一种基于混合二叉树结构的多类支持向量机分类算法

冷强奎[1] 刘福德[2] 秦玉平[3]
(渤海大学信息科学与技术学院 辽宁 锦州 121000)[1] (渤海大学大学基础教研部 辽宁 锦州 121000)[2]
(渤海大学工学院 辽宁 锦州 121000)[3]

摘 要 为提高多类支持向量机的分类效率,提出了一种基于混合二叉树结构的多类支持向量机分类算法。该混合二叉树中的每个内部结点对应一个分割超平面,该超平面通过计算两个距离最远的类的质心而获得,即该超平面为连接两质心线段的垂直平分线。每个终端结点(即决策结点)对应一个支持向量机,它的训练集不再是质心而是两类(组)样本集。该分类模型通常是超平面和支持向量机的混合结构,其中超平面实现训练早期的近似划分,以提升分类速度;而支持向量机完成最终的精确分类,以保证分类精度。实验结果表明,相比于经典的多类支持向量机方法,该算法在保证分类精度的前提下,能够有效缩短计算时间,提升分类效率。

关键词 支持向量机,多类分类,混合二叉树,质心表达
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.05.037

Multi-class Classification Algorithm for SVM Based on Hybrid Binary Tree Structure

LENG Qiang-kui[1] LIU Fu-de[2] QIN Yu-ping[3]
(College of Information Science and Technology, Bohai University, Jinzhou, Liaoning 121000, China)[1]
(Research and Teaching Institute of College Basics, Bohai University, Jinzhou, Liaoning 121000, China)[2]
(College of Engineering, Bohai University, Jinzhou, Liaoning 121000, China)[3]

Abstract In order to improve the classification efficiency of mutli-class support vector mechine, a multi-class classification algorithm for support vector machine(SVM) based on hybrid binary tree structure was proposed. In the structure, each internal node corresponds to a partition hyperplane, which is obtained as perpendicular bisectors of linking two centroid segements of the two farthest classes from each other. Each terminal node(i.e., decision node) is associated with a SVM, whose training set is two sets of samples instead of two centroids. In general, the resulting classification model represents a hybrid form, consisting of hyperplanes and SVMs. The approximate hyperplanes by centroids can provide fast partition in the early stages of the training phase, whereas the SVMs will perform the final precise decision. Experimental results show that compared with the classical multi-class SVM, the proposed algorithm can reduce the computational time and improve the classification efficiency with similar classification accuracy.

Keywords SVM, Multi-class classification, Hybrid binary tree, Centroid representation

1 引言

支持向量机(Support Vector Machine, SVM)[1-2]作为一种典型的分类算法被广泛关注。最初设计SVM来解决两类分类问题,而在实际应用中,分类场景往往多于两类,如文本分类、人脸识别和手写字符识别等。为了将SVM推广到多类情形,学者们提出了诸多方法。其中,最常见的是一对一[3-4]、一对多[5-6]、导向无环图[7-8]和二叉树[9-11]等。另外,随着研究的深入,一些新方法也被提出,如单类集成方法[12]、多类孪生SVM[13-14]、纠错输出码[15]和最小紧闭球[16]等。

这些方法通常按某种方式组合多个二类分类器来实现多类分类。下面以一个$K(K\geqslant 2)$类问题为例,从直观上说明几种典型多类分类器的组合方式及特点。

1)一对一(见图1(a)):该方法需要建立$K(K-1)/2$个子分类器,每个子分类器用于划分二分类问题。测试样本需经过所有的子分类器,类别决策通过投票机制产生。

2)一对多(见图1(b)):该方法需要建立K个子分类器。每个子分类器在其中一类和其余所有类间训练得到。测试样本的类别与具有最大决策函数值的类别相同。

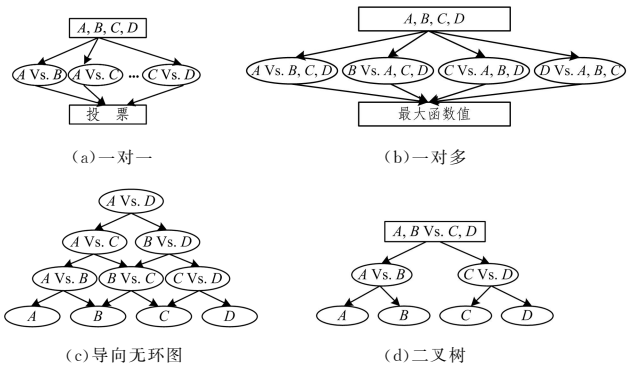
3)导向无环图(见图1(c)):该方法需要建立一个带根结

到稿日期:2017-05-18 返修日期:2017-07-05 本文受国家自然科学基金项目(61602056),辽宁省博士科研启动基金项目(201601348),辽宁省教育厅科研项目(LZ2016005)资助。

冷强奎(1981-),男,博士,副教授,CCF会员,主要研究方向为模式识别、机器学习,E-mail:qkleng@gmail.com(通信作者);**刘福德**(1964-),男,副教授,主要研究方向为机器学习,E-mail:fdliu@gmail.com;**秦玉平**(1965-),男,博士,教授,主要研究方向为模式识别、机器学习,E-mail:jzqinyuping@gmail.com。

点的导向无环图，它包括 $K(K-1)/2$ 个内部结点和 K 个叶子结点。当测试样本到达叶子结点时，类别被最终确定。

4）二叉树（见图 1(d)）：该方法通过递归地将当前类分割为两个子类来构建一棵带根的二叉树。每个叶子结点通常只包含一类，当测试样本到达叶子结点时，类别也随之确定。



（a）一对一　（b）一对多　（c）导向无环图　（d）二叉树

图 1　多类 SVM

Fig. 1　Multi-class SVM

一对一方法和一对多方法的缺点在于类别决策需要经过较多的二类分类器，实时性受到制约。导向无环图虽然在一定程度上能够减少测试经过的结点数，但它仍需要训练 $K(K-1)/2$ 个二类分类器，特别是当类别数较大时，其需要更长的生成时间。而对于二叉树结构的多类分类器，它只需要训练 $K-1$ 个二类分类器，并且对于一个测试样本，它只需测试 $\log_2 K$ 次即可获得类别输出。这种二叉树结构的分类器已经被多种不同的方法实现[9-11]，但它们有一个共同点，即均采用“分而治之”策略将较大型的原始问题分解为若干个子问题，以便求解。

本文实现了一种新的基于二叉树结构的多类 SVM 分类算法，旨在提升分类效率。该算法受到 Kostin 提出的决策树[17]的启发，在每个内部结点，通过两个最远类间的质心来计算分割超平面，从而实现在训练早期对模式类进行快速划分。然后，该超平面将继承到的当前结点的所有类分为两组，从而产生新的子树结点。当到达决策结点时，计算相应的 SVM 模型，以实现最终的分类决策。这样，一棵二叉分割树被建立，该二叉树通常是包含超平面和 SVM 的混合结构。其中，超平面实现训练早期的快速划分，而 SVM 完成最终的精确分类。

本文第 2 节简单介绍了 SVM；第 3 节给出了混合二叉树的构建过程，并描述该二叉树的生成算法；第 4 节给出了算法在标准数据集上的实验结果并对其进行了分析；最后总结全文并讨论进一步的工作。

2　SVM 简介

给定训练集 $S=\{(x_i,y_i),i=1,2,\cdots,l\}$，SVM 通过在特征空间中计算一个超平面 $H:\langle \boldsymbol{w},\phi(\boldsymbol{x})\rangle+b=0$ 来划分两类样本。其中，$\boldsymbol{x}_i\in\boldsymbol{R}^n$ 表示输入样本，$y_i\in\{1,-1\}$ 表示相应的输出标签，l 表示样本个数，$\phi(\cdot)$ 表示 $\boldsymbol{R}^n$ 到更高维特征空间的映射，$\langle\cdot\rangle$ 表示两个向量的点积。为了得到超平面 H，SVM 需要求解以下优化问题：

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s. t. } y_i(\langle \boldsymbol{w},\phi(\boldsymbol{x}_i)\rangle+b)\geqslant 1,i=1,\cdots,l \tag{1}$$

式(1)是一个带线性不等式约束的二次优化问题，通常将其求解过程转化为相应的对偶问题：

$$\min\left(\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle-\sum_{i=1}^{l}\alpha_i\right)$$
$$\text{s. t. } \alpha_i\geqslant 0,i=1,\cdots,l \tag{2}$$

根据 Mercer 定理[18]，可以使用原空间的正定核函数来表示特征空间中两个向量的点积。使用核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 来代替 $\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle$，式(2)可改写为：

$$\min\left(\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{i=1}^{l}\alpha_i\right)$$
$$\text{s. t. } \alpha_i\geqslant 0,i=1,\cdots,l \tag{3}$$

为了处理非线性的情况并允许一定的错分，引入软间隔 SVM：

$$\min\left(\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{i=1}^{l}\alpha_i\right)$$
$$\text{s. t. } \sum_{i=1}^{l}\alpha_i y_i=0,0\leqslant\alpha_i\leqslant C,i=1,\cdots,l \tag{4}$$

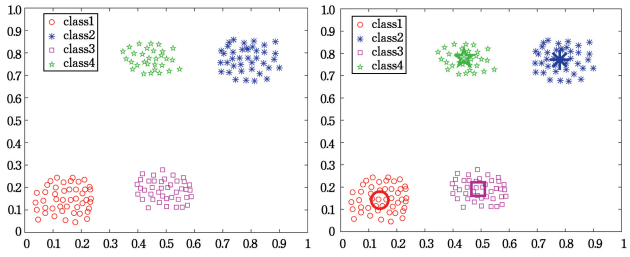
其中，C 为惩罚系数，表示对离群点的惩罚，C 越大表示越重视离群点。在求得对偶问题的最优解后，对于一个新的样本 $\boldsymbol{x}$，SVM 使用式(5)来预测其类别：

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{x})+b\right) \tag{5}$$

其中，sgn 为符号函数，α_i 为式(4)的解。Karush-Kuhn-Tucher (KKT)条件保证了唯一解的充分必要性，称 $\alpha_i>0$ 的样本为支持向量，偏置 b 的值可通过 KKT 条件计算得到。

3　混合二叉树结构的多类 SVM

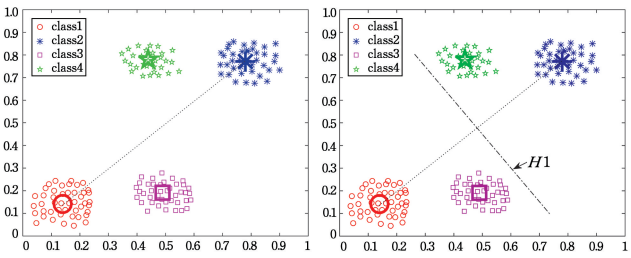
本节将标准的 SVM 推广到多类情形，推广结果以递归算法的方式进行表达。为了更清晰和直观地进行描述，以一个二维平面为例（见图 2－图 4）来说明该算法的生成过程。



（a）原始数据　（b）质心

图 2　类的质心表达

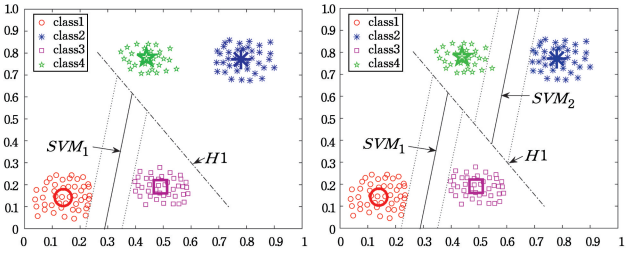
Fig. 2　Centroid representation of classes



（a）极端质心选择　（b）分类超平面

图 3　二叉分割树根结点的生成

Fig. 3　Generation of root of binary partition tree



(a)左子树 SVM_1 (b)右子树 SVM_2

图 4 使用 SVM 进行分类决策的示意图

Fig. 4 Classification decision by using SVM

令 $K(K\geqslant 2)$ 为 $\boldsymbol{R}^n$ 中模式类的类别数目，相应地，可构造一个类集 $T=\{Class_1,Class_2,\cdots,Class_K\}$，其中每个类 $Class_j=\{\boldsymbol{X}_l^j \mid l=1,2,\cdots,N_j\}$ 由 $\boldsymbol{R}^n$ 中的样本组成，即：

$$\boldsymbol{X}_l^j=(\boldsymbol{x}_{1l}^j,\boldsymbol{x}_{2l}^j,\cdots,\boldsymbol{x}_{nl}^j),j=1,2,\cdots,K \tag{6}$$

在初始状态下，本文尝试将 T 中的所有类分为两组。受文献[17]的启发，分组计算不涉及每个类中的样本，而是通过它们的质心来进行划分。某个类 $Class_j$ 的质心为：

$$C^j=(c_1^j,c_2^j,\cdots,c_i^j,\cdots,c_n^j) \tag{7}$$

其中，

$$c_i^j=\frac{1}{N_j}\sum_{l=1}^{N_j}\boldsymbol{x}_{il}^j,j=1,2,\cdots,K \tag{8}$$

因此，一个质心的集合 C 被确定，它与类集 T 相对应。

$$C=\{C^j \mid j=1,2,\cdots,K\} \tag{9}$$

图 2(a)给出了一个四类原始问题，图 2(b)给出了相应的质心表达，即 $C=\{C^1,C^2,C^3,C^4\}$。接下来，通过计算任意两个质心的欧氏距离来确定两个极端质心，并将其标记为 C^p 和 C^q，即它们之间的距离为最远。该寻找过程可通过文献[19]中描述的启发式方法来完成。

每一个极端质心可以标识一个组 G_C^p 或 G_C^q。C 中的其他质心根据与 C^p 和 C^q 的位置关系来确定分组。这里使用 $dist$ 表示欧氏距离函数，对于质心 $C^j(j\neq\{p,q\})$，如果 $dist(C^j,C^p)\leqslant dist(C^j,C^q)$，则 $C^j\in G\_C^p$；否则，$C^j\in G\_C^q$。在图 3(a)中，两个极端质心 C^1 和 C^2 首先被找到，然后确定两个分组 G_C^1 和 G_C^2。

接着递归地创建二叉分割树的每一个结点。根据组的划分情况，如果 G_C^p 和 G_C^q 中包含的类别数均多于一个，则通过极端质心 C^p 和 C^q 创建一个内部结点(包括根结点)，并计算一个分割超平面 H：

$$\sum_{i=1}^{n}\{(c_i^p-c_i^q)\cdot\boldsymbol{x}_i-\frac{1}{2}[(c_i^p)^2-(c_i^q)^2]\}=0 \tag{10}$$

该超平面为连接两极端质心 C^p 和 C^q 的线段的垂直平分线。如图 3(b)所示，超平面 $H1$ 由 C^1 和 C^2 根据式(10)计算得到。

如果 G_C^p 或 G_C^q 中只包含一个类，那么一个终端结点将被创建。在终端结点处使用 SVM 进行分类决策，需要说明的是，与此结点相关的训练集由两组中的样本来构造，而不仅仅使用质心。当一个测试样本从根结点到达二叉分割树的叶子结点时，它的类别也被最终确定。图 4 给出了使用 SVM 进行最终分类决策的结果，其中 SVM_1 分开了类 $Class_1$ 和 $Class_3$，SVM_2 分开了类 $Class_2$ 和 $Class_4$。与之对应的二叉树如图 5 所示，/代表超平面，⚡代表 SVM。

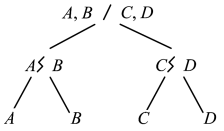
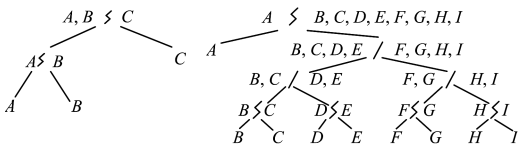


图 5 4 类示例对应的二叉树结构

Fig. 5 Binary tree structure corresponding to four-class example

通常情况下，生成的二叉分割树是一种包含了质心计算的超平面和 SVM 的混合结构。其中，超平面能够对当前结点中的类别进行近似划分，其生成速度较快；而 SVM 能够实现终端结点的精确分类，从而保证分类质量。但需要说明的是，在某些情况下获得的二叉树并不总是混合的、平衡的，还包括一些特别的形式，如图 6 所示。



(a) 非混合二叉树 (b) 非平衡二叉树

图 6 特殊形式的二叉树

Fig. 6 Binary tree with special forms

算法 1 给出了混合结构二叉分割树的生成算法(Binary Partition Tree with Hybrid Structure，BPT-HS)。为了便于理解，在算法 1 中加入了详细的描述语句。

算法 1 混合结构二叉分割树生成算法

输入：1 个 K 类训练集，核参数，精度参数

输出：一棵二叉分割树

1. 计算 K 类数据的质心 $C=\{C^1,C^2,\cdots,C^K\}$。
2. 计算两个极端质心 C^p 和 C^q，然后将 C 分为两组 G_C^p 和 G_C^q：$\forall C^j(j\neq\{p,q\})$，如果 $dist(C^j,C^p)\leqslant dist(C^j,C^q)$，则 $C^j\in G\_C^p$；否则，$C^j\in G\_C^q$。
3. 如果 G_C^p 或 G_C^q 只包含 1 个类，则在两组样本集间构造一个 SVM，并为单个类建立 1 个叶子结点：若 G_C^p 包含的类别数大于 1，则 $C^{left}=G\_C^p$，构建左子树；否则 $C^{right}=G\_C^q$，构建右子树。
4. 否则，根据式(10)在两个质心 C^p 和 C^q 间计算分割超平面，并且：$C^{left}=G\_C^p$，构建左子树；$C^{right}=G\_C^q$，构建右子树。
5. 若所有 K 个类均产生叶子结点，则返回。

BPT-HS 算法是递归的，它将贯序地分割原始样本空间，从而进入更小的子空间，直到产生叶子结点为止。对于一个 K 类分类问题，BPT-HS 需要训练 $K-1$ 个二分类器。通常情况下，这些分类器既包含由质心计算得到的超平面又包含 SVM。假定训练集中包含 l 个样本，BPT-HS 的时间复杂度可被估计为：

$$T_{BPT\text{-}HS}=O(mKl)+O[(K-1-m)Kl^2],\quad m=0,1,\cdots,K-1 \tag{11}$$

其中，Kl 对应 m 个超平面，Kl^2 对应 $K-1-m$ 个 SVM。显然，SVM 的训练时间高于超平面，因此希望生成的二叉分割树是近似平衡的，这样就会有大约 $K/2$ 个 SVM 被计算，由超平面来完成其余结点的分割，从而使分类模型快速生成。

4　实验结果及分析

实验使用 8 个标准数据集来评估 BPT-HS 的性能。数据集的描述如表 1 所列，其中 vehicle，segment，letter 3 个数据集来自 Statlog[20]，其余 5 个数据集来自 UCI[21]。对于数据集 vowel 和 letter，本文直接使用 LIBSVM[22]标出的训练集和测试集。对于其他 6 个数据集，随机将其分为两部分，一部分用于训练，另一部分用于测试。实验将一对一多类 SVM 作为基准，该方法在文献[23]中被详细分析并被推荐使用。SVM 使用高斯核函数，惩罚参数设置为 1，核宽度参数设置为 $1/dim$（dim 表示维度），精度参数设置为 10^{-3}。机器环境为：I5-3230M CPU 2.60 GHz，4 GB 内存，Windows8.1 操作系统。

表 1　实验所用数据集

Table 1　Datasets used in experiments

name	# class	# training	# testing	# dim
iris	3	75	75	4
wine	3	88	90	13
glass	6	105	109	10
vowel	11	528	462	10
sensorless	11	29249	29260	48
vehicle	4	422	424	18
segment	7	1155	1155	19
letter	26	15000	5000	16

表 2 列出了 BPT-HS 与一对一多类 SVM 在分类精度上的对比。BPT-HS 在 5 个数据集上（vowel，sensorless，vehicle，segment，letter）的性能不如一对一多类 SVM，在 2 个数据集上（iris，wine）取得相同的精度，只在 1 个数据集上（glass）的性能优于一对一多类 SVM。产生这种差异的原因在于，BPT-HS 产生的分类模型中通常会包含部分的分割超平面（参见表 2 中的后两列），而这些超平面是由两类数据的质心计算得到的，代表了一种近似划分的结果，因此会引起分类精度的下降。尽管总体上 BPT-HS 的分类精度不如一对一多类 SVM，但这种差异并非十分显著，如在数据集 letter 上表现出的最大差异仅为 4.70%。

表 2　BPT-HS 的分类精度和模型结构

Table 2　Accuracy and model structure of BPT-HS

name	BPT-HS	1-Vs.-1 SVM	# hyperplanes	# SVM
iris	94.37%	94.37%	0	2
wine	100.00%	100.00%	0	2
glass	65.14%	59.63%	2	3
vowel	77.92%	80.87%	3	7
sensorless	72.06%	74.71%	4	6
vehicle	67.92%	70.28%	1	2
segment	90.56%	90.82%	2	4
letter	77.58%	82.28%	8	17

表 3 列出了 BPT-HS 与一对一多类 SVM 的训练时间和测试时间。可以看出，BPT-HS 通常能花费更少的训练时间来产生二叉分割树模型，其原因在于对于一个 K 类问题，BPT-HS 至多需要训练 $K-1$ 个 SVM，而一对一多类 SVM 却需要训练 $K(K-1)/2$ 个 SVM。例如，在数据集 sensorless 上，BPT-HS 执行训练进程只需花费约 4.30×10^4 ms，而一对一多类 SVM 却需要花费 1.34×10^5 ms。

表 3　训练时间和测试时间

Table 3　Training time and testing time

（单位：ms）

name	Training time		Testing time	
	BPT-HS	1-Vs.-1 SVM	BPT-HS	1-Vs.-1 SVM
iris	3.82	0.85	0.15	0.52
wine	5.98	0.87	0.41	1.50
glass	7.90	12.68	0.71	1.77
vowel	26.98	57.42	0.36	25.94
sensorless	4.30×10^4	1.34×10^5	3015.06	1.56×10^5
vehicle	22.04	23.85	20.81	23.47
segment	100.30	157.55	31.04	88.83
letter	1374.72	1.11×10^4	259.18	1.18×10^4

特别地，当类别数 $K=3$ 时，BPT-HS 不会再产生分割超平面，树中的每一个结点均由 SVM 构造。此时，BPT-HS 的分类精度能够最大程度地得到保证，但会消耗较多的训练时间，具体可参见数据集 iris 和 wine 上的实验结果。当 $K=2$ 时，BPT-HS 退化为一个二类分类器。

在测试时间方面，BPT-HS 明显优于一对一多类 SVM。例如，在数据集 letter 上，BPT-HS 只需花费约 259 ms，而一对一多类 SVM 却需要花费约 1.18×10^4 ms。这对于将 BPT-HS 集成到智能设备中是非常重要的，它可以实现实时响应，从而实现快速分类决策。

结束语　本文提出了一种混合二叉树算法来设计多类 SVM。在训练早期，使用两类质心来计算分割超平面，从而实现模式类的快速划分。当分类进程到达终端结点时，使用 SVM 做精确的分类决策。分类模型通常是超平面和 SVM 的混合结构，该结构表达了对给定模式类的分层处理。实验结果表明，相比于一对一多类 SVM，该方法具有明显的时间优势，计算效率较高，能够实现快速分类响应。当类别数目较小时，它能够获得不错的分类精度。它适合被集成在一些小型嵌入式系统或便携设备中，以利用它的实时性优势对一些实际应用问题进行快速决策。

下一步将引入一些启发式进程来提升该方法的分类精度，从而使其在一些实际应用中被优先考虑。

参 考 文 献

[1] VAPNIK V. The nature of statistical learning theory[M]. New York: Springer-Verlag, 1995.

[2] VAPNIK V. Statistical learning theory[M]. New York: Wiley-Interscience, 1998.

[3] HSU C W, LIN C J. A comparison of methods for multiclass support vector machines[J]. IEEE Transactions on Neural Networks, 2002, 13(2): 415-425.

[4] ROKACH L. Ensemble-based classifiers[J]. Artificial Intelligence Review, 2010, 33(1/2): 1-39.

[5] KREβEL U H G. Pairwise classification and support vector machines[M] // Advances in Kernel Methods. MIT Press, 1999: 255-268.

[6] LORENA A C, DE CARVALHO A C, GAMA J M P. A review on the combination of binary classifiers in multiclass problems [J]. Artificial Intelligence Review, 2008, 30(1): 19-37.

（下转第 237 页）