软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(5):1132−1147 [doi: 10.3724/SP.J.1001.2012.04064]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于选择性集成的最大化软间隔算法*

方育柯+, 傅 彦, 周俊临, 佘 莉, 孙崇敬

(电子科技大学 计算机科学与工程学院,四川 成都 611731)

Selective Boosting Algorithm for Maximizing the Soft Margin

FANG Yu-Ke+, FU Yan, ZHOU Jun-Lin, SHE Li, SUN Chong-Jing

(School of Computer Science and Engineering, University of Electronic and Science Technology of China, Chengdu 610054, China)

+ Corresponding author: E-mail: liusha.fang@gmail.com, fangyuke@uestc.edu.cn

Fang YK, Fu Y, Zhou JL, She L, Sun CJ. Selective boosting algorithm for maximizing the soft margin. *Journal of Software*, 2012,23(5):1132−1147. http://www.jos.org.cn/1000-9825/4064.htm

Abstract: Research of traditional boosting algorithms mainly focuses on maximizing the hard or soft margin of the convex combination among weak hypotheses. The weak learners are often all used in the combination, even though some of them are more, or less related. This increases the time complexity of the hypotheses' training and test. To ease the redundancies of the base hypotheses, this paper presents a selective boosting algorithm called SelectedBoost for classifying binary labeled samples, which is based on LPBoost. The main idea of the algorithm is to discard as many hypotheses as possible according to their relevance and diversity. Furthermore, this paper introduces an edge constraint for every strong hypothesis to speed up the convergence when maximizing the soft margin of the combination of the weak hypotheses. The experimental results show that this algorithm can achieve both better performance and less generalization error compared to some representative boosting algorithms.
Key words: boosting learning; selective boost; soft margin; correlation; linear programming

摘 要: 当前,boosting 集成学习算法研究主要集中于最大化弱学习器凸组合的间隔或软间隔,该凸组合几乎使用了生成的所有弱学习器,然而这些弱学习器间存在大量的相关性和冗余,增加了训练和分类过程的时空复杂度.针对这一问题,在 LPBoost 基础上提出了一种选择性 boosting 集成学习算法,称为 SelectedBoost.在每次迭代生成新的弱学习器以后,通过计算新生成的弱学习器与已有弱学习器的相关度和差异度,并结合当前集成的强学习器的准确率来判断是否选择该弱学习器.另外,当前的一系列 boosting 算法(如 AdaBoost,LPBoost,ERLPBoost 等),本质上是基于已生成的 1 个或者多个弱学习器来更新样本权重,但与弱学习器相比,强学习器更能代表当前的决策面.因此,SelectedBoost 通过在带约束的间隔最大化问题中引入更加严格的强学习器边界约束条件,使得该算法不仅参考弱学习器边界,同时还参考已生成的强学习器来更新样本权重,进而提高算法的收敛速度.最后,与其他有代表性的集成学习算法进行实验比较,结果表明,该方法在收敛率、分类准确性以及泛化能力等方面均具有比较明显的优势.
关键词: 集成学习;选择性集成;软间隔;相关度;线性规划
中图法分类号: TP181 **文献标识码**: A

* 基金项目: 国家自然科学基金(60903073, 60973120, 61003231); 四川省科技攻关项目(2008GZ0009)
收稿时间: 2010-09-10; 修改时间: 2011-01-20, 2011-04-28, 2011-05-18; 定稿时间: 2011-06-20

尽管当前 boosting 集成学习算法的研究使其收敛速度达到了对数级的水平,仍存在两个方面的问题:

(1) boosting 集成学习算法虽然已经在生成弱学习器的同时删除了部分的弱学习器(如错误率大于 0.5),但是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),这些生成的弱分类器之间仍然存在较大的相关性和冗余信息.Duangsoithong,Windeatt 和 Kuncheva 等人[1−3]认为,弱分类器间的差异性和相关性对集成效果的影响至关重要.周志华等人[4−6]的研究结果表明,从已生成的弱学习器中选择之后再集成,可以获得更好的性能.换句话说,为了达到期望的性能,未必使用更多的学习器,而应减少分类器间的相关性同时增加差异性.

(2) AdaBoost 系列算法样本权重更新是基于上次迭代生成的弱分类器,而当前流行的间隔最大化算法(如 LPBoost,TotalBoost,SoftBoost,ERLPBoost 等)虽然是基于全局多个弱分类器更新样本权重,但与弱分类器相比,强分类器更能代表当前样本分布的决策超平面,因此更应该基于强分类更新样本权重.

从这两个思路出发,本文提出了选择性集成学习算法(SelectedBoost),其主要贡献是:

(1) 提出在 boosting 生成弱学习器的同时进行弱学习器的选择,这样大大提高了 boosting 生成所有弱学习器以后再选择集成的效率.具体方法是,基于 LPBoost 算法,在生成弱分类器的同时,使用相关性评价指标选择性地删除相关度较高的弱分类器,从而有效地减少了生成的弱分类器个数,并降低了整体弱分类器间的相关性,增加了最终分类器间的多样性.这不仅在一定程度上提高了收敛速度,而且使最终的分类结果准确率有了进一步的提高.

(2) SelectedBoost 通过在带约束的间隔最大化问题中引入更加严格的强分类器边界约束条件,使得样本权重的更新不仅仅参考已生成的弱分类器边界,而且还参考了当前的强分类边界信息,从而进一步提高了 SelectedBoost 收敛速度.

本文第 1 节介绍 boosting 集成学习理论中间隔最大化算法研究的相关工作.第 2 节给出基本符号定义与 LPBoost 相关概念.第 3 节分析之前 boosting 算法(如 LPBoost,SoftBoost)存在的问题,并针对这些问题给出相应的解决方案,进一步提出选择性集成学习算法 SelectedBoost.第 4 节具体给出算法伪代码.第 5 节针对 AdaBoost 以及当前有代表性的集成学习算法进行深入的实验比较.第 6 节进行总结,并讨论未来进一步的研究方向.

1 相关工作

集成学习理论来源于 PAC(probably approximately correct)[7,8]学习理论,其研究的主要问题是:一种仅仅比随机猜测稍好一点的弱学习算法是否可以通过提升,达到一个任意精度的强学习算法.Freund 和 Schapire[9−11]提出的 AdaBoost 学习算法是第一种可以实用的 boosting 学习算法.它是一种有效的分阶段优化算法,即从一系列容易得到的弱学习器出发,通过线性组合方式集成为一个强学习器.可以证明,AdaBoost 最小化误差上界,此上界为训练集上间隔的指数函数[12,13],并且 AdaBoost 通过最小化指数和来确定弱分类器权重.许多 boosting 学习算法变种都是通过修改目标函数来确定弱弱学习器的权重.

当前,boosting 学习算法的研究热点主要集中于生成的弱分类器的线性组合与最大化间隔的关系,以及在最大化间隔的同时降低集成学习迭代界[14−18].Schapire,Freund 和 Rätsch 等人[15,19]指出,AdaBoost 可以生成一个具有较大间隔(margin)的弱学习器的组合,但是没有最大化硬间隔(hard margin).基于这个发现,产生了很多最大化间隔的 Boosting 算法[14,17,18,20,21].Grove 等人[20]通过理论分析发现,间隔最大化问题可以使用线性规划理论求解其对偶问题得出最优解,并提出了 LPBoost 算法.Warmuth 和 Rätsch 等人[21]认为,AdaBoost 关于样本权重的更新是一种基于上一次迭代的局部性更新算法,并提出了基于全局的样本权重更新算法 TotalBoost.主要使用熵投影,使得新的样本权重分布与之前生成的所有弱分类器正交,同时最小化相对熵以保证权值分布更新的平滑性,使得在 $\frac{2\ln(N)}{\delta^2}$ 迭代界内以δ的准确率最大化硬间隔(hard margin).这里,N 为样本个数.AdaBoost*[18]通过间隔最大化理论构造样本权值和弱分类器权值更新算法,达到了和 TotalBoost 同样的迭代界.

另外,在考虑噪声点或者难分样本点情况下,上述的间隔最大化算法(如 LPBoost,TotalBoost 等)就会牺牲大部分样本的间隔来增加噪声点或难分样本点的间隔,导致过拟合现象[14,20,22].基于此,产生了一系列的软间隔算法.为了抑制过分集中于难分类样本点(或者噪声点)的问题,使得训练集中的一部分噪声样本或者难分样本间隔至少为ρ或者在ρ的值上取折中.这类算法主要包括带 soft margin(忽略难分样本点的 margin)的 Adaboost[19],BrownBoost[23],MadaBoost[24],v-arc[25,26],SmoothBoost[27],SoftBoost[28]和 ERLPBoost[29]等等.在高噪声环境下,这些算法与最初的 AdaBoost 算法相比有较为显著的性能提高.其中,MadaBoost[24]和 SmoothBoost[27]只是与最大化软间隔有关,但其迭代次数与其他因素有关,很难估计何时能够收敛到最大软间隔(maximum soft margin);而 SoftBoost 的最优化问题是较吸引人的,因为它直接最大化训练集子集的间隔,这个特性在减小泛化误差界的过程中起到了关键的作用.SoftBoost 类似于 LPBoost,通过线性规划直接求解最大间隔解,最大的差别就是 SoftBoost 使用了相对较弱的约束条件,以及引入对初始均匀分布的相对熵作为目标函数.这使得训练集上的样本分布在更新以后仍然趋向于均匀分布.SoftBoost[22]最终以δ的准确率达到了 $O\left(\frac{\ln N}{\delta^2}\right)$对数级的迭代界.尽管 SoftBoost 最小化当前样本分布到样本初始分布的相对熵,满足所有弱分类器的边界约束,但随着间隙上界的逐渐减小,SoftBoost 的泛化误差在早期下降缓慢.针对此问题,Warmuth 等人[29]提出的 ERLPBoost 通过增加 $1/\eta$ 因子到初始分布的相对熵来在最大化软间隔目标和最小化相对熵之间进行折中,解决了 SoftBoost 中泛化误差早期下降缓慢的问题.

本文所提出的算法基于 LPBoost 中的软间隔最大化算法,引入强分类器边界约束来最大化软间隔.同时,在生成弱分类器时,使用弱分类器间的相关度与冗余度,结合弱学习器的准确率来选择性地删除一些弱学习器,降低了整体弱分类器间的相关度,增加最终分类器间的差异性.不仅在一定程度上提高了算法收敛速度,而且使最终的分类准确率有了进一步的提高.

2 符号定义与 LPBoost

本节主要介绍误差、间隔、边界和 LPBoost 的相关知识,首先引入两个概念:边界(edge)和间隔(margin),然后证明边界和间隔互为对偶性问题.另外,本文中如没有特殊说明,学习器(learner)、假设(hypothesis)和分类器(classifier)所表示的概念相同.

一个弱分类器 h 的性能评价可以基于 h 在样本权重分布 $\bar{d}$ 上的边界γ_h来计算.γ_h 定义为

$$\gamma_h = \sum_{m=1}^{M} d_m y_m h(x_m),$$

其中,M 为样本个数,d_m 为样本(x_m,y_m)对应的权重,$y_m\in\{-1,1\}$,$h(x_m)\in\{-1,1\}$.并定义 h 在样本集上的误差为

$$\varepsilon_h = \sum_{m=1}^{M} d_m (y_m \neq h(x_m)).$$

可以看出,弱分类器 h 的误差和边界的关系 $\varepsilon_h(d) = \frac{1}{2} - \frac{1}{2}\gamma_h$. 如果一个弱分类器预测能力好,则$\gamma_h$=1;反之,则$\gamma_h$=−1,随即分类器的$\gamma_h$=0.边界$\gamma$值越大,分类效果越好.

Boosting 算法的最终输出 f 是弱分类器的一个凸线性组合形式:

$$f_w(x_m) = \sum_{t=1}^{T} w_t h^t(x_m),$$

其中,h^t 是添加的弱分类器,w_t 是其对应的权重.与边界含义不同,间隔(margin)是对于样本而言的,简单来说是样本到分类器的距离.样本(x_m,y_m)对于分类器 f_w 的间隔被定义为$\rho_m=y_m f_w(x_m)$.训练集对于分类器 f_w 的间隔(hard margin)是所有样本间隔的最小值.换句话说,间隔描述了分类器在训练集上的泛化能力,间隔越大,分类器的泛化能力越好.

近年来的理论研究表明,在最小化训练集误差的同时,应该使训练集上的 margin 越大越好,这个问题可以看作是带约束的最优化问题.Breiman 和 Grove 等人[20,30]指出,这个最优化问题可以通过线性规划方法来求解.这里参照文献[20]给出一个简化的形式说明,构造代价矩阵如图 1 所示,对于弱分类器 h_j 和训练集 X,定义代价矩阵

U,其元素 $u_{i,j}=h_j(x_i)*y_i$.此时,在每个样本(x_i,y_i)上的间隔对应为 $\rho_i=\sum_{j=1}^{t}w_ju_{i,j}=\overline{w}\cdot\overline{u_{i\cdot}}$;而训练集的间隔可以表示为 $\rho_X=\min_{i=1}^{m}\overline{w}\cdot\overline{u_{i\cdot}}$,即训练集上的样本间隔的最小值.

	h_1	...	h_t	样本权重
x_1	$u_{1,1}$	...	$u_{1,n}$	d_1
...	...	...	...	...
x_m	$u_{m,1}$	...	$u_{1,n}$	d_n
弱分类器权重	w_1	...	w_t	

Fig.1 Cost matrix

图 1 代价矩阵

基于上述符号定义,集成学习的目标是找到一个权重向量,使得在满足约束条件 $w_j\geqslant 0,\sum_j w_j=1$的同时使间隔$\rho_X$ 最大化.这是一个最大最小化问题,选择弱分类器和权重 $\overline{w}$ 来最大化 $\min_i \overline{w}\cdot u_i$, 同时满足约束条件 $w_j\geqslant 0,\sum_j w_j=1$. 这个问题用线性规划形式表示如公式(1)所示(注意,$u_{i\cdot}$ 表示 U 的第 i 行元素,$u_{\cdot j}$ 表示 U 的第 j 列元素).

$$\begin{cases}\max\limits_{w}\ \min\limits_{i\in\{1,\ldots,m\}} w*u_{i\cdot}\\ \text{s.t. } w_j\geqslant 0,\sum\limits_j w_j=1\end{cases}\ \text{或者}\ \begin{cases}\max\limits_{w}\ \rho\\ \text{s.t. } \sum\limits_{j=1}^{t}w_j.u_{i,j}\geqslant\rho,\ \text{for } i=1,\ldots,m\\ w_j\geqslant 0,\sum\limits_j w_j=1\end{cases}\tag{1}$$

上述最大最小问题(1),由 Von-Neumann 最小最大定理[31]可以证明其对偶问题为最小最大问题(2)的形式.

$$\begin{cases}\min\limits_{d}\ \max\limits_{j\in\{1,2,\ldots,t\}} u_{\cdot j}*d\\ \text{s.t. } d_i\geqslant 0,\sum\limits_i d_i=1\end{cases}\ \text{或者}\ \begin{cases}\min\limits_{d}\ \gamma\\ \text{s.t. } \sum\limits_{i=1}^{m}u_{i,j}d_i\leqslant\gamma,\ \text{for } j=1,\ldots,t\\ d_i\geqslant 0,\sum\limits_i d_i=1\end{cases}\tag{2}$$

同时,问题(1)和问题(2)满足公式(3):

$$\rho^*=\max_w\min_i w*u_{i\cdot}\leqslant\ \gamma^*=\min_d\max_j u_{\cdot j}.d\tag{3}$$

由问题(2)可以进一步推导出另外一种 boosting 过程,使用线性规划来计算样本权重,这个过程能够在整个基假设空间达到最优间隔.在对偶问题中,维护每个样本的权重 d_i,使得每个弱假设满足约束 $\sum_i u_{i,j}d_i\leqslant\gamma$. 这里,$\gamma$ 是对偶问题中弱分类器满足的界.对偶问题可被看作选择 $(\overline{d},\gamma)$ 来最小化γ,同时满足:

$$\sum_i d_iu_{i,j}\leqslant\gamma,\ j=1,\ldots,\text{t, and }\sum_i d_i=1,d_i\geqslant 0.$$

注意,这些约束条件可以自然地解释.向量 $\overline{d}$ 为训练样本上的概率分布,U 的第 j 列为弱分类器 h_j 对样本分类子序列,那么,$\sum_i d_iu_{i,j}$ 就是在新的样本权值分布下对假设(或者弱分类器 h_j)的简单评分.我们重新描述对偶问题如下:找到一个样本集下的权值分布,使最佳弱分类器的评分γ尽可能地小.即调整样本权值分布,使错分样本权值增加,而减小正确分类的样本权值.通过寻找一种最优的样本权值分布,使得最终的强分类器间隔最大.上述最小最大问题(2)就是 LPBoost 的核心思想.

另外,在样本可分的情况下,LPBoost 性能优于 AdaBoost.一旦样本集中存在难分样本或者噪声样本情况,LPBoost 过于偏向这些难分样本或噪声点,就会处于无解状态.因此,我们需要考虑如何针对少部分不可分样本(可能由噪声产生)进行约束条件的弱化,进而继续使用 LPBoost 算法来求解,这样就产生了最大化软间隔(maximizing soft margin)的算法,这里的“soft”是 margin 约束条件的减弱.现在允许部分样本处于 margin 中,通过松弛变量ζ_i 来对这一部分样本进行惩罚.此时,其对偶问题就是带约束条件的最小化弱分类器的最大边界,这就

是 SoftBoost 和 ERLPBoost 的思想.假定训练样本集上分类超平面的间隔为ρ,每个样本 i 到分类超平面的间隔记为$\rho-\zeta_i$,将原问题和对偶问题写成如下形式:

$$\text{原问题:}\begin{cases}\max\limits_{w,\rho,v}\ \rho-\dfrac{v}{m}\sum\limits_{i=1}^{m}\zeta_i\\ \text{s.t. } w.u_i\geqslant\rho-\zeta_i, i=1,...,m;\\ \zeta_i\geqslant 0;\\ w_j\geqslant 0,\sum\limits_j w_j=1\end{cases};\quad \text{对偶问题:}\begin{cases}\min\limits_{d,\gamma}\ \gamma\\ \text{s.t. } \sum\limits_{i=1}^{m}u_{i,j}d_i\leqslant\gamma,\ \text{for } 1\leqslant j\leqslant t;\\ 0\leqslant d_j\leqslant\dfrac{v}{m},\sum\limits_j d_j=1\end{cases}\tag{4}$$

其中,v来源于支持向量机社区中的 v-SVC[32]中的v思想,表示了在训练样本中处于分界面间隔内的样本点所占比例.

3 SelectedBoost 的提出

如上所述,即便 SoftBoost,ERLPBoost 能够容忍部分噪声点或者难分样本点,并且迭代界已经达到对数界的水平,但由于数据规模以及数据分布的复杂性,使得这些算法仍然存在以下两个问题:

(1) LPBoost,ERLPBoost 等算法的弱分类器凸线性组合是基于最优分类器边界最小化原则而生成的,过于侧重最大化训练集样本间隔而忽视了训练样本分类准确率,使得这些算法在迭代过程中产生的弱分类器的摇摆,最终导致中后期生成的弱分类器之间存在很大的相关性,极易导致其强分类器的准确率增长速度缓慢.因此,如果能够消除这种相关性,则有助于减少生成的弱分类器个数,并且增加准确率.

(2) 由公式(2)与公式(4)可以看出,LPBoost 系列算法中,样本权重分布的更新,本质上都是基于弱分类器分类结果而言的,然而与弱学习器相比,强学习器更能代表当前的决策平面,样本权重如果能够参照强学习更新,则有助于增加算法的收敛速度.从另一个角度来说,弱学习器的凸线性组合形成的强分类器边界并不一定小于弱学习器的最大边界.我们由此来研究间隔最大化的对偶问题中,是否可以通过增强对偶问题的约束条件来提升间隔最大化收敛速度.

3.1 弱学习器间的相关性

如本文开始部分所描述,弱分类器间的差异性对最终生成的强分类器的效果影响非常明显.提高弱分类器间的差异性是得到较好集成性能的必要条件.然而,对于这种弱分类器间的差异性度量并没有一个统一的标准.与差异度相对的还有依赖性、相关性、正交性、互补性等等,但对于这些,至今都没有一个严格的定义.当前,较常见的弱分类器差异度量方法主要有 10 种,按照计算的对象可以归为两大类[1]:基于对的(pair-wise)和非基于对的(non-pair-wise).基于对的差异性计算方法从两个弱分类器上进行评价,如 Q 统计量、ρ(correlation coefficient)、一致性度量 A(agreement measure)、双误性(double fault);非基于对的方法则是用于直接评估全体弱分类器上的差异度,如投票熵、困难度索引值(difficulty index)、Kohavi-Wolpert 方差、评判一致性(interrater agreement)、泛化差异度、一致性失败差异度等等.正如文献[1]中所说,当弱分类器准确率普遍较高时,差异度必然下降.因此,如何在差异度和准确率两个因素之间取折中很重要.然而并没有具体的理论或者实验结果提供一种差异性度量,直接降低集成后的强学习器泛化误差.

由于本文使用基于弱分类间的差异度指标来删除某些高度相关的弱分类器,并且使用基于对的差异度计算复杂度要远小于非基于对的差异度计算,因此这里主要考虑介绍基于对的差异度计算方法,参照上面提到的 4 种计算方法,提出了基于错误率和弱分类器相关度的选择性评判,用于对新生成的弱分类器进行选择,使得最终的弱分类器数量大为减少,强分类器分类性能保持最优.对于非基于对的度量计算方法,具体参考文献[1,2].

分类器 i,j 对样本集分类输出结果可以看作长度为 M 的二值向量:

$$\overline{y_i}=(y_{i,0},...,y_{i,M}),\ \overline{y_j}=(y_{j,0},...,y_{j,M}),$$

其中,$y_{j,k}$=1,如果分类器 j 正确分类样本 k,k=1,…,M.接下来,给出以下符号定义:

$$N_{i,j}^{11}=\sum_{k=1}^{M}y_{i,k}\wedge y_{j,k},$$
$$N_{i,j}^{00}=\sum_{k=1}^{M}\widehat{y_{i,k}}\wedge\widehat{y_{j,k}},$$
$$N_{i,j}^{10}=\sum_{k=1}^{M}y_{i,k}\wedge\widehat{y_{j,k}},$$
$$N_{i,j}^{01}=\sum_{k=1}^{M}\widehat{y_{i,k}}\wedge y_{j,k},$$

其中,上面的∧表示逻辑与运算,$\widehat{y_{j,k}}$ 表示逻辑非运算.基于上述符号定义,Q 统计量、ρ、一致性度量 A、双误性的计算公式如下(公式(5)~公式(8)):

$$Q_{i,j}=\frac{N_{i,j}^{11}N_{i,j}^{00}-N_{i,j}^{01}N_{i,j}^{10}}{N_{i,j}^{11}N_{i,j}^{00}+N_{i,j}^{01}N_{i,j}^{10}} \tag{5}$$

$$A_{i,j}=\frac{N_{i,j}^{11}+N_{i,j}^{00}}{N_{i,j}^{11}+N_{i,j}^{10}+N_{i,j}^{01}+N_{i,j}^{00}} \tag{6}$$

$$\rho_{i,j}=\frac{N_{i,j}^{11}N_{i,j}^{00}-N_{i,j}^{01}N_{i,j}^{10}}{\sqrt{(N_{i,j}^{11}+N_{i,j}^{10})(N_{i,j}^{01}+N_{i,j}^{00})(N_{i,j}^{11}+N_{i,j}^{01})(N_{i,j}^{10}+N_{i,j}^{00})}} \tag{7}$$

$$DF_{i,j}=\frac{N_{i,j}^{00}}{N_{i,j}^{11}+N_{i,j}^{10}+N_{i,j}^{01}+N_{i,j}^{00}} \tag{8}$$

由上述公式不难看出:分类器 i,j 共同分类正确的样本越多,$Q_{i,j}$ 越趋近于 1,反之则趋近于–1.如果它们是统计独立的两个分类器,则 $Q_{i,j}$ 等于 0;相关系数 $\rho_{i,j}$ 与 Q 统计量类似,它们有相同的符号,并且可以证明$|\rho|\leqslant|Q|$;一致性度量 $A_{i,j}$ 是直接观察两个分类器分类结果相同的样本个数与样本总数的比值;双误性是评价分类器 i,j 同时误分类样本的比例.

另外,上述指标(公式(5)~公式(8))只是描述两个弱分类间的差异性度量.为了观察全局弱分类器间总体差异度,本文使用平均差异度统计量 C_{avg},其计算公式如公式(9)所示.值得注意的是,C_{avg} 只是便于我们的实验观察,而在实际算法中并不计算该参数,只需对每个新生成的弱分类器与已有分类器计算差异度.

$$C_{avg}=\frac{2}{M(M-1)}\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}C_{i,j} \tag{9}$$

其中,$C_{i,j}\in\{Q_{i,j},\rho_{i,j},A_{i,j},DF_{i,j}\}$.

总体来说,这 4 个指标均与弱分类器间的差异度成相似的关系;同时,每个指标各有侧重.因此,本文综合使用这 4 种指标来进行差异性的评价,使用 $C_{i,j}$ 作为最终弱分类器之间的相关度,如公式(10)所示.最终,结合准确率和相关度对弱分类器进行选择.

$$C_{i,j}=(Q_{i,j}+A_{i,j}+\rho_{i,j}+DF_{i,j}) \tag{10}$$

3.2 强分类器边界约束作用

对于问题(2),我们希望使用形式化的方法加以阐述.参照之前的符号说明,给出以下符号定义:

$$\gamma_{\min}=\min_j\left(\sum_{i=1}^{m}h_j(x_i)y_id_i\right)\text{ for }j=1\text{ to }t,$$
$$\gamma_{\max}=\max_j\left(\sum_{i=1}^{m}h_j(x_i)y_id_i\right)\text{ for }j=1\text{ to }t,$$
$$H(x_i)=\sum_{j=1}^{t}h_j(x_i)w_j,$$
$$H'(x_i)=\begin{cases}1, & \text{if } H(x_i)>0\\ -1, & \text{if } H(x_i)\leqslant 0\end{cases}.$$

显然,不等式(11)成立.但是对于 $H(x_i),H'(x_i)$,情况会如何呢?

$$\gamma_{\min} \leqslant \sum_{i=1}^{m} h_j(x_i) y_i d_i \leqslant \gamma_{\max}, \text{ for } j=1,...,m \tag{11}$$

对于 $H(x_i)$,进行如下推导:

$$\sum_{i=1}^{m} H(x_i) d_i y_i = \sum_{i=1}^{m} \sum_{j=1}^{t} h_j(x_i) w_j d_i y_i = \sum_{j=1}^{t} w_j \sum_{i=1}^{m} h_j(x_i) d_i y_i \tag{12}$$

结合公式(11),可以得到:

$$\gamma_{\min} \leqslant \sum_{i=1}^{m} H(x_i) d_i y_i \leqslant \gamma_{\max} \tag{13}$$

由边界的定义,可以从公式(13)看出弱分类器的凸线性组合 $H(x_i)$,即产生的强分类器的分类边界小于各弱分类器的最大边界.按照这个结论推导并结合边界和错误率的关系,可以得出结论:强分类器的错误率大于弱分类器错误率的最小值.显然,这个结论与我们平时的结论是相违背的,也与集成学习相关理论相悖.那么,我们的推导和结论是否错了呢?仔细观察可以看出,最终形成的强分类器应该是 $H'(x_i)$而不是 $H(x_i)$.也就是说,对于 $H'(x_i)$,并不一定满足

$$\gamma_{\min} \leqslant \sum_{i=1}^{m} H'(x_i) d_i y_i \leqslant \gamma_{\max} \tag{14}$$

这给我们一个启发:把每一次迭代生成的强分类器 $H'(x_i)$添加到最小化最大边界问题(2)或问题(4)的约束条件中,使约束条件更加严格,从而增加了本文所提出算法 SelectedBoost 的收敛速度.本文实验部分将对此进行详细验证.

4 SelectedBoost 算法

针对第 3 节的讨论,对于问题(1),在新生成的弱分类器时,计算与之前已有弱分类器的差异度、相关度;对问题(2),引入强分类器边界约束,最终的带约束二次优化问题如公式(15)所示.

$$\begin{cases} \min\limits_{d_t,\gamma} \gamma + \eta.\Delta(d^t, d^0) \\ \text{s.t. } \sum\limits_{i=1}^{m} u_{i,j} d_i \leqslant \gamma, \text{ for } 1 \leqslant j \leqslant t; \\ \sum\limits_{i=1}^{m} H'_{t-1}(x_i) y_i d_i \leqslant \gamma; \\ 0 \leqslant d_j \leqslant \dfrac{v}{m}, \sum\limits_{j} d_j = 1 \end{cases} \tag{15}$$

其中,η为折中因子,0<η<1.综合上述两个问题的解决方案,本文提出了 SelectedBoost 算法,具体的算法伪代码如图 2 所示.在基于软间隔最大化的同时,最小化样本权值分布的相对熵,同时引入强分类器边界约束,并综合准确率、相关度和差异度来判断是否使用新生成的弱分类器.

在每次的迭代过程中,先使用参数 d^{t-1} 调用 oracle 函数,产生弱分类器 h^t,然后基于新的弱分类器集合使用基于软间隔最大化方法求解出新的样本权值 d^t,同时计算当前弱分类器的凸线性组合系数 $\overline{w_t}$ 及生成的强分类器 $f^t(x) = \sum_{q=1}^{t} w_q^t h^q(x)$,进一步计算当前强分类器的准确率 $err(t)$和新生成的 h^t 与已有弱分类器的相关度 c^t,判定是否使用新需要删除弱分类器,如果 h^t不能保证准确率上升并且还增加了系统的相关度(冗余),则选择与 h^t相关度最高的 $h^{t'}$,然后重新求解剩下弱分类间的权重以及样本权重分布,并进入下一次迭代.由于本算法主要对弱分类器进行选择,并没有增加迭代界,其迭代界如文献[28]所述:如果想要达到最大最小软间隔的ε范围内,需要 $O\left(\frac{1}{\varepsilon^2}\ln\frac{N}{v}\right)$的迭代次数.

算法1. SelectedBoost.

1. Input : $S=\{(x_1,y_1),...,(x_m,y_m)\}$,准确率参数$\varepsilon>0$,置信度$\delta$,
   折中因子η,平滑参数$0\leqslant\nu\leqslant 1$,最大迭代次数Max*Iter*,
   弱分类器生成器 : *oracle*(d),弱分类器集合H.
2. Initialize : d^0为均匀分布,$\delta^0=\varepsilon$,$err(0)=1$.
3. For $t=1$ to Max*Iter*
   (a) 以参数d^t调用oracle产生新的弱分类器h^t,$H=H\cup\{h^t\}$;
   (b) 求解如下带约束最优化问题:

$$\min_{d_t,\gamma}\ \gamma+\eta.\Delta(d^t,d^0)$$

$$\text{s.t. }\sum_{i=1}^{m}u_{i,j}d_i\leqslant\gamma,\ \text{for}\ 1\leqslant j\leqslant t;$$

$$\sum_{i=1}^{m}f^{t-1}(x_i)y_id_i\leqslant\gamma;$$

$$0\leqslant d_j\leqslant\frac{\nu}{m},\sum_j d_j=1;$$

   更新分布d^t,$\delta^t=\gamma+\eta.\Delta(d^t,d^0)$;
   (c) 调用LPBoost对H中所有弱分类器使用最大化间隙方法求解弱分类器权重$\bar{w}$:

$$f^t(x)=sign\left(\sum_{q=1}^{t}w_qh^q(x)\right),$$

$$err(t)=1-\frac{\left(\sum_i(f^t(x_i)==y_i)\right)}{n};$$

   (d) if $(err(t)>err(t-1))$
   计算相关度$c_{t,j}$, $j=1,...,t-1$;
   $h^{t'}=\arg\max_{j=1}^{t-1}(c_{t,j})$, 其中, $c_{t,j}$按照公式(10)得到;
   从H中删除弱分类器$h^{t'}$;重新求解(b)中最优化问题,并更新d^t,δ^t;
   continue;
   end
   (e) if $(\delta^t-\delta^{t-1})<\varepsilon/2$
   break;
   end
4. Output : 使用LPBoost对H中所有弱分类器基于最大化间隙方法求解弱分类器权重w:

$$f_{final}(x)=sign\left(\sum_{q=1}^{t}w_qh^q(x)\right),t\text{为最终的弱分类器个数; }sign(\cdot)\text{为符号函数.}$$

Fig.2 SelectedBoost algorithm pseudo-code

图 2 SelectedBoost 算法伪代码

5 实验验证

为了评价 SelectedBoost 的性能,本文与当前具有代表性的 4 种集成学习算法 AdaBoost,LPBoost,SoftBoost, ERLPBoost 进行了大量的实验比较.另外,为了便于描述,SelectedBoost 在图表中用 slpBoost 表示.

5.1 实验准备

与文献[19,28]中使用的评测数据集类似,本文使用 11 个人工生成和真实数据集.这些数据集来源于 UCI, DELVE 标准测试数据集:banana,breast cancer,diabetes,german,heart,image segment,ringnorm,new-thyroid, twonorm, waveform,spiral.然而这些数据集并不能直接用来做实验,需要对数据进行如下预处理:

1. 如果数据集不是二分类问题,即类别标签个数大于 2,则随机的把其类别属性分成两类标签,并尽可能地使样本类别分布平衡.

2. 如果数据集中样本属性值缺失,则删除这些样本.也就是说,实验中所有样本集中每个属性都有值;

3. 针对样本集中的符号属性将其转换为 1 到 N 的一个数字,N 为其属性取值个数.

其中,Spiral(双螺旋型)和 Banana(香蕉形)数据集为人工生成数据集,主要目的是便于观察各集成学习算法在边界、间隔、迭代界上的不同.最终,本文使用的数据集描述见表 1.

Table 1 Dataset description in the experiments

表 1 本文实验所使用各数据集描述

数据集	属性个数	原始类别个数	样本分布	样本数
Banana	2	2	1000/1000	2 000
Breast cancer	10	2	357/212	569
Heart	14	2	150//120	270
Image segment	19	7	990/1320	2 310
Ringnorm	21	2	3700/3700	7 400
New-Thyroid	5	3	150/75	215
Twonorm	21	2	3700/3700	7 400
Waveform	21	3	2000/4000	6 000
Spiral	3	2	900/900	1 800
German	20	2	700//300	1 000
Diabetes	8	2	500/268	768

实验中使用单节点决策树作为基分类器,上述 5 种算法最大迭代次数均为 200 次,其他参数为默认最优值.将每个数据集分为训练集和测试集(其比例为 80%:20%),并使用 10 轮交叉验证来产生 10 个训练模型.在此基础上,我们得到每种集成算法的平均效果及方差作为最终的评判标准,这样使得比较实验更加完备和可信.对于软间隔算法,参数设置参照文献[29]设置,具体设置为ε=0.01, ν=0.05.最后, $\eta = 2/\varepsilon \ln N/\nu$.

5.2 实验过程

5.2.1 分类准确率实验

本节主要评价 SelectedBoost 算法的分类准确率效果,这里,在 11 个数据集上对比其他 4 种集成学习算法,使用的评价指标为准确率(accuracy)、精确度 precision)、召回率(recall)、$F1$ 测度($F1$-score).实验基于 10 轮交叉验证的均值与方差,终止条件是达到一定迭代界和误差范围的终止条件.实验结果如图 3 所示,条状图上方部分为方差,下方部分为均值.对于某些算法在部分数据集上方差趋近于 0,故方差部分不明显.另外,由于 5 种算法在数据集 Banana 上的分类准确率均为 1,故图 3 只展示其余的 10 种数据集上的评测结果.

从图 3 中的准确率结果来看,除了在 heart 和 waveform 两个数据集上 SelectedBoost 稍低于 AdaBoost 的分类性能之外,在大部分数据集上都表现出优异的性能.另外,从分类的精确度、召回率以及 F1 测度实验结果来看,AdaBoost 算法通常具有较高的精确度,但召回率结果较差.相对于 LPBoost,SoftBoost 和 ERLPBoost,本文所提出的 SelectedBoost 方法普遍达到了较好的评价效果.

另外,为了描述这 5 种算法在各数据集上的差异程度,本文使用单因素方差分析和 paired t-检验来具体观察准确率上的差异程度.基于上述 10 轮交叉验证得到的结果数据,使用单因素方差分析,结果显示,5 种算法在 ringnorm,diabetes,image,newthyroid 这 4 个数据集上准确度均值有显著性差异(置信度为 0.1).进一步使用 paired t-检验查看 SelectedBoost 与其他算法的准确率差异程度,概率值结果见表 2,表中黑体表示本文所提出的 SelectedBoost 算法显著性水平明显好于其他算法(置信度为 0.1).

Table 2 Significance test with paired t-test at confidence level 0.1 over 10 datasets

表 2 10 种数据集上显著性检验的 paired t-检验值(置信度为 0.1)

SlpBoost vs.	German	Ring-Norm	Twonorm	Diabetes	Image	New-Thyroid	Heart	Wave-Form	Breast cancer	Spiral
RealBoost	0.318	**0.001**	0.272	**0.024**	**0.002**	**0.096**	0.973	0.651	0.120	0.230
LPBoost	0.124	**0.000**	0.500	**0.030**	0.114	0.223	0.780	0.531	0.272	0.258
SoftBoost	0.148	**0.008**	0.104	**0.037**	**0.027**	**0.076**	0.959	**0.076**	**0.043**	**0.087**
ErlpBoost	0.115	**0.000**	0.358	0.306	0.012	0.152	0.858	0.342	0.905	0.146

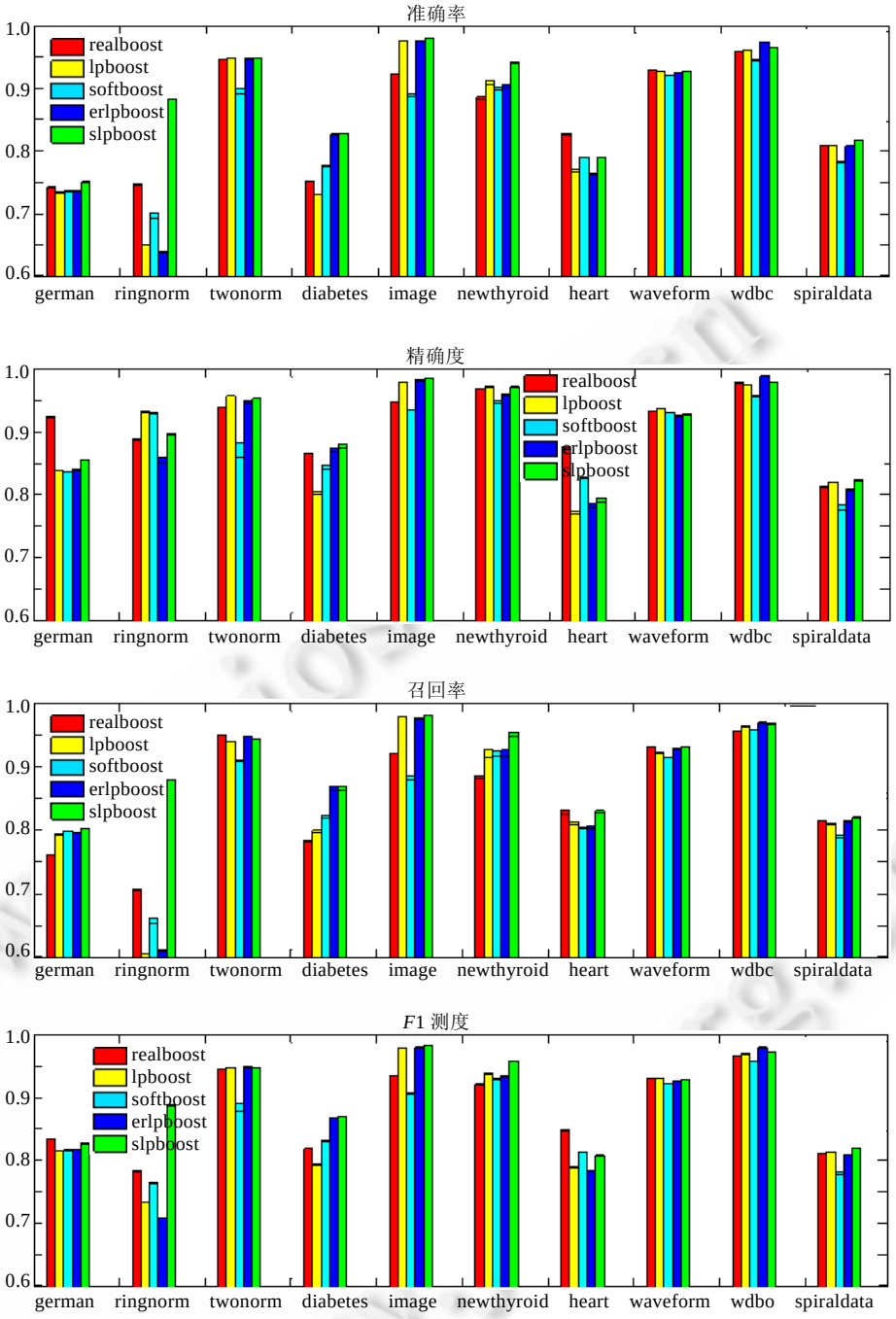


Fig.3 Comparison chart of 4 classification evaluation measures for 5 algorithms over 10 benchmark datasets

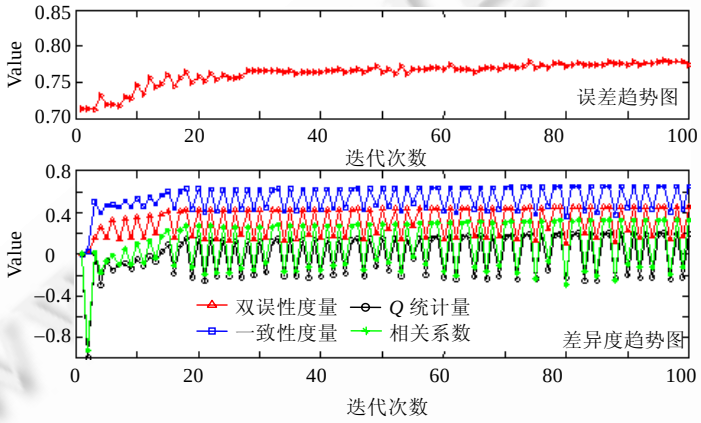
图 3 10 个标准评测数据集上使用 4 种分类性能评价指标对 5 种算法的对照图

结合图 3 和表 2 可以看出,在其他数据中,与另外 4 种算法相比,SelectedBoost 表现出相当的,甚至更好的优势.

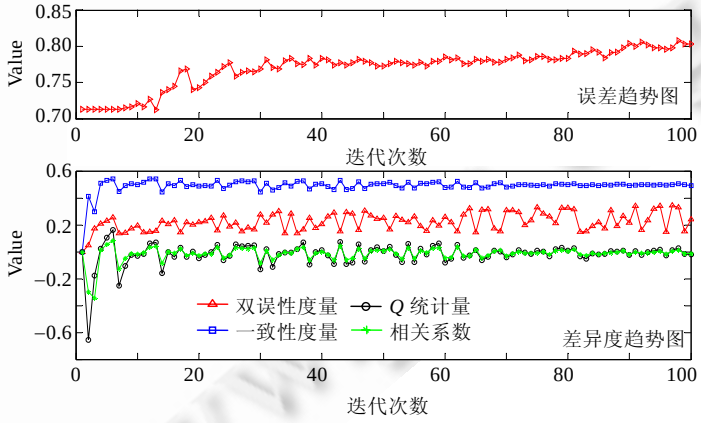
5.2.2 差异度对准确率的影响

本节观察差异度和准确率随迭代次数的变化关系.为了说明本文所提出的基于差异度的集成学习的效果,这里实验数据为 german 数据集,选用的对比算法为 LPBoost 与 SelectedBoost.迭代次数均为 100 次.

由图 4 可知,4 种差异性度量指标与准确率总体上呈现相似的正比例关系,在迭代开始阶段(1 步~20 步),4 种指标均出现了大幅度的调整,随后随着准确率的稳定,也趋于稳定的波动变化.比较图 4(a)与图 4(b)可以看出,LPBoost 在多个差异度指标上波动范围较大.从一致性度量与双误性度量的差值来看,说明新生成的弱分类器与现有弱分类器之间虽然共同正确分类的样本数量稳定,但共同的误分样本数量波动变化较剧烈,而每次迭代生成的弱分类器稳定性较差;并且从新生成的弱分类器与已有弱分类器间的相关性来看,呈剧烈波动变化,通俗地说就是一次正相关一次负相关,一次与整体分类性能一致一次不一致的交替变化.而对 SelectedBoost 来说,生成的弱分类器与已有弱分类器共同误分的样本数量变化幅度较大,而一致性度量上则趋于稳定.另外,从 Q 统计量和相关系数ρ来看,新生成的弱分类器与已有弱分类器间的相关性较弱,基本趋近于 0.从这个实验结果中可以看出,SelectedBoost 中的弱学习器具有较小的相关性,或者具有较大的差异性;另外,LPBoost 算法产生的新弱分类在后期存在较大的振荡现象.



(a) LPBoost 中 4 种差异度随迭代次数趋势图



(b) SelectedBoost 中 4 种差异度随迭代次数趋势图

Fig.4

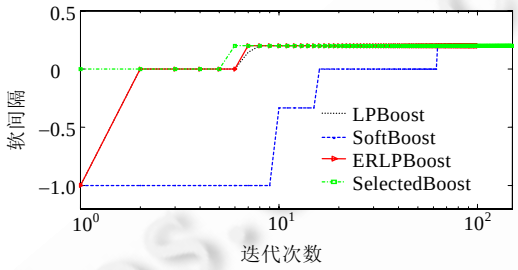
图 4

5.2.3 最大化间隔对准确率的影响

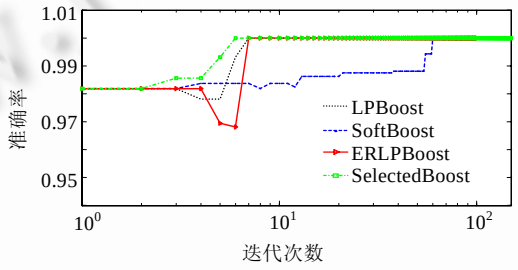
下面我们评价各集成学习算法在间隔最大化、准确率和迭代次数方面的性能.

为了方便与 SoftBoost,ERLPBoost 观察比较,本节参照文献[28,29]的实验方法,在 Banana 数据集上进行比较.另外,由于 AdaBoost 并不是基于间隔最大化理论的算法,因此本部分实验不考虑 AdaBoost.

图 5 是 LPBoost,SoftBoost,ERLPBoost 和 SelectedBoost 在 Banana 数据集上同一次的实验结果.图 5(a)描述了 4 种集成学习算法在 Banana 数据集上的间隔随迭代次数的走势图,可以看出,SelectedBoost 能够最快速地收敛到实际间隔值.注意,与文献[22,23]的实验结果一致,LPBoost 基本上与 ERLPBoost 的收敛趋势吻合,ERLPBoost 收敛速度初期优于 SoftBoost.图 5(b)说明了在 Banana 数据集上各算法的准确率随迭代次数的提升速度.结合图 5(a)和图 5(b)可以看出,本文所提出的算法能够以较快的速度使用不相关的弱分类收敛到最大间隔,以最快速度达到准确率最优值.



(a) LPBoost,SoftBoost,ERLPBoost 和 SelectedBoost 在 Banana 数据集上的间隔随迭代次数趋势图



(b) LPBoost,SoftBoost,ERLPBoost 和 SelectedBoost 在 Banana 数据集上分类器的准确率走势图

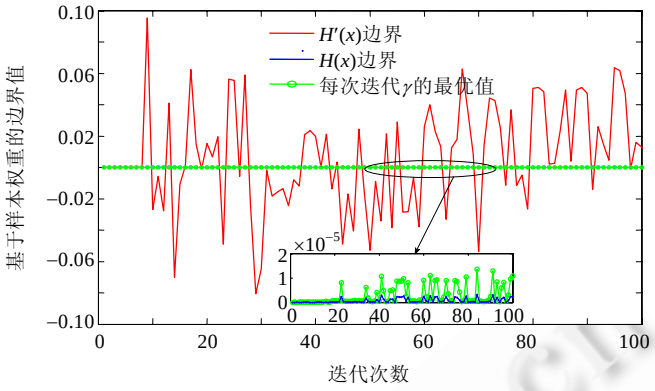
Fig.5

图 5

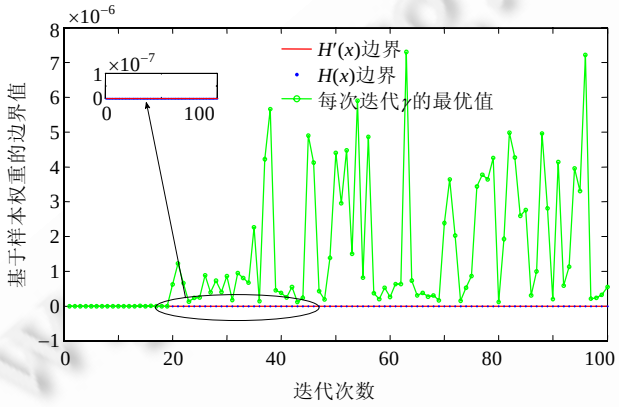
5.2.4 强分类器约束对收敛性的影响

本节实验是基于 spiral 数据集上(因为 Banana 数据集是明显可分数据集,在经过几次迭代学习就可以达到稳定的最优值,并不能明显看出引入强分类器 $H'(x)$边界约束前后的收敛性变化情况,所以本部分实验主要是在 spiral 数据集上来实现,没有选用 Banana 数据集),引入强分类器 $H'(x)$边界前后,$H'(x)$,$H(x)$与γ在每一次迭代中的关系曲线.

图 6(a)表示了在求解对偶问题(5)以后,$H'(x)$边界(右三角实点线)、$H(x)$边界(虚线)和弱分类器中的最大弱分类器边界γ(矩形虚点线)这 3 个边界的大小关系图.$H'(x)$边界值远远高于 $H(x)$边界和γ,由于 $H(x)$与γ的值太接近 0(10^{-6})而在图中不容易看出来,在图 6(a)中以图中图的方式将其放大显示出来.可以看出,与第 3 节的分析一致,$H(x)$边界(虚线)小于最大弱分类器边界(矩形虚点线).在将强分类器边界约束引入到问题(5)以后,所得到的边界曲线如图 6(b)所示.在对引入 $H'(x)$边界约束后的对偶问题求解以后,$H'(x)$与 $H(x)$的边界曲线均接近于 0,甚至 $H'(x)$的边界为 0.可以看出,对于问题(14),本文引入的强分类器边界约束起到了明显的作用.



(a) 引入强分类器 $H'(x)$边界约束前,$H'(x)$,$H(x)$边界与γ的关系



(b) 引入强分类器 $H'(x)$边界约束后,$H'(x)$,$H(x)$边界与γ的关系

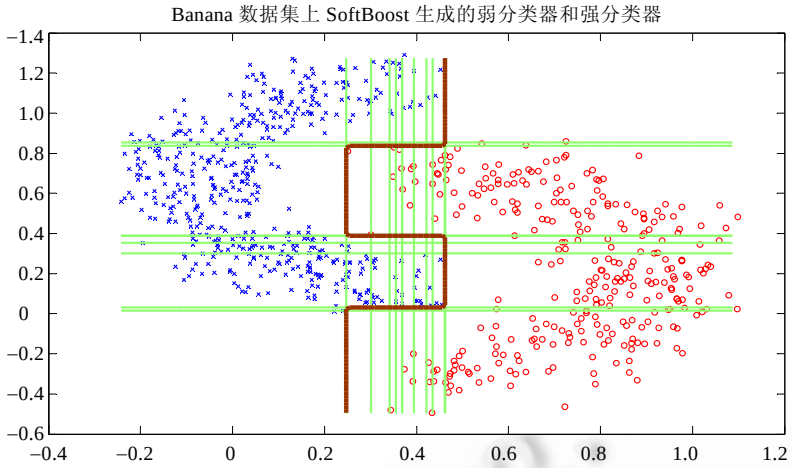
Fig.6

图 6

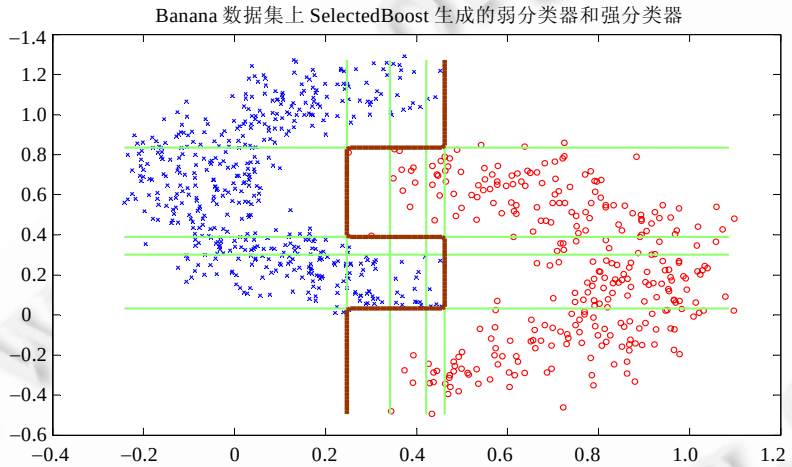
5.2.5 强分类器边界约束对弱分类器数量的影响

这一部分实验主要观察引入强分类器边界对弱分类器生成个数的影响.为了简化实验,便于观察,这部分实验数据只使用 800 个样本点的 Banana 数据集.

图 7(a)给出了 SoftBoost 算法生成的所有弱分类器和最终的强分类器,共有 20 个弱分类器(4 个重复).可以看出,生成的弱分类器存在严重冗余性和相关性.图 7(b)展示了引入强分类器边界约束以后,SelectedBoost 算法生成的弱分类器和强分类器.虽然两种算法生成的最终决策面接近一致,但是可以看出,引入强分类器边界约束以后,能够明显减少生成的弱分类器个数,快速收敛到最大间隔.由于使用的弱分类器个数大为减少,从而提高了分类速度.



(a) Banana 数据集上 SoftBoost 算法生成的最终强分类器(黑线)及弱分类器(灰线)



(b) Banana 数据集上 SelectedBoost 算法生成的最终强分类器(黑线)及弱分类器(灰线)

Fig.7

图 7

6 结束语

本文首先阐述了当前 boosting 集成学习算法的研究进展,深入分析了当前 LPBoost 系列集成算法存在的两个问题,并针对每个问题给出了相应的解决方法.如对 LPBoost 等集成学习算法中存在的弱学习器间的相关性和冗余性进行了分析,提出了基于弱分类器相关度和离散度的选择性集成 boosting 学习算法;并对于 LPBoost 系列算法中间隔最大化的对偶问题,即样本权值更新算法的带约束线性规划问题,在其约束条件中引入了更加严格的约束限制条件,使用强分类器边界限制条件作为约束,使得本文所提出的 SelectedBoost 算法与之前的 LPBoost 系列算法相比具有更快的收敛速度和更高的准确率,并在标准评测集的基础上,分别对收敛速度、准确率以及弱分类器间的相关度进行了实验分析和验证.实验结果表明,SelectedBoost 能够进一步减小弱分类器间的相关性,并引入强分类器边界约束条件,进一步提高了 SelectedBoost 的收敛速度.

我们将来的工作是在高噪声情况下,进一步降低 LPBoost 算法过程中生成的弱学习器间的相关性和冗余度,使得最终使用尽可能少的弱分类器来提高分类速度,并且进一步分析强分类器对间隙收敛性的影响.最后,

将会对这一系列集成学习算法引入概念漂移和样本倾斜分布的情况,以考虑算法的优化和准确率的提升.

致谢 在此谨向为本文工作提供支持和审稿意见的老师表示由衷的感谢.

References:

[1] Kuncheva LI, Whitaker CJ. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning, 2003,51(2):181−207. [doi: 10.1023/A:1022859003006]

[2] Duangsoithong R, Windeatt T. Relevance and redundancy analysis for ensemble classifiers. In: Perner P, ed. Proc. of the Machine Learning and Data Mining in Pattern Recognition. LNCS 5632, Heidelberg: Springer-Verlag, 2009. 206−220. [doi: 10.1007/978-3-642-03070-3_16]

[3] Rokach, L. Ensemble-Based classifiers. Artificial Intelligence Review, 2010,33(1-2):1−39. [doi: 10.1007/s10462-009-9124-7]

[4] Zhou ZH, Wu JX, Tang W. Ensembling neural networks: Many could be better than all. Artificial Intelligence, 2002,137(1-2): 239−263. [doi: 10.1016/S0004-3702(02)00190-X]

[5] Zhou ZH, Chen SF. Neural network ensemble. Chinese Journal of Computers, 2002,25(1):1−8 (in Chinese with English abstract).

[6] Tang W, Zhou ZH. Bagging-Based selective clusterer ensemble. Journal of Software, 2005,16(4):496−502 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/496.htm [doi: 10.1360/jos160496]

[7] Kearns MJ, Vazirani UV. An Introduction to Computational Learning Theory. Cambridge: MIT Press, 1994.

[8] Valiant LG. A theory of the learnable. Communications of the ACM, 1984,27(11):1134−1142. [doi: 10.1145/1968.1972]

[9] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997,55(1):119−139. [doi: 10.1006/jcss.1997.1504]

[10] Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. In: Proc. of the 11th Annual Conf. on Computational Learning Theory. 1998. 80−91. [doi: 10.1023/A:1007614523901]

[11] Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 1999,37(3):297−336. [doi: 10.1023/A:1007614523901]

[12] Kivinen J, Warmuth MK. Boosting as entropy projection. In: Proc. of the 12th Annual Conf. on Computer Learning Theory. New York: ACM Press, 1999. 134−144. [doi: 10.1145/307400.307424]

[13] Lafferty J. Additive models, boosting, and inference for generalized divergences. In: Proc. of the 12th Annual Conf. on Computional Learning Theory. New York: ACM Press, 1999. 125−133. [doi: 10.1145/307400.307422]

[14] Breiman L. Prediction games and arcing algorithms. Neural Computation, 1999,11(7):1493−1517. [doi: 10.1162/089976699300016106]

[15] Bartlett PL, Freund Y, Lee WS, Schapire RE. Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics, 1998,26(5):1651−1686. [doi: 10.1214/aos/1024691352]

[16] Demiriz A, Bennett KP, Shawe-Taylor J. Linear programming boosting via column generation. Machine Learning, 2002,46(1-3): 225−254. [doi: 10.1023/A:1012470815092]

[17] Rudin C, Daubechies I, Schapire RE. The dynamics of adaboost: Cyclic behavior and convergence of margins. Journal of Machine Learning Research, 2004,5:1557−1595.

[18] Rätsch G, Warmuth MK. Efficient margin maximizing with boosting. Journal of Machine Learning Research, 2005,6:2131−2152.

[19] Rätsch G, Onoda T, Müller KR. Soft margins for AdaBoost. Machine Learning, 2001,42(3):287−320. [doi: 10.1023/A:1007618119488]

[20] Grove AJ, Schuurmans D. Boosting in the limit: Maximizing the margin of learned ensembles. In: Proc. of the 15th National Conf. on Artifical Intelligence. 1998. 692−699.

[21] Warmuth MK, Liao J, Rätsch G. Totally corrective boosting algorithms that maximize the margin. In: Proc. of the ICML 2006. ACM Press, 2006. 1001−1008. [doi: 10.1145/1143844.1143970]

[22] Li HX, Shen CH. Boosting the minimum margin: LPBoost vs. AdaBoost. In: Proc. of the Digital Image Computing: Techniques and Applications (DICTA 2008). 2008. 533−539. [doi: 10.1109/DICTA.2008.47]

[23] Freund Y. An adaptive version of the boost by majority algorithm. Machine Learning, 2001,43(3):293−318. [doi: 10.1023/A:1010852229904]

[24] Domingo C, Watanabe O. Madaboost: A modification of Adaboost. In: Proc. of the COLT 2000. 2000. 180−189.

[25] Rätsch G, Schölkopf B, Smola AJ, Mika S, Onoda T, Müller KR. Robust ensemble learning. In: Smola AJ, Bartlett PL, Schölkopf B, Schuurmans D, eds. Advances in Large Margin Classifiers. Cambridge: MIT Press, 2000. 207−219.

[26] Rätsch G. Robust boosting via convex optimization: Theory and applications [Ph.D. Thesis]. Potsdam: University of Potsdam, 2001.

[27] Servedio RA. Smooth boosting and learning with malicious noise. Journal of Machine Learning Research, 2003,4(4):633−648. [doi: 10.1162/153244304773936072]

[28] Warmuth MK, Glocer K, Rätsch G. Boosting algorithms for maximizing the soft margin. In: Platt J, Koller D, Singer Y, Roweis S, eds. Advances in Neural Information Processing Systems 20. Cambridge: MIT Press, 2007.

[29] Warmuth MK, Glocer KA, Vishwanathan SVN. Entropy regularized lpboost. In: Proc. of the Algorithmic Learning Theory (ALT). 2008. [doi: 10.1007/978-3-540-87987-9_23]

[30] Breiman L. Arcing the edge. Technical Report, 486, Statistics Department, U. C. Berkeley, 1997.

[31] Von Neumann J. Zur theorie der gesellschaftsspiele (on the theory of parlor games). Mathematische Annalen, 1928,100(1): 295−320. [doi: 10.1007/BF01448847]

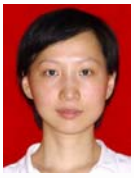
[32] Kuncheva LI, Whitaker CJ. Measures of diversity in classifier ensembles. Machine Learning, 2003,51(2):181−207. [doi: 10.1023/A:1022859003006]

附中文参考文献:

[5] 周志华,陈世福.神经网络集成.计算机学报,2002,25(1):1−8.

[6] 唐伟,周志华.基于 Bagging 的选择性聚类集成.软件学报,2005,16(4):496−502. http://www.jos.org.cn/1000-9825/16/496.htm [doi: 10.1360/jos160496]

方育柯(1984－),男,河南禹州人,博士,主要研究领域为机器学习,流数据挖掘,信息推荐.



佘莉(1978－),女,博士,讲师,CCF 会员,主要研究领域为数据挖掘,Web 文本挖掘.

傅彦(1962－),女,教授,博士生导师,主要研究领域为数据挖掘,信息安全.

孙崇敬(1986－),男,博士生,主要研究领域为数据挖掘,机器学习,隐私保护.

周俊临(1981－),男,博士,讲师,主要研究领域为数据挖掘,异常检测,信息推荐.