基于 Boosting 的不平衡数据分类算法研究

李秋洁 茅耀斌 王执铨
(南京理工大学自动化学院 南京 210094)

摘 要 研究基于 boosting 的不平衡数据分类算法,归纳分析现有算法,在此基础上提出权重采样 boosting 算法。对样本进行权重采样,改变原有数据分布,从而得到适用于不平衡数据的分类器。算法本质是利用采样函数调整原始 boosting 损失函数形式,进一步强调正样本的分类损失,使得分类器侧重对正样本的有效判别,提高正样本的整体识别率。算法实现简单,实用性强,在 UCI 数据集上的实验结果表明,对于不平衡数据分类问题,权重采样 boosting 优于原始 boosting 及前人算法。

关键词 不平衡数据分类,Boosting,采样
中图法分类号 TP391 **文献标识码** A

Research on Boosting-based Imbalanced Data Classification

LI Qiu-jie MAO Yao-bin WANG Zhi-quan
(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract This paper aimed to investigate boosting-based imbalanced data classification algorithms. Through the deep analysis of existing algorithms, a weight-sampling boosting algorithm was proposed. Changing the data distribution by weight sampling, the trained classifier was made suitable for imbalanced data classification. The natural of the proposed algorithm is that the loss function of naïve boosting is adjusted by the sampling function and the positive examples are emphasized so that the classifier focuses on correctly classifying these examples and finally the recognition rate of positive examples is improved. The new algorithm is simple and practical and has been shown to outperform naive boosting and previous algorithms in the problem of imbalanced data classification on the UCI data sets.

Keywords Imbalanced data classification, Boosting, Sampling

不平衡数据分类问题中,各类样本数目存在显著差异。此类问题在实际应用中经常碰到,如欺诈识别、入侵检测、医疗诊断等。当数据包含两类时,样本数目较少的类别称为少数类/正类(minority/positive class),样本数目较多的类别称为多数类/负类(major/negative class)。传统学习方法最小化分类准确度(classification accuracy),数据类别分布不平衡时,分类器对负类的分类效果很好,但对正类的分类效果较差,当不平衡程度严重时更是如此。为此,如何设计合适的学习算法,以得到适用于不平衡数据的分类器,成为目前机器学习和数据挖掘的一个研究热点[1]。

Boosting 是一类能够显著提高弱学习器性能的强学习算法,已在模式识别的各个领域得到成功应用[2]。原始 boosting 算法对正负类别样本等同对待,不适用于不平衡数据分类情况。本文研究基于 boosting 的不平衡数据分类算法。现有方法可分为数据采样法和代价敏感法两类。数据采样法通过数据采样技术平衡 boosting 训练样本集数据分布;代价敏感法强调正样本的分类损失,将数据不平衡分类问题转化为代价敏感分类问题,是广为使用的一种方法。

数据采样法计算量大,容易引入噪声或丢失有用数据信息;代价敏感法中代价敏感损失函数设计不够合理,如何有效引入不同类别的损失代价是个问题。本文提出一种基于权重采样的 boosting 算法,通过对样本进行权重采样改变原有数据分布,得到适用于不平衡数据的分类器。算法实现时,新的数据分布由样本权重乘以相应采样因子得到,算法并不产生或删除数据,实现简单,避免引入噪声和丢失有用信息。算法本质是利用采样函数调整原始 boosting 损失函数形式,进一步强调正样本的分类损失,使得分类器侧重于有效判别正样本,提高正样本的整体识别率。与代价敏感法相比,算法无需直接设计代价敏感损失函数。此外,采用独立于权重更新的权重采样机制,避免了将代价敏感损失引入权重更新而引起不稳定问题。以 AUC 为测度、原始 boosting 为基准算法,在 UCI 数据集上的实验结果表明,权重采样 boosting 优于原始 boosting 及前人算法。此外,本文研究不同的采样函数对算法性能的影响,通过理论分析和实验比较得出重点采样边界附近的正样本优于对所有正样本过采样和仅对误分正样本过采样的结论。

到稿日期:2011-01-20 返修日期:2011-05-04 本文受国家自然科学基金(60974129,70931002)资助。
李秋洁(1983-),女,博士生,主要研究方向为计算机视觉、模式识别和机器学习,E-mail:liqiujie_1@163.com;茅耀斌(1971-),男,博士,副教授,主要研究方向为图像处理与模式识别,E-mail:myb_njust@163.com(通信作者);王执铨(1939-),男,教授,博士生导师,主要研究方向为系统工程。

本文第 1 节介绍原始 boosting 算法；第 2 节归纳现有基于 boosting 的数据不平衡分类算法，分析其作用机制，指出存在的问题；第 3 节提出一种基于权重采样的 boosting 算法，并讨论采样函数设计；第 4 节给出相关实验数据及分析；最后对本文进行总结。

1 boosting 算法

boosting 是集成学习(ensemble learning)中最具代表性和应用前景的方法，是一种可提升任意给定弱学习算法(weak learning algorithm)性能的强学习算法(strong learning algorithm)。

1.1 算法流程

boosting 通过反复调用同一弱学习器(weak learner)产生多个弱分类器(weak classifier)，将其线性组合成最终的强分类器(strong classifier)[2]。算法实现时，每个样本与一个权重相关联，boosting 维护整个训练集上的权重/分布。样本初始权重相等，每轮迭代后，增加错误分类样本的权重，使得后轮弱分类器侧重对较难分类样本的正确分类。以 Discrete AdaBoost 为例，算法流程如下：

输入：

- 训练样本集 $\{(x_i,y_i)\}_{i=1}^N$，其中 $x_i\in X, y_i\in Y=\{-1,+1\}$；
- 弱学习器 L，在给定样本分布 D 下学习弱分类器 $f:X\to Y$；
- 迭代次数 M。

初始化：$F_0(x)=0$，$w_1(i)=1$。

$m=1:M$：

(1)计算样本分布：$D_m(i)=\dfrac{w_m(i)}{\sum\limits_i w_m(i)}$；

(2)弱学习：用 L 学习分布 D_m 下的弱分类器 $f_m:X\to Y$，弱分类器权重为 $\alpha_m=\dfrac{1}{2}\ln\dfrac{1-e_m}{e_m}$，误差 $e_m=\Pr_{i\sim D_m}[f_m(x_i)\neq y_i]$；

(3)权重更新：$$w_{m+1}(i)=w_m(i)\times\begin{cases}e^{-\alpha_m}, & \text{if } f_m(x_i)=y_i\\ e^{\alpha_m}, & \text{if } f_m(x_i)\neq y_i\end{cases}$$
$$=w_m(i)e^{-y_i\alpha_m f_m(x_i)}$$；

(4)强分类器更新：$F_m(x)=F_{m-1}(x)+\alpha_m f_m(x)$。

输出：强分类器 $\mathrm{sgn}(F_M(x))$，$F_M(x)=\sum\limits_{m=1}^M\alpha_m f_m(x)$。

boosting 算法中，第 m 次迭代时，弱学习器 L 负责学习当前样本分布 D_m 下的弱分类器 $f_m(x):X\to Y$，其性能用样本分布 D_m 下的分类误差来度量：

$$e_m=\Pr_{i\sim D_m}[f_m(x_i)\neq y_i]=\sum_{i;f_m(x_i)\neq y_i}D_m(i)$$

样本分布由样本权重归一化计算得出：

$$D_m(i)=\frac{w_m(i)}{\sum\limits_i w_m(i)}$$

因此 e_m 又称为加权分类误差(weighted classification error)。这里，弱学习器 L 可以是任意学习算法。若 L 无法使用样本权重，可根据分布 D_m 对训练集进行重采样，用得到的无权重重采样样本进行训练。

1.2 间隔和损失函数

文献[3]指出，boosting 可看作在函数空间内按梯度下降法寻找一组基函数(弱分类器)$\{f_m(x)\}_{m=1}^M$ 来最小化训练集累积损失 $\sum\limits_{i=1}^N L(y_i,F(x_i))=\sum\limits_{i=1}^N L(y_i,\sum\limits_{m=1}^M\alpha_m f_m(x_i))$。

分类误差定义如下：

$$\mathrm{error}(y,F(x))=\begin{cases}1, & \text{if } yF(x)<0\\ 0, & \text{if } yF(x)\geqslant 0\end{cases}$$

损失函数 $L(y,F(x))=e^{-yF(x)}$，可看作分类误差上界，用来衡量分类损失。称 $yF(x)$ 为样本 (x,y) 关于判决函数 $F(x)$ 的间隔(margin)，间隔可同时反映分类结果和分类置信度。间隔为正时，分类正确；反之，分类错误。且间隔越大，代表分类置信度越高。损失函数是关于间隔的减函数，间隔越大，损失越小；间隔较小时，损失显著增大。

第 m 次迭代，给出 F_{m-1}，求取 f_m 和 α_m，以最小化当前训练集累积损失：

$$(\alpha_m,f_m)=\arg\min_{\alpha,f}\sum_{i=1}^N L(y_i,F_{m-1}(x_i)+\alpha f(x_i))$$

按类牛顿法推导得到的解即为 boosting 弱学习器生成的最优弱分类器及其权重(具体见文献[4])。此时，样本权重

$$w_m(i)=\prod_{l=1}^{m-1}e^{-y_i\alpha_l f_l(x_i)}=e^{-y_i F_{m-1}(x_i)}=L(y_i,F_{m-1}(x_i))$$

为样本 (x_i,y_i) 在当前分类器 F_{m-1} 下的损失。难以区分的样本其间隔较小，对应损失较大，样本权重也较大。本轮弱分类器 f_m 更侧重对其正确分类。

2 基于 boosting 的不平衡数据分类算法

原始 boosting 算法对不同类别的样本采用相同的损失函数 $L(y,F(x))$，不适用于不平衡数据分类。目前已有不少基于改进 boosting 的不平衡数据分类方法，本文将其分为数据采样法和代价敏感法两类。下面分别进行介绍。

2.1 数据采样法

数据采样法将数据采样技术与 boosting 算法相结合，在每轮迭代使用过采样(oversampling)/欠采样(undersampling)降低类别间的不平衡程度。SMOTEBoost 在每轮迭代初始采用 SMOTE(Synthetic Minority Over-sampling Technique，少类样本合成过采样技术)生成新的人造正样本，以改善数据不平衡给分类器带来的不良影响[5]；RUSBoost 采用随机降采样平衡样本分布[6]；DataBoost-IM 将当前权重较大的样本作为种子样本，产生具有平衡分布的数据[7]。数据采样法计算量大，采样过程容易引入噪声或丢失有用数据信息，采样数量难以确定，不易操作。

2.2 代价敏感法

代价敏感法基于代价敏感学习(cost-sensitive learning)，通过对正负类别定义不同的误分代价，使得分类器侧重于正样本的正确判别。代价敏感法将数据不平衡问题转化为代价敏感问题[8-11]，是广为使用的一种方法。

样本分类代价定义如下：

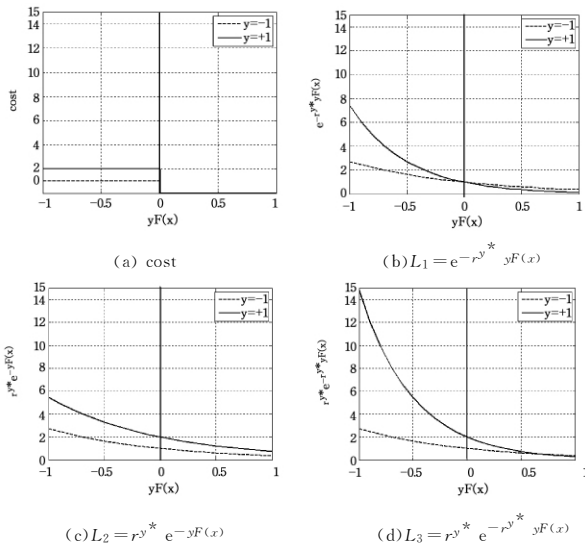
$$\mathrm{cost}(y,F(x))=\begin{cases}r, & \text{if } y=1, yF(x)<0\\ 1, & \text{if } y=-1, yF(x)<0\\ 0, & \text{if } yF(x)\geqslant 0\end{cases}$$

上式说明样本正确分类代价为 0。正负样本错误分类代价比值为 $r>1$，称 r 为代价因子。根据分类代价，构造 3 种代价敏感损失函数 $L(y,F(x),r)$：

$$L_1(y,F(x),r)=e^{-r^{y^*}yF(x)}$$
$$L_2(y,F(x),r)=r^{y^*}e^{-yF(x)}$$
$$L_3(y,F(x),r)=r^{y^*}e^{-r^{y^*}yF(x)}$$

式中，$y^*=(y+1)/2$。图 1 显示了分类代价和代价敏感损失

函数随间隔 $yF(x)$ 变化的曲线，其中 $r=2$。



(a) cost　(b) $L_1=e^{-r^{y^*}yF(x)}$

(c) $L_2=r^{y^*}e^{-yF(x)}$　(d) $L_3=r^{y^*}e^{-r^{y^*}yF(x)}$

图 1　分类代价和代价敏感损失函数

代价敏感损失函数中负样本损失与原始 boosting 相同，正样本损失大于负样本，使得算法更侧重于正样本的正确分类。在函数空间内按梯度下降法最小化代价敏感损失函数在训练集上的期望 $\sum_{i=1}^{N}L(y_i,F(x_i),r)$，可得到代价敏感 boosting 算法。然而，此种方法往往得不到弱分类器及其权重的解析解，需要在执行时迭代求取[11]。更为直接的方法是修改原始 boosting 算法中的权重更新机制，用样本在当前弱分类器下的代价敏感损失更新权重，使得后轮迭代更侧重于正样本的正确分类，即

$$w_{m+1}(i)=w_m(i)L(y_i,\alpha_m f_m(x_i),r)$$

由于在权重更新中引入代价敏感损失，当损失函数设计不好时，随着迭代次数增加，容易产生算法性能不稳定的情况[8]。此时权重为

$$\begin{aligned}w_{m+1}(i)&=w_m(i)L(y_i,\alpha_m f_m(x_i),r)\\&=w_{m-1}(i)L(y_i,\alpha_{m-1}f_{m-1}(x_i),r)L(y_i,\alpha_m f_m(x_i),r)\\&=\cdots\\&=\prod_{l=1}^{m}L(y_i,\alpha_m f_m(x_i),r)\end{aligned}$$

取不同的代价敏感函数，第 $m+1$ 次迭代时权重分别为

$$\begin{aligned}w_{m+1}(i)&=\prod_{l=1}^{m}L_1(y_i,\alpha_m f_m(x_i),r)=e^{-r^{y_i^*}\sum_{l=1}^{m}\alpha_l f_l(x_i)}\\&=L_1(y_i,F_m(x_i),r)\end{aligned}$$

$$\begin{aligned}w_{m+1}(i)&=\prod_{l=1}^{m}L_2(y_i,\alpha_m f_m(x_i),r)=r^{my_i^*}e^{-y_i\sum_{l=1}^{m}\alpha_l f_l(x_i)}\\&=L_2(y_i,F_m(x_i),r^m)\end{aligned}$$

$$\begin{aligned}w_{m+1}(i)&=\prod_{l=1}^{m}L_3(y_i,\alpha_m f_m(x_i))=r^{my_i^*}e^{-ry_i\sum_{l=1}^{m}\alpha_l f_l(x_i)}\\&=r^{(m-1)y_i*}L_3(y_i,F_m(x_i),r)\end{aligned}$$

对于 L_1 和 L_2，样本权重即为其当前分类器 $F_m(x)$ 下的代价敏感损失。不同的是，L_2 中的代价因子随迭代次数呈指数增长。L_3 与 L_2 类似，样本权重为 $r^{(m-1)y_i^*}$ 和当前代价敏感损失的乘积。随着迭代次数增加，正样本损失远远高于负样本，容易造成算法性能不稳定。

3　权重采样 boosting 算法

如第 2 节所述，现有基于 boosting 的不平衡数据分类算法存在一些问题：数据采样法计算量大，容易引入噪声或丢失有用数据信息，代价敏感法中修改权重更新机制容易造成算法性能不稳定。本节提出一种基于权重采样的 boosting 算法，即无需设计代价敏感损失函数，通过采样函数对原始损失函数进行纠正。算法实现时，根据采样函数计算每个样本的采样因子，样本权重乘以采样因子，得到新的数据分布。这种权重采样方式并不产生新数据或删除原有数据，操作简单，避免引入噪声和丢失有用信息。本节对采样函数设计进行深入分析，提出 3 种采样函数形式并在实验部分予以比较。

3.1　算法流程

权重采样 boosting(weight-sampling boosting)算法流程如下：

输入：

- 训练样本集 $\{(x_i,y_i)\}_{i=1}^{N}$，其中 $x_i\in X$，$y_i\in Y=\{-1,+1\}$；
- 弱学习器 L，在给定样本分布 D 下学习弱分类器 $f:X\to Y$；
- 迭代次数 M。

初始化：$F_0(x)=0$，$w_1(i)=1$。

$m=1:M$：

(1)计算样本分布：$D_m^S(i)=\dfrac{s_m(i)w_m(i)}{\sum_i s_m(i)w_m(i)}$，采样因子 $s_m(i)=S(y_i,F_{m-1}(x_i),r)$；

(2)弱学习：用 L 学习采样后分布 D_m^S 下的弱分类器 $f_m:X\to Y$，弱分类器权重为 $\alpha_m=\dfrac{1}{2}\ln\dfrac{1-e_m}{e_m}$，其误差为 $e_m=\Pr_{i\sim D_m^S}[f_m(x_i)\neq y_i]$；

(3)权重更新：$w_{m+1}(i)=w_m(i)e^{-y_i\alpha_m f_m(x_i)}$；

(4)强分类器更新：$F_m(x)=F_{m-1}(x)+\alpha_m f_m(x)$。

输出：强分类器 $\mathrm{sgn}(F_M(x))$，$F_M(x)=\sum_{m=1}^{M}\alpha_m f_m(x)$。

权重采样 boosting 不删除现有样本或生成新样本，而是对每个样本 (x_i,y_i) 关联一个采样因子 $s(i)$，弱学习器在权重采样后的分布 $D_m^S(i)=\dfrac{s_m(i)w_m(i)}{\sum_i s_m(i)w_m(i)}$ 下学习弱分类器 $f_m(x)$。采样因子 $s(i)$ 由采样函数 $S(y,F(x),r)$ 产生，采样函数与样本 (x,y)、样本当前分类结果 $F(x)$ 和代价因子 r 有关，其设计在 3.2 节讨论。图 2 为权重采样 boosting 每轮迭代示意图。

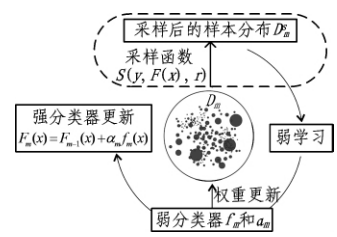


图 2　权重采样 boosting

与原始 boosting 算法相比，新算法仅增加了权重采样操作，不同于修改权重机制的代价敏感 boosting。权重采样 boosting 中权重更新机制保持不变，弱学习器在采样后的权重分布下学习。如此避免了将代价敏感函数直接引入权重更新机制而造成的算法性能不稳定情况。

3.2　采样函数

称 $w_m^s(i)=s_m(i)w_m(i)$ 为采样权重，

$w_m^s(i)=s_m(i)w_m(i)$

$$=S(y_i,F_{m-1}(x_i),r)L(y_i,F_{m-1}(x_i))$$
$$=L^S(y_i,F_{m-1}(x_i),r)$$

不难发现，采样权重实质为原始损失函数经由采样函数修正后的结果。本节讨论 3 种形式不同的采样函数设计，给出经采样函数修正后的损失函数。3 种采样函数均对正样本过采样，即样本为正时，函数值>1；对负样本不进行采样操作，即样本为负时，函数值为 1。

3.2.1 **等同过采样**(Equal-Oversampling，**简称** EOS)

等同过采样增加所有正样本的出现概率，以平衡数据分布。采样函数只与样本类别有关，与样本分类结果无关，所有正样本的采样因子均为 r：

$$S(y,F(x),r)=r^{y^*}=\begin{cases}r, & \text{if } y=1\\ 1, & \text{if } y=-1\end{cases}$$

修正后的损失函数为

$$L^S(y,F(x),r)=r^{y^*}\,\mathrm{e}^{-yF(x)}$$

3.2.2 **误分过采样**(Misclassification-Oversampling，**简称** MOS)

误分过采样通过强调错误分类的正样本，以达到提高正样本检出率的目的。采样函数与样本分类结果有关，正样本错误分类时采样因子为 r，否则不进行过采样：

$$S(y,F(x),r)=\begin{cases}r, & \text{if } y=1,yF(x)<0\\ 1, & \text{otherwise}\end{cases}$$

修正后的损失函数为

$$L^S(y,F(x),r)=\begin{cases}r\mathrm{e}^{-yF(x)}, & \text{if } y=1,yF(x)<0\\ \mathrm{e}^{-yF(x)}, & \text{otherwise}\end{cases}$$

3.2.3 **分界面过采样**(Boundary-Oversampling，**简称** BOS)

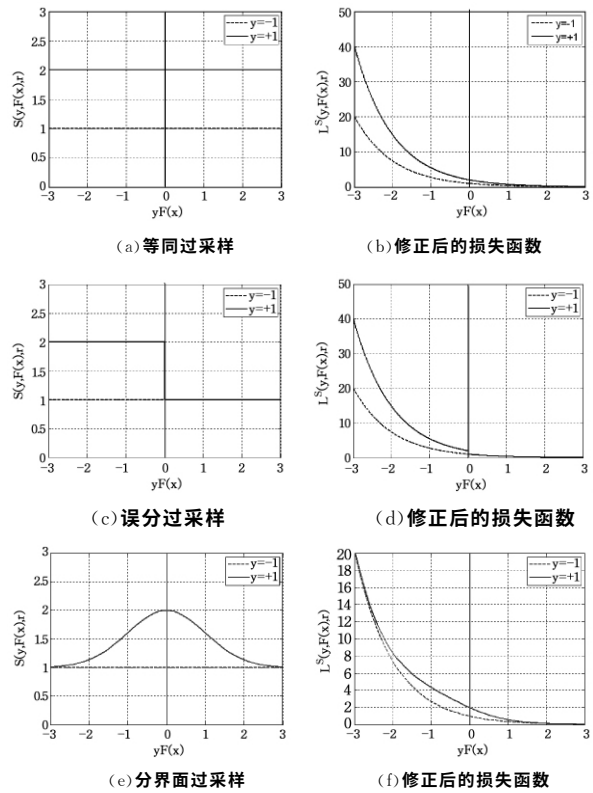
分界面过采样强调分界面附近的正样本，使得分类器侧重于此类样本的正确分类，从而提高正样本整体检出率。采样函数与样本分类结果有关，设计为关于间隔的正态分布函数，对分界面(即间隔 $yF(x)=0$)附近的正样本进行重点采样，而忽略远离分界面的样本：

$$S(y,F(x),r)=1+(r^{y^*}-1)\mathrm{e}^{-\frac{(yF(x))^2}{r}}$$
$$=\begin{cases}1+(r-1)\mathrm{e}^{-\frac{(yF(x))^2}{r}}, & \text{if } y=1,\\ 1, & \text{if } y=-1\end{cases}$$

修正后的损失函数为

$$L^S(y,F(x),r)=\left(1+(r^{y^*}-1)\mathrm{e}^{-\frac{(yF(x))^2}{r}}\right)\mathrm{e}^{-yF(x)}$$
$$=\begin{cases}\left(1+(r-1)\mathrm{e}^{-\frac{(yF(x))^2}{r}}\right)\mathrm{e}^{-yF(x)}, & \text{if } y=1\\ \mathrm{e}^{-yF(x)}, & \text{otherwise}\end{cases}$$

r 取 2 时，3 种采样函数及其修正后的损失函数如图 3 所示。等同过采样修正后的损失函数与代价敏感损失函数 L_2 相同，所有正样本损失均为同间隔下负样本的 r 倍；样本错误判断时，误分过采样与 L_2 相同，样本正确判断时，正负样本损失相同；而经由分界面过采样修正后的损失函数使得分界面附近的正样本损失为负样本的 r 倍，远离分界面的正负样本损失相近。实验部分比较了 3 种采样函数对算法性能的影响。



(a)等同过采样　(b)修正后的损失函数

(c)误分过采样　(d)修正后的损失函数

(e)分界面过采样　(f)修正后的损失函数

图 3　采样函数及经其修正后的损失函数

4　实验

为验证算法有效性，在 UCI 数据集上展开实验，比较以下几种算法：(1) Naïve：原始 boosting 算法；(2) CS1：采用 $L_1(y,F(x),r)=\mathrm{e}^{-r^{y^*}yF(x)}$ 进行权重更新的代价敏感 boosting 算法；(3) CS2：采用 $L_2(y,F(x),r)=r^{y^*}\mathrm{e}^{-yF(x)}$ 进行权重更新的代价敏感 boosting 算法；(4) CS3：采用 $L_3(y,F(x),r)=r^{y^*}\mathrm{e}^{-r^{y^*}yF(x)}$ 进行权重更新的代价敏感 boosting 算法；(5) EOS：采用等同过采样的权重采样 boosting 算法；(6) MOS：采用误分过采样的权重采样 boosting 算法；(7) BOS：采用分界面过采样的权重采样 boosting 算法。为适应数据不平衡问题，所有算法初始权重修改为

$$w_1(i)=\begin{cases}0.5/N_p, & \text{if } y=1\\ 0.5/N_N, & \text{if } y=-1\end{cases}$$

式中，N_p，N_N 分别为正负样本个数。

4.1　数据集

选取 UCI 机器学习数据库中的 5 个不平衡数据集：sonar，wpbc，wdbc，pima 和 bupa[12]，数据集描述如表 1 所列，除 sonar 外，其余均为医疗诊断数据。

表 1　数据集归纳

名称	属性个数	数据集大小	正负类比值
sonar	60	97 个正样本，111 个负样本	0.874
wpbc	33	46 个正样本，148 个负样本	0.311
wdbc	30	212 个正样本，357 个负样本	0.594
bupa	7	145 个正样本，200 个负样本	0.725
pima	8	268 个正样本，500 个负样本	0.536

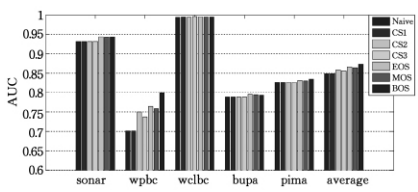
4.2　性能度量

数据集类间分布不平衡时，分类准确性不能有效衡量算法性能，此时用 ROC 曲线(Receiver Operating Characteristic)

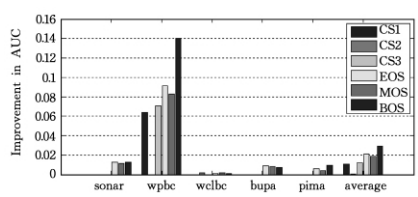
较为合适。ROC 反映分类器判决阈值变化时检出率和虚警率之间的关系，是一种独立于数据集类间分布的性能评价方法，对于数据集的不平衡性有很好的鲁棒性。由于 ROC 曲线并没有给出具体的评价数值，常用曲线下方面积 AUC(Area Under ROC Curve)作为评价指标。AUC 是基于 ROC 曲线的惟一数值，其值越大代表算法性能越好。事实上，AUC 具有统计意义，它表示任取一个正样本和负样本时正样本预测值大于负样本的概率[13]。

4.3 实验结果

实验中，弱分类器为二结点树，采用五折交叉验证，所有算法均进行 200 次迭代。对于不平衡数据分类算法，代价因子 r 以 0.2 为间隔，在 1 至 10 间调节，以获取 AUC 度量下的最好性能。图 4(a)给出了 7 种算法在 5 个数据集上的最佳 AUC 及其平均值，图 4(b)给出了以原始 boosting 算法为基准，6 种不平衡数据分类算法的 AUC 相对提高量。与原始 boosting 算法相比，6 种改进算法均有不同程度的提升，平均提升量 BOS＞EOS＞MOS＞CS3＞CS1＞CS2。在所有数据集上，分界面过采样均得到最好结果，尤其是对于不平衡程度最为严重的 wpbc 数据集(正负类样本比值为 0.311)，分界面过采样明显优于其它 5 种算法。



(a)AUC



(b)AUC 提高量

图 4 实验结果

3 种权重采样 boosting 均优于前人算法，分界面过采样最好，误分过采样相对较差。误分过采样过分强调错误分类的正样本。尤其当样本间隔较小时，正样本损失远大于负样本，使得分类器容易过于拟合小间隔正样本，造成泛化性能下降；等同过采样强调所有正样本而不仅是误分正样本，因此其泛化性能优于误分过采样；分界面过采样强调分界面附近的正样本，忽略远离分界面的正样本，经其修正的损失函数较为合理，可避免因过度拟合间隔较小正样本而造成泛化性能下降。

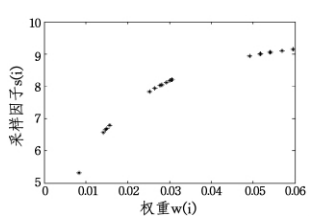


图 5 样本权重与采样因子

图 5 给出分界面过采样中采样因子随样本权重变化的一个实例，代价因子 r 取最优值 9.4。图中的点是某次 wpbc 数据集中用于训练的正样本点，横坐标是某次迭代样本归一化权重，纵坐标为分界面过采样得到的样本采样因子。此时，所有正样本均被正确分类(间隔大于 0)。小权重样本分类置信度高，远离分界面；大权重样本分类置信度低，在分界面附近。从图中可看出，分界面过采样强调分界面附近的样本，即权重越大采样率越高。

结束语 本文研究基于 boosting 的不平衡数据分类算法，提出一种基于权重采样的 boosting，通过采样函数调整原始 boosting 损失函数形式，使得分类器侧重于正样本的有效判别。在此基础上，提出 3 种采样函数形式，其中过采样强调分界面附近的正样本，能得到最好的结果。

与前人算法相比，权重采样 boosting 有以下优点：

(1)采样后的数据集分布由样本权重乘以相应采样因子得到，算法并不真正产生或删除数据，实现简单，避免引入噪声和有用信息丢失；

(2)算法通过采样函数修正原始损失函数，无需直接设计代价敏感损失函数；

(3)采用独立于权重更新机制的权重采样，避免了将代价敏感损失引入权重更新引起的不稳定问题。

参 考 文 献

[1] 高嘉伟，梁吉业. 非平衡数据集分类问题研究进展[J]. 计算机科学，2008，35(4)：10-13

[2] 涂承胜，陆玉昌. Boosting 视角[J]. 计算机科学，2005，32(5)：140-143

[3] Mason L，Baxter J，Bartlett P，et al. Boosting algorithms as gradient descent[C]// Neural Information Processing Systems 12. Cambridge：MIT Press，2000：512-518

[4] Friedman J，Hastie T，Tibshirani R. Additive logistic regression：a statistical view of boosting[J]. The Annals of Statistics，2000，28(2)：337-407

[5] 李正欣，赵林度. 基于 SMOTEBoost 的非均衡数据集 SVM 分类器[J]. 系统工程，2008，26(5)：116-119

[6] Seiffert C，Khoshgoftaar T M，Hulse J V，et al. RUSBoost：Improving classification performance when training data is skewed [C]// Proceedings of 19th International Conference on Pattern Recognition. Washington DC：IEEE Computer Society，2008：1-4

[7] Guo H Y，Viktor H L. Learning from imbalanced data sets with boosting and data generation：the DataBoost-IM approach[J]. SIGKDD Explorations，2004，6(1)：30-39

[8] Sun Y，Kamel M S，Wong A K C，et al. Cost-sensitive boosting for classification of imbalanced data[J]. Pattern Recognition，2007，40(12)：3358-3378

[9] Ge J F，Luo Y P. A Comprehensive Study for Asymmetric AdaBoost and Its Application in Object Detection[J]. Acta Automatica Sinica，2009，35(11)：1403-1409

[10] Li Q J，Mao Y B，Wang Z Q，et al. Cost-sensitive boosting：fitting an additive asymmetric logistic regression model[C]// Proceedings of the 1st Asian Conference on Machine Learning：Advances in Machine Learning (ACML '09). Berlin：Springer，2009：234-247

[11] Masnadi-Shirazi H，Vasconcelos N. Cost-sensitive boosting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2010，33(2)：294-309

[12] Newman D，Hettich S，Blake C，et al. UCI repository of machine learning data bases[DB/OL]. http://www.ics.uci.edu/~mlearn/MLRepository.html，2011-05-01

[13] Hanley J A，McNeil B J. The meaning and use of the area under a receiver operating characteristic(ROC) curve[J]. Radiology，1982，143(1)：29-36