基于神经网络的纠错输出编码方法研究

周进登[1],周红建[1],杨 云[1],郭长华[2],胡洪宇[3]
(1.空军装备软件测评中心,北京 100076;2.广空装备部军通处,广东广州 510071;3.驻成飞公司军事代表室,四川成都 610091)

摘 要: 构造基于数据编码矩阵是目前利用纠错输出编码解决多类分类问题的研究重点.为此提出利用单层感知器作为学习框架,结合解码策略把输出编码矩阵各码元值映射为感知器网络中的权值,同时引入含权值取值约束的目标函数作为该网络代价函数,并对其进行学习,最终得到基于子类划分的数据编码矩阵.实验中利用人工数据集和UCI数据集并选择线性逻辑分类器作为基分类器分别进行测试,通过与几种经典编码方法比较,结果表明该编码方法能在编码长度较小情况下得到更好的分类效果.

关键词: 多类分类;纠错输出编码;神经网络
中图分类号: TN912-34 **文献标识码:** A **文章编号:** 0372-2112 (2013) 06-1114-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.06.012

Coding Design for Error Correcting Output Codes Based on Neural Network

ZHOU Jin-deng[1], ZHOU Hong-jian[1], YANG Yun[1], GUO Chang-hua[2], HU Hong-yu[3]
(1. *Air Force Equipment Software Testing Center*, *Beijing* 100076, *China*;
2. *Equipment Department of Air Force of Guangzhou Military Region*, *Guangzhou*, *Guangdong* 510071, *China*;
3. *Military Representatives Office of PLA*, *Chengdu*, *Sichuan* 610091, *China*)

Abstract: It is known that error-correcting output codes (ECOC) is a common way to model multiclass classification problems, in which the research of encoding based on data especially attracts attentions. In this paper, we proposed a method for learning error-correcting output codes with the help of a single layered perception neural network. To achieve this goal, the code elements of ECOC are mapped to the weights of network for the given decoding strategy, and an object function with the constrained weights used as a cost function of network. After the training, we can obtain a coding matrix including lots of subgroups of class. Experimental results on artificial data and UCI with logistic linear classifier (LOGLC) as the binary learner show that our scheme provides better performance of classification with shorter length of coding matrix than other state-of-the-art encoding strategies.

Key words: multiclass categorization; error-correcting output codes; neural network

1 引言

纠错输出编码作为一种多类分解框架,能有效的把多类问题分解为多个二类问题,进而把多类分类简化为二类分类问题,从而有效利用经典的两类分类方法.在利用纠错输出编码解决多类分类问题领域中,如何构建基于数据的编码矩阵是有效利用此类方法的基础.在此方面的研究有:Alpaydin 和 Mayoraz 于 1999 年率先研究了基于样本数据特征的编码问题,提出基于数据的反向传播编码确定方法(back propagation algorithm)[1].2001 年,Utschick 和 Weichselberger 利用期望最大化法则(expectation maximization algorithm)通过对最大似然目标函数进行优化找出最适合样本空间的编码矩阵[2].2002 年,Crammer 和 Singer 研究证明在基分类器确定的前提下,计算最优编码矩阵是一个 NP 难问题,从理论上指出了基于问题域设计最优编码矩阵的困难[3].2006 年,Pujol 和 Radeva 等人提出一种判别式 ECOC 编码方法,其针对样本集特征,利用决策树结构试探性地逐步构造类间间隔最大的子类从而确定编码矩阵[4].2008 年,Escalera 和 Tax 等人提出针对样本集线性不可分问题,提出对基类子集再分割的 ECOC 编码方法[5].在国内,2008 年,尹安容和谢湘等人利用 Hadamard 纠错码来简化 ECOC 的构造,该方法实现简单,且容易构造出性能优越的纠错码本[10].蒋艳凰和赵强利等人提出一种搜索编码方法,通过把编码阵每一行看成是某一整数的二进制码串,进而利用编码阵构造的约束条件在某一整数区间搜索出满足要求的纠错输出编码[6].相关研究还见于文献[7,11].这些研究都有力的促进了 ECOC 编码

收稿日期:2011-09-25;修回日期:2012-10-11

不断发展.

基于数据编码矩阵的应用能很好的提高 ECOC 的分类效果.究其原因,文献[5]指出:基于数据编码矩阵能最大可能获得数据类别子类划分(subgroups partition),而类别子类划分往往为相关性较小且最易于分类的二类划分,因此基于此类划分构造的二分类器就能达到较高准确率,从而实现分类效果的整体提高.然而目前存在的问题是缺乏寻找类别子类划分的有效手段,虽然文献[5]在此方面做了一定的探索,但其方法过于复杂,学习效率低下,且每一次获得最优子类划分都需要利用训练集进行学习并利用验证集进行验证,其学习效率和最终分类效果都不是很理想.为此,本文提出一种结合解码策略,利用单层感知器学习网络把编码矩阵中各码元值的确定转化为网络中权值学习的过程,并最终得到满足要求的基于数据编码矩阵.

2　纠错输出编码(ECOC)

ECOC 框架即用一种二元或三元的编码矩阵实现多类类别分解和基分类器集成.在其编码矩阵中,二元码用 $\{-1,+1\}$ 表示,三元码用 $\{-1,0,+1\}$ 表示,"-1"代表一类,"$+1$"代表另一类,"0"表示该码字位所对应的类在其列所形成的二类划分中被忽略(即不参与由该列所产生的基分类器的训练).图 1 给出了四种常见的 ECOC 分类系统示意图,以编码矩阵来区分,图 1(a)是"一对多"编码阵(one-versus-all)、图 1(b)是"一对一"编码阵(one-versus-one)、图 1(c)是密集随机阵(dense random)、图 1(d)是稀疏随机阵(sparse random).

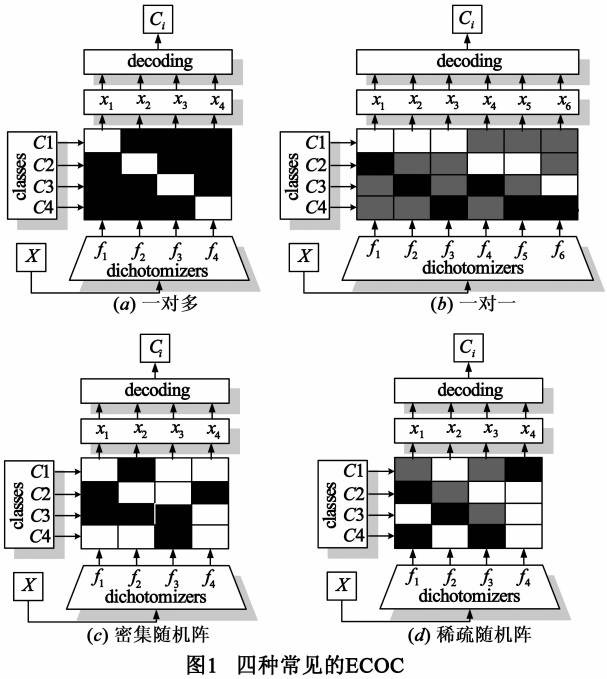

图1　四种常见的ECOC

图 1 中所有编码阵的每一行代表某一类 $Ci(i=1,2,3,4)$的码字,每一列代表样本的一种二类划分,码元"1"、"-1"和"0" 分别用白色、黑色和灰色表示.在训练阶段,每一个基分类器 $f_i(i=1,2,\cdots,6)$的训练样本根据其在编码阵对应的列重新划分,然后分别训练得到与该列对应的二类分类器.例如,在图 1(d)中对基分类器 f_3 进行训练时,白色对应的 $C2$ 为一类,黑色对应的 $C4$ 为另一类,而灰色对应的 $C1$ 和 $C3$ 不参与该基分类器的训练.以此类推可训练得到四个二类分类器 $\{f_1,f_2,f_3,f_4\}$.在测试阶段,给定一个测试样本 X,同时利用这四个二类分类器对其进行分类,结果为一码字向量(x_1,x_2,x_3,x_4)(其中 $x_i\in\{-1,+1\}$),最后根据某种解码规则(即融合策略)对其进行解码即可得最终分类结果.下一节我们将重点讨论本文提出的基于数据编码矩阵确定框架.

3　基于感知器的编码矩阵确定方法

给定初始编码矩阵 $\boldsymbol{M}_{\text{init}}\in\{-1,0,+1\}^{K\times L}$,$K$ 为类别数,L 为基分类器个数.利用训练样本对该编码矩阵各列所对应的基分类器进行训练,可得 L 个基分类器$(f_1,f_2,\cdots,f_L)$,则对样本集中每一个元素 $x^{(t)}$,可得由基分类器输出组成的向量 $\boldsymbol{a}^{(t)}=(f_1^{(t)}(x),f_2^{(t)}(x),\cdots,f_L^{(t)}(x))$,而训练样本的所属类别 $y^{(t)}$已知,故对任意训练样本可得一组合向量 $\boldsymbol{b}^{(t)}=(\boldsymbol{a}^{(t)},y^{(t)})$.本节将重点介绍如何利用该组合向量构造基于数据的编码矩阵,对此我们提出结合解码策略并以神经网络为学习理论依据,把编码矩阵中各码元值的确定转化为单层感知器网络中权值确定的方法—基于感知器的编码矩阵确定方法(Encoding Based on Pereceptron for Error Correcting Output Codes,PECOC).

对任意训练样本 x,基分类器的输出为$(f_1(x),f_2(x),\cdots,f_L(x))$,解码规则如下:

$$o_k=\sum_{l=1}^{L}m_{kl}f_l(x) \tag{1}$$

$$y=\arg\max_k o_k \tag{2}$$

其中 m_{kl}为编码矩阵第 k 行第 l 列元素.对于给定的训练样本,由于其类别已知且对每一个训练样本可得其基分类器输出向量,那么根据式(1)"反推"m_{kl}则成为可能.这里需要解决的问题是根据什么样的方法去实现这样一个"反推".在此单层感知器网络将作为这种"反推"方法的具体实现.在该实现框架中,我们抽取部分训练样本作为编码矩阵分类效果验证集,剩余训练样本对应各基分类器的输出为该感知器输入层的输入,感知器的输出为样本对应所属类别,感知器中每个节点的权值为编码矩阵对应列中的码元,如图 2 所示.

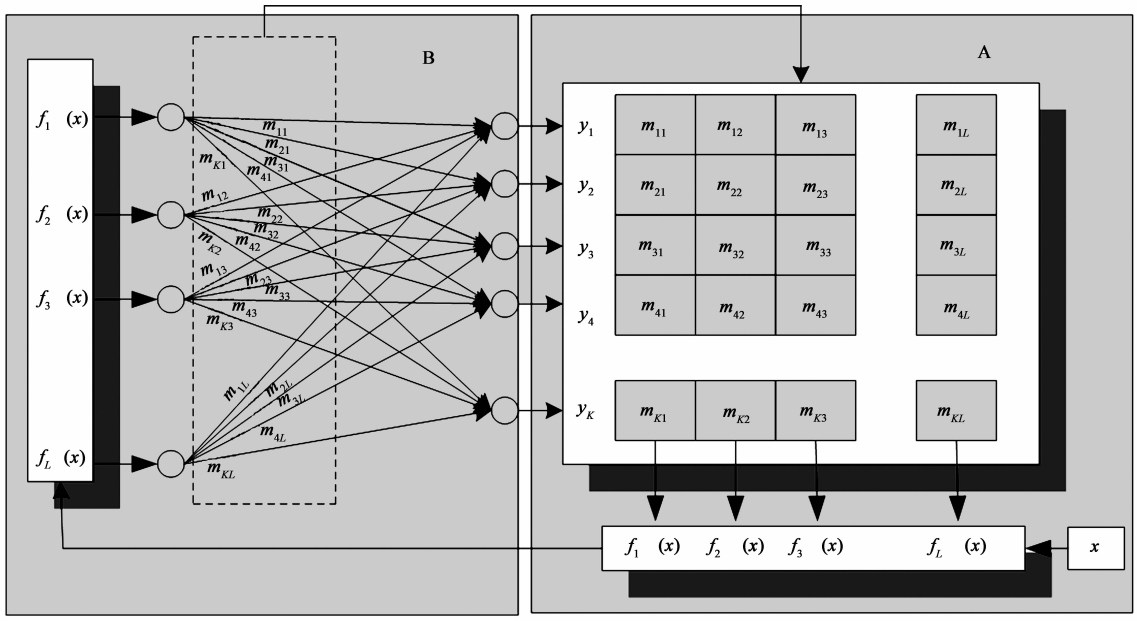

图2 基于单层感知器编码矩阵确定框架

在该编码矩阵确定框架中,首先在框架的 A 部分,对每个训练样本利用训练好的基分类器对其进行分类得到一输出向量,把该输出向量作为 B 部分单层感知器的输入进行学习,该部分学习的目标是通过不断修正权值 m_{kl},使感知器的代价函数尽可能小.当满足学习停止条件时,感知器网络的权值便作为 A 部分编码矩阵各码元的更新值,进而重构获得新编码矩阵,利用该新编码矩阵更新基分类器对应的二类划分并对其进行训练,应用该训练后的基分类器及更新后的编码矩阵根据式(1)和式(2)对验证集进行分类决策并最终得到验证集的分类错误率,若该错误率满足给定条件则"反推"结束,得到编码矩阵为最终编码阵;否则,则把训练时各训练样本的输出向量重新作为框架 B 部分感知器的输入并再次进行学习,如此循环直到"反推"结束条件满足,因此我们可得算法 1 基于单层感知器网络学习的编码矩阵确定方法(PECOC)具体流程.

要顺利实现算法 1 所述"反推"过程,其中最重要的步骤是单层感知器学习的权值更新过程,而权值更新可通过算法 1 所示步骤实现:

算法 1 基于单层感知器网络学习的编码矩阵确定方法(PECOC)

输入:初始编码矩阵 $\boldsymbol{M}_{\text{init}} \in \{-1,0,+1\}^{K\times L}$,训练样本集 R,单层感知器满足收敛最小错误率 λ 及判断是否收敛学习最大次数 ζ,反推结束条件 r

Step1 从训练样本集 R 中随机抽取部分样本作为验证集 R_{validate},更新训练样本集 $R' = R - R_{\text{validate}}$,令编码矩阵 $\boldsymbol{M} = \boldsymbol{M}_{\text{init}}$.

Step2 基于编码矩阵 $\boldsymbol{M}$ 和训练样本集 R' 训练得到 L 个基分类器,对验证集 R_{validate} 中每一个样本 x 利用此 L 个基分类器分别进行决策得到输出向量 $(f_1(x), f_2(x), \cdots, f_L(x))$,根据公式(1)和(2)判定其所属类别,据此计算验证集分类错误率 $\text{ER}_{\text{validate}}$.若 $\text{ER}_{\text{validate}} \leqslant r$ 则转至 Step4,否则对训练集 R' 中每一个样本利用已获得的 L 个基分类器分别进行决策分类并得到对应输出向量.

Step3 将上一步得到 R' 中各元素对应的基分器输出向量作为单层感知器网络输入层的输入,选定感知器网络节点激励函数 g 和网络学习总代价函数 E,对该单层感知器网络进行训练并判断其学习次数是否达到收敛最大学习次数 ζ,若达到且 $E > \lambda$ 则说明感知器网络不收敛,若未达到或刚好达到且 $E \leqslant \lambda$ 则感知器收敛,当可判断感知器收敛或不收敛时感知器停止学习,利用该网络各节点权值更新编码矩阵 M 中对应的码元值,并转至 Step2.

Step4 反推结束,保存最终得到的编码矩阵.

输出:编码矩阵 $\boldsymbol{M}_{\text{final}}$

$$m_{kl}(T+1) = m_{kl}(T) + \eta \Delta m_{kl}(T) \tag{3}$$

η 为学习效率,取值范围一般为 $(0,1]$.如何计算 $\Delta m_{kl}(T)$ 便成为我们研究的重点.注意到算法 1 所示编码矩阵确定方法中初始编码矩阵 $\boldsymbol{M}_{\text{init}}$ 的选择是事先给定的,那么初始编码矩阵选择是否会影响最终编码矩阵的优劣也是需要认真考虑的,对此我们将分别进行讨论.

3.1 感知器的学习问题

引入代价函数:

$$E = -\sum_t \sum_k y_k^{(t)} \ln p_k^{(t)} \tag{4}$$

其中 $y_k^{(t)} = \begin{cases} 1, & \text{if } x^{(t)} \in \text{class}_k \\ 0, & \text{otherwise} \end{cases}$,考虑该代价函数的建立是以解决多类分类为目标,因此其必须包含各类别的约束信息,故该代价函数以最小化样本对应各类后验概率负对数和为目标,其意义为若每个测试样本的预测值与其真实值相同,那么该代价函数则最小,p_k 为样本 x 对应第 k 类的后验概率:

$$p_k(x)=\frac{\exp\left(\sum_{l=0}^{L} m_{kl}f_l(x)\right)}{\sum_{k=1}^{K}\exp\left(\sum_{l=0}^{L} m_{kl}f_l(x)\right)} \tag{5}$$

其中 m_{k0}为偏差项.式(5)即为感知器网络中的节点激励函数,即 $g(x)=p(x)$,把式(5)带入式(4)得到感知器训练的代价函数为:

$$E=-\sum_t\left(\sum_{l=0}^{L} m_{kl}f_l^{(t)}(x)-\ln\left(\sum_{i=1}^{K}\exp\left(\sum_{l=0}^{L} m_{kl}f_l^{(t)}(x)\right)\right)\right) \tag{6}$$

考虑到权值 m_{kl}的取值范围为一离散集合即 $m_{kl}\in\{-1,0,+1\}$,而一般的神经网络训练过程中,其对应节点的权值为连续值,故在求取上述代价函数最小值时必须考虑权值 m_{kl}的取值影响,为此参考文献[8],我们假设权值 m_{kl}为一混合分布的抽样取值,且该混合分布由三个均值分别为-1、0 和 1 的高斯分布组成即:

$$p(m_{kl})=a_1p_1(m_{kl})+a_2p_2(m_{kl})+a_3p_3(m_{kl}) \tag{7}$$

其中 $a_i(i=1,2,3)$为先验概率,$p_i(m_{kl})$为高斯分布:$p_i(m_{kl})\sim N(\mu_i,\sigma_i^2)$

$$p_i(m_{kl})=\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(m_{kl}-\mu_i)^2}{2\sigma_i^2}\right) \tag{8}$$

其中 $\mu_1=-1,\mu_2=0,\mu_3=1$. 为了使 m_{kl}能最大可能的取自于该混合高斯分布,我们设其基于训练样本的似然估计为:

$$l(m_{kl})=\prod_{k,l}\sum_{i=1}^{3} a_ip_i(m_{kl}) \tag{9}$$

则使该估计为最大似然估计的 m_{kl}即为满足要求的值,为使该约束条件能统一到式(6)所示的代价函数中去,我们取 $\Omega=-\ln l(m_{kl})$作为该代价函数的正则化参数项(regularization term),并联合式(6)得到最终代价函数为:

$$\begin{aligned}E'=E+\nu\Omega=&-\sum_t\left(\sum_{l=0}^{L} m_{kl}f_l^{(t)}(x)\right.\\&\left.-\ln\left(\sum_{i=1}^{K}\exp\left(\sum_{l=0}^{L} m_{kl}f_l^{(t)}(x)\right)\right)\right)\cdots\\&-\nu\sum_{k,l}\ln\left(\sum_{i=1}^{3} a_ip_i(m_{kl})\right)\end{aligned} \tag{10}$$

其中 ν 为正则化系数(regularization coefficient),该参数值定义了式(10)中两子项对代价函数重要性关系.为求该代价函数的极小值,利用梯度下降法对式(10)求沿 m_{kl}方向的梯度得到:

$$\frac{\partial E}{\partial m_{kl}}=-\sum_t y_k^{(t)}\frac{f_l(x)\left(\sum_{k=1}^{K}\exp\left(\sum_{l=0}^{L} m_{kl}f_l(x)\right)-\exp\left(\sum_{l=0}^{L} m_{kl}f_l(x)\right)\right)}{\sum_{k=1}^{K}\exp\left(\sum_{l=0}^{L} m_{kl}f_l(x)\right)}=-\sum_t y_k^{(t)}f_l(x)(1-p_k^{(t)}) \tag{11}$$

$$\frac{\partial\Omega}{\partial m_{kl}}=-\frac{-a_1p_1(m_{kl})\frac{m_{kl}-\mu_1}{\sigma_1^2}--a_2p_2(m_{kl})\frac{m_{kl}-\mu_2}{\sigma_2^2}-a_3p_3(m_{kl})\frac{m_{kl}-\mu_3}{\sigma_3^2}}{\sum_{i=1}^{3} a_ip_i(m_{kl})}=\sum_{i=1}^{3}\pi_i(m_{kl})\frac{m_{kl}-\mu_i}{\sigma_i^2} \tag{12}$$

其中 $\pi_i(m_{kl})$属于各分布的后验概率即:

$$\pi_i(m_{kl})=\frac{a_ip_i(m_{kl})}{\sum_{i=1}^{3} a_ip_i(m_{kl})}\ (i=1,2,3) \tag{13}$$

联合式(10)、(11)和(12),最终得到 $\Delta m_{kl}(T)$的计算公式为:

$$\begin{aligned}\Delta m_{kl}(T)=&-\sum_t y_k^{(t)}f_l(x)(1-p_k^{(t)})\\&+\nu\sum_{i=1}^{3}\pi_i(m_{kl})\frac{m_{kl}-\mu_i}{\sigma_i^2}\end{aligned} \tag{14}$$

联合式(3)、(14)并代入公式(13)便可得到网络中每一个权值

$$m_{kl}=\begin{cases}-1,\text{when }\pi_1(m_{kl})=\max(\pi_i)\\0,\text{when }\pi_2(m_{kl})=\max(\pi_i)\\+1,\text{otherwise}\end{cases} \tag{15}$$

由式(14)得到感知器网络每一次学习中权值更新值 $\Delta m_{kl}(T)$.注意到经过更新后的 m_{kl}不再是离散集合 $\{-1,0,+1\}$中的元素,而当该网络达到学习停止条件时,利用式(15)将其转化成该离散集合中的某一元素,因此在每一次利用感知器网络权值更新输出编码矩阵时都会引入误差,即更新后编码矩阵中的每一个码元值并不对应着收敛感知器网络的最优权值.解决这一问题方法是在感知器学习过程中通过不断修正正则化系数 ν,找到一组最小的 $\sigma_i(i=1,2,3)$值,这样 m_{kl}的取值就可以集中分布于均值的两侧,从而使由式(15)得到的权值与最优化网络得到的权值尽可能接近,以此降低此误差带来的影响.

3.2　初始编码矩阵的选择问题

在算法 1 所示基于单层感知器网络学习编码矩阵确定方法(PECOC)中,初始编码矩阵的选定是否能影响该算法学习效率和最终编码的优劣是值得思考的问题.考虑到单层感知器学习过程中其初始权值的选择是不影响学习收敛性的,而感知器网络的权值对应输出编码矩阵各码元值,因此可以肯定的是初始编码矩阵的选择不会对感知器网络学习收敛速度产生影响.

然而,我们知道初始编码矩阵决定了感知器输入层的节点数即二类划分数,同时也为基分类器的个数,而该个数却影响基于输出纠错编码多类分类的学习效率和分类准确率,一个好的基于输出编码矩阵多类分类框架应该是能利用最少的基分类器达到较高的分类准确率的框架.

为此,要使本文所提基于数据编码矩阵确定方法能达到此要求通常有两个途径,一是在选择初始编码矩阵时选择能将问题域粗略划分为某几个子类的编码矩阵,由此确定解决该分类问题的最佳基分类器个数范围;二是随机选择初始编码矩阵,在感知器的训练中把输入层节点个数变为可变量,在保证收敛的前提下利用一定的搜索策略(如顺序浮点搜索算法(SFFS))找到最优输入层个数,从而确定最优基分类器个数.在第4节中为了实验操作的方便性,第一种方案将作为具体实施方案.

下一节我们将通过实验验证本文所提基于数据编码矩阵的分类效果,并通过与已有几种数据编码和经典的事前编码进行比较,对比分析其各自分类效果.

4 实验

在本节中我们将利用人工数据集和UCI数据集分别用于验证上节所述结论和比较现有各种基于数据编码及经典的事前编码与本文所提方法的优劣.为此,我们将从实验数据、实验设计和实验结果及分析分别加以介绍.

4.1 实验数据

实验中采用的第一种数据为5类二维正态分布数据集,各类别数据的先验概率相同,其概率密度函数如式(16)所示,各类别分布参数如表1所示.

$$p(x\mid \text{class}_k)=\frac{1}{2\pi\sigma_k^2}\exp\left[\frac{\|x-\boldsymbol{\mu}_k\|^2}{-2\sigma_k^2}\right],k=1,2,\cdots 5 \quad (16)$$

使用表1所示参数,根据贝叶斯法则我们可以得到各类别的后验概率.因此,由贝叶斯分类器可以得到各类别的决策分界面,且通过计算可以得到该五类人工数据集的贝叶斯分类错误率为25.76%.数据分布及其分类决策面如图3所示.

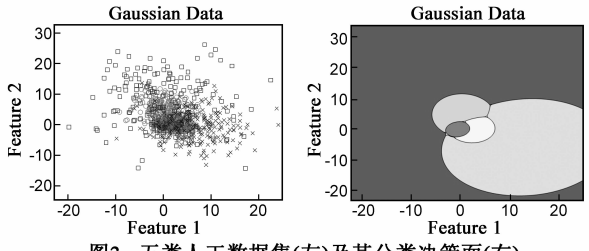

图3 五类人工数据集(左)及其分类决策面(右)

第二种为公共数据集,本文将选取UCI数据库作为公共数据集,各数据分布属性如表2所示.

表1 五类人工数据集各类别分布参数

Class	Prior Probabilities	Mean Vectors	Covariance Matrices
C_1	$P(C_1)=\frac{1}{5}$	$\boldsymbol{\mu}_1=(0,0)^{\mathrm{T}}$	$\boldsymbol{\Sigma}_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$
C_2	$P(C_2)=\frac{1}{5}$	$\boldsymbol{\mu}_2=(3,0)^{\mathrm{T}}$	$\boldsymbol{\Sigma}_2=\begin{pmatrix}2&0\\0&2\end{pmatrix}$
C_3	$P(C_3)=\frac{1}{5}$	$\boldsymbol{\mu}_3=(0,5)^{\mathrm{T}}$	$\boldsymbol{\Sigma}_3=\begin{pmatrix}3&0\\0&3\end{pmatrix}$
C_4	$P(C_4)=\frac{1}{5}$	$\boldsymbol{\mu}_4=(7,0)^{\mathrm{T}}$	$\boldsymbol{\Sigma}_4=\begin{pmatrix}5&0\\0&5\end{pmatrix}$
C_5	$P(C_5)=\frac{1}{5}$	$\boldsymbol{\mu}_5=(0,9)^{\mathrm{T}}$	$\boldsymbol{\Sigma}_5=\begin{pmatrix}8&0\\0&8\end{pmatrix}$

表2 UCI数据集各数据属性描述

#	Problem	# Train	# Attributes	# Classes
(*a*)	Yeast	1484	8	10
(*b*)	Segmentation	2310	19	7
(*c*)	Satimage	6435	36	6
(*d*)	Glass	214	9	7
(*e*)	Vehicle	846	18	4
(*f*)	Zoo	101	18	7
(*g*)	Wine	178	13	3
(*h*)	Vowel	990	10	11
(*i*)	Ecoli	336	8	8
(*j*)	Iris	150	4	3

4.2 实验设计

为了验证本文所提基于数据编码确定方法确实能产生简单易分的二类划分,我们事先针对第一类人工数据集定义能产生较复杂二类划分编码矩阵 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$,并将此编码矩阵作为本文所提基于感知器学习编码矩阵确定方法的初始编码阵,利用该框架产生符合要求的编码阵,通过对比此两种编码矩阵各基分类器决策边界,判断其是否有效(能产生简单易分的二类划分),进而验证所得结论.编码阵 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$ 及其所基分类器对应的二类划分决策子边界如图4所示.

在验证本文所得基于数据编码矩阵的最终分类效果时,将分别利用人工数据集和UCI数据集进行验证.人工数据集用于比较上述两种不同初始化编码矩阵 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$ 同经PECOC学习得到后的最终编码矩阵的各基分类器的分类效果及编码矩阵最终分类效果.UCI数据集用于本文PECOC框架所得基于数据编码矩阵同几种经典的基于数据编码和事前编码方法进行比较,这些经典的方法有:一对一编码(1-vs-1)、一对多编码(1-vs-all)、密集随机编码(dense random)、稀疏随机编码(sparse random)、判别式编码(Discriminant Error Correcting

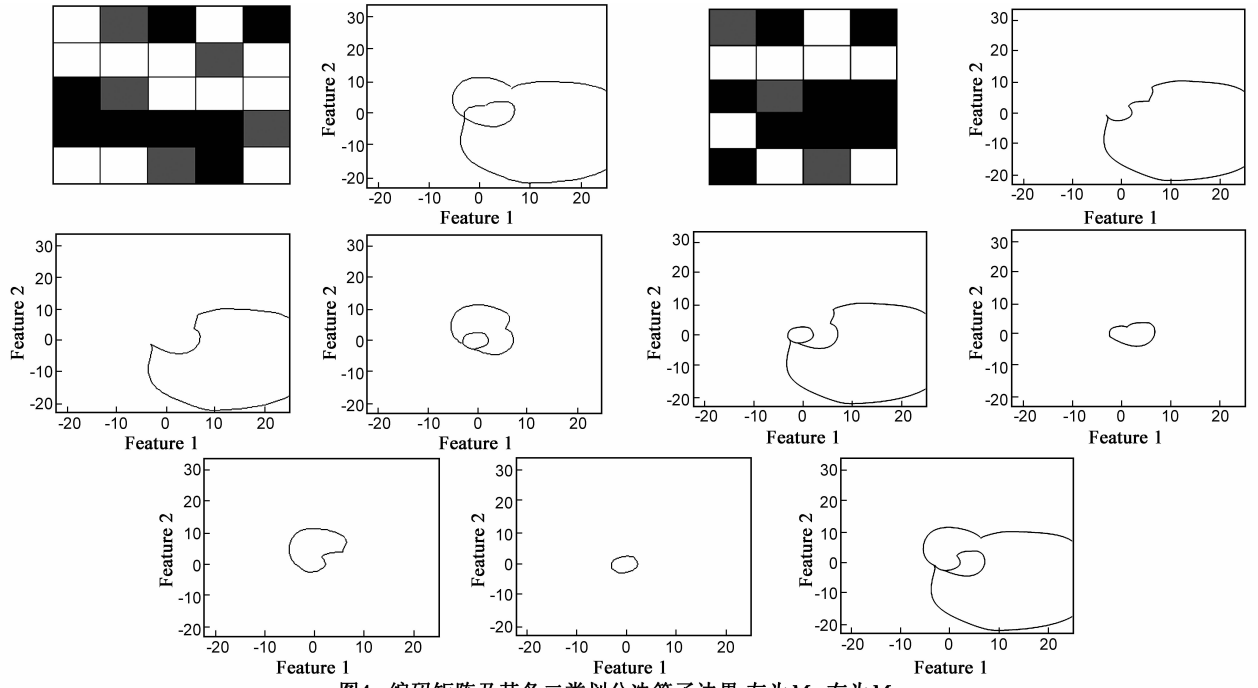

图4　编码矩阵及其各二类划分决策子边界,左为M_1, 右为M_2

Output Codes,DECOC)以及子类编码(Subclass Error Correcting Output Codes,SECOC).在构造 PECOC 时为简化实验初始编码矩阵选择文献[4]提出的判别式纠错输出编码(DECOC).在对解码策略选择时,四种解码策略:Hamming距离解码、欧式距离解码、线性损失函数解码和指数损失函数解码将分别用于各编码矩阵分类效果的比较,同时基分类器为 LOGLC 分类器.

在估计分类错误率时我们采用十重交叉验证来进行,并利用双边估计 t 检验法来计算置信水平为 0.95 的分类错误率置信区间作为最终结果,计算公式如下:

$$\frac{|\bar{x}-\mu|}{\sigma/\sqrt{n}} \geqslant t_{0.025}(n-1) \tag{17}$$

μ,σ 分别表示十重交叉验证的均值和标准差,$t_{0.025}(9)=2.2622$. 同时为了对实验结果进行统计分析,采用 Nemenyi 检验法对各编码矩阵分类效果之间的差异显著性进行检验.

4.3　实验结果和分析

4.3.1　人工数据集

图 5 所示为两种复杂二类划分初始编码矩阵 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$ 经 PECOC 得到的最终输出编码矩阵各二类划分决策面,对比两种编码矩阵前后不同的二类划分决策面,可以清楚的看到经 PECOC 方法生成的最终输出编码矩阵,其二类划分的决策面有较大的不同,图 5(a)所示的二类划分决策面相比其初始化编码矩阵 $\boldsymbol{M}_1$ 的二类划分决策面保留了原编码矩阵中第 2、4、5 列所对应的决策面,原因是此三列所对应的决策面较为简单,在表 4 实验结果中我们看到基于此划分训练出的基分类器其错误率较低.此外,$\boldsymbol{M}_1$ 中原本较为复杂的决策面(第 1、3 列对应的二类划分)经 PECOC 框架学习后,可以很清楚的看到其复杂度降低.当初始编码矩阵为 $\boldsymbol{M}_2$ 时,仅有第 3 列对应的二类划分决策面与最终生成的矩阵相同,其余各二类划分决策面经 PECOC 学习后,复杂性大为降低.注意到此两种不同初始编码矩阵经 PECOC 学习后生成的最终编码矩阵中包含大量相同的简单二类划分决策面,如图示 5(a)、(b)所示.为此,进一步验证了 PECOC 确实能找到简单、易分的二类划分决策分类面.

表 3　人工数据集中编码矩阵各基分类器分类错误率及编码矩阵最终分类错误率

	f_1	f_2	f_3	f_4	f_5	Final
$\boldsymbol{M}_1$	61.67%	35.23%	30.21	25.98%	16.35%	38.75%
$\boldsymbol{M}_{\text{PECOC:1}}$	46.43%	35.23%	19.56%	25.98%	16.35%	29.54%
$\boldsymbol{M}_2$	39.45%	50.53%	24.67%	48.46%	—	41.64%
$\boldsymbol{M}_{\text{PECOC:2}}$	35.23%	16.35%	24.67%	34.32%	—	32.56%

表 4 列出了两种不同初始化编码矩阵 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$ 及其经过 PECOC 学习后产生的编码矩阵 $\boldsymbol{M}_{\text{PECOC:1}}$、$\boldsymbol{M}_{\text{PECOC:2}}$各基分类器的分类错误率以及最终编码矩阵的分类效果,可以看出两种基于数据学习后产生的编码矩阵所包含的基分类器与对应初始化矩阵包含的基分类器相比较,其分类错误率有较明显降低,此外其最终分类效果也具有明显差异.

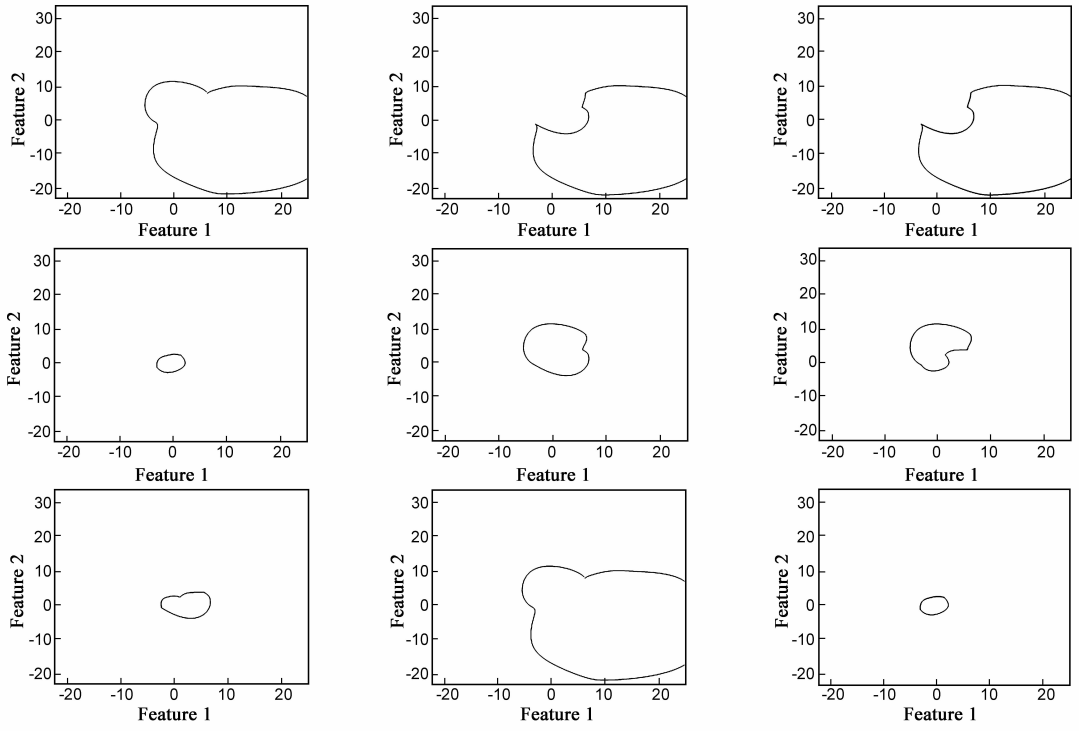

图5 经PECOC后产生的最终编码矩阵各二类划分决策面,(a)M_{init}=M_1,(b)M_{init}=M_2

4.3.2 UCI 数据集

在本实验中四种不同解码策略以及两种不同基分类器被用于比较本文所提 PECOC 编码方法与前文所述六种经典编码方法.在完成这些实验之后,我们总共进行了 800 次十重交叉验证测试,为了得到具有统计意义的实验结论,我们利用秩和检验法对实验结果进行分析,其中秩水平计算如下:

$$R_j = \frac{1}{J}\sum_i r_i^j \tag{18}$$

r_i^j 为每种编码方式在第 i 类问题中用基于第 j 种解码策略所得到的秩大小,J 为每种方法所进行的实验次数,在本次实验中 J 为:4 种解码策略 × 10 种 UCI 数据.表 4 为各方法所对应的秩和平均数,其中加粗部分为基于各解码策略中秩和平均最小值即对应分类错误率最小的编码方式.

表 4 各编码策略秩和平均数比较

Coding	1 vs all	1vs 1	dense	sparse	DECOC	SECOC	PECOC
HD	4.60	4.00	3.10	4.80	4.80	3.30	3.40
AED	6.20	3.70	3.20	4.40	2.90	4.90	2.70
LLB	5.40	4.20	3.60	4.20	3.70	3.50	3.00
ELB	5.40	4.90	3.90	4.20	3.80	3.50	2.00
Global Rank	5.40	4.20	3.45	4.40	3.80	3.80	2.78

从表 4 中可以看出本文所提 PECOC 秩和平均数最小为 2.78,dense 次之,1 vs all 最大.为了验证这七种不同经典编码矩阵分类效果具有统计意义上的显著差别,我们利用 Nemenyi 检验方法—即两种方法具有显著性差异当此两种方法的秩和平均差大于临界值 CD (Critical Difference value)[9]

$$\mathrm{CD} = q_\alpha\sqrt{\frac{k(k+1)}{6J}} \tag{19}$$

其中 q_α 可通过查询"The Studentized Range Statistic"表得到,k 为所要验证的方法数,J 为每次实验的次数.在本实验中我们比较了 7 种方法在置信水平为 $\alpha = 0.05$ 下的分类效果如表 4 所示,即 $k = 7$, $q_{0.05} = 1.860$,代入式(19)可得差异临界值为 1.218.观察表 4 可知 PECOC 的秩平均数比其余编码矩阵秩平均数都要小且差值都大于差异临界值,因此我们可以说 PECOC 在 95% 的置信区间都要好于其它编码方法.

5 结论

构建基于数据编码矩阵是目前利用纠错输出编码解决多类分类问题的研究热点之一.利用已知类别训练样本信息构建适合问题域最优编码矩阵的难点是如何以最少的基分类器个数包含尽可能多的子类划分.本文所提 PECOC 能有效的实现该目标并能最终提高对多类问题的分类效果.此外,注意到在 PECOC 框架中基分类器的学习与感知器网络的学习是分别进行的,即在构建基于数据的编码矩阵时我们并未考虑基分类器学习的因素,然而事实上基分类器的学习也会影响利用纠错输出编码解决多类分类问题的最终分类效果,

如何将这两者的学习统一起来将是我们下一步研究的方向.

参考文献

[1] E Alpaydin, E Mayoraz. Learning error-correcting output codes from data[A]. Proceedings of International Conference on Artificial Neural Networks (ICANN'99)[C] Verlag: Springer, 1999.743 – 748.

[2] W Utschick, W Weichselberger. Stochastic organization of output codes in multiclass learning problems[J]. Neural Comput, 2001,13(5):1065 – 1102.

[3] K Crammer, Y Singer. On the learnability and design of output codes for multiclass problems[J]. Machine Learning, 2002, 47(2):201 – 233.

[4] O Pujol, P Radeva, J Vitria. Discriminate ECOC: A heuristic method for application dependent design of error correcting output codes[J]. IEEE Trans Pattern Analysis and Machine Intelligence, 2006, 28(6):1001 – 1007.

[5] S Escalera, David M J Tax, O Pujol, P Radeva, Robert P W Duin. Subclass problem-dependent design for error-correcting output codes[J]. IEEE Trans Pattern Analysis and Machine Intelligence, 2008, 30(6):1041 – 1054.

[6] Jiang Yan-huang, Zhao Qiang-li, Yang Xue-jun. A search coding method and its application in supervised classification[J]. Journal of Software, 2005, 16(6):1081 – 1088.

[7] Jin Deng Zhou, Xiao Dan Wang, Heng Song. Research on the unbiased probability estimation of error-correcting output coding [J]. Pattern Recognition, 2011, 44(7):1552 – 1565.

[8] S Nowlan, G Hinton. Simplifying neural networks by soft weight sharing[J]. Neural Comp, 1992, 23(4):473 – 493.

[9] J Demsar. Statistical comparisons of classifiers over multiple data sets[J]. Journal of Machine Learning Research, 2006, 7(1):1 – 30.

[10] 尹安容,谢湘,匡镜明. Hadamard 纠错码结合支持向量机在多分类问题中的应用[J]. 电子学报 2008,36(1):122 – 126.
Yin An-rong, Xie Xiang, Kuang Jing-ming. Application of hadamard ECOC in multi-class problems based on SVM[J]. Acta Electronic Sinica, 2008, 36(1):122 – 126. (in Chinese)

[11] 周进登,王晓丹,权文. 加权解码在解决纠错输出编码 Consistent-Diverse 平衡问题的应用[J]. 电子学报, 2011, 20(7):1514 – 1522.
Zhou Jin-deng, Wang Xiao-dan, Quan Wen. Application of weighted decoding for the consistent-diverse balance problem of error correcting output codes[J]. Acta Electronic Sinica, 2011, 20(7):1514 – 1522. (in Chinese)

作者简介

周进登(通信作者)　男,1984 年生于江西鹰潭.博士.研究方向为模式识别和智能信息处理.

E-mail: zhoujin198417@yahoo.com.cn

杨　云　男,1969 年生于甘肃,高级工程师,研究方向为软件工程装备论证.

周红建　男,1972 生于湖北荆州,博士,研究方向为软件工程和系统仿真.