文章编号: 1001-0920 (2013) 07-0978-07

基于样本特性欠取样的不均衡支持向量机

陶新民, 郝思媛, 张冬雪, 李 震

(哈尔滨工程大学 信息与通信工程学院，哈尔滨 150001)

摘 要: 针对传统支持向量机在数据失衡的情况下分类效果很不理想的问题, 提出一种基于样本特性欠取样的不均衡SVM分类算法. 该算法首先在核空间中依据样本信息量选择一定比例的靠近不均衡分类界面的多数类样本; 然后根据样本密度信息选择最具有代表性的均衡多数类样本点, 在减少多数类样本的同时使分类界面向多数类方向偏移. 实验结果表明, 所提出的算法与其他不均衡数据预处理方法相比, 能有效提高SVM算法在失衡数据中少数类的分类性能、总体分类性能和鲁棒性.

关键词: 不均衡数据；支持向量机；样本特性；欠取样

中图分类号: TP391 **文献标志码:** A

Support vector machine for unbalanced data based on sample properties under-sampling approaches

TAO Xin-min, HAO Si-yuan, ZHANG Dong-xue, LI Zhen

(College of Information and Communication Engineering，Harbin Engineering University，Harbin 150001，China. Correspondent：TAO Xin-min，E-mail：taoxinmin@hrbeu.edu.cn)

Abstract: The classification result of classical support vector machine algorithm in the case of unbalanced data set is not satisfactory. Therefore, a under-sampling algorithm based on sample properties is presented. According to sample information in the kernel space, a certain percentage of majority instances located near the classification interface are selected. Then according to the sample's density, the representive majority samples in the selected samples are selected, which can not only reduce the number of majority instances, but also make the SVM classification interface bias toward the majority instances. The experimental results show that compared with other data-preprocess methods for unbalanced dataset classification, the proposed method can improve the classification performance of SVM in the minority class data, the overall classification performance and robustness.

Key words: unbalanced data；support vector machine；sample properties；under-sampling

0 引 言

支持向量机(SVM)是以统计学习理论为基础的一种新型机器学习方法[1]. 它克服了神经网络和传统分类器的过学习、局部极值点和维数灾难等诸多缺点, 具有较强的泛化能力, 现已成为机器学习领域的一个新的研究热点.

SVM方法属于有监督分类算法, 因此需要对数目接近的不同类别样本进行训练才能获得较好的泛化能力. 但是, 实际情况中很多数据样本都是不均衡的, 例如商业欺诈、疾病诊断、文本分类等[2-4]数据集. 对不均衡数据集进行分类时, 各个类别的样本数目存在很大差异, 从而导致不同类别的样本对训练算法提供的信息不对称, 使得SVM算法在处理不均衡数据时, 训练后得到的分类面会向少数类样本偏移[5], 使SVM过度拟合多数类样本点, 低估了少数类样本点, 造成对少数类样本的错分率增大. 如何实现SVM算法在不均衡数据下的正确分类已成为众多学者关注的重点.

目前, 提高不均衡数据下SVM算法性能的研究大多主要集中在算法层面和数据层面. 算法层面的方法是指从SVM分类算法本身入手, 修改已有的分类算法或提出新的算法, 如文献[6]提出的代价敏感算

收稿日期: 2012-03-22；**修回日期:** 2012-08-25.

基金项目: 国家自然科学基金面上项目(61074076)；中国博士后科学基金项目(20090450119)；中国博士点新教师基金项目(20092304120017)；黑龙江省博士后基金项目(LBH-Z08227).

作者简介: 陶新民(1973−), 男, 副教授, 从事智能信号处理、智能计算等研究；郝思媛(1987−), 女, 硕士生, 从事模式识别、信号处理的研究.

法等. 而数据层面研究较多的是如何将数据预处理方法与SVM算法相结合, 其中数据预处理方法又分数据过取样和欠取样. 与过取样结合的方法有: 文献[7]提出的基于随机过取样代价敏感SVM算法、文献[8]提出的基于SMOTE (synthetic minority over-sampling technique) 代价敏感SVM算法及基于边界BSMOTE过取样的SVM算法等. 然而, 过取样算法本身是一个数据依赖算法, 它要求少数类样本集合是个凸集, 即位于两个少数类样本间的实例必须是少数类样本, 同时由于过取样算法额外增加了很多新的训练样本, 导致SVM模型计算代价增大. 欠取样算法则是一个与过取样相反的方法, 它通过减少多数类样本的方式实现数据均衡, 其中包括随机欠取样[9], 借鉴实例简约的DROP算法和CNN算法[10]. 但是, 由于欠取样算法只随机选取了多数类的一个子集, 而选出的子集对改善SVM分类界面是否有效却未知, 如选择不当可能会导致分类效果很不理想[11-16]. 因此, 如何在保证数据均衡的同时, 使得保存的样本信息对决策界面的生成更有效是利用欠取样来提高不均衡数据下SVM算法分类性能的关键.

参考前期研究工作可以发现[17], SVM对数据的不均衡本身并不十分敏感, 只要将SVM分类边界朝多数类进行适当的偏移便可以使更多的少数类样本不被误判. 对于SVM 算法而言, 那些距离分类边界近的样本对分类边界的影响最大, 为此本文提出一种基于样本特性的欠取样策略. 首先利用SVM算法对不均衡数据进行分类; 然后选择距离分类边界近的最具有代表性的多数类样本, 并与少数类样本组合作为训练集对SVM算法进行训练, 如此操作可以实现SVM分类界面向多数类样本偏移, 进而提高SVM算法在不均衡数据下的泛化性能. 将本文算法同其他取样与SVM相结合的算法进行了实验比较, 比较结果表明本文算法在数据不均衡情况下分类性能较其他算法有较大幅度的提高.

1　SVM算法及其不均衡数据分类性能分析

1.1　支持向量机简介

SVM算法是建立在统计学习结构风险最小化原理基础上, 根据有限的样本信息在模型复杂性与学习能力之间寻求最佳折衷, 以期获得最好的泛化能力. 它通过核函数将原始特征空间中的非线性分类界面映射到更高维的特征空间, 以便样本在高维特征空间中变得线性可分.

以两类训练样本集为例, 设给定的训练样本集为 $\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, $y_i\in\{+1,-1\}(i=1,2,\cdots,n)$ 代表样本类别, 核函数为 K, 构造如下代价函数使其最小化:

$$\min\frac{1}{2}||w||^2+C\sum_{i=1}^{n}\xi_i. \tag{1}$$

约束条件为

$$y_i(w^{\mathrm{T}}x_i+b)\geqslant 1-\xi_i,\ \xi_i\geqslant 0,\ i=1,2,\cdots,n. \tag{2}$$

其中: ξ_i 为松弛变量, 表示训练样本的错分程度; C 为惩罚常数, 控制对错分样本的惩罚程度; w 和 b 分别为判决函数 $f(x)=(w\cdot x)+b$ 的权向量和阈值.

拉格朗日函数为

$$L(w,b,\alpha)=\frac{1}{2}||w||^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i[y_i(w^{\mathrm{T}}\cdot x_i+b)-1+\xi_i]-\sum_{i=1}^{n}\beta_i\xi_i, \tag{3}$$

其中 α_i 和 β_i 为拉格朗日算子. 根据KKT条件

$$0\leqslant\alpha_i\leqslant C,\ i=1,2,\cdots,n, \tag{4}$$

$$\sum_{i=1}^{n}\alpha_i y_i=0, \tag{5}$$

$\alpha_i>0$ 的样本是支持向量, 判别函数为

$$f(x)=\mathrm{sgn}\Big(\sum_{i=1}^{n}\alpha_i^* y_i K(x,x_i)+b^*\Big). \tag{6}$$

1.2　SVM在不均衡数据下分类边界的偏移

传统的SVM算法均基于数据集中各类样本数目基本均衡的假设, 显然这一假设在现实应用领域大多不成立. 实际上, 在大多数的应用领域中很多类别往往并不均衡, 数据集中某个类别的样本数可能会远多于其他类别. 另外, 不同类别的分类错误带来的损失也不尽相同, 这就引出了不均衡数据集的分类问题.

为了测试数据不均衡对SVM分类器的影响, 本文选用高斯函数生成的数据集作为测试样本集, 其中一类样本中心为 $(0.3,0.5)$, 另一类样本中心为 $(-0.3,-0.5)$, 方差定为0.5. SVM算法参数设置如下: 选择高斯核函数, 核宽度为10, 惩罚常数选择为 $C=10$, 两类样本数目比例为100:1, 其中少数类样本数为5. SVM算法的分类情况如图1所示.

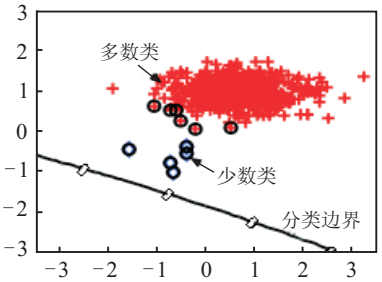

图 1　数据样本比例为100:1时SVM算法的分类边界

从图1的结果可以清楚地看到, SVM分类边界向着少数类方向进行了偏移, 这是由于SVM算法本身的优化函数对不同类别的错误分类采用了相同的

惩罚系数. 在这种条件设置下, 由于少数类样本密度小, 训练后得到的总体训练误差也小. 因此, 为了能使间隔尽可能大的同时尽量降低错分经验风险, 算法学习得到的分类超平面会向样本数量小的类别移动. 这样势必会导致最终的SVM分类器对小数量的样本类别产生较大的测试误差. 因此, 为了提高SVM分类算法对不均衡数据的分类性能, 必须解决在此情况下SVM分类边界偏向于少数类样本的问题.

2 基于样本特性欠取样不均衡SVM分类算法

2.1 传统欠取样算法分析

在多数类样本中存在大量重复信息, 这些冗余信息会导致多数类与少数类样本的数目不均衡, 严重影响了SVM分类器的界面生成, 因此传统的欠取样算法都是通过剔除远离边界的冗余多数类样本并保留有效多数类边界样本的方式来实现数目均衡, 如DROP和CNN等算法. 然而, 这些减少多数类样本数目的欠取样方法并不适合于SVM算法, 这是因为SVM算法的分类边界只与支持向量有关. 因此, 通过删除远离边界的多数类冗余样本来减少多数类样本, 即使能够实现多数类与少数类样本数目间的均衡, 但依然不能改变SVM分类边界的位置, 即无法实现分类边界向多数类样本偏移. 为了说明这一问题, 仍以上例样本为例, 通过删除远离边界冗余样本且只保留原有支持向量的方式实现数据均衡, 训练后的SVM算法分类界面的变化情况如图2所示. 其中: 空心圆标记的是少数类样本, 加号圆标记的是多数类样本(详见图中标识). 通过与图1对比后不难发现, 图1和图2的分类边界没有任何变化, 这是由于初始的支持向量都存在于边界附近, 而由SVM判别公式(6)不难发现, 分类边界的形成只与支持向量有关, 因此均衡前后分类边界没有发生任何变化. 该示例表明, 传统欠取样算法减少多数类样本的方式并不适合于改善不均衡数据下SVM算法的分类性能.

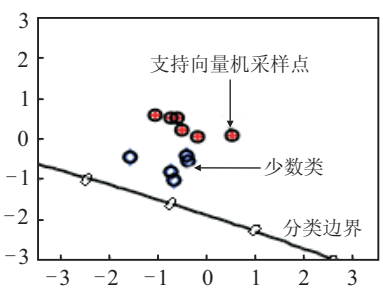

图 2 只保存支持向量样本后SVM算法分类边界的变化

2.2 基于样本特性的欠取样

由SVM原理可以看出, 决策面是只与支持向量有关的超平面, SVM通过使分类间隙最大来设计决策超平面, 以期获得最好的推广能力. 众所周知, 样本的信息量, 即点到决策超平面的距离是判断该点分类性质的主要因素, 距离越近, 对分类界面的影响越大, 因此为了实现决策面向多数类样本偏移, 需在多数类边界样本附近进行采样. 另外, 所选择的样本是否具有一定的空间代表性也是欠取样算法的关键所在.

本文设计一种既能代表多数类样本分布特征, 又能对分类界面有一定影响的样本特性欠取样方法. 该方法采用距离不均衡分类界面的距离作为信息性的度量值, 这是因为距离边界越近, 信息性越强. 首先根据信息度值对多数类样本进行排序, 具体公式如下:

$$\phi^{\mathrm{ME}}(x) = -\|w^{\mathrm{T}} \cdot x + b\|. \tag{7}$$

其中: $\phi^{\mathrm{ME}}(x)$ 为信息量, w 和 b 分别为不均衡分类界面的权向量和阈值.

对于样本代表性的衡量, 常常采用基于高斯核的密度值, 其中相似性度量公式如下:

$$\mathrm{sim}(\boldsymbol{x}, x^{(u)}) = \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_{(u)}\|^2}{2\delta^2}\Big). \tag{8}$$

其中: $u \in U$ 为计算 x 样本密度时考虑的最近邻样本集合; δ 为高斯核半径参数, 该参数对该项的影响很大, 过大则太过泛化, 太小则局部化太强, 因此本文拟采用一种所有多数类样本最小距离中最大距离的倍数, 即

$$\delta = \gamma D = \gamma \max_{x_k}[\min_{x_l}(\|x_k - x_l\|)^2], \tag{9}$$

γ 为固定数值.

为了消除所选出的样本彼此之间的相关性, 即考虑选出的样本之间的差异性, 这里拟采用考虑样本集合间差异性的信息量评测标准, 即

$$\phi^{\mathrm{MID}}(x) = \Big(\lambda \frac{1}{|U|}\sum_{u=1}^{|U|}\mathrm{sim}(x, x^{(u)}) + (1-\lambda)\frac{1}{|Q|}\sum_{q=1}^{|Q|}\mathrm{diff}(x, x^{(q)})\Big), \tag{10}$$

$$\mathrm{diff}(x, x^{(q)}) = 1 - \mathrm{sim}(x, x^{(q)}). \tag{11}$$

其中: Q 为当前采样后得到的多数类样本子集, λ 用于控制二者的权重.

2.3 基于样本特性欠取样不均衡SVM分类算法

为减少训练时的计算量以及不均衡数据给SVM分类界面带来的偏移影响, 本文提出一种基于样本特性欠取样策略的不均衡SVM分类算法. 首先利用原有数据训练得到分类界面(不均衡分类界面); 然后利用式(7)对多数类样本排序, 并选取一定数量($L \gg \mathrm{MI}$, MI为少数类样本)最具信息量的多数类样本; 之后利用式(10)对多数类样本点进行采样, 保留具有一定信息量并具有代表性的MI个多数类样本

作为新的训练样本进行 SVM 学习.

当取 $L = 5 \times \mathrm{MI}$ 的多数类样本时, SVM 算法分类界面的变化情况如图 3 所示. 图中: 正方形点为采样得到的多数类点, 十字形点为多数类样本点, 菱形为少数类样本点. 由图 3 不难看出, 通过本文提出的样本特性欠取样算法处理后, SVM 的分类性能得到了很好改善, 分类边界向着多数类方向偏移.

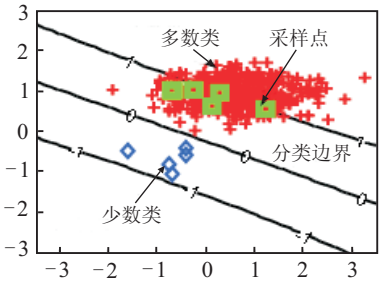

图 3　经过样本特性欠取样后, SVM 算法分类边界的变化

2.4　计算复杂度分析

本文算法的时间复杂度主要取决于 SVM 算法复杂度、排序计算和密度值计算 3 部分. 定义 n 为全部样本数, MD 为多数类样本数, MI 为少数类样本数. SVM 算法的渐近复杂度为 $O(n^2)$; 排序计算的渐近复杂度为 $O(\mathrm{MD}\log\mathrm{MD})$; 而高斯密度计算可利用 SVM 算法中的 Hessian 矩阵, 计算复杂度为 $O(|U|\times L\times \mathrm{MD})$, 由于 $L \ll \mathrm{MD}$, $|U| \ll \mathrm{MD}$, 其渐近复杂度为 $O(\mathrm{MD})$; 最后根据采样的样本训练 SVM 算法的复杂度为 $O(\mathrm{MI}^2)$, 因此算法总的复杂度为 $O(n^2+\mathrm{MD}\log\mathrm{MD}+\mathrm{MD}+\mathrm{MI}^2)$. 对于不均衡数据而言, $\mathrm{MI} \ll \mathrm{MD}$, 渐近复杂度为 $O(n^2+\mathrm{MD}(\log\mathrm{MD}+1))$.

3　实验分析及对比

3.1　不均衡数据分类性能评估指标

以往适用于均衡数据分类的以整体分类错误为目标的传统性能评估指标, 已不再适用于不均衡数据集分类. 传统性能评估方法都从整体分类器考虑, 以此为指导训练学习得到的不均衡数据分类器存在容易将少数类样本错分的问题. 针对传统性能指标存在的缺陷, 近年来很多学者提出了一些用于不均衡数据集分类的性能评测指标, 最常见的有以下几种.

首先定义在不均衡数据集中少数类 (正类) 为 P, 多数类 (负类) 为 N. FP 是指将多数类样本错分成少数类的数目, FN 是指将少数类样本错分成多数类的数目. 同理, TP 和 TN 分别表示少数类和多数类样本被正确分类的个数. 由此可以得到:

少数类样本正确率 (TPR)

$$\mathrm{Sensitivity} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}),$$

$$\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP}+\mathrm{TN}); \tag{12}$$

多数类样本正确率

$$\mathrm{Specificity} = \mathrm{TN}/(\mathrm{FP}+\mathrm{TN}); \tag{13}$$

少数类查准率

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{FP}+\mathrm{TP}); \tag{14}$$

几何平均正确率 G-mean

$$G = \sqrt{\mathrm{Sensitivity}\cdot\mathrm{Specificity}}; \tag{15}$$

少数类的 F-measure

$$F = \frac{2\times\mathrm{Sensitivity}\times\mathrm{Precision}}{\mathrm{Sensitivity}+\mathrm{Precision}}. \tag{16}$$

表 1　混合矩阵

	预测正类	预测负类
真正正类	TP	FN
真正负类	FP	TN

性能指标 G 综合考虑了少数类和多数类两类样本的分类性能, 如果分类器分类偏向于其中一类则会影响另一类的分类正确率, 从而 G 值会很小. 性能指标 F 则考虑将少数类样本的查全率与查准率相结合, 其中任何一个值都能影响 F 值的大小, 所以它能综合地体现出分类器对少数类的分类效果.

3.2　实验数据

本文选用来源于国际机器学习标准数据库 UCI 中的 6 组不同的数据集对算法进行实验, 数据特征信息见表 2, 其中类别表示被选出作为少数类和多数类样本的代表类别.

表 2　实验数据集描述

数据集	属性	少数类/多数类	类别
haberman	4	126/225	2:1
german	25	300/700	B/A
pima	9	268/500	1:0
wpbc	35	46/148	R:N
abalone	8	634/689	10:9
yeast	9	429/463	NUC:CYT

3.3　不同算法的分类性能比较

为比较本文算法在不均衡数据下的分类性能, 应用本文算法 (SPU-SVM) 对上述数据集进行分类, 并与基于随机欠取样的 SVM 算法 (RU)、基于 SMOTE 过取样的 SVM 算法、基于 BSMOTE 过取样的 SVM 算法、基于代价敏感的 SVM 算法 (SVM-WEIGHT) 以及基于随机欠取样与 SMOTE 相结合的 SVM 算法 (RU-SMOTE-SVM)、自适应人工样本上采样 SVM 算法 (AdaSyn) 的结果进行比较. 对于每一个数据集, 采用 10 次交叉验证的方法进行实验, 对每次交叉实验运行 10 次以防止随机影响, 最后计算这些实验的 F-measure、G-mean 性能评测指标的统计平均值. 为了考察不均衡数据下算法的分类性能, 实验中选择 1:10 的比例进行随机选择, 并且每种算法的参数设置都选择各自的最优设置, 具体设置如下.

分类器SVM参数设置为: 核函数为高斯函数, 核宽度数为10, 惩罚因子$C = 1\,000$, Smote、BSmote算法中最近邻算法参数k选择为6, 其他欠取样算法保留着与少数类样本数目相同的多数类样本. 代价敏感SVM算法的多数类的代价与少数类的代价比值设置为$C_{\rm MI}/C_{\rm MA} = 10$, 本文算法中$L = 5\times{\rm MI}, u = 7, \lambda = 2$. 为了能与SVM算法很好地融合, 本文算法中$\delta$的值设置为与SVM算法核宽度相同.

对不同数据集的F-measure和G-mean性能指标的实验对比结果如表3所示.

表 3 10: 1不均衡数据下数据集F-measure、G-mean和AUC性能比较

Dataset	Methods	Specificity	Sensitivity	G-mean	F-measure
haberman	SVM	$\mathbf{1.0 \pm 0}$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SPU	0.823 ± 0.106	0.460 ± 0.073	$\mathbf{0.612} \pm 0.061$	0.599 ± 0.069
	RU	0.961 ± 0.063	0.128 ± 0.106	0.281 ± 0.211	0.207 ± 0.169
	Smote	0.769 ± 0.161	0.469 ± 0.082	0.588 ± 0.057	0.597 ± 0.061
	BSmote	0.769 ± 0.171	$\mathbf{0.469 \pm 0}.092$	0.590 ± 0.062	0.597 ± 0.071
	Weight	0.777 ± 0.158	0.460 ± 0.109	0.586 ± 0.053	0.589 ± 0.082
	RUS	0.645 ± 0.142	0.562 ± 0.052	0.599 ± 0.083	$\mathbf{0.661 \pm 0.051}$
	AdaSyn	0.750 ± 0.138	0.491 ± 0.093	0.599 ± 0.053	0.614 ± 0.071
german	SVM	$\mathbf{0.998 \pm 0.004}$	0.035 ± 0.005	0.039 ± 0.046	0.007 ± 0.011
	SPU	0.776 ± 0.98	$\mathbf{0.604 \pm 0.092}$	$\mathbf{0.679 \pm 0.026}$	$\mathbf{0.719 \pm 0.024}$
	RU	0.856 ± 0.037	0.482 ± 0.053	0.641 ± 0.034	0.630 ± 0.048
	Smote	0.824 ± 0.029	0.481 ± 0.032	0.629 ± 0.024	0.626 ± 0.029
	BSmote	0.831 ± 0.036	0.472 ± 0.031	0.626 ± 0.025	0.619 ± 0.028
	Weight	0.794 ± 0.058	0.555 ± 0.054	0.662 ± 0.030	$0.568\,5 \pm 0.042$
	RUS	$0.807 \pm 0.037\,8$	0.384 ± 0.045	0.555 ± 0.031	0.531 ± 0.046
	AdaSyn	0.811 ± 0.029	0.486 ± 0.032	0.627 ± 0.025	0.629 ± 0.029
pima	SVM	$\mathbf{1.0 \pm 0.0}$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SPU	0.718 ± 0.134	$\mathbf{0.767 \pm 0.114}$	$\mathbf{0.733 \pm 0.043}$	$\mathbf{0.833 \pm 0.068}$
	RU	0.898 ± 0.075	0.521 ± 0.046	0.682 ± 0.038	0.673 ± 0.038
	Smote	0.768 ± 0.073	0.687 ± 0.045	0.725 ± 0.034	0.789 ± 0.029
	BSmote	0.772 ± 0.076	0.685 ± 0.041	0.726 ± 0.035	0.788 ± 0.027
	Weight	0.778 ± 0.064	0.697 ± 0.033	0.736 ± 0.036	0.797 ± 0.024
	RUS	0.778 ± 0.058	0.611 ± 0.064	0.687 ± 0.025	0.733 ± 0.45
	AdaSyn	0.76 ± 0.052	0.704 ± 0.039	0.730 ± 0.030	0.800 ± 0.027
wpbc	SVM	$\mathbf{1.0 \pm 0.0}$	0.019 ± 0.027	0.086 ± 0.114	0.037 ± 0.051
	SPU	0.717 ± 0.119	0.532 ± 0.117	$\mathbf{0.612 \pm 0.076}$	$\mathbf{0.633 \pm 0.091}$
	RU	0.824 ± 0.058	0.371 ± 0.138	0.543 ± 0.109	0.498 ± 0.144
	Smote	0.811 ± 0.080	$\mathbf{0.426 \pm 0.072}$	0.584 ± 0.044	0.559 ± 0.063
	BSmote	0.825 ± 0.077	0.422 ± 0.089	0.586 ± 0.057	0.556 ± 0.081
	Weight	0.831 ± 0.079	0.439 ± 0.076	0.600 ± 0.049	0.574 ± 0.068
	RUS	0.763 ± 0.073	0.451 ± 0.095	0.585 ± 0.072	0.573 ± 0.084
	AdaSyn	0.804 ± 0.073	0.435 ± 0.079	0.589 ± 0.055	0.566 ± 0.072
abalone	SVM	$\mathbf{1.0 \pm 0.0}$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SPU	0.658 ± 0.047	0.606 ± 0.105	$\mathbf{0.630 \pm 0.189}$	0.740 ± 0.165
	RU	0.966 ± 0.047	0.112 ± 0.105	0.268 ± 0.189	0.186 ± 0.165
	Smote	0.630 ± 0.112	$\mathbf{0.628 \pm 0.082}$	0.623 ± 0.035	0.753 ± 0.059
	BS=mote	0.635 ± 0.118	0.629 ± 0.077	0.626 ± 0.046	0.754 ± 0.058
	Weight	0.639 ± 0.110	0.620 ± 0.089	0.623 ± 0.029	0.747 ± 0.064
	RUS	0.616 ± 0.063	0.580 ± 0.077	0.595 ± 0.041	0.716 ± 0.060
	AdaSyn	0.593 ± 0.089	0.657 ± 0.074	0.620 ± 0.037	$\mathbf{0.773 \pm 0.055}$
yeast	SVM	$\mathbf{1.0 \pm 0.0}$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	SPU	0.749 ± 0.093	0.520 ± 0.089	$\mathbf{0.618 \pm 0.033}$	0.667 ± 0.075
	RU	0.963 ± 0.042	0.121 ± 0.088	0.297 ± 0.170	0.204 ± 0.144
	Smote	0.719 ± 0.095	$\mathbf{0.546 \pm .065}$	0.623 ± 0.033	0.689 ± 0.052
	BSMote	0.715 ± 0.094	0.547 ± 0.064	0.622 ± 0.029	$\mathbf{0.690 \pm 0.050}$
	Weight	0.750 ± 0.102	0.504 ± 0.057	0.611 ± 0.028	0.656 ± 0.046
	RUS	0.671 ± 0.082	0.520 ± 0.033	0.590 ± 0.037	0.667 ± 0.028
	AdaSyn	0.704 ± 0.095	0.560 ± 0.064	0.625 ± 0.036	0.700 ± 0.052

从表3可以看出, SVM算法针对不均衡数据集分类而言, 出现了严重向少数类样本方向偏移的问题, 其中针对大部分数据集, 其SVM算法的Specificity性能指标多为1, Sensitivity性能指标基本为零, 而其他不均衡数据分类算法在二者指标上都有明显的提高. 其中本文算法的G-mean性能在各个数据集分类

上都优于其他不均衡数据SVM分类算法. 由于G-mean性能既考虑了多数类的样本分类性能, 也考虑了少数类样本的分类性能, 可以说本文算法在整体性能上最优. 观察另一个F-measure性能评测指标可以发现, 本文算法和SVM-Weight算法在该性能指标上表现较好, 而同样是欠取样算法的随机欠取样算法RU, 由于对多数类采样的盲目性使得该算法对不均衡数据分类性能的改善不如本文算法显著.

为进一步说明本文算法与其他算法的比较结果, 本文进行了T-检验, 置信区间为0.05, 并显示了win-tie-loss的比较结果(见表4和表5). 由表4和表5可以看出, 本文算法与其他算法相比都具有明显的优势.

表 4　10: 1 不均衡数据下 F-measure 的性能 T-检验比较

Dataset	RU	Smote	BSmote	Weight	RUS	AdaSyn	avg.
haberman	win	tie	tie	tie	loss	win	2-3-1
german	win	win	win	win	win	win	6-0-0
pima	win	tie	tie	tie	win	tie	2-4-0
wpbc	win	tie	tie	tie	tie	tie	1-5-0
abalone	win	tie	tie	tie	tie	win	2-4-0
yeast	win	tie	tie	tie	tie	loss	1-4-1
avg.	6-0-0	1-5-0	1-5-0	1-5-0	2-3-1	3-2-1	

表 5　10: 1 不均衡数据下 G-mean 的性能 T-检验比较

Dataset	RU	Smote	BSmote	Weight	RUS	AdaSyn	avg.
haberman	tie	tie	tie	tie	tie	tie	0-6-0
german	tie	win	win	tie	win	win	4-2-0
pima	tie	win	win	tie	win	tie	3-3-0
wpbc	tie	tie	tie	tie	tie	tie	0-6-0
abalone	tie	tie	tie	tie	win	tie	1-5-0
yeast	tie	tie	tie	tie	win	win	2-4-0
avg.	0-6-0	2-4-0	2-4-0	0-6-0	4-2-0	2-4-0	

3.4　不同比例下不均衡数据分类性能比较

为考察算法的鲁棒性, 采用文献[18]中的鲁棒性分析方法对8种算法在求解以上6个问题时的鲁棒性进行比较. 具体地, 算法m在某一特定数据集上的相对性能用该算法在求解该问题时得到的Adjusted Rand Index的值与最大Adjusted Rand Index值的比值进行衡量, 即

$$b_m = \frac{R_m}{\max\limits_k R_m}. \tag{17}$$

因此, 在某个数据集上表现最好的算法m^*的相对性能$b_{m^*} = 1$, 而其他算法的相对性能$b_m \leqslant 1, b_m$值越大, 算法m在所有算法中的相对性能越好. 算法m在所有数据集上的b_m值的总和可以用来客观评价算法的鲁棒性, 总和越大鲁棒性越好. 为了验证本文提出的基于样本特性欠取样分类算法的鲁棒性能, 同样选择上面数据库中数据集作为测试数据, 将多数类数据与少数类数据数目的比例按20: 1的比例进行选取, 然后利用10次交叉验证法进行测试, 将测试结果与其他算法进行比较, 其中算法参数设置同上. 图4为8种算法的G-mean鲁棒性比较结果, 每一个算法对应的柱状图高度为对应算法在所有6个问题上的b_m值的总和.

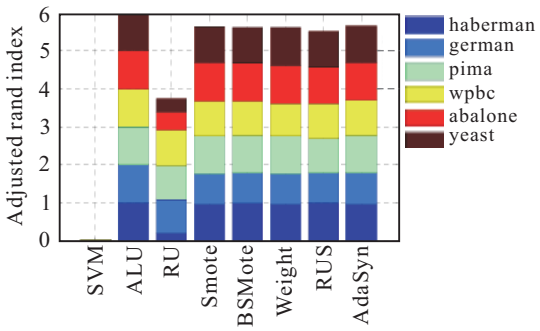

图 4　20: 1 不均衡数据下 G-mean 的 Adjusted Rand Index 鲁棒性能比较

从图4可以看出, 本文算法获得了最高的总和值, 达到了6个, 这充分说明了基于样本特性欠取样的不均衡SVM算法具有很好的鲁棒性. 本文算法的b_m值对测试的6个问题均为1, 表明SPU-SVM对不同空间结构以及不同维度的数据不均衡分类问题均表现出很好的性能, 其在所有比较的8种算法中具有最好的鲁棒性. 这是由于本文算法充分结合了SVM算法的特点, 利用基于样本特性欠取样选择多数类样本点, 使得训练得到的SVM算法分类界面向着多数类样本方向进行适当地偏移所带来的结果.

3.5　参数(高斯核半径)对算法性能的影响

为了测试高斯核半径参数对本文算法分类性能的影响, 选用haberman、german数据集作为测试数据, 多数类样本数目和少数类样本数目按20: 1的比例选取, 用10次交叉验证法测试, γ参数选定在$(0.5, 4.5)$区间, 其他参数设置同上, 其中本文的相似度计算以及SVM算法的计算均在同一个参数的特征空间进行. 测试结果如图5所示.

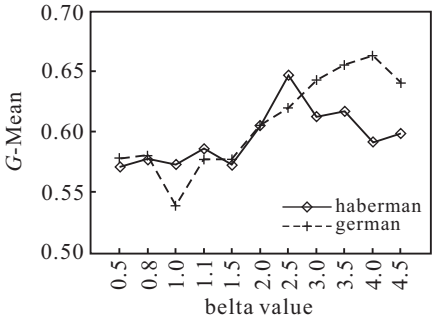

图 5　G-mean 性能随高斯核半径值的变化情况

从图5可以得出: 本文算法随着参数的增大, 算法性能呈明显上升趋势, 这表明在该段高斯核半径范围内, 高斯核半经参数越大分类性能越好. 这是因为本文基于样本特性欠取样算法选取的多数类样本点都具有一定的空间代表性, 因此为了能在算法中发挥

代表作用, 需将自身的邻域半径扩大以便使其受影响的面积增多. 然而, 随着参数的进一步增大, 分类性能开始出现下降趋势, 这是因为随着参数的增大, 每个样本的影响区域不断增大而导致学习能力降低. 因此选择一个合适的参数对于本文算法的性能提升具有一定的帮助, 从实验结果可以看出参数在 $[2, 4]$ 区间性能最优.

4 结 论

本文针对SVM算法在不均衡数据下分类性能差的问题, 提出了一种基于样本特性欠取样的不均衡SVM分类算法. 该算法通过在SVM特征空间中选择具有信息性并同时具有一定代表性的多数类样本点, 实现了SVM分类界面向着多数类样本方向偏移的目的. 将本文算法与其他不均衡数据分类算法进行了实验比较, 比较结果表明, 本文算法在不同数据集下的分类性能优于其他算法, 并具有较强的鲁棒性. 最后, 为了考察高斯核半径参数对算法性能的影响, 本文利用不同参数值对不同数据集进行实验, 由实验结果发现, 本文算法在将参数设置在一定范围时分类性能较好, 这一现象也同样符合欠取样算法的机理.

参考文献(References)

[1] Vapnik V N. The nature of statistical learning theory[M]. New York: Springer, 2000: 138-167.

[2] He H B, Edwardo A. Learning from imbalanced data[J]. IEEE Trans on Knowledge and Data Engineering, 2009, 21(8): 1263-1284.

[3] Liu X Y, Zhou Z H. Exploratory under-sampling for class-imbalance learing[J]. IEEE Trans on Systems, Man and Cybernetics, 2009, 39(2): 539-550.

[4] Liu X Y, Zhou Z H. Training cost-sensitive neural networks with methods addressing the class imbalance problem[J]. IEEE Trans on Knowledage and Data Engineering, 2006, 18(1): 63-77.

[5] Van H J, Khoshgoftaar T M, Napolitano A. Experimental perspectives on learning from imbalanceed data[C]. Proc of the 24th Int Conf on Machine Learning. New York: ACM, 2007: 143-146.

[6] Weiss G M. Mining with rarity: A unifying framework[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 7-19.

[7] Estabrooks A, Jo T. A multiple resampling method for learning from imbalanced data sets[J]. Computational Intelligence, 2004, 20(11): 18-36.

[8] Han H, Wang W Y, Mao B H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning[C]. Proc Int Conf of Intelligent Computing. Berlin Heidelberg: Springer, 2005: 878-887.

[9] Akban I R, Kwek S, Japkow I. Applying support vector machines to imbalanced datasets[C]. Proc of the 15th European Conf on Machines Learning. Berlin Heidelberg: Springer, 2004: 39-50.

[10] Bastista G E, Prati R C, Monard M C. A study of the Behavior of several methods for balancing machine learning training data[J]. ACM SIGKDD Exploration Newsletter, 2004, 6(1): 20-29.

[11] 陶新民, 徐晶, 童稚靖. 不均衡数据下基于阴性免疫的过抽样算法[J]. 控制与决策, 2010, 25(6): 867-873.
(Tao X M, Xu J, Tong Z J. Over-sampling algorithm based on negative immune in imbalanced data sets learning[J]. Control and Decision, 2010, 25(6): 867-873.)

[12] Sun Y, Kamel M S, Wong A K C. Cost-sensitive boosting for classification of imbalanced data[J]. Pattern Recognition, 2007, 40(11): 3358-3378.

[13] 曾志强, 吴群, 廖备水, 等. 一种基于核SMOTE的非平衡数据集分类方法[J]. 电子学报, 2009, 39(10): 2489-2495.
(Zeng Z Q, Wu Q, Liao B S, et al. A classfication method for imbalance data set based on kernel SMOTE[J]. Acta Electronica Sinica, 2009, 39(10): 2489-2495.)

[14] He H, Bai Y. ADASYN: Adaptive synthetic sampling approach for imbalanced learning[C]. Proc Int Conf of Neural Networks. Hong Kong, 2008: 1322-1328.

[15] 毕华, 梁洪力. 重采样方法与机器学习[J]. 计算机学报, 2009, 32(5) :862-877.
(Bi H, Liang H L. Resampling methods and machine learning[J]. Chinese J of Computers, 2009, 32(5): 862-877.)

[16] Liu Y, Yu X H. Combining integrated sampling with SVM ensembles for learning from imbalanced datasets[J]. Information Processing & Management, 2011, 47(4): 617-631.

[17] 陶新民, 童智靖, 刘玉. 基于ODR和BSMOTE结合的不均衡数据SVM分类算法[J]. 控制与决策, 2011, 26(10): 1535-1541.
(Tao X M, Tong Z J, Liu Y. SVM classifier for unbalanced data based on combination of ODR and BSMOTE[J]. Control and Decision, 2011, 26(10): 1535-1541.)

[18] 公茂果, 焦李成, 马文萍. 基于流行距离的人工免疫无监督分类与识别算法[J]. 自动化学报, 2008, 34(3): 367-375.
(Gong M G, Jiao L C, Ma W P. Unsupervised classification and recognition using an artificial immune system based on manifold distance[J]. Acta Automatica Sinica, 2008, 34(3): 367-375.)