文章编号: 1001-0920 (2012) 12-1761-08

基于谱聚类欠取样的不均衡数据 SVM 分类算法

陶新民, 张冬雪, 郝思媛, 付丹丹

(哈尔滨工程大学 信息与通信工程学院，哈尔滨 150001)

摘 要: 提出一种基于谱聚类欠取样的不均衡数据支持向量机 (SVM) 分类算法. 该算法首先在核空间中对多数类样本进行谱聚类; 然后在每个聚类中根据聚类大小和该聚类与少数类样本间的距离, 选择具有代表意义的信息点; 最终实现训练样本间的数目均衡. 实验中将该算法同其他不均衡数据预处理方法相比较, 结果表明该算法不仅能有效提高 SVM 算法对少数类的分类性能, 而且总体分类性能及运行效率都有明显提高.

关键词: 不均衡数据；SVM 算法；谱聚类；欠取样

中图分类号: TP391 **文献标志码:** A

SVM classifier for unbalanced data based on spectrum cluster-based under-sampling approaches

TAO Xin-min, *ZHANG Dong-xue*, *HAO Si-yuan*, *FU Dan-dan*

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

Abstract: An under-sampling unbalanced dataset support vector machine(SVM) algorithm based on spectrum cluster is presented. Majority instances are clustered by using spectrum cluster in kernel space for resampling reprentative samples with cluster information. The number of selected samples in each cluster is dependent on the size of each cluster and the distance of the cluster to the all minority instances, which can not only reduce the number of majority instances, but also the SVM classification performance under unbalanced dataset is improved by using the proposed method. In the experiments, the proposed approach is compared with other data-preprocess methods for unbalanced dataset classification. The experimental results show that the proposed method can not only improve classification performance of SVM algorithm in the minority class data, but also increase the overall classification performance and effectivity.

Key words: unbalanced data；support vector machine algorithm；spectrum cluster；under-sampling

1 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, 具备较强的泛化能力, 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, 需要数目相同的不同类别样本进行训练才能获得较好的泛化能力. 但在现实生活中, 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. 针对不均衡数据集进行分类时, 由于各个类别的样本数目存在较大差异, 导致不同类别的样本对于训练算法提供的信息不对称, 这使得 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], 从而使支持向量机过度拟合多数类样本点, 低估了少数类样本点, 导致 SVM 算法对少数类样本的错分率增大. 因此, 如何实现 SVM 算法在不均衡数据下的正确分类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

收稿日期: 2011-07-21；**修回日期:** 2011-10-08.

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；张冬雪(1987−), 女, 硕士生, 从事模式识别、信号处理的研究.

据过取样和欠取样. 在与过取样结合的方法中, 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 oversampling technique)代价敏感SVM算法以及基于边界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], 以及借鉴实例简约的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, 而这些选取出来的子集对改善SVM分类界面是否有效却未知, 如选择不当则可能会导致分类效果不理想[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, 是利用欠取样提高不均衡数据下SVM算法分类性能的关键.

鉴于此, 本文提出一种基于谱聚类[17]的欠取样算法. 首先在核空间中对多数类样本进行聚类; 然后根据聚类结果选择具有代表性的多数类样本子集来实现训练数据样本均衡; 同时通过尽量多地选取离少数类距离远的聚类中样本的方式来实现SVM分类界面向多数类样本的偏移, 以此提高SVM算法在不均衡数据下的分类性能. 实验部分将建议的基于谱聚类欠取样的SVM算法同其他取样与SVM相结合的算法进行比较, 结果表明建议的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.

2 SVM算法及其不均衡数据分类问题分析

2.1 支持向量机简介

SVM是建立在统计学习理论中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(即对特定训练样本的学习精度)和学习能力(即无错误地识别样本的能力)之间寻求最佳折衷, 以期获得最好的推广能力. 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中, 以便分类界面在高维特征空间中变得线性可分, 使分类效果更好.

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

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

$$\text{s.t. } y_i(w^{\mathrm{T}}x_i+b)\geqslant 1-\xi_i;$$

$$\xi_i\geqslant 0,\ i=1,2,\cdots,n. \tag{2}$$

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

$$L(w,b,\alpha)=\frac{1}{2}||w||^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\beta_i\xi_i-$$

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

其中 α_i 和 β_i 为拉格朗日算子. 根据KKT条件, 有

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

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

$\alpha_i>0$ 的样本是支持向量. 判别函数为

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

2.2 SVM在不均衡数据下分类边界的偏移

传统的SVM类算法都是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中多数时候并不成立. 实际上, 在大多数的应用领域中很多类别并不均衡, 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, 从而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, 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算法的参数设置如下: 选择高斯核函数, 核宽度为10, 惩罚常数选择 $C=10$, 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.

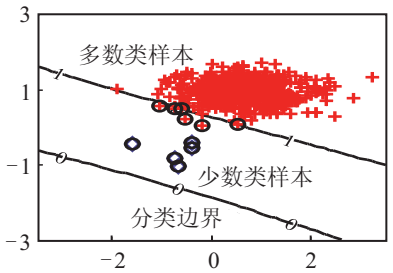

图1 数据样本比例为100:1时SVM算法的分类边界

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, 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将会向样本数量小的类别移动. 这样一来, 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 分类器的分类性能, 必须解决 SVM 算法在不均衡数据下分类边界偏向于少数类样本的问题.

3　基于谱聚类欠取样的不均衡数据 SVM 算法

3.1　传统欠取样算法分析

在多数类样本中存在大量的重复信息, 这些冗余信息会导致多数类与少数类样本的数目不均衡, 从而严重影响 SVM 分类器的界面生成. 因此, 传统的欠取样算法通过剔除这些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间的均衡. 这种方法包括: DROP、CNN 和 ODR[10]等算法. 然而, 这些减少多数类样本数目的欠取样算法并不适合于 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数目, 即使实现了多数类和少数类样本数目间的均衡, 也不能改变 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.

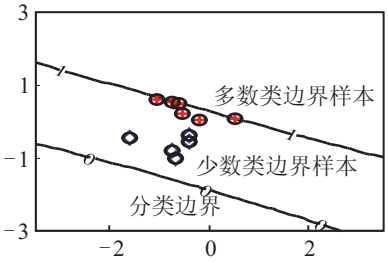

图 2　保留边界样本后 SVM 算法的分类边界的变化

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的变化, 这是由于支持向量都存在于边界附近, 而保留边界样本无疑会导致原有的未进行数据均衡前的支持向量仍然存在, 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, 这个示例说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.

3.2　基于谱聚类的欠取样

在多数类样本中存在着噪声样本和大量的重复信息, 这些冗余信息将会导致 SVM 分类器的界面向着少数类一方偏移. 因此, 为了能使分类界面向着多数类方向偏移, 就需要剔除这些冗余样本. 然而, 传统的欠取样算法只是将远离边界的样本进行了删除, 或者随机选取多数类的一个子集, 没能考虑采样后子集的信息是否有效, 这种方法虽然能实现训练样本数目间的均衡, 但对改善 SVM 的分类界面没有任何影响. 为了能有效改善 SVM 分类界面的位置, 使其向着多数类方向偏移, 需要删除部分边界样本. 为了不改变多数类样本集合空间结构, 本文利用聚类算法对多数类样本聚类, 然后选择那些聚类中具有局部空间代表意义的样本作为新的训练样本, 即可实现对多数类样本集合有目的地筛选. 在聚类算法的选择上, 由于传统的聚类算法 (如 KMEANS 算法、EM 算法等) 都是建立在凸球形的样本空间上, 当样本空间不为凸时算法会陷入局部最优. 为了能实现在任意形状的样本空间上聚类并收敛于全局最优解, 学者们开始研究一类新型的聚类算法, 称为谱聚类算法. 该算法首先根据给定的样本数据集定义一个描述成对数据点相似度的亲合矩阵; 然后计算矩阵的特征值和特征向量; 最后选择合适的特征向量聚类不同的数据点. 谱聚类算法最初用于计算机视觉等领域, 最近才开始用于机器学习中并迅速成为机器学习领域的研究热点. 这种算法不用对数据的全局结构作假设, 并且具有识别非凸分布聚类的能力, 因此非常适合于许多实际问题. 另外, 谱聚类算法能在核空间聚类, 这使其能与 SVM 算法实现无缝连接. 下面简单介绍一下谱聚类算法.

定义一个无方向图 $G=(V,E)$, 其中定点集合 $V=\{v_1,v_2,\cdots,v_n\}$, n 是样本个数. 假设 G 是个加权图, 它的两个定点 v_i 和 v_j 的边由 $w_{ij}>0$ 表示. 进一步, 定义该图的加权连接矩阵为 $W=(w_{ij})_{i,j=1,2,\cdots,n}$. 如果 $w_{ij}=0$, 则表示 v_i 和 v_j 顶点间没有边. 因 G 是个无向图, 所以 $w_{ij}=w_{ji}$. 其中顶点 $v_i\subset V$ 的度及图的度矩阵分别定义为

$$d_i=\sum_{j=1}^{n}w_{ij},\ D=\mathrm{diag}(d_1,d_2,\cdots,d_n). \tag{7}$$

对应该图的标准化图拉普拉斯矩阵定义为

$$L_{\mathrm{sym}}=I-D^{-1/2}WD^{-1/2}. \tag{8}$$

谱聚类具体算法描述如下:

输入: 相似矩阵 $S\in R^{n\times n}$ 和聚类个数 k;

输出: 聚类 $A_1,\cdots,A_k$, $A_i=\{j|t_j\in C_j\}$.

1) 根据相似矩阵构造无方向相似图 $G=(V,E)$, 其中 W 作为它的加权连接矩阵;

2) 计算标准化图拉普拉斯矩阵 L_{sym};

3) 计算拉普拉斯矩阵 L_{sym} 的前 k 个特征矢量 $\mu_{.1},\mu_{.2},\cdots,\mu_{.k}$;

4) 将 $\mu_{.1},\mu_{.2},\cdots,\mu_{.k}$ 作为列生成 $U\in R^{n\times k}$;

5) 通过对 U 矩阵中的每一行进行标准化处理, $t_{ij}=\mu_{ij}\Big/\Big(\sum\limits_{k}\mu_{ij}^2\Big)^{1/2}$ 生成 $T\in R^{n\times k}$;

6) 对于每一行 $i=1,2,\cdots,n, t_i \in R^k$ 作为矩阵 T 的第 i 行的向量;

7) 对 $(t_i)_{i=1,2,\cdots,n}$ 利用 K-means 算法进行聚类, 形成 $C_1, C_2, \cdots, C_k$.

3.3 基于谱聚类欠取样不均衡数据 SVM 算法

利用上述基于谱聚类的欠取样算法对多数类样本进行预处理, 然后将取样后的多数类样本子集同全部少数类样本共同组合成新的训练样本集, 输入到 SVM 算法中进行训练学习, 具体步骤如下.

1) 设置预取样的多数类样本点个数 Major$N$ = m * Minor N, m 为两者数量上的比例.

2) 算法首先利用多数类样本集合建立一个基于高斯核的相似矩阵 $S \in R^{n\times n}$, n 是原有多数类样本的个数.

3) 利用上述的谱聚类算法对多数类样本进行聚类分析, 生成聚类 $A_1, \cdots, A_k$, $k=$ Major$N$.

4) 选择每一个聚类中具有代表性的样本点, 其中在每一个聚类中的样本选择数取决于该聚类的大小以及该聚类中的样本与少数类样本点的平均距离的大小, 聚类越大, 选择的样本数越多; 离少数类样本越近则选择的越少. 如此选择是为了有目的地删除多数类中的边界样本信息点, 具体如下:

$$K\mathrm{Dist}_{il} = K(x_i,x_i) + K(x_l,x_l) - 2K(x_i,x_l), \tag{9}$$

$$I\mathrm{Dist}_i = \frac{1}{K\mathrm{size}_i \times \mathrm{Minor}N} \sum_{j=1}^{K\mathrm{size}_i} \sum_{l=1}^{\mathrm{Minor}N} K\mathrm{Dist}_{il}, \tag{10}$$

$$\mathrm{Radio}_i = \left(K\mathrm{size}_i \Big/ \sum_{i}^{k} K\mathrm{size}_i\right) \times \left(I\mathrm{Dist}_i \Big/ \sum_{i=1}^{k} I\mathrm{Dist}_i\right), \tag{11}$$

$$S\mathrm{size}_{MA}^{i} = \mathrm{Major}N \cdot \mathrm{Radio}_i \Big/ \sum_{i=1}^{k} \mathrm{Radio}_i. \tag{12}$$

其中: $K\mathrm{size}_i$ 为聚类 A_i 的大小, $I\mathrm{Dist}_i$ 为聚类 A_i 到少数类的平均距离, $S\mathrm{size}_{MA}^{i}$ 是每一个聚类中选择的样本个数.

5) 对于每一个聚类, 选择前 $S\mathrm{size}_{MA}^{i}$ 个离少数类样本点平均距离最小的样本组合后形成新的多数类训练样本子集.

6) 将取样得到的多数类训练样本子集和全部的少数类样本组合作为新的训练样本, 输入到 SVM 算法中进行训练学习, 其中核参数与相似矩阵 S 的核参数相同, 即谱聚类和 SVM 分类算法都在一个空间中进行.

7) 根据训练得到的分类界面进行新样本的类别辨识.

图 3 为当 $m=5$ 时 SVM 算法分类界面的变化情况. 不难看出, 通过谱聚类欠取样算法处理后 SVM 的分类性能得到了很好改善, 分类边界已经向着多数类方向偏移. 其中正方形点为多数类到少数类的最远点, 六角形为多数类到少数类的中心点, 倒三角形为多数类到少数类的最近点.

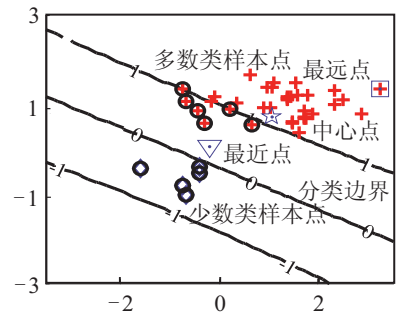

图 3 经过谱聚类欠取样后 SVM 算法分类边界的变化

4 实验分析及对比

4.1 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, 已不再适合不均衡数据集分类. 由于传统的性能评估是从整体分类器考虑的, 以此为指导训练学习得到的不均衡数据分类器容易将少数类样本错分. 这是因为少数类样本数目所占比例不大, 将其分错对总体的分类性能指标的影响不大. 针对传统性能指标存在的缺陷, 近年来很多学者提出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):

少数类样本查全率

$$\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}); \tag{13}$$

多数类样本查全率

$$\mathrm{TNR} = \mathrm{TN}/(\mathrm{TN}+\mathrm{FP}); \tag{14}$$

少数类样本查准率

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}); \tag{15}$$

多数类样本错分率

$$\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP}+\mathrm{TN}); \tag{16}$$

几何平均正确率 G-mean

$$G = \sqrt{\mathrm{TPR} * \mathrm{TNR}}; \tag{17}$$

少数类的 F-measure

$$F = \frac{2 * \mathrm{TPR} * \mathrm{Precision}}{\mathrm{TPR} + \mathrm{Precision}}. \tag{18}$$

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 则是考虑少数类样本的查全率和查准率的结合, 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. AUC (area under the ROC curve) 是另一个有效的不均衡数据分类性能评价手段, 对于一个给定的两分类问题, ROC 曲线是利用多个 (FPR, TPR) 对描述性能的方法. AUC 是这个曲线形成的面积, 它评测的是 FPR 所有可能值对应的分类方法的性能, 因此被证明是一个非常有效的不均衡分类性能评测标准.

在实验中, 通过改变 SVM 算法的输出 $f(x)$ 的阈值从 $(-\infty, \infty)$ 得到 (FPR, TPR) 对, 然后利用文献 [3] 中的算法计算 AUC 值.

4.2　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 9 组不同的数据集对算法进行实验, 数据特征信息见表 1, 其中用类别表示选择出来作为少数类和多数类样本的代表类别.

表 1　实验数据集描述

数据集	属性	少数类/多数类	类别
wdbc	30	212/357	B/M
glass	10	17/76	3:2
letter	16	766/789	2(B):1(A)
car	6	384/1210	Acc:Unacc
abalone	8	634/689	10:9
haberman	4	126 /225	2:1
ionosphere	35	126/225	Bad:Good
pima	9	268/500	1:0
yeast	9	429/463	NUC: CYT

4.3　不同算法的分类性能比较

为了比较本文算法在不均衡数据下的分类性能, 试验中应用本文算法 (SC-SVM) 对上述数据集进行分类, 并与基于随机欠取样的 SVM 算法、基于 DROP 欠取样的 SVM 算法、基于 CNN 欠取样的 SVM 算法、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和 SMOTE 相结合的 SVM 算法 (RU-SMOTE-SVM) 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, 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 和 AUC 性能评测指标的统计平均值. 本次实验选取 1:30 的比例进行随机选择, 以考察不均衡数据下算法的分类性能. 其中分类器 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 和 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 算法的少数类的代价与多数类的代价比值设置为 $C_{MI}/C_{MA} = 30$, 本文中的谱聚类算法的相似矩阵为高斯矩阵, 核宽度数为 10, 为了实现连接矩阵的稀疏化, 本文选择 $K = 7$ 的近邻矩阵, 算法初始 m 值为 5. 实验结果如表 2 和表 3 所示, 表 2 为前 4 个易分类数据集 (根据本文算法 F-measure 值大于 90% 的数据集称为易分类数据集) 的实验对比结果, 表 3 为后 5 个难分类数据集的实验对比结果.

由表 2 和图 4 的结果可以得知, 对于文中的这 4 种易分类的数据而言, 本文算法的 F-measure 性能和 G-mean 性能都优于其他的不均衡数据 SVM 分类算法. 对于 wdbc 数据而言, RU-SVM 算法性能最优, 但其对 car 数据的分类性能却较差; 而对于 SMOTE-SVM 以及 BSOMTE-SVM 算法而言, 虽然在 wdbc 及 letter 数据集上表现出较好的分类性能, 但其在 glass 数据上分类性能却不强, 而本文算法的 F-measure 的性能对于 4 种数据集而言都表现出了较好的分类性能. 由于 F-measure 主要测量的是对少数类样本的分类精度, 该性能指标的提升也说明本文算法的确实现了将 SVM 分类边界向着多数类偏移的目的. 观察 G-mean 性能指标可以看出, DROP-SVM 算法和 CNN-SVM 算法在 4 种数据上的性能都较低, 是由于 SVM 分类界面已经超过真正分类界面的位置向着多数类方向偏移过大造成的, 可以说传统的实例简约算法并不适合处理 SVM 算法的不均衡数据分类问题. SMOTE-SVM 和 BSMOTE-SVM 算法在该项指标同其他 RU-SVM、WEIGHT-SVM 和 RUSMOTE-SVM 算法相比性能较稳定, 尤其在 wdbc 数据集上表现尤为突出. 而本文算法在 4 种数据上都表现出了最优的分类性能, 这说明本文算法利用谱聚类算法选择代表性样本的欠取样算法更适合于将 SVM 分类界面向着多数类方向进行偏移. 由于 G-mean 性能考虑了多数类和少数类的分类性能, 可以说本文算法的综合性能最优, 能够实现在没有损失多数类样本分类性能的前提下提高少数类样本的分类精度. 观察另一个 AUC 性能评测指标可以发现, SMOTE-SVM、BSMOTE-SVM 和 WEIGHT-SVM 算法在该性能指标上表现较好, 而本文算法虽然稍逊一筹但差别并不明显. 为了能综合比较这些算法的分类性能, 本文将这些算法进行叠加求和, 考察他们之间的性能对比, 结果如图 4 所示. 从结果中发现, 本文算法在 All-ACC 性能上表现最优, 其值为 2.887 1.

本文针对后 5 种较难分类的数据集进行了实验比较, 有关 F-measure、G-mean 和 AUC 的性能指标比较结果如表 3 所示. 从结果中同样发现, DROP-SVM 和 CNN-SVM 算法除了 ionosphere 外在其他 4 个数据

表 2 不同易分数据集的 F-measure、G-mean 和 AUC 性能比较

F-measure	wdbc	glass	letter	car	Avg
SC-SVM	0.959±0.043	**0.989 4±0.017 1**	0.981 2±0.017 8	0.920 1±0.016 7	**0.962 4±0.023 7**
RU	**0.988 7±0.0**	0.963 0±0	0.967 2±0	0.907 1±0	0.956 5±0
DROP	0.918 1±0.001	0.786 7±0.011 5	0.949 3±0.000 2	0.850 6±0.001 0	0.876 2±0.003 4
CNN	0.918 1±0.001 0	0.786 7±0.011 5	0.949 3±0.000 2	0.850 4±0.000 5	0.876 1±0.003 3
SMTE	0.967 1±0.023 2	0.907 7±0.061 6	0.982 3±0.016 8	0.945 1±0.024 6	0.950 6±0.031 6
BSMOTE	0.968 6±0.007 6	0.920 1±0.061 5	0.977 7±0.015 0	0.932 7±0.019 3	0.949 8±0.025 9
WEIGHT	0.002 0±0.004 2	0.963 6±0.062 2	**0.985 0±0.015 4**	**0.948 7±0.019 7**	0.724 8±0.025 4
RUS	0.951 5±0.008 6	0.966 9±0.069 6	0.986 9±0	0.926 2±0.019 1	0.957 9±0.024 3
G-mean	wdbc	glass	letter	car	Avg
SC-SVM	**0.892 5±0.063 1**	**0.983 4±0.027 7**	**0.987 6±0.017 7**	**0.922 9±0.019 4**	**0.946 6±0.032 0**
RU	0.873 1±0	0.963 6±0	0.967 7±0	0.885 2±0	0.922 4±0
DROP	0±0	0±0	0±0	0.033 9±0.057 6	0.008 5±0.014 4
CNN	0±0	0±0	0±0	0.027 3±0.043 9	0.006 8±0.011 0
SMOTE	0.901 7±0.023 2	0.909 8±0.053 8	0.982 0±0.015 8	0.921 4±0.027 8	0.928 7±0.030 2
BSMOTE	0.908 2±0.027 4	0.924 3±0.056 0	0.977 5±0.015 0	0.910 1±0.026 1	0.930 0±0.031 1
WEIGHT	0.014 1±0.029 8	0.965 7±0.058 4	0.984 7±0.014 7	0.919 9±0.022 8	0.721 1±0.031 4
RUS	0.768 8±0.043 9	0.969 1±0.063 8	0.987 1±0	0.916 0±0.021 2	0.910 3±0.032 2
AUC	wdbc	glass	letter	car	Avg
SC-SVM	0.943 6±0.061 2	0.997 3±0.008 5	0.999 8±0	0.971 5±0.009 4	0.978 1±0.019 8
RU	0.987 6±0	1.000 0±0	0.997 1±0	0.955 7±0	0.985 1±0
DROP	0.551 6±0.091 5	0.984 4±0.039 0	0.986 6±0.005 8	0.747 1±0.033 0	0.817 4±0.042 3
CNN	0.559 7±0.078 4	0.996 2±0.008 1	0.975 8±0.015 4	0.742 9±0.031 5	0.818 7±0.033 4
SMOTE	0.962 4±0.030 0	**1.000 0±0**	0.998 9±0.001 4	0.972 7±0.010 7	**0.983 5±0.010 5**
BSMOTE	0.961 5±0.022 8	1.000 0±0	0.999 0±0.001 0	0.968 9±0.014 2	0.982 4±0.009 5
WEIGHT	**0.962 6±0.026 7**	1.000 0±0	**0.999 8±0.000 3**	0.970 1±0.012 1	0.983 1±0.009 8
RUS	0.910 2±0.052 9	1.000 0±0	0.999 5±0	**0.980 0±0.009 6**	0.972 4±0.015 6

表 3 不同难分数据集的 F-measure、G-mean 和 AUC 性能比较

F-measure	abalone	haberman	ionosphere	pima	yeast	Avg
SC-SVM	0.764 8±0.035 5	0.631±0.057 3	**0.733 2±0.0639**	0.456 9±0.057	0.774 7±.053 9	**0.672 1±0.053 5**
RU	0.802 27±0.032	0.548 7±0.012	0.713 5±0	0.432 4±0	0.605 4±0	0.620 5±0.008 8
DROP	0.908 9±.030 0	**0.866 5**±0.002 7	0±0	**0.909 4±0.001**	**0.852 2±0**	0.707 4±0.006 7
CNN	**0.917 1±0.017 9**	0.866 5±0.002 7	0±0	0.909 4±0	0.758 4±0.397 2	0.690 3±0.083 6
SMOTE	0. 8±0.066 9	0.478 0±0.084 4	0.720 2±0.074 7	0.389±0.043 7	0.711 7±0.204 9	0.574 7±0.107 7
BSMOTE	0.8±0.072 0	0.505 3±0.084 2	0.709 7±0.086 1	0.369±0.077 6	0.683 3±0.212 0	0.613 6±0.106 4
WEIGHT	0.713 4±0.115 2	0.332 7±0.083 4	0.678 7±0.127 5	0.035±0.042 4	0.517 0±0.218 9	0.455 5±0.117 5
RUS	0.712 4±0.047 7	0.520 5±0.069 6	0.668 7±0.076 3	0.386 1±0.074 9	0.654 4±0.066 9	0.588 4±0.063 2
G-mean	abalone	haberman	ionosphere	pima	yeast	Avg
SC-SVM	**0.575 5±0.030 9**	0.533 3±0.045 3	**0.744 8±0.045 0**	**0.504 0±0.048 1**	**0.594 9±0.050 4**	**0.590 5±0.044 0**
RU	0.489 9±0	0.500 9±0	0.730 7±0	0.479 0±0	0.437 1±0	0.527 5±0
DROP	0.260 7±.114 8	0±0	0± 0	0±0	0±0	0.052 1±0.023 0
CNN	0.276 2±0.067 1	0±0	0 ±0	0±0	0±0	0.055 2±0.013 4
SMOTE	0. 5±0.057 9	0.512 2±0.063 6	0.737 9±0.054 9	0.469 3±0.030 0	0.487 8±0.147 1	0.551 8±0.070 7
BSMOTE	0. 5±0.056 5	**0.537±0.058 4**	0.738 4±0.066 2	0.452 1±0.061 2	0.411 4±0.223 9	0.534 9±0.093 2
WEIGHT	0.542 8±0.053 4	0.417±0.060 7	0.716±0.100 1	0.128 8±0.042 4	0.451 5±0.175 0	0.451 2±0.086 3
RUS	0.573 3±0.036 9	0.526 8±0.055 3	0.704 1±0.060 4	0.460 2±0.046 0	0.593 4±0.032 0	0.571 6±0.046 1
AUC	abalone	haberman	ionosphere	pima	yeast	Avg
SC-SVM	**0.624 9±0.035 7**	0.593±0.064 7	0.864 9±0	0.664 7±0.052 3	**0.655 4±0.053 3**	**0.680 7±0.041 2**
RU	0.531 7±0	0.471 4±0	**1±0**	0.591±0.117 0	0.406 6±0	0.600 2±0.023 4
DROP	0.505 8±0.044 7	0.490 7±0.111 7	1±0	0.515 4±0.063 1	0.621 8±0.140 0	0.626 7±0.071 9
CNN	0.492 4±0.035 6	0.554±0.089 0	1±0	0.527 9±0.084 0	0.525 4±0.172 6	0.620 1±0.076 2
SMOTE	0.6±0.036 3	0.567 2±0.072 2	0.852 0±0.033 8	**0.686 8±0.034 7**	0.651 2±0.029 8	0.671 4±0.041 4
BSMOTE	0.6±0.045 8	**0.602±0.069 4**	0.865 5±0.046 4	0.675±0.057 7	0.624 7±0.058 3	0.673 6±0.055 5
WEIGHT	0.604 5±0.038 8	0.558±0.072 6	0.876±0.037 3	0.646 0±0.035 9	0.651 2±0.050 3	0.667 3±0.047 0
RUS	0.616 6±0.045 1	0.601±0.083 3	0.794±0.047 4	0.669 6±0.076 4	0.642 1±0.028 6	0.664 8±0.056 2

集上的 F-measure 性能较高, 这表明其对少数类样本的分类性能较好, 但在 G-mean 性能指标上却表现的十分不好, 而 G-mean 是既考虑少数类又考虑多数类分类性能的综合评测指标. 因此该现象说明, 这些传统的实例简约欠取样算法是通过损失多数类分类性能的基础上提高少数类算法性能的. 本文算法无论

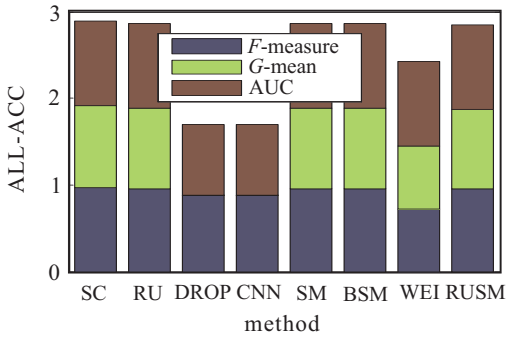

图 4　不同易分数据集的 All-ACC 的对比结果

在 F-measure、G-mean 和 AUC 的性能指标上都表现良好且较稳定, 尤其是 G-mean 性能在每个数据集上都达到了最优. 从图 5 可以看出, 本文算法 All-ACC 性能最好, 其值为 1.943 3.

为了比较各种算法的效率性能, 本文将训练后的支持向量的个数作为评测指标, 这是因为 SVM 分类算法的决策函数只与支持向量有关, 所以其计算量也同样与其个数有关. 结果如表 4 所示, 虽然从结果上看 DROP-SVM、CNN-SVM 的支持向量最少, 但其分类性能很差, 而本文算法与其他算法比较而言支持向量的个数最少, 尤其对比过取样算法, 因此可以说本文算法在不损失算法性能的前提下大大降低了 SVM 算法的检测时间和训练时间.

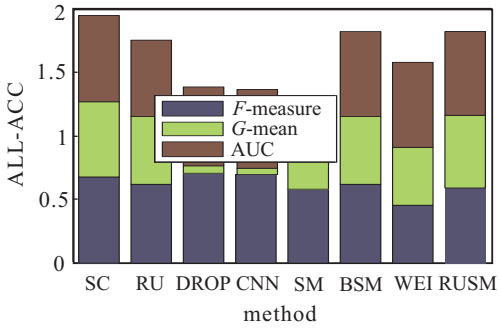

图 5　不同难分数据集的 All-ACC 的对比结果

表 4　不同难分数据集不同算法的支持向量的个数比较

SV	abalone	haberman	ionosphere	pima	yeast	Avg
SC-SVM	102.0±11.31	14.1±3.90	16.4± 3.37	228.7 ±7.29	62.2±9.07	84.7±6.99
RU	298.0±0	41.0 ±0	20.0± 0	233.0±0	210.0 ±0	160.4±0
DROP	31.10±6.10	8.6±1.26	1±0	22.2±0.92	28.9±9.80	18.4±3.62
CNN	25.3±3.50	8.3±1.49	1±0	21.2±1.14	28.9± 9.80	**16.9±3.19**
SMOTE	98.7±1.08	170.5± 48.39	93.6±30.67	595.1±24.95	824.2±11.83	356.4±23.39
BSMOTE	1045.2±117.90	186.1±43.02	80.9±19.91	614.0± 38.86	815.5±25.86	548.3±49.11
WEIGHT	580.2±23.68	69.1±18.04	40.9±12.52	322.1 ± 15.50	420.2±12.49	286.5±16.44
RUS	178.7±8.77	24.5±7.81	17.9±3.48	156.2± 3.49	122.6±18.72	99.9±8.45

4.4　不同比例下不均衡数据分类性能比较

为了验证本文提出的基于谱聚类欠取样算法在不同比例下不均衡数据中的分类性能, 本文选择上面数据库中的难分类的 5 种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照 40:1 和 50:1 的比例进行选取; 然后利用 10 次交叉验证法进行测试, 将测试结果同其他算法进行比较, 其中算法参数设置同上. 比较算法在不同比例下的 F-measure、G-mean、AUC 和性能评价指标之和 All-ACC 的值. 实验结果如图 6 和图 7 所示. 从实验结果可以看出, 对于不同比例的 5 种数据集而言, 本文算法的 All-ACC 性能都比其他算法优越, 这是因为本文算法充分结合了 SVM 算法的特点, 利用谱聚类欠取样有目的地选择多数类样本点, 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了适当的偏移.

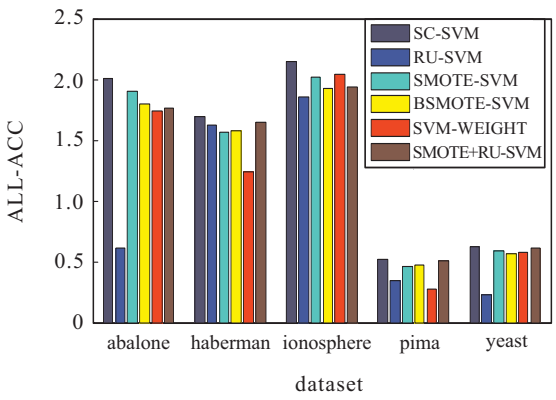

图 6　1:40 的不均衡数据的性能比较

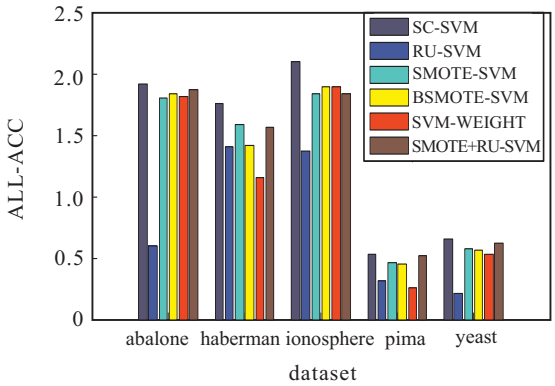

图 7　1:50 的不均衡数据的性能比较

4.5　高斯核半径参数对算法性能的影响

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 5 种难分数据集作为测试数据, 多数类样本数目和少数类样本按 50:1 的比例选取, 利用 10 次交叉验证法测试, 参数选定在 (0,160) 区间, 其他参数

设置同上, 其中本文中的谱聚类算法相似矩阵以及距离的计算都在同一个参数下的特征空间进行, 测试结果如图 8 所示.

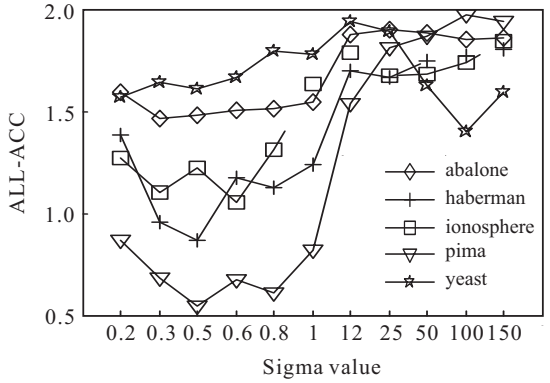

图 8 不同参数下本文算法 All-ACC 性能比较

由图 8 可以得出: 本文算法随着参数的增大, 算法性能呈明显的上升趋势, 说明针对欠取样算法而言高斯核半径参数越大分类性能越好. 这是由于本文选取的欠取样算法是基于谱聚类算法的, 本身选择的样本点都具有一定的代表性. 因此, 为了能在算法中发挥代表作用, 需要将自身的邻域半径进行扩大以便使受其影响的面积增多, 而这一特点与本文欠取样算法自身的性质是相关的.

5 结 论

本文针对 SVM 算法在不均衡数据下分类性能差的问题, 提出了一种基于谱聚类的欠取样 SVM 分类算法. 首先在 SVM 特征空间中通过对多数类样本进行聚类, 选择具有代表性的多数类样本点, 为了能进一步使训练得到的 SVM 分类界面向着多数类样本方向偏移, 在样本选择时根据与少数类样本间的距离比例进行选择, 通过适当减少边界样本的方式达到提高 SVM 算法不均衡数据分类性能的目的. 将本文算法同其他不均衡数据分类算法进行了比较, 结果表明, 本文算法在不同数据集和不同不均衡数据比例下的分类性能都优于其他算法, 同时 SVM 算法的训练时间和检测时间也大大降低. 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, 结果发现, 本文算法在参数设置较大时分类性能较好, 这一现象也同样符合本文基于谱聚类欠取样算法的机理. 需要说明的是, 鉴于谱聚类算法的参数敏感性, 是否能与集成算法相结合来进一步提升 SVM 算法处理不均衡数据集分类的性能, 将是本课题下一阶段研究的重点.

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 Knowledge 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 of Int Conf on Intelligent Computing. Hefei, 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: Spring-Verlag, 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 SM0TE[J]. Acta Electronica Sinica, 2009, 39(10): 2489-2495.)

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

(下转第1775页)