文章编号: 1001-0920 (2014) 02-0208-07 DOI: 10.13195/j.kzyjc.2012.1472

基于 AdaBoost 和匹配追踪的选择性集成算法

姚 旭[1,2], 王晓丹[1], 张玉玺[1], 雷 蕾[1]

(1. 空军工程大学 防空反导学院，西安 710051；2. 93767部队，河北 张家口 075000)

摘 要: 为了平衡集成学习中差异性和准确性的关系并提高学习系统的泛化性能, 提出一种基于 AdaBoost 和匹配追踪的选择性集成算法. 其基本思想是将匹配追踪理论融合于 AdaBoost 的训练过程中, 利用匹配追踪贪婪迭代的思想来最小化目标函数与基分类器线性组合之间的冗余误差, 并根据冗余误差更新 AdaBoost 已训练基分类器的权重, 进而根据权重大小选择集成分类器成员. 在公共数据集上的实验结果表明, 该算法能够获得较高的分类精度.

关键词: 选择性集成；AdaBoost 算法；匹配追踪；差异性

中图分类号: TP391 **文献标志码:** A

Selective ensemble algorithm based on AdaBoost and matching pursuit

YAO Xu[1,2], *WANG Xiao-dan*[1], *ZHANG Yu-xi*[1], *LEI Lei*[1]

(1. School of Air and Missile Defense，Air Force Engineering University，Xi'an 710051，China；2. The Army of 93767，Zhangjiakou 075000，China. Correspondent：YAO Xu，E-mail：ffxy132@163.com)

Abstract: To balance the diversity and the accuracy in ensemble learning and improve the generalization performance of learning system, a selective ensemble algorithm based on AdaBoost and matching pursuit is proposed. In the algorithm, matching pursuit is fused into the training of AdaBoost, in which the residual between the target function and the linear combination of basis classifiers is minimized with a greedy iterative idea. Then the weight of each basis classifier is updated by the residual which is generated during the last iteration and then the optimal weights for every classifier are gained, by which the component classifiers are selected. Experimental results on common data sets show that the algorithm can get higher classification accuracy.

Key words: selective ensemble；AdaBoost algorithm；matching pursuit；diversity

0 引 言

集成学习是将多个不同的单个模型组合成一个模型, 其目的是利用这些单个模型之间的差异来改善模型的泛化性能. 早在 1997 年, 国际机器学习界的权威人士 Dietterich 就将集成学习列为机器学习四大研究方向之首[1]. 近年来, 人们提出了很多创新性的方法[2-6]. 然而, 使用大量的个体学习器虽然能够获得更好的性能, 但也增加了更大的计算和存储开销, 而且个体学习器的差异性也越来越难以获得. 2002 年, 周志华等[7]提出了"选择性集成"的概念, 并证明通过选择部分个体学习器进行集成可能比使用全部个体学习器进行集成效果更好. 针对如何选择部分个体学习器, 已有很多代表性的方法, 如: 周志华等[7]使用遗传算法进行选择; Rokach[8]提出了 CAP (Collective-agreement-based pruning) 选择性集成学习算法; 张春霞等[9]采用 Boosting 思想进行选择; 杨晓霜等[10]提出了基于 Moore-Penrose 逆矩阵的选择性集成学习算法; 杨长盛等[11]提出了一种基于成对差异性度量的选择性集成方法; 方育柯等[12]提出了一种选择性 Boosting 集成学习算法; Mao 等[13]提出了一种基于差异性的贪婪优化方法; Li 等[14]提出了基于差异性正则化的选择性集成方法. 此外, 张春霞等[15]对现有的选择性集成学习算法进行了详细综述, 按照算法采用的选择策略进行了分类, 并分析了各种算法的主要特点, 为集成学习的研究提供了宝贵的信息资源.

AdaBoost[16]算法是当前最流行的集成学习算法之一. 由于 AdaBoost 的差异性是基于样本扰动的, 对于决策树、神经网络等不稳定分类器集成效果显著. 而对于支持向量机等稳定的学习算法, 训练数据集中的变化只在分类器上引起很小的变化, 以此作为基分

收稿日期: 2012-10-06；**修回日期:** 2013-03-21.

基金项目: 国家自然科学基金项目(60975026, 61273275).

作者简介: 姚旭(1982−), 女, 博士, 从事智能信息处理和机器学习的研究；王晓丹(1966−), 女, 教授, 博士生导师, 从事智能信息处理、机器学习等研究.

类器的集成系统的分类性能、泛化性能如何是近年来备受人们关注的问题[17]. 同时, 尽管当前 AdaBoost 算法的研究使其收敛速度达到了对数级的水平, 但也存在着缺陷. 如 AdaBoost 对噪声数据比较敏感, 虽然在生成基分类器的同时删除了部分基分类器 (如错误率大于 0.5), 但由于集成算法本身的局限性和数据分布的复杂性 (尤其是在噪声点或者难分样本点存在时), 生成的基分类器之间仍然存在较大的相关性和冗余信息, 用大量的基分类器进行集成容易造成过拟合.

针对 AdaBoost 算法的上述缺陷, 并受文献 [13] 的启发, 本文引入匹配追踪算法, 提出一种基于 AdaBoost 和匹配追踪的选择性集成算法, 旨在基分类器的准确性和差异性之间寻求折衷, 最终提高集成系统的性能. 算法的基本思想是利用匹配追踪贪婪迭代, 通过不断最小化预设的目标函数与 AdaBoost 已训练基分类器的线性组合之间的冗余误差来获得每一个基分类器的权重, 从而得到一个最优权重向量, 进而依据权重大小选择参与集成的基分类器集合. 为了验证所提出算法的性能, 分别以稳定的学习算法支持向量机 (SVM) 和不稳定的学习算法决策树 (DT) 为分类器, 在公共数据集上进行仿真实验, 取得了较好的结果.

1　匹配追踪算法

匹配追踪算法是 Mallat 等[18]于 1993 年提出的. 基本的匹配追踪算法原理如下[13]: 令 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ 和 $\{y_1, y_2, \cdots, y_N\}$ 分别表示样本集 D 的观测值和类标签. 设 H 为一个 Hilbert 空间, $S = \{s_1, s_2, \cdots, s_M\}$ 为 H 中的一组函数构成的函数库. 假设 $q \in H$ 为目标函数, 每一个样本 $\boldsymbol{x}_i\,(i = 1, 2, \cdots, N)$ 对应着一个类标签 $y_i\,(i = 1, 2, \cdots, N)$, 则匹配追踪算法的目的是在函数库 S 中寻找 q 的一个稀疏逼近, 即

$$q' = \sum_{i=1}^{n} \alpha_i g_i. \tag{1}$$

其中: $g_i \subset S\,(i = 1, 2, \cdots, n)$ 为基函数, $\alpha_i \in R^n (i = 1, 2, \cdots, n)$ 为对应的基函数的系数, q' 是由函数库 S 中 n 个不同的基函数组成的 q 的一个近似. 于是二者的冗余误差为

$$\|\boldsymbol{R}_n\|^2 = \|\boldsymbol{q} - \boldsymbol{q}'\|^2 = \sum_{i=1}^{N} (y_i - q'(x_i))^2. \tag{2}$$

在匹配追踪算法中, 为了最小化 $\boldsymbol{R}_n$, 贪婪迭代方法被用来选择基函数 $\{g_1, g_2, \cdots, g_n\}$ 和它们对应的系数 $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$. 系数 α 的计算公式为

$$\alpha = \langle \boldsymbol{g}, \boldsymbol{R}_n \rangle / \|\boldsymbol{g}\|^2. \tag{3}$$

由上面的描述可知, 匹配追踪算法即是利用贪婪迭代的思想, 寻找一组能够最小化冗余误差的基函数. 下面将介绍如何根据匹配追踪算法对基分类器进行选择性集成.

2　基于 AdaBoost 和匹配追踪的选择性集成算法

2.1　算法描述

在集成学习中, 分类器之间的差异性和基分类器的准确性是两个重要因素. 对于集成差异性的定义目前仍未统一, 人们最普遍接受的定义是各个分类器对待测样本做出不同错误的趋势[19]. 2000 年, 周志华等[20]给出了集成系统的泛化误差与差异性和准确性的关系, 如下式所示:

$$\mathrm{Error} = \overline{\mathrm{Error}} - \overline{D}. \tag{4}$$

其中: Error 为集成系统的泛化误差, $\overline{\mathrm{Error}}$ 为个体分类器的平均错误率, $\overline{D}$ 为分类器之间的差异性. 因此, 在增加差异性的同时保证基分类器的准确性, 便可提高集成系统的性能. 然而, 差异性和准确性本身是一个矛盾体, 即差异性的增加一般是以牺牲准确性为代价的, 如何在二者之间找到一个平衡是集成学习研究的重点.

本文提出一种基于 AdaBoost 和匹配追踪的选择性集成算法, 简称 AMPSEN. 该算法的基本思想是: 将匹配追踪理论融合到 AdaBoost 算法的训练过程中, 利用匹配追踪算法贪婪迭代的思想, 不断地最小化目标函数与 AdaBoost 已训练基分类器线性组合之间的冗余误差, 并更新基分类器的权重, 进而得到所有基分类器的一个最优权重向量; 最后根据权重大小, 在不降低基分类器间差异性的情况下, 从 AdaBoost 生成的基分类器中去除一些冗余的基分类器 (这一点将在 2.2 节中详细介绍). 在该算法中, 冗余误差 R_0 初始化为训练集的真实类标签, 基分类器的决策函数 h 作为基函数. 因此, 通过最小化冗余误差, 便可得到一个基函数的最优线性组合. 以这些基分类器组成集成系统对待测样本进行分类, 便能得到与样本真实类标签最相近的分类结果, 该集成效果是最好的. 下面对算法的实现过程进行详细描述.

初始化最大的权重系数 $\alpha_{\max} = \langle \boldsymbol{h}, \boldsymbol{R}_0 \rangle / \|\boldsymbol{h}\|^2$. 对于第 t 次迭代操作, 每个基分类器的权重需要根据第 $t-1$ 次的冗余误差进行更新, 权重和冗余误差的更新公式分别为

$$\alpha_{\max} = \langle \boldsymbol{h}_t, \boldsymbol{R}_t \rangle / \|\boldsymbol{h}_t\|^2, \tag{5}$$

$$R_t = R_{t-1} - \alpha_t h_t. \tag{6}$$

式 (5) 中, 第 t 次迭代中所有的基分类器权重根据第 $t-1$ 次的冗余误差进行更新. 如果所有基分类器权重的最大值 $\max\{\alpha_1, \cdots, \alpha_{t-1}, \alpha_t\}$ 大于 $\alpha_{\max}$, 则更新 $\alpha_{\max} = \max\{\alpha_1, \cdots, \alpha_{t-1}, \alpha_t\}$, 并且 $\alpha_t = \alpha_{\max}$. 由式

(5) 和 (6) 可以看出, 随着迭代次数的增加, 冗余误差越来越小, 即基分类器的线性组合越来越接近于目标值. 算法的具体步骤如下.

输入: 样本集 $D=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),\cdots,(\boldsymbol{x}_N,\boldsymbol{y}_N)\}$, 迭代次数 T, 选择的基分类器个数 L;

输出: 总体分类器的判决函数值

$$H(\boldsymbol{x})=\mathrm{sign}\Big[\sum_{i=1}^{L}\alpha_i h_i(\boldsymbol{x})\Big].$$

Step 1: 初始化各样本对应的权值 $w_1(i)=1/N$, $i=1,2,\cdots,N$; 初始化冗余误差 $R_0=\{y_1,y_2,\cdots,y_N\}$.

Step 2: 依据 w_1 训练基分类器 C_0 得到 h_0, 计算 $\alpha_0=\langle \boldsymbol{h}_0,\boldsymbol{R}_0\rangle/\|\boldsymbol{h}_0\|^2$.

Step 3: for $t=1$ to T

1) 依据 w_t 抽取训练集 D_t, 训练基分类器 C_t 得到 h_t.

2) for $m=1$ to t

计算基分类器 C_t 的权重 $\alpha_m=\langle \boldsymbol{h}_m,\boldsymbol{R}_{t-1}\rangle/\|\boldsymbol{h}_m\|^2$;

End for

令 $\alpha_{\max}^t=\max\limits_{m}\{\alpha_m\}$, 如果 $\alpha_{\max}^t>\alpha_{\max}$, 则 $\alpha_{\max}=\alpha_{\max}^t$.

3) 计算分类器的错误率 $\varepsilon_t=\sum\limits_{i=1}^{N}w_t(i),\ y_i\neq h_t(\boldsymbol{x}_i)$.

4) 如果 $\varepsilon_t=0$ 或 $\varepsilon_t\geqslant 0.5$, 则重置样本权重 $w_t(i)=1/N$, $i=1,2,\cdots,N$, 转 1).

5) 令 $\alpha_t=\alpha_{\max}$, 更新冗余误差 $R_t=R_{t-1}-\alpha_t h_t$.

6) 更新训练样本的权值

$$w_{t+1}(i)=\frac{w_t(i)\exp\{-\alpha_t y_i h_t(\boldsymbol{x}_i)\}}{Z_t}=\frac{w_t(i)}{Z_t}\times\begin{cases}\mathrm{e}^{-\alpha_t},\ \boldsymbol{y}_i=h_t(\boldsymbol{x}_i);\\ \mathrm{e}^{\alpha_t},\ \boldsymbol{y}_i\neq h_t(\boldsymbol{x}_i).\end{cases}$$

其中 Z_t 为归一化系数, 使得 $\sum\limits_{i=1}^{N}w_{t+1}(i)=1$.

End for.

Step 4: 将权重向量 $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_T)$ 按降序排列, 选择前 L 个基分类器作为集成系统的成员.

Step 5: 对测试样本 $\boldsymbol{x}$ 融合各分类器的输出结果, 有

$$H(\boldsymbol{x})=\mathrm{sign}\Big[\sum_{i=1}^{L}\alpha_i h_i(\boldsymbol{x})\Big].$$

2.2 算法分析

对于如 Bagging、Boosting 等传统的集成策略, 差异性是通过在不同的训练集上训练基分类器构造的. 但是仅仅扰动训练集有时并不能保证生成的基分类器之间彼此互不相关. 本文提出的 AMPSEN 算法是基于基分类器间的差异性, 对 AdaBoost 算法生成的基分类器进行选择性集成. 下面针对算法的差异性进行分析.

由 2.1 节中的算法介绍可知, 该算法是基于最小化冗余误差进行搜索的, 权重系数和冗余误差根据式 (5) 和 (6) 进行更新. 因此在第 $t+1$ 次迭代中, 基函数 h_i 的权重系数 α_i 可通过下式计算:

$$\alpha_i=\langle \boldsymbol{h}_i,\boldsymbol{R}_t\rangle/\|\boldsymbol{h}_i\|^2. \tag{7}$$

根据式 (6), 式 (7) 可以表示为如下形式:

$$\alpha_i=\frac{\langle \boldsymbol{h}_i,\boldsymbol{R}_t\rangle}{\|\boldsymbol{h}_i\|^2}=\frac{\langle \boldsymbol{h}_i,\boldsymbol{R}_{t-1}\rangle-\alpha_t\boldsymbol{h}_i}{\|\boldsymbol{h}_i\|^2}=\frac{\langle \boldsymbol{h}_i,\boldsymbol{R}_{t-1}\rangle}{\|\boldsymbol{h}_i\|^2}-\frac{\langle \boldsymbol{h}_i,\alpha_t\boldsymbol{h}_t\rangle}{\|\boldsymbol{h}_i\|^2}=\frac{\langle \boldsymbol{h}_i,\boldsymbol{R}_{t-1}\rangle}{\|\boldsymbol{h}_i\|^2}-\alpha_t\frac{\langle \boldsymbol{h}_i,\boldsymbol{h}_t\rangle}{\|\boldsymbol{h}_i\|^2}. \tag{8}$$

其中: h_t 为权重系数 α_t 对应的基函数, $\{h_i\}$ 为第 $t+1$ 次迭代中所有基函数的集合. 当 h_i 与 h_t 相似时, 可得

$$\begin{cases}\lim\limits_{h_i\to h_t}\langle \boldsymbol{h}_i,\boldsymbol{R}_{t-1}\rangle/\|\boldsymbol{h}_i\|^2=\alpha_t,\\ \lim\limits_{h_i\to h_t}\langle \boldsymbol{h}_i,\boldsymbol{h}_t\rangle/\|\boldsymbol{h}_i\|^2=1.\end{cases} \tag{9}$$

由式 (8) 和 (9) 可得

$$\lim_{h_i\to h_t}\alpha_i=0. \tag{10}$$

在 AMPSEN 算法中, 匹配追踪算法中的函数库是由所有基分类器构成的, 即每一个基分类器代表一个基函数. 由上面的分析可知, 当两个基函数 h_i 和 h_t 相关性很大时, 在第 $t+1$ 次迭代中 h_i 获得的权重近似为 0, 因此在依据权重选择基分类器的过程中, h_i 被选的概率基本为 0. 反之, 当二者差异性很大时, h_i 将获得较大的权重系数. 因此, AMPSEN 算法中基分类器的选择是基于差异性的, 它通过为基分类器分配近似为 0 的系数来去除一些冗余信息. 同时, 在优化迭代过程中, 初始化冗余误差为真实类标签, 随着迭代次数的增加, 不断地最小化冗余误差, 使得基分类器间的线性组合越来越接近真实类标签. 因此, 算法中差异性的增加并不以牺牲准确性为代价.

综上所述, AMPSEN 算法具有以下优点: 1) 集成分类器成员的选择基于一个优化过程, 每一个基分类器的权重系数根据上一次迭代的结果进行自动更新, 保证了最终的组合最接近于目标函数; 2) 算法基于分类器之间的差异性进行选择, 且在增加差异性的同时保证了基分类器的准确性, 最终提高了集成的性能.

3 实验结果及分析

3.1 实验数据

实验中的数据来自 UCI 数据库和 Statlog 数据库[21], 实验选择了其中 12 组数据 (特征维数范围为 8 ~ 60, 样本范围为 208 ~ 20 000), 关于实验数据的详细描述如表 1 所示.

表 1　公共数据集各数据描述

数据集	样本数	维数	类别	来源
Sonar	208	60	2	UCI
Ionosphere	351	34	2	UCI
Diabetes	768	8	2	UCI
Breast-w	699	9	2	UCI
Heart	270	13	2	Statlog
Soybean	307	35	19	UCI
Vehicle	846	18	4	Statlog
Segment	2 310	19	7	UCI
Glass	214	10	7	UCI
Letter	20 000	16	26	Statlog
Satimage	6 435	36	6	UCI
Pendigits	10 992	36	6	UCI

在估计分类错误率时为保证估计的准确性, 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 重交叉验证来进行, 并利用双边估计 t 检验法计算置信水平为 0.95 的分类错误率置信区间得到最终结果, 计算公式如下:

$$\frac{|\bar{x}-\mu|}{\sigma/\sqrt{n}} \geqslant t_{0.025}(n-1). \tag{11}$$

其中: μ、σ 分别表示 n 重交叉验证的均值和标准差, $t_{0.025}(4)=2.776\,4$, $t_{0.025}(9)=2.262\,2$. 实验中成员分类器均来自 PRTool (http://www.prtools.org) 工具箱. 支持向量机分类器采用径向基核函数的 SVM, 其中参数 $C=1\,000$, σ 按文献 [17] 选取. 实验机器配置为 2 G 内存, 2.80 G CPU, 算法基于 Matlab 7.10 (R2010a) 实现.

3.2 实验结果和分析

为了验证本文算法 AMPSEN 的性能, 在 UCI 数据集和 Statlog 数据集上进行实验, 并与单分类器 Single、Bagging、AdaBoost、随机子空间方法 (RSM) 和 GASEN[7] 几种集成方法进行比较. 实验选取 SVM 和决策树作为基分类器, 从收敛性分析、分类误差和差异性分析 3 个角度对本文提出的算法进行验证.

3.2.1 算法的收敛性能分析

为了表明本文算法的收敛性能, 实验给出了在 Sonar、Heart-statlog、Soybean、Segment 四个数据集上分别以 SVM 和决策树为基分类器时迭代次数与训练误差的关系, 如图 1 所示. 从图 1 的实验结果可以看出, 随着迭代次数的增加, 训练误差明显降低, 虽然在有些数据集上, 当误差达到最小值时偶尔有上升的浮动, 但最终都趋于稳定. 从实验结果可以看出, 当迭代次数达到 20 次时, 测试误差已经达到平稳值. 因此在实验中设置选择的基分类器个数 $L=20$.

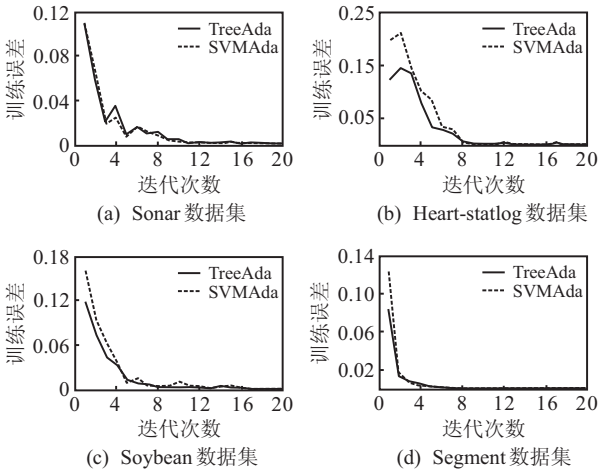

图 1　不同数据集上迭代次数与收敛性能的关系

3.2.2 分类误差的比较

实验中采用交叉验证法来估计分类误差. 在每个数据集上进行 10 次实验, 实验结果取 10 次实验的平均值. Bagging、AdaBoost、RSM 的集成规模均为 50, GASEN 中遗传算法的迭代次数为 100, AMPSEN 算法中 $T=50$. 实验结果如表 2 和表 3 所示. 其中表 2 的基分类器为 SVM, 表 3 的基分类器为决策树, 各个数据集上最小的分类误差在表中用黑体表示.

分析表 2 可以看出, 当选取 SVM 为分类器时, 除了在 Diabetes、Heart-statlog 和 Satimage 三个数据集上, 本文提出的算法均获得了最小的分类误差. 相对于单分类器, 几种集成方法 Bagging、AdaBoost、RSM、GASEN 和本文算法的平均分类误差分别降低了 1.52%、2.51%、1.93%、3.66%、4.87%. 从表 2 的实验结果还可以看出, 当以 SVM 这种稳定的学习算法为基分类器时, 仅依靠对训练集进行扰动构造差异性的 Bagging 集成方法与单分类器相比没有显著的优势.

分析表 3 可以看出, 当选取决策树为分类器时, 除了在 Diabetes、Vehicle 和 Letter 三个数据集上, 本文提出的算法均获得了最小的分类误差. 相对于单分类器, Bagging、AdaBoost、RSM、GASEN 和本文算法的平均分类误差分别降低了 2.62%、3.84%、2.90%、4.88%、5.92%. 从表 3 中还可看出, 对于如决策树这样的不稳定学习算法, 基于样本扰动的集成方法的效果是显著的. 通过分析表 2 和表 3 的实验结果可知, 本文方法在大部分数据集上都获得了最小的分类误差, 充分说明了本文提出的基于损失函数的样本权重更新策略和基于匹配追踪思想的选择方法可以有效提高基分类器的准确性和差异性, 从而提高了集成系统的

表 2 基于 SVM 的正确率及置信水平为 0.95 的置信区间 %

Dataset	Single	Bagging	AdaBoost	RSM	GASEN	AMPSEN
Sonar	76.54±1.65	76.10±3.80	77.68±2.52	76.61±3.40	78.15±3.02	**80.49±2.62**
Ionosphere	86.18±1.70	87.75±1.10	88.74±1.88	87.57±2.65	91.19±2.64	**92.92±2.06**
Diabetes	75.43±2.75	76.14±1.35	76.32±1.95	76.03±3.14	**78.23±2.01**	78.22±1.99
Breast-w	92.98±1.87	94.42±1.70	95.55±2.29	95.16±1.44	96.15±1.87	**97.56±1.98**
Heart	71.85±6.02	73.33±6.70	75.19±6.50	76.17±6.37	**76.98±5.79**	76.56±4.91
Soybean	86.76±3.66	91.71±2.23	92.77±2.52	91.43±2.50	92.79±2.39	**94.72±2.09**
Vehicle	79.69±3.19	77.41±1.59	78.19±1.08	77.85±1.33	80.36±1.46	**81.40±1.67**
Segment	95.89±0.91	95.45±0.91	95.93±0.78	95.89±1.18	96.09±1.05	**96.41±0.91**
Glass	72.52±1.08	73.07±1.40	73.61±1.60	72.63±1.57	76.21±1.56	**79.52±1.08**
Letter	68.33±0.49	76.86±0.91	77.82±0.37	76.43±1.18	78.03±1.47	**79.84±1.36**
Satimage	85.32±0.54	86.58±0.81	87.84±0.64	87.52±0.99	**88.69±0.95**	88.67±0.84
Pendigits	93.42±0.21	94.35±0.54	95.48±0.26	94.87±0.35	96.01±0.56	**97.06±0.48**
Average	82.08±2.00	83.60±1.92	84.59±1.87	84.01±2.18	85.74±2.06	**86.95±1.83**

表 3 基于 DT 的正确率及置信水平为 0.95 的置信区间 %

Dataset	Single	Bagging	AdaBoost	RSM	GASEN	AMPSEN
Sonar	67.32±8.20	73.11±6.66	75.08±5.95	74.46±6.57	75.46±7.02	**76.57±6.13**
Ionosphere	89.22±3.38	89.18±2.47	92.30±2.90	91.79±3.57	92.46±3.04	**93.37±3.20**
Diabetes	68.09±4.58	65.77±4.79	69.14±4.85	68.75±4.56	**69.43±3.46**	69.40±2.52
Breast-w	94.71±1.19	95.71±1.52	95.98±1.67	95.85±1.46	96.12±1.98	**96.87±1.69**
Heart	64.81±4.88	65.67±4.50	65.93±3.91	64.07±5.31	66.79±4.56	**69.81±7.09**
Soybean	55.76±1.54	69.02±2.37	70.41±2.12	65.56±2.15	71.32±2.36	**75.37±1.67**
Vehicle	70.25±3.89	73.63±1.90	73.01±2.22	73.28±2.86	**75.42±2.34**	74.98±1.73
Segment	94.32±1.37	95.88±1.24	96.04±0.98	95.94±1.32	96.25±1.35	**96.83±1.06**
Glass	73.05±1.44	74.66±2.18	75.59±1.62	74.67±2.30	77.98±2.34	**79.10±2.38**
Letter	72.72±0.47	74.86±0.87	77.05±0.53	76.71±0.36	**80.37±0.79**	79.62±0.67
Satimage	86.54±2.87	88.84±3.65	90.05±2.30	89.19±2.21	91.23±2.31	**92.05±2.16**
Pendigits	93.67±0.87	95.65±0.25	95.98±0.24	95.04±0.35	96.25±0.31	**97.59±0.32**
Average	77.54±2.89	80.16±2.70	81.38±2.44	80.44±2.75	82.42±2.66	**83.46±2.55**

性能. 同时实验结果也表明, 基于样本扰动的集成方法对于不稳定的学习算法效果较为显著, 并且由于基分类器之间存在冗余或无用的信息, 进行选择性集成是必要的.

为了使实验结果更为清晰, 用统计的观点对文中所涉及分类算法的相对性能进行分析, 同时利用文献[22]中提出的一些统计量进行讨论. 表 4 和表 5 给出了在所有数据集上, 每种方法的误差比较. 表 4 和表 5 中的最后一行是每种方法的误差在所有数据集上的平均值. 如果用“row”表示表中每一行所列算法的误差, “col”表示表中每一列所列算法的误差, 则表 4 和表 5 中“$\dot{r}$”一行的值表示“row/col”的几何平均值; “s”对应的行给出的是 win/tie/loss 统计量, 其中的 3 个值分别表示 col<row, col = row, col>row 的数据集个数.

从表 4 和表 5 的实验结果可以看出, 无论基分类器选择 SVM 还是决策树, AMPSEN 算法都具有最小的平均误差. 分析平均误差、“$\dot{r}$”和“s”三个统计量可以看出, 6 种方法按照分类效果由好到差依次为 AMPSEN、GASEN、AdaBoost、RSM、Bagging、Single. 单独考虑 AMPSEN 的分类效果, 它与 Single 的误差比率的几何平均值比其他方法相对于 Single 的误差比率的几何平均值都小; 同时, 与 Bagging、AdaBoost、

表 4 基于 SVM 的各数据集上误差 (Error) 比较

Algorithm		Single	Bagging	AdaBoost	RSM	GASEN	AMPSEN
Single	$\dot{r}$		0.901	0.820	0.866	0.753	0.650
	s		9/0/3	11/0/1	10/1/1	12/0/0	12/0/0
Bagging	$\dot{r}$			0.910	0.961	0.836	0.722
	s			12/0/0	7/0/5	12/0/0	12/0/0
AdaBoost	$\dot{r}$				1.055	0.918	0.793
	s				1/0/11	12/0/0	12/0/0
RSM	$\dot{r}$					0.870	0.751
	s					12/0/0	12/0/0
GASEN	$\dot{r}$						0.864
	s						9/0/3
Error/%		17.92	16.40	15.41	15.99	14.26	**13.05**

表 5 基于决策树的各数据集上误差 (Error) 比较

Algorithm		Single	Bagging	AdaBoost	RSM	GASEN	AMPSEN
Single	$\dot{r}$		0.856	0.790	0.837	0.742	0.664
	s		10/0/2	12/0/0	11/0/1	12/0/0	12/0/0
Bagging	$\dot{r}$			0.922	0.977	0.867	0.775
	s			11/0/1	8/0/4	12/0/0	12/0/0
AdaBoost	$\dot{r}$				1.060	0.940	0.841
	s				1/0/11	12/0/0	12/0/0
RSM	$\dot{r}$					0.887	0.793
	s					12/0/0	12/0/0
GASEN	$\dot{r}$						0.894
	s						9/0/3
Error/%		22.46	19.84	18.62	19.56	17.58	**16.54**

RSM、GASEN 的误差比率的几何平均值也表明本文算法的效果比较好. 此外, 与其他几种方法相比, AMPSEN 在更多的数据集上具有较小的分类误差, 从而表明该算法是有效的.

3.2.3　差异性比较

为了得出更具统计意义的实验结论, 利用秩和检验法对上面的结果进行分析. 秩水平计算如下:

$$R_j = \frac{1}{J}\sum_i r_i^j. \tag{12}$$

其中: r_i^j 为在第 i 个数据集上用第 j 种方法所得到的秩的大小; J 为每种方法所进行的实验次数, 文中 $J = 10$. 表 6 给出了每种方法误差的秩和平均数.

表 6　各种方法误差秩和平均数比较

Algorithm	Single	Bagging	AdaBoost	RSM	GASEN	AMPSEN
Error(SVM)	6.6	5.8	3.8	5.3	2.1	**1.5**
Error(DT)	6.9	5.7	3.8	5.2	2.1	**1.5**
Global	6.75	5.75	3.8	5.25	2.1	**1.5**

从表 6 可以看出, AMPSEN 所得到的秩和平均数最小, GASEN 次之, Single 最大. 为了验证这几种方法的分类效果具有统计意义上的显著差别, 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如下临界值[23]时:

$$\mathrm{CD} = q_\alpha\sqrt{\frac{k(k+1)}{6J}}. \tag{13}$$

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.

在本实验中比较了 6 种方法在置信水平为 $\alpha = 0.05$ 下的分类效果, 即 $k = 6$, $q_{0.05} = 1.860$, 代入式 (13) 可得差异临界值 (CD) 为 1.556. 观察表 6 可知, AMPSEN 的秩平均数比其余方法秩平均数都要小. AMPSEN 算法与 Single、Bagging、AdaBoost、RSM、GASEN 的秩和平均差分别为 5.25、4.25、2.30、3.75、0.6, 除了 GASEN, 均大于差异临界值, 因此本文算法的分类效果与这几种方法具有统计意义上的显著差别. 虽然与 GASEN 差别不大, 但是本文方法与 GASEN 相比具有较高的分类正确率和较低的运行时间, 因此本文算法是可行的, 它能够有效地提高集成系统的性能.

为了使实验结果更直观, 这里引入 Kappa-Error 图[24]来分析每种集成方法中基分类器之间的差异性. Kappa-Error 图为分析集成分类器提供了一种可视化方法. 图中每个点对应一对基分类器, 其位置由两个基分类器在验证集上的平均分类错误率和一致性度量统计量 Kappa 决定, 横轴为 Kappa 的值, 纵轴为平均识别错误率. Kappa 值和分类错误率越小, 该方法的性能越好. 因此, Kappa 图位于第一象限左下方的方法性能最好. 图 2 和图 3 给出了 Sonar 数据集上以 SVM 和决策树为基分类器的 Kappa-Error 图, 每个数据集所对应的 4 种集成方法的 Kappa-Error 图上的坐标刻度一致, 以便于直观比较结果.

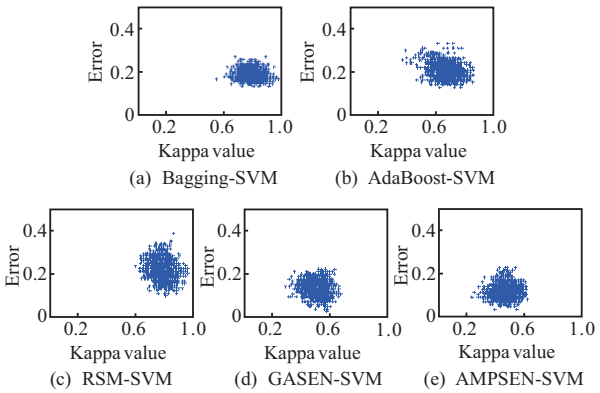

图 2　基于 SVM 的集成方法在“Sonar”数据集的 Kappa-Error 图

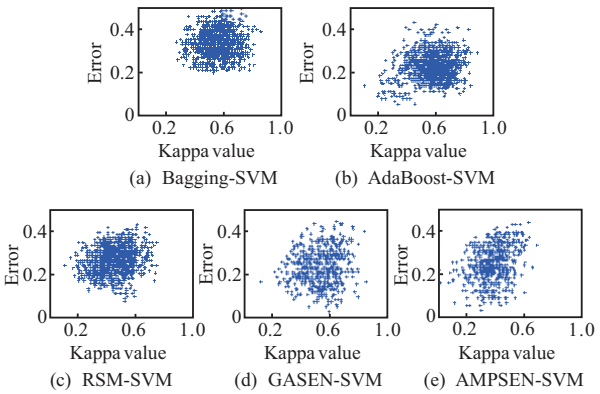

图 3　基于决策树的集成方法在“Sonar”数据集的 Kappa-Error 图

分析图 2 和图 3 可以看出, AMPSEN 方法在分类误差和差异性方面都是几种集成方法中最好的. AMPSEN 方法和 GASEN 方法差异性相差不大, 二者与 Bagging、AdaBoost、RSM 三种传统集成策略相比, 在差异性和准确率方面有着更大的优势.

4　结　论

为了去除 AdaBoost 集成算法中的冗余基分类器, 并且平衡集成系统中差异性和准确性的关系, 本文提出了一种基于 AdaBoost 和匹配追踪的选择性集成算法. 该算法将匹配追踪贪婪迭代的寻优过程融合到 AdaBoost 的训练过程中, 通过最小化目标函数 (训练集的真实类标签) 与 AdaBoost 已训练基分类器之间的冗余误差, 为每一个基分类器分配一个权重系数, 最终可以获得所有基分类器的一个最优权重向量, 并根据权重大小选择一部分基分类器参与集成. 此外, 为了改善 AdaBoost 算法在样本权重更新过程中过度专注于难分样本的缺陷, 文中给出了一种新的样本权重更新策略, 实验表明该策略可以保证基分类器训

练的准确性. 同时算法中基分类器的选择是基于成对差异性的, 并且差异性的增加并不以牺牲基分类器的准确性为代价. 为了验证本文所提出算法的有效性, 以SVM和决策树为基分类器, 在UCI数据集和Statlog数据集上进行实验. 实验结果表明, 该算法能够获得低于Bagging、AdaBoost、RSM和GASEN的分类误差. 因此, 本文提出的算法能有效提高集成的性能.

参考文献(References)

[1] Dietterich T G. Machine learning research: Four current directions[J]. AI Magazine, 1997, 18(4): 97-136.

[2] Ho T K. Complexity of classification problems and comparative advantages of combined classifiers[C]. Multiple Classifier Systems. Berlin: Springer, 2000: 97-106.

[3] Kuncheva L I. Evaluation of stability of k-means cluster ensembles with respect to random initialization[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2006, 28(11): 1798-1808.

[4] Garcia-Pedrajas N, Fyfe C. Construction of classifier ensembles by means of artificial immune systems[J]. J of Heuristics, 2008, 14(3): 285-310.

[5] 唐耀华, 高静怀. 一种新的选择性支持向量机集成学习算法[J]. 西安交通大学学报, 2008, 42(10): 1221-1225.
(Tang Y H, Gao J H. Novel selective support vector machine ensemble learning algorithm[J]. J of Xi'an Jiaotong University, 2008, 42(10): 1221-1225.)

[6] Dos Santos E M, Sabourin R, Maupin P. Overfitting cautious selection of classifier ensembles with genetic algorithms[J]. Information Fusion, 2009, 10(2): 150-162.

[7] Zhou Z H, Wu J, Tang W. Ensembling neural networks: Many could be better than all[J]. Artificial Intelligence, 2002, 137(1/2): 239-263.

[8] Rokach Lior. Collective-agreement-based pruning of ensembles[J]. Computational Statistics and Data Analysis, 2009, 53(4): 1015-1026.

[9] Zhang C X, Zhang J S, Zhang G Y. Using boosting to prune double-bagging ensembles[J]. Computational Statistics and Data Analysis, 2009, 53(4): 1218-1231.

[10] Yang X S, Wang Y Y. Selective ensemble based on Moore-Penrose pseudo-inverse[J]. Opto-Electronic Engineering, 2009, 36(11): 140-144.

[11] 杨长盛, 陶亮, 曹振田. 基于成对差异性度量的选择性集成方法[J]. 模式识别与人工智能, 2010, 23(4): 265-571.
(Yang C S, Tao L, Cao Z T. Pairwise diversity measures based selective ensemble method[J]. Pattern Recognition and Artificial Intelligence, 2010, 23(4): 265-571.)

[12] 方育柯, 傅彦, 周俊临, 等. 基于选择性集成的最大化软间隔算法[J]. 软件学报, 2012, 23(5): 1132-1147.
(Fang Y K, Fu Y, Zhou J L, et al. Selective boosting algorithm for maximizing the soft margin[J]. J of Software, 2012, 23(5): 1132-1147)

[13] Mao Sha-sha, Jiao Li-cheng, Xiong Lin, et al. Greedy optimization classifiers ensemble based on diversity[J]. Pattern Recognition, 2011, 44(6): 1245-1261.

[14] Li N, Yu Y, Zhou Z H. Diversity regularized ensemble pruning[C]. Proc of the European Conf on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Bristol, 2012: 330-345.

[15] 张春霞, 张讲社. 选择性集成学习算法综述[J]. 计算机学报, 2011, 34(8): 1400-1410.
(Zhang C X, Zhang J S. A survey of selective ensemble learning algorithm[J]. Chinese J of Computers, 2011, 34(8): 1400-1410.)

[16] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C]. Proc of the 13th Int Conf on Machine Learning. Bari: Morgan Kaufmann, 1996: 148-156.

[17] 王晓丹, 孙东延, 郑春颖, 等. 一种基于AdaBoost的SVM分类器[J]. 空军工程大学学报: 自然科学版, 2006, 7(6): 54-57.
(Wang X D, Sun D Y, Zheng C Y, et al. A combined SVM classifier based on AdaBoost[J]. J of Air Force Engineering University: Natural Science Edition, 2006, 7(6): 54-57.)

[18] Mallat S G, Zhang Z F. Matching pursuit with time-frequency dictionaries[J]. IEEE Trans on Signal Processing, 1993, 41(12): 3397-3415.

[19] Dietterich T G. Ensemble methods in machine learning[C]. Multiple Classifier Systems. Berlin: Springer, 2000: 1-15.

[20] Wu J X, Zhou Z H, Shen X H, et al. A selective constructing approach to neural network ensemble[J]. J of Computer Research and Development(Chinese), 2000, 37(9): 1039-1044.

[21] King R D. Statlog databases[D]. Glasgow: Department of Statistics and Modelling Science, University of Strathclyde, 1992.

[22] Webb G I. MultiBoosting: A technique for combining boosting and wagging[J]. Machine Learning, 2000, 40(2): 159-196.

[23] Demsar J. Statistical comparisons of classifiers over multiple data sets[J]. J of Machine Learning Research, 2006, 7(2): 1-30.

[24] Kuncheva L I, Whitaker C J. Measures of diversity in classifier ensembles andtheir relationship with the ensemble accuracy[J]. Machine Learning, 2003, 51(2): 181-207.

(责任编辑: 李君玲)