{ji}实际上就是调整第 j 个样本的类别。如果 $\|\boldsymbol{x}^j-\boldsymbol{w}_i(k)\|^2=\min_l\{\|\boldsymbol{x}^j-\boldsymbol{w}_l(k)\|^2\}$，即如果 $\boldsymbol{x}^j$ 离第 i 类的聚类中心最近，则将 d_{ji}定为 1，即将 $\boldsymbol{x}^j$ 归入第 i 类；否则不归入第 i 类，即将 d_{ji}置为 0，因此第③步是修改分类。

第③步修改分类后，应修正聚类中心，这就是第④步要完成的工作。下式计算

$$\boldsymbol{w}_i(k+1)=\sum_{j=1}^{N}d_{ji}\boldsymbol{x}^j/\sum_{j=1}^{N}d_{ji},\quad i=1,2,\cdots,c \qquad (5.1)$$

就是分别求各类的新的聚类中心。

从这个算法的第③、④步可以看出，这个算法是在每次把全部样本都调整完毕后才重新计算一次各类的聚类中心，因此是成批样本修正法。当然，我们也可以采用逐个样本修正法，每调整一个样本的类别就重新计算一次各类的聚类中心。这只要将第③步修改如下：

依次输入一样本 $\boldsymbol{x}^j$，按下式修正 d_{ji}，$i=1,2,\cdots,c$.

$$d_{ji}=\begin{cases}1,\text{若}\|\boldsymbol{x}^j-\boldsymbol{w}_i(k)\|^2=\min_l\{\|\boldsymbol{x}^j-\boldsymbol{w}_l(k)\|^2\\0,\text{其它}\end{cases}$$

显然，这里只调整了一个样本的类别。因此就成了逐个样本修

正法。

下面我们给出能完成聚类分析功能的聚类器的结构，如图 5-3 所示。

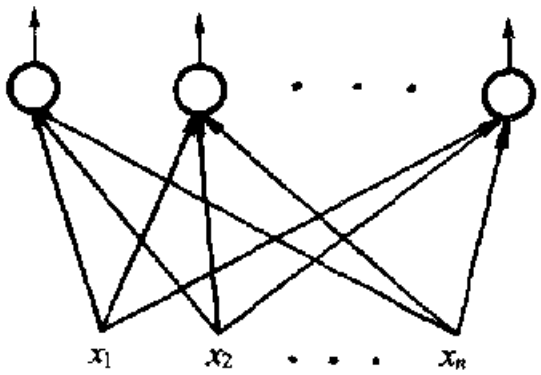


图 5-3

这是一个单层的网络，只有一层即输出层，输出节点数等于聚类的类别数。第 i 个节点的权向量对应于第 i 类的聚类中心 w_i，节点的输入不是权向量与输入样本的点积，而是样本与权向量间的距离。

如果 $\| x^j - w_m \| = \min_i \{ \| x^j - w_i \| \}$，我们称第 m 个节点获胜。按 c-均值算法，则应将样本 x^j 归入第 m 类，即令 $d_{jm}=1$，$d_{ji}=0$，$i=1,2,\cdots,c$；$i \neq m$，然后按式(5.1)修正各类的聚类中心，即修正各节点的权向量 w_i。

5.4 Kohonen 聚类网络

c-均值法中的权向量是按式(5.1)计算得到的。而神经网络中的权各量是逐步修正的，即新的权向量是在原来的权向量基础上加一个修正量得到的：

$$w_i(k+1) = w_i(k) + \triangle w_i(k)$$

另外，我们前面介绍的神经元的输入是权向量与样本的点积，而这里，节点的输入是样本与权向量的距离，也不太符合神经元组合输入信号的规律，因此，如果要用神经网络执行 c -均值算法的话，必须将算法作一些修改。

c -均值算法采用的准则函数为误差平方和准则函数：

$$J = \sum_{i=1}^{c} \sum_{x^j \in R_i} \| x^j - w_i \|^2$$

这里，聚类中心 w_i 是变量，不同的 w_i 得到的 J 值是不同的。确定了 w_i 后，按就近原则，所有的样本就各归其类。因此关键是确定 w_i。我们所求的是使 J 取得极小值的 w_i。这是一个优化问题，我们采用梯度下降法。J 对 w_i 的梯度为

$$\nabla J = -2(x - w_i)$$

这样，权向量的修正公式为

$$w_i(k+1) = w_i(k) + \eta(x - w_i(k))$$

为了减小计算量，或许是模仿生物神经网络的学习规律，算法只对获胜节点及其邻域内的节点的权向量进行修正。

Kohonen 聚类网络算法如下：

① 给定类别数 c，邻域大小 l，步长 η，允许误差 Emax，并初始化权向量 $w_i, i=1,2,\cdots,c$；

② 输入一样本 x^j，计算 $\mu_{ji} = \| x^j - w_i(k) \|$，$i=1,2,\cdots,c$；并按升序排列为

$$\mu_{jm_1}, \mu_{jm_2}, \cdots, \mu_{jm_l}, \cdots, \mu_{jm_c}$$

节点 m_1 为获胜节点，令 $d^j_{jm_l}=1$，$d^j_{ji}=0$，$i=1,2,\cdots,c$；$i \neq m_1$；

③ 修正获胜节点及其邻域内节点的权向量：

$$w_i(k+1) = w_i(k) + \eta(x^j - w_i(k)), i = m_1, m_2, \cdots, m_l.$$

④ 计算误差 $e(k+1) = \sum_{i=1}^{c} \| w_i(k+1) - w_i(k) \|^2$；

如果 $e(k+1) < \text{Emax}$，算法结束；否则，$k \leftarrow k+1$，转第②步。

在该算法中，邻域大小 l 随迭代次数增大而逐渐减小。

上述算法的第②步中，计算 μ_{ji}，目的是判断输入样本 x^j 与哪一个聚类中心最匹配，采用的相似性测度为距离。也可以用角度相似性函数作为相似性测度：

$$S(w_i, x^j) = \left(\frac{w_i}{\| w_i \|}\right)^T \left(\frac{x^j}{\| x^j \|}\right)$$

将权向量和样本作归一化处理后，节点的输入又成为归一化的权向量与输入样本的点积了。

这样，上述算法的第②步可改为输入一样本 x^j。若 $w_m^T x^j = \max\{w_i^T x^j\}$，则第 m 个节点获胜。令 $d_{jm}=1, d_{ji}=0, i=1,2,\cdots,c; i\neq m$.

上述 c -均值算法是在类别数给定的情况下进行的。当类别数未知的情况下，我们在使用 c -均值算法时，可以假设类别数是逐步增加的，显然准则函数 J 是随 c 的增加而单调减小的。当样本集表现为 $\hat{c}$ 个很集中的聚类时，J 随着从一个聚类增加到 $\hat{c}$ 个聚类而迅速减小。当 c 再继续增加时，相当于将本来是较密集的群再行分开，因此 J 虽有所减小，但减小的速度将比较缓慢，直到 $c=N$ 时，$J=0$. 作一条 $J-c$ 的曲线，如图 5-4 所示，其拐点对应的类数就比较接近于最优的聚类数。例如图 5-4 中的拐点 A 所对应的 $\hat{c}=3$ 为较合适的聚类类别数。

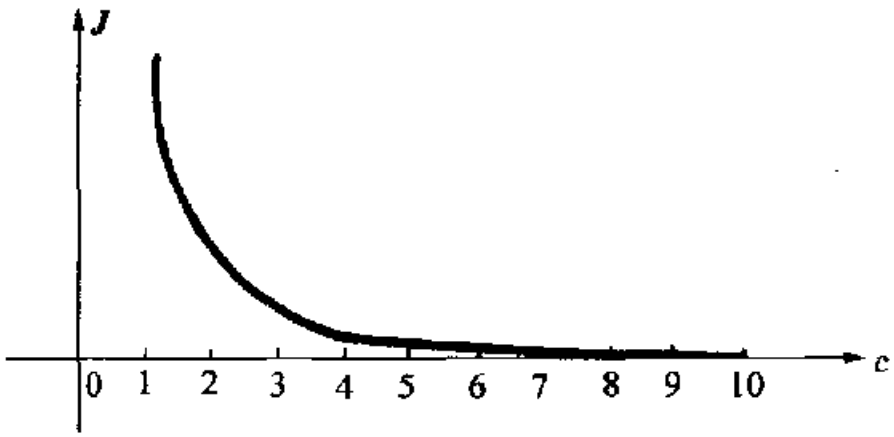


图 5-4

遗憾的是，并非所有的情况都能找到这样的拐点 A。当无明显的转折点时，这种选择最佳聚类类别数的方法失效。

5.5 ISODATA 算法

ISODATA 算法与 c-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来决定的。但 ISODATA 算法还加入一些试探步骤，能自动地进行类的合并和分裂，从而得到类数较合理的聚类结果。

下面我们给出 ISODATA 算法：

已知样本集为 $\{\boldsymbol{x}^1,\boldsymbol{x}^2,\cdots,\boldsymbol{x}^N\}$。

(1) 规定下列控制参数：

K＝期望得到的聚类数；

Q_N＝一个聚类中的最少样本数；

Q_S＝标准偏差参数；

Q_C＝合并参数；

L＝每次迭代允许合并的最大聚类对数；

I＝允许迭代的次数。

设初始的聚类数 c 和初始的聚类中心 $\boldsymbol{w}_i$，$i=1,2,\cdots,c$.

(2) 按照下述关系

若

$$\|\boldsymbol{x}-\boldsymbol{w}_i\| < \|\boldsymbol{x}-\boldsymbol{w}_j\|, j=1,2,\cdots,c; j\neq i$$

则

$$\boldsymbol{x}\in R_i$$

将所有样本分到各个聚类中去。R_i 是第 i 个聚类，其中心为 $\boldsymbol{w}_i$.

(3) 若有任何一个 R_i，其基数 $N_i<Q_N$，则舍去 R_i，并令 $c=c-1$。

(4) 按照下列关系

$$w_i = \frac{1}{N_i}\sum_{x \in R_i} x, \quad i = 1,2,\cdots,c$$

计算更新的均值向量。式中 N_i 是第 i 个聚类的样本数目(基数)。

(5) 计算 R_i 中的所有样本距其相应的聚类中心 w_i 的平均距离 $\bar{D}_i$

$$\bar{D}_i = \frac{1}{N_i}\sum_{x \in R_i} \| x - w_i \|, \quad i = 1,2,\cdots,c$$

(6) 计算所有样本距其相应的聚类中心的平均距离

$$\bar{D} = \frac{1}{N}\sum_{i=1}^{c} N_i \bar{D}_i$$

(7) (*a*)若这是最后一次迭代(由参数 I 确定),则置 $\theta_C = 0$,转第(11)步;

(*b*) 若 $c \leqslant K/2$,转第(8)步;

(*c*) 若是偶数次迭代,或若是 $c \geqslant 2K$,则转第(11)步。否则,往下进行。

(8) 对每一个聚类 R_i,用下列公式求标准偏差 $\sigma_i = (\sigma_{i1}, \sigma_{i2}, \cdots, \sigma_{in})^T$

$$\sigma_{ij} = \sqrt{\frac{1}{N_i}\sum_{x^k \in R_i} (x_j^k - w_{ij})^2}$$

式中,x_j^k 是第 k 个样本的第 j 个分量;w_{ij}是第 i 个聚类中心的第 j 个分量;σ_{ij}是第 i 个聚类的标准偏差的第 j 个分量;n 是样本 x 的维数。

(9) 对每一个聚类,求出具有最大标准偏差的分量 $\sigma_{\max}$,$i = 1, 2, \cdots, c$.

(10) 若对任一个 σ_{imax},$i = 1,2,\cdots,c$,存在 $\sigma_{\max} > \theta_S$,并且有:

(*a*) $D_i > \bar{D}$且 $N_i > 2(\theta_N + 1)$;

或(*b*) $c \leqslant K/2$.

则把 R_i 分裂成两个聚类,其中心相应为 w_i^+ 和 w_i^-,把原来的 w_i 取

消，且令 $c=c+1$。w_i^+ 和 w_i^- 的计算如下：

给定一个 α 值，$0<\alpha\leqslant 1$，令

$$\gamma_i = \alpha\sigma_{i\max}$$

则 $w_i^+=w_i+\gamma_i$，　$w_i=w_i-\gamma_i$

式中的 α 值应选得使 R_i 中的样本到 w_i^+ 和 w_i^- 的距离不同，但又应使 R_i 中的样本仍然在这两个新的集合中。

(11) 对于所有的聚类中心，计算两两之间的距离

$$D_{ij} = \| w_i - w_j \|, \qquad i = 1,2,\cdots,c-1$$
$$j=i+1,i+2,\cdots,c$$

(12) 比较 D_{ij} 和 θ_C，将 $D_{ij}<\theta_C$ 的值按上升次序排列：

$$D_{i_1j_1} < D_{i_2j_2} < \cdots < D_{i_lj_l}, \quad l \leqslant L$$

(13) 从最小的 $D_{i_1j_1}$ 开始，将距离为 $D_{i_lj_l}$ 的两个聚类中心 w_{i_l} 和 W_{j_l} 合并，得新的聚类中心

$$w_l = \frac{1}{N_{i_l}+N_{j_l}}\left(N_{i_l}w_{i_l} + N_{j_l}w_{j_l}\right)$$

并令 $c=c-1$.

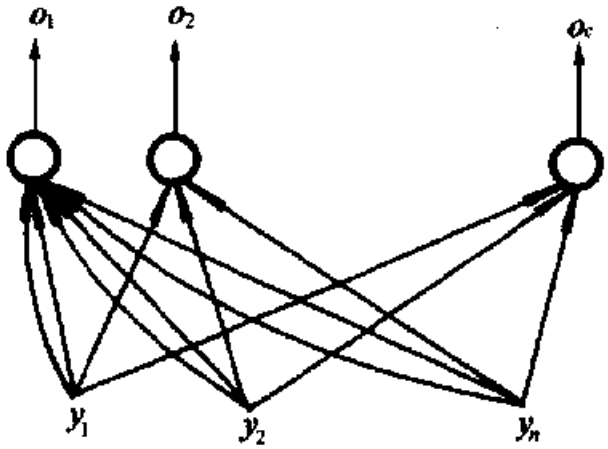
(14) 若这是最后一次迭代，则算法终止。否则，若根据经验需要改变参数，则转第(1)步；若不需要改变参数，则转第(2)步。

本步中，还应将迭代计数器加 1.

5.6　自适应聚类网络

Kohonen 聚类网络与 c -均值算法一样，聚类的类别数 c 必须预先确定。本节要介绍的一种聚类网络，类别数不需预先确定。该网络在聚类过程中能依样本分布情况自动地修改类别数，以适应样本分布的实际情况，所以我们称之为自适应聚类网络。该网络的结构如图 5－5 所示。

该网络有两组权向量，一组是由下而上的权向量 $\boldsymbol{b}_k$，$k=1,2$，



(从上而上的连接权值b_i只在一个节点上标出)

图 5-5

…,c;另一组是由上而下的权向量t_k,$k=1,2,\cdots,c$。权向量t_k相当于第k类的聚类中心。

这种网络与 Kohonen 网络一样,采用竞争学习规则。假设输入样本x^j及其归一化样本y^j,输出节点展开竞争,结果,输出节点m获胜:

$$o_m = \max_k \{b_k^T y^j\}$$

然后修正获胜节点的权向量b_m,其它输出节点的权向量b_k不修正。对于获胜节点,修正权向量b_m的策略是使b_m与y^j趋于一致。

采用这种学习方法,常常出现这样一种情况,如果在相隔较远的两个时间点上两次输入同一个样本,而在此期间,输入过其它样本,那么第二次分类结果会出现与第一次分类结果不一致的现象。这说明第一次分类后新学习得到的记忆内容有可能冲掉原有的学习记忆内容,从而导致了第二次分类的错误。这也说明简单的竞争学习规则不能保证记忆具有足够的牢固性。

为了解决这个矛盾,可以在竞争学习算法中再加上一个由上而下的自稳机制。在网络中增加了由上而下的权向量t_k,$k=1,2$,…,c。t_k相当于第k类的聚类中心。假设输入样本x^j后,输出节点m获胜,那么该不该把样本x^j归入第m类,需要进行测试。测试的

办法是，计算样本 x^j 与聚类中心 t_m 之间的距离 $\| x^j - t_m \|$。如果距离小于警戒参数 ρ，应将 x^j 归入第 m 类，并修改聚类中心 t_m. 否则，再看看是否该归在其它类中。如果将 x^j 归在现有的所有各类都不合适，则另辟新类，增加一个新的输出节点。这样，网络的结构能依样本分布的不同而不同。

下面给出自适应聚类网络算法。

(1) 设初始类别数 $c=2$，设定警戒参数 ρ 和步长 η. 初始化由上而下的权向量 $t_k(1)$，并令 $b_k(1)=t_k(1)/\| t_k(1) \|$，$k=1,2$.

(2) 输入一样本 x^j 及其归一化样本 y^j.

(3)找出获胜节点 m：

$$o_m = \max_{\substack{k=1,2,\cdots,c \\ k\text{不是已屏蔽节点}}} \{ b_k^T(l) y^j \}$$

(4) 警戒测试，计算 $s = \| x^j - t_m(l) \|$.

如果　$s<\rho$，表示警戒测试通过；

如果　$s\geqslant\rho$，表示警戒测试未通过。

① 如果获胜节点通过警戒测试，转第(6)步；

② 如果获胜节点没有通过警戒测试，则屏蔽该获胜节点，返回第(3)步，选择新的获胜节点；

③ 如果现有的 c 个输出节点都没有通过警戒测试，则转第(5)步。

(5) $c \leftarrow c+1$，增加一个输出节点，并初始化该节点的权向量，转第(7)步。

(6) 修正权向量

$$t_m(l+1) = t_m(l) + \eta(x^j - t_m(l))$$

$$b_m(l+1) = t_m(l+1)/\| t_m(l+1) \|$$

(7) 还要学习吗？需要，则转第(2)步，否则算法结束。

上述算法中，警戒参数对聚类结果影响很大。ρ 越小，聚类结果，类别越多；ρ 越大，类别越少。ρ 在实际应用中要适当选择。

5.7 聚类分析的遗传算法方法

在各种聚类算法中，c-均值聚类算法的应用最为广泛。然而，c-均值算法本质上是一种局部搜索技术，它采用了所谓的梯度下降法来寻找最优解。因此容易陷入局部最小值，而得不到全局最优解。遗传算法已成功地应用于各种优化问题，遗传算法仿效了生物的进化过程。通过一群基因串一代又一代地繁殖和交换，遗传算法能搜索到多个局部极值，从而增加了找到全局最优解的可能性。本节给出一种基于遗传算法的聚类分析方法，目的是提高聚类算法找到全局最优的可能性。

将遗传算法同聚类问题联系起来首先要解决：

① 如何构造适应度函数来度量每条基因串对问题的适应程度。如果某条基因串代表着良好的聚类结果，其适应度就高；反之，其适应度就低。

② 如何将问题的解编码到基因串中去。

c-均值算法采用的聚类准则函数为误差平方和函数：

$$J=\sum_{i=1}^{c}\sum_{j=1}^{N}d_{ji}\,\|\boldsymbol{x}^{j}-\boldsymbol{w}_{i}\|$$

其中，N 为样本数；$\boldsymbol{w}_i$ 为第 i 类的聚类中心；d_{ji} 表示第 j 个样本属于第 i 类（$d_{ji}=1$）或不属于第 i 类（$d_{ji}=0$）.

我们定义适应度函数为

$$f=\frac{1}{1+J}$$

我们可以采用两种编码方案。在第一种方案中，设 N 个样本要分成 c 类，用基因串 $A=\{\alpha_1,\alpha_2,\cdots,\alpha_j,\cdots,\alpha_N\}$ 来表示某一分类结果，其中 α_j 取值为 k 时（$1\leqslant k\leqslant c$），表示第 j 样本归入第 k 类。将 α_i 用二进制编码后，实际的搜索空间中的点数将大于 c^N 个点。第二种方案，用基因串 $B=\{\beta_1,\beta_2,\cdots,\beta_i,\cdots\beta_c\}$ 来表示一种分类结

果。其中 β_i 是第 i 类聚类中心的编码。可将 n 维聚类中心的每一维量化为 256 级，这样每一个聚类中心用 $8\times n$ 个比特位表示，基因串的长度就成了 $8\times n\times c$ 个比特位。此时的搜索空间有 $256^{n\times c}$ 个点。

遗传算法的流程大致如下：

① 设定每代中基因串的个数 l，取 l 个基因串作为第一代。

② 用适应度函数计算每条基因串的适应度，并根据适应度将其复制到配对池中。

③ 根据 P_c 和 P_m 对配对池中的基因串进行交叉和变异操作，产生新的一代。

④ 如果达到设定的繁衍代数，算法结束。在最后一代中找出最好的基因串，并由此得到最好的聚类结果。否则，转第②步，继续繁衍下一代。

第六章　模糊模式识别

模糊(Fuzzy)集合的概念是由美国控制论专家扎德(Zadeh)首次提出的。1965年，他发表了奠基性的论文“Fuzzy Sets”，标志着模糊数学的诞生。

在人类的日常活动中，几乎处处是模糊现象或模糊概念。例如，“年青人”，“大胡子”，“性能良好”等等。是否可以这样讲，世界上的现象，模拟性是绝对的，而清晰性或精确性是相对的。人脑中所形成的概念几乎都是模糊的，由此形成的判断和推理也都是模糊的。

“概念”是人们常使用的名词。例如“男人”就是一个概念。一个概念有其内涵和外延。所谓内涵是指符合此概念所具有的共同属性，例如男人这一概念就是男人所有的特征。而外延指的是符合此概念的全体对象所组成的集合，男人这一概念的外延就是全体男人。

集合可以表现概念，集合之间还有运算和变换，这些运算和变换可以表现判断和推理。因此建立在集合论基础上的现代数学便成了可以描述和表现各门学科的形式语言和系统。

集合论是康托创立的。他的主要思想方法之一就是概括原则。所谓概括原则是指任给一个性质 p，便能把所有满足性质 p 的对象，也仅由具有性质 p 的对象汇集在一起构成一个集合，用符号来表示就是

$$A = \{a \mid p(a)\}$$

其中，a 表示 A 的任何一个元素；$p(a)$ 表示 a 具有性质 p；{　} 表示把所具有性质 p 的 a 汇成一个集合。

这里要强调指出，康托要求组成集合的那些对象是确定的，彼

此有区别的。实际上是要求用以构成集合的性质 p 必须是界线分明的，即要求任何对象要么具有性质 p，要么不具有性质 p. 按照这一要求，集合所表明的概念，真就是真，假就是假，形成一种二值逻辑。而人脑中的概念，几乎都是没有明确外延的模糊概念。那么二值逻辑能不能处理模糊概念呢？不能！二值逻辑把真与假绝对化，只允许有真(记为 1)和假(记为 0)这两个值。对于模糊概念，仅用这两个值是不够的，必须在 1 与 0 之间，采用其它中间的逻辑值来表示不同的真确程度。这样，模糊集合论就应运而生了。

模式识别是让计算机模拟人的思维方式对客体进行分类判别。而客体的特征常常带有某些模糊性，因此模式识别与模糊数学的关系很紧密。将模糊理论引入模式识别会提高模式识别系统的柔性处理能力。

6.1 模糊数学的基本知识

6.1.1 模糊集合

在实际问题中，集合总是作为某个概念的外延而出现的。

定义 6.1 被讨论的全体对象称为论域。论域中的每个对象称为元素。给定论域 U，U 中一部分元素的全体，称为 U 上的一个集合。

1. 模糊集合的定义

定义 6.2 所谓给定了论域 U 上的一个模糊子集 $\underset{\sim}{A}$ 是指：对任何 $u\in U$，都指定了一个数 $\mu_{\underset{\sim}{A}}(u)\in[0,1]$ 与之对应，称 $\mu_{\underset{\sim}{A}}(u)$ 为 u 对 $\underset{\sim}{A}$ 的隶属度。这意味着构造了一个映射

$$\mu_{\underset{\sim}{A}}:\quad U\to[0,1]$$
$$u\to\mu_{\underset{\sim}{A}}(u)$$

这个映射称为 $\underset{\sim}{A}$ 的隶属函数。

模糊集合完全由其隶属函数所刻划。特别当 $\mu_{\tilde{A}}(u)=\{0,1\}$ 时，$\mu_{\tilde{A}}$ 蜕化为一个普通集合的特征函数，$\tilde{A}$ 便蜕化成一个普通集合。普通集合是模糊集合的特殊情况：

$$A=\{u\in U\mid \mu_{\tilde{A}}(u)=1\}$$

如果把论域 U 上全体子集所组成的集合记为 $P(U)$，把论域 U 上全体模糊子集所组成的集合记作 $F(U)$，则 $F(U)\supset P(U)$

例 6.1　如图 6－1 所示，a,b,c,d 是 4 个小块，它们组成论域 U。“圆块块”是 U 上的一个模糊子集，记为 $\tilde{A}$，它的隶属函数为

$$\mu_{\tilde{A}}:\quad U\to[0,1]$$

$$a\to 1$$

$$b\to 0.75$$

$$c\to 0.5$$

$$d\to 0.25$$

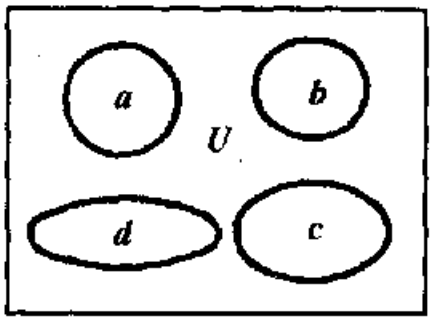


图 6－1

可采用扎德的记法，$\tilde{A}$ 表示为

$$\tilde{A}=\frac{1}{a}+\frac{0.75}{b}+\frac{0.5}{c}+\frac{0.25}{d}$$

上式右端并非分式求和，它仅仅是一种记号，分母位置放的是论域 U 中的元素，分子位置放的是相应元素的隶属度。当某一元素的隶属度为 0 时，那一项可以省略。

例 6.2　以人类的年龄作为论域 $U=[0,100]$，则“年青”可以表示为 U 上的一个模糊子集，记作 $\tilde{Y}$，其隶属函数为

$$\mu_{\tilde{Y}}(u)=\begin{cases}1, & 0\leqslant u\leqslant 25\\ \left[1+\left(\dfrac{u-25}{5}\right)^2\right]^{-1}, & 25<u\leqslant 100\end{cases}$$

“年老”也可表示为 U 上一个模糊子集，记为 $\tilde{O}$，隶属函数为

$$\mu_{\tilde{O}}(u)=\begin{cases}0, & 0\leqslant u\leqslant 50\\ \left[1+\left(\dfrac{u-50}{5}\right)^{-2}\right]^{-1}, & 50<u\leqslant 100\end{cases}$$

扎德的记法可推广到任何论域 U 上

$$\underset{\sim}{A}=\int_{u\in U}(\mu_{\underset{\sim}{A}}(u)/u)$$

此处积分号不是通常积分的意思，而是表示各个元素与其隶属度对应关系的一个总括。

2. 模糊集合的运算

定义 6.3　设 $\underset{\sim}{A},\underset{\sim}{B}\in F(U)$，规定模糊集合之间的包含 $\underset{\sim}{A}\supset\underset{\sim}{B}$，相等 $\underset{\sim}{A}=\underset{\sim}{B}$，并 $\underset{\sim}{A}\cup\underset{\sim}{B}$，交 $\underset{\sim}{A}\cap\underset{\sim}{B}$，以及余运算 $\underset{\sim}{A}^c$ 如下：

包含　$\underset{\sim}{A}\supset\underset{\sim}{B}\Leftrightarrow(\forall\ u\in U)(\mu_{\underset{\sim}{A}}(u)\geqslant\mu_{\underset{\sim}{B}}(u))$

相等　$\underset{\sim}{A}=\underset{\sim}{B}\Leftrightarrow\underset{\sim}{A}\supset\underset{\sim}{B}$ 且 $\underset{\sim}{B}\supset\underset{\sim}{A}$

并　$\underset{\sim}{C}=\underset{\sim}{A}\cup\underset{\sim}{B}\Leftrightarrow(\forall\ u\in U)(\mu_{\underset{\sim}{C}}(u)=\mu_{\underset{\sim}{A}}(u)\vee\mu_{\underset{\sim}{B}}(u))$

交　$\underset{\sim}{D}=\underset{\sim}{A}\cap\underset{\sim}{B}\Leftrightarrow(\forall\ u\in U)(\mu_{\underset{\sim}{D}}(u)=\mu_{\underset{\sim}{A}}(u)\wedge\mu_{\underset{\sim}{B}}(u))$

余　$\underset{\sim}{E}=\underset{\sim}{A}^c\Leftrightarrow(\forall\ u\in U)(\mu_{\underset{\sim}{E}}(u)=1-\mu_{\underset{\sim}{A}}(u))$

其中，符号“∨”表示取最大值；“∧”表示取最小值。

例 6.3　在例 6.1 中再定义模糊子集“方块块”$\underset{\sim}{B}$：

$$\underset{\sim}{B}=\frac{0.1}{a}+\frac{0.3}{b}+\frac{0.5}{c}+\frac{0.7}{d}$$

这时，有

$$\underset{\sim}{A}\cup\underset{\sim}{B}=\frac{1\vee0.1}{a}+\frac{0.75\vee0.3}{b}+\frac{0.5\vee0.5}{c}+\frac{0.25\vee0.7}{d}$$

$$=\frac{1}{a}+\frac{0.75}{b}+\frac{0.5}{c}+\frac{0.7}{d}$$

$$\underset{\sim}{A}\cap\underset{\sim}{B}=\frac{1\wedge0.1}{a}+\frac{0.75\wedge0.3}{b}+\frac{0.5\wedge0.5}{c}+\frac{0.25\wedge0.7}{d}$$

$$=\frac{0.1}{a}+\frac{0.3}{b}+\frac{0.5}{c}+\frac{0.25}{d}$$

$$\underset{\sim}{A}^c=\frac{1-1}{a}+\frac{1-0.3}{b}+\frac{1-0.5}{c}+\frac{1-0.25}{d}$$

$$=\frac{0}{a}+\frac{0.7}{b}+\frac{0.5}{c}+\frac{0.75}{d}$$

6.1.2 模糊关系

设 U,V 是两个论域，记

$$U \times V = \{(x,y) | x \in U, y \in V\}$$

为 U 与 V 的笛卡尔乘积集。

1. 模糊关系的定义

定义 6.4 设 U,V 是两个论域，由 U,V 作出一个新的论域 $U \times V$. $U \times V$ 上任何一个模糊集合 $\underset{\sim}{R} \in (U \times V)$ 都称为 U 与 V 之间的模糊关系，即

$$\mu_{\underset{\sim}{R}}: \quad U \times V \to [0,1]$$

$$(u,v) \to \mu_{\underset{\sim}{R}}(u,v)$$

其中 $\mu_{\underset{\sim}{R}}(u,v)$ 称为 u 与 v 关于 $\underset{\sim}{R}$ 的关系强度。当 $U=V$ 时，称 $\underset{\sim}{R}$ 为 U 上的模糊关系。

例 6.4 在实数集上规定一个“远远大于”关系，记作“$>>$”

$$\mu_{>>}(x,y) = \begin{cases} 0, & x \leqslant y \\ [1 + \dfrac{100}{(x-y)^2}]^{-1}, & x > y \end{cases}$$

例 6.5 用模糊关系表示苹果、书和桃子的相似关系。本例，$U=V=$ {苹果，书，桃子}。

设用专家评分的办法给出它们的相似程度，如表 6-1。

表 6-1

相似程度	苹果	书	桃子
苹果	1	0	0.5
书	0	1	0
桃子	0.5	0	1

这个模糊关系可用矩阵形式表示

$$R=\begin{pmatrix}1 & 0 & 0.5\\ 0 & 1 & 0\\ 0.5 & 0 & 1\end{pmatrix}$$

一般地，对于有限论域 $U=\{u_1,u_2,\cdots,u_n\}$，$V=\{v_1,v_2,\cdots,v_m\}$ 之间的模糊关系 $\underset{\sim}{R}$ 可用 n 行 m 列的模糊矩阵 R 表示

$$R=(r_{ij})_{n\times m}$$

其中，$r_{ij}=\mu_{\underset{\sim}{R}}(u_i,v_j)$。

2. λ 截关系

设 $\underset{\sim}{R}$ 为 $U\times V$ 上的一个模糊关系，其模糊矩阵为 R，对任意 $\lambda\in[0,1]$，记 $R_\lambda=(\lambda_{r_{ij}})$，其中

$$\lambda_{r_{ij}}=\begin{cases}1, & r_{ij}\geqslant\lambda\\ 0, & r_{ij}<\lambda\end{cases}$$

称 R_λ 为 $\underset{\sim}{R}$ 的 λ 截矩阵，即 R_λ 为 $\underset{\sim}{R}$ 的 λ 截关系。R_λ 是 U 与 V 之间的一个普通关系。

例 6.6　模糊关系如上例：

$$R=\begin{pmatrix}1 & 0 & 0.5\\ 0 & 1 & 0\\ 0.5 & 0 & 1\end{pmatrix}$$

则

$$R_1=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$$

$$R_{0.5}=\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}$$

$$R_0=\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$$

3. 模糊关系的合成运算

定义 6.5　设 U,V,W 是三个论域，$\underset{\sim}{Q}\in F(U\times V)$，$\underset{\sim}{R}\in F(V\times W)$，由 $\underset{\sim}{Q}$ 与 $\underset{\sim}{R}$ 作出一个新的模糊关系 $\underset{\sim}{Q}\circ\underset{\sim}{R}\in F(U\times W)$，称为 $\underset{\sim}{Q}$ 与 $\underset{\sim}{R}$ 的合成模糊关系，它的隶属函数规定为

$$\mu_{\underset{\sim}{Q}\circ\underset{\sim}{R}}(u,w)=\bigvee_{v\in V}(\mu_{\underset{\sim}{Q}}(u,v)\wedge\mu_{\underset{\sim}{R}}(v,w))$$

当 $U=V=W$，$\underset{\sim}{R}\in F(U\times U)$ 时，记

$$R^2=R\circ R$$

$$R^n=R^{n-1}\circ R,\quad n=2,3,\cdots$$

当 U,V,W 均为有限论域时，即 $U=\{u_1,u_2,\cdots,u_n\}$，$V=\{v_1,v_2,\cdots v_m\}$，$W=\{w_1,w_2,\cdots,w_l\}$，$\underset{\sim}{Q}$，$\underset{\sim}{R}$ 和 $\underset{\sim}{S}=\underset{\sim}{Q}\circ\underset{\sim}{R}$ 均可表示为矩阵形式：

$$Q=(q_{ij})_{n\times m},\ R=(r_{jk})_{m\times l},\ S=(s_{ik})_{n\times l}$$

其中

$$s_{ik}=\bigvee_{j=1}^{m}(q_{ij}\wedge r_{jk})$$

例 6.7　设

$$Q=\begin{pmatrix}0.3 & 0.7 & 0.2\\ 1 & 0 & 0.4\\ 0 & 0.5 & 1\\ 0.6 & 0.7 & 0.8\end{pmatrix},\quad R=\begin{pmatrix}0.1 & 0.9\\ 0.9 & 0.1\\ 0.6 & 0.4\end{pmatrix}$$

则

$$S=Q\circ R=\begin{pmatrix}0.7 & 0.3\\ 0.4 & 0.9\\ 0.6 & 0.4\\ 0.7 & 0.6\end{pmatrix}$$

其中　$s_{11}=(q_{11}\wedge r_{11})\vee(q_{12}\wedge r_{21})\vee(q_{13}\wedge r_{31})$

$=(0.3\wedge 0.1)\vee(0.7\wedge 0.9)\vee(0.2\wedge 0.6)$

$=0.1\vee 0.7\vee 0.2$

$=0.7$

类似地可以求出其它的 s_{ik}.

4. 模糊等价关系和模糊相似关系

定义 6.6　称模糊关系 $\underset{\sim}{R}\in F(U\times U)$ 为 U 上的一个模糊等价关系，如果 $\underset{\sim}{R}$ 满足条件

(1)自反性　$(\forall\ u\in U)(\mu_{\underset{\sim}{R}}(u,u)=1)$，即 $R\supset I$；

(2)对称性　$(\forall\ u,v\in U)(\mu_{\underset{\sim}{R}}(u,v)=\mu_{\underset{\sim}{R}}(v,u))$，即 $R^T=R$；

(3)传递性　$\forall\ (u,w)\in U\times U$，有

$$\bigvee_{u\in V}(\mu_{\underset{\sim}{R}}(u,v)\wedge\mu_{\underset{\sim}{R}}(v,w))\leqslant\mu_{\underset{\sim}{R}}(u,w)，即\ R\circ R\subset R$$

定义 6.7　称 $\underset{\sim}{R}\in F(U\times U)$ 为 U 上的模糊相似关系，如果 $\underset{\sim}{R}$ 满足条件

(1)自反性　$(\forall\ u\in U)(\mu_{\underset{\sim}{R}}(u,u)=1)$，即 $R\supset I$；

(2)对称性　$(\forall\ u,v\in U)(\mu_{\underset{\sim}{R}}(u,v)=\mu_{\underset{\sim}{R}}(v,u))$，即 $R^T=R$.

从定义 6.6 和定义 6.7 可能看出，如果模糊相似矩阵 R 满足条件 $R\circ R\subset R$，则 $R\circ R$ 所对应模糊关系就成了模糊等价关系。由此可见，为了从模糊相似关系得到模糊等价关系，可将模糊相似矩阵自乘，即 $R\circ R\triangleq R^2$，$R^2\circ R^2\triangleq R^4$，…，直到 $R^{2k}=R^k$. 至此，R^k 便是模糊等价矩阵，R^k 所对应的模糊关系为模糊等价关系。

下面给出一个重要定理：

定理　$\underset{\sim}{R}\in F(U\times U)$ 是 U 上的模糊等价关系，当且仅当，对任何 $\lambda\in[0,1]$，R_λ 都是 U 上的普通等价关系。

我们介绍等价关系的目的是为了将集合划分成若干等价类。有了上述定理后，我们就可以将根据模糊等价关系矩阵 R 划分集合转化为根据相应的普通等价关系 R_λ 来划分集合。

设 $\underset{\sim}{R}$ 是论域 U 上的模糊等价关系，λ 从 1 下降至 0，依次截得等价关系 R_λ，它们都将 U 作了分类。由于满足条件

$$\lambda_2\leqslant\lambda_1\Rightarrow R_{\lambda_2}\supset R_{\lambda_1}$$

所以，$\forall\ u,v\in U$，若 u 与 v 相对于 R_{λ_1} 来说属于同一类，$(u,v)\in R_{\lambda_1}$，则 $(u,v)\in R_{\lambda_2}$，即 u 与 v 相对于 R_{λ_2} 来说也属于同一类。这意味

着由 R_{λ_2} 所得到的分类是由 R_{λ_1} 所得到的分类的加粗。

当 λ 从 1 下降至 0 时，分类由细变粗，逐渐归并，形成一个分级聚类树。

例 6.8　设 $U=\{u_1,u_2,u_3,u_4,u_5\}$，给定 U 上一个模糊等价关系

$$R=\begin{pmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{pmatrix}$$

让 λ 从 1 变到 0，写出 R_λ，再按 R_λ 分类，这里 u_i 与 u_j 归为同一类是指 $\lambda_{ij}=1$.

$$R_1=\begin{pmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{pmatrix}$$

相应的分类为

$$\{u_1\},\{u_2\},\{u_3\},\{u_4\},\{u_5\}。$$

$$R_{0.8}=\begin{pmatrix}1 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{pmatrix}$$

相应的分类为

$$\{u_1,u_3\},\{u_2\},\{u_4\},\{u_5\}。$$

这时，u_1 与 u_3 归为一类。

$$R_{0.6}=\begin{pmatrix}1&0&1&0&0\\0&1&0&0&0\\1&0&1&0&0\\0&0&0&1&1\\0&0&0&1&1\end{pmatrix}$$

相应的分类为

$$\{u_1,u_3\},\{u_2\},\{u_4,u_5\}。$$

这时，u_4 与 u_5 又并为一类。

$$R_{0.5}=\begin{pmatrix}1&0&1&1&1\\0&1&0&0&0\\1&0&1&1&1\\1&0&1&1&1\\1&0&1&1&1\end{pmatrix}$$

相应的分类为

$$\{u_1,u_3,u_4,u_5\},\{u_2\}。$$

这时，u_1,u_3,u_4,u_5 并为一类，u_2 单独一类。

$R_{0.4}$为全 1 矩阵，相应的分类为$\{u_1,u_2,u_3,u_4,u_5\}$。

于是我们得出分级聚类树，如图 6－2 所示。

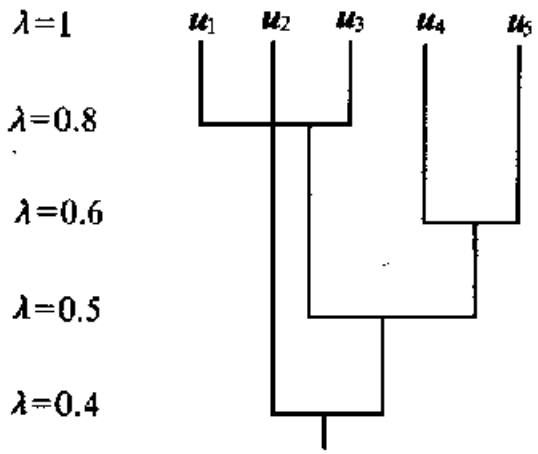


图 6－2

6.2 模糊识别的直接方法

我们还是从细胞识别讲起。假设将所有细胞分为四类，用模糊数学的术语说，在全体细胞上构造四个模糊子集：癌细胞 $\underset{\sim}{M}$，重度核异质细胞 $\underset{\sim}{N}$，轻度核异质细胞 $\underset{\sim}{R}$和正常细胞 $\underset{\sim}{K}$. 现在要识别细胞 x 属于哪一类，根据什么原则来判别？一个很显然的原则：如果细胞 x 属某一类的隶属度最大，则决策细胞 x 属该类。这就是最大隶属原则。

最大隶属原则　设 $\underset{\sim}{A}_1,\underset{\sim}{A}_2\cdots\cdots,\underset{\sim}{A}_m\in F(U)$，$x$ 是 U 中的一个元素：

若　$\mu_{\underset{\sim}{A}_i}(x)>\mu_{\underset{\sim}{A}_j}(x)$，　$j=1,2,\cdots,c;j\neq i$，

则　x 隶属于 $\underset{\sim}{A}_i$，即将 x 判属第 i 类。

例 6.9　细胞识别。

论域 $U=\{x\mid x=(NA,NL,A,L,NI,MI,ME)^T\}$。式中，$NA$ 为核(拍照)面积，NL 为核周长，A 为细胞面积，L 为细胞周长，NI 为核内总光密度，MI 为核内平均光密度，ME 为核内平均透过率。

根据病理医生的实际经验，选出下列六种主要因素，它们都是 U 上的模糊子集。

$\underset{\sim}{A}$：核增大

$$\mu_{\underset{\sim}{A}}(x)=\left(1+\alpha_1\left(\frac{NA_0}{NA}\right)^2\right)^{-1}$$

式中，NA_0 是正常核面积。

$\underset{\sim}{B}$：核增深

$$\mu_{\underset{\sim}{B}}(x)=\left(1+\frac{\alpha_2}{(NI)^2}\right)^{-1}$$

$\underset{\sim}{C}$：核浆比倒置

$$\mu_{\underset{\sim}{C}}(x)=\left(1+\frac{\alpha_3}{(NA)^2}\right)^{-1}$$

$\underset{\sim}{D}$：染色质不匀

$$\mu_{\underset{\sim}{D}}(x)=\left(1+\frac{\alpha_4(ME)^2}{(ME+\lg MI)^2}\right)^{-1}$$

$\underset{\sim}{E}$：核畸形

$$\mu_{\underset{\sim}{E}}(x)=\left(1+\frac{\alpha_5}{((NL)^2/NA-4\pi)^2}\right)^{-1}$$

$\underset{\sim}{F}$：细胞畸形

$$\mu_{\underset{\sim}{F}}(x)=\left(1+\frac{\alpha_6}{(L^2/A-L_0^2/A_0)^2}\right)^{-1}$$

其中，A_0，L_0 是正常值。上面的 $\alpha_1,\cdots,\alpha_6$ 是调整参数。

再由上述六个因素构造出表示四类细胞的模糊子集：

$\underset{\sim}{M}$：癌细胞

$$\mu_{\underset{\sim}{M}}(x)=(\mu_{\underset{\sim}{A}}(x)\wedge\mu_{\underset{\sim}{B}}(x)\wedge\mu_{\underset{\sim}{C}}(x)\wedge(\mu_{\underset{\sim}{D}}(x)\vee\mu_{\underset{\sim}{E}}(x))\vee\mu_{\underset{\sim}{F}}(x)$$

$\underset{\sim}{N}$：重度核异质细胞

$$\mu_{\underset{\sim}{N}}(x)=\mu_{\underset{\sim}{A}}(x)\wedge\mu_{\underset{\sim}{B}}(x)\wedge\mu_{\underset{\sim}{C}}(x)\wedge(1-\mu_{\underset{\sim}{M}}(x))$$

$\underset{\sim}{R}$：轻度核异质细胞

$$\mu_{\underset{\sim}{R}}(x)=\mu_{\underset{\sim}{A}}^{\frac{1}{2}}(x)\wedge\mu_{\underset{\sim}{B}}^{\frac{1}{2}}(x)\wedge\mu_{\underset{\sim}{C}}^{\frac{1}{2}}(x)\wedge(1-\mu_{\underset{\sim}{M}}(x))\wedge(1-\mu_{\underset{\sim}{N}}(x))$$

$\underset{\sim}{K}$：正常细胞

$$\mu_{\underset{\sim}{K}}(x)=(1-\mu_{\underset{\sim}{M}}(x))\wedge(1-\mu_{\underset{\sim}{N}}(x))\wedge(1-\mu_{\underset{\sim}{R}}(x))$$

上面构造了四个模糊子集的隶属函数。有了这些隶属函数后，若要识别某细胞属哪一类，只要将该细胞的特征值代入上述各隶属函数进行运算，按最大隶属原则取 $\mu_{\underset{\sim}{M}}(x)$，$\mu_{\underset{\sim}{N}}(x)$，$\mu_{\underset{\sim}{R}}(x)$，$\mu_{\underset{\sim}{K}}(x)$ 中最大的一类，即为识别结果。

从这个例子可以看出，这种识别方法本身是简单的。关键的问题是要设计出各类的隶属函数，但要确定恰当的隶属函数并不容易，这需要对模式的特征有足够的了解，还需要有一定的数学技巧。

6.3 模糊 BP 网

神经网络和模糊识别虽然都是仿效生物体信息处理系统以获得柔性信息处理能力，但是它们本身的含义是不同的。神经网络是从微观上模拟人脑，采取由底到顶的方法，通过对大量范例的学习，将获取的知识淀积在网络权系数上，形成自适应非线性动态系统，这种系统能够处理不能被语言化的模式信息。模糊识别系统仿效以语言和概念为代表的脑的宏观功能，采取由顶到底的方法，通过主观引入的隶属函数和模糊规则，从逻辑上处理含模糊性的模式信息。

如何将模糊理论与神经网络有机地结合起来，取长补短，构成具有强大学习能力和识别能力的模糊神经网络，是目前最受人注目的课题之一。

模糊神经网络应该是将模糊理论渗透到网络学习算法的各个环节，包括网络权系数修正规则等。本节讨论的模糊 BP 网在网络结构和学习算法上与 BP 网一致，只在输出表达形式上体现了模糊化。

在传统的 BP 网学习算法中，如果样本 x 属于第 k 类，那么样本的希望输出，除第 k 个输出节点的输出为 1 外，其它输出节点的输出为 0，输出只取 0，1 两值，非 1 即 0，一点也不含糊。模糊 BP 网对此作了修改，将样本的希望输出改为样本相对于各类的希望隶属度。在模糊 BP 网学习阶段，送入网络的是样本及其相对于各类的希望隶属度。通过训练，使网络具有反映训练集中输入与输出的隶属关系的能力。在识别时，输出就能给出待识模式的隶属度来。下面介绍一下样本的希望隶属度的计算。

假定，第 k 类训练样本的均值向量为 m_k，标准方差向量为 V_k，则训练样本 x 相对于第 k 类的加权距离可定义为

$$y_k = \sqrt{\sum_{i=1}^{n} \left(\frac{x_i - m_{ki}}{v_{ki}}\right)^2}, \quad k = 1,2,\cdots,c$$

其中，n 为样本 x 的维数，c 为类别数。加权值 $1/v_{ki}$ 用来考虑类的方差，即方差大的特征在分类时其加权较小。

样本 x 对第 k 类的希望隶属度定义为

$$\mu_k(x) = \frac{1}{1 + \left(\frac{y_k}{f_g}\right)^{f_c}}, \qquad k = 1,2,\cdots,c$$

其中，正常数 f_g 和 f_c 为两个参数，控制样本的隶属程度的对比度。显然，$\mu_k(x) \in [0,1]$。从上式可知，若样本 x 对第 k 类的距离越大，则其隶属度越小；若距离为 0，则其隶属度为 1。

最后重述一下，这里的模糊 BP 网与传统的 BP 网的学习算法基本相同。区别仅在于：学习阶段，对于 BP 网，送入网络的是样本及其希望输出，而对于模糊 BP 网，送入网络的是样本及其希望隶属度，即用希望隶属度代替希望输出。当然也可将输入模糊化，即输入网络的不是样本本身，而是样本相对于各类的隶属度。这里不作讨论。

6.4 基于模糊等价关系的聚类分析

基于模糊等价关系的聚类分析的步骤大致如下。

第一步　建立模糊相似矩阵。

设 $S = \{x^1, x^2, \cdots, x^N\}$ 是待聚类的样本的全体。每一个样本都有 n 个特征。建立模糊相似矩阵 $R = (r_{jk})_{N \times N}$ 的过程叫做标定，r_{jk} 表示样本 x^j 与 x^k 按 n 个特征相似的程度，称为相似系数。显然标定的关键在于如何合理地求出相似系数 r_{jk}. 求取 r_{jk} 的方法很多，下面列出三种方法，可根据实际情况选取一种方法。

(1)线段打分法

请有经验者在线段上做标记。设有 l 个人参加评分。第 m 个

人($1 \leqslant m \leqslant l$),就任何两个样本 x^j 和 x^k,在图 6-3 第一个线段上做标记 $y_{jk}^{(m)}$,在第二个线段上做标记 $a_{jk}^{(m)}$(表示他做标记 $y_{jk}^{(m)}$ 的把握程度)。

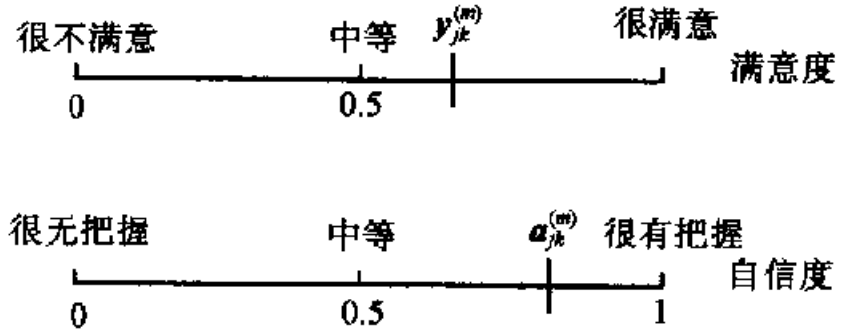


图 6-3

计算相似系数

$$r_{jk} = \frac{1}{l}\sum_{m=1}^{l} a_{jk}^{(m)} y_{jk}^{(m)}$$

(2)绝对值减数法

$$r_{jk} = 1 - \alpha\sum_{i=1}^{n} |x_i^j - x_i^k|$$

其中,α 为适当选取的常数,使 r_{jk} 在[0,1]中且分散开。

(3)夹角余弦法

$$r_{jk} = \frac{(x^j)^T x^k}{\| x^j \| \cdot \| x^k \|}$$

如果 r_{jk} 出现负值,可采用下面的方法将全体 r_{jk} 进行调整:

取

$$r'_{jk} = \frac{r_{jk} + 1}{2}$$

则

$$r'_{jk} \in [0,1]$$

第二步　聚类。

由第一步建立起来的模糊矩阵 R，一般情况下是模糊相似矩阵，即 R 只满足对称性和自反性，不满足传递性，还需要将 R 改造成模糊等价矩阵 R^*，由 R^* 作出分级聚类树，取适当 $\lambda\in[0,1]$，由截矩阵 R_λ^* 得出所需的分类。

例 6.10　设 $S=\{x^1,x^2,x^3,x^4,x^5\}$ 表示父、子、女、邻居、母五人构成的论域。请陌生人对这五人按相貌相像的程度进行聚类。

首先，对 S 中任意两个人 x^i 和 x^k，让打分者采用线段打分法，得到相似系数 r_{ik}，于是得模糊相似矩阵

$$R=\begin{pmatrix}1 & 0.8 & 0.6 & 0.1 & 0.2\\ 0.8 & 1 & 0.8 & 0.2 & 0.85\\ 0.6 & 0.8 & 1 & 0 & 0.9\\ 0.1 & 0.2 & 0 & 1 & 0.1\\ 0.2 & 0.85 & 0.9 & 0.1 & 1\end{pmatrix}$$

由于

$$R^2=\begin{pmatrix}1 & 0.8 & 0.8 & 0.2 & 0.8\\ 0.8 & 1 & 0.85 & 0.2 & 0.85\\ 0.8 & 0.85 & 1 & 0.2 & 0.9\\ 0.2 & 0.2 & 0.2 & 1 & 0.2\\ 0.8 & 0.85 & 0.9 & 0.2 & 1\end{pmatrix}\neq R$$

故 R 不是模糊等价矩阵。注意到

$$R^4=\begin{pmatrix}1 & 0.8 & 0.8 & 0.2 & 0.8\\ 0.8 & 1 & 0.85 & 0.2 & 0.85\\ 0.8 & 0.85 & 1 & 0.2 & 0.9\\ 0.2 & 0.2 & 0.2 & 1 & 0.2\\ 0.8 & 0.85 & 0.9 & 0.2 & 1\end{pmatrix}=R^2$$

于是得模糊等价矩阵 $R^*=R^2$.

当 $\lambda\in[0.9,1]$ 时，由 R_λ^* 得到的分类为

$$\{x^1\},\quad \{x^2\},\quad \{x^3\},\quad \{x^4\},\quad \{x^5\}$$

当 $\lambda\in[0.85,0.9]$时，S 分成四类

$$\{x^1\},\quad \{x^2\},\quad \{x^3,x^5\},\quad \{x^4\}$$

当 $\lambda\in[0.2,0.8]$时，S 分为两类

$$\{x^1,x^2,x^3,x^5\},\quad \{x^4\}$$

当 $\lambda\in[0,0.2]$时，S 自身为一类。

图 6－4 是它的分级聚类树。

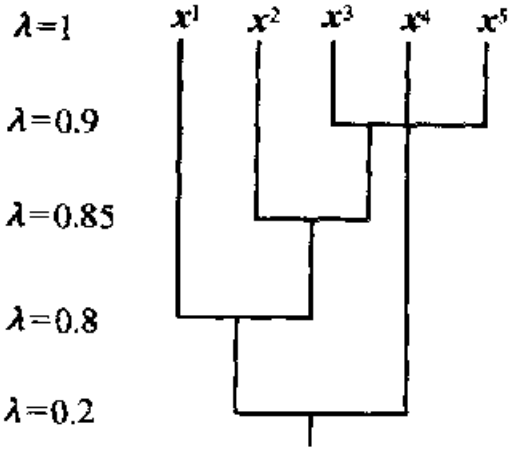


图 6－4

例 6.11　区域环境污染分量。

每个区域的污染情况由空气、水、土壤和噪声污染数据来描述。假设有五个区域 x^1,x^2,x^3,x^4,x^5，它们的污染数据如表 6－2 所示。

表 6－2

	空气	水	土壤	噪声
x^1	5	5	3	2
x^2	2	3	4	5
x^3	5	5	2	3
x^4	1	5	3	1
x^5	2	4	5	1

取论域 $S=\{x^1,x^2,x^3,x^4,x^5\}$，按绝对值减数法进行标定，取 $\alpha=0.1$，得模糊相似矩阵

$$R=\begin{pmatrix}1 & 0.1 & 0.8 & 0.5 & 0.3\\ 0.1 & 1 & 0.1 & 0.2 & 0.4\\ 0.8 & 0.1 & 1 & 0.3 & 0.1\\ 0.5 & 0.2 & 0.3 & 1 & 0.6\\ 0.3 & 0.4 & 0.1 & 0.6 & 1\end{pmatrix}$$

经计算可知

$$R^8=R^4=\begin{pmatrix}1 & 0.4 & 0.8 & 0.5 & 0.5\\ 0.4 & 1 & 0.4 & 0.4 & 0.4\\ 0.8 & 0.4 & 1 & 0.5 & 0.5\\ 0.5 & 0.4 & 0.5 & 1 & 0.6\\ 0.5 & 0.4 & 0.5 & 0.6 & 1\end{pmatrix}$$

此即例 6.7 中的模糊等价矩阵，其分级聚类树见图 6-2。若 $\lambda=0.6$，则 S 可分为三类

$$\{x^1,x^3\},\quad \{x^2\},\quad \{x^4,x^5\}。$$

6.5 模糊 c-均值算法

在聚类分析一章中，我们讨论过 c-均值算法。c-均值算法是应用最广泛的一种聚类算法，它采用的聚类准则函数为误差平方和函数：

$$J=\sum_{i=1}^{c}\sum_{j=1}^{N}d_{ji}\|x^j-w_i\|^2$$

其中，w_i 是第 i 类的聚类中心。d_{ji} 用来标明样本 x^j 是否属于第 i 类，如果 x^j 属于第 i 类，则 $d_{ji}=1$；如果 x^j 不属于第 i 类，则 $d_{ji}=0$. 这就是说，d_{ji} 要么为 1，要么为 0，一点也不含糊！实际问题中，并不是这么绝对，x^j 属于各类的隶属程度常常是 0 与 1 之间的一个数。因此，我们将 d_{ji} 改为 μ_{ji}，$\mu_{ji}\in[0,1]$。这样，聚类准则函数改为

$$J_m = \sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ji})^m \| x^j - w_i \|^2$$

其中 ①$\mu_{ji}\in[0,1]$, $i=1,2,\cdots c$; $j=1,2,\cdots,N$;

②$\sum_{i=1}^{c}\mu_{ji}=1$, $j=1,2,\cdots,N$

这表明每一个样本属于各类的隶属度之和为 1;

③$\sum_{j=1}^{N}\mu_{ji}>0$, $i=1,2,\cdots c$

这表明每一类模糊集不可能是空集合;

④为了加强 x^j 属于各类的隶属程度的对比度,在准则函数中添上参数 m,$1<m<\infty$,m 越大,对比度越大。

显然,模糊聚类结果有无穷多个,我们要求的是使 J_m 达到最小值的聚类结果。为此,我们将 J_m 分别对 w_i 和 μ_{ji}求导,并令它们的导数为 0,再代入条件

$$\sum_{i=1}^{c}\mu_{ji} = 1$$

即可得

$$\mu_{ji}=\frac{\left(\frac{1}{\| x^j-w_i \|^2}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c}\left(\frac{1}{\| x^j-w_i \|^2}\right)^{\frac{1}{m-1}}}, \quad i=1,2,\cdots,c;\ j=1,2,\cdots,N \tag{6.1}$$

$$w_i=\frac{\sum_{j=1}^{N}(\mu_{ji})^m x^j}{\sum_{j=1}^{N}(\mu_{ji})^m} \tag{6.2}$$

下面给出模糊 c-均值算法步骤:

①给定类别数 c,参数 m,容许误差 Emax 的值,令 $k=1$;

②初始化聚类中心:$w_i(1)$, $i=1,2,\cdots,c$;

③按式(6.1)计算隶属度 $\mu_{ji}(k)$,$i=1,2,\cdots,c$;$j=1,2,\cdots,N$;

④按式(6.2)修正所有的聚类中心 $w_i(k+1)$,$i=1,2,\cdots,c$;

⑤计算误差

$$e = \sum_{i=1}^{c} \| w_i(k+1) - w_i(k) \|^2$$

如果 $e<$Emax，则算法结束；否则，$k \leftarrow k+1$，转到第③步。

算法结束后，可按下列两种方法将所有的样本进行归类。

方法一：

若 $\mu_{ji} > \mu_{jl}$，$l=1,2,\cdots,c$；$l \neq i$

则将 x^j 归入第 i 类。

方法二：

若 $\| x^j - w_i \|^2 < \| x^j - w_l \|^2$，$l=1,2,\cdots,c$；$l \neq i$

则将 x^j 归入第 i 类。

6.6 模糊 Kohonen 聚类网络

将模糊理论与神经网络相结合有利于提高系统的柔性处理能力，模糊 BP 网就是一种尝试。不过模糊 BP 网只在输出表达形式上体现了模糊化。本节介绍的模糊 Kohonen 聚类网络在权系数修正规则中也体现了模糊化，将样本的隶属度引入到权系数的修正规则中。

模糊 Kohonen 聚类网络的结构与 Kohonen 聚类网络的结构相同。

下面给出模糊 Kohonen 聚类网络算法步骤。

①给定类别数 c，参数 $m(1)>1$，容许误差 Emax 和允许迭代次数 K，令 $k=1$。

初始化权向量 $w_i(1)$，$i=1,2,\cdots,c$；

②计算隶属度

$$\mu_{ji}(k) = \frac{\left(\dfrac{1}{\| x^j - w_i(k) \|^2} \right)^{\frac{1}{m(k)-1}}}{\sum_{i=1}^{c} \left(\dfrac{1}{\| x^j - w_i(k) \|^2} \right)^{\frac{1}{m(k)-1}}}$$

③计算步长

$$\alpha_{ji}(k) = (\mu_{ji}(k))^{m(k)}$$

④修正权向量

$$w_i(k+1) = w_i(k) + [\sum_{j=1}^{N}\alpha_{ji}(k)(x^j - w_i(k))]/\sum_{j=1}^{N}\alpha_{ji}(k)$$

⑤计算误差

$$e = \sum_{i=1}^{c} \| w_i(k+1) - w_i(k) \|^2$$

如果 $e<\text{Emax}$，则算法结束；

否则

$$\Delta m = (m(1)-1)/k$$
$$m(k+1) = m(k) - \Delta m$$
$$k \leftarrow k+1$$

转到第②步。

模糊 Kohonen 聚类网络所采用的算法实际上就是模糊 c -均值算法，可证明如下。

模糊 Kohonen 网的权系数修正量为

$$[\sum_{j=1}^{N}\alpha_{ji}(k)(x^j - w_i(k))]/\sum_{j=1}^{N}\alpha_{ji}(k)$$
$$= \sum_{j=1}^{N}\alpha_{ji}(k)x^j/\sum_{j=1}^{N}\alpha_{ji}(k) - w_i(k)\sum_{j=1}^{N}\alpha_{ji}(k)/\sum_{j=1}^{N}\alpha_{ji}(k)$$
$$= \sum_{j=1}^{N}\alpha_{ji}(k)x^j/\sum_{j=1}^{N}\alpha_{ji}(k) - w_i(k)$$

因此，模糊 Kohonen 网的权系数修正规则为

$$w_i(k+1) = w_i(k) - w_i(k) + \sum_{j=1}^{N}\alpha_{ji}(k)x^j/\sum_{j=1}^{N}\alpha_{ji}(k)$$
$$= \frac{\sum_{j=1}^{N}(\mu_{ji}(k))^{m(k)}x^j}{\sum_{j=1}^{N}(\mu_{ji}(k))^{m(k)}}$$

由上式可见，模糊 Kohonen 网的权系数修正规则与模糊 c -均值法相同。

第七章　特征选择与提取

特征抽取的目的是获取一组“少而精”的分类特征，即获取特征数目少且分类错误概率小的特征向量。

特征抽取常常分几步进行。

第一步：特征形成　根据被识别的对象产生一组原始特征。它们可以是传感器的直接测量值，也可以是将传感器的测量值作某些计算后得到的值。以细胞识别为例，通过图象输入得到细胞的数字图象。根据细胞的数字图像可以计算出细胞总面积，总光密度，胞核面积，核浆比等等，这些值我们称为原始特征，产生这些原始特征的过程称为特征形成。

第二步：特征选择　由特征形成过程得到的原始特征可能很多，如果把所有的原始特征都作为分类特征送往分类器，不仅使得分类器复杂，分类计算判别量大，而且分类错误概率也不一定小。因此需要减少特征数目。减少特征数目的方法有两种，一种是特征选择，另一种是特征提取。

从一组特征中挑选出一些最有效的特征的过程叫特征选择。

第三步：特征提取　特征提取是另一种减少特征数目的方法。通过映射（或变换）的方法把高维的特征向量变换为低维的特征向量。

特征形成得到原始特征后，可以只作特征选择，也可以只作特征提取，当然也可以先进行特征选择再作特征提取，可视具体情况而定。

0 本章只讨论特征选择和特征提取的方法。

7.1 类别可分性准则

特征选择或特征提取的任务是从 n 个特征中求出对分类最有效的 m 个特征($m<n$)。对于特征选择来讲，从 n 个特征中选择出 m 个特征，有 C_n^m 种组合方式，哪一种特征组的分类效果最好，这需要有一个比较标准，即需要一个定量的准则来衡量选择结果的好坏。同样，对于特征提取来讲，把 n 维特征向量变换成 m 维特征向量，有各种变换，哪一种变换得到的 m 维特征向量对分类最有效，也需要用一个准则来衡量。

很自然地会想到，既然我们的目的是设计分类器，那么用分类器的错误概率作为准则就行了，也就是说，使分类器错误概率最小的那组特征，就应当是一组最有效的特征。从理论上讲，这是完全正确的，但在实际使用中却有很大困难。

从第二章中错误概率的计算公式就会发现，即使在类条件概率密度已知的情况下错误概率的计算就很复杂，何况实际问题中概率分布常常不知道，这使得直接用错误概率作为准则来评价特征的有效性比较困难。

我们希望找出另外一些更实用的准则来衡量各类间的可分性，并希望可分性准则满足下列几条要求：

①与错误概率有单调关系，这样使准则取最大值的效果一般来说其错误概率也较小。

②度量特性：

$$J_{ij}>0, \quad \text{当 } i\neq j \text{ 时}$$
$$J_{ij}=0, \quad \text{当 } i=j \text{ 时}$$
$$J_{ij}=J_{ji}$$

这里 J_{ij} 是第 i 类和第 j 类的可分性准则函数，J_{ij} 越大，两类的分离程度就越大。

③单调性，即加入新的特征时，准则函数值不减小。

$$J_{ij}(x_1,x_2,\cdots,x_d)\leqslant J_{ij}(x_1,x_2,\cdots,x_d,x_{d+1})$$

下面介绍二种常用的类别可分性准则。

7.1.1 基于距离的可分性准则

各类样本之间的距离越大，则类别可分性越大。因此，我们可以用各类样本之间的距离的平均值作为可分性准则

$$J_d=\frac{1}{2}\sum_{i=1}^{c}P_i\sum_{j=1}^{c}P_j\frac{1}{N_iN_j}\sum_{x^i\in\omega_i}\sum_{x^j\in\omega_j}D(\boldsymbol{x}^i,\boldsymbol{x}^j)\qquad(7.1)$$

式中，c 为类别数；N_i 为 ω_i 类中样本数；N_j 为 ω_j 类中样本数；P_i，P_j 是相应类别的先验概率；$D(\boldsymbol{x}^i,\boldsymbol{x}^j)$是样本 $\boldsymbol{x}^i$ 与 $\boldsymbol{x}^j$ 之间的距离。

如果采用欧氏距离，即有

$$D(\boldsymbol{x}^i,\boldsymbol{x}^j)=(\boldsymbol{x}^i-\boldsymbol{x}^j)^T(\boldsymbol{x}^i-\boldsymbol{x}^j)\qquad(7.2)$$

用 $\boldsymbol{m}_i$ 表示第 i 类样本集的均值向量

$$\boldsymbol{m}_i=\frac{1}{N_i}\sum_{x^i\in\omega_i}\boldsymbol{x}^i\qquad(7.3)$$

用 $\boldsymbol{m}$ 表示所有各类的样本集总平均向量

$$\boldsymbol{m}=\sum_{i=1}^{c}P_i\boldsymbol{m}_i\qquad(7.4)$$

将式(7.2)，(7.3)，(7.4)代入式(7.1)得

$$J_d=\sum_{i=1}^{c}P_i\Big[\frac{1}{N_i}\sum_{x^i\in\omega_i}(\boldsymbol{x}^i-\boldsymbol{m}_i)^T(\boldsymbol{x}^i-\boldsymbol{m}_i)+(\boldsymbol{m}_i-\boldsymbol{m})^T(\boldsymbol{m}_i-\boldsymbol{m})\Big]\qquad(7.5)$$

我们也可以用下面定义的矩阵写出 J_d 的表达式。

令

$$S_b=\sum_{i=1}^{c}P_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^T\qquad(7.6)$$

以及

$$S_w=\sum_{i=1}^{c}P_i\frac{1}{N_i}\sum_{x^i\in\omega_i}(\boldsymbol{x}^i-\boldsymbol{m}_i)(\boldsymbol{x}^i-\boldsymbol{m}_i)^T$$

$$= \sum_{i=1}^{c} P_i C_i \tag{7.7}$$

则

$$J_d = t_r(S_w + S_b)$$

其中，$t_r(S_w+S_b)$表示取矩阵S_w+S_b的迹。S_w为类内离散度矩阵，S_b为类间离散度矩阵。

我们希望类内离散度尽量小，类间离散度尽量大，因此除$J_d(\boldsymbol{x})$外，还可以提出下列准则函数

$$J_2 = t_r(S_w^{-1}S_b)$$

$$J_3 = |S_w^{-1}S_b|$$

7.1.2 基于熵函数的可分性准则

最佳分类器由后验概率确定，所以可由特征的后验概率分布来衡量它对分类的有效性。如果对某些特征，各类后验概率是相等的，即

$$P(\omega_i/\boldsymbol{x}) = \frac{1}{c}$$

其中c为类别数，则我们将无从确定样本所属类别，或者我们只能任意指定$\boldsymbol{x}$属于某一类(假定先验概率相等或不知道)，此时其错误概率为

$$P_e = 1 - \frac{1}{c} = \frac{c-1}{c}$$

另一个极端情况是，如果能有一组特征使得

$$P(\omega_i/\boldsymbol{x}) = 1, \text{且 } P(\omega_j/\boldsymbol{x}) = 0, \forall\ j \neq i$$

此时$\boldsymbol{x}$划归ω_i类，其错误概率为0.

可见后验概率越集中，错误概率就越小。后验概率分布越平缓(接近均匀分布)，则分类错误概率就越大。

为了衡量后验概率分布的集中程度，需要规定一个定量准则，我们可以借助于信息论中关于熵的概念。

设ω为可能取值$\omega_i, i=1,2,\cdots,c$的一个随机变量，它的取值

依赖于分布密度为 $p(x)$的随机向量 x(特征向量),即给定 x 后 ω_i 的概率是 $P(\omega_i/x)$. 现在进行观察变量 x 以及相应的 ω 值的实验。我们想知道的是:给定某一 x 后,我们从观察 ω 的结果中得到了多少信息?或者说 ω 的不确定性减少了多少?显然,假如对某一 x,有 $P(w_i/x)=1$,且 $P(\omega_j/x)=0, \forall\ j\neq i$,则观察结果必然是 $\omega=\omega_i$. 因此观察的结果并未使我们得到任何信息。反之,若对所有的 i, $P(\omega_i/x)$都相等,我们只能任意猜测 ω 的可能结果,而从观察到的实际发生的 ω_i 事件中得到的信息量就不再等于零,而相应于观察前的不确定程度。

从特征抽取的角度看,显然用具有最小不确定性的那些特征进行分类是有利的。在信息论中用"熵"作为不确定性的质量,它是 $P(\omega_1/x), P(\omega_2/x), \cdots, P(\omega_c/x)$的函数。可定义如下形式的广义熵:

$$J_C^{\alpha}[P(\omega_1/x), P(\omega_2/x), \cdots, P(\omega_c/x)]$$
$$= (2^{1-\alpha}-1)^{-1}\left[\sum_{i=1}^{c} P^{\alpha}(\omega_i/x) - 1\right]$$

式中,α 是一个实的正参数,$\alpha\neq 1$。

不同的 α 值可以得到不同的熵分离度量,例如当 α 趋近于 1 时,据 L'Hospital 法则有

$$J_C^{1}[P(\omega_1/x), P(\omega_2/x), \cdots, P(\omega_c/x)]$$
$$=\lim_{\alpha\to 1}(2^{1-\alpha}-1)^{-1}\left[\sum_{i=1}^{c} P^{\alpha}(w_i/x)-1\right]$$
$$=-\sum_{i=1}^{c} P(\omega_i/x)\log_2 P(\omega_i/x)$$

这就是 Shannon 熵。

当 $\alpha=2$ 时,得到平方熵

$$J_C^{2}[P(\omega_1/x), P(\omega_2/x), \cdots, P(\omega_c/x)] = 2\left[1 - \sum_{i=1}^{c} P^2(\omega_i/x)\right]$$

显然,为了对所提取的特征进行评价,我们要计算空间每一点

的熵函数。在熵函数取值较大的那一部分空间，不同类的样本必然在较大的程度上互相重叠。因此熵函数的期望值

$$J(\cdot) = \mathrm{E}\{J_C^a[P(\omega_1/\boldsymbol{x}), P(\omega_2/\boldsymbol{x}), \cdots, P(\omega_c/\boldsymbol{x})]\}$$

可以表征类别的分离程度，它可用来作为所提取特征的分类性能的准则函数。

7.2 特征选择

从 n 个特征中挑选出 $m(m<n)$个最有效的特征，这就是特征选择的任务。最直接的特征选择方法是根据专家的知识挑选那些对分类最有影响的特征。另一种是用数学方法进行筛选比较，找出最有分类信息的特征。本节只讨论用数学方法进行特征选择。

要完成特征选择的任务，必须解决两个问题，一个是选择的标准，这可以用前面讲的类别可分性准则，选出使某一可分性达到最大的特征组来。另一个问题是找一个较好的算法，以便在较短的时间内找出最优的那一组特征。

有两个极端的特征选择算法，一个是单独选择法，另一个是穷举选择法。

单独选择法就是把 n 个特征每个特征单独使用时的可分性准则函数值都算出来，然后按准则函数值按从大到小排序，如

$$J(\boldsymbol{x}_1) > J(\boldsymbol{x}_2) > \cdots > J(\boldsymbol{x}_m) > \cdots J(\boldsymbol{x}_n)$$

然后，取使 J 较大的前 m 个特征作为选择结果。这样得到的 m 个特征是否就是一个最优的特征组呢？如果是，特征选择就简单了。然而，即使当所有特征都相互独立时，除一些特殊情况外，一般来说，前 m 个最有效的特征并非最优的特征组。

从 n 个特征中挑选 m 个，所有可能的组合数为 C_n^m，比如 $n=20, m=10$，从 20 个特征中选出 10 个特征，有 $C_{20}^{10}=184756$ 种特征组合方式。我们把 184756 种特征组合的可分性准则函数值都算出来，然后看哪一种特征组合的准则函数值最大，我们就选中该种组

合的 10 个特征。这就是穷举选择法。显然，穷举法的计算量太大，以至无法实现。因此，我们常采用一些优化算法进行特征选择。

分支定界算法是一种自上而下的搜索方法，但具有回溯功能，可使所有可能的特征组合都被考虑到。由于合理地组织搜索过程，使得有可能不必计算某些组合，而又不影响得出最优结果。使用分支定界算法必须具备一个先决条件：所采用的可分性准则函数必须具有单调性。即，从 n 个特征中选出 m 个特征组成 m 维特征向量，再从这 m 个特征中选出 k 个特征组成 k 维特征向量，其准则函数应满足 $J_n \geqslant J_m \geqslant J_k$ 的条件。分支定界算法正是利用了这种单调性，才可能减少了对一些特征组合的搜索。

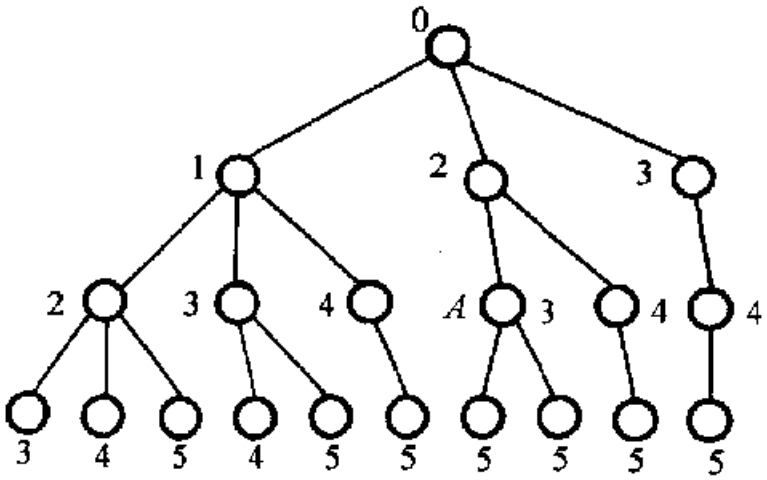


图 7－1

整个搜索过程可用树表示出来，图 7－1 为从五个特征中选择三个的例子，节点上的标号是去掉的特征序号。每一级在上一级基础上再去掉一个特征。例如节点 A 表示去掉第 2、3 号二个特征后的特征组，即 (x_1, x_4, x_5)。级数正好是已去掉的特征数。六个特征中选二个，四级即可。

若某一支已搜索到叶节点，比如已搜索到图中右侧第一支的

叶节点，该节点表示的特征组为 $x^* = (x_1, x_2)^T$，计算出相应的准则函数值作为界：$B = J(x^*)$。此时树中某个节点，比如节点 A，节点 A 表示的特征组为 (x_1, x_4, x_5)，计算出相应的准则函数值 J_A，若 $J_A \leqslant B$，则节点 A 以下各点都不必去计算，因为根据单调性，它们的 J 值都不会大于 B。

如果在搜索过程中发现一个叶节点所对应的准则函数值 $J > B$，则可断言，它是当前搜索到的最优特征组，于是修改 x^* 和 B，然后继续搜索，直到全树搜索完毕为止。

7.3 基于距离可分性准则的特征提取

特征选择是在一定准则下从 n 个特征中挑选出最优的 m 个特征，其余的 $n-m$ 个特征被取消了。一般来说，原来的 $n-m$ 个特征多多少少都含有一些分类信息，简单地把这些特征抛弃了，有点可惜。那么能不能把 n 个特征的分类信息尽量集中到 m 个特征中去？这就是特征提取要研究的问题。

特征提取就是通过某种数学变换，把 n 个特征压缩为 m 个特征，即

$$y = A^T x$$

式中，x 是具有 n 个特征的向量，变换矩阵 A 是一个 $n \times m$ 阶的矩阵，经变换后的向量 y 是 m 维的，$m < n$.

特征提取的关键问题是求出最佳的变换矩阵，使得变换后的 m 维模式空间中，类别可分性准则值最大。

前面介绍的可分性准则都可用于特征提取。本节介绍基于距离的可分性准则的特征提取。

我们以可分性准则 $J_2 = t_r(S_w^{-1} S_b)$ 为例，详细讨论一下基于 J_2 的特征提取问题。

采用变换 $y = A^T x$ 后，我们希望在 m 维的 y 空间里，样本的类别可分性好，即希望在 y 空间里，准则函数 J_2 达到最大值。

y 空间里的协方差矩阵 C_y 与 x 空间里的协方差矩阵 C_x 有如下关系：

$$C_y = A^T C_x A$$

这是因为

$$\begin{aligned} C_y &= \mathrm{E}[(y - m_y)(y - m_y)^T] \\ &= \mathrm{E}[(A^T x - A^T m_x)(A^T x - A^T m_x)^T] \\ &= \mathrm{E}[A^T(x - m_x)(x - m_x)^T A] \\ &= A^T C_x A \end{aligned}$$

这样，y 空间里的类内离散度矩阵 S_{yw} 可用 x 空间里的类内离散度矩阵 S_{xw} 计算得到：

$$S_{yw} = A^T S_{xw} A$$

同样有

$$S_{yb} = A^T S_{xb} A$$

因此，特征提取问题就变成求变换矩阵 A，使得 y 空间里的准则函数 J_2 达到最大值。

$$J_2 = t_r(S_{yw}^{-1} S_{yb}) = tr[(A^T S_{xw}^{-1} A)^{-1}(A^T S_{xb} A)]$$

将上式对 A 求导：

$$\frac{\partial J_2}{\partial A} = -2 S_{xw} A S_{yw}^{-1} S_{yb} S_{yw}^{-1} + 2 S_{xb} A S_{yw}^{-1}$$

令上式等于零，得

$$(S_{xw}^{-1} S_{xb}) A = A(S_{yw}^{-1} S_{yb}) \tag{7.8}$$

利用线性变换 $z = B^T y = (AB)^T x$，可将对称矩阵 S_{yb}、S_{yw} 同时对角化：

$$B^T S_{yb} B = \Lambda, \quad B^T S_{yw} B = I \tag{7.9}$$

式中，B 是非奇异的 $m \times m$ 阶方阵。

从 y 到 z 的这种非奇异变换是不会改变准则函数的值：

$$\begin{aligned} t_r(S_{zw}^{-1} S_{zb}) &= t_r[(B^T S_{yw} B)^{-1}(B^T S_{yb} B)] \\ &= t_r[B^{-1} S_{yw}^{-1} (B^T)^{-1} B^T S_{yb} B] \end{aligned}$$

$$= t_r[S_{yw}^{-1}S_{yb}BB^{-1}]$$
$$= t_r(S_{yw}^{-1}S_{yb})$$

利用式(7.9)，式(7.8)可改写为

$$(S_{xw}^{-1}S_{xb})A = A(B\Lambda B^{-1})$$

即
$$(S_{xw}^{-1}S_{xb})(AB)=(AB)\Lambda$$

或
$$(S_{xw}^{-1}S_{xb})\varphi_i=\varphi_i\lambda_i, \quad i,=1,2,\cdots,n$$

上式说明 λ_i 和 φ_i 是 $S_{xw}^{-1}S_{xb}$的本征值和本征向量。

设 $S_{xw}^{-1}S_{xb}$的本征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$，按大小顺序排列为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_m \geqslant \cdots \geqslant \lambda_n$$

选前 m 个本征值对应的本征向量构成变换矩阵

$$A^T = (\varphi_1,\varphi_2\cdots,\varphi_m)^T$$

此时准则函数 J_2 可取得最大值：

$$J_2 = t_r(S_{yw}^{-1}S_{yb}) = t_r[(B^TS_{xw}B)^{-1}(B^TS_{xb}B)]$$
$$= t_r[I^{-1}\Lambda]$$
$$= \sum_{i=1}^{m}\lambda_i$$

例 7.1　给定先验概率相等的两类，其均值向量分别为 $\boldsymbol{m}_1=(1,3,-1)^T$，$\boldsymbol{m}_2=(-1,1,1)^T$，协方差矩阵分别为

$$C_1=\begin{pmatrix}4&1&0\\1&4&0\\0&0&1\end{pmatrix} \qquad C_2=\begin{pmatrix}2&1&0\\1&2&0\\0&0&1\end{pmatrix}$$

求用 J_2 的特征提取。

解　两类问题的类间离散度矩阵为

$$S_b = \sum_{i=1}^{2}P_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^T$$
$$= P_1P_2(\boldsymbol{m}_2-\boldsymbol{m}_1)(\boldsymbol{m}_2-\boldsymbol{m}_1)^T$$

由于 S_b 是由 c 类均值向量组成的矩阵，它是 c 个秩等于或小于 1 的矩阵之和，其秩等于或小于 $c-1$，故 $S_w^{-1}S_b$ 的秩等于或小于 $c-1$，对于两类情况，$S_w^{-1}S_b$ 只有一个非零本征值 λ_1，为了求

$S_w^{-1}S_b$ 的本征向量应解方程：

$$S_w^{-1}S_b\varphi_1 = \lambda_1\varphi_1$$

即

$$S_w^{-1}P_1P_2(m_2 - m_1)(m_2 - m_1)^T\varphi_1 = \lambda_1\varphi_1$$

$$\varphi_1 = \frac{P_1P_2}{\lambda_1}S_w^{-1}(m_2 - m_1)(m_2 - m_1)^T\varphi_1$$

$$= \alpha S_w^{-1}(m_2 - m_1)$$

系数 α 可略去，略去不影响分类效果。

$$S_w = \frac{1}{2}(C_1 + C_2) = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

$$S_w^{-1} = \frac{1}{8}\begin{bmatrix} 3 & -1 & 0 \\ -3 & 3 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

$$\varphi_1 = S_w^{-1}(m_2 - m_1) = \frac{1}{8}(1,5,8)^T$$

故变换矩阵为 φ_1^T，这就是要求的解。

7.4 基于 K-L 变换的特征提取

7.4.1 离散 K-L 展开式

K-L 变换是一种常用的正交变换。

假设 x 为 n 维的随机向量，x 可以用 n 个基向量的加权和来表示：

$$x = \sum_{i=1}^{n}\alpha_i\varphi_i \tag{7.10}$$

式中，φ_i 为基向量；α_i 为加权系数。

式(7.10)还可以用矩阵形式表示：

$$x = (\varphi_1, \varphi_2 \cdots, \varphi_n) \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{Bmatrix} = \Phi\alpha \tag{7.11}$$

式中，$\Phi = (\varphi_1, \varphi_2, \cdots \varphi_n)$，$\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)^T$

我们取基向量为正交向量，即

$$\varphi_i^T \varphi_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Φ 由正交向量构成，所以 Φ 是正交矩阵，即

$$\Phi^T \Phi = I \tag{7.12}$$

将式(7.11)两边左乘 Φ^T，并考虑到 Φ 为正交矩阵，得

$$\alpha = \Phi^T x \tag{7.13}$$

即

$$a_j = \varphi_j^T x, \quad j = 1, 2, \cdots, n \tag{7.14}$$

我们希望向量 α 的各个分量间互不相关。那么如何保证 α 的各个分量间互不相关呢？这取决于选用什么样的正交向量集 $\{\varphi_j\}$ 了！

设随机向量 x 的总体自相关矩阵为

$$R = \mathrm{E}[xx^T]$$

将 $x = \Phi\alpha$ 代入上式，得

$$R = \mathrm{E}(\Phi\alpha\alpha^T\Phi^T) = \Phi\mathrm{E}[\alpha\alpha^T]\Phi^T$$

我们要求向量 α 的各个分量间互不相关，即应满足下列关系：

$$\mathrm{E}[a_j a_k] = \begin{cases} \lambda_j, & j = k \\ 0, & j \neq k \end{cases}$$

写成矩阵形式，应使

$$E[\alpha\alpha^T]=\begin{bmatrix}\lambda_1 & & & 0\\ & \ddots & & \\ & & \lambda_j & \\ & & & \ddots \\ 0 & & & \lambda_n\end{bmatrix}=\Lambda$$

则
$$R=\Phi\Lambda\Phi^T$$

将上式两边右乘 Φ，得

$$R\Phi=\Phi\Lambda\Phi^T\Phi$$

因 Φ 是正交矩阵，所以得

$$R\Phi=\Phi\Lambda$$

即
$$R\varphi_j=\lambda_j\varphi_j,\quad j=1,2,\cdots,n$$

可以看出，λ_j 是 x 的自相关矩阵 R 的本征值，φ_j 是对应的本征向量。因为 R 是实对称矩阵，其不同本征值对应的本征向量应正交。

综上所述，K－L 展开式的系数可用下列步骤求出：

①求随机向量 x 的自相关矩阵

$$R=E[xx^T]$$

②求出自相关矩阵 R 的本征值 λ_j 和对应的本征向量 φ_j，$j=1,2,\cdots,n$. 得到矩阵

$$\Phi=(\varphi_1,\varphi_2,\cdots,\varphi_n)$$

③展开式系数即为

$$\alpha=\Phi^Tx$$

7.4.2 基于 K－L 变换的数据压缩

我们从 n 个本征向量中取出 m 个组成变换矩阵 A，即

$$A=(\varphi_1,\varphi_2,\cdots,\varphi_m),\quad m<n$$

这时 A 是一个 $n\times m$ 维矩阵，x 为 n 维向量，经过 A^Tx 变换，得到降维为 m 的新向量。现在的问题是选取哪 m 个本征向量构成变换

矩阵 A，使降维的新向量在最小均方误差准则下接近原来的向量 $\boldsymbol{x}$？

对于式(7.10)，即

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

现在只取 m 项，对略去的项用预先选定的常数 b_j 来代替，这时对 $\boldsymbol{x}$ 的估计值为

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

由此产生的误差为

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

均方误差为

$$\bar{\varepsilon}^2 = \mathrm{E}[\| \Delta \boldsymbol{x} \|^2] = \sum_{i=m+1}^{n} \mathrm{E}[(a_i - b_i)^2] \tag{7.15}$$

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

$$\begin{aligned} \frac{\partial \bar{\varepsilon}^2}{\partial b_j} &= \frac{\partial}{\partial b_j} \sum_{i=m+1}^{n} \mathrm{E}[(a_i - b_i)^2] \\ &= \frac{\partial}{\partial b_j} \mathrm{E}[(a_j - b_j)^2] \\ &= \mathrm{E}[-2(a_j - b_j)] = 0 \end{aligned}$$

所以

$$b_j = \mathrm{E}[a_j] \tag{7.16}$$

这就是说，对于省略掉的那些 α 中的分量，应该用它们的期望值来代替。

如果在 K-L 变换前，将模式总体的均值向量作为新坐标系的原点，即在新坐标系中，$\mathrm{E}[\boldsymbol{x}]=0$，根据式(7.16)得

$$b_j = \mathrm{E}[a_j] = \mathrm{E}[\boldsymbol{\varphi}_j^{\mathrm{T}} \boldsymbol{x}] = \boldsymbol{\varphi}_j^{\mathrm{T}} \mathrm{E}[\boldsymbol{x}] = 0$$

这样由式(7.15)给出的均方误差变为

$$\bar{\varepsilon}^2 = \sum_{j=m+1}^{n} \mathrm{E}[a_j^2] = \sum_{j=m+1}^{n} \mathrm{E}[(\boldsymbol{\varphi}_j^{\mathrm{T}} \boldsymbol{x})(\boldsymbol{\varphi}_j^{\mathrm{T}} \boldsymbol{x})^{\mathrm{T}}]$$

$$= \sum_{j=m+1}^{n} \mathrm{E}[\boldsymbol{\varphi}_j^T \boldsymbol{x}\boldsymbol{x}^T \boldsymbol{\varphi}_j]$$

$$= \sum_{j=m+1}^{n} \boldsymbol{\varphi}_j^T R \boldsymbol{\varphi}_j$$

$$= \sum_{j=m-1}^{n} \lambda_j$$

式中，λ_j 是 $\boldsymbol{x}$ 的自相关矩阵 R 的第 j 个本征值；$\boldsymbol{\varphi}_j$ 是与 λ_j 对应的本征向量。显然，所选的 λ_j 值越小，均方误差也越小。

综上所述，基于 K－L 变换的数据压缩的步骤如下：

① 平移坐标系，将模式总体的均值向量作为新坐标系的原点；

② 求出自相关距阵 R；

③ 求出 R 的本征值 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 及其对应的本征向量 $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_n$；

④ 将本征值按从大到小排序，如：

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_m \geqslant \cdots \geqslant \lambda_n$$

取前 m 个大的本征值所对应的本征向量构成变换矩阵：

$$A = (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_m)$$

⑤ 将 n 维的原向量变换成 m 维的新向量：

$$\boldsymbol{y} = A^T \boldsymbol{x}$$

例 7.2　给出样本数据如下：

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

$$\begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \end{pmatrix}, \begin{pmatrix} 4 \\ 5 \end{pmatrix}$$

试用 K－L 变换作一维的数据压缩。

解　①求样本总体均值向量

$$\boldsymbol{m} = \frac{1}{10}\left[\begin{pmatrix} -5 \\ -5 \end{pmatrix} + \begin{pmatrix} -5 \\ -4 \end{pmatrix} + \cdots + \begin{pmatrix} 4 \\ 5 \end{pmatrix}\right] = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

无需作坐标系平移。

② 求自相关矩阵

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

③ 求本征值和本征向量。解本征值方程

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

即

$$(25.4-\lambda)^2-25.0^2=0$$

解得本征值

$$\lambda_1=50.4,\quad \lambda_2=0.4$$

由 $R\varphi_j=\lambda_j\varphi_j$，可解得本征向量为

$$\varphi_1=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix},\quad \varphi_2=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

④ 取 φ_1 作为变换矩阵，即

$$A=\varphi_1$$

⑤ 将原样本变换为一维的样本：

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

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

其结果如图 7-2 所示。从图中可见，用 φ_1 作为变换矩阵是将原样本投影在 φ_1 向量轴上。

7.4.3 基于 K-L 变换的特征提取

K-L 变换适用任何概率分布，它是在均方误差最小的意义下获得数据压缩的最佳变换。如果采用大本征值对应的本征向量构成变换矩阵，则能对应地保留原样本中方差最大的数据分量，所以 K-L 变换起了减小相关性、突出差异性的效果，有人称之为主分量变换。不过，采用 K-L 变换作为模式分类的特征提取时要注

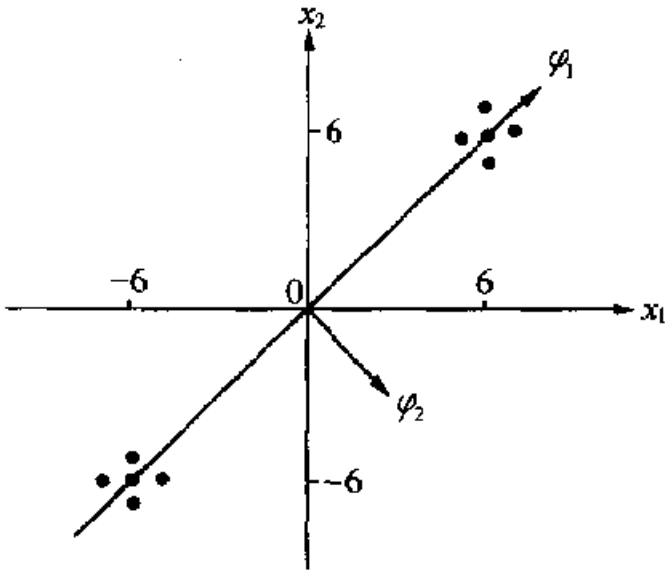


图 7－2

意保留不同类别的模式分类识别信息。单纯考虑尽可能准确地代表原模式的主分量，有时分类效果并不好。

(1) 采用模式总体自相关矩阵作 K－L 变换

这是把多类模式合并起来看成是一个总体分布，按其自相关矩阵作 K－L 变换，采用与大本征值对应的本征向量构成变换矩阵，使降维模式能在均方误差最小的条件下逼近原来的模式，这就是上面 7.4.2 小节中讨论过的情况。采用自相关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，用总体自相关矩阵的 K－L 变换便能尽量多地保留可分性信息。

(2) 采用离散度矩阵 $S_w^{-1}S_b$ 作 K－L 变换

为了强调模式的分类识别信息，可用 $S_w^{-1}S_b$ 作 K－L 变换。

如果模式类别数为 c，则类间离散度矩阵 S_b 的秩不会大于 $c-1$。假使类内离散度矩阵 S_w 为满秩矩阵，则 $S_w^{-1}S_b$ 的秩也不会大于 $c-1$，$S_w^{-1}S_b$ 只多有 $c-1$ 个非零本征值。我们可求出 $S_w^{-1}S_b$ 的 $c-1$ 个非零本征值，按从大到小排序，如：

$$\lambda_1 > \lambda_2 > \cdots > \lambda_{r-1}$$

选出 m 个与大本征值对应的本征向量构成变换矩阵。这实际上就是 7.3 节讨论的基于距离可分性准则的特征提取。

7.5 基于神经网络的特征提取

从基于 K－L 变换的数据压缩方法中可以看出，这种方法的目的是在 n 维的数据空间中找出一组 m 个正交向量，这组向量可最大可能地表示出数据的方差。将数据从原来的 n 维空间投影到由这组正交向量所组成的 m 维子空间上，从而完成维数压缩的作用。

需要指出的是，第一个主分量是沿最大方差方向，第二个主分量则被限制在垂直于第一个主分量的子空间内。在该子空间内，它沿着最大方差方向，以后的主分量按此规则排列下去。通常第 j 个主分量沿自相关矩阵的第 j 个最大本征值所对应的本征向量上。

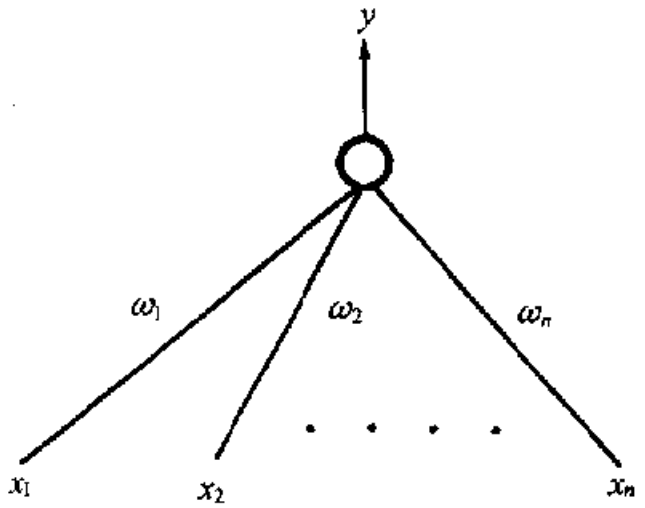


图 7－3

7.5.1 最大主分量的自适应提取

图 7-3 给出的神经网络可以完成最大主分量的自适应提取。

该网络只有一个神经元，其输入为 $x_1, x_2, \cdots, x_n$，权系数为 $w_1, w_2, \cdots, w_n$，该神经元的激励函数为线性函数，显然其输出 y 为

$$y = \sum_{i=1}^{n} w_i x_i = \boldsymbol{w}^T \boldsymbol{x}$$

网络结构确定了，下面的问题是该网络的权向量 $\boldsymbol{w}$ 按什么规则进行学习。

我们定义目标函数为

$$f(\boldsymbol{w}) = \frac{\mathrm{E}[y^2]}{\boldsymbol{w}^T\boldsymbol{w}} = \frac{\boldsymbol{w}^T \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}}$$

$$= \frac{\boldsymbol{w}^T R \boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}}$$

式中，$R = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T]$ 是 $\boldsymbol{x}$ 的自相关矩阵。

为了获取使 $f(\boldsymbol{w})$ 达到最大值的权向量 $\boldsymbol{w}$，可由梯度下降法来实现。

f 相对于 $\boldsymbol{w}$ 的梯度为

$$\nabla f = \frac{2R\boldsymbol{w}(\boldsymbol{w}^T\boldsymbol{w}) - (\boldsymbol{w}^T R \boldsymbol{w})2\boldsymbol{w}}{(\boldsymbol{w}^T\boldsymbol{w})^2}$$

我们规定，权向量 $\boldsymbol{w}$ 为归一化向量：$\boldsymbol{w}^T\boldsymbol{w} = 1$，可得

$$\nabla f = 2R\boldsymbol{w} - 2(\boldsymbol{w}^T R \boldsymbol{w})\boldsymbol{w} \tag{7.17}$$

即

$$\nabla f = 2E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{w} - 2E[y^2]\boldsymbol{w}$$

用样本值代替随机向量，则上式为

$$\nabla f = 2\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{w} - 2y^2\boldsymbol{w}$$

$$= 2\boldsymbol{x}y - 2y^2\boldsymbol{w}$$

$$= 2y(\boldsymbol{x} - y\boldsymbol{w})$$

这样，权向量 $\boldsymbol{w}$ 的修正量为

$$\triangle w = \eta y(x - yw)$$

该网络权向量的修正规则为

$$w(k+1) = w(k) + \eta y(k)(x(k) - y(k)w(k)) \tag{7.18}$$

该网络采用式(7.18)的权向量修正规则，经若干步迭代后，网络收敛。网络收敛后有以下三个结论：

(1) $\| w \|^2 = 1$

网络收敛后，可以认为权向量的修正量$\triangle w$的均值为0，即

$$0 = \frac{E[\triangle w]}{\eta} = E[yx - y^2 w]$$
$$= E[xx^T w - w^T xx^T ww]$$
$$= Rw - [w^T Rw]w$$

即

$$Rw = [w^T Rw]w$$

或

$$Rw = \lambda w$$

上式表明，w是R的本征向量，λ为R的本征值。$\lambda = w^T Rw = w^T \lambda w = \lambda \| w \|^2$，故$\| w \| = 1$.

(2) w位于R的最大本征向量方向上

令φ_1为R的一个归一化本征向量，即

$$R\varphi_1 = \lambda_1 \varphi_1, \quad 且 \quad \| \varphi_1 \| = 1$$

设网络收敛后，w逼近φ_1，即

$$w = \varphi_1 + \varepsilon$$

其中，ε为小扰动项。

由上式得

$$\triangle w = \triangle \varepsilon$$

$$E[\triangle \varepsilon] = E[\triangle w] = \eta E[\nabla f]$$

正系数η不影响本征向量方向，可略去，得

$$
\begin{aligned}
\mathrm{E}[\triangle\varepsilon] &= Rw-(w^TRw)w \\
&= R(\varphi_1+\varepsilon)-[(\varphi_1+\varepsilon)^TR(\varphi_1+\varepsilon)](\varphi_1+\varepsilon) \\
&= R\varepsilon-2\lambda_1[\varepsilon^T\varphi_1]\varphi_1-\lambda_1\varepsilon+0(\varepsilon^2) \\
&\approx R\varepsilon-2\lambda_1[\varepsilon^T\varphi_1]\varphi_1-\lambda_1\varepsilon
\end{aligned}
$$

假定另有一个 R 的归一化本征向量 φ_2，$\varphi_2\neq\varphi_1$，为了计算 $\mathrm{E}[\triangle\varepsilon]$ 在 φ_2 上的投影，对上式两边左乘 φ_2^T，得

$$\varphi_2^T\mathrm{E}[\triangle\varepsilon]=(\lambda_2-\lambda_1)\varphi_2^T\varepsilon \tag{7.19}$$

分两种情况讨论。

第一种情况　$\varphi_2^T\varepsilon>0$

即 ε 在 φ_2 方向上的投影为正，则当 $\lambda_2>\lambda_1$ 时，式(7.19)的右边为正。

第二种情况　$\varphi_2^T\varepsilon<0$

即 ε 在 φ_2 方向上的投影为负，则当 $\lambda_2>\lambda_1$ 时，式(7.19)的右边为负。

综合这两种情况，$\mathrm{E}[\triangle\varepsilon]$ 总是向 φ_2 正方向变化，即 w 总是向较大本征向量方向上变化。因此 网络收敛后，w 必然位于 R 的最大本征向量方向上。

(3) 输出方差最大，即 w 位于使 $\mathrm{E}[y^2]$ 最大的方向上

$$\mathrm{E}[y^2]=w^TRw \tag{7.20}$$

由线性代数知识可知：当 w 位于 R 最大本征向量方向上时，二次型 w^TRw 将达到最大值。

以上结论告诉我们，该网络在式(7.18)学习规则下迭代，权向量 w 将收敛于 R 的最大本征值的归一化本征向量。因此，该网络完成了将 n 维的数据压缩为一维的数据，而且保证压缩结果，均方误差最小。

7.5.2　多主分量的自适应提取

图 7－3 中的神经网络只有一个输出节点，可获得第一主分量。下面考虑图 7－4 所示的具有多个输出节点的神经网络。

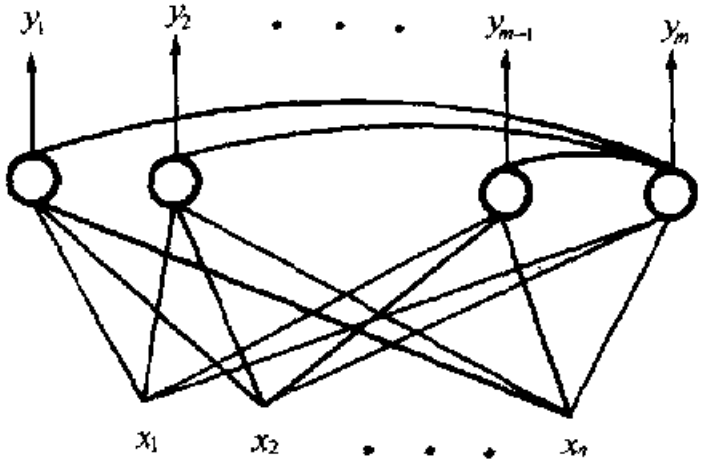


图 7－4

假定网络的前 $m-1$ 个输出神经元的权向量已收敛于样本自相关矩阵 R 的前 $m-1$ 个最大的本征向量，该网络经过学习，第 m 个神经元的权向量可以收敛于 R 的第 m 个最大的本征向量，该权向量与前 $m-1$ 个权向量正交。

假设样本向量为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^T$，由前 $m-1$ 个神经元的输出构成的向量为 $\boldsymbol{y}=(y_1,y_2,\cdots,y_{m-1})^T$，由前 $m-1$ 个神经元的权向量构成的矩阵为 $W=(\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_{m-1})^T$，第 m 个神经元的权向量为 $\boldsymbol{w}_m=(w_{m1},w_{m2},\cdots,w_{mn})$，前 $m-1$ 个神经元与第 m 个神经元的连接权向量为 $\boldsymbol{s}=(s_1,s_2,\cdots,s_{m-1})$。

这样，网络的输入输出关系为

$$\boldsymbol{y}(t)=W(t)\boldsymbol{x}(t) \tag{7.21}$$

$$y_m(t)=\boldsymbol{w}_m(t)\boldsymbol{x}(t)+\boldsymbol{s}(t)\boldsymbol{y}(t)\,. \tag{7.22}$$

我们确定网络权向量修正规则如下：

$$\boldsymbol{w}_m(t+1)=\boldsymbol{w}_m(t)+\beta[y_m(t)\boldsymbol{x}^T(t)-y_m^2(t)\boldsymbol{w}_m(t)] \tag{7.23}$$

$$\boldsymbol{s}(t+1)=\boldsymbol{s}(t)+\gamma[y_m(t)\boldsymbol{y}^T(t)+y_m^2(t)\boldsymbol{s}(t)] \tag{7.24}$$

不难看出，式(7.21)与 7.5.1 节中的权向量修正规则是相同的，式(7.22)具有正交归一化的功能。

下面我们分析这种算法的收敛特性。

假设前 $m-1$ 个神经元的权向量 $\boldsymbol{w}_1(t),\boldsymbol{w}(t)_2,\cdots,\boldsymbol{w}_{m-1}(t)$ 已分别收敛于 R 的前 $m-1$ 个最大的本征向量 $\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\cdots,\boldsymbol{\varphi}_{m-1}$，即

$$W(t)=(\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\cdots,\boldsymbol{\varphi}_{m-1})^T$$

我们将 $\boldsymbol{w}_m(t)$ 展开成

$$\boldsymbol{w}_m(t)=\sum_{i=1}^{n}\theta_i(t)\boldsymbol{\varphi}_i \tag{7.25}$$

将式(7.21)和(7.22)代入式(7.23)，得

$$\boldsymbol{w}_m(t+1)=\boldsymbol{w}_m(t)+\beta[\boldsymbol{w}_m(t)\boldsymbol{x}(t)\boldsymbol{x}^T(t)+\boldsymbol{s}(t)W(t)\boldsymbol{x}(t)\boldsymbol{x}^T(t)-y_m^2(t)\boldsymbol{w}_m(t)]$$

对上式两边求统计平均，可得

$$\begin{aligned}\boldsymbol{w}_m(t+1)&=\boldsymbol{w}_m(t)+\beta[\boldsymbol{w}_m(t)R+\boldsymbol{s}(t)W(t)R-\mathrm{E}(y_m^2(t))\boldsymbol{w}_m(t)]\\&=\boldsymbol{w}_m(t)+\beta[(\boldsymbol{w}_m(t)+\boldsymbol{s}(t)W(t))R-\sigma(t)W_m(t)]\end{aligned} \tag{7.26}$$

式中，

$$\sigma(t)=\mathrm{E}(y_m^2(t)) \tag{7.27}$$

根据式(7.25)和式(7.26)，可得 θ_i 的修正规则为

$$\theta_i(t+1)=\theta_i(t)+\beta\lambda_i\theta_i(t)+\beta s_i(t)\lambda_i\theta_i(t)-\beta\sigma(t)\theta_i(t)$$

即

$$\theta_i(t+1)=[1+\beta(\lambda_i-\sigma(t))]\theta_i(t)+\beta\lambda_i s_i(t) \tag{7.28}$$

式中，λ_i 为 R 的第 i 个本征值。

类似地，对式(7.24)两边求统计平均可得

$$s_i(t+1)=-\gamma\lambda_i\theta_i(t)+[1-\gamma(\lambda_i+\sigma(t))]s_i(t) \tag{7.29}$$

将式(7.28)和(7.29)两式联立，并写成矩阵形式

$$\begin{bmatrix}\theta_i(t+1)\\ s_i(t+1)\end{bmatrix}=\begin{bmatrix}1+\beta(\lambda_i-\sigma(t) & \beta\lambda_i\\ -\gamma\lambda_i & 1-\gamma(\lambda_i+\sigma(t)\end{bmatrix}\begin{bmatrix}\theta_i(t)\\ s_i(t)\end{bmatrix} \tag{7.30}$$

由于 $i\geqslant m$ 时，$s_i(t)=0$，因此我们对式(7.30)的讨论可以分成 $i=1,2,\cdots,m-1$ 和 $i=m,m+1,\cdots,n$ 两种情况来进行。

第一种情况($i\leqslant m-1$)：

如果 $\beta=\gamma$，那么式(7.30)的系数矩阵有一个二重本征值：

$$\rho_i(t) = 1 - \beta\sigma(t)$$

只要 β 足够小，使得 $\rho_i(t)<1$，那么有

$$\lim_{t\to\infty}\theta_i(t) = \lim_{t\to\infty}s_i(t) = 0 \tag{7.31}$$

第二种情况 $(i\geqslant m)$：

此时，式(7.28)变为

$$\theta_i(t+1) = [1 + \beta(\lambda_i - \sigma(t))]\theta_i(t) \tag{7.32}$$

式(7.27)变为

$$\begin{aligned}\sigma(t) &= \mathrm{E}[y_m^2(t)]\\ &= \mathrm{E}[\boldsymbol{w}_m(t)\boldsymbol{x}(t)\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{w}_m^{\mathrm{T}}(t)]\\ &= \boldsymbol{w}_m(t)R\boldsymbol{w}_m^{\mathrm{T}}(t)\\ &= \Big(\sum_{i=1}^{n}\theta_i(t)\boldsymbol{\varphi}_i\Big)R\Big(\sum_{i=1}^{n}\theta_i(t)\boldsymbol{\varphi}_i^{\mathrm{T}}\Big)\\ &= \sum_{i=1}^{n}\lambda_i\theta_i^2(t)\end{aligned}$$

当 $t\to\infty$ 时，有

$$\sigma(t) = \sum_{i=m}^{n}\lambda_i\theta_i^2(t) \tag{7.33}$$

假定 $\theta_m(t)\neq 0$，并令

$$\alpha_i(t) = \frac{\theta_i(t)}{\theta_m(t)}, \qquad i = m+1, m+2, \cdots, n$$

由式(7.32)可得

$$\alpha_i(t-1) = \frac{1 + \beta(\lambda_i - \sigma(t))}{1 + \beta(\lambda_m - \sigma(t))}\alpha_i(t)$$

由于 R 的本征值排序为 $\lambda_1>\lambda_2>\cdots>\lambda_m>\cdots>\lambda_n>0$，所以

$$\frac{1 + \beta(\lambda_i - \sigma(t))}{1 + \beta(\lambda_m - \sigma(t))} < 1$$

进而

$$\lim_{t\to\infty}\alpha_i(t) = 0, \qquad i = m+1, m+2, \cdots, n$$

由于 $\theta_m(t)$ 有界，所以

$$\lim_{t\to\infty}\theta_j(t)=0,\qquad j=m+1,m+2,\cdots,n$$

这样，式(7.33)变为

$$\sigma=\lambda_m\theta_m^2(t)$$

将上式代入式(7.32)可得

$$\theta_m(t+1)=[1+\beta\lambda_m(1-\theta_m^2(t))]\theta_m(t)$$

由上式可得

$$\lim_{t\to\infty}\theta_m(t)=1$$

综合以上两种情况可得：

$$\lim_{t\to\infty}\theta_m(t)=1$$
$$\lim_{t\to\infty}\theta_i(t)=0\quad,\quad i\neq m$$
$$\lim_{t\to\infty}s_i(t)=0$$

这样由式(7.25)可得

$$\lim_{t\to\infty}w_m(t)=\varphi_m$$

至此，我们证明了在 $m-1$ 个神经元权向量收敛于 R 的前 $m-1$ 个最大本征向量 $\varphi_1,\varphi_2,\cdots,\varphi_{m-1}$ 的情况下，上述多主分量提取算法给出的第 m 个神经元的权向量 $w_m(t)$ 将收敛于 φ_m（即 R 的第 m 个最大本征向量）。当 $m=1$ 时，上述算法就是最大主分量提取算法，权向量收敛于 R 的最大本征向量 φ_1.

多主分量自适应提取算法如下：

① 取 $m=1$；

② 随机选择初始值 $w_m(0)$ 和 $s(0)$；

③ 适当选择步长 β 和 γ；

④ 利用式(7.23)和(7.24)计算 $w_m(t)$ 和 $s(t)$；

⑤ 计算误差 $\|w_m(t+1)-w_m(t)\|$ 和 $\|s(t+1)-s(t)\|$，

若误差大于设定值，则转到第④步；

若误差小于设定值，则令 $m=m+1$；此时，若 $m<p$（p 为所需的主分量个数），转第②步，否则算法结束。

上述多主分量提取算法有如下特点：

(1) 算法利用前 $m-1$ 神经元的权向量来递推计算第 m 个神经元的权向量，这样可大大减少算法的计算量。

(2) 由于 $\sigma(t)=\mathrm{E}[y_m^2(t)]$，而 $\lim\sigma(t)=\lambda_m$，因此，从网络的输出就可直接得到自相关矩阵 R 的本征值。

(3) 可以采用变步长学习，对于第 m 个本征向量的提取，可选用步长

$$\beta=\gamma=\frac{1}{N\sigma}$$

这样可加快算法的收敛速度。

第八章　图象特征形成

特征形成与具体识别对象关系较大。本章以图象识别为例，详细介绍图象特征形成。

图象识别的范围很广泛。例如，印刷体或手写体的字母、数字和文字等字符识别；航空或卫星的遥感图象识别；指纹、脸谱识别；应用于工业自动化或军事自动导航、侦察方面的机器视觉图象识别；细胞、血球、染色体等显微图象识别；X -光片、超声影象等识别。

为了适合于输入到数字计算机进行处理和运算，首先必须将一幅景物图象经数字转换器转换成一幅数字化图象。常用的输入设备是显微光密度计、飞点扫描器和电视摄象机的数字化转换器。数字化图象通常用两维光强度函数 $f(x,y)$ 形式表示，此处 x 和 y 是指空间的坐标，而在任意点 (x,y) 上的 f 值正比于图象在该点的灰度级。

一幅数字图象 $f(x,y)$ 指的是在空间坐标上和灰度上都已离散化了的图象。我们可以把一幅数字图象考虑为一个矩阵，其行和列标出了图象中的一个点，而矩阵中相应元素的值则标出了该点的灰度等级。这样的数字阵列中的元素叫图象元素或简称为象素（象点）。

虽然数字图象的大小随具体问题的不同而不同，但在数字化过程中即离散采样的过程中其空间分辨率（即空间的采样间隔距离）是可以选择的。通常一幅数字图象往往选以 2 的整数幂作为方阵列的大小，如 16×16，256×256，1024×1024 等等。灰度分辨率（即灰度等级）也是可以选择的，如 16、32、64、128、256 个灰度级等等。

8.1 图象分割

一幅图象通常是由代表物体的图案与背景组成，简称物体与背景。为了对物体进行特征抽取和识别，首先需要将物体从背景中划分出来。例如，要确定航空照片中的森林、耕地、城市区域等，首先需要将这些部分从图象中分割出来；如果要辨认文件中的文字，也需要先将这些文字分选出来。这种把图象空间按照一定的要求分成一些“有意义”的区域的技术叫做图象分割。

图象分割的数学定义是：对于图象 F，若存在不相交的非空子集 $F_1, F_2, \cdots, F_m$ 使得：

① $\bigcup_{i=1}^{m} F_i = F$；

② 每一个 F_i 是 4－连通或 8－连通域；

③ 对每一个 F_i 来说都满足某种均匀性（或一致性）的要求；

④ 任何数目的相邻子集之和都不满足上述均匀性（或一致性）的要求；

则称 $(F_1, F_2, \cdots, F_m)$ 是 F 的一种分割。

图象分割也可以看成图象象点的分类问题。可以根据象点的灰度等特性判别哪些象点是属于物体和哪些象点是属于背景，或者判别哪些象点是属于各个区域内部的象点和哪些象点是属于区域之间边界的象点。

到目前为止，图象分割的基本方法或者基于区域内部的特性具有某种均匀性的原则，或者基于区域之间的特性具有某种不连续性（或突变性）的原则。具体可归纳为以下几种常见的分割技术：

① 阈值分割技术；

② 区域生长技术；

③ 区域的分裂与合并技术；

④ 边缘检测与边界跟踪技术。

8.1.1 阈值分割技术

图象分割问题可看成根据象点的某种特征将图象中的象点进行分类问题。

设图象 F 是由背景和物体组成。一般情况背景比物体的亮度大，即背景的灰度要比物体的灰度低。于是可采用灰度阈值的方法将物体区域分割出来。例如我们可以选择一个灰度阈值 θ，通过以下定义产生一个新的阈值化后的图象 F_t，其 (x,y) 位置上的象点获得新的灰度级 $f_t(x,y)$ 如下：

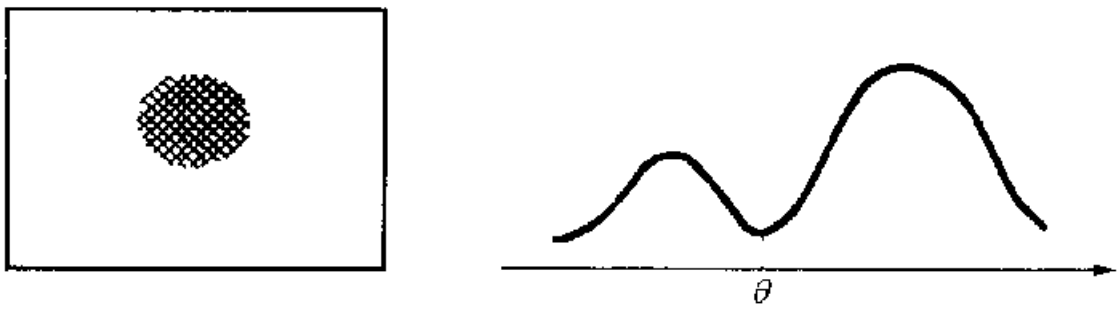
$$f_t(x,y)=\begin{cases}1, & \text{若 } f(x,y)\geqslant\theta\\ 0, & \text{其它}\end{cases}$$

此时新的阈值化后的图象称为二值(0,1)图。此外还可定义另一种阈值化图象：

$$f_t(x,y)\begin{cases}f(x,y), & \text{若 } f(x,y)\geqslant\theta\\ 0, & \text{其它}\end{cases}$$

当阈值 θ 选得过高时，有一部分物体象点将误划为背景，使检出的物体区域偏小；反之又会使分割出来的物体区域偏大。为了将物体能正确地分割出来，其关键在于选择合适的阈值 θ. 对于简单的图象如图8－1(a)，由于物体和背景所处的灰度段明显不同，在图 8－1(b)所示的灰度频数直方图上出现两个峰。此时选择峰之间的谷底灰度作为阈值 θ 将会获得很好的分割效果。

在实际问题中，所遇到的灰度频数直方图往往呈现的不是双峰态而是多峰态的分布，或者呈现峰谷不明显，谷底平坦，双峰不对称等现象。对于这些情况都会给阈值的正确选择带来困难。这时我们可以借助象点的其它特征，比如灰度差分(梯度)，对象点进行分类。对于灰度分布较均匀的区域内部象点一般都具有较低的灰度梯度值，而处于灰度分布不均匀的区域边缘象点往往具有较高的灰度梯度值。因此象点的灰度梯度值可反映象点的特性。假设象点 (x,y) 的灰度梯度为 $\nabla f(x,y)$，在统计灰度频数直方图时



(a)　　(b)

图 8－1

若用 $1/(1+\nabla f(x,y))$对(x,y)象点进行频数加权，即相当于对区域内部象点给予较大的频数加权，而对边缘象点给于较小的频数加权，那么经过加权变换后所获得的直方图将会比原直方图呈现更明显的峰谷。

8.1.2　区域生长技术

区域生长技术的基本思想是，首先在各个区域里寻找一些核心点(或核心区)，通过适当的生长准则将其周围邻近的象点(或子区)归并进来以使区域逐渐生长扩大。

对于区域生长技术一般需要解决两个关键问题：一是起始点(或起始区)的选择；二是生长准则(或称归并准则)的确定。

关于区域生长技术人们提出了许多具体实用的方法，其中区域近邻搜索法是最基本的。现给出该方法的具体实现步骤：

① 将整个图象分成许多等面积的样本单元，其大小可取成 2×2、4×4 或 8×8；

② 计算每个样本单元的灰度统计量；

③ 从左上角第一个单元开始，将它与其相邻的单元逐个进行统计量的比较。若两者相似，符合归并准则，那么就将两者归并形成一个小片，并且计算小片的灰度统计量。若相邻的某单元与它不

相似，不符合归并准则，那么就将该单元标以“未完成”；

④ 继续将小片与其相邻的单元逐个进行统计量的比较。凡是与小片相似者则并入小片使小片逐渐生长扩大，直到没有再可归并的单元为止。然后将此生长完毕的小片标以“已完成区”；

⑤ 对于下一个“未完成”的单元重复上述生长小片的步骤，直到所有的单元都已标为“已完成区”为止。

除灰度外，形状或纹理等也可作为小片生长的依据。

区域生长技术不仅考虑区域的特性具有某种均匀性的要求，而且还考虑到区域连通性的要求，而阈值分割技术并未考虑象点间的空间邻域关系，因此被分割的区域不一定能满足连通性的要求。

8.1.3 区域的分裂与合并技术

分裂与合并算法需用一种特殊的金字塔式（或称四分树式）的数据结构。假设图象由 $2^L \times 2^L$ 个象点组成，那么四分树就有 $L+1$ 层。现以 $L=2$ 为例，图象由 $2^2 \times 2^2 = 16$ 个象点组成，其四分树的数据结构如图8－2所示。

第 0 层有一个节点，它是树的根，对应于整幅图象。第一层有 4 个节点，分别对应于图象的四个等分块 1，2，3 和 4。第 2 层有 16 个节点，它们是树的叶子，每个叶子对应于图象中的一个象点。其中 11，12，13 和 14 分别对应于上一层图象块 1 的四个等分块；……；41，42，43 和 44 分别对应于上一层图象块 4 的四个等分块。

使用分裂与合并算法时必须先确定分裂与合并准则。当某一层中某一块内的象点特征（比如灰度）存在不均匀性时就将该块分裂成为下一层中的四小块，如图 8－3。

当某一层中依次四个小块的象点特征具有某种一致性时就将它们合并成为上一层中的一个稍大的块，如图 8－4。

分裂与合并准则反映了对象点特征的某种均匀性（或一致性）的要求。例如，当四小块的灰度最大值与最小值之差小于某一阈值

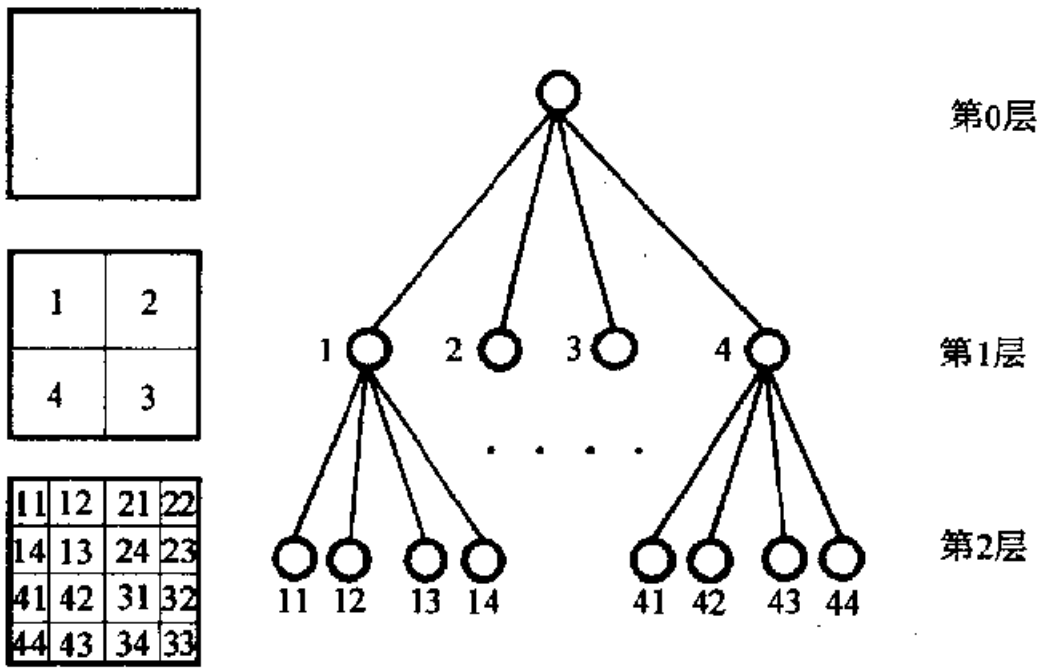


图 8－2

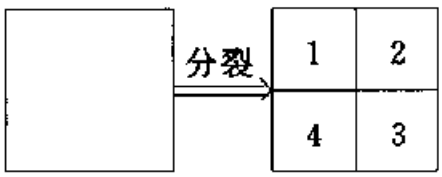


图8－3

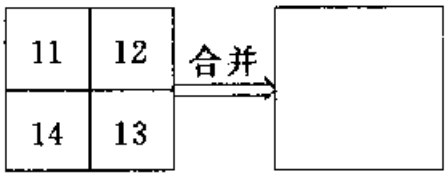


图8－4

即 $f_{max}-f_{min}<T$，则合并；当一块内的 $f_{max}-f_{min}\geqslant T$，则分裂。可根据具体问题设计或选择恰当的准则。

初始分割的选择有以下三种方案：

① 从整个图象（树根）出发，不断地分裂，直到每个块的灰度特性都满足均匀性要求为止；

② 从单个象点(树叶)出发,不断地合并,直到再合并将会使灰度特性出现不均匀性为止;

③ 从中间某层的初始分割开始,有的合并而有的分裂。这种方案最节省计算量。

8.1.4 边缘检测与边界跟踪技术

上面讨论的图象分割技术,都是基于物体和背景的各区域内部在灰度特性上具有某种均匀性,而将物体区域从背景中分割出来。图象分割的另一主要途径就是利用边缘检测技术,这种分割方法是基于物体与背景之间在灰度特性上存在某种不连续性(或突变性),这种不连续性是由于从物体区过渡到背景区存在区域边缘所引起的,从而利用边缘检测技术通过区域边缘的检出同样可以达到分割区域和检出物体的目的。

一、边缘点的检测

"边缘"是指它的两侧分属于两个区域,每个区域的灰度相对均匀一致,而两个区域之间在灰度特性上存在着一定差异。"边缘"实际上是一个边缘点集,它是由一系列边缘点所组成。边缘点是指具有一定边界强度的象点,即灰度在该象点处存在某种程度的变化。下面介绍几种常用的边缘点检测方法。

根据区域的边缘处灰度呈现不连续性的特点,可以通过对图象的灰度特性 $f(x,y)$ 进行微分运算来达到边缘点检测的目的。微分运算的结果表明,凡是灰度迅速变化的点具有较高的微分值。最简单的微分运算是一阶偏导数 $\partial f/\partial x, \partial f/\partial y$。它们给出了灰度在该点的 x 和 y 方向上的变化率。设某象点 (x,y) 的灰度为 $f(x,y)$,该象点的灰度梯度定义如下:

$$\nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right)^T$$

梯度的幅值

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

梯度的方向

$$\theta(x,y) = \mathrm{tg}^{-1}\left[\left(\frac{\partial f(x,y)}{\partial y}\right)/\left(\frac{\partial f(x,y)}{\partial x}\right)\right]$$

对于数字化的离散图象(如图 8－5)可用差分代替微分，一阶差分梯度算子为

$$|\nabla f(x,y)| \approx [(\triangle_x f(x,y))^2 + (\triangle_y f(x,y))^2]^{\frac{1}{2}}$$

$$= [(f(x,y) - f(x,y+1))^2 + (f(x,y) - f(x+1,y))^2]^{\frac{1}{2}}$$

$f(x-1,y-1)$　$f(x-1,y)$　$f(x-1,y+1)$

$f(x,y-1)$　$f(x,y)$　$f(x,y+1)$

$f(x+1,y-1)$　$f(x+1,y)$　$f(x+1,y+1)$

图 8－5

为了计算方便，可用求绝对值近似代替求平方及开方

$$|\nabla f(x,y)| \approx |f(x,y) - f(x,y+1)| + |f(x,y) - f(x+1,y)|$$

或用求最大代替求和

$$|\nabla f(x,y)| \approx \max\{|f(x,y) - f(x,y+1)|, |f(x,y) - f(x+1,y)|\}$$

Robert's 算子是另一种常用的梯度算子。我们在上面求象点(x,y)的梯度时，只用到它的两个邻点$f(x+1,y)$和$f(x,y+1)$的值，而没有用到另一个邻点$f(x+1,y+1)$的值。Robert's 算子为了更充分利用邻域信息而用到了这点的值。Robert's 算子定义为

$$|\nabla f(x,y)| \approx |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)|$$

或

$$|\nabla f(x,y)| \approx \max\{|f(x,y) - f(x+1,y+1)|, |f(x+1,y) - f(x,y+1)|\}$$

拉普拉斯算子是一种较高阶的微分算子，它是对 $f(x,y)$ 求二阶微分

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

同样也可用二阶差分近似地代替二阶微分

$$\nabla^2 f(x,y) \approx f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

为了保证取得非负值，可以使用它的绝对值。

拉普拉斯算子检测边缘的效果不如梯度算子。

下面介绍使用一些模板去和局部图象进行匹配的边缘检测方法。

Sobel 模板是一种常用模板，它由 S_1 和 S_2 两个方向模板组成，如图 8-6 所示。

-1	0	1
-2	0	2
-1	0	1

S_1

1	2	1
0	0	0
-1	-2	-1

S_1

图 8-6

象点 (x,y) 的边界强度计算如下：

$$g(x,y) = \{(g_1(x,y))^2 + (g_2(x,y))^2\}^{1/2}$$

或

$$g(x,y) = \max\{g_1(x,y), g_2(x,y)\}$$

其中 $g_1(x,y)$ 和 $g_2(x,y)$ 为对图象分别采用模板 S_1 和 S_2 进行卷

积运算的结果，即

$$g_1(x,y)=\sum_{k=-1}^{1}\sum_{l=-1}^{1}S_1(k,l)f(x+k,y+l)$$
$$=[f(x-1,y+1)+2f(x,y+1)+f(x+1,y+1)]$$
$$-[f(x-1,y-1)+2f(x,y-1)+f(x+1,y-1)]$$
$$g_2(x,y)=\sum_{k=-1}^{1}\sum_{l=-1}^{1}S_2(k,l)f(x+k,y+l)$$
$$=[f(x-1,y-1)+2f(x-1,y)+f(x-1,y+1)]$$
$$-[f(x+1,y-1)+2f(x+1,y)+f(x+1,y+1)]$$

若 $g_1(x,y)>g_2(x,y)$，说明在象点(x,y)处有垂直方向的边缘通过，反之有水平方向的边缘通过。

KIRSCH 模板由 $K_0\sim K_7$ 共八个方向模板组成，如图 8-7 所示。象点(x,y)的边界强度计算如下：

$$g(x,y)=\max\{g_0(x,y),g_1(x,y),\cdots,g_7(x,y)\}$$

其中

$$g_i(x,y)=\sum_{K=-1}^{1}\sum_{l=-1}^{1}K_i(k,l)f(x+k,y+l)$$

若 $g_i(x,y)$最大，说明在(x,y)处有 $i\cdot 45°$方向上的边缘通过。

5	5	5
−3	0	−3
−3	−3	−3

K_0

−3	5	5
−3	0	5
−3	−3	−3

K_1

−3	−3	5
−3	0	5
−3	−3	5

K_2

−3	−3	−3
−3	0	5
−3	5	5

K_3

−3	−3	−3
−3	0	−3
5	5	5

K_4

−3	−3	−3
5	0	−3
5	5	−3

K_5

5	−3	−3
5	0	−3
5	−3	−3

K_6

5	5	−3
5	0	−3
−3	−3	−3

K_7

图 8-7

当对图象进行微分运算后，每个象点获得一个梯度值 $|\nabla f$

$(x,y)|$，此梯度值直接反映该点的边界强度。当对图象采用模板进行卷积运算后，每个象点同样也获得一边界强度。假设给定一个梯度(或边界强度)阈值 G，那么仅当边界强度 $g(x,y)\geqslant G$ 或 $|\nabla f(x,y)|\geqslant G$ 的那些象点才称为边缘点。

应当指出，边缘检测对图象中的噪声十分敏感。通常在边缘检测之前需对图象进行预处理，例如可采用平滑技术减少或消除局部噪声；采用边缘增强技术增强边缘的反差等。

二、边界的跟踪

上面介绍的边缘点检测方法是一种并行处理技术，即判断一个象点是否是边缘点只根据该点的灰度与其邻域点的灰度之间的差别来决定的，并不取决于其它象点已判定的结果。边界跟踪的方法属于串行处理技术，即在确定新的边界点时需要利用前面已被跟踪过的边界点的信息。由于利用了这些信息，可以在边缘灰度反差小处也能确定边界点，并可以把不在边界上的噪声边缘点去掉。这是边界跟踪法的主要优点。

大多数边界跟踪方法包括下述步骤：

(1) 确定起始点

确定起始点的方法取决已获得的关于边界的信息。如果已知在某区域内有边界点，则只需在此范围内进行搜索来寻找起始点；如果已知边界的方向，则可选择最适合于该方向的边缘检测算子来检测边缘点。如果没有任何先验信息，则需对整幅图象用边缘检测算子从头进行搜索，直到可靠的边缘点被发现。

用于确定起始点的边界强度阈值应取得足够大，以保证选出可靠的边界点。

(2) 预测下一个边界点

下一个边界点的预测依赖于已得到的边界点。通常的作法是，利用刚得到的若干边界点，进行直线拟合，然后在拟合直线的延长线上得出下一边界点的预测位置。以预测位置为中心设置一定大

小的区域，由该区域内的象点构成下一个边界点的候选点集。

(3) 确定下一个边界点

从候选点集中确定下一个边界点要综合考虑候选点的梯度值(或边界强度)和梯度方向。假设 θ 为前面已得到的边界的方向，$\theta(x,y)$ 和 $|\nabla f(x,y)|$ 分别为候选点 (x,y) 的梯度方向和梯度值。可用下面的评价函数作为下一个边界点的判定标准：

$$J(x,y) = |\nabla f(x,y)|\cos(\theta - \theta(x,y))$$

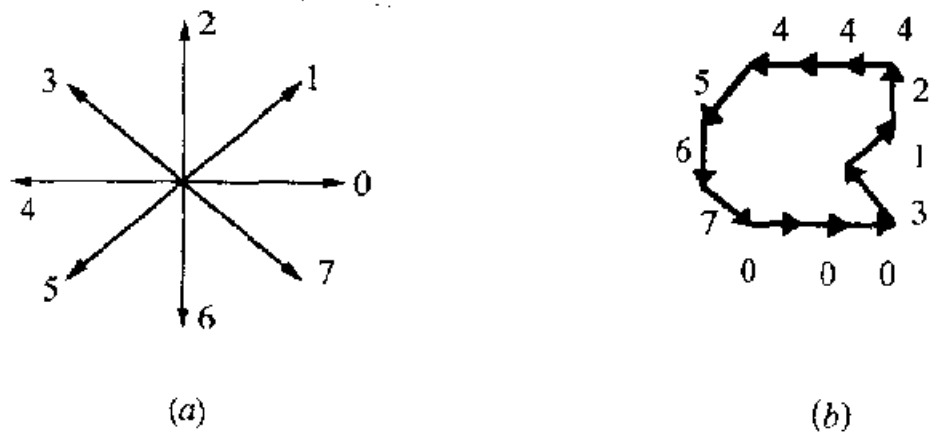
取 $J(x,y)$ 为最大值的候选点作为下一个边界点。

三、边界的链码表示

通过边界跟踪获得一个边界点序列，其输出可以是各边界点的坐标，也可以用 Freeman 链码来表示边界。以链码的方式输出边界可节省存储空间。

链码是边界上相邻象点之间短连接线的方向码序列，有 8 种方向码：0 到 7，其取向如图 8－8(a)所示。偶数方向码表示水平或垂直线段，线段长度为 l(l 为象点间距)。奇数方向码表示对角线段。线段长度为 $\sqrt{2}l$.

对于图 8－8(b)所示的边界，当 s 为起始点，按逆时针方向顺



(a)　　(b)

图 8－8

序获的方向链码为 567000312444.

8.2 线特征描述

经过边缘检测与边界跟踪后，得到了不同区域之间的边界。每一条边界由一个边缘点序列组成。一般来讲边缘点序列不适宜作为分类特征，需要进一步进行参量描述，然后将参量作为分类特征。

8.2.1 分段折线拟合

对边界点序列的一种很有用的表示方法是采用曲线方程描述。一般情况下，一个点序列不适宜用单独一个方程描述。通常可取的作法是将点列分段，各段可较好地由一个方程来近似表征。

采用的方程取决于边界点列的形状。最简单的曲线方程是直线方程。因此，对一条边界点列的最简单的表示方法是采用分段直线来近似。对于一条给定的曲线点列，寻找合适的折线近似的关键是确定出能够产生最佳分段折线的转角位置(亦称角点)。

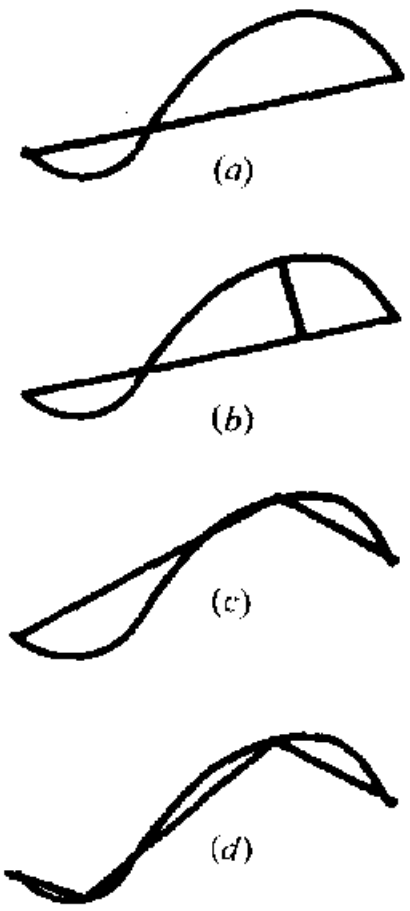


图 8-9

下面介绍一种采用曲线切分来得到分段折线的方法。算法如下：

① 对于给定的点列，设两端点为角点。在两角点间引直线段，如图 8-9(a)所示。

② 对于点列上的每一个点，计算该点与直线段的距离。如果各点距离均在某阈值之内，算法结束；否则，找

出最大距离的点，设其为新的角点。如图8－9(*b*)所示。

③ 在各相邻角点间引直线段，如图8－9(*c*)所示，然后转第②步。

最后的结果示于图8－9(*d*)。这种算法运算简单，而且能用多段直线在设定的误差范围内对任一条曲线进行近似。其缺点是该方法对点列上局部的突变很敏感。

8.2.2 曲线拟合

对边界点列的曲线拟合就是要决定某种函数关系

$$\hat{y}=f(x)$$

使边界点列$\{(x_i,y_i)\}$与函数对应点列$\{(x_i,f(x_i))\}$之间的误差测度为最小。典型的误差测度有

幅度误差 $\varepsilon=\sum_{i=1}^{K}|y_i-f(x_i)|$

均方误差 $\varepsilon=\sum_{i=1}^{K}[y_i-f(x_i)]^2$

峰值误差 $\varepsilon=\max_i\{|y_i-f(x_i)|\}$

式中K为边界点列中的点数。

拟合函数通常取多项式，即

$$\hat{y}=f(x)=a_0+a_1x+a_2x^2+\cdots+a_Nx^N$$

将边界点列$\{(x_i,y_i)\}$中的K个点代入上式，得

$$\begin{bmatrix}\hat{y}_1\\ \hat{y}_2\\ \vdots\\ \hat{y}_K\end{bmatrix}=\begin{bmatrix}1 & x_1 & x_1^2 & \cdots & x_1^N\\ 1 & x_2 & x_2^2 & \cdots & x_2^N\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 1 & x_K & x_K^2 & \cdots & x_K^N\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_N\end{bmatrix}$$

写成矩阵形式为

$$\hat{\boldsymbol{y}}=\boldsymbol{Xa}$$

我们可以采用误差平方和准则函数：

$$J(\boldsymbol{a}) = \| \boldsymbol{y} - X\boldsymbol{a} \|^2/2$$

式中 $\boldsymbol{y}=(y_1, y_2, \cdots, y_K)^T$。

使误差平方和准则函数 $J(\boldsymbol{a})$达到极小的 $\boldsymbol{a}$ 为

$$\boldsymbol{a} = [X^T X]^{-1} X^T \boldsymbol{y}$$

多项式的系数 $a_1, a_2, \cdots, a_N$ 可作为描述该边界的特征。

8.2.3 Hough 变换

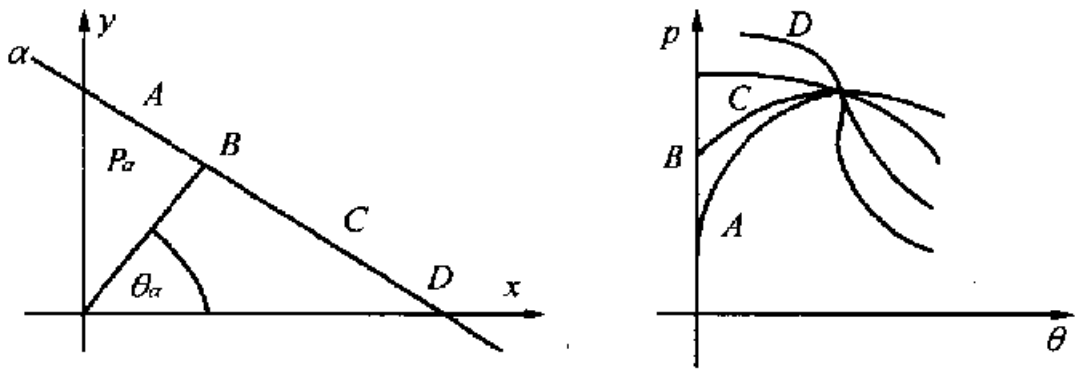
如果在图象空间中某条边界上的点被适当地变换到另一空间(参数空间)，参数空间中的映射点将形成某种聚类形式，每一类中的点表征了一条边界。这种方法称为 Hough 变换法。Hough 变换当初用于直线的检测，后来进行了推广，成为广义 Hough 变换，可用于对曲线的检测。

设 A 是图象上的一个边缘点，如图 8-10(a)所示，通过 A 的直线 α 可表示为

$$x_A\cos\theta_\alpha + y_A\sin\theta_\alpha = p_\alpha$$

其中 θ_α、p_α 为该条直线的参数，由 θ_α 和 p_α 唯一地表示了这条直线。

现在考虑如图 8-10(b)所示的参数空间。两个正交坐标轴分



(a)　　(b)

图 8-10

别为 θ 和 p. 图象空间中的一个点(比如点 A)映射为参数空间中的一条正弦曲线。设图象空间中点 A、B、C、D 位于同一条直线 α 上，则这四点在参数空间分别映射为一条正弦曲线(参看图 8-10)。这四条曲线交会于一点 $\alpha=(\theta_\alpha, p_\alpha)$，该点就是图象空间的直线 α 的映射。

为了应用计算机对数字图象执行上述变换过程，需将参数空间离散化。可将参数 θ 的取值范围$[0, \frac{\pi}{2}]$离散化为 I 级，将参数 p 的取值范围$[0, L]$离散化为 J 级。对图象中每个边缘点 e_k 可计算出一个 0,1 矩阵 $H_k=[h_{ij}]$，如果边缘点 e_k 对应的参数空间中的离散正弦曲线经过位置(i, j)，则 $h_{ij}=1$，否则为 0. 如果图象中有 K 个边缘点 $e_1, e_2, \cdots, e_K$，则可分别计算出矩阵 $H_1, H_2, \cdots, H_K$，然后将所有这 K 个矩阵相加，得到矩阵 $T=(t_{ij})$，元素 t_{ij} 为位置(i, j)上经过的离散正弦曲线的条数，也即图象空间中相应直线上的边缘点数。由于离散化效应以及噪声的影响，一条直线上的边缘点在参数空间中对应的离散正弦曲线并非总是相交于一点，因此矩阵 T 中的元素值会出现聚类形式，形成若干个“峰”，峰顶的位置参数(θ, p)就是图象空间中相应直线的参数。如果有 M 个峰，则表示图象中有 M 个直线段边界，每段边界可用其峰顶位置(θ_m, p_m)来描述。

Hough 变换的一种推广是对形式已知而参数未知的曲线的检测。假设曲线方程为

$$y = ax^2 + c$$

设立一个参数空间，以 a、c 作为正交坐标轴。图象空间中的每个边缘点，映射到参数空间后将是一条直线。其后的处理步骤同前述作法。

Hough 变换方法所依据的是图像的整体信息，因此参数空间中的边缘点聚类对于局部噪声不敏感。这种方法的不足之处在于，如果要检测多参数曲线，那么参数空间的维数将增加，这给参数空

间中的边缘点聚类带来困难。

8.2.4 付里叶描绘子

由于区域的边界是一条封闭的曲线，因此相对于边界上某一固定的起始点 b_0 来说，沿边界曲线上的一个动点 b 的坐标变化则是一个周期函数，周期函数可以展开成付里叶级数。付里叶级数中的系数是与边界曲线的形状有关的，可用作形状的描述，称为付里叶描绘子。当系数项取到足够多的阶次时，几乎可将边界形状信息完全提取出来。

令 C 表示区域 R 的封闭边界曲线。s 表示从 C 上的起始点 b_0 到沿 C 曲线反时针方向上某一动点 b 之间的孤长。S 表示边界曲线 C 的周长。如图 8-11 所示。

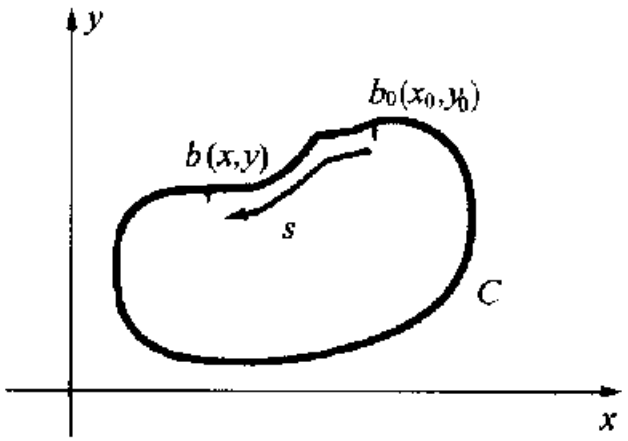


图 8-11

动点 b 的坐标 $b(x(s),y(s))$既是 x,y 的函数，又是孤长 s 的函数。曲线的参数方程可用复数形式表示为

$$U(s) = x(s) + jy(s) \tag{8.1}$$

它是一个周期函数，即

$$U(s+S) = U(s), \qquad 0 \leqslant s \leqslant S$$

现设 $t=2\pi s/S$，则方程(8.1)可改写为

$$U(t)=x(t)+jy(t), \qquad 0\leqslant t<2\pi$$

式中的 $U(t)$ 是一个以 2π 为周期的周期函数。它的付里叶展开式为

$$U(t)=\sum_{n=-\infty}^{+\infty}p_n e^{jnt}=p_0+\sum_{n=1}^{\infty}(p_n e^{jnt}+p_{-n}e^{-jnt}),$$
$$0\leqslant t<2\pi \qquad (8.2)$$

式中的付里叶系数为

$$p_n=\frac{1}{2\pi}\int_0^{2\pi}U(t)e^{-jnt}\mathrm{d}t, \qquad n=0,\pm 1,\pm 2,\cdots\cdots \quad (8.3)$$

在数字图象中，区域的边界可用沿边界曲线 C 反时针方向的方向链码 $c_1c_2\cdots c_M$ 来表示。

将[0,2π]区间划分为

$$t_m=2\pi s_m/S, \qquad m=0,1,2,\cdots,M$$

即 $0=t_0<t_1<t_2<\cdots<t_{M-1}<t_M=2\pi$。由式(8.3)可得：

$n\neq 0$ 时

$$\begin{aligned}p_n&=-\frac{1}{2n\pi j}U(t)e^{-jnt}\Big|_0^{2\pi}+\frac{1}{2n\pi j}\int_0^{2\pi}e^{-jnt}\mathrm{d}U(t)\\&=\frac{1}{2n\pi j}\int_0^{2\pi}e^{-jnt}\mathrm{d}U(t)\\&\approx\frac{1}{2n\pi j}\sum_{m=1}^{M}e_m^{-jnt}[U(t_m)-U(t_{m-1})]\end{aligned} \qquad (8.4)$$

$n=0$ 时

$$\begin{aligned}p_0&=\frac{1}{2\pi}\int_0^{2\pi}U(t)\mathrm{d}(t)\\&\approx\frac{1}{2\pi}\sum_{M=1}^{M}U(t_m)(t_m-t_{m-1})\\&=\frac{1}{2\pi}U(t_M)t_M+\frac{1}{2\pi}\sum_{m=1}^{M-1}U(t_m)t_m-\frac{1}{2\pi}U(t_1)t_0\\&\quad-\frac{1}{2\pi}\sum_{m=2}^{M}U(t_m)t_{m-1}\end{aligned}$$

$$= U(t_0) - \frac{1}{2\pi}\sum_{m=2}^{M}[U(t_m) - U(t_{m-1})]t_{m-1} \qquad (8.5)$$

式中 $U(t_0)=U(0)=x_0+jy_0$ 对应于起始点 b_0，因此 p_0 项与坐标有关。

为了建立链码与付里叶系数间的关系，设

$$a_k = \begin{cases} 1 & \text{，若 } c_k \text{ 为偶数} \\ \sqrt{2} & \text{，若 } c_k \text{ 为奇数，} \end{cases} \qquad k = 1,2,\cdots,M$$

周长

$$S \approx \sum_{k=1}^{M} a_k \qquad (8.6)$$

参变量

$$t_m = \frac{2\pi s_m}{S} \approx 2\pi \sum_{k=1}^{m} a_k / \sum_{k=1}^{M} a_k, \quad m = 1,2,3,\cdots,M \qquad (8.7)$$

式(8.4)和(8.5)中的 $U(t_m)-U(t_{m-1})$ 是从 t_m 到 t_{m-1} 由方向码 C_m 对应的一个既有幅度又有幅角的向量，幅度为 a_m，幅角为 $\frac{\pi}{4}c_m$，因此该向量可表示为

$$U(t_m) - U(t_{m-1} \approx a_m e^{j\frac{\pi}{4}c_m}, \quad m = 1,2,\cdots,M \qquad (8.8)$$

将式(8.7)和式(8.8)代入式(8.5)后得

$$p_0 \approx U(0) - \sum_{m=2}^{M} a_m e^{j\frac{\pi}{4}c_m}\left(\sum_{k=1}^{m-1} a_k / \sum_{k=1}^{M} a_k\right)$$

将式(8.7)和式(8.8)代入式(8.4)后得

$$p_n \approx \frac{1}{2n\pi j}\sum_{m=1}^{M} a_m e^{j\left(\frac{\pi}{4}c_m - 2n\pi \sum_{k=1}^{M} a_k / \sum_{k=1}^{M} a_k\right)}$$

$$n = \pm 1, \pm 2, \cdots$$

对闭合曲线进行付里叶描绘还有其它作法。例如，曲线函数不是取作复平面上的位置向量 $U(s)$，而是取作曲线的曲率函数 $K(s)$，或曲线与内部某点(如形心)的距离函数 $D(s)$等。

8.3 区域特征描述

通过图象分割，已将有意义的区域分割出来了，对分割出来的区域的描述对图象识别是十分有用的。

8.3.1 几何特征

区域的几何特征是反映区域的几何性质的，它仅与区域的象点有关而与这些象点的灰度无关。因此这类特征一般可在二值(0，1)图象或边界链码上进行度量。下面介绍常用的几何特征。

(1) 区域的面积

区域的面积 A 为区域内象点的总数。

(2) 区域的周长

假设区域的边界链码为 $c_1c_2\cdots c_M$，那么区域的周长 P 为

$$P = l\sum_{i=1}^{M}\left(\sqrt{2}\right)^{d_i}$$

式中，l 为象点间距，$d_i = c_i \text{ MOD } 2$.

(3) 区域的密集度

密集度为区域周长的平方与其面积的 4π 倍之比，即

$$C = \frac{P^2}{4\pi A}$$

在相同面积的条件下，具有光滑周界的圆形的周长最短，可称为最密集的形状，$C=1$. 随着周界凹凸变化加剧，周长增大，C 值随之加大。

(4) 宽长比

宽长比为区域最小外接矩形的宽 W 与长 L 之比。方形或圆形区域的宽长比都接近于 1，而细长的区域则宽长比将小于 1。

(5) 占空比

占空比为区域面积 A 与区域最小外接矩形面积 A_R 之比，即

$$R = \frac{A}{A_R}$$

对于矩形区域，R 取最大值 1；对于圆形区域，$R=\pi/4$；对于弯曲的细长区域或凹区域，R 值将较小。

以上这些特征直观性强，计算简单。缺点是所提取的区域形状信息不完全，特征值与具体形状之间不是一一对应的关系，而是一对多的关系，不适用于鉴别形状的细节。

8.3.2 矩

当一个区域是以其内部点的形式给出时，还有其它一些描述区域的方法，用“矩”来描述就是其中的一种。

给定二维连续函数 $f(x,y)$，其 $(p+g)$ 阶原点矩为

$$m_{pg} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^g f(x,y)\mathrm{d}x\mathrm{d}y$$

式中 $p,q=0,1,2,\cdots$。

相应的 $(p+g)$ 阶中心矩可表示为

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\mathrm{d}x\mathrm{d}y$$

式中

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad , \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{8.9}$$

根据唯一性定理可知，如果 $f(x,y)$ 是分段连续的，而且只在 xy 平面的有限部分中有非零值，则所有各阶皆存在，并且矩序列 m_{pq} 或 μ_{pq} 也唯一地被 $f(x,y)$ 所确定；反之，m_{pq} 或 μ_{pq} 也唯一地确定了 $f(x,y)$.

对于数字图象可用求和代替积分：

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{8.10}$$

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)$$

三阶以内的中心矩为

$$\mu_{00}=m_{00}, \qquad \mu_{11}=m_{11}-\bar{y}m_{10}$$

$$\mu_{10}=0, \qquad \mu_{30}=m_{30}-3\bar{x}m_{20}+2\bar{x}^2m_{10}$$

$$\mu_{01}=0, \qquad \mu_{03}=m_{03}-3\bar{y}m_{02}+2\bar{y}^2m_{01}$$

$$\mu_{20}=m_{20}-\bar{x}m_{10}, \qquad \mu_{12}=m_{12}-2\bar{y}m_{11}-\bar{x}m_{02}+2\bar{y}^2m_{10}$$

$$\mu_{02}=m_{02}-\bar{y}m_{01}, \qquad \mu_{21}=m_{21}-2\bar{x}m_{11}-\bar{y}m_{20}+2\bar{x}^2m_{01}$$

例如：

$$\mu_{12}=\sum_x\sum_y(x-\bar{x})(y-\bar{y})^2f(x,y)$$

$$=\sum_x\sum_y(xy^2-2x\bar{y}y+x\bar{y}^2-\bar{x}y^2+2\bar{x}y\bar{y}-\bar{x}\bar{y})f(x,y)$$

利用式(8.9)和式(8.10)可得

$$\mu_{12}=m_{12}-2\bar{y}m_{11}-\bar{x}m_{02}+2\bar{y}^2m_{10}$$

零阶原点矩 m_{00}可以看作区域内的灰度质量，$\bar{x}$ 和 $\bar{y}$ 便是区域灰度重心的坐标。

中心矩 μ_{pq}是反映区域中的灰度相对于灰度重心是如何分布的度量。例如 μ_{20}和 μ_{02}分别表示区域围绕通过灰度重心的垂直和水平轴线的惯性矩。若 $\mu_{20}>\mu_{02}$，那么这可能是一个水平方向拉长的区域。μ_{30}和 μ_{03}的幅值可以度量区域对于垂直和水平轴线的不对称性。如果是完全对称的形状，其值应为零。

用零阶中心矩对其余各中心矩进行规格化，可得到规格化中心矩

$$\eta_{pq}=\mu_{pq}/\mu_{00}^{r}$$

式中 $r=\dfrac{p+q}{2}$ ， $p+q=2,3,\cdots$

利用二阶和三阶规格化中心矩可以导出下面七个不变矩组：

$$\varphi_1=\eta_{20}+\eta_{02}$$

$$\varphi_2=(\eta_{20}-\eta_{20})^2+4\eta_{11}^2$$

$$\varphi_3=(\eta_{30}-3\eta_{12})^2+(3\eta_{21}+\eta_{03})^2$$

$$\varphi_4=(\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2$$

$$\varphi_5=(\eta_{30}-3\eta_{12})(\eta_{30}-\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]$$
$$+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]$$
$$\varphi_6=(\eta_{20}-\eta_{02})[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]$$
$$+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})$$
$$\varphi_7=(3\eta_{12}-\eta_{30})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]$$
$$+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{03}+\eta_{12})^2-(\eta_{12}+\eta_{03})^2]$$

我们常常希望特征在平移、旋转、尺度和起始点等条件变化下是一个不变量。可以证明这个矩组对平移、旋转和大小比例变化都是不变的。

参考文献

[1] Tou J T, Gonzalez R C. Pattern Recognition principles. Massachusetts: Addison-Wesley, 1974.

[2] Duda R O, P. E. Hart P E. Pattern Classification and Scene Analysis. New York: john Wiley & Son, 1973.

[3] Sklanky J, Wassel G N. Pattern clasifiers and Trainable Machines. New york: Springer—Verlag, 1981.

[4] Zurada J M. Introduction to Artificial Neural Systems. Los Angeles: West Publishing Company, 1992.

[5] Carpenter G A, Grcssberg S, Rosen D. Fuzzy ART: Fast Stable learning and Categorization of Analog Patterns by an Adaptive Resonance System. Neural Networks, 1991(4): 759—771

[6] Kim Y S, Mitra S. An Aaptive Integrated Fuzzy Clustering Model for Pattern Recognition. Fuzzy Sets and Systems, 1994(2): 297—310

[7] (日)福永圭之介著. 统计图形识别导论. 陶笃纯 译, 北京, 科学出版社, 1978.

[8] 蔡元龙. 模式识别. 西安: 西北电讯工程学院出版社, 1986.

[9] 边肇祺. 模式识别. 北京: 清华大学出版社, 1988.

[10] 杨行峻, 郑君里. 人工神经网络. 北京: 高等教育出版社, 1992.

[11] 焦李成. 神经网络的应用与实现. 西安: 西安电子科技大学出版社, 1995.

[12] 李洪兴, 汪培庄 编著. 模糊数学. 北京: 国防工业出版社, 1994.

[13] 黄凤岗. 句法统计距离及其在联机手写体汉字识中的应用. 哈尔滨船舶工程学院学报, 1991, 12(1): 57—61

[14] 黄凤岗 等. 基于离散马尔可夫模型的三维运动目标识别. 模式识别与人工智能, 1996, 9(1): 91—95

[15] 黄凤岗 等. 一种集成模糊聚类神经网络. 哈尔滨工程大学学报, 1997, 18(3): 82—85