文章编号: 1001-0920 (2011) 09-1295-08

基于最小 k 近邻错分率编码确定方法及其在多类分类中的应用

周进登[1,2], 王晓丹[1]

(1. 空军工程大学 导弹学院，陕西 三原 713800；2. 94719部队，江西 吉安 343700)

摘 要: 构造输出编码矩阵是将多类分类问题分解为多个两类分类问题的有效方法之一, 如何判断一个编码阵的好坏是此类问题的关键. 提出以最小 k 近邻错分率作为评价标准, 把构造问题简化为一个搜索问题. 在 M 类的所有二类划分空间中, 通过行交换规则和有限启发式搜索策略搜索出 k 近邻错分率最小的 ℓ 个二类划分, 并依据编码规则得到最终输出编码矩阵. 实验中用人工数据集和UCI数据集分别测试, 通过与几种经典的编码方法比较, 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；k 近邻；输出编码；启发式搜索

中图分类号: TP181 **文献标识码:** A

Designing of output codes based on minimal k nearest neighbor classifying error and its application in multi-class classification

ZHOU Jin-deng[1,2], *WANG Xiao-dan*[1]

(1. Missile Institute，Air Force Engineering University，Sanyuan 713800，China；2. Air Force 94719，Ji'an 343700, China．Correspondent：ZHOU Jin-deng，E-mail：zhoujin198417@yahoo.com.cn)

Abstract: Generating the matrix of output codes is one of the most effective methods to reduce multiclass to binary. How to establish the effect of coding matrix is the key to solve this problem. Therefore, the k-nearest neighbor classifying error rateis taken as a metewand, which can translate construction issue into searching problem. The matrix of standard output codes is generated by using ℓ binary partitions based on row-exchange rule and limited heuristic search strategy in the space of M classes. Experimental results on artificial data and UCI with logistic linear classifier(loglc) and support vector machine(SVM) as the binary learner show that the proposed method has better performance of classification with shorter length of coding matrix.

Key words: multi-class classification；k-nearest neighbor；output codes；heuristic search

1 引 言

多类分类是机器学习的一个重要研究领域. 目前, 解决多类问题的方法大致分为两种: 1) 直接构造多类分类器, 对待解决问题进行求解; 2) 把多类问题转化为二类问题, 结合目前在两类分类领域所取得的研究成果, 设计一种映射模式, 从而达到对多类进行分类的目的. 从现有的研究趋势看, 基于输出编码的映射方式被认为是能有效提高分类准确率的有效方法之一.

Dietterich 等人于 1995 年提出了纠错编码ECOC (error-correcting codes) 解决多类分类[1], 它属于第2种多类分类方法, 其主要包括训练和分类两个步骤. 在训练阶段通过构造编码矩阵把多类问题转换成多个两类分类问题; 在分类阶段将由训练获得的一组两类分类器对样本进行分类, 得到一个结果向量, 通过基于Hamming 距离解码对该向量进行判定. Hastie 等人[2]于 1998 年提出了一对余类编码方法, Allwein 等人[3]于 2000 年提出了成对分类编码方法, 这两种方法都可以看成是ECOC 的两个特例, 其Hamming 距离分别为 2 和 $\left(\binom{M}{2}-1\right)/2$ (M 为类别数)[3]. 此外还有随机编码[4-5]、无遗编码[1,3]. Crammer 等人[6]在对现有的基于输出编码多类分类的问题分析和总结后, 提出了多类分类编码的 3 个问题:

1) 给定编码矩阵, 如何找到使分类错误率最小的一组二类分类器;

2) 给定一组二类分类器, 如何确定是分类错误率

收稿日期: 2010-05-17；**修回日期:** 2010-08-17.

基金项目: 国家自然科学基金项目(60975026)；博士学位论文创新基金项目(DY11102).

作者简介: 周进登(1984−), 男, 博士生, 从事智能信息处理、机器学习的研究；王晓丹(1966−), 女, 教授，博士生导师, 从事智能信息处理、机器学习研究.

最小的编码矩阵;

3) 怎样同时确定输出编码矩阵和与之相对应的一组二类分类器使分类错误率最小.

从现有的研究成果看, 目前绝大部分是针对问题 1) 进行的[1-3,5,7]. 然而, 现实分类问题中判断待分类的数据适合用哪一种编码矩阵具有很大的主观性, 缺乏定性的判断; 加之现有的这些常用编码矩阵不能满足所有类型数据分类, 往往会得出与真实分类结果有较大差异的结论. 随着二类分类器研究的深入, 特别是基于结构风险最小化的支持向量机 (SVM) 的提出, 使得任何数据都可以找到与之对应的二类分类函数, 即测试样本总能找到一个 N 维空间, 在这 N 维空间中线性可分. 为此, 如果能很好地解决问题 2), 则多类分类问题将被完美地解决. 遗憾的是问题 2) 被证明是一个 NPC 问题[6]. 虽然有很多学者针对问题 2) 提出了改进方案[6,8], 但它们不是过于复杂难以实用化, 就是约束条件太苛刻, 减弱了其泛化能力. 因此可将目光集中到问题 3) 的解决, 问题 3) 表面上看似更复杂, 但由问题 2) 的研究可知, 任何数据在高维空间里总是线性可分的, 且通过定义适当的核函数可以实现这种可分性. 这时问题 3) 可以简化成针对待分类数据按一定标准找出一组最利于线性分类器分类的划分, 即确定输出编码矩阵. 此类方法可以归结为基于数据的编码方法. 文献 [9] 提出了类似的想法, 它通过构建二叉树实现了对数据的二类划分, 并根据类别间的交互信息 (MI) 确定了划分的顺序. 文献 [10] 对此方法进行了改进.

对一组 M 类数据, 其所有二类划分共有 $2^{M-1}-1$ 个, 从这些划分中找出一组满足要求的划分并组成一个编码组是本文研究的核心. 本文提出了一种行交换 (RE) 规则来解决空间复杂度问题, 并探索了一种有限启发式搜索策略 (LHSS), 以减少时间复杂度.

2 输出编码设计

在解决问题 3) 时, 本文提出了一种基于最近邻域多类编码设计, 在对待分类数据进行二类划分 (即编码) 时基于以下两点假设:

假设 1 一组编码 (编码矩阵的一列) 对应一个划分.

假设 2 好的划分能使待分类样本及其邻域内的样本更容易归为同类.

如何判定一个划分好坏将是本文研究的重点, 这个判定标准应尽可能简单直观, 这样才能在分类类别数较大时更快更准确地进行判定. 本文利用文献 [11] 提出的最小化 k 邻域错分率作为判定标准, 以达到快速判定的目的.

对一个 M 类问题设计输出编码矩阵 H, 假设有 ℓ 个二类划分, 则 H 为 $M\times\ell$ 矩阵, 本文对 H 矩阵有如下设计要求[1]:

1) 行分离. 每一行只代表一类, 当考虑 H 的纠错能力时, 应尽可能地使这种分离性最大, 即使得两行的 Hamming 距离最大, 以便 H 具有更强的纠错能力.

2) 列分离. 每一列只代表一种二类划分, 故列与列之间不能相同或相似, 也不能互补. 相似的两列被认为具有相似的划分, 据此训练出来的二类分类器也将具有相似的分类能力, 产生的错误率将叠加. 互补的两列被认为是同一种划分.

以上是输出编码矩阵设计的两条基本原则, 事实上以往研究设计输出编码矩阵时经常利用这两条基本原则作为设计目标, 进而设计出行列分离性较大的编码矩阵, ECOC 编码的设计即是最大化行分离性, 使编码阵的最小 Hamming 距离最大, 使纠错能力更强. 在实际应用中设计输出编码矩阵是为了减少多类分类的复杂性, 利用性能良好的二类分类器作为基分类器, 并由编码阵给定的二类划分进行训练, 如果训练错误率减小, 则分类器的准确率将提高, 最终使得分类错误率最小. 因此, 在设计编码阵时应考虑以训练错误率最小为目标, 方能达到设计要求. 对于输出编码阵与训练错误率的关系, 文献 [3] 进行了深入的研究, 并得出了以下结论: 一个输出编码阵的训练错误率是其基于二类分类器组平均训练错误率 $\bar{\varepsilon}$ 的 ℓ/ρ 倍, ℓ 为编码阵的列数, ρ 为编码阵的最小 Hamming 距离, $\bar{\varepsilon}$ 的计算公式如下:

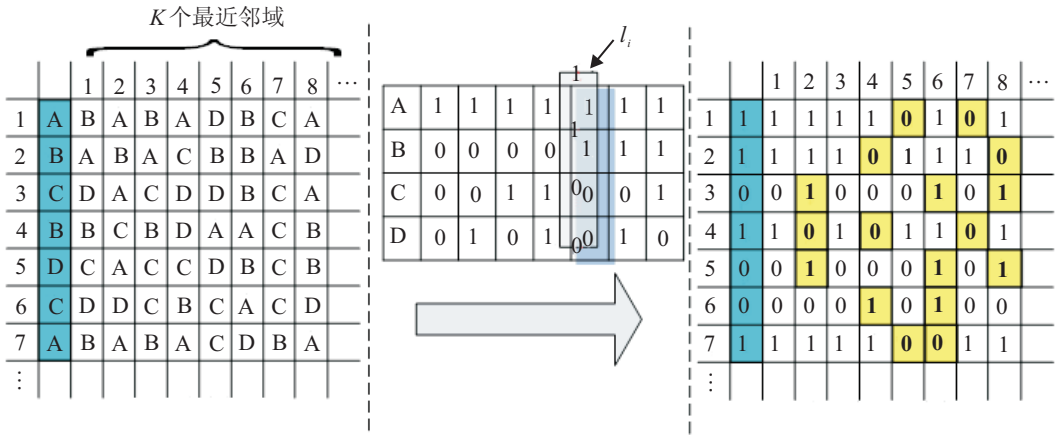
$$\bar{\varepsilon}=\frac{1}{m\ell}\sum_{i=1}^{m}\sum_{s=1}^{\ell}L\left(H\left(y_i,s\right)\ f_s\left(x_i\right)\right). \tag{1}$$

其中: m 为样本数目, H 为编码阵, f 为二分类器. 上述结果表明, 当编码阵的列数 ℓ 很大时 ($\ell\gg\rho$), 其训练错误率也很大. 本文在设计输出编码阵时为简化问题的复杂性暂不考虑 ρ 的大小, 但可通过设计尽可能小的 $\ell(\log_2 M\leqslant\ell\leqslant 2^{M-1}-1)$ 来弥补这种缺憾. 实验中取 $\ell=\max(M,10\times\log_2(M)/3)$.

3 基于最近邻域多类编码设计

3.1 最近邻域评价准则

如何对假设 2 做定量分析是本节研究的主要内容, 为此引进最近邻域 kNN 的概念, 并提出以最小化 k 近邻错分率作为两类划分的评价标准. 参考文献 [11], 具体做法如下: 随机取出 m' 个训练样本 (当样本数不是特别大时, 取 $m'=m$, 即全部样本数, 为方便说明本文均取全部样本数), 对每个样本分别求取 k 个最近邻域样本, 并将其按图 1(a) 所示排列, 深颜色标示为所有训练样本列, 用输出编码矩阵的每一列对其做一次映射. 如图 1(b) 用编码阵的第 5 列 $l_5=[1\ 1\ 0\ 0]'$ 对图 1(a) 做一次映射得到图 1(c),



(a)　编码前类标签　　(b)　输出编码阵　　(c)　按l_i编码后类标签

图 1　kNN 映射图

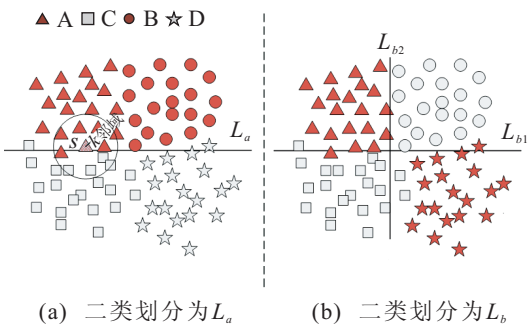
则 l_5 列的 k 邻域错分率 ε'_5 可由下式计算得到:

$$\varepsilon'_j = \frac{1}{m}\sum_{i=1}^{m} I(f_j(x_i) \neq c_{y_i j}),\ j = 1, 2, \cdots, \ell. \tag{2}$$

其中: $I(\cdot)$ 为符号函数, m 为样本数, $I(\cdot) = 1$ 对应图 1(c) 带阴影黑色字体标示的块. ε'_j 越小, 说明用 ℓ_i 对应的二类划分使得训练样本 k 近邻错分率越小. 注意到这里的错分是指每个样本的 k 近邻样本与其本身的类标签不同, 这也符合假设 2 的规定. 得到 ℓ_i 的 k 近邻错分率后, 整个输出编码阵的 k 近邻错分率为

$$\bar{\varepsilon}' = \frac{1}{\ell}\sum_{i=1}^{\ell} \varepsilon'_i. \tag{3}$$

由上可以看出, 这种方法具有简单快速的优点, 特别当类别数目较大时, 能有效简化计算的复杂性且保持较高的准确率. 下面将讨论这种方法的合理性. 由图 2 可以看出, 每组训练样本的 k 近邻错分率都是由处于类别边缘的样本所确定的, 图 2(a) 中的样本 s, 其 k 近邻共包含 10 个样本, 其中正类样本 6 个, 负类样本 4 个, 样本 s 本身为正类样本. 根据 k 近邻错分率公式计算可得 s 的 k 近邻错分率为 0.4. 图 2(a) 对应的二类划分为 $\ell_a = (1\ 1\ 0\ 0)'$, 其类边缘样本数由直线 L_a 确定; 图 2(b) 对应的二类划分为 $\ell_b = (1\ 0\ 0\ 1)'$, 其类边缘样本数由直线 L_{b1} 和 L_{b2} 确定. 可以看出, ℓ_a 划分的 k 近邻错分率 ε'_{ℓ_a} 要小于 ℓ_b 划分的 k 近邻错分率 ε'_{ℓ_b} (事实上 $\varepsilon'_{\ell_a} \approx \varepsilon'_{\ell_b}/2$), 故 ℓ_a 划分要好于 ℓ_b 划分.



(a)　二类划分为L_a　　(b)　二类划分为L_b

图 2　k近邻错分示意图

3.2　行交换规则

在对二类划分优劣的评价标准确定之后, 下一步即要找出所有待评价的划分组, 根据前文所述一个 M 类数据组, 其所有二类划分共有 $2^{M-1}-1$ 个, 当 M 增加时, 划分数将以指数级增长, 要找出最优解就必须穷尽所有划分, 这在实际应用中是难以实现的, 首先存储这些划分需要消耗大量内存, 其次从中搜索最优的 ℓ (编码矩阵的列数) 个划分的算法时间复杂度为 $O(2^M)$. 在此不可能列举出所有划分, 但可以通过一个简单的行交换找出所有的划分.

下面证明这种行交换的可行性, 假设 $M = 4$, 其无遗编码矩阵 S 如表 1 所示, 初始编码矩阵如表 2 所示.

表 1　类别数为 4 的无遗编码矩阵

Row	Column						
	1	2	3	4	5	6	7
1	**1**	**1**	**1**	1	1	1	1
2	**0**	**0**	**0**	0	1	1	1
3	**0**	**0**	**1**	1	0	0	1
4	**0**	**1**	**0**	1	0	1	0

表 2　初始编码矩阵

Class	Cord Word		
	f_1	f_2	f_3
C_1	**1**	**1**	**1**
C_2	**0**	**0**	**0**
C_3	0	0	1
C_4	0	1	0

从表 2 可以看出, 该编码阵取无遗编码的前 3 列, 无遗编码阵包含了所有二类划分. 现进行行交换 (如表 3 所示): $C_1 \Leftrightarrow C_2$, 得到 f'_1, f'_2, f'_3 三个新的二类划分, 可以看出 $f'_2 = f_3$, $f'_3 = f_2$, 得到 $S(4)$, 即 $C_1 \Leftrightarrow C_3$, 见表 4. 由此得到 f''_1, f''_2, f''_3 三个新的二类划分, 可以看出 $f''_3 = f'_2 = f_3$, 得到 $S(5)$ 和 $S(6)$, 即 $C_1 \Leftrightarrow C_4$, 见

表5. 由此得到 f_1', f_2', f_3' 三个新的二类划分, 可以看出 $f_2''' = f_3' = f_2$, $f_3''' = f_2''$, 得到 $S(7)$.

表 3 第 1 和第 2 行交换后的编码阵

Class	Cord Word		
	f_1'	f_2'	f_3'
C_1'	**0**	**0**	**0**
C_2'	**1**	**1**	**1**
C_3'	0	0	1
C_4'	0	1	0

表 4 第 1 和第 3 行交换后的编码阵

Class	Cord Word		
	f_1''	f_2''	f_3''
C_1''	**0**	**0**	**1**
C_2''	0	0	0
C_3''	**1**	**1**	**1**
C_4''	0	1	0

表 5 第 1 和第 4 行交换后的编码阵

Class	Cord Word		
	f_1'''	f_2'''	f_3'''
C_1'''	**0**	**1**	**0**
C_2'''	0	0	0
C_3'''	0	0	1
C_4'''	**1**	**1**	**1**

至此, 通过3次行交换遍历了 $M = 4$ 时其无遗编码矩阵 S 的所有列 $S(i)\,(i = 1, 2, \cdots, 7)$. 事实上只要初始编码矩阵设计合理, 通过行交换总能遍历其无遗编码矩阵的所有列, 因为 $M! > 2^{M-1} - 1 (M \geqslant 2)$.

需要注意的是初始编码阵的选择, 由上述分析可知, 当 $M = 4$ 时其所有二类划分可归纳为2种, 即 $\{k_1, k_2 | k_1 : (1^+, 3^-), k_2 : (2^+, 2^-)\}$, k_1 代表1个正类和3个负类, k_2 代表2个正类和2个负类. 表6列出了几种类别数的二类划分的种类. 要保证能遍历完所有的二类划分, 初始化编码阵必须包含所有种类的二类划分. 如表6所示, 当 $M = 5$ 时初始编码阵应包含2列: $\{\ell_1, \ell_2 | \ell_1 \in k_1, \ell_2 \in k_2\}$.

3.3 有限启发式搜索

在解决了空间复杂度问题之后, 如何利用3.1节提出的最小化 k 近邻错分率准则搜索出最好的 ℓ 个二类划分并组成输出编码阵, 是本文又一个亟待解决的问题. 在此提出有限启发式搜索策略(LHSS), 并与文献[11]提出的方法做比较, 以说明其优点. 下面将对该搜索问题的特点进行分析.

M 类数据组共有 $2^{M-1} - 1$ 个二类划分, 当 M 不是很大时可用穷尽法搜索策略找出符合要求的 ℓ 个二类划分; 当 M 较大(一般 $M > 7$)时, 由于这种搜索算法的时间复杂度为 $O(2^M)$, 此时不能利用这种方法进行求解. 如何找出一个时间复杂度为多项式且能准确地搜索出最优二类划分组, 是一个有待更加深入研究的问题. 参考文献[10]提出了一个近似解决方案, 并称之为有限启发式搜索, "有限"的原因是该方法需要事先给定一个常用的输出编码矩阵(如一对一编码阵、随机编码阵等), 并认为基于这些编码阵的行交换能搜索出性能更好的二类划分. 该搜索算法依赖初始编码阵选择恰当与否, 启发的程度有限. 事实上, 由3.2节可知, 要获得最优划分应尽可能搜索更多的划分空间, 故初始编码阵必须包含更全的二类划分种类. 本文采用的初始化编码阵为随机编码阵, 随机编码矩阵分两种: dense random codes 和 sparse random codes, 本文取第1种. 构造过程是对编码阵的每一个元素都随机地从 $\{-1, +1\}$ 中选取, 其最小Hamming距离为 $\ell/2$. 文献[4]中 ℓ 的最优取值为 $10\lceil \log_2(M) \rceil$, 但因为 $\log_2 M \leqslant \ell \leqslant 2^{M-1} - 1$, 只有当 $M \geqslant 6$ 时满足, 所以本文编码方法只针对类别数 $M \geqslant 6$ 的情况. LHSS 搜索算法如下:

Step 1: 初始化编码矩阵 $H =$ Random Codes.

Step 2: 计算 H 的 k 近邻错分率 $\bar{\varepsilon}'$, 并保存每个 ε_j' 到 $E = \{\varepsilon_j' | j = 1, 2, \cdots, \ell\}$, $E' = \text{sortascending}(E)$.

Step 3: While $\bar{\varepsilon}'$ is not improved do.

Step 3.1: for $i = 1$ to M do.

Step 3.1.1: for $j = 1$ to M, $i \neq j$ & $E'(i) = E'(j)$ do.

交换第 i 和第 j 行并产生新的编码阵 H'; 计算 H' 的 k 近邻错分率 $\bar{\varepsilon}''$, 同时保存每个列的 ε_j'' 到 E''.

Step 3.1.2: if $\bar{\varepsilon}'' < \bar{\varepsilon}'$ then Set $H = H'$, $\bar{\varepsilon}' = \bar{\varepsilon}''$, $E' = \min(\varepsilon_i', \varepsilon_i'')(\varepsilon_i' \in E', \varepsilon_i'' \in E'', i = 1, 2, \cdots, \ell)$.

表 6 不同类别数的二类划分数

类别数	二类划分种类
$M = 5$	$\{k_1, k_2 \| k_1 : (1^+, 4^-), k_2 : (2^+, 3^-)\}$
$M = 6$	$\{k_1, k_2, k_3 \| k_1 : (1^+, 5^-), k_2 : (2^+, 4^-), k_3\,(3^+, 3^-)\}$
$M = 7$	$\{k_1, k_2, k_3 \| k_1 : (1^+, 6^-), k_2 : (2^+, 5^-), k_3\,(3^+, 4^-)\}$
$M = 8$	$\{k_1, k_2, k_3, k_4 \| k_1 : (1^+, 7^-), k_2 : (2^+, 6^-), k_3\,(3^+, 5^-), k_4\,(4^+, 4^-)\}$
$M = n$	$\{k_1, k_2, \cdots, k_n \| k_1 : (1^+, (n-1)^-), k_2 : (2^+, (n-2)^-), \cdots, k_n : ((\lfloor n/2 \rfloor)^+, (\lceil n/2 \rceil)^-)\}$

Step 4: 输出 E' 和 H.

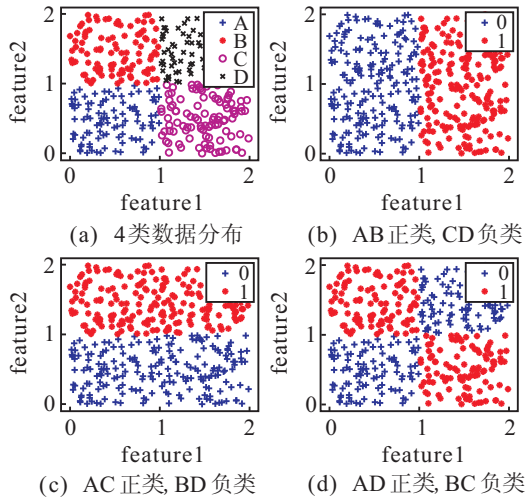
上述算法与文献 [11] 的算法不同之处在于: 行交换是根据 E' 确定的, E' 由每次循环产生 H' 的各列 k 近邻错分率按升序组成的 E'' 与之比较并取相对应的较小元素组成. E' 最后也是按降序排列. 在进行行交换时, 根据 $E'(i)$ 是否与 $E'(j)$ 相等来决定是否交换, 相等即交换. 这样做的目的是每次进行行交换时总能保证前一次的最小 ε'_j 所对应的二类划分保存在本次新构造的 H' 中, 此外按此行交换后能更快地搜索出更好的二类划分.

3.4　算法描述

由于本文算法的核心思想是基于第 2 节中的两个假设, 对于该假设是否合理关系到本文算法的成败与否, 为此本节先就假设的有效性做论证, 最后提出算法步骤.

假设 2 可以理解为: 对于给定的一组编码, 决定性能优劣的是其 ℓ 个列编码, 当其编码合理时往往反映出该列编码组对样本的最优二类划分.

在图 3 中有 $Y = (A, B, C, D)$ 四类数据, $\ell = (1\ 1\ 0\ 0)$ 为图 3(b) 编码划分, $\ell = (1\ 0\ 1\ 0)'$ 为图 3(c) 编码划分, $\ell = (1\ 0\ 0\ 1)'$ 为图 3(d) 编码划分. 很容易看出 $\ell(1), \ell(2)$ 列编码是最优的, 因为按此两种编码方法其数据是线性可分的, 而基于 $\ell(3)$ 列编码划分的数据为线性不可分. 线性可分说明绝大部分相邻样本被划分到同一类, 如图 3(b) 和图 3(c), 图 3(d) 线性不可分. 可以看出相邻样本不为同一类, 与假设 2 一致.



(a) 4类数据分布　(b) AB正类, CD负类

(c) AC 正类, BD 负类　(d) AD 正类, BC 负类

图 3　二类划分示意图

构造 ℓ 个最优列编码且满足行列的分离性即可得到最优编码矩阵. 注意到, 在这里暂且没有考虑最大化编码矩阵的 Hamming 距离, 虽然这样在解码时可能会增加误分率, 但可通过设计不同于基于 Hamming 距离解码的其他解码方法来克服由此带来的不足, 文献 [3,12] 对此方法进行了卓有成效的探索.

由于不考虑编码矩阵的最大化 Hamming 距离问题, 只需满足编码的两个基本设计准则, 在进行编码矩阵设计时可认为编码阵的行向量设计不影响整个编码阵性能的优劣. 表 7 和表 8 所示的两种编码阵为等效编码矩阵.

表 7　编码矩阵 M_1

Class	Cord Word		
	f_1	f_2	f_3
C_1	1	1	1
C_2	0	0	0
C_3	**0**	**0**	**1**
C_4	**0**	**1**	**0**

表 8　编码矩阵 M_2

Class	Cord Word		
	f'_1	f'_2	f'_3
C_1	1	1	1
C_2	0	0	0
C_3	**0**	**1**	**0**
C_4	**1**	**0**	**0**

从表 7 和表 8 可以看出, 两种编码矩阵 M_1 和 M_2 的列向量分别对应: $f_1 = f'_3$, $f_2 = f'_1$, $f_3 = f'_2$, 虽然 C_3 和 C_4 两类的编码方式不同 (图中用加粗标示), 但仍认为 $M_1 = M_2$, 即 M_1 和 M_2 等效.

在不考虑编码矩阵的类编码 (行向量) 设计问题时, 问题主要集中在设计 ℓ 个二类划分编码, 即编码阵的 ℓ 个列向量, 使其满足 2.1 节所述的最近邻域评价准则以达到最优化目的, 然后利用设计输出编码的两个基本原则 (行列分离性) 对其进行检验. 算法步骤如下:

输入: 训练样本集 A, 最近邻域个数 k;

输出: 输出编码矩阵 H.

Step 1: 初始化编码阵 H. 若 H 给定, 则按给定编码阵对其初始化; 若未给定, 则利用随机编码阵对其初始化.

Step 2: 运用 2.3 节提出的 LHSS 找出 E' 和 H.

Step 3: 取 E' 前 ℓ 个元素对应的二类划分组成输出编码阵 H'', 判断是否满足行列分离性. 若满足, 则转 Step 5; 若不满足, 则转下一步.

Step 4: 对 H'' 进行列变换, 找出满足行列分离性的等效编码阵 H'''. 若找出, 则转下一步; 否则, 令 $\ell = \ell + 1$, 并转 Step 3.

Step 5: 利用式 (3) 比较 $H''(H''')$ 和 H 的 $\bar{\varepsilon}'$. 若 $\bar{\varepsilon}'_{H''(H''')} < \bar{\varepsilon}'_H$, 则最终输出编码矩阵为 $H''(H''')$; 反之, 则为 H.

注意到, 算法的 Step 5 对 $\bar{\varepsilon}'_{H''(H''')}$ 和 $\bar{\varepsilon}'_H$ 进行比较之后才能最终确定输出编码矩阵, 原因是当算法执行了 Step 4 之后, 由于取出的 ℓ 个二类划分不再是 E' 前 ℓ 个元素对应的划分, 其 $\bar{\varepsilon}'$ 不一定会比迭代后的 H 小, 因此 Step 5 是必要的.

4　实验分析

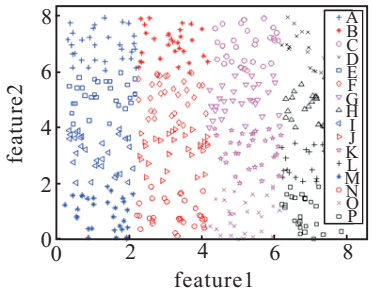
本节将通过实验验证所提出的结论, 重点验证的有:

1) 2.3 节提出的有限启发式搜索策略.

2) 2.4 节提出的基于最小化 k 近邻错分率的多类输出编码矩阵.

4.1 实验数据

在实验中将用到两类数据集, 第 1 类是文献 [11] 提供的人工数据集, 第 2 类是选取 UCI 数据库的数据集. 图 4 为人工数据集及其初始编码矩阵, 表 9 为 UCI 数据集及各类数据描述. 本文针对部分 UCI 数据集进行了归一化处理, 并删除了一些样本数很小的类. 表 9 中类别数列括号里的数为原样本类数, 同时为了提高分类速度, 实验中对高维数据使用主成分分析法 (PCA) 对其进行降维处理, 在属性列中括号内为原属性数目.



(a) 16 类人工数据集

Class	c1	c2	c3	c4	c5	c6
A	1	1	1	0	0	0
B	1	1	1	1	0	0
C	1	1	1	1	1	0
D	1	1	1	1	1	1
E	0	1	1	0	0	0
F	0	1	1	1	0	0
G	0	1	1	1	1	0
H	0	1	1	1	1	1
I	0	0	1	0	0	0
J	0	0	1	1	0	0
K	0	0	1	1	1	0
L	0	0	1	1	1	1
M	0	0	0	0	0	0
N	0	0	0	1	0	0
O	0	0	0	1	1	0
P	0	0	0	1	1	1

(b) 初始输出编码矩阵

图 4 人工数据集及初始输出编码矩阵

表 9 UCI 数据集各数据描述

Problem	#Train	#Attributes	#Classes
Glass	214	9(9)	6(7)
Soybean	307	10(35)	18(19)
ecoli	336	8(8)	8
yeast	1484	8(8)	10
Segmentation	2310	8(19)	7
Isolet	7797	20(617)	26
Letter	20000	16	26

4.2 实验设计

为比较本文所提出的有限启发搜索策略与文献 [11] 所提搜索方法的优劣, 数据采用图 4(a) 所示的 16 类人工数据集, 初始编码阵为给定编码阵 (如图 4(b)) 进行 30 次行变换后所得到的. 图 4(b) 所示的输出编码阵为理想编码阵.

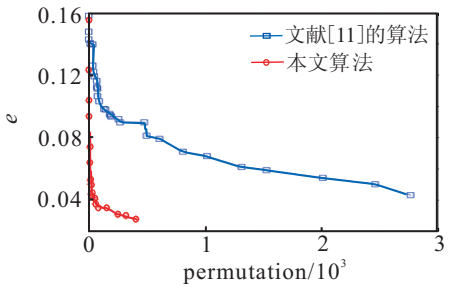
在验证基于最小 k 近邻错分率输出编码矩阵的优劣时, 将其与 1vs1, 1vs All 以及 Dense 和 Sparse 两种随机编码作比较. 此外, 选择 logistic linear classifier 和 SVM (核函数取多项式核函数, $C = 2$) 作为基分类器, 分别比较在这两种不同分类器下的分类错误率. 在估计分类错误率时采用 10 重交叉验证来进行, 并将利用双边估计 t 检验法来计算置信水平为 0.95 的分类错误率置信区间作为最终结果, 计算公式如下:

$$\frac{|\bar{x}-u|}{\sigma/\sqrt{n}} \geqslant t_{0.05}(n-1). \tag{4}$$

u, σ 分别表示 10 重交叉验证的均值和标准差[13]. 实验中所用分类器均来自 PRTool(http://www.prtools.org) 工具箱, 实验机器配置为 1 G 内存, 2.30 G CPU, 算法基于 Matlab 7.7(R2008b) 实现.

4.3 实验结果与分析

从图 5 所示的结果可以看出, 本文所提出的搜索算法比文献 [11] 所提算法能更快地搜索到最优输出编码矩阵, 由图 5 可以看出, 本文算法只搜索了不到 500 个二类划分的组合就使 k 近邻错分率小于 0.04, 而文献 [11] 的算法搜索了近 3 000 个二类划分组合, k 近邻错分率才接近 0.04. 另外, 从图 5(b) 中可以看出, 两种搜索算法所得到的最终输出编码矩阵与图 4(b) 所示的理想编码矩阵十分接近 (矩阵对应各列相似), 从而说明了算法的有效性.



(a) 两种算法错误率比较

利用本文搜索算法得到的最终输出编码矩阵	利用文献[10]搜索算法得到的最终输出编码矩阵
1 1 1 1 1 1	1 1 1 1 1 1
1 1 1 1 1 0	1 1 1 1 1 0
1 1 1 1 0 0	1 1 1 0 0 0
1 1 1 0 0 0	0 1 1 0 0 0
0 1 1 1 1 1	0 1 1 1 1 1
0 1 1 1 1 0	1 1 1 1 0 0
0 1 1 1 0 0	0 1 1 1 0 0
0 1 1 0 0 0	0 0 1 0 0 0
0 0 0 1 1 1	0 1 1 1 1 0
0 0 1 1 1 1	0 0 1 1 0 0
0 0 1 1 0 0	0 0 1 1 1 0
0 0 1 0 0 0	0 0 1 1 1 1
0 0 0 0 0 0	0 0 0 1 1 0
0 0 1 1 1 0	0 0 0 1 0 0
0 0 0 1 0 0	0 0 0 0 0 0
0 0 0 1 1 0	0 0 0 1 1 1

(b) 对应最终输出编码矩阵

图 5 两种搜索算法结果比较

表 10 所示结果为各类型编码在基于人工数据集下的分类错误率及其置信区间, ER 代表错分率, CI 为置信水平为 0.95 的置信区间. 在用 loglc 做基分类器时, 1vs1 编码的错误率最小, 本文编码方法次之, 但 1vs1 所需分类时间最长, 本文编码方法最短. 从表 11 可以看出, 1vs1 方法的编码长度最长, 其包含可分性好的二类划分的可能性越大, 故其分类错误率相比

其他编码长度小的方法小, 而本文通过选取最小 k 近邻错分率的 6 个二类划分组成编码阵, 可以看出其错分率仅次于 1vs1 编码方法. 由此可以看出, 本文提出的最小 k 近邻错分率准则是有效的, 相比其他编码方法, 本文方法能在保证较小错分率的前提下得出最短的编码阵, 使多类分类的复杂度有明显的降低, 分类所需时间也最短. 图 6 为各编码方法在不同 k 下的邻域错分率比较, 从中也可以看出本文方法的 k 近邻错分率仅比 1vs all 大, 这里需要注意的是之所以 1vs all 的 k 近邻错分率最小, 原因是实验中产生的人工数据集是可分性较好的数据, 各类之间可以找到明显的分类线, 在对 1vs all 计算 k 近邻错分率时每次只有其中的一类会产生错分数, 其余各类均不产生, 故其 k 近邻错分率最小. 但从本文下面的实验结果中可以看出, 基于 1vs all 的分类错误率并不比其他编码方法小, 因此在对可分性好的数据集进行多类分类时, 应避免使用 1vs all 编码作为初始编码矩阵.

表 12 和表 13 分别为 5 种编码方法在 7 种 UCI 数据下的错分率及置信水平为 0.95 的置信区间, 其中每 1 格的第 2 行为分类所需的总时间, 表 12 的基分类器为 loglc, 表 13 的基分类器为 svm. 从这两表格中可以看出, 本文编码方法与 Dense 编码方法的分类效果基本相同, 这是由于本文采用 Dense 作为初始编码矩阵, 但分类所需时间要远小于基于 Dense 编码的分类时间, 原因是本文编码方法产生的编码长度相比 Dense 要小, 表 14 为各编码方法的编码长度. 在实验结果中, 基于 loglc 的分类器中本文编码方法表现最好的数据有 Glass 数据集 (18.88±1.57), ecoli 数据集 (7.83±0.74), Isolet 数据集 (22.21±0.26); 基于 svm 的分类器中本文编码方法表现最好的数据有 Glass 数据集 (14.53±0.91), Segmentation 数据集 (4.8±0.69), 在其他数据中错分率居中.

表 10　十重交叉验证错误率及置信水平为 0.95 的置信区间

编码类型	基分类器			
	loglc		svm	
	ER & CI(%)	Time(s)	ER & CI(%)	Time(s)
1 vs 1	**21.29±0.78**	54	**22.67±0.64**	105
1 vs all	42.08±0.48	8	30.4±1.02	41
Dense	35.71±0.63	18	28.58±0.91	143
Sparse	37.81±0.37	25	25.54±0.5	85
OCMkNN	26.75±0.59	3	23.77±0.67	25

表 11　基于人工数据集各编码类型的编码长度

编码类型	编码长度
1 vs 1	120
1 vs all	16
Dense	40
Sparse	60
OCMkNN	6

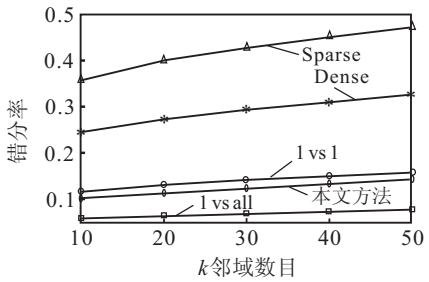


图 6　各编码方法 k 近邻错分率比较

表 12　基于 loglc 的 UCI 数据集分类错误率

	1 vs 1	1 vs all	Dense	Sparse	OCMkNN
Glass	19.06±1.06 9.5	22.1±1.06 2.8	20.33±0.87 11.4	—	**18.88±1.57** 3.8
Soybean	**6.14±0.91** 83.5	10.36±1.25 14.1	9.18±1.24 33.4	7.68±0.76 48.9	8.53±0.95 14.5
ecoli	9.58±0.99 13.4	10.63±0.66 3.6	8.15±0.78 12.5	8.04±0.53 19.6	**7.83±0.74** 4.5
yeast	**20.71±0.36** 20.2	29.95±0.32 5.4	22.88±0.30 18.9	24.38±0.38 23.1	24.26±0.31 12.1
Segmentation	**4.4±0.66** 11.6	9.21±0.66 4.5	6.26±0.49 16.2	6.54±0.71 22.6	5.4±0.87 10.8
Isolet	23.12±0.84 206.2	24.1±0.94 82.18	22.3±0.59 86.1	26.85±0.53 95.8	**22.21±0.26** 78.9
Letter	**19.04±0.33** 160.3	37.86±0.27 17.3	34.45±0.38 27.2	32.77±0.40 35.9	30.37±0.40 19.6

表 13　基于 svm 的 UCI 数据集分类错误率

	1 vs 1	1 vs all	Dense	Sparse	OCMkNN
Glass	16.36±1.04 13.4	17.1±1.05 9.4	16.96±1.03 39.5	—	**14.53±0.91** 12.8
Soybean	**8.14±0.95** 119.5	8.30±0.69 34.2	8.04±0.65 74.1	12.84±087 64.7	8.82±0.78 25.3
ecoli	**7.95±0.62** 25.8	14.14±0.63 14.4	9.58±0.52 54.7	12.02±0.30 56.6	9.20±0.46 19.5
yeast	**22.1±0.44** 132.3	36.58±0.29 883．2	23.68±0.36 4022.3	23.38±0.26 1012.5	25.46±0.28 2384.2
Segmentation	6.29±0.55 32.5	7.59±0.89 84.9	4.96±0.45 387.4	5.93±0.90 187.2	**4.8±0.69** 231.5
Isolet	26.77±0.63 304	19.77±0.90 310.8	**15.02±0.71** 481.4	17.87±0.67 148.2	22.94±0.61 336.5
Letter	**19.88±0.52** 288.7	25.07±0.87 1191.4	20.88±0.53 2914.5	23.26±0.36 581.3	27.28±0.47 2101.6

表 14 基于 UCI 数据集各编码类型的编码长度

编码类型	编码长度						
	Glass	Soybean	ecoli	yeast	Segment	Isolet	Letter
1 vs 1	15	153	28	45	21	325	325
1 vs all	6	18	8	10	7	26	26
Dense	26	42	30	34	29	48	48
Sparse	—	63	45	50	43	71	71
OCMkNN	8	28	10	11	9	32	32

5 结 论

多类分类是学习领域的一个重点和难点, 借助两类分类的研究成果对其进行简化是目前最有效的方法之一, 如何使多类分类问题准确地映射为多个两类分类问题是进行这种简化的关键. 针对传统多类分类编码方法不依据数据集产生输出编码矩阵的不足, 本文提出了最小k近邻错分率准则, 依据该准则产生最有利于多类分解为二类问题的输出编码矩阵. 在构造输出编码矩阵时, 本文把构造问题简化成搜索问题, 并以此得出最优输出编码矩阵. 通过对人工数据集和 UCI 数据集的测试, 证明了本文所提出的方法能有效地构造出使分类效果突出的输出编码. 对比几种经典的编码方法, 本文所得编码矩阵能在保证分类精度的情况下最大限度地减少编码长度.

下一步将重点研究针对输出编码矩阵确定的ℓ个二类划分如何找出与之分别对应的最优二类分类器, 目前大多数文献的做法是选择一种常用的基分类器作为每个划分的学习器, 这种做法的合理性有待研究. 另外, 设计出更好评价输出编码矩阵优劣的评价标准也是一个值得研究的问题.

参考文献(References)

[1] Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes[J]. J of Artificial Intelligence Research, 1995, 11(2): 263-286.

[2] Hastie T, Tibshirani R. Classification by pair wise coupling[J]. The Annals of Statistics, 1998, 26(1): 451-471.

[3] Allwein E, Schapire R, Singer Y. Reducing multiclass to binary: A unifying approach for margin classifiers[J]. Machine Learning Research, 2002, 12(1): 113-141.

[4] Dietterich T G, Bakiri G. Error-correcting output codes: A general method for improving multiclass inductive learning programs[C]. Proc of the Ninth National Conf on Artificial Intelligence (AAAI-91). Menlo Park: AAAI Press, 1991: 572-577.

[5] Schapire R E. Using output codes to boost multiclass learning problems[C]. Proc of the Fourteenth Int Conf on Machine Learning. Tennessee, 1997: 123-131.

[6] Crammer K, Singer Y. On the learnability and design of output codes for multiclass problems[C]. Proc of the 13th Annual Conf on Computational Learning Theory. California, 2000: 567-572.

[7] 蒋艳凰, 赵强利, 杨学军. 一种搜索编码法及其在监督分类中的应用[J]. 软件学报, 2005, 16(6): 1081-1088.
(Jiang Y H, Zhao Q L, Yang X J. A search coding method and its application in supervised classification[J]. J of Software, 2005, 16(6): 1081-1088.)

[8] Nobuhiko Yamaguchi, Naohiro Ishii. Constructing error correcting output coding classifiers[C]. Proc of the 9th Int Conf on Neural Information. Hong Kong, 2003: 432-439.

[9] Pujol O, Radeva P, Vitria J Discriminant. 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.

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

[11] PaoloSimeon, David M J Tax, Robert P W Duin, et al. A fast approach to improve classification performance of ECOC classification systems[J]. Computer Science Trans Structural, Syntactic, and Statistical Pattern Recognition, 2008, 3(4): 459-468.

[12] Ludmila I Kuncheva. Using diversity measures for generating error-correcting output codes in classier ensembles[J]. Pattern Recognition Letters, 2005, 26(1): 83-90.

[13] 盛骤, 谢式千, 潘承毅. 概率论与数理统计[M]. 北京: 高等教育出版社, 2003: 219-221.
(Sheng Z, Xie S Q, Pan C Y. Probability theory & mathematical statistic[M]. Beijing: Higher Education Press, 2003: 219-221.)