文章编号: 1001-0920(2018)02-0211-08 DOI: 10.13195/j.kzyjc.2016.1616

基于子类划分和粒子群优化的自适应编码多类分类方法

薛爱军, 王晓丹†

(空军工程大学 防空反导学院, 西安 710051)

摘 要: 纠错输出编码(ECOC)可以有效地解决多类分类问题. 基于数据的编码是主要的编码方法之一. 对此,提出一种基于子类划分和粒子群优化(PSO)的自适应编码方法,利用混淆矩阵衡量各类别的相关性,基于规则的方法对类别进行自适应组合,根据组合方案构建类别的二类划分并最终形成编码矩阵,通过引入PSO算法寻找最优阈值,从而得到最优编码矩阵. 实验结果表明,所提出的编码方法可以得到更好的分类性能.

关键词: 模式识别; 纠错输出编码; 多类分类; 子类划分; 粒子群优化

中图分类号: TP391 **文献标志码:** A

Multiclass classification of adaptive error-correcting output codes based on subclass and particle swarm optimization

XUE Ai-jun, WANG Xiao-dan†

(Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China)

Abstract: Error correcting output codes(ECOC) is an effective way to solve multiclass classification problems. Encoding based on data is one of important methods to having coding matrix. Thereforem, an adaptively encoding method based on subclass and particle swarm optimization(PSO) is proposed. Firstly, the similarity between each pair of patterns is measured by using the confusion matrix, and classes are regrouped based on rules. Then binary partitions are gotten based on pattern combination. Finally, the PSO algorithm is introduced to find the most suitable thresholds, thus obtaining a data driven coding matrix. Experimental results show that the proposed method can provide better performance.

Keywords: pattern recognition; error-correcting output codes; multi-class classification; subclass; particle swarm optimization

0 引 言

多类分类是机器学习研究的核心问题之一. 目前,求解多类分类问题大致有3种思路: 1) 直接构造多类分类器,求解全局最优化代价函数; 2) 基于Bayes定理估计属于多类的概率密度; 3) 将多类分类问题分解为多个二类问题. 纠错输出编码(ECOC)作为解决多类分类问题的通用分解框架,能有效地利用经典的二类分类方法解决多类分类问题,具有简单性和继承性等特点,受到人们的广泛关注. 目前已成功应用于人脸识别[1]、自动语音识别[2]、血管样本数据分类[3]以及交通指示牌识别[4]等诸多领域,并取得了很好的识别效果.

编码矩阵决定了每个二类问题分类的难易程度和整个框架的纠错能力,是影响ECOC分类性能的重要因素. 好的编码矩阵应该能够反映样本数据的分布特征,从而得到易于分类的二类分类问题. 因此,如何基于实际问题自适应构造编码矩阵(也称基于数据的编码),是有效利用ECOC解决多类分类问题的重点. 编码矩阵的每一列对应着一个二类划分,好的二类划分应使得两类之间的可分性良好. 为此,文献[5]提出了一种判别式ECOC编码方法; 文献[6]提出了对基类子集再分割的ECOC编码方法; 文献[7]提出了一种基于数据的联合学习模型,同时,还提出了获取编码矩阵的最优化方法[8]. 另一方面,编码矩阵各列之间的独立性是保证ECOC具有较强纠错能力的关键. 为此,文献[9]提出了子空间ECOC编码方法; 文献[10]提出了利用先验原始类结构信息提高ECOC分类性能的方法. 另外,将多类划分为二类的结果有很多. 如何从这些划分结果中挑选出性能较优(主要是指二类的分类结果较优)的一组二类划分,

收稿日期: 2016-12-20; **修回日期:** 2017-05-15.

基金项目: 国家自然科学基金项目(61273275).

作者简介: 薛爱军(1989–), 男, 博士生, 从事机器学习的研究; 王晓丹(1966–), 女, 教授, 博士生导师, 从事机器学习、目标识别等研究.

†通讯作者. E-mail: wang_afeu@126.com

进而组成编码矩阵,近年来也得到了关注. 基于此,文献[11-12]提出了基于遗传进化算法构造最优编码矩阵的方法. 文献[13]针对此类进化算法的适应度函数构造问题进行了研究. 此外,文献[14-15]提出了相关编码矩阵的自适应确定方法,这些工作为基于问题的ECOC编码方法的研究提供了新的思路. 其中,文献[14]提出的方法综合了每列类别可分性良好和整体纠错能力最佳的优点,理论上,具有较好的分类性能,因此,本文对其进行更进一步的研究.

针对文献[14]中相似阈值和相异阈值取值的问题,本文详细分析该方法的工作原理和分类性能,进而提出一种基于子类划分和粒子群优化的自适应编码方法. 该方法首先根据基于规则的子类划分算法得到子类划分,由子类划分构造编码矩阵;然后,以编码矩阵在训练数据集上的分类错误率作为代价函数,利用PSO算法对子类划分算法中相似阈值和相异阈值的取值进行优化,最优阈值对应的编码矩阵即为最终得到的最优编码矩阵;最后,采用两种不同的数据集验证本文方法的分类性能.

1 纠错输出编码

ECOC是一种利用二元或三元的编码矩阵将多类分类问题分解为若干个二类分类问题的通用集成框架. 其中:二元编码矩阵可以表示为$M \in \{-1,+1\}^{k\times l}$, “$k$”代表类别数, “$l$”代表编码长度,编码矩阵的行代表某一类,编码矩阵的列代表一个二类划分, “-1”表示其所在行对应的类在二类划分中被划分为负类, “$+1$”表示其所在行对应的类在二类划分中被划分为正类;三元编码矩阵可以表示为$M \in \{-1,0,+1\}^{k\times l}$, “$0$”表示其所在行对应的类在二类划分中被忽略. 三元编码矩阵的引入使得ECOC编码矩阵具有更加广泛的意义,二元编码矩阵甚至可以视为三元编码矩阵的一种特殊形式. 图1给出了4种常见的ECOC编码矩阵,分别是: “一对多”编码阵、“一对一”编码阵、密集随机阵、稀疏随机阵. 图1中码元“$+1$”、“-1”和“0”分别用白色、黑色和灰色表示.

利用ECOC解决多类分类问题时,通常可以分为3个阶段: 编码阶段、训练阶段和解码阶段. 编码阶段将多类分类问题分解为多个二类分类问题,形成多个二类划分,每个二类划分对应一个二类分类任务. 这一阶段是多类分类向二类分类转化的重要阶段,对分类效果具有决定性作用. 因此,编码方法的研究已成为ECOC研究的核心问题之一.

在训练阶段,根据编码矩阵的列生成二类分类器

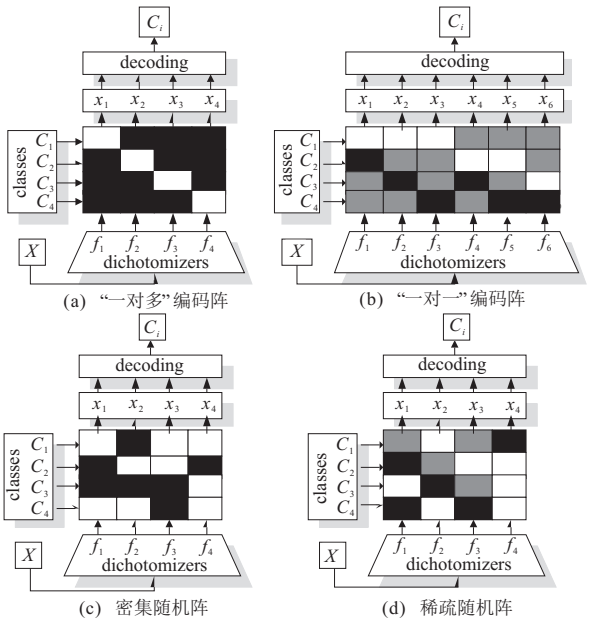

图1 4种常见的ECOC

的训练数据集,完成二类分类器$f_i(i=1,2,\cdots,l)$的训练任务. 例如,在图1(d)中对二类分类器f_3进行训练时,选择白色对应的C_2为正类,黑色对应的C_4为负类,构成训练数据集,不考虑灰色对应的类别C_1和C_3. 在此训练数据集上训练得到二类分类器f_3.

在解码阶段,根据多个二类分类器的分类结果,利用某种解码规则(融合策略)得到最终的分类结果. 例如,在图1(d)给定一个测试样本X,利用训练得到的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$x_i \in \{-1,+1\}$),最后根据码字向量与编码矩阵行之间的最小汉明距离得到分类结果.

2 基于混淆矩阵的自适应编码方法

ECOC编码方法研究的本质就是研究如何将多类分类问题分解为若干二类分类问题. 面临不同的样本数据时,ECOC编码方法具有不同的分类性能,所以理想的结果是ECOC编码可以基于样本数据自适应地构造. 自适应编码方法即是能根据样本数据分类信息构造最优编码矩阵的过程. 满足下面两个条件可以使编码方法具有好的分类性能:一是子类的类别可分性最佳;二是子类集成的纠错能力最佳. 要满足这两个条件,自适应构造编码矩阵的关键是样本数据的类别可分性计算.

文献[14]利用混淆矩阵计算样本数据的类别可分性,得到各类别的相关性度量矩阵,根据相似阈值α和相异阈值β获得子类划分,进而得到所需的模式二类划分集,组合二类划分集得到自适应编码矩阵. 分析上述过程不难看出,编码矩阵的构造依赖于子类划分,而子类划分又取决于α和β两个阈值的选

取. 因此,α和β的取值将最终影响编码矩阵的构造.

下面举例说明α和β的选取对编码矩阵构造的影响. 假设样本数据符合高斯分布,各类别的二维分布如图2所示. 选择决策树作为预分类器进行初始分类,得到混淆矩阵,再经计算得到各类别的相关性度量矩阵.

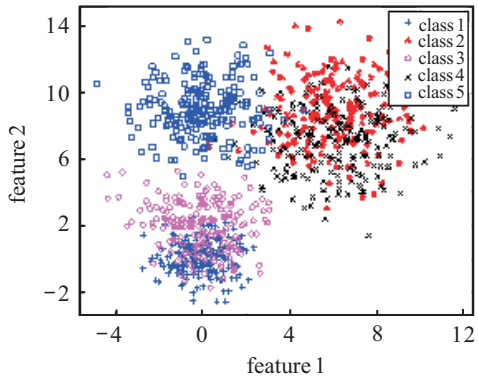

图 2　5类高斯分布样本数据

取$\alpha = 0.8, \beta = 0.83$,分别用斜体表示小于$\alpha$部分,黑体表示大于$\beta$部分,其他均为介于两者之间值部分,于是相关性度量矩阵根据阈值可分为3部分,如表1所示.

表 1　5类样本数据的相关性度量矩阵

	class1	class2	class3	class4	class5
class1	0	0.823	*0.738*	**0.868**	**0.965**
class2	0	0	0.814	*0.729*	0.826
class3	0	0	0	0.821	**0.836**
class4	0	0	0	0	**0.857**
class5	0	0	0	0	0

基于Fisher准则的子类划分算法,最终可得子类划分集$J = \{\{C_1, C_3\}, \{C_2, C_4\}, \{C_5\}\}$. 基于子类划分构造编码矩阵为

$$M = \begin{Bmatrix} 1 & -1 & -1 & 1 & 0 \\ -1 & 1 & -1 & 0 & 1 \\ 1 & -1 & -1 & -1 & 0 \\ -1 & 1 & -1 & 0 & -1 \\ -1 & -1 & 1 & 0 & 0 \end{Bmatrix}.$$

若取$\alpha = 0.82, \beta = 0.86$,则相关性度量矩阵如表2所示.

表 2　5类样本数据的相关性度量矩阵

	class1	class2	class3	class4	class5
class1	0	0.823	*0.738*	**0.868**	**0.965**
class2	0	0	*0.814*	*0.729*	0.826
class3	0	0	0	0.821	0.836
class4	0	0	0	0	0.857
class5	0	0	0	0	0

基于Fisher准则的子类划分算法,最终可得子类划分集

$$J = \{\{C_1, C_2, C_3, C_4\}, \{C_5\}\}.$$

基于子类划分构造编码矩阵为

$$M = \begin{Bmatrix} -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & -1 & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{Bmatrix}.$$

比较不同阈值下得到的编码矩阵可知,阈值的取值不同最终将构造出不同的编码矩阵. 不同的编码矩阵应用于分类时,其对应的分类错误率是不同的. 表3给出了阈值取值不同时,对5类样本数据的分类错误率. 由表3可知,当$\alpha = 0.8, \beta = 0.83$时,5类样本数据的分类错误率最小,这是因为基于此阈值得到了最优的子类划分. 而随着样本类别数的增加,编码矩阵的构造对阈值取值的敏感程度也将增加. 因此,如何选择α和β的取值并得到最优的子类划分是自适应构造最优编码矩阵的关键.

表3　5类样本数据的分类错误率　%

α	β			
	0.82	0.83	0.84	0.86
0.73	34.2	30.3	29.6	28.7
0.76	24.3	28.6	19.4	23.9
0.80	16.3	**12.2**	14.6	17.5
0.82	21.5	20.3	19.8	25.1

文献[14]中α和β的取值是根据经验确定的. 但是对于不同的样本数据,阈值的取值很难快速准确地选择,因此,如何自适应地选择阈值,使基于该阈值构造的编码矩阵分类错误率最小,是一个值得研究的问题. 同时,还应注意到当$\alpha < \mathrm{tm}_{ij} < \beta$时,可以认为$C_i$与$C_j$仍然具有很大的相似性,但基于Fisher准则的子类划分算法,将C_i和C_j分开考虑,分别寻找包含这两类的已有的子类划分,相当于认为C_i与C_j的相关性大于阈值β.

针对上述问题,本文对子类划分算法加以改进,提出基于规则的子类划分算法. 该算法针对两类别处于相关与不相关之间的情形,根据已有的可组合类别集和不可组合类别集,判断两类别是否可以组合为一个子类划分,得到更加符合类别分布特点的子类划分结果. 最后,引入PSO算法寻找最优阈值取值,该最优阈值对应的编码矩阵即为最优编码矩阵.

3 基于子类划分和粒子群优化的自适应编码方法

基于数据的编码方法可以看作是从样本数据到编码矩阵的一个映射,文献[16]证明了计算最优编码矩阵是NP难问题. 贪心算法是解决NP难问题、寻找较优解的常用方法. 根据贪心算法的基本思想,在计算问题解的过程中,只要每一步是最优的,则最终得到的解一定是较优解. 本文从样本数据到编码矩阵的映射过程可以描述为从样本数据到子类划分,再由子类划分构造编码矩阵的过程. 因此,对子类划分算法和编码矩阵构造方法的优化是寻找最优编码矩阵的基础. 在此基础上,利用PSO算法对整个映射过程寻优,最终得到最优编码矩阵. 下面从子类划分算法、编码矩阵构造方法和PSO三个方面对本文方法进行描述.

3.1 基于规则的子类划分算法

为了使子类的分类能力最佳,最优的子类划分应该是将相关性较强的类别划分为一个子类,而将相关性较弱的类别划分到不同的子类. 为此,本文基于规则的方法制定若干准则,判断类别间是否相似,从而形成类别的组合方案. 基于规则的子类划分算法描述如下.

基于混淆矩阵得到样本数据类别的相关性度量矩阵TM,并对其进行归一化处理. 给定一组α和β,对TM中的每一个元素按照如下规则对不同类别进行自适应组合,得到子类划分.

规则1 若$\mathrm{tm}_{ij} \leqslant \alpha$,则类别$i$与类别$j$相似,将两者进行组合,得到可组合类别集$G_m = \{C_i, C_j\}$.

规则2 若$\mathrm{tm}_{ij} \geqslant \beta$,则类别$i$与类别$j$相异,两类别相关性弱,不能组合到一起,从而得到不可组合类别集$G'_n = \{C_i, C_j\}$.

规则3 对于任意两个可组合类别集G_{m1}和G_{m2},若$G_{m1} \bigcap G_{m2} \neq \varnothing$,则将二者合并为一个可组合类别集.

规则4 若$\alpha < \mathrm{tm}_{ij} < \beta$,即两种类别处于相关与不相关之间,则类别组合时需根据已有的可组合类别集G和不可组合类别集G'进行如下考虑:

规则4.1 若$\alpha < \mathrm{tm}_{ij} < \beta$,且同时存在一个可组合类别集$G = \{C_i, C_k\}$和一个不可组合类别集$G' = \{C_j, C_k\}$,则类别$j$不能添加进可组合类别集$G$中;

规则4.2 若$\alpha < \mathrm{tm}_{ij}, \mathrm{tm}_{ik} < \beta$,且同时存在可组合类别集$G_1 = \{C_j, C_k\}, G_2 = \{C_i, C_h\}$,则只有当$\mathrm{tm}_{hj}$、$\mathrm{tm}_{hk}$均小于$\beta$时,$G_1$和$G_2$方可进行组合.

最终得到的子类划分为可组合类别集G和不可组合类别集G'.

3.2 编码矩阵构造方法

自适应编码矩阵的构造还要求使子类集成的纠错能力最佳. 分析现有的编码方法可以看到:“一对多”编码具有基分类器个数少、计算复杂度低的优点,但存在冗余信息较少,纠错能力较差的问题,适合于类别可分性较好的数据分类;“一对一”编码基分类器个数多,是所有编码方法中提供冗余信息最多的一种编码方法,具有较强的纠错能力,但计算复杂度高,适合于类别数较少的数据分类. 因此,将“一对多”与“一对一”编码方法相结合,可以在保证计算复杂度较低的同时,具有较强的纠错能力.

对上节得到的子类划分,采用“一对一”与“一对多”相结合的编码方法构造编码矩阵. 根据子类划分算法,可组合类别集中各类别的相关性较强,从直观上看各类别的数据分布存在较大的交叉和重叠,数据的可分性较差. 因此,对于可组合类别集采用“一对一”的编码方法,可以提高分类器的纠错能力. 对于不可组合类别集,其各类别的相关性较弱,直观上看各类别数据分布相对独立,数据的可分性好. 因此,对于不可组合类别集采用“一对多”的编码方法,可以降低计算复杂度.

3.3 粒子群优化

给定一组α和β,由上节基于规则的子类划分算法和编码矩阵构造方法,可以得到对应的编码矩阵. 下面引入PSO算法对α和β取值进行优化,最终得到最优取值对应的最优编码矩阵.

粒子群源于对鸟群捕食行为的研究,基于迭代的方法寻找最优解是群体智能算法的典型代表. 如果每个粒子的邻域是全部粒子,则称为全局粒子群算法. 文献[17]应用全局粒子群算法在解空间中搜索最优的阈值取值.

PSO算法的主要步骤是: 1) 确定解空间; 2) 进行随机初始化; 3) 迭代搜索解空间,更新极值找到满足迭代条件的最优解. 在算法的迭代过程中,极值的更新包括两个方面:一是个体最优极值(y_i),是粒子本身所找到的最优解;二是全局最优极值($\hat{y}$),是整个种群历史上找到的最优解.

设$p_i(t)$为第i个粒子t时刻在解空间的位置. 粒子通过增加一个速度分量$v_i(t)$更新到新的位置. 粒子i的位置更新公式为

$$p_i(t+1) = p_i(t) + v_i(t+1). \tag{1}$$

其中: $p_i(0) \propto U(p_{\min}, p_{\max})$, U为均匀分布.

粒子i的速度更新公式为

$$v_i(t+1)=\omega v_i(t)+c_1r_1(t)[y_i(t)-p_i(t)]+ c_2r_2(t)[\hat{y}(t)-p_i(t)]. \tag{2}$$

其中: $v_i(t)$为第i个粒子t时刻的速度, r_1和r_2是[0,1]之间的随机数, ω为惯性因子, c_1和c_2为学习因子.

在PSO算法中, 适应度函数的选择对最优解的获取具有重要作用. 适应度函数衡量每个粒子的适应度, 计算个体极值和全局极值, 通过更新极值, 引导粒子群向最优化的方向运动. 在本文中, 每一个粒子代表一组可能的阈值取值. 对α和β的取值优化的目的是为了得到对应分类错误率最小的编码矩阵. 因此, 本文选取编码矩阵在训练数据集上的分类错误率E作为适应度函数. 设训练数据集为$\Re = \{(s_1,l(s_1)),\cdots,(s_v,l(s_v))\}$, 适应度函数的计算公式为

$$E=\frac{\sum_{j=1}^{v}\sigma(\Delta(M,s_j),l(s_j))}{v}. \tag{3}$$

$$\sigma(m,n)=\begin{cases}0, & m=n;\\ 1, & \text{otherwise};\end{cases} \tag{4}$$

$$\Delta(M,s_j)=\arg\min_i \delta(M_i,H(s_j)),\quad i\in\{1,2,\cdots,N\}. \tag{5}$$

其中: δ为解码方法, N为训练数据集的类别数, M_i为编码矩阵的行向量, $H(s_j)$为各二类分类器的输出向量. 基于PSO的α和β优化方法的具体流程如图3所示.

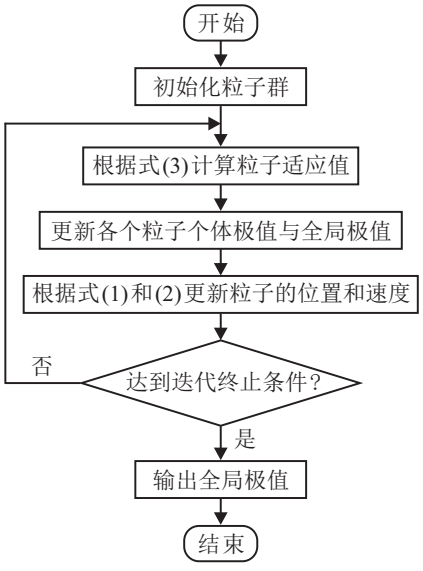

图3　基于PSO的阈值优化流程

由图3可知, PSO算法的终止需要满足迭代终止条件. 一般而言有两种: 一是预先设定最大迭代次数; 二是预先设定理想适应值. 本文采用第1种方法作为终止条件.

至此, 便形成了完整的基于子类划分和PSO的自适应编码方法, 下面将通过实验验证该方法所得编码矩阵在分类中的应用效果.

4　实验结果及分析

采用两种数据集验证本文所提出方法的有效性.

4.1　实验数据

两种数据集为: UCI(University of California Irvine)数据集[18]和五类飞机目标的一维距离像数据集(HRRP). 其中, UCI数据集被普遍用来检验多类分类算法的性能, 表4为UCI数据集及各类数据描述. 同时, 为了提高分类速度, 对UCI数据集中维数较高的数据集利用主成分分析方法进行降维处理. 五类飞机目标的HRRP数据集为实验室条件下得到的五类不同机型的雷达回波数据, 对其进行分类对于提高雷达的目标识别能力具有重要意义.

表4　UCI数据集各数据描述

数据名称	样本数	特征维数	类别数
Yeast	1 484	8	10
Segmentation	2 310	19	7
Satimage	6 435	36	6
Glass	214	9	7
Zoo	101	18	7
Wine	178	13	3
Vowel	990	10	11
Iris	150	4	3
Vehicle	846	18	4

4.2　实验设计

首先, 利用UCI数据集检验基于子类划分和PSO的自适应编码矩阵的分类性能. 分类结果分别与“一对一”编码、“一对多”编码、密集随机编码、稀疏随机编码、判别式编码[5]、神经网络编码[15]以及混淆矩阵编码[14]进行比较. 解码策略选择为Hamming距离解码和欧氏距离解码. 基分类器为支持向量机(SVM), 核函数为径向基核函数. 实验中, α和β的取值范围设定为相关性度量矩阵非零元素的最大值和最小值. 同时, 考虑到α和β具有依赖关系, 即$\alpha\leqslant\beta$, 因此, 对不满足这种依赖关系和超出取值范围的粒子, 将其适应度函数设置为一个较大的值, 从而促使粒子向最优解的方向运动. 最大迭代次数设置为100.

然后, 用五类飞机的HRRP数据集检验本文所提出方法在实际应用领域的分类效果. 实验中, 分别提取3个不同角度范围(0° ~ 100°, 80° ~ 150°和0° ~ 150°)的HRRP进行识别, 选择汉明距离解码作为解

码策略.

在构造混淆矩阵时,选择决策树作为预分类器.为保证估计分类正确率时的准确性,样本数据个数大于500时采用10重交叉验证,小于500时采用5重交叉验证;同时,利用双边估计t检验法计算置信水平为0.95的分类正确率置信区间,将其作为最终结果,其计算公式为

$$\frac{|\bar{x}-u|}{\sigma/\sqrt{n}} \geqslant t_{0.025}(n-1). \tag{6}$$

其中:u和σ分别表示n重交叉验证的均值和标准差,$t_{0.025}(4)=2.776\,4, t_{0.025}(9)=2.262\,2$.

4.3 实验结果与分析

4.3.1 UCI数据集

各编码矩阵分类正确率的结果见表5和表6. 表

表5 基于Hamming距离解码及SVM作为基分类器的置信区间为95%的各编码矩阵分类正确率 %

数据名称	一对多	一对一	密集随机编码	稀疏随机编码	判别式编码	神经网络编码	混淆矩阵编码	PSO编码
Yeast	23.86±3.25	**58.43±3.61**	29.84±5.10	47.04±2.93	32.02±4.43	44.68±3.46	52.09±3.12	**56.40±2.05**
	10 × 10	**10 × 45**	10 × 10	10 × 10	10 × 9	10 × 18	10 × 12	**10 × 12**
Segment	73.94±1.39	**94.16±1.87**	77.36±2.44	56.80±1.23	77.57±1.27	77.57±0.58	86.93±1.40	**92.16±0.54**
	7 × 7	**7 × 21**	7 × 7	7 × 7	7 × 6	7 × 12	7 × 7	**7 × 10**
Satimage	68.24±0.82	**85.63±0.52**	76.46±0.87	60.79±1.10	**84.03±0.24**	81.89±1.87	83.15±0.82	83.22±0.91
	6 × 6	**6 × 15**	6 × 6	6 × 6	6 × 5	6 × 10	6 × 8	6 × 8
Glass	60.87±4.53	**94.03±4.04**	66.02±7.07	49.51±3.93	71.45±3.08	86.42±1.32	71.18±7.01	**88.98±8.41**
	6 × 6	**6 × 15**	6 × 6	6 × 6	6 × 5	6 × 10	6 × 8	**6 × 8**
Zoo	75.33±17.21	**78.14±12.27**	68.38±16.45	35.71±30.02	69.36±10.42	68.27±13.02	67.29±10.58	**76.24±7.76**
	7 × 7	**7 × 21**	7 × 7	7 × 7	7 × 6	7 × 12	7 × 8	**7 × 8**
Wine	98.28±1.95	**98.32±3.10**	96.65±2.88	97.75±2.90	93.30±1.42	96.03±0.84	**97.71±2.97**	97.21±2.51
	3 × 3	**3 × 3**	3 × 3	3 × 3	3 × 2	3 × 4	**3 × 3**	3 × 3
Vowel	13.23±2.15	**76.06±5.13**	30.00±3.86	27.58±3.78	31.31±3.02	**67.57±1.46**	39.60±2.96	66.87±2.83
	11 × 11	**11 × 55**	11 × 11	11 × 11	11 × 10	**11 × 20**	11 × 14	11 × 15
Iris	74.67±2.27	96.67±4.14	74.00±11.48	79.33±4.53	97.33±2.14	97.33±3.47	**98.00±3.70**	97.33±3.46
	3 × 3	3 × 3	3 × 3	3 × 3	3 × 2	3 × 4	**3 × 3**	3 × 3
Vehicle	67.61±1.73	**80.50±1.95**	68.35±7.38	62.88±6.05	74.23±2.45	76.92±1.12	75.63±3.23	**79.54±2.59**
	4 × 4	**4 × 6**	4 × 4	4 × 4	4 × 3	4 × 6	4 × 5	**4 × 5**

表6 基于欧氏距离解码及SVM作为基分类器的置信区间为95%的各编码矩阵分类正确率 %

数据名称	一对多	一对一	密集随机编码	稀疏随机编码	判别式编码	神经网络编码	混淆矩阵编码	PSO编码
Yeast	31.33±2.00	**57.40±3.85**	45.35±1.17	51.01±3.08	54.92±2.42	53.44±2.71	49.19±1.30	**55.54±0.20**
	10 × 10	**10 × 45**	10 × 10	10 × 10	10 × 9	10 × 18	10 × 12	**10 × 12**
Segment	74.50±1.65	**94.42±1.87**	72.77±0.86	60.48±1.66	80.95±1.51	80.95±1.56	85.37±1.91	**92.30±1.36**
	7 × 7	**7 × 21**	7 × 7	7 × 7	7 × 6	7 × 12	7 × 7	**7 × 10**
Satimage	68.33±0.59	**85.72±0.79**	64.02±0.87	78.51±0.48	83.36±0.73	81.96±0.95	83.28±1.13	**84.92±0.96**
	6 × 6	**6 × 15**	6 × 6	6 × 6	6 × 5	6 × 10	6 × 8	**6 × 8**
Glass	60.70±2.80	**93.57±3.96**	75.37±8.42	79.04±7.83	71.80±11.34	**88.74±6.30**	69.52±8.01	81.24±8.02
	6 × 6	**6 × 15**	6 × 6	6 × 6	6 × 5	**6 × 10**	6 × 8	6 × 8
Zoo	79.19±17.81	72.48±20.87	66.10±23.17	64.43±12.18	73.45±12.66	77.36±10.92	**80.19±7.62**	**83.38±10.45**
	7 × 7	7 × 21	7 × 7	7 × 7	7 × 6	7 × 12	**7 × 8**	**7 × 8**
Wine	97.74±1.57	98.33±3.04	**98.35±1.87**	97.17±4.98	96.53±3.54	95.57±3.50	97.19±3.50	**98.30±1.93**
	3 × 3	3 × 3	**3 × 3**	3 × 3	3 × 2	3 × 4	3 × 3	**3 × 3**
Vowel	9.19±0.28	**76.16±3.47**	30.81±0.30	29.19±3.02	36.57±2.28	66.36±2.13	38.18±1.14	**69.39±5.20**
	11 × 11	**11 × 55**	11 × 11	11 × 11	11 × 10	11 × 20	11 × 13	**11 × 14**
Iris	75.33±9.53	**9**8.00±**5**.55	74.00±6.14	75.33±6.93	95.33±4.52	95.33±4.52	**97.33±3**.46	96.67±5.07
	3 × 3	**3 × 3**	3 × 3	3 × 3	3 × 2	3 × 4	**3 × 3**	3 × 3
Vehicle	66.66±2.48	**79.68±2.65**	65.96±4.20	65.47±6.15	70.95±3.78	76.85±3.33	75.31±3.83	**77.55±2.89**
	4 × 4	**4 × 6**	4 × 4	4 × 4	4 × 3	4 × 6	4 × 5	**4 × 4**

5和表6中黑体表示分类正确率排名前两位的编码方法. 从表中可以看出,“一对一”编码方法在大多数数据集上均取得了最好的分类结果. 这与文献[12]的实验结果是一致的,这是因为“一对一”编码可以获得更多的冗余分类信息,具有较强的纠错能力. 基于数据的编码的目标就是用更少的编码长度实现接近于“一对一”编码的分类性能. 同时可以看出,本文方法通过优化阈值取值和子类划分算法,与其他编码方法相比,可以有效提高分类性能,从而验证了本文方法的有效性.

4.3.2 HRRP数据集

3种不同角度范围下各编码方法的分类结果如表7～表9所示. 表中每个分类正确率下的值均是置信度为95%的置信区间值. 从表中数据可以看出,与其他编码方法相比,在面对实际分类问题时,本文提出的编码方法具有更好的实用效果.

表7　角度为0°～100°时HRRP分类正确率　%

飞机类型	一对多	一对一	密集随机编码	稀疏随机编码	判别式编码	神经网络编码	混淆矩阵编码	PSO编码
B-52	48.50	**97.63**	95.40	**97.60**	66.53	70.12	54.10	77.90
	4.11	**1.55**	0.97	**1.13**	2.34	1.23	3.72	3.14
J-6	40.90	**98.76**	40.50	41.30	44.15	51.26	60.65	**85.47**
	3.15	**2.12**	4.97	4.54	3.26	2.41	1.93	**1.04**
F-117	97.80	98.10	97.70	**98.20**	96.56	98.02	97.60	**98.99**
	1.00	1.86	1.17	**1.25**	0.78	1.54	0.50	**0.33**
F-16	97.40	**98.32**	96.60	86.50	88.26	88.54	97.20	**97.59**
	1.40	**1.26**	1.59	2.37	1.59	2.03	0.81	**0.59**
FY-2000	59.30	**98.23**	66.60	63.40	44.33	70.46	77.19	**89.94**
	3.98	**2.43**	3.41	1.31	3.69	3.41	1.94	**1.57**

表8　角度为80°～150°时HRRP分类正确率　%

飞机类型	一对多	一对一	密集随机编码	稀疏随机编码	判别式编码	神经网络编码	混淆矩阵编码	PSO编码
B-52	96.86	**98.60**	95.29	95.86	76.25	95.23	97.57	**97.70**
	1.72	**1.25**	1.67	1.40	1.02	1.46	1.08	**0.68**
J-6	90.86	99.23	93.57	**99.86**	94.53	**98.66**	91.71	98.51
	2.78	2.54	2.82	**0.32**	1.45	**2.01**	1.65	1.58
F-117	**98.86**	98.42	92.61	96.67	96.01	94.12	97.19	**98.07**
	1.31	2.02	1.87	1.94	1.86	0.78	0.94	**0.63**
F-16	95.14	**99.42**	97.14	87.71	90.43	95.33	96.57	**97.58**
	2.42	**2.16**	1.45	3.98	1.89	2.03	1.61	**1.42**
FY-2000	65.71	**99.03**	82.29	33.86	47.02	59.23	65.86	**89.79**
	5.84	**3.84**	3.77	4.70	3.62	3.12	4.88	**1.25**

表9　角度为0°～150°时HRRP分类正确率　%

飞机类型	一对多	一对一	密集随机编码	稀疏随机编码	判别式编码	神经网络编码	混淆矩阵编码	PSO编码
B-52	34.67	91.60	**97.47**	45.60	38.56	59.563	32.73	**81.90**
	2.00	2.55	**0.74**	3.41	4.12	3.10	2.97	**3.32**
J-6	18.70	**94.70**	15.60	18.81	13.26	27.15	17.87	**28.71**
	2.28	**1.12**	3.26	1.46	2.02	2.14	2.56	**3.79**
F-117	97.47	98.27	91.73	63.87	90.43	78.98	**98.87**	**98.99**
	0.38	1.06	2.19	2.50	0.56	1.35	**0.68**	**1.55**
F-16	88.80	**96.82**	94.40	94.53	91.74	90.18	86.93	**95.80**
	1.12	**1.18**	1.28	0.95	0.33	1.26	2.04	**1.85**
FY-2 000	22.20	94.12	22.40	98.27	77.32	**98.36**	**99.93**	88.40
	1.37	1.86	2.43	1.06	2.16	**3.25**	**0.15**	2.43

5 结 论

基于混淆矩阵的自适应编码方法根据相似阈值和相异阈值获得子类划分,自适应构造编码矩阵. 针对相似阈值和相异阈值根据经验设定带来的问题,本文提出了利用PSO方法寻找最优的阈值取值,进而形成了更加完善的基于子类划分和PSO的自适应编码方法. 利用UCI数据集和HRRP数据集分别对其进行实验,所得结果表明了本文方法能显著地提高对多类问题的最终分类效果.

参考文献(References)

[1] Sergio E, David M. Online error correcting output codes[J]. Pattern Recognition Letters, 2011, 32(3): 458-467.

[2] Omid D, Bin M. Discriminative feature extraction for speech recognition using continuous output codes[J]. Pattern Recognition Letters, 2012, 33(13): 1703-1709.

[3] Sergio E, Oriol P, Josepa M. IVUS tissue characterization with subclass error-correcting output codes[J]. Computer Vision and Pattern Recognition, 2008, 34(5): 1-8.

[4] Sergio E, Oriol P, Petia R. Re-coding ECOCs without retraining[J]. Pattern Recognition Letters, 2010, 31(5): 555-562.

[5] Oriol P, Petia R, Jordi V. Discriminate ECOC: A heuristic method for application dependent design of error correcting output codes[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2006, 28(6): 1007-1012.

[6] Sergio E, David M, Oriol P, et al. Subclass problem-dependent design for error-correcting output codes[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2008, 30(6): 1041-1054.

[7] Zhong Guoqiang, Huang Kaizhu, Liu Chenglin. Joint learning of error-correcting output codes and dichotomizers from data[J]. Neural Computing and Application, 2012, 21(4): 715-724.

[8] Zhong Guoqiang, Huang Kaizhu, Liu Chenglin. Learning ECOC and dichotomizers jointly from data[J]. Lecture Notes in Computer Science, 2010, 6443(3): 494-502.

[9] Mohammad A B, Gholam A M, Ehsanollah K. A subspace approach to error correcting output codes[J]. Pattern Recognition Letters, 2013, 34(1): 176-184.

[10] Wang Yunyun, Chen Songcan, Xue Hui. Can under-exploited structure of original-classes help ECOC-based multi-class classification[J]. Neurocomputing, 2012, 89(15): 158-167.

[11] Mohammad A B, Qigang G, Sergio E. A genetic-based subspace analysis method for improving error-correcting output coding[J]. Pattern Recognition, 2013(46): 2830-2839.

[12] Miguel A B, Sergio E, Xavier B, et al. On the design of an ECOC-compliant genetic algorithm[J]. Pattern Recognition, 2014(47): 865-884.

[13] Nicolas G P, Colin F. Evolving output codes for multiclass problems[J]. IEEE Trans on Evolutionary Computation, 2008, 12(1): 93-106.

[14] 周进登,王晓丹,周红建. 基于混淆矩阵的自适应纠错输出编码多类分类方法[J]. 系统工程与电子技术, 2012, 34(7): 1518-1524.
(Zhou J D, Wang X D, Zhou H J. Multiclass classification of adaptive error-correcting out-put codes based on confusion matrix[J]. Systems Engineering and Electronics, 2012, 34(7): 1518-1524.)

[15] Zhou Jindeng, Wang Xiaodan, Zhou Hongjian, et al. Coding design for error correcting output codes based on perception[J]. Optical Engineering, 2012, 51(5): 1-16.

[16] Koby C, Yoram S. On the learnability and design of output codes for multiclass problems[J]. Machine Learning, 2002, 47(2): 201-233.

[17] Dimitrios B, Nikolaos A, Anastasios T. Optimizing linear discriminant error correcting output codes using particle swarm optimization[J]. Lecture Notes in Computer Science, 2011, 6792(1): 79-86.

[18] Asuncion A, Newman D. UCI machine learning repository[D]. Irvine: School of Information and Computer Sciences, University of California, 2007.

（责任编辑：李君玲）