一种新的不平衡数据学习算法 PCBoost

李雄飞[1)] 李 军[1),2)] 董元方[1),3)] 屈成伟[1)]

[1)](吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

[2)](长春理工大学应用数学系 长春 130022)

[3)](长春理工大学经济管理学院 长春 130022)

摘 要 现实世界中广泛存在不平衡数据,其分类问题是机器学习研究中的一个热点.多数传统分类算法假定类分布平衡或误分类代价均衡,在处理不平衡数据时,效果不够理想.文中提出一种不平衡数据分类算法-PCBoost.算法以信息增益率为分裂准则构建决策树,作为弱分类器.在每次迭代初始,利用数据合成方法添加合成的少数类样例,平衡训练信息;在子分类器形成后,修正"扰动",删除未被正确分类的合成样例.文中讨论了数据合成方法,给出了训练误差界的理论分析,并分析了集成学习参数的选择.实验结果表明,PCBoost算法具有处理不平衡数据分类问题的优势.

关键词 数据挖掘;不平衡数据;集成学习;提升;扰动

中图法分类号 TP18 **DOI号**: 10.3724/SP.J.1016.2012.00202

A New Learning Algorithm for Imbalanced Data—PCBoost

LI Xiong-Fei[1)] LI Jun[1),2)] DONG Yuan-Fang[1),3)] QU Cheng-Wei[1)]

[1)](*Key Laboratory of Symbolic Computation and Knowledge Engineering for Ministry of Education, Jilin University, Changchun* 130012)

[2)](*Department of Applied Mathematics, Changchun University of Science and Technology, Changchun* 130022)

[3)](*School of Economics and Management, Changchun University of Science and Technology, Changchun* 130022)

Abstract Imbalanced data exists widely in the real world, and its classification is a hot topic in machine learning. Most traditional classification algorithms assume balanced class distribution or equal misclassification costs, while they do not work when dealing with the imbalanced data. On the one hand, an imbalanced data classification algorithm, named as PCBoost, is proposed in this paper. The algorithm constructs decision tree with information gain ratio as the splitting criterion, and regards the decision tree as a weak classifier. At the beginning of each iteration, the algorithm makes use of data synthesize method to add synthetic minority class examples in order to balance training information. After the sub-classifier is formed, the algorithm corrects the perturbation and deletes the synthetic examples that are not correctly classified. On the other hand, the data synthesize method is discussed, the theoretical analysis of training error boundary is put forward, and the choice of ensemble learning parameters is analyzed. The experimental results show that the PCBoost algorithm has advantages on imbalanced data classification problem.

Keywords data mining; imbalanced data; ensemble learning; boosting; perturbation

---

收稿日期:2010-07-09;最终修改稿收到日期:2012-01-09.本课题得到国家科技支撑计划项目(2006BAK01A33)、吉林省科技发展计划项目(20070321,20090704)资助.**李雄飞**,男,1963年生,教授,博士生导师,主要研究领域为模式识别、数据挖掘.E-mail:lxf@jlu.edu.cn.**李 军**(通信作者),男,1974年生,博士,副教授,主要研究方向为模式识别、数据挖掘.E-mail:lijun.yq@163.com.**董元方**,女,1975年生,博士,讲师,主要研究方向为模式识别、数据挖掘.E-mail:yf.dong@163.com.**屈成伟**,男,1985年生,硕士,主要研究方向为数据挖掘.E-mail:cw.qu@hotmail.com.

1 引　言

近年来，不平衡数据学习问题（Imbalanced Data Learning，IDL）得到了学术界、工业界和政府基金机构的广泛关注，有与之相关的重要研讨会、会议和专刊，包括 AAAI 不平衡数据学习研讨会（AAAI'00）[1]、ICML 的不平衡数据学习研讨会（ICML'03）[2]以及 2004 年 ACM SIGKDD 的专题通讯[3]等.

如果数据集类别分布不均匀，使得其中某个类别占支配地位，则称其为不平衡数据.作为一种特殊的分类学习，不平衡数据学习问题关注的重点是：在数据未被充分表达或严重类分布不平衡情况下，学习算法的性能.多数现有的学习算法假定或期望类分布平衡或误分类代价相等，因此，当处理复杂的不平衡数据集时，这些算法不能有效地表现数据的分布特征，分类结果不能令人满意.

不平衡数据广泛存在于各种领域，如医疗诊断、雷达图像检测、诈骗检测、电信设备故障预测等[4].鉴于不平衡数据学习的重要现实意义，研究者对该问题进行了大量研究，提出的主要解决方案包括数据层面的方案和算法层面的方案，其主要目标是提高少数类的分类精度.本文融合数据采样和 boosting 技术，提出一种不平衡数据学习算法——PCBoost（Perturbation Correction Boosting）.

2 相关工作

多类别数据的不同类之间存在不平衡情形，由于篇幅关系，本文只考虑两类别不平衡学习问题.两类别情况下，通常称少数类为正类，多数类为负类.类别标签的取值分别为$\{-1,+1\}$.

数据采样、代价敏感学习（cost-sensitive learning）、boosting 技术、核方法、主动学习（active learning）以及单类别学习等方法，是处理不平衡数据的常见策略[4].

数据采样技术通过添加少数类样例（过采样，oversampling）或移出多数类样例（欠采样，undersampling）的方式平衡数据的类分布.两者各有优缺点，由于欠采样删除部分训练样例，故其主要缺点是引起信息丢失，而其优点是通过削减训练数据集可以降低训练模型的时间.过采样的主要缺点是若简单地复制原始数据，可能导致过拟合.

与采用不同策略平衡类分布的采样方法不同，代价敏感学习关注错分样例的代价[5].研究表明：代价敏感学习和不平衡数据学习之间存在很强的联系，代价敏感学习的相关理论和算法可以用来解决不平衡数据的学习问题[6].

应用核方法解决不平衡数据学习问题时，主要有 3 种策略：将支持向量机（Support Vector Machine，SVM）与 boosting 结合、核更新方法和偏差惩罚方法.其主要思想是调整由于数据不平衡导致的偏斜的类边界[7].Ertekin 等人[8]提出一种基于 SVM 的主动学习方法，该方法选择最富信息的"未见过"的训练样例，也即，离当前分类超平面最近的样例，重新训练 SVM.间隔（margin）中的数据不平衡比率，低于整个数据集的不平衡比率，因此，该方法可以避免搜索全部数据.但是，搜索最富信息样例的过程计算量很大.

作为集成学习方法的 boosting 技术用于提高分类性能[9].无论数据是否不平衡，都可以通过 boosting 迭代创建集成模型，提升弱分类器的性能.将 boosting 用于不平衡学习问题的优势在于：（1）数据空间重采样自动降低探测最优类分布和代表样例的额外学习代价；（2）通过组合多个分类器避免模型过拟合；（3）降低特定学习算法的偏倚.将 boosting 算法应用于不平衡数据的文献有两类：一类是可以直接应用到大多数分类器的学习算法.例如，AdaCost[10]、CSB1、CSB2[11]、RareBoost[12]和 BABoost[13]等，此类算法主要讨论代价敏感学习和 boosting 技术的结合[14]；另一类是将数据合成方法和 boosting 技术结合的算法，如 SMOTEBoost 和 DataBoost-IM 等.

Song 等人[13]提出一种改进的不平衡数据分类算法 BABoost（Balanced AdaBoost），通过分别计算每个类上的错误率和调整参数 λ 给错分的少数类样例设置更高的权值，实验结果表明 BABoost 显著降低了少数类的预测误差，同时对多数类预测误差影响不大.

SMOTEBoost[15]算法将 Adaboost.M2 与算法 SMOTE[16]整合，在每次迭代中引入合成样例，使每个子分类器更多关注少数类.由于各子分类器建立在不同的数据样本上，集成分类器具有更宽泛的和良性定义的少数类边界.

RUSBoost[17]是 SMOTEBoost 的变形，应用随机欠采样从多数类中随机移出样例，与 SMOTEBoost 相比，算法具有简单易于实现、训练时间短等性能优势.

DataBoost-IM 算法将数据合成技术融合到

AdaBoost.M1,在获得少数类较高的预测精度的同时,并未牺牲多数类的预测精度[18]. 在 boosting 迭代时识别多数类和少数类的困难样例,然后根据困难样例分别生成多数类和少数类的合成样例. DataBoost-IM 依据类间学习样例的比率生成合成样例. 最后,根据新加入合成样例,更新权值分布.

SMOTEBoost 采用 SMOTE 算法通过"插值"的方式添加合成样例,造成合成样例只分布在原始样例的连线上,不能很好反映样例的分布. 与之相比,随机采样方式能更好地模拟数据的真实分布,而且,SMOTE 算法相对于随机采样过于复杂. RUSBoost 算法虽然克服了过采样引起信息丢失的问题,但其缺点是没有最大限度地"拓展"少数类边界. DataBoost-IM 算法同时生成两类样例,而且没有及时"修正"错误添加的合成样例,其迭代过程将面临过多的训练数据.

本文提出一种融合数据采样和 boosting 技术的不平衡数据分类算法——PCBoost(Perturbation Correction Boosting). 在每次迭代初始,利用随机采样方式的数据合成方法,添加合成的少数类样例,平衡训练信息,并及时进行"扰动修正"(Perturbation Correction),删除错分的合成样例. PCBoost 发挥 boosting 技术提升分类器性能的作用,有效"拓展"少数类边界. 理论和实验结果表明,PCBoost 在处理不平衡数据时性能更优.

本文第 3 节描述 PCBoost 算法,并给出数据合成方法和对训练权值更新的讨论;第 4 节讨论训练误差的界,子分类器集成的权重选择以及合成样例连续型属性的处理;第 5 节给出 UCI 数据上的实验结果;第 6 节给出结论和进一步研究展望.

3 不平衡数据挖掘算法 PCBoost

PCBoost 算法包括 3 个阶段:首先对原始数据集的每个样例设置相同的初始权值;其次,调用数据合成方法,生成 m 个合成样例平衡少数类训练信息,合成样例添加后需要规范化权值;第 3 阶段调用弱学习算法,形成子分类器,并通过权值更新过程使得下一次迭代时,被当前子分类器错分的"困难样例"能够得到更多关注,同时修正扰动数据,消除"错误"添加的合成样例对集成学习的影响. 第 2 阶段和第 3 阶段重复执行,直到达到迭代次数 T. 最后,将子分类器集成.

3.1 数据合成方法

数据合成的目的是通过添加合成样例,平衡少数类的训练信息. 以 O 表示所有原始样例集,$O_{\min}$ 表示所有少数类样例集. 在每一次迭代开始时,将少数类 $O_{\min}$ 中的样例视为"种子",利用数据合成方法生成合成样例. 以 S 表示所有合成样例集,S_t 表示第 t 次迭代时生成的合成样例集,其数目与原始少数类样例数目相同,也即 $|S_t|=|O_{\min}|=m$. 合成样例的类别标记与少数类相同.

生成合成样例时,基于如下方法生成属性值:对于离散型属性,首先获得在该属性下少数类样例的属性值分布,然后,数据合成算法根据该分布随机选择一个属性值. 设属性 a 为离散型属性,在该属性下少数类样例的属性值域为 $\{a_1,a_2,\cdots,a_m\}$,每个属性值的出现频率分别为 $p_1,p_2,\cdots,p_m$,设 $e(a)$ 表示合成样例 e 的属性 a 的属性值,首先利用随机数发生器给出一个(0,1)上的均匀分布随机数 ξ,然后根据下式选取合成样例的属性值:

$$e(a)=\begin{cases}a_1, & 0\le\xi\le p_1\\ a_j, & \sum_{i=1}^{j-1}p_i<\xi\le\sum_{i=1}^{j}p_i,\ j=2,\cdots,m\end{cases} \tag{1}$$

由于 ξ 是均匀分布于区间[0,1]的,因此合成数据的属性取值的概率分布是对真实分布的近似.

对于连续型属性,数据合成方法根据该属性分布的均值和方差,随机生成属性值. 设属性 a 为连续型属性,其取值的均值和方差分别为 μ 和 σ^2,在生成合成数据时,利用随机数发生器给出服从正态分布 $N(\mu,\sigma^2)$ 的随机数 ξ,表示合成样例的属性值 $e(a)$. 如果有诸多因素起作用,而每一种因素都不起主导作用时,根据中心极限定理可知,符合这种特点的随机变量近似服从正态分布. 另外,正态分布是最常见的连续型随机变量分布,生产生活中的许多现象服从或近似服从正态分布. 因此,PCBoost 采用正态分布生成合成样例连续属性值. 采用正态分布效果不理想时,可以按 4.3 节给出的方法进行处理.

在生成合成样例时,采用独立的方式生成合成样例属性值,所产生的数据分布与真实分布有偏差. PCBoost 算法的步 6 利用式(3)更新样例权值,将错分的合成样例权值设为 0,也即删除错分的合成样例,从而,可以减少不符合真实分布的合成样例对分类效果的影响.

按照上述方法生成每个属性的取值后,即可得到一个合成少数类样例. 数据合成过程结束后,将 S_t 中的合成样例添加到 S 中,并以 $O\cup S$ 作为训练数据集训练第 t 个子分类器,由于逐次添加了若干合成样例,因此,数据分布逐渐趋于平衡.

3.2 训练数据权值更新

第 t 次迭代过程中，有两次训练数据权值更新，分别是合成样例添加之后和子分类器形成之后.

假设第 t 次迭代添加合成样例之后的训练数据集的基数 $|O\cup S|=n_t$，由于 $|S_t|=|O_{\min}|=m$，因此对于合成样例集 S_t 中的每个合成样例，设定其权值为 $\frac{1}{n_t}$，为规范化权值，$|O\cup S|$ 的原有样例权值应调整为原来权值的 $\frac{n_t-m}{n_t}$，合成样例加入后的权值更新为

$$D_t^{\text{new}}(i)=\begin{cases}\dfrac{1}{n_t}, & x_i\in S_t\\ D_t^{\text{old}}(i)\times\dfrac{n_t-m}{n_t}, & x_i\notin S_t\end{cases}\tag{2}$$

其中，D_t^{old} 和 D_t^{new} 分别表示第 t 次迭代时，合成样例加入前及加入后的样例权值.

定理 1. $D_t^{\text{new}}(i)$ 是规范化的.

由式(2)，易知定理 1 成立.

第 t 次迭代训练结束时，得到子分类器 $h_t:x\to\{-1,1\}$，$(t=1,\cdots,T)$，$h_t(x)$ 给出样例 x 所属类别. 根据子分类器 h_t 的分类情况，更新样例权值，减少正确分类样例权值，增加错分原始样例的权值，以便在下一次迭代时，“困难”样例得到更多关注. 同时，被子分类器 h_t 错分的合成样例，相当于扰动数据. 因此，需将错分的合成样例权值设为 0，使得在下次迭代前，这些样例已经删除，从而起到扰动修正的作用. 子分类器 h_t 形成后的权值更新公式如下：

$$D_{t+1}(i)=\begin{cases}\dfrac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}, & x_i\in O\cup S-S_t\ \vee\\ & (x_i\in S_t\wedge y_i=h_t(x_i))\\ 0, & x_i\in S_t\wedge y_i\neq h_t(x_i)\end{cases}\tag{3}$$

其中 Z_t 为规范化因子：

$$Z_t=\sum_{\substack{x_i\in O\cup S-S_t\\ \vee(x_i\in S_t\wedge y_i=h_t(x_i))}}D_t(i)\exp(-\alpha_t y_i h_t(x_i))$$

PCBoost 算法

输入. 样例集合 $\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$，$y_i\in\{-1,1\}$ 为类别标签，设其中的少数类样例个数为 m 个；迭代次数 T；弱学习算法 WeakLearn.

初始化. 对于 $\forall i$，令 $D_0(i)=\frac{1}{n}$

步骤. for $t=1,\cdots,T$：

1. 调用数据合成方法，生成 m 个合成样例，以平衡少数类训练信息；

2. 添加合成样例，利用式(2)更新训练数据集权值. 原始样例集 O 与合成样例集 S 形成训练数据集 $O\cup S$；

3. 在 $O\cup S$ 上用弱学习算法 WeakLearn，获得子分类器 $h_t:x\to\{-1,1\}$；

4. 计算子分类器 h_t 的误差：ε_t（见 4.2 节式(6)）；

5. 令 $\alpha_t=\frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$（见 4.2 节式(7)）；

6. 利用式(3)更新样例权值

输出. $H(x)=\operatorname{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right)$.

其中，符号函数 sign 定义为

$$\operatorname{sign}(x)=\begin{cases}1, & x\geq 0\\ -1, & x<0\end{cases}.$$

4 训练误差的界与参数选择

4.1 训练误差的界

Schapire 和 Singer 给出了 AdaBoost 算法训练误差的界，并讨论了参数 α_t 的选择[19]. 定理 2 给出本文 PCBoost 算法训练误差的界.

定理 2. 最终分类器 H 的训练误差的界为

$$\frac{1}{n}|\{x_i:x_i\in O\wedge H(x_i)\neq y_i\}|\leq\prod_{t=1}^{T}Z_t.$$

证明. 消去权值更新规则的递归，得

如果 $x_i\in O$，则

$$D_{T+1}(i)=\frac{\exp\left(-\sum_t\alpha_t y_i h_t(x_i)\right)}{n\prod_t Z_t}=\frac{\exp(-y_i H(x_i))}{n\prod_t Z_t}\tag{4}$$

若 $H(x_i)\neq y_i$，则 $y_i H(x_i)\leq 0$，从而，

$$\exp(-y_i H(x_i))\geq 1,$$

因此有

$$I(H(x_i)\neq y_i)\leq\exp(-y_i H(x_i))\tag{5}$$

其中 I 为示性函数. 由式(4)、式(5)可知训练误差的界为

$$\begin{aligned}&\frac{1}{n}|\{x_i:x_i\in O\wedge H(x_i)\neq y_i\}|\\ &\leq\frac{1}{n}\sum_{x_i\in O}\exp(-y_i H(x_i))\\ &=\sum_{x_i\in O}\left(\prod_{t=1}^{T}Z_t\right)D_{T+1}(i)\\ &\leqslant\sum_{x_i\in O\cup S}\left(\prod_{t=1}^{T}Z_t\right)D_{T+1}(i)\\ &=\prod_{t=1}^{T}Z_t\sum_{x_i\in O\cup S}D_{T+1}(i)=\prod_{t=1}^{T}Z_t.\end{aligned}$$

证毕.

定理 2 给出了训练误差的界与迭代中的规范化因子 Z_t 的关系.

4.2 参数 α_t 的选择

由定理 2 可知，可以通过最小化每次迭代的 Z_t

来最小化误差界,从而降低训练误差.

注意,当 $x_i \in S \wedge y_i \neq h_t(x_i)$ 时,$D_t(i)=0$,从而

$$\sum_{\substack{x_i \in S \\ y_i \neq h_t(x_i)}} D_t(i)\left(\frac{1-y_i h_t(x_i)}{2} e^{\alpha_t}\right)=0,$$

另外,注意 $y_i \in \{-1,1\}$,$h_t(x_i) \in \{-1,1\}$,因此

$$\begin{aligned} Z_t &= \sum_i D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \\ &= \sum_{x_i \in O} D_t(i)\left(\frac{1+y_i h_t(x_i)}{2} e^{-\alpha_t} + \frac{1-y_i h_t(x_i)}{2} e^{\alpha_t}\right) + \\ &\quad \sum_{\substack{x_i \in S \\ y_i = h_t(x_i)}} D_t(i)\left(\frac{1+y_i h_t(x_i)}{2} e^{-\alpha_t}\right). \end{aligned}$$

$$\begin{aligned} \frac{dZ_t}{d\alpha_t} &= \sum_{\substack{x_i \in O \\ y_i = h_t(x_i)}} -D_t(i)e^{-\alpha_t} + \sum_{\substack{x_i \in O \\ y_i \neq h_t(x_i)}} D_t(i)e^{\alpha_t} + \\ &\quad \sum_{\substack{x_i \in S \\ y_i = h_t(x_i)}} -D_t(i)e^{-\alpha_t}. \end{aligned}$$

令 $\frac{dZ_t}{d\alpha_t}=0$,可得 α_t 的形式如下:

$$\begin{aligned} \alpha_t &= \frac{1}{2}\log\left(\frac{\sum_{\substack{x_i \in O \\ y_i = h_t(x_i)}} D_t(i) + \sum_{\substack{x_i \in S \\ y_i = h_t(x_i)}} D_t(i)}{\sum_{\substack{x_i \in O \\ y_i \neq h_t(x_i)}} D_t(i)}\right) \\ &= \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right), \end{aligned}$$

其中

$$\begin{aligned} \varepsilon_t &= \sum_{\substack{x_i \in O \\ y_i \neq h_t(x_i)}} D_t(i) + \sum_{\substack{x_i \in S \\ y_i \neq h_t(x_i)}} D_t(i) \\ &= \sum_{\substack{x_i \in O \\ y_i \neq h_t(x_i)}} D_t(i) \\ &= \frac{1}{2}\sum_{i=1}^{n_t} D_t(i) \mid h_t(x_i) - y_i \mid \end{aligned} \tag{6}$$

显然,由式(6)的形式可知,ε_t 是第 t 次迭代所得子分类器的训练误差. 基于上述讨论,在 PCBoost 迭代算法中,为最小化误差界,参数 α_t 应取为

$$\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{7}$$

4.3 合成样例连续型属性的分布

对于连续型属性,采用正态分布生成属性值,效果不够理想时,可以按如下方法处理:对常见连续型分布(均匀分布、正态分布、拉普拉斯分布、韦布尔分布、指数分布、对数正态分布等)做皮尔逊 χ^2 检验,利用分布函数的拟合优度检验,确定属性值的分布类型;然后,依据具体的分布类型给出合成样例的连续属性值.

假设 $F_k(x;\theta_1,\theta_2,\cdots,\theta_{l_k})$,$k=1,\cdots,K$ 是待检验的 K 个连续型分布函数,F_k 含有 l_k 个未知参数. a 是连续型属性. 对于第 k 个待检验的分布 F_k,首先,用未知参数的极大似然估计量 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_{l_k}$ 代替 $\theta_1,\theta_2,\cdots,\theta_{l_k}$,使得 $F_k(x;\theta_1,\theta_2,\cdots,\theta_{l_k})$ 不含未知参数,然后,按照以下步骤进行皮尔逊 χ^2 检验:

(1) 对所讨论连续型属性 a 计算属性值的频数分布;

(2) 根据分布 $F_k(x;\theta_1,\theta_2,\cdots,\theta_{l_k})$ 计算理论频数;

(3) 建立皮尔逊 χ^2 统计量,在给定的显著性水平 α 之下,进行显著性检验.

通过调整显著性水平 α,使得仅有一种分布假设被接受. 在合成样例时,利用该分布假设生成连续属性值. 数据合成方法根据该分布的参数,利用随机数发生器给出服从相应分布的随机数 ξ,表示合成样例的属性值 $e(a)$.

5 实验分析

5.1 数据集

为评估算法的性能,选择 10 组具有不同实际应用背景的 UCI 数据. 对于含有多个类别的数据,合并某些类别或只取两个类别.

表 1 是用于实验的数据信息,包括数据集大小、少数类样例的比例、属性个数等. 其中 Glass 的 headlamps 类作为少数类,合并其它类别作为多数类. Vowel 的 hed 类作为少数类,其它类合并作为多数类. Vehicle 的 van 类作为少数类,其它类合并作为多数类. 取 Segment 的 grass 类作为少数类,其它类合并作为多数类. 取 Satimage 的 damp grey soil 类为少数类,其它类合并作为多数类. 取 Abalone 数据的第 18 类为少数类,第 9 类为多数类.

表 1 UCI 数据集信息

数据集	样例数目	少数类	多数类	类分布	属性(连续/离散)
Sonar	208	97	111	0.47:0.53	60/0
Monk2	169	64	105	0.37:0.63	0/6
Ionosphere	351	126	225	0.35:0.65	34/0
Breast-W	699	241	458	0.34:0.66	9/0
Vehicle	846	199	647	0.23:0.77	18/0
Segment	2310	330	1980	0.14:0.86	19/0
Glass	214	29	185	0.13:0.87	9/0
Satimage	6435	626	5809	0.097:0.903	33/0
Vowel	990	90	900	0.09:0.91	10/3
Abalone	731	42	689	0.06:0.94	7/1

一般认为当少数类与多数类的类分布比例低于 1∶2 时,数据集具有不平衡特征. 表 1 所示数据集具有不同不平衡比例. 其中,Sonar 数据集基本是平衡数据集,选取该数据集的目的是验证 PCBoost 算

法对一般数据集的有效性.

5.2　评价度量

在评价分类性能和指导分类器建模时，评估度量起着至关重要的作用. 机器学习领域对于不平衡数据分类的常用评价标准包括 ROC 曲线、AUC 以及基于混淆矩阵的若干度量，如查全率(*recall*)、查准率(*precision*)、*F-measure* 和 *G-mean* 等. 在两类别情形下，将训练样例少，但具有高识别重要性的少数类视为正类，多数类视为负类. 经过分类过程后，训练样例可以分为混淆矩阵中所表示的 4 种情况，如表 2 所示.

表 2　混淆矩阵

	预测正类	预测负类
实际正类	True Positives (TP)	False Negatives (FN)
实际负类	False Positives (FP)	True Negatives (TN)

利用混淆矩阵，可以派生出几个度量：

真实正类率：

$$TP_{\text{rate}} = TP/(TP+FN).$$

真实负类率：

$$TN_{\text{rate}} = TN/(TN+FP).$$

正类预测值：

$$PP_{\text{value}} = TP/(TP+FP).$$

如果只考虑正类的性能，真实正类率 TP_{rate} 和正类预测值 PP_{value} 是重要的度量. 在信息检索领域，将真实正类率 TP_{rate} 定义为查全率 *recall*，表示检索到的相关对象占实际正类的比例. 将正类预测值 PP_{value} 定义为查准率 *precision*，表示相关对象占检索出的所有对象的比例.

F-measure 是查全率和查准率的调和均值，其取值接近两数的较小者，因此，较大 *F-measure* 值表示 *recall* 和 *precision* 都较大：

$$F\text{-}measure = \frac{(1+\beta^2)recall \times precision}{\beta^2 \times recall + precision},$$

其中 β 用于调节 *precision* 和 *recall* 的相对重要度，通常取为 1.

如果同时关注两个类的性能，也即，希望 TP_{rate} 和 TN_{rate} 都取较大值，可以使用 *G-mean* 度量学习算法在两个类上的平均性能：

$$G\text{-}mean = \sqrt{TP_{\text{rate}} \times TN_{\text{rate}}}.$$

本文采用 *F-measure* 和 *G-mean* 作为评价度量.

5.3　实验结果

使用基于信息增益率为分裂属性的剪枝决策树作为 PCBoost 弱学习算法，对每个数据集实施 10-折交叉验证.

在 PCBoost 算法逐次添加合成样例过程中，少数类边界得到有效拓展，使得分类边界更接近于多数类. 随着合成样例的添加，*TP*、*FP* 将增大，*TN*、*FN* 将减小. 根据 TP_{rate}、TN_{rate}、PP_{rate} 的定义可以看出，这将导致 TP_{rate}(即 *recall*)增大，TN_{rate} 降低，但 PP_{rate}(即 *precision*)的变化取决于 *TP*、*FP* 变化的程度，当 *TP* 增加的比例大于 *FP* 增加的比例时，PP_{rate} 将增大.

根据 *F-measure* 的定义，其取值接近于 *recall* 和 *precision* 的较小者，因此，如果算法有效地获得较大的 *F-measure*，则说明算法的 *recall* 和 *precision* 都取较大值，也即，算法的少数类查全率和查准率都较高. 同理，根据 *G-mean* 的定义，如果算法有效地获得较大的 *G-mean*，则说明算法的 TP_{rate} 和 TN_{rate} 都取较大值，也即算法在两个类别上的精度都较高.

表 3 给出在 10 个不同的不平衡数据上 PCBoost 与 C4.5、AdaBoost.M1、SMOTEBoost 和 DataBoost-IM 的比较结果. 图 1 及图 2 分别给出不同数据集上各算法的 *F-measure* 和 *G-mean* 的比较.

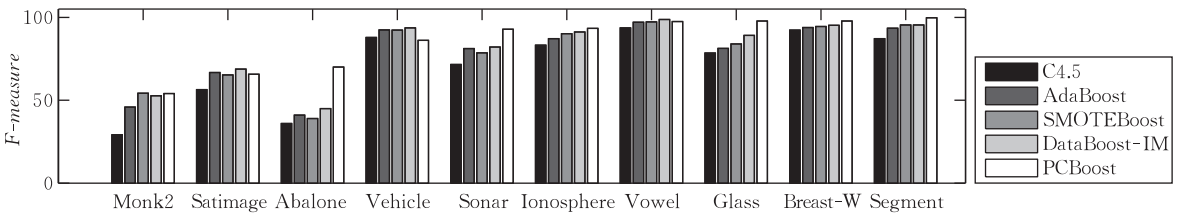

图 1　不同算法的 *F-measure* 比较

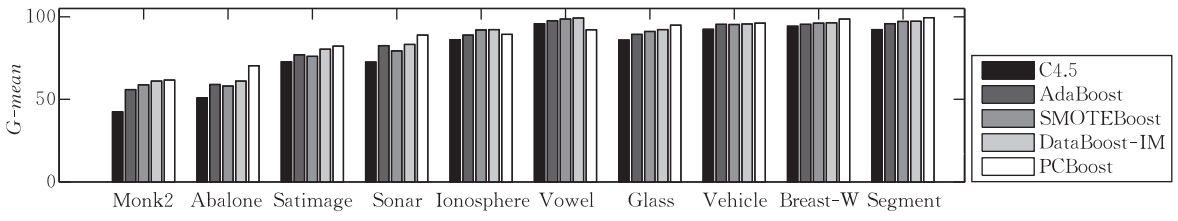

图 2　不同算法的 *G-mean* 比较

表 3 PCBoost 算法与其它算法的比较

数据集	方法	*F-measure*	*G-mean*
Glass	C4.5	78.5	85.9
	AdaBoost.M1	81.3	89.4
	SMOTEBoost	84.0(N=100)	91.1(N=100)
	DataBoost-IM	89.2	92.3
	PCBoost	97.8	94.9
Satimage	C4.5	56.4	72.7
	AdaBoost.M1	66.7	77.0
	SMOTEBoost	65.3(N=300)	76.0(N=300)
	DataBoost-IM	68.8	80.4
	PCBoost	65.7	82.2
Vowel	C4.5	93.7	95.8
	AdaBoost.M1	97.1	97.6
	SMOTEBoost	97.3(N=100)	98.7(N=100)
	DataBoost-IM	98.8	99.3
	PCBoost	97.5	92.1
Abalone	C4.5	36.0	50.8
	AdaBoost.M1	41.0	59.0
	SMOTEBoost	39.0(N=300)	58.1(N=300)
	DataBoost-IM	45.0	61.1
	PCBoost	70.0	70.4
Segment	C4.5	87.2	92.2
	AdaBoost.M1	93.5	95.9
	SMOTEBoost	95.4(N=300)	97.2(N=300)
	DataBoost-IM	95.5	97.3
	PCBoost	99.8	99.5
Sonar	C4.5	71.6	72.6
	AdaBoost.M1	81.2	82.5
	SMOTEBoost	78.6(N=100)	79.3(N=100)
	DataBoost-IM	82.1	83.3
	PCBoost	92.9	88.9
Monk2	C4.5	29.2	42.4
	AdaBoost.M1	45.9	55.9
	SMOTEBoost	54.3(N=300)	58.8(N=300)
	DataBoost-IM	52.7	61.1
	PCBoost	54.1	61.7
Ionosphere	C4.5	83.3	86.2
	AdaBoost.M1	87.2	88.9
	SMOTEBoost	90.2(N=100)	92.0(N=100)
	DataBoost-IM	91.2	92.3
	PCBoost	93.4	89.4
Breast-W	C4.5	92.4	94.3
	AdaBoost.M1	93.9	95.4
	SMOTEBoost	94.5(N=500)	96.2(N=500)
	DataBoost-IM	95.2	96.4
	PCBoost	97.8	98.7
Vehicle	C4.5	87.9	92.5
	AdaBoost.M1	92.5	95.5
	SMOTEBoost	92.4(N=100)	95.3(N=100)
	DataBoost-IM	93.7	95.7
	PCBoost	86.2	96.2

可以看出在 Glass、Abalone、Segment、Breast-W、Sonar 上，PCBoost 的两个度量指标均优于其它算法；在 Monk2、Ionosphere、Vehicle、Satimage 上 PC-Boost 的一个度量指标优于其它算法，并且另一度量指标也是良好的. 在 Vowel 上的度量指标略低于 DataBoost-IM，其 *F-measure* 与 SMOTE-Boost 相当. Vowel 数据集上 *G-mean* 较低的原因，主要是由于随着合成样例的添加，TN_{rate} 降低程度大于 TP_{rate} 增大程度造成的.

虽然 Bartlett 和 Traskin 证明了 AdaBoost 算法具有一致性[20]，但是，Schapire 指出 AdaBoost 算法的收敛性仍然是一个值得研究的问题[21]. 对于每一个数据集，在调用 PCBoost 算法时，为了选取能够取得最佳预测性能的分类器，需要确定恰当的迭代次数. 实验中选取若干不同的迭代次数，然后根据在验证集上的结果优劣，选出能够获得最佳分类性能的迭代次数. 与 AdaBoost 算法相比，PCBoost 不仅改变样例权值，而且增加新的样例，这更有利于分类边界的确定. 实验结果表明，PCBoost 算法能够更快地收敛.

6 结 论

本文提出一种新的不平衡数据分类算法—PCBoost，算法融合了数据合成采样技术和 boosting 技术，逐步渐进地增加合成的少数类样例，平衡训练信息，并及时删除误分的合成样例，通过扰动修正，避免了不恰当的人工合成样例对集成分类器的影响. 同时，从理论上分析了训练误差的界，并讨论了子分类器集成权重的选择依据. 通过 UCI 数据集实验，以 *F-measure* 和 *G-mean* 为度量对算法进行评价，与决策树算法、标准 AdaBoost 算法以及两种基于采样和 boosting 融合的算法进行比较，实验结果表明 PCBoost 算法具有处理不平衡数据的优势.

考虑属性间相关性的数据合成方法对 PCBoost 算法效率的作用，是值得深入研究的课题. 进一步研究将从实验上研究不同数据合成方法对 PCBoost 算法的影响以及从理论上分析 PCBoost 算法误差界与 AdaBoost 算法误差界的关系.

致 谢 审稿专家提出了宝贵建议. 在此表示感谢!

参 考 文 献

[1] Japkowicz N. Learning from imbalanced data sets: A comparison of various strategies//Proceedings of the AAAI 2000 Workshop, 2000: 10-15

[2] Chawla N V, Japkowicz N, Kotcz A. Workshop on learning from imbalanced data sets//Proceedings of the ICML'2003. Washington, DC, USA, 2003

[3] Chawla N V, Japkowicz N, Kolcz A. Editorial: Special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 1-6

[4] He Hai-Bo, Garcia E A. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(9): 1263-1284

[5] Ling Charles X, Sheng Victor S. A comparative study of cost-sensitive classifiers. Chinese Journal of Computers, 2007, 30(8): 1203-1212(in Chinese)
(凌晓峰, Sheng Victor S. 代价敏感分类器的比较研究. 计算机学报, 2007, 30(8): 1203-1212)

[6] Liu X Y, Zhou Z H. The influence of class imbalance on cost-sensitive learning: An empirical study//Proceedings of the 6th International Conference on Data Mining(ICDM'06). Hong Kong, China, 2006: 970-974

[7] Wang B X, Japkowicz N. Boosting support vector machines for imbalanced data sets. Lecture Notes in Artificial Intelligence, 2008, 4994: 38-47

[8] Ertekin S, Huang J, Bottou L, Giles L. Learning on the border: active learning in imbalanced data classification// Proceedings of the ACM Conference on Information and Knowledge Management. Lisbon, Portugal, 2007: 127-136

[9] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997, 55(1): 119-139

[10] Fan W, Stolfo S, Zhang J, Chan P. AdaCost: Misclassification cost-sensitive boosting//Proceedings of the 16th International Conference on Machine Learning. Slovenia, 1999: 97-105

[11] Ting K M. A comparative study of cost-sensitive boosting algorithms//Proceedings of the 17th International Conference on Machine Learning. Stanford University, USA, 2000: 983-990

[12] Joshi M V, Kumar V, Agarwal R C. Evaluating boosting algorithms to classify rare classes: Comparison and improvements//Proceedings of the 1st IEEE International Conference on Data Mining (ICDM'01). California, USA, 2001: 257-264

[13] Song J, Lu X L, Wu X Z. An improved AdaBoost algorithm for unbalanced classification data//Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery. Tianjin, China, 2009: 109-113

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

[15] Chawla N V, Lazarevic A, Hall L O, Bowyer K W. SMOTEBoost: Improving prediction of the minority class in boosting//Proceedings of the 7th European Conference Principles and Practice of Knowledge Discovery in Databases. Cavtat-Dubrovnik, Croatia, 2003: 107-119

[16] Chawla N V, Bowyer K, Hall L, Kegelmeyer W. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 2002, 16: 321-357

[17] Seiffert C, Khoshgoftaar T M, Hulse J V, Napolitano A. RUSBoost: Improving classification performance when training data is skewed//Proceedings of the 19th IEEE International Conference on Pattern Recognition. Tampa, FL, USA, 2008: 1-4

[18] Guo H, Viktor H L. Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM Approach. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 30-39

[19] Schapire R E and Singer Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 1999, 37(3): 297-336

[20] Bartlett P L, Traskin M. AdaBoost is consistent. Journal of Machine Learning Research, 2007, 8: 2347-2368

[21] Schapire R E. The convergence rate of AdaBoost [open problem]//Proceedings of the 23rd Conference on Learning Theory. Haifa, Israel, 2010

LI Xiong-Fei, born in 1963, professor, Ph. D. supervisor. His current research interests include pattern recognition, knowledge discovery and data mining.

LI Jun, born in 1974, Ph. D., associate professor. His current research interests include pattern recognition and data mining.

DONG Yuan-Fang, born in 1975, Ph. D., lecturer. Her current research interests include pattern recognition and data mining.

QU Cheng-Wei, born in 1985, M. S.. His current research interest is data mining.

Background

This work is supported by National Key Technology R&D Program (grand No. 2006BAK-01A33), Technology Development Program of Jilin Province (grand Nos. 20070321, 20090704).

In recent years, imbalanced data learning problem receives more and more attentions in both theoretical and practical aspects. Imbalanced data sets exists in many real-world domains, such as medical diagnostics, radar image detection, fraud detection, telecommunication equipment failures forecast. The main objective of this paper is to propose a new imbalanced data classification algorithm named as PCBoost which combines the data synthesize sampling and boosting technology. In this work, a new data synthesize method is introduced, adding synthetic data is used to balance training information of minority class, and the definition of perturbation correction is introduced to avoid the influence of inappropriate synthetic examples to ensemble classifier. Theoretical analysis and experimental results show that the PCBoost algorithm has advantages on imbalanced data learning problem.