全国第七届研究生数学建模竞赛

题 目 **肿瘤基因图谱信息提取和分类方法研究**

摘 要：

本文讨论了肿瘤基因图谱信息提取方法，主要工作是提取结肠癌基因表达谱的特征基因信息，并利用神经网络进行分类识别。

对于问题一、二，本文采用了一种致癌基因信息提取与分类方法，该方法主要由四步构成：（1）利用 Bhattacharyya 距离法滤除无关基因；（2）采用两两冗余分析，剔除强相关冗余基因；同时，采用主成分分析方法对所选择的基因样本数据进行降维处理，得到样本的主成分量；（3）对特征提取后的基因数据采用四种神经网络（BP 神经网络及三种径向基神经网络）进行分类学习，训练获得分类网络模型；（4）采用获得的分类网络模型对测试肿瘤样本进行分类，并采用留一交叉检验法和独立检验法评估四种神经网络分类器性能。结果表明：本文所采用的特征提取方法能有效提出与肿瘤相关的信息基因，选取的特征基因子集包含 9 个基因，且采用概率神经网络（PNN）的分类识别准确率（77.27%）最高。

对于问题三，采用小波去噪方法消除基因信息采集过程中的随机误差。为最大限度地消除噪声并保证原始数据信息损失最小，本文分别采用 db3、db5、sym8、haar 等不同小波基进行去噪，通过对概率神经网络分类的结果比较可知：选择 haar 小波基对所有数据进行消噪，提取到的特征信息基因更为有效。在独立测试实验中，概率神经网络对 22 个样本数据的识别准确率为 100%。

对于问题四，本文采用信息融合的方法，利用加权评分法建立了融入生理学确定信息基因的多信源信息融合模型。通过对 PNN-WG 模型求解，验证了提出融合模型可将原有概率神经网络的分类准确性从 77.27%提高到 86.36%，达到了多源信息融合的目的。

最后，评价了本文模型和算法的优点，并给出了进一步研究的方向。

关键词：肿瘤基因；特征提取；分类识别；小波去噪；信息融合

参赛队号 10613001

队员姓名 臧天磊 邹大云 黄飞

参赛密码 _____

（由组委会填写）

中山大学承办

一、问题重述

癌症起源于正常组织在物理或化学致癌物的诱导下，基因组发生的突变，即基因在结构上发生碱基对的组成或排列顺序的改变，因而改变了基因原来的正常分布（即所包含基因的种类和各类基因以该基因转录的 mRNA 的多少来衡量的表达水平）。所以探讨基因分布的改变与癌症发生之间的关系具有深远的意义。

DNA 微阵列（DNA microarray），也叫基因芯片，是最近数年发展起来的一种能快速、高效检测 DNA 片段序列、基因表达水平的新技术。它将数目从几百个到上百万个不等的称之为探针的核苷酸序列固定在小的玻璃或硅片等固体基片或膜上，该固定有探针的基片就称之为 DNA 微阵列。根据核苷酸分子在形成双链时遵循碱基互补原则，就可以检测出样本中与探针阵列中互补的核苷酸片段，从而得到样本中关于基因表达的信息，这就是基因表达谱，因此基因表达谱可以用一个矩阵或一个向量来表示，矩阵或向量元素的数值大小即该基因的表达水平。

随着大规模基因表达谱技术的发展，人类各种组织的正常的基因表达已经获得，各类病人的基因表达分布图都有了参考的基准，因此基因表达数据的分析与建模已经成为生物信息学研究领域中的重要课题。从 DNA 芯片所测量的成千上万个基因中，找出决定样本类别的一组基因“标签”，即“信息基因”是正确识别肿瘤类型、给出可靠诊断和简化实验分析的关键所在。

通常由于基因数目很大，在判断肿瘤基因标签的过程中，需要剔除掉大量“无关基因”，从而大大缩小需要搜索的致癌基因范围。事实上，在基因表达谱中，一些基因的表达水平在所有样本中都非常接近。因此，必须对这些“无关基因”进行剔除。

但信噪比肯定不是衡量基因对样本分类贡献大小的唯一标准，肿瘤是致癌基因、抑癌基因、促癌基因和蛋白质通过多种方式作用的结果，在确定某种肿瘤的基因标签时，应该设法充分利用其他有价值的信息。有专家认为在基因分类研究中忽略基因低水平表达、差异不大的表达的倾向应该被纠正，与临床问题相关的主要生理学信息应该融合到基因分类研究中。

面对提取基因图谱信息这样前沿性课题，以下几点是解决前沿性课题的有价值的工作。

（1）由于基因表示之间存在着很强的相关性，所以对于某种特定的肿瘤，似乎会有大量的基因都与该肿瘤类型识别相关，但一般认为与一种肿瘤直接相关的突变基因数目很少。对于给定的数据，如何从上述观点出发，选择最好的分类因素？

（2）相对于基因数目，样本往往很小，如果直接用于分类会造成小样本的学习问题，如何减少用于分类识别的基因特征是分类问题的核心，事实上只有当这种特征较少时，分类的效果才更好些。对于给定的结肠癌数据如何从分类的角度确定相应的基因“标签”？

（3）基因表达谱中不可避免地含有噪声，有的噪声强度甚至较大，对含有噪声的基因表达谱提取信息时会产生偏差。通过建立噪声模型，分析给定数据中的噪声能否对确定基因标签产生有利的影响？

（4）在肿瘤研究领域通常会已知若干个信息基因与某种癌症的关系密切，建立融入了这些有助于诊断肿瘤信息的确定基因“标签”的数学模型。

二、问题分析

本文问题的关键是解决如何从基因表达数据中提取肿瘤分类特征信息以达到对基因表达谱数据进行大幅度降维的目的。

对于第一个问题，由于基因表示之间存在着很强的相关性，所以对于某种特定的肿瘤，似乎会有大量的基因都与该肿瘤类型识别相关，但一般认为与一种肿瘤直接相关的突变基因数目很少。所以，首先应基于给定数据，采用某一种基因排序方法进行基因初选。通常根据数据分布得到的经验值得到选择信息基因的个数。

对于第二个问题，相对于基因数目，样本往往很小，如果直接用于分类会造成小样本的学习问题，如何减少用于分类识别的基因特征是分类问题的核心，事实上只有当这种特征较少时，分类的效果才更好些。所以，第二步需要采用特征提取方法从初选出的信息基因子集中提取分类特征信息，因为这些初选出来的信息基因相互之间存在高度的相关性，而具有这个特点的数据集也适合于采用诸如主成分分析这类降维方法。

对于第三个问题，由于基因表达谱中不可避免地含有噪声，有的噪声强度甚至较大，对含有噪声的基因表达谱提取信息时会产生偏差。所以，需要通过建立噪声模型，分析给定数据中的噪声能否对确定基因标签产生有利的影响。

对于第四个问题，在肿瘤研究领域通常会已知若干个信息基因与某种癌症的关系密切，所以需要采用信息融合的方法，建立融入有助于诊断肿瘤信息的确定信息基因的数学模型。

三、模型假设

1、所给基因数据不含奇异数据；

2、基因的功能与作用是多个基因集体作用的结果；

3、与一种肿瘤直接相关的突变基因数目很少；

4、基因表达谱中含有噪声主要由基因采集过程中随机性产生的误差构成；

5、本文利用的临床生理学信息：大约 90%结肠癌在早期有 5 号染色体长臂 APC 基因的失活，而只有 40%~50%的 ras 相关基因突变。可看作基于 APC 基因的分类可信度为 0.9，而基于 ras 相关基因的分类可信度为 0.4~0.5。

四、符号约定

$B(g)$：基因 g 的 Bhattacharyya 距离；

$Corr\_Coef(g_i, g_j)$：基因 g_i, g_j 在训练样本集中表达水平间的 Pearson 相关系数；

R：相关系数矩阵；

$p_k = (a_1, a_2, \ldots, a_n)$：网络输入向量；

$T_k = (y_1, y_2, \ldots, y_q)$：网络目标向量；

$S_k=(s_1,s_2,\ldots,s_p)$：中间层单元输入向量；

$B_k=(b_1,b_2,\ldots,b_p)$：中间层单元输出向量；

$L_k=(l_1,l_2,\ldots,l_q)$：输出层单元输入向量；

$C_k=(c_1,c_2,\ldots,c_q)$：输出层单元输出向量；

w_{ij}：输入层至中间层的连接权，$i=1,2,\cdots,n;\ j=1,2,\cdots,p$；

v_{it}：中间层至输出层的连接权，$j=1,2,\cdots,p;t=1,2,\cdots,p$；

θ_j：中间层各单元的输出阈值，$j=1,2,\cdots,p$；

γ_j：输出层各单元的输出阈值，$j=1,2,\cdots,p$；

α_i：信息基因的可信度，$i=1,2,\cdots,n$；

λ_i：神经网络输出结果赋予权值，$i=1,2,\cdots,n$；

$S=(S_1,S_2,\cdots,S_p)$：加权评分向量；

$f(i)$：真实信号；

$e(i)$：信号中的噪声；

$s(i)$：含噪声的信号。

五、问题一、二的建模与求解

肿瘤分类特征基因选取的目的在于从原始基因集合中提取出一组最能反映样本分类特性的基因以准确地刻画出事物的分类模型，从而为最终确定肿瘤分类与分型的基因标记物提供可靠线索。该特征基因集合应包含尽可能完整的样本分类信息，即不丢失原始基因集合中所蕴含的样本分类信息，可利用有效的分类器实现对基因样本的准确分类。

鉴于基因表达数据存在维数高、噪音大、样本数量小以及基因表达之间存在很大相关性等特点，本文设计了一种致癌基因信息提取与分类方法。该方法的框架模型主要由下述五步构成：

Step1 信息基因选择。采用 Bhattacharyya 距离衡量基因含有样本分类信息的多少，滤除无关基因；

Step2 冗余基因剔除。采用两两冗余分析，剔除强相关冗余基因；

Step3 提取主成分分量。采用主成分分析（PCA）方法对所选择的基因样本数据进行降维处理，得到样本的主成分分量；

Step4 分类模型训练和最优基因组合筛选。对特征提取后的基因数据形成的 2^N 个候选基因子集分别采用神经网络（BP 神经网络及三种径向基神经网络）进行分类学习，训练网络权值，得到分类网络模型和最优基因组合；

Step5 测试分类模型。采用获得的分类网络模型对测试肿瘤样本进行分类，并采用留一检验法和独立检验法评估四种神经网络分类器性能。

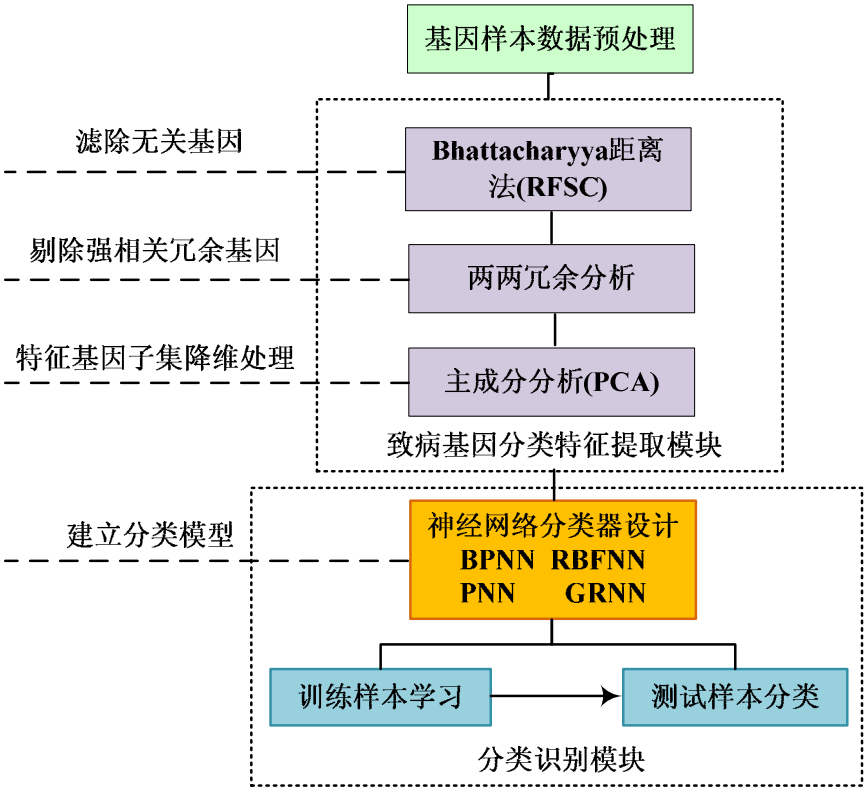


图 5-1 致癌基因信息提取与分类方法框架

5.1 数据的预处理

5.1.1 均值中心化

由于样本中存在大量的来源于一个基因样本的多次重复试验数据，为保证在特征提取和分类时，这些数据之间的相关性不对结果产生较大影响，本文对这些数据进行均值中心化处理，该过程同时可在一定程度上消除数据的系统偏差。处理后，原始数据由 2000 维降到 1909 维。

5.1.2 归一化

本文使用的特征提取方法和人工神经网络分类识别算法要求首先对输入数进行归一化处理。

$$x_{ik} = (x_{ik} - \mu_k) / \sigma_k \qquad i = 1, 2, \ldots, m; \quad k = 1, 2, \ldots, p \tag{5-1}$$

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x_{ik} \tag{5-2}$$

$$\sigma_k^2 = \frac{1}{m-1} \sum_{i=1}^{m} (x_{ik} - \mu_k)^2 \tag{5-3}$$

5.2 类别无关基因的滤除

由于只有少数基因与样本某一特定的表型（生物类别）相关，其余大部分基因是同该表型无关的“类别无关基因”，或者可以理解为“噪声基因”。为有效选取样本的分类特征，本文首先利用基因之间的 Bhattacharyya 距离作为衡量基因含有样本分类信息多少的度量。

Bhattacharyya 距离[1]体现了属性在两个不同样本中分布的差异，这种差异既

包含了属性在不同类别分布均值的差异，同时也考虑了样本分布方差不同对分类的贡献。其具体模型为：

$$B(g)=\frac{(\mu_+(g)-\mu_-(g))^2}{4(\sigma_+^2(g)+\sigma_-^2(g))}+\frac{1}{2}\ln\left(\frac{\sigma_+^2(g)+\sigma_-^2(g)}{2\sigma_+(g)\sigma_-(g)}\right) \qquad (5\text{-}5)$$

式中 μ_+ 和 μ_- 分别为基因 g 在两类不同样本中的表达水平的均值，σ_+ 和 σ_- 为相应的标准差。基因的 Bhattacharyya 距离越大，该基因在两类样本中表达水平的分布差异也就越大，对样本分类的能力也就越强。

根据公式（5-5）计算了每个基因的 Bhattacharyya 距离，并作出了基因的 Bhattacharyya 距离分布的直方图，如图 5-2 所示。

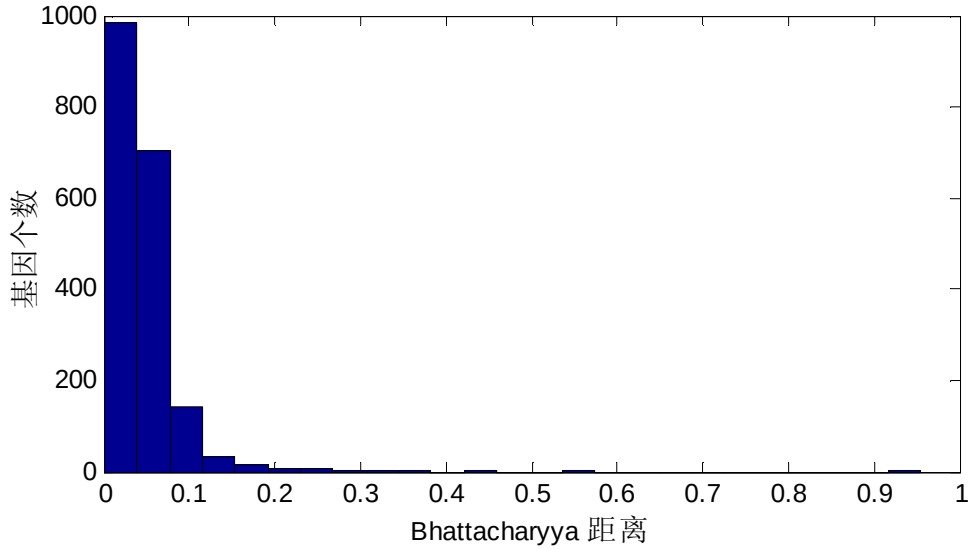


图 5-2 基因 Bhattacharyya 距离分布的直方图

本文选择 $B(g)<0.1$ 的 1709 个具有较小 Bhattacharyya 距离的基因作为类别无关基因予以剔除，余下的 200 个基因作为进一步分析的基础。其 Bhattacharyya 距离值可认为是基因信息指数。

5.3 强相关性冗余基因的剔除

从分类的角度看，上文得到 200 个基因就可以作为分类特征基因。然而，在这 200 个基因中还可能存在冗余，这些冗余基因的存在与否并不会影响到整个分类特征基因集合的样本分类能力。因此，本文进一步应用两两冗余分析算法[2]计算初选后的任意两个基因表达水平间的相关系数，若其相关系数大于指定阈值，则认为两个基因是强相关的，排除二者中分类信息指数较小的那个基因，使排除冗余后的分类特征基因集合具有较大的分类信息指数。

两两冗余分析算法的伪代码如下：

（1） 对 *m* 个基因按分类信息指数由大到小排序，得到有序基因集合
F = {g_1 , g_2 , …, g_m}

（2） FSet = {g_1}

（3） **For** *i* = 1 to *card* (F)
(a) g_i = F(*i*)
(b) Corr = False
(c) **For** *j* = 1 to *card*(FSet)

g_j = FSet (j)

If Corr_Coef (g_i, g_j) ≥Threshold **Then** Corr = True

(d) **If** Corr = False **Then** FSet = FSet ∪ {g_i}

(4) **Return** Fset

其中，*card* F 为 F 的势，$Corr\_Coef(g_i, g_j)$ 用来计算基因 g_i, g_j 在训练样本集中表达水平间的 Pearson 相关系数，具体计算公式如下：

$$Corr\_Coef(g_i, g_j) = \frac{\sum_{k=1}^{n}(x_{gik} - \overline{x}_{gi})(x_{gik} - \overline{x}_{gj})}{\sqrt{\sum_{k=1}^{n}(x_{gik} - \overline{x}_{gi})^2 \sum_{k=1}^{n}(x_{gik} - \overline{x}_{gj})^2}} \quad (5\text{-}6)$$

式中 x_{gik}， x_{gjk} 为基因 g_i、g_j 在训练集第 k 个样本中的表达水平值，$\overline{x}_{gi}$，$\overline{x}_{gj}$ 分别为 g_i、g_j 在训练集所有样本中表达水平的均值。Threshold 为指定的相关系数阈值。

两两冗余分析算法的程序流程如下图所示：

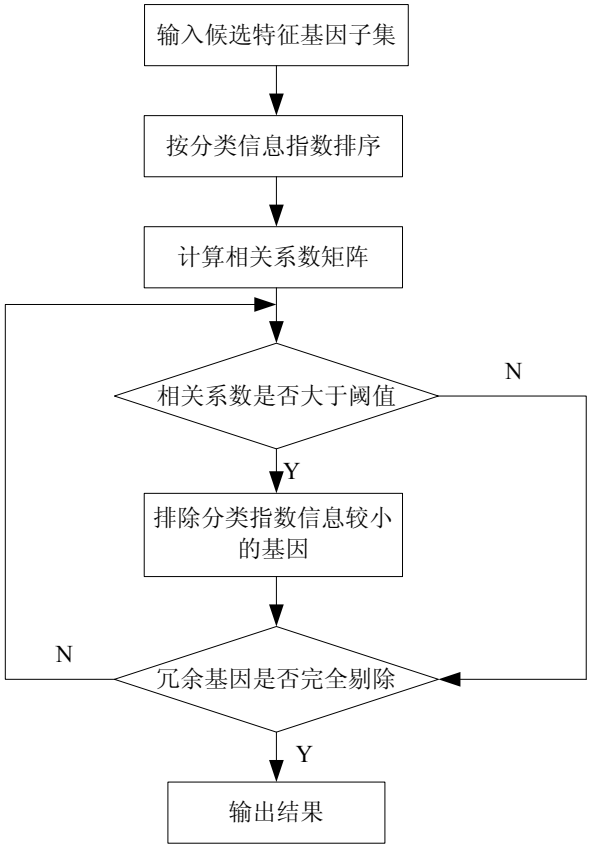


图 5-3 两两冗余分析算法的程序流程

本文选用阈值为 0.5，最后得到 27 个信息基因。

5.4 基于主成分分析的降维处理

上述特征选择方法提取出信息基因维数仍然较高，所以需要进行降维处理，即用较少的几个综合指标来代替原来较多的变量指标，而且使这些较少的综合指标既能尽量多地反映原来较多指标所反映的信息，同时它们之间又是彼此独立的。主成分分析是把原来多个变量化为少数几个综合指标的一种统计分析方法，

本文将利用主成分分析法（PCA）[3,4]对信息基因进行降维处理。

对$\mathbf{M}_s$进行主成分分析并从中提取主成分分量。为使样本集$\mathbf{M}_s$在降维过程中所引起的平方误差最小，必须进行两方面的工作：一是用雅可比方法求解正交变换矩阵；二是选取w个主成分分量，$w<p$。PCA 的计算过程主要分三步进行：

Step1 将矩阵$\mathbf{M}_s$中的数据进行标准化处理（均值为 0，方差为 1），即对样本集中元素x_{ik}作变换：

$$x_{ik}=(x_{ik}-\mu_k)/\sigma_k \qquad i=1,2,\ldots,m;\quad k=1,2,\ldots,p \tag{5-7}$$

$$\mu=\frac{1}{m}\sum_{i=1}^{m}x_{ik} \tag{5-8}$$

$$\sigma_k^2=\frac{1}{m-1}\sum_{i=1}^{m}(x_{ik}-\mu_k)^2 \tag{5-9}$$

Step2 为消除量纲对评价结果的影响，得到标准化后的矩阵$\mathbf{M}_b$。计算样本矩阵$\mathbf{M}_b$的相关系数矩阵 R 。

$$R(i,j)=\frac{\mathrm{cov}(i,j)}{\sqrt{\mathrm{cov}(i,i)\,\mathrm{cov}(j,j)}} \tag{5-10}$$

Step3 对于相关系数矩阵 R，采用雅可比方法求特征方程$|\mathrm{R}-\lambda I|=0$的p个非负特征值$\lambda_1>\lambda_2>\cdots>\lambda_p\geq 0$，$\lambda_i$的特征向量为$v_i=(v_{i1},v_{i2},\ldots,v_{ip})$，$i=1,2,\ldots,p$并且满足

$$v_i v_j=\sum_{k=1}^{p}v_{ik}v_{jk}=\begin{cases}1, i=j\\0, i\neq j\end{cases} \tag{5-11}$$

Step4 选择w个主成分分量，使得前面w个主成分的方差和占全部总方差的比例$\eta=(\sum_{i=1}^{w}\lambda_i)/(\sum_{i=1}^{p}\lambda_i)$并使所选的这$w$个主成分尽可能多地保留原来$p$个基因的信息，得到的主成分矩阵记为$\mathbf{M}_{\omega}$。

采用上述数据处理措施后选取的特征基因子集中含15个基因：X53799、M29273、U21914、L00352、D14520、X90858、R80427、X75208、D29808、M59807、D13627、M22760、R56070、Y00062、R50158。

5.5 基于神经网络模型的致癌基因分类方法

5.5.1 本文选用的几种神经网络模型

（1）BP 神经网络模型（BPNN）

BPNN 由输入层、输出层以及一个或多个隐含层组成。本文采用单隐含层的三层神经网络拓扑结构，如图 5-4 所示。输入层神经元数目和样本数相同，隐含层神经元个数一般通过实验或根据经验值选取，输出层神经元个数为 1。

BP 算法的学习训练过程由正向传播和反向传播两阶段组成。在正向传播过程中，样本数据从输入层经过隐含层传递函数的处理传向输出层。如果输出层得不到期望的输出，则转入反向传播过程，将误差信号沿原来正向传播的通路返回，利用均方误差和梯度下降法来实现对网络连接权的修正，以调整网络的实际输出与指导学习信号之间的均方误差值。此过程反复进行，直至满足指定误差要求或达到最大训练次数终止。

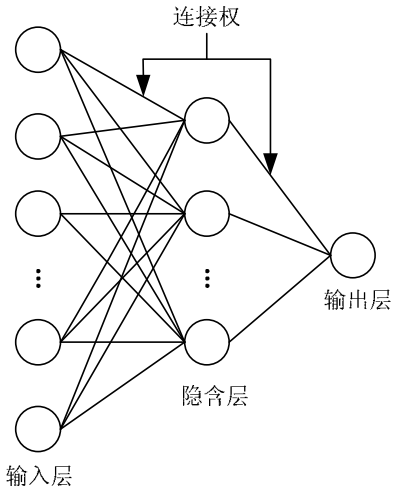


图 5-4 BP 神经网络的拓扑结构

设网络输入向量为 $X=(x_1,x_2,\cdots,x_n)$，实际输出为 $Y=(y_1,y_2,\cdots,y_n)$，期望的输出为 $TY=(ty_1,ty_2,\cdots,ty_n)$。$TY$ 中数据分为两类，0 表示正常，1 表示异常。给定隐含层或输出层的神经元，其输入为 $I_j=\sum_i w_{ij}O_i+\theta_j$，其中 w_{ij} 是由上一层的神经元 i 到神经元 j 的连接权；传递函数 $O_i=\frac{1}{1+e^{-I_j}}$ 是神经元 i 的输出；θ_j 是神经元 j 的偏置。对于训练集中的第 k 个样本，其误差函数为 $E=\frac{1}{2n}\sum_{j=1}^{n}(ty_{kj}-y_{kj})^2$，通常利用梯度下降法求误差函数的极小值，即 $\Delta w_{ij}=-\eta\frac{\partial E}{\partial w_{ij}}$，其中，$\eta$ 为学习速率，其值大于零。

（2）径向基函数神经网络模型（RBFNN）

RBF 网络的学习过程分为三个阶段。第一阶段：根据所有输入样本决定高斯基函数的中心值和平滑因子；第二阶段：利用最小二乘原则，求出输出层的权值；第三阶段：根据指导学习信号校正网络参数，以进一步提高网络的精度。

RBF 网络的拓扑结构由输入层、径向基层和输出层组成，如图 5-5 所示。

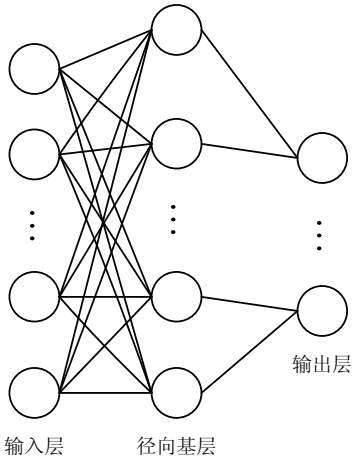


图 5-5 径向基神经网络的拓扑结构

输入层接收来自训练样本的值，其神经元数目和样本数相同，传递函数是线性的。

径向基层神经元采用高斯函数作传递函数(也称基函数)，第 i 个径向基层神经元的输出为 $u_i = \exp[-\frac{(X - X_i)^T (X - X_i)}{2\sigma^2}]$，输出范围在 0 和 1 之间，其中 σ 是平滑因子，其取值确定了以样本点 X_i 为中心的高斯函数的宽度，一般 σ 的选取要足够大，以保证径向基神经元的响应在输入空间能够交迭。高斯函数是一种中心径向对称衰减的非负非线性函数，表示形式简单且解析性好，便于进行理论分析。高斯函数对输入信号将在局部产生响应，当输入样本越靠近基函数的中央范围时，径向基层节点输出值越大，因而 PNN 具有局部逼近能力，学习速度更快。

输出层的传递函数为径向基层神经元输出的线性组合。

（3）概率函数神经网络模型（PNN）

概率神经网络（ProbabilsticNeuralNetwork, PNN）是一种径向基神经网络模型，采用 Parzen 提出的由高斯函数为核函数形成联合概率密度分布的估计方法和贝叶斯优化规则。它基于统计原理，计算能逼近贝叶斯最优判决式的非线性决策边界，在分类功能上与最优贝叶斯分类器等价。其拓扑结构由输入层、模式层、累加层和输出层(决策层)组成，如图 5-6 所示。

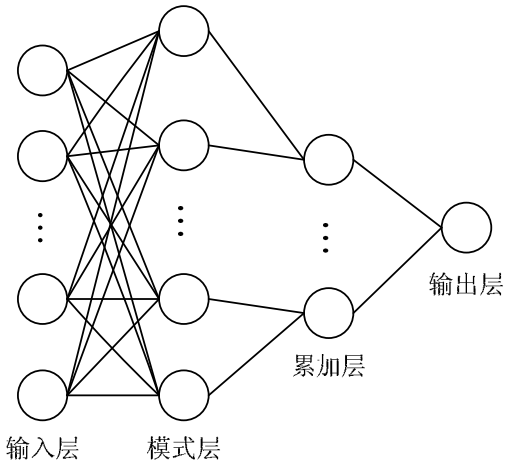


图 5-6 概率神经网络的拓扑结构

输入层接收来自训练样本的值，其传递函数是线性的，直接将输入样本传递给模式层。

模式层神经元将输入向量的各个分量进行加权求和后采用一个非线性算子运算 $g(z_i) = \exp((z_i - 1)/\sigma^2)$，其中，$\sigma$ 是平滑因子，表示各类样本之间的影响程度。

累加层接收来自模式层的运算结果，其节点只与相应类别的样本节点相连，计算概率密度函数，从而得到输入样本属于某类的最大可能性。

输出层为模式后验概率估计。神经元数目等于训练样本数据的种类数，每个神经元分别对应于一个数据类别。该层神经元是一种竞争神经元，它接收从累加层输出的各类概率密度函数，寻找概率密度函数最大的神经元，所对应的类即为待识别的样本模式类别。

（4）广义回归神经网络模型（GRNN）

广义回归神经网络（General Regression Neural Network, GRNN）是在概率神经网络基础之上提出的另一种径向基神经网络模型，建立在非参数核回归的数理统计基础上，以样本数据为后验条件，执行 Parzen 非参数估计，网络最后收敛于样本量积聚最多的优化回归面。GRNN 的拓扑结构由输入层、模式层、累加层和输出层组成。

输入层接收来自训练样本的值，传递函数是线性的，直接将输入样本传递给模式层。

模式层又称隐回归层，神经元的个数等于训练样本数。模式层中采用高斯函数作传递函数。训练过程中通过改变平滑因子σ的值，从而调整模式层中各神经元的传递函数，以获得最佳的回归估计结果。σ取值越大则基函数越平滑，在训练样本数目一定的情况下，平滑因子值的变化影响概率密度函数值的变化，进而影响最终预测结果。

累加层接收来自模式层的运算结果，神经元数目为样本向量的维数 p 加 1，包括两种类型神经元，其中 p 个神经元计算所有模式层神经元输出的加权和，称为分子单元；另一个神经元计算所有模式层神经元的输出之和，称为分母单元。

输出层将累加层分子单元和分母单元的输出相除，算得样本的估计值。

5.5.2 神经网络致癌基因分类模型的建立

神经网络模型的输入层节点数m设置为训练样本x的基因个数；隐层节点数n为输入层节点数的 2 倍；由于输出目标为区分肿瘤样本和正常样本，故输出层节点数k设为 1。输出目标函数$T(X)$的值表示训练样本类别，其中“0”表示正常样本，“1”表示肿瘤样本。输入向量X的第i个分量x_i对应训练集中样本的第i个基因。

基于给定数据本文建立了四种神经网络分类器模型，分别为下面以 BP 神经网络为例，给出网络的训练过程及步骤。

（1）初始化。给每个连接权值w_{ij}、v_{it}、阀值θ_j与γ_j赋予区间（-1，1）内的随机值。

（2）用输入基因样本$P_k=(a_1^k,a_2^k,\cdots,a_n^k)$、连接权$w_{ij}$和阀值$\theta_j$计算中间层各单元的输入$s_j$，然后用$s_j$通过传递函数计算中间层各单元的输出$b_j$。

$$b_j=f(s_j)\quad j=1,2,\cdots,p \tag{5-12}$$

（3）利用中间层的输出b_j、连接权v_{jt}和阀值γ_j计算输出层各单元的输出L_t，然后利用通过传递函数计算输出层各单元的响应C_t。

$$L_t=\sum_{j=1}^{p}v_{jt}b_j-\gamma_j\quad t=1,2,\cdots,q \tag{5-13}$$

$$C_j=f(L_j)\quad t=1,2,\cdots,q \tag{5-14}$$

（4）利用网络目标向量$T_k=(y_1^k,y_2^k,\cdots,y_p^k)$，网络的实际输出$C_t$，计算输出层的各单元一般化误差$d_t^k$。

$$d_t^k=(y_t^k-C_t).C_t(1-C_t)\quad t=1,2,\cdots,q \tag{5-15}$$

（5）利用连接权v_{jt}、输入层的一般化误差d_t和中间层的输出b_j计算中间层

各单元的一般化误差e_t^k。

$$e_t^k = [\sum_{t=1}^{q} d_t . v_{jt}] b_j (1-b_j) \tag{5-16}$$

（6）利用输出层各单元的一般化误差d_t^k与中间层各单元的输出b_j来修正连接权v_{jt}和阀值γ_j。

$$v_{jt}(N+1) = v_{jt}(N) + \alpha . d_t^k . b_j \tag{5-17}$$

$$\gamma_t(N+1) = \gamma_t(N) + \alpha . d_t^k \tag{5-18}$$

（7）利用中间层各单元的一般化误差，输出层各单元的输入来修正连接权和阀值。

$$w_{ij}(N+1) = w_{ij}(N) + \beta e_j^k a_i^k \tag{5-19}$$

$$\theta_j(N+1) = \theta_j(N) + \beta e_j^k \tag{5-20}$$

（8）随机选取下一个学习样本向量提供给网络，返回到步骤（3），直到m个训练样本样本完毕。

（9）重新从m个学习样本中随机选取一组输入和目标样本，返回步骤（3），直到网络全局误差ε小于预先设定的一个极小值，即网络收敛。如果学习次数大于预先设定的值，网络就无法收敛。

（10）训练学习结束。

通用的神经网络训练的基本流程如下图所示：

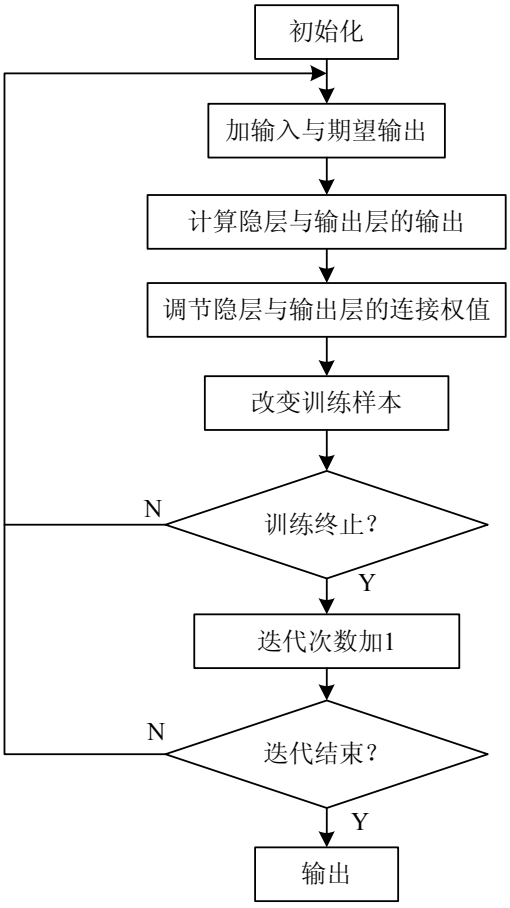


图 5-7 通用神经网络训练的基本流程

5.5.3 分类实验过程与结果分析

1、实验数据分类

在提取特征信息基因后，本文将正常样本和肿瘤样本按接近 2:1 的比例随机地分配到训练集和测试集中。如图 5-8 所示，训练集有 40 个样本，测试集有 22 个样本。

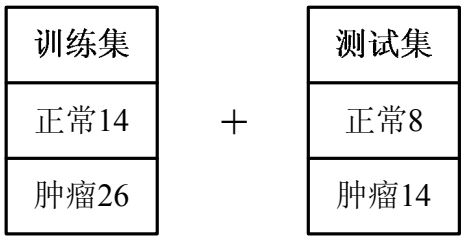


图 5-8 基因表达谱实验数据集分类

2、特征基因子集筛选

采用主成分分析法得到特征基因集合中含有的 11 个特征基因，可以构成 2^{15}=32768 个不同的基因组合，每个基因组合被称为一个特征子集。本文采用遍历搜索算法对特征子集构成的空间进行搜索，通过二进制编码对 32768 个基因组合进行标记，通过对正确辨识率的排序，筛选出具有最佳分类能力和最少基因个数的特征基因子集，以此作为分类器的基因“标签”（信息基因）。通过上述方法对本文进行试验，得到其中 9 种基因的结合具有最佳分类能力。

表5-1列出了采用上述数据处理措施后选取的特征基因子集中9个基因的基因标签和简单功能描述。

表5-1　最佳基因组合及功能描述

序号	基因	基因功能描述
1	X53799	Human mRNA for macrophage inflammatory protein-2alpha (MIP2alpha).
2	M29273	MYELIN-ASSOCIATED GLYCOPROTEIN PRECURSOR (HUMAN).
3	U21914	Human duplicate spinal muscular atrophy mRNA, clone 5G7, partial cds.
4	L00352	Human low density lipoprotein receptor gene, exon 18.
5	X90858	H.sapiens mRNA for uridine phosphorylase.
6	R80427	C4-DICARBOXYLATE TRANSPORT SENSOR PROTEIN DCTB (Rhizobium leguminosarum).
7	X75208	H.sapiens HEK2 mRNA for protein tyrosine kinase receptor.
8	D29808	Human mRNA for T-cell acute lymphoblastic leukemia associated antigen 1 (TALLA-1), complete cds.
9	M59807	NATURAL KILLER CELLS PROTEIN 4 PRECURSOR (HUMAN); contains element MSR1 repetitive element.

3、训练与测试实验

由于实验样本少，为了获得对候选特征子集分类能力的较为可靠的估计，采取留一交叉校验和独立测试实验在训练集和测试集上分别检验分类准确率。

(1)训练集中，采用“留一法”[2](Leave-One-Out Cross Validation, LOOCV) 校验样本类型：每次保留 1 个样本为测试样本，其余 39 个样本用作神经网络的训

练样本。重复该过程，直到所有 40 个样本都被用作过测试样本为止，从而统计得到留一交叉检验的准确率。

(2) 对于测试集，用训练集上的所有 40 个样本训练神经网络，然后用训练好的神经网络识别测试集中 22 个样本的类型，从而统计得到“独立测试实验”[5](Independent Test, IT) 的分类准确率。

4、实验结果分析

表 5-2　四种神经网络的分类准确性

神经网络分类器	BPNN	RBFNN	PNN	GRNN
留一交叉校验准确率	97.5%	97.5%	97.5%	97.5%
独立测试实验准确率	61.36%	63.64%	77.27%	63.64%

由表 5-2 可知，对四种网络分类器，留一法检验正确率都比较高，而对于训练集采用独立测试实验时，概率神经网络（PNN）对分类的效果最好。所以，在下文的分析中，均采用 PNN 分类器进行分类识别。

六、问题三的建模与求解

对于第三个问题，由于基因表达谱中不可避免地含有噪声，有的噪声强度甚至较大，对含有噪声的基因表达谱提取信息时会产生偏差。为保证特征提取的有效性和分类识别的准确性，本文采用不同小波基函数对基因数据进行小波变换去噪，然后提取去噪后数据的特征信息基因，将其输入到概率神经网络分类器，得到了更为准确的分类结果。

6.1 基于小波变换的去噪方法

一个含有噪声的一维信号的模型为：

$$s(i)=f(i)+\sigma\cdot e(i) \qquad i=0,1,\cdots,n-1 \tag{6-1}$$

其中，$f(i)$ 为真实信号，$e(i)$ 为噪声，$s(i)$ 为含噪声的信号。信号消噪的目的就是要将信号 $s(i)$ 中的噪声 $e(i)$ 对真实信号 $f(i)$ 的影响减小到最小的程度。

在信号处理中，有用信号通常表现为低频信号或是一些比较平稳的信号，而噪声信号则通常表现为高频信号。在本问题中，信息基因数据为有用信号，而信息基因采集过程中产生的随机误差等为噪声，表现为高频分量。

基于小波的去噪方法就是寻找到从含噪信号空间到小波函数空间的最佳映射，即找到 $f=(f_1,...,f_n)'$ 的估计值 $\tilde{f}$，使得其均方误差（mean-square error）$R(\tilde{f},f)$ 最小：

$$R(\tilde{f},f)=n^{-1}\sum_{i=1}^{n}E(\tilde{f}_i-f_i)^2 \tag{6-2}$$

多分辨率理论认为，在尺度 $2^J\le 2^j\le 2^L$ 上，$y\in L^2(\mathbf{R})$ 可分解成小波系数：

$$[\{d_j\}_{J\le j\le L},a_L]\ ,\ k\in z \tag{6-3}$$

小波去噪算法首先把含噪信号小波分解，并设定一阈值，低于该阈值的小波系数被认为是噪声产生的，从而被清零，留下的有效系数经小波逆变换后得到被

测信号的估计值，大体流程可用下式表示

$$y \xrightarrow{DWT} \{a_L, d_j\} \xrightarrow{Threshold} \{\tilde{a}_L, \tilde{d}_j\} \xrightarrow{IDWT} \tilde{f} \qquad (6\text{-}4)$$

小波消噪可按以下 3 个步骤进行：

(1) 首先对信号进行小波分解。选择小波并确定小波分解的层次 N，然后对信号 S 进行 N 层小波分解。如进行三层分解（噪声通常含在 cd1，cd2，cd3 中），分解过程如图 6-1 所示。

(2)小波分解高频系数的阈值量化。对于第 1 层到第 N 层的每一层高频系数，选择一个阈值，并且对高频系数用阈值收缩处理。

(3) 对信号进行重构。根据小波分解的第 N 层的低频系数和阈值量化处理后的第 1 层到第 N 层的高频系数，进行小波重构。重构过程如图 6-2 所示。

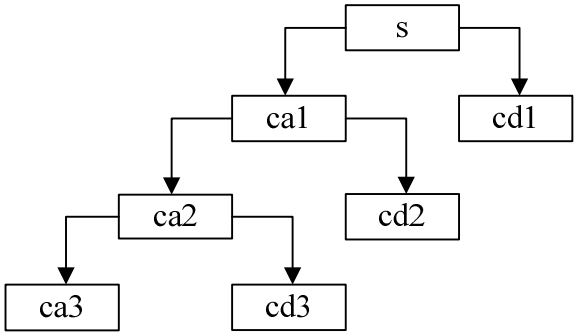


图 6-1 信号的小波分解树

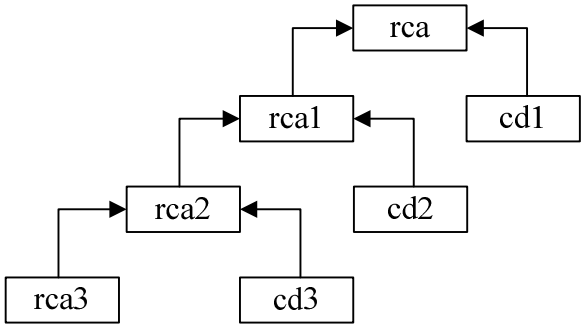


图 6-2 信号的小波重构树

6.2 基于小波变换消除基因数据噪声误差

本文使用 Matlab 小波工具箱（Wavelet）进行阈值消噪处理。首先，在 Matlab 中利用 *ddencmp* 函数产生信号默认阈值，然后利用 wden 函数进行消噪处理。

语法结构：

[*XD, CXD, LXD*]=*wden*(*X, tptr, sorh, scal, n, 'wavename'*)

[*XD, CXD, LXD*]=*wden*(*C, L, tptr, sorh, scal, n, 'wavename'*)

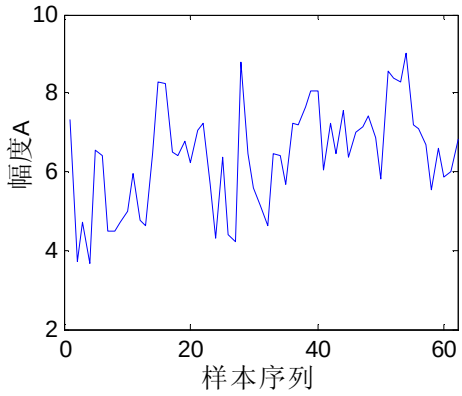
说明：

[*XD, CXD, LXD*]=*wden*(*X, tptr, sorh, scal, n, 'wavename'*)使用小波系数阈值，返回输入信号 *X* 除噪后的信号 *XD*，输出参数[*CXD, LXD*]表示 *XD* 的小波分解结构。

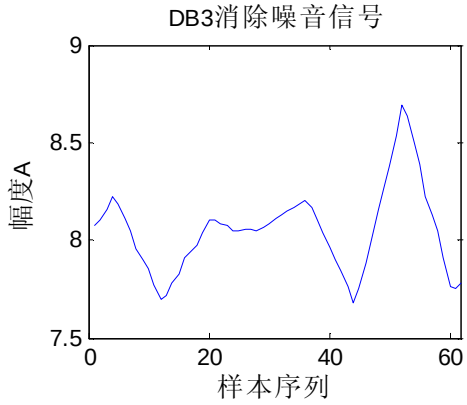
输入参数中，*tptr* 同 *thselect* 函数；sorh 为‘*s*’或‘*h*’表示软硬阈值；*n* 表示在 *n* 层上的小波分解；*wavename* 指定小波名称；*scal* 定义阈值调整比例。

[*XD, CXD, LXD*]=*wden*(*C, L, tptr, sorh, scal, n, 'wavename'*)使用同上面一样选项，返回直接对小波分解结构[*C, L*]除噪后的信号 *XD*，在 *n* 层上，使用‘*wavename*’指定的正交小波。

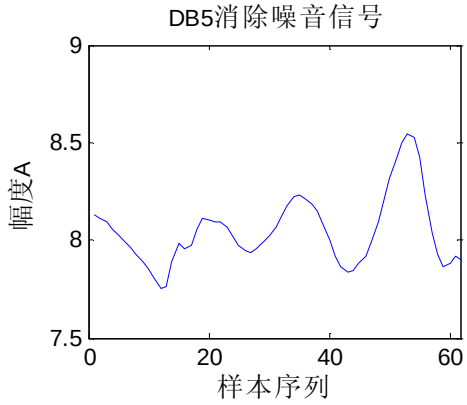
为最大限度地消除噪声，同时保证原始数据信息损失最小。本文分别选择用 db3、db5、sym8、haar 四种不同小波基进行去噪，去噪前后对比如图 6-3 所示。



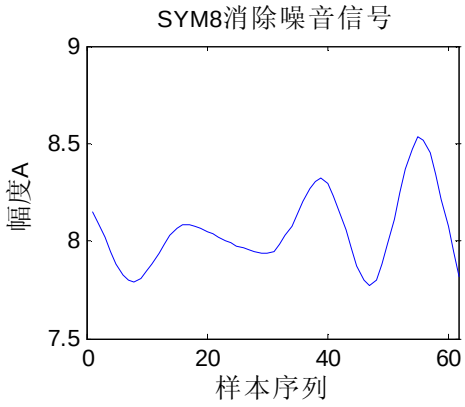
(a) 原始数据



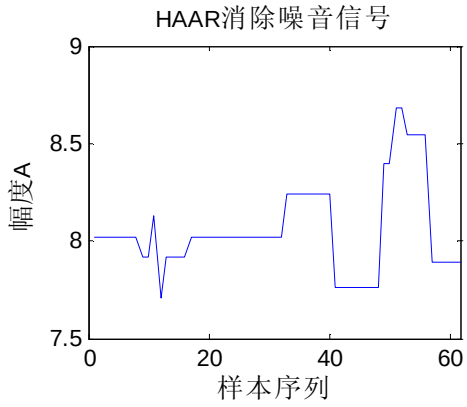
(b) db3 小波基去噪



(c) db5 小波基去噪



(d) sym8 小波基去噪



(e) haar 小波基去噪

图 6-3 不同小波基去噪前后对比图

由上图可知：选择各种小波基去噪都达到了预期效果，同时很好地保留了基因数据的原始信息，所以本文分别对将这四种小波基函数去噪后的数据，利用前面提出的特征提取方法，再将获得特征信息基因集合，输入神经网络进行训练和测试，得到的分类实验结果如下表所示。

表 6-1　四种小波基去噪后选取的特征基因及分类准确率

选用的小波基函数	db3 小波	db5 小波	sym8 小波	Haar 小波
提取的特征信息基因	R08183 X73478 H78386 M26252 — —	T65758 H72965 M59371 X72018 — —	T62496 X78817 T84049 R37464 — —	H29293 U37012 R37482 M31303 R46069 X72018
留一交叉校验准确率	97.5%	97.5%	97.5%	97.5%
独立测试实验准确率	77.27%	72.73%	81.82%	100%

由上表可知：利用 db5 小波基去噪处理后的基因数据进行独立测试实验的准确率低于去噪前，说明基于该小波基函数的去噪使原始数据损失了有效信息。而利用 sym8 和 Haar 小波基函数去噪后的基因数据均得到了较高的分类准确率。经 Haar 小波基去噪后提取出 6 个特征信息基因，利用 PNN 进行独立测试实验的分类准确率达到了 100%。

七、问题四的建模与求解

在肿瘤研究领域通常会已知若干个信息基因与某种癌症的关系密切，所以本文采用信息融合的方法，利用加权评分法(Weighted Grade, WG)建立了融入生理学确定信息基因的多信源信息融合模型。通过对 PNN-WG 模型求解，验证了提出融合模型可有效提高原有分类器的准确性，达到了多源信息融合的目的。

7.1 PNN-WG 多信源信息融合模型的建立

本文设计了包含 n 个概率神经网络和加权评分机制的多信源信息融合模型（PNN-WG），模型框架如图 7-1 所示。

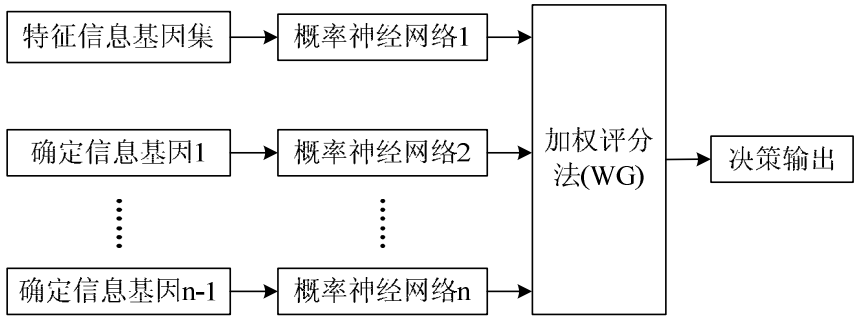


图 7-1 PNN-WG 多信源信息融合模型

为验证本文方法的有效性，选用未经小波去噪的数据所提取的特征基因子集（概率神经网络的分类准确率为 77.27%）输入概率神经网络进行分类识别。

设基于特征信息基因进行神经网络分类的输出向量为 $T_1=(s_1^1,s_2^1,\cdots,s_p^1)$（即概率神经网络 1 的输出向量），该网络分类准确率 α_1，可看作网络结果的可信度。

利用其他确定信息基因进行神经网络分类的输出向量为

$$T_2=(s_1^2,s_2^2,\cdots,s_p^2),\cdots,T_n=(s_1^n,s_2^n,\cdots,s_p^n) \tag{7-1}$$

其中 n 表示概率神经网络个数，确定信息基因为 $n-1$ 个，确定信息基因的可信度分别设为 $\alpha_2,\alpha_3,\cdots,\alpha_n$。

对 n 个神经网络输出结果赋予权值 $\lambda_i,i=1,2,\cdots,n$ 为

$$\lambda_i=\frac{\alpha_i}{\sum_{i=1}^{n}\alpha_i} \tag{7-2}$$

$S=(S_1,S_2,\cdots,S_p)$ 为加权评分向量，其中

$$S_j=\sum_{i=1}^{n}\lambda_i s_j^i \tag{7-3}$$

由于确定信息基因的分类结果可作为重要参考，具有纠正概率神经网络 1 的错误分类的作用，故分别取 $s_j^k=0$或1，$s_j^k=0$ 表示判断结果为正常样本，$s_j^k=1$ 表示判断结果为肿瘤样本。设定投票评分法的阈值判据模型为：

$$\begin{cases} if \quad S_j\le\lambda_1, \quad then \quad s_j^k=0 \\ if \quad S_j\ge\lambda_2+\lambda_3+\cdots+\lambda_n, \quad then \quad s_j^k=1 \end{cases} \tag{7-4}$$

7.2 模型的求解

下面结合特征提取的信息基因，引入题中给定的确定信息基因——5 号染色体长臂 APC 基因与 ras 相关基因为确定基因样本数据，利用多信源融合网络进行分类识别，确定基因样本数据见表 7-1 所示。

表 7-1　确定基因样本编号及功能描述

基因类型	基因编号	基因功能描述
APC基因	L35545	Homo sapiens endothelial cell protein C/APC receptor (EPCR) mRNA, complete cds.
ras相关基因	H04311	RAS GTPASE-ACTIVATING-LIKE PROTEIN IQGAP1 (Homo sapiens)
	H42477	RAS-RELATED C3 BOTULINUM TOXIN SUBSTRATE 1 (Homo sapiens)
	M28214	RAS-RELATED PROTEIN RAB-3B (HUMAN);.
	R22779	RAS-RELATED PROTEIN RAB-11 (HUMAN);.
	R53941	RAS-RELATED C3 BOTULINUM TOXIN SUBSTRATE 1 (Homo sapiens)
	T70197	RAS-RELATED C3 BOTULINUM TOXIN SUBSTRATE 1 (Homo sapiens)
	T71207	RAS-RELATED C3 BOTULINUM TOXIN SUBSTRATE 2 (Homo sapiens)
	X54871	H.sapiens mRNA for ras-related protein Rab5b.
	Z29677	H.sapiens mRNA for ras-related GTP-binding protein.

如前所述，实验中同样选取 40 个训练样本，22 个测试样本。加权评分过程如下：

Step1 将特征基因信息子集输入概率神经网络分类器，得到测试样本输出向量：$T_1=(s_1^1,s_2^1,\cdots,s_{22}^1)$

Step2 分别将 APC 相关基因和 ras 相关基因输入概率神经网络分类器，分别得到测试样本的输出向量：$T_2=(s_1^2,s_2^2,\cdots,s_{22}^2)$ 和 $T_3=(s_1^3,s_2^3,\cdots,s_{22}^3)$

Step3 得到加权得分向量

$$S=\sum_{i=1}^{3}\lambda_i T_i \tag{7-5}$$

其中，三个向量的可信度分别为 $\alpha_1=\frac{17}{22},\alpha_2=0.9,\alpha_3=0.5$，可计算得到三个神经网络的权重分别为 $\lambda_1=0.355648,\alpha_2=0.414226,\alpha_3=0.230126$

Step4 由加权评分法的阈值判据

$$\begin{cases} if \quad S_j\le\lambda_1, \quad then \quad s_j^k=0 \\ if \quad S_j\ge\lambda_2+\lambda_3, \quad then \quad s_j^k=1 \end{cases} \tag{7-6}$$

得到最终分类结果向量

$$S'=(1,0,0,0,0,0,0,1,1,1,1,0,1,1,1,1,1,1,1,1,1)$$

由结果可知，前 8 个正常个体测试样本分类正确 6 个，后 14 个肿瘤测试样本分类正确 13 个，正确率为 $\frac{6+13}{22}\times100\%=86.36\%$。与第二问分类效果相比，准确率提高了 9.09%。

通过加权评分法对不同特征信息子集的神经网络输出进行信息融合，可以综合不同类型信息，从而克服由单一特征信息提取和识别带来的误判。

八、模型和算法评价与改进方向

8.1 模型和算法的优点

（1）本文的主要结果验证了提出的特征基因信息提取方法的准确性和有效性，且设计的神经网络分类器可通过调用 Matlab 神经网络工具箱，具有编程简单、可移植性好、求解速度快的优点；

（2）由第三问的求解可知，利用小波变换去噪方法消除数据的随机误差，使数据信息更加准确，为特征提取提供了有利条件。利用去噪后的数据得到的分类识别准确率高的优点；

（3）利用加权评分法建立的融入生理学的确定信息基因的多信源信息融合模型可有效提高原有分类器的准确性，达到了多源信息融合的目的，同时模型和算法具有简洁高效、易于编程实现等优点。

8.2 进一步研究的方向

（1）如果把基因表达谱看成一种信号，那么我们就可以采用信号处理的方法来处理肿瘤基因表达谱样本。

（2）有兴趣深入的研究思路：PNN-DS 多信源信息融合模型

由于时间有限，下述思路未能完全实现，在此述及以供后续工作参考之用。

为网络使模型获得更好的分类结果，可设计了包含 n 个概率神经网络和 D-S 证据理论的多信源信息融合模型（PNN-DS），模型框架如图 8-1 所示。

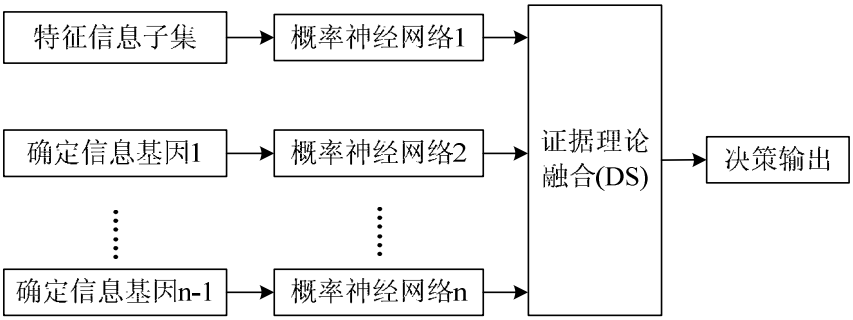


图 8-1 PNN-DS 多信源信息融合模型

PNN-DS 多信源信息融合模型的两个重要步骤表述如下：

1）概率神经网络单一特征基因信息分类识别

各概率神经网络分别对不同的特征基因信息进行分类识别，分别处理不同信息来源的数据样本（包括提取的特征信息基因和临床生理学信息等），由此通过区分不同来源的信息基因子集而形成 n 个概率神经网络。

2）DS 证据理论决策融合

将每个神经网络的输出值经过转换后作为证据理论在不同特征信息基因下的独立证据，即成为各类信息的基本概率分配。每个网络的诊断能力和可靠程度是不同的，因此每个网络存在一个可靠性系数 α，表示对专家判定结果的信任程度。其具体公式如下[9]：

设第 i 个网络的第 j 个输出值为 $o_i(j)$，则有

$$m_i(j)=\frac{o_i(j)}{\sum_{j=1}^{q} o_i(j)}\times\alpha, m_i(\theta)=1-\alpha_i \tag{8-1}$$

式中，$m_i(j)$ 代表第 i 个证据对状态 j 的概率分配，$m_i(\theta)$ 为不确定性 θ 的基本概率分配函数，q 代表神经网络的个数。然后根据证据理论合并规则公式得到合并后的各状态的基本概率分配。

最后通过如下决策规则得到最终决策输出：

设 $\exists A_1, A_2 \subset U$，满足

$$m(A_1)=\max\{m(A_i), A_i \subset U\} \tag{8-2}$$

$$m(A_2)=\max\{m(A_i), A_i \subset U 且 A_i \neq A_1\} \tag{8-3}$$

若有

$$\begin{cases} m(A_1)-m(A_2)>\varepsilon_1 \\ m(\theta)<\varepsilon_2 \\ m(A_1)>m(\theta) \\ m(A_1)>\varepsilon_3 \end{cases} \tag{8-4}$$

则 A_1 即为判定是否为致癌基因，其中 ε_1、ε_2、ε_3 为预先设定的阈值。

从方法的机理上看，PNN-DS 多信源信息融合模型具有以下优点：

1）可降低每个神经网络处理数据样本的维数，充分利用概率神经网络收敛速度快和计算机并行处理能力，可以加快神经网络训练速度和诊断决策时间，进而解决高维输入神经网络训练收敛速度慢和诊断时间长等问题。分类各信息基因子集的神经网络工作相互独立，新特征基因信息增加方便，该分类识别系统具有可扩展性强的特点；

2）通过 DS 证据理论对不同特征信息子集的神经网络输出进行信息融合，可以综合不同类型信息，从而克服由单一特征信息提取和识别带来的误判。

九、参考文献

[1] 李颖新，刘全金，阮晓钢. 急性白血病的基因表达谱分析与亚型分类特征的鉴别[J]. 中国生物医学工程学报，2005，24(2) :240-244.

[2] 李颖新，阮晓钢. 基于基因表达谱的肿瘤亚型识别与分类特征基因选取研究[J]. 电子学报，2005，33(4) :651-655.

[3] 王树林. 生物子序列频数分布与肿瘤亚型分类模型研究[D]. 长沙:国防科技大学，2007.

[4] 王树林，王戟，陈火旺，张波云. 基于主成份分析的肿瘤分类检测算法研究[J]. 计算机工程与科学，2007，29(9) :84-90.

[5] 刘全金，李颖新，阮晓钢. 基于BP 网络灵敏度分析的肿瘤亚型分类特征基因选取[J]. 中国生物医学工程学报，2008，27(5) :710-715.

[6] 刘全金，李颖新，朱云华，阮晓钢. 基于BP 神经网络的肿瘤特征基因选取[J]. 计算机工程与应用，2005，34 :184-186.

[7] 黄德生. 基因识别和微阵列数据识别算法研究[D]. 北京:中国医科大学，2009.

[8] 崔光照，曹祥红，张华. 基于小波变换的基因表达数据去噪聚类分析[J]. 信号处理，2005，21(4A) :463-466.

[9] 张海平，何正友，张钧. 基于量子神经网络和证据融合的小电流接地选线方法[J]. 电工技术学报，2009，24(12) :171-178.

附录

附录清单

附录一：两两冗余法和主成分分析法提取特征基因；

附录二：加权得分程序

附录一：两两冗余法和主成分分析法提取特征基因

```
clear all;
clc;
fid1=fopen('pre_pro.txt','r');
data1=fscanf(fid1,'%g',[62,1909]);
data=data1';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%数据归一化
sum_total=0;
std_total=0;
for i=1:1909
    for j=1:62
        sum_total=sum_total+data(i,j);
    end
end
ave=sum_total/(62*1909);
for i=1:1909
    for j=1:62
        std_total=std_total+(data(i,j)-ave)^2;
    end
end
std=sqrt(std_total/(62*1909-1));
for i=1:1909
    for j=1:62
        data_guiyihua(i,j)=(data(i,j)-ave)/std;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 计算正常与癌变的均值与标准差
for i=1:1909
    normal_ave(i)=sum(data_guiyihua(i,1:12))/12;
    for j=1:12
        normal_biaozhuncha(i)=sqrt(sum(data_guiyihua(i,j)-normal_ave(i))^2/(12-1));
    end
end
```

```
for i=1:1909
        cancer_ave(i)=sum(data_guiyihua(i,13:62))/40;
        for j=13:62
                cancer_biaozhuncha(i)=sqrt(sum(data_guiyihua(i,j)-cancer_ave(i))^2/(40-1));
        end
end

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 方法一：d 为基因信噪比
% for i=1:1909
%
d(i)=abs((normal_ave(i)-cancer_ave(i))/(normal_biaozhuncha(i)+cancer_biaozhuncha(i)));
% end
% dd=d';
%
% [A,ind]=sort(d,'descend');
% d_juli=zeros(1909,2);
%
% for i=1:1909
%         d_juli(i,1)=ind(i);
%         d_juli(i,2)=A(i);
% end
%
% for i=1:300
%                 choose_300(i,1)=d_juli(i,1);
%                 choose_300(i,2)=d_juli(i,2);
% end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 方法二：b 为 Bhattacharyya 距离
for i=1:1909

b(i)=1/4*(normal_ave(i)-cancer_ave(i))^2/(normal_biaozhuncha(i)^2+cancer_biaozhuncha(i)^2)...

+1/2*log((normal_biaozhuncha(i)^2+cancer_biaozhuncha(i)^2)/(2*normal_biaozhuncha(i)*cance
r_biaozhuncha(i)));
end
bb=b';

[B,ind2]=sort(b,'descend');
b_juli=zeros(1909,2);
```

```
for i=1:1909
        b_juli(i,1)=ind2(i);
        b_juli(i,2)=B(i);
end



nn=200;
for i=1:nn
                choose_300(i,1)=b_juli(i,1);
                choose_300(i,2)=b_juli(i,2);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

index_huanyuan=zeros(1909,1);
%提取的归一化数据
for i=1:nn
        temp=choose_300(i,1);
        index_huanyuan(i)=temp;
        for j=1:62
                data_tiqu(i,j)=data_guiyihua(temp,j);
        end
end
%提取的未归一化的数据
for i=1:nn
        temp=choose_300(i,1);

index_huanyuan(i)=temp;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%对应抽取出的 300 个基因的标号
        for j=1:62
                data_tiqu1(i,j)=data(temp,j);
        end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%以下为冗余算法
for i=1:nn
        ave_yangben(i)=sum(data_tiqu(i,:))/62;
end
coef1=zeros(nn,nn);
coef2=zeros(nn,1);
```

```
for i=1:nn
    for j=1:nn
        for k=1:62
        coef1(i,j)=coef1(i,j)+(data_tiqu(i,k)-ave_yangben(i))*(data_tiqu(j,k)-ave_yangben(j));
        end
    end
end
for i=1:nn
    for j=1:62

        coef2(i)=coef2(i)+(data_tiqu(i,j)-ave_yangben(i))^2;

    end
end
for i=1:nn
    for j=1:nn

        coef(i,j)=coef1(i,j)/sqrt(coef2(i)*coef2(j));
    end
end

newB=choose_300(:,2);
newindex11=choose_300(:,1);

for i=1:nn
    for j=i+1:nn

        if (coef(i,j)>0.5)
            newB(j)=0;
            newindex11(j)=0;


        else
            break;
        end

    end
end
j=1;
for i=1:nn

    if newB(i)~=0;
        newBB(j)=newB(i);
        newindex22(j)=newindex11(i);
```

```
        j=j+1;
    end
end
newBB=newBB';
newindex22=newindex22';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  主成分分析法
% index_huanyuan=zeros(1909,1);
% for i=1:length(newBB)
%       temp1=newindex22(i);
%       index_huanyuan11(i)=temp1;
%       for j=1:62
%             data_tiqu_hou(i,j)=data_guiyihua(temp1,j);
%       end
% end
for i=1:length(newBB)
    temp_1=newindex22(i);

index_huanyuan_1(i)=temp_1;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%对应抽取出的 300 个基因的标号
    for j=1:62
        data_tiqu1_hou(i,j)=data(temp_1,j);
    end
end
%主成分分析——数据归一化(飞哥提取的 300 个基因)
sum_total_1=0;
std_total_1=0;

for i=1:length(newBB)
    for j=1:62
        sum_total_1=sum_total_1+data_tiqu1(i,j);
    end
end
ave_1=sum_total_1/(62*length(newBB));
for i=1:length(newBB)
    for j=1:62
        std_total_1=std_total_1+(data_tiqu1(i,j)-ave_1)^2;
    end
end
std_1=sqrt(std_total_1/(62*1909-1));
for i=1:length(newBB)
    for j=1:62
```

```
            tiqu_guiyihua(i,j)=(data_tiqu1(i,j)-ave_1)/std_1;
        end
end
%主成分分析——求相关系数矩阵
 tiqu_zhuanzhi=tiqu_guiyihua';
% %C=cov(tiqu_zhuanzhi);
% R=corrcoef(tiqu_zhuanzhi);
% fprintf('相关系数矩阵:\n');
std=CORRCOEF(tiqu_zhuanzhi);       %计算相关系数矩阵
% fprintf('特征向量(vec)及特征值(val)：\n')
[vec,val]=eig(std);       %求特征值(val)及特征向量(vec)
newval=abs(diag(val)) ;
%newval_biaohao_zhi=[index_huanyuan_1,newval];%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%将标号与特征值组成 300*2 的向量
[y,index]=sort(newval) ;          %对特征根进行排序，y 为排序结果，index 为索引
%    fprintf('特征根排序：\n');
 for z=1:length(y)
       newy(z)=y(length(y)+1-z);
 end
%    fprintf('%g\n',newy);
 rate=y/sum(y);
%    fprintf('\n 贡献率：\n');
 newrate=newy/sum(newy);
 sumrate=0;
 newindex=[];
for k=length(y):-1:1
       sumrate=sumrate+rate(k);
       newindex(length(y)+1-k)=index(k);
       if sumrate>0.93 break;
       end
end                          %记下累积贡献率大 85%的特征值的序号放入 newindex 中
fprintf('主成分数：%g\n\n',length(newindex));
```

附录二：加权得分程序

```
clc;
clear all;
```

```
% P 是样本数据, T 是表示样本类别的下表矩阵
fid1=fopen('pre_pro.txt','r');
data1=fscanf(fid1,'%g',[62,1909]);
data=data1';

biaohao=[1696 643 1560 457 1855 1094 1798 67 691];%冗余+主成分 0.5 最优组合


%biaohao=[540];%加权 APC
%biaohao=[114 259 533 635 859 994 1354 1360 1700 1896];%加权 RAS


for i=1:length(biaohao)
    for j=1:1909
        if (j==biaohao(i))
            P1=data(j,1:14);
            P2=data(j,23:48);
            P(i,1:40)=[P1 P2];
            break;
        end
    end
end

 %T1=[1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2];
 T1=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ];
%  将下标矩阵变为单值矢量组作为网络的目标输出
T=ind2vec(T1);

%   设计概率神经网络
sp=0.1; %扩展常数
net=newpnn(P,T,sp);

%   对网络进行仿真，并绘出分类结果
Y=sim(net,P);
Y1=vec2ind(Y);

%   对一组新的数据进行分类
for i=1:length(biaohao)
    for j=1:1909
        if (j==biaohao(i))
            P3=data(j,15:22);
            P4=data(j,49:62);
            PP(i,1:22)=[P3 P4];
            break;
```

```
            end
        end
end

 Y=sim(net,PP);
 Y1=vec2ind(Y);
  correct=0;
 for i=1:22
        if (Y1(i)==1)&(i<=8)
              correct=correct+1;
        elseif (Y1(i)==2)&(i>8)
              correct=correct+1;
        end
 end
 %%%%%%%%%%%%%%%%%%%%%
%    a1=0.35565;
%    a2=0.414226;
%    a3=0.230126;
 for i=1:22
        if (Y1(i)==1)
              X1(i)=0;
        elseif (Y1(i)==2)
              X1(i)=1;
        end
 end

biaohao=[540];%加权 APC
%biaohao=[114 259 533 635 859 994 1354 1360 1700 1896];%加权 RAS


for i=1:length(biaohao)
      for j=1:1909
            if (j==biaohao(i))
                  P1=data(j,1:14);
                  P2=data(j,23:48);
                  P(i,1:40)=[P1 P2];
                  break;
            end
      end
end

 %T1=[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2];
 T1=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ];
%  将下标矩阵变为单值矢量组作为网络的目标输出
```

```
T=ind2vec(T1);

%  设计概率神经网络
sp=0.1; %扩展常数
net=newpnn(P,T,sp);

%  对网络进行仿真，并绘出分类结果
Y=sim(net,P);
Y1=vec2ind(Y);

%  对一组新的数据进行分类
for i=1:length(biaohao)
      for j=1:1909
          if (j==biaohao(i))
              P3=data(j,15:22);
              P4=data(j,49:62);
              PP(i,1:22)=[P3 P4];
              break;
          end
      end
end

 Y=sim(net,PP);
 Y2=vec2ind(Y);
  correct=0;
 for i=1:22
      if (Y2(i)==1)&(i<=8)
          correct=correct+1;
      elseif (Y2(i)==2)&(i>8)
          correct=correct+1;
      end
 end
 %%%%%%%%%%%%%%%%%%%%%%
%   a1=0.35565;
%   a2=0.414226;
%   a3=0.230126;
 for i=1:22
      if (Y2(i)==1)
          X2(i)=0;
      elseif (Y2(i)==2)
          X2(i)=1;
      end
 end
```

```
 %biaohao=[540];%加权 APC
%biaohao=[114 259 533 635 859 994 1354 1360 1700 1896];%加权 RAS
biaohao=[114 259 533 635 859 994 1354 1896];%加权 RAS 较优，正确 19 个

for i=1:length(biaohao)
      for j=1:1909
            if (j==biaohao(i))
                  P1=data(j,1:14);
                  P2=data(j,23:48);
                  P(i,1:40)=[P1 P2];
                  break;
            end
      end
end

 %T1=[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2];
 T1=[1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ];
% 将下标矩阵变为单值矢量组作为网络的目标输出
T=ind2vec(T1);

%   设计概率神经网络
sp=0.1; %扩展常数
net=newpnn(P,T,sp);

%   对网络进行仿真，并绘出分类结果
Y=sim(net,P);
Y1=vec2ind(Y);

%   对一组新的数据进行分类
for i=1:length(biaohao)
      for j=1:1909
            if (j==biaohao(i))
                  P3=data(j,15:22);
                  P4=data(j,49:62);
                  PP(i,1:22)=[P3 P4];
                  break;
            end
      end
end

 Y=sim(net,PP);
 Y3=vec2ind(Y);
  correct=0;
 for i=1:22
```

```
        if (Y3(i)==1)&(i<=8)
            correct=correct+1;
        elseif (Y3(i)==2)&(i>8)
            correct=correct+1;
        end
 end
 %%%%%%%%%%%%%%%%%%%%%
 a1=0.35565;
 a2=0.414226;
 a3=0.230126;
%    a1=0.4843;
%    a2=0.35897;
%    a3=0.15673;

 for i=1:22
        if (Y3(i)==1)
            X3(i)=0;
        elseif (Y3(i)==2)
            X3(i)=1;
        end
 end
%    X2=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 ];
%    X3=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 ];
 Score=a1*X1+a2*X2+a3*X3;
 for i=1:22
        if (Score(i)<=a1)
            Result(i)=0;
        elseif (Score(i)>=(a2+a3))
            Result(i)=1;
        end
 end

 correct=0;
 for i=1:22
        if (Result(i)==0)&(i<=8)
            correct=correct+1;
        elseif (Result(i)==1)&(i>8)
            correct=correct+1;
        end
 end
correct_ratio=correct/22;
fprintf('正确个数：%g\n\n',correct);
fprintf('正确率：  %g\n\n',correct_ratio);
```