一种基于核 SMOTE 的非平衡数据集分类方法

曾志强[1,2],吴 群[2],廖备水[2],高 济[2]
(1. 厦门理工学院计算机科学与技术系,福建厦门 361024;2. 浙江大学计算机科学与技术学院,浙江杭州 310027)

摘 要: 本文提出一种基于核 SMOTE(Synthetic Minority Over-sampling Technique)的分类方法来处理支持向量机(SVM)在非平衡数据集上的分类问题.其核心思想是首先在特征空间中采用核 SMOTE 方法对少数类样本进行上采样,然后通过输入空间和特征空间的距离关系寻找所合成样本在输入空间的原像,最后再采用 SVM 对其进行训练.实验表明,核 SMOTE 方法所合成的样本质量高于 SMOTE 算法,从而有效提高 SVM 在非平衡数据集上的分类效果.

关键词: 非平衡数据集; 支持向量机; 输入空间; 特征空间; 原像
中图分类号: TP181 **文献标识码**: A **文章编号**: 0372-2112 (2009) 11-2489-07

A Classfication Method For Imbalance Data Set Based on Kernel SMOTE

ZENG Zhi-qiang[1,2], WU Qun[2], LIAO Bei-shui[2], GAO Ji[2]
(1. *Department of Computer Science and Technology, Xiamen University of Technology, Xiamen, Fujian* 361024, *China*;
2. *College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang* 310027, *China*)

Abstract: An approach based on kernel SMOTE (Synthetic Minority Over-sampling Technique) to solve classification on imbalance data set by Support Vector Machine (SVM) is presented. The method first oversamples the minority class in feature space by kernel SMOTE algorithm, then the pre-images of the synthetic instances are found based on a distance relation between feature space and input space. Finally, these pre-images are appended to the original data set to train a SVM. Experiments on real data sets indicate that compared with SMOTE approach, the samples constructed by the kernel SMOTE algorithm have the higher quality. As a result, the effectiveness of classification by SVM on imbalance data set is improved.

Key words: imbalance data set; support vector machine; input space; feature space; pre-image

1 引言

近年来,非平衡数据集的分类问题已成为分类技术研究中最具挑战性的难点之一,因而得到研究者的广泛重视.非平衡数据集是指某类样本数量明显少于其它类样本的数据集,传统的分类方法如决策树、神经网络等在非平衡数据集上难以取得令人满意的分类效果.

支持向量机(Support Vector Machine,简称 SVM)是在统计学习理论基础上发展起来的一种新的机器学习方法,它基于结构风险最小化原则,在解决小样本、非线性及高维模式识别问题中表现出许多特有的优势,在手写数字识别、语音识别、文本分类等许多实际应用中都取得了成功.然而,同其它分类方法一样,SVM 在非平衡数据集上的分类效果也不尽如人意,主要在于 SVM 训练所得的最优分类超平面会向少数类偏移,从而容易错分少数类(为方便起见,后文中将少数类样本定义为正类数据,多数类样本定义为负类数据).研究者们提出了许多策略来解决此问题,Veropoulos[1]等人提出了代价敏感训练算法,该方法通过赋予错分的正负类样本不同的惩罚系数来降低分类超平面的偏移度,此种方法简单易行并且具有一定效果,因此被视为 SVM 处理非平衡数据集的标准方法.然而,当正类样本过分稀疏时,采用此种方法会因分类超平面过分拟合正类样本而影响分类效果[2].文献[3～5]中,作者们通过对负类样本进行下采样来降低数据的不平衡率,然而,此类方法忽略了潜在有用的负类样本,因而可能降低分类器的性能[6].文献[7]中,Chawla 等人采用 SMOTE(Synthetic Minority Over-sampling TEchnique)算法人工构造少数类样本来增加正类样本的数量,从而减少数据失衡程度.同其它方法相比,SMOTE 方法对数据的预处理更为有效,因而引起研究者的广泛兴趣,后续提出许多 SMOTE 和 SVM 相结合的改进算法,如 Akbani 等人提出了 SMOTE 和代

收稿日期:2009-02-24;修回日期:2009-09-03
基金项目:国家自然科学基金项目(No. 60773177);福建省青年人才项目(No. 2008F3108)

价敏感训练相结合的 SDC(SMOTE with Different Costs)[2]算法,Yang Liu 等人提出了 SMOTE 和 Boost 相结合的 EnSVM(Ensemble of SVM)[8]算法等,实验表明,这些方法在非平衡数据集上都取得了较好的分类效果. 然而,SMOTE 算法操作在输入空间,而 SVM 工作在特征空间,在输入空间所构造的最佳样本在特征空间中不一定是最佳的,并且目前对 SMOTE 和 SVM 相结合类型算法(简称 SMOTE_SVM)所带来二次优化问题的变化的阐述是基于经验观察,而非理论分析. 本文鉴于此,1. 提出一种新颖的 SMOTE_SVM 类型算法,同上述算法不同的是,新算法在特征空间中合成新样本,从而解决在不同空间处理训练样本所带来的不一致问题,提高了合成样本的质量. 2. 从理论角度分析了 SMOTE_SVM 类型算法和 SVM 处理非平衡数据集的标准算法 Biased SVM 所对应的二次优化问题之间的差别.

2 SVM 及 Biased SVM

2.1 SVM

训练 SVM 的本质是求解一最优分类超平面问题,给定训练样本 (x_i, y_i), $i = 1, \cdots, l$,其中 $x_i \in R^h$, $y_i \in \{-1, 1\}$,求解最优分类超平面可转化为二次优化问题:

$$\min_{w,b,\xi} \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{l} \xi_i \tag{1}$$

subject to $y_i(w^{\mathrm{T}}\phi(x_i) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$, $i = 1, \cdots, l$.

其中 ξ_i, $i = 1, \cdots, l$ 为松弛因子, $C > 0$ 为误分样本的惩罚系数,用 Lagrange 乘子法可获得式(1)的对偶问题:

$$\min_{\alpha} \frac{1}{2} \alpha^{\mathrm{T}} Q \alpha - e^{\mathrm{T}} \alpha \tag{2}$$

subject to $0 \leq \alpha_i \leq C$, $i = 1, \cdots, l$, $y^{\mathrm{T}} \alpha = 0$.

其中 e 是都为 1 的向量, α_i, $i = 1, \cdots, l$ 为 Lagrange 乘子, Q 为 $l \times l$ 的对称矩阵,称为 Hessian 矩阵, $Q_{ij} = y_i y_j k(x_i, x_j) = y_i y_j \phi(x_i)^{\mathrm{T}} \phi(x_j)$, $k(x_i, x_j)$ 为核函数,它对应于采用非线性映射 $\phi: R^h \rightarrow F$ 将训练样本从输入空间映射到某一特征空间 F,在该特征空间中,样本是线性可分的. SVM 训练所得分类超平面的函数形式如下所示:

$$f(x) = \sum_{i=1}^{N_{sv}} \alpha_i y_i k(x_i, x) + b \tag{3}$$

对于非平衡数据集,SVM 训练所得的分类超平面会向正类数据偏移,因此易将正类数据误分为负类数据,从而影响分类效果.

2.2 Biased SVM

Veropoulos 等人提出了解决 SVM 在非平衡数据集上所产生的分类超平面偏移问题的策略,即 Biased SVM,其主要思想就是赋予错分的正负类样本不同的惩罚系数 C^+ 和 C^-,则目标函数变更为:

$$\min_{w,b,\xi} \frac{1}{2} w^{\mathrm{T}} w + C^+ \sum_{\{i \mid y_i = +1\}} \xi_i + C^- \sum_{\{i \mid y_i = -1\}} \xi_i \tag{4}$$

subject to $y_i(w^{\mathrm{T}}\phi(x_i) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$, $i = 1, \cdots, l$.

采用 Lagrange 乘子法可以获得(4)式的对偶问题:

$$\min_{\alpha} \quad W(\alpha) = \frac{1}{2} \alpha^{\mathrm{T}} Q \alpha - e^{\mathrm{T}} \alpha, \tag{5}$$

subject to $0 \leq \alpha_i \leq \eta C$, if $y_i = +1$

and $0 \leq \alpha_i \leq C$, if $y_i = -1$,

$$y^{\mathrm{T}} \alpha = 0$$

其中 $C = C^-$, $\eta = C^+ / C^-$. 比较式(2)和式(5)可以看出,除了 Lagrange 乘子 α 的约束不同外,两者的对偶形式完全相同. 采用 Biased SVM,那些有非零松弛变量的正类样本对应的 α_i,要大于有非零松弛变量的负类样本对应的 α_i,这样就会将分类面推向多数类,因此能够在一定程度上缓解分类超平面的偏移问题,此种方法简单易行并且具有相当效果,因此被视为 SVM 处理非平衡数据集的标准方法.

3 基于核 SMOTE 的分类算法

本文所提出的非平衡数据集分类方法的基本思想就是首先通过核函数将正类数据映射到特征空间,在特征空间中采用扩展 SMOTE 算法合成正类样本,然后根据距离约束获得合成的正类样本在输入空间的原像,最后再进行 SVM 训练,从而提高 SVM 在非平衡数据集上的分类效果.

3.1 核 SMOTE 算法

文献[7]中,Chawla 提出了通过 k 近邻构造正类样本的 SMOTE 算法,取得了较好效果. 然而,SMOTE 算法操作在输入空间,而 SVM 工作在特征空间,显然在输入空间中通过 SMOTE 算法所构造的最佳样本在特征空间中不一定是最佳的. 因此,对 SMOTE 算法进行扩展,通过在特征空间中合成新样本,以解决不同空间处理训练样本所带来的不一致问题,提高所合成样本的质量. 扩展后的 SMOTE 算法称为核 SMOTE(Kernel SMOTE)算法,简称 KSMOTE.

假设待处理的正类样本集为 $D^+ = \{x_1, x_2, \cdots, x_n\}$, $x_i \in R^h$, $i = 1, \cdots, n$,核函数 $k(,)$ 与非线性映射 $\phi: R^h \rightarrow F$ 相关,ϕ 将集合 D^+ 中的元素从输入空间映射到特征空间 F,则 KSMOTE 算法框架描述如下:

输入:正类样本集 D^+,最近邻数目 k,待构造正类样本数量与原样本数量 $|D^+|$ 的比值 N

输出:合成的正类样本集合 D_s

函数:

getRandomPoint(S):返回集合 S 中的任意一个元素;getFeatureNeighbors(x, S, k):在特征空间中从集合 S 中获得样本点 x 的 k 个最近邻并返回之;getRandom-

Number(*value*1, *value*2):返回(*value*1 *value*2)区间的一任意值;getASCOrder(*S*, *A*):按照升序方式对集合 *S* 中的元素进行排序,并返回排序完的集合.排序的指标为数组 *A* 的值,数组 *A* 的元素和集合 *S* 中的元素一一对应;getFirstPoints(*S*, *k*):返回集合 *S* 中的前 *k* 个元素组成的集合.

算法:

(1) $T:=|D^+|$, $D_s:=\varnothing$;
(2) if $N<1$ then
(3) $T:=\lfloor N\times T\rfloor$, $N:=1$;
(4) end if
(5) $N:=\lfloor N\rfloor$, $Z:=D^+$;
(6) for $i:=1$ to T
(7) { $x_i:=$getRandomPoint(Z);
(8) $D_i^+:=$getFeatureNeighbors(x_i, $D^+-\{x_i\}$, k);
(9) for $j:=1$ to N
(10) { $x_j:=$getRandomPoint(D_i^+);
(11) $\lambda_{ij}:=$getRandomNumber(0,1);
/*在特征空间中合成新样本*/
(12) $O_{ij}:=\phi(x_i)+\lambda_{ij}\times(\phi(x_j)-\phi(x_i))$; (6)
(13) $D_s:=D_s\cup\{O_{ij}\}$;
(14) $D_i^+:=D_i^+-\{x_j\}$;}
(15) $Z:=Z-\{x_i\}$;}
(16) return D_s

算法第(8)步中所用函数 getFeatureNeighbors() 的功能是寻找样本点在特征空间中的 *k* 近邻,涉及到特征空间中样本点的距离计算,在每次迭代中该函数几乎承担了所有时间开销,其对应算法框架描述如下:

函数:getFeatureNeighbors(x, S, k)

(17) for $i:=1$ to $|S|$
{/*在特征空间中计算 x_i 与 x 的距离, $x_i\in S$*/
(18) $d_i:=\|\phi(x_i)-\phi(x)\|=\sqrt{k(x_i,x_i)-2k(x_i,x)+k(x,x)}$; (7)
(19) $A[i]:=d_i$;}
(20) $S:=$getASCOrder(S, A);
(21) $B:=$getFirstPoints(S, k);
(22) return B

3.2 寻找合成样本的原像

从式(6)可以看出,特征空间中的合成样本在形式上表现为原始正类样本的线性组合,由于映射 ϕ 未知,不能直接用于 SVM 训练,因此须寻找合成样本在输入空间的原像 u_{ij},使得 $\phi(u_{ij})=O_{ij}$.然而,由于非线性映射 $\phi^{-1}:F\to R^h$ 是未知的,所以不可能精确得到合成样本在输入空间中的原像 $u_{ij}=\phi^{-1}(O_{ij})$,只能通过其它方法得到它的近似解.本文采用文献[9]中的策略,利用输入空间和特征空间之间的距离关系来寻找合成样本 O_{ij} 在输入空间的原像 u_{ij}.

要确定原像 u_{ij},首先必须建立输入空间和特征空间之间的距离关系.虽然目前只能对各向同性核函数 $k(x,y)=K(\|x-y\|)$(例如高斯核函数)确立这种距离关系,但考虑到此类核函数在实际应用中使用最广泛,因此本算法仍具有显著的实用性.在特征空间中,原始正类集合 D^+ 中的任一元素 x_l 到合成样本 O_{ij} 的距离计算如下式所示:

$$\begin{aligned}\tilde{d}_l^2(O_{ij},\phi(x_l))&=\tilde{d}_l^2(\phi(x_i)+\lambda_{ij}\times(\phi(x_j)-\phi(x_i)),\phi(x_l))\\&=\|(\phi(x_i)+\lambda_{ij}\times(\phi(x_j)-\phi(x_i)))-\phi(x_l)\|^2\\&=(1+2\lambda_{ij})k(x_l,x_l)-2k(x_l,x_i)-2\lambda_{ij}k(x_l,x_j)+\\&\quad(\lambda_{ij}-1)^2k(x_i,x_i)+2\lambda_{ij}(1-\lambda_{ij})k(x_i,x_j)+\lambda_{ij}^2k(x_j,x_j)\end{aligned}\tag{8}$$

对于高斯核函数而言,x_l 与 O_{ij} 的原像 u_{ij} 在特征空间中的距离 $\tilde{d}_l^2(\phi(u_{ij}),\phi(x_l))$ 与输入空间中的距离 $d_l^2(u_{ij},x_l)$ 维持如下关系[9]:

$$\begin{aligned}&\tilde{d}_l^2(\phi(u_{ij}),\phi(x_l))\\&=\|\phi(u_{ij})-\phi(x_l)\|^2=k(u_{ij},u_{ij})-2k(u_{ij},x_l)+k(x_l,x_l)\\&=2-2\exp(-\|u_{ij}-x_l\|^2/(2\sigma^2))=2-2\exp(-d_l^2(u_{ij},x_l)/(2\sigma^2))\end{aligned}$$

$$\Rightarrow d_l^2(u_{ij},x_l)=-2\sigma^2\ln\left(1-\frac{1}{2}\tilde{d}_l^2(\phi(u_{ij}),\phi(x_l))\right)\tag{9}$$

因为 $\tilde{d}_l^2(O_{ij},\phi(x_l))=\tilde{d}_l^2(\phi(u_{ij}),\phi(x_l))$,则根据式(8)、式(9)可以得到 $d_l^2(u_{ij},x_l)$,即样本点 x_l 与 O_{ij} 的原像 u_{ij} 在输入空间中的距离.通常,样本点与其近邻的距离在确定样本点位置过程中起着至关重要的作用,所以,在求 u_{ij} 的过程中主要考虑合成样本 O_{ij} 在特征空间中与集合 D^+ 中的 t 个近邻$\{\phi(x_1^{ij}),\phi(x_2^{ij}),\cdots,\phi(x_t^{ij})\}\subset D^+$ 在输入空间中的距离(用二次方来衡量).定义向量

$$d^2=[d_1^2,d_2^2,\cdots,d_t^2]^{\mathrm{T}}\tag{10}$$

其中 $d_l, l=1,\cdots,t$ 为 O_{ij} 的原像 u_{ij} 和它的近邻 x_l^{ij} 在输入空间中的距离.文献[9,10]采用某个未知坐标的点和其它点之间的距离约束来确定此点在空间中的坐标,笔者借鉴其思想来寻找 O_{ij} 在输入空间中的原像 u_{ij}.对于 O_{ij} 在特征空间中的 t 个近邻$\{\phi(x_1^{ij}),\phi(x_2^{ij}),\cdots,\phi(x_t^{ij})\}$,确定此 t 个近邻在输入空间中的原像$\{x_1^{ij},x_2^{ij},\cdots,x_t^{ij}\}\in R^h$ 的均值 $\bar{x}=(1/t)\sum_{l=1}^{t}x_l^{ij}$,并构建一个新的坐标系.首先创建一个 $h\times t$ 的矩阵 $X=[x_1^{ij}\ x_2^{ij}\ \cdots\ x_t^{ij}]$ 和一个 $t\times t$ 的中心矩阵

$$H=I-\frac{1}{t}LL^{\mathrm{T}}\tag{11}$$

其中 I 是一个 $t\times t$ 的单位矩阵,$L=[1,1,\cdots,1]^{\mathrm{T}}$ 为 $t\times 1$ 的向量,则矩阵 XH 是以 $\bar{x}$ 为中心的 $h\times t$ 中心矩阵:

$$XH=\left[\begin{array}{cccc} x_1^{ij}-\bar{x} & x_2^{ij}-\bar{x} & \cdots & x_t^{ij}-\bar{x} \end{array}\right] \tag{12}$$

假设矩阵 XH 的秩为 q,对其进行奇异值分解:

$$XH=[E_1,E_2]\left[\begin{array}{cc} \Lambda_1 & O \\ O & O \end{array}\right]\left[\begin{array}{c} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \end{array}\right]=E_1\Lambda_1V_1^{\mathrm{T}}=E_1\Gamma \tag{13}$$

其中 $E_1=[e_1,e_2,\cdots,e_q]$ 为一组标准正交列向量 e_i 组成的 $h\times q$ 矩阵,$\Gamma=\Lambda_1V_1^{\mathrm{T}}=[c_1,c_2,\cdots,c_t]$ 为一 $q\times t$ 矩阵,列向量 c_l 为向量 $x_l^{ij}-\bar{x}$ 在 E_1 上的投影,此时 $\|c_l\|^2=\|x_l^{ij}-\bar{x}\|^2$, $l=1,\cdots,t$,定义一个 $t\times1$ 的向量 $d_0^2=[\|c_1\|^2,\|c_2\|^2,\cdots,\|c_t\|^2]^{\mathrm{T}}$. 显然,为了获得较为精确的原像 u_{ij},距离 $d^2(u_{ij},x_l^{ij})$, $l=1,\cdots,t$ 应尽可能等于式(10)中的值,即

$$d^2(u_{ij},x_l^{ij})\approx d_l^2,\ l=1,\cdots,t \tag{14}$$

定义 $\tilde{c}\in R^{q\times1}$ 并且 $E_1\tilde{c}=u_{ij}-\bar{x}$,则

$$\begin{aligned} d_l^2&\approx\|u_{ij}-x_l^{ij}\|^2=\|(u_{ij}-\bar{x})-(x_l^{ij}-\bar{x})\|^2 \\ &=\|\tilde{c}\|^2+\|c_l\|^2-2(u_{ij}-\bar{x})^{\mathrm{T}}(x_l^{ij}-\bar{x}) \end{aligned},\ l=1,\cdots,t \tag{15}$$

对等式(15)从 1 累加到 t,由于 XH 为中心矩阵,则式(15)中内积项的累加和为零. 累加完的等式如下式所示:

$$\sum_{l=1}^{t}d_l^2=t\|\tilde{c}\|^2+\sum_{l=1}^{t}\|c_l\|^2\Rightarrow\|\tilde{c}\|^2=\frac{1}{t}\sum_{l=1}^{t}(d_l^2-\|c_l\|^2),\ l=1,\cdots,t \tag{16}$$

将式(16)中 $\|\tilde{c}\|^2$ 的表达式代入式(15)并重新排列可得,

$$2(x_l^{ij}-\bar{x})^{\mathrm{T}}(u_{ij}-\bar{x})=\|c_l\|^2-d_l^2-\frac{1}{t}\sum_{l=1}^{t}(\|c_l\|^2-d_l^2),\ l=1,\cdots,t \tag{17}$$

采用矩阵的形式来表达式(17)可得

$$2\Gamma^{\mathrm{T}}\tilde{c}=(d_0^2-d^2)-\frac{1}{t}LL^{\mathrm{T}}(d_0^2-d^2) \tag{18}$$

由于 Γ 为中心矩阵,所以 $\Gamma LL^{\mathrm{T}}=0$. 对式(18)进行适当变换可得

$$\tilde{c}=\frac{1}{2}(\Gamma\Gamma^{\mathrm{T}})^{-1}\Gamma(d_0^2-d^2)=\frac{1}{2}\Lambda_1^{-1}V_1^{\mathrm{T}}(d_0^2-d^2) \tag{19}$$

最后,将 $\tilde{c}$ 转换回输入空间中的原始坐标系,可以得到合成样本 O_{ij} 在输入空间中的原像的近似值

$$u_{ij}=\frac{1}{2}E_1\Lambda_1^{-1}V_1^{\mathrm{T}}(d_0^2-d^2)+\bar{x} \tag{20}$$

获得特征空间中合成的正类样本原像后,就可将其加入原始的正类数据集,从而增大正类样本的数量,减小训练样本的失衡程度,合并正负两类数据集进行 SVM 学习,即可获得最终的分类超平面.

3.3 计算复杂度分析

定义 n, m 分别表示原始正负类样本数量,基于 KSMOTE 的非平衡数据集训练算法的时间开销由以下三部分组成:

(1)合成新样本的 KSMOTE 算法,该算法每次迭代的时间花销在于根据式(7)获得样本点的 N 个近邻,所需时间为 $O(nt)$, $O(t)$ 代表核函数计算花销. 一般情况下,算法迭代次数 $T=n$,因此,KSMOTE 算法所需时间为 $O(n^2t)$;(2)原像算法,确定每个原像的主要时间花销在于根据式(8)确定合成样本在特征空间中的 t 个近邻,所需时间为 $O(3t)=O(t)$,则确定所有合成样本对应原像所需时间为 $O(nNt)$;(3)SVM 训练算法,本文采用具有较快训练速度的 SMO 算法,其时间花销为 $O((n(N+1)+m)\rho)$,其中 ρ 表示 SMO 算法迭代次数. 通常情况下 $n(N+1)+m\leqslant2m$,因此,SMO 算法的时间复杂度为 $O(m\rho t)$.

由于 $nN\leqslant m$,并且一般情况下 $t\leqslant\rho$,因此,综合以上三部分可以得出基于 KSMOTE 的训练算法的时间复杂度为 $O(n^2+nNt+m\rho)=O(n^2+m\rho)$(以核函数计算次数来度量).

该算法的空间复杂度取决于 SVM 训练算法存储对应于最终训练样本集的 Hessian 矩阵所需空间,因此,该算法的空间复杂度为 $O((n(N+1)+m)^2)=O(m^2)$.

3.4 相关二次优化问题分析

SMOTE-SVM 类型算法由于在非平衡数据集上取得良好的分类效果而被广泛采用,然而目前缺乏对该类型算法的理论阐述,受文献[3]启发,本文从理论角度分析 SMOTE-SVM 类型算法和 Biased SVM 所对应二次优化问题之间的差别,这对进一步研究该类型算法在非平衡数据集上的分类问题,具有积极意义.

假设原始正,负类数据集分别为 $D^+=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, $D^-=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$,则原始数据集 $D=D^+\cup D^-$. 定义 $J((x_i,y_i))=\{(u_{i,1},y_{i,1}),(u_{i,2},y_{i,2}),\cdots,(u_{i,N},y_{i,N})\}$ 表示采用 SMOTE 类算法对 D^+ 进行预处理后,对应于原始正类样本 (x_i,y_i) 的合成样本集(即 (x_i,y_i) 与它的 k 个近邻合成的样本集),其中 N 表示合成的正类样本数量与 D^+ 的比值,则总的合成样本集 $D_s=\bigcup_{i=1}^{n}J((x_i,y_i))$,和数据集 $\tilde{D}=D_s\cup D$ 相关的 SVM 对偶问题为:

$$\begin{aligned} \min_{\tilde{\alpha}}\quad & \tilde{W}(\tilde{\alpha})=\frac{1}{2}\tilde{\alpha}^{\mathrm{T}}\tilde{Q}\tilde{\alpha}-e^{\mathrm{T}}\tilde{\alpha} \\ \text{subject to}\quad & 0\leqslant\tilde{\alpha}_i\leqslant C,\ i=1,\cdots,|\tilde{D}| \\ & \tilde{y}^{\mathrm{T}}\tilde{\alpha}=0 \end{aligned} \tag{21}$$

这就是 SMOTE-SVM 类型算法所对应的目标问题. 和 Biased SVM(简称 BSVM)对应的优化问题式(5)相比,SMOTE-SVM 对应的优化问题有何变化呢? 首先定义集合

$$\bar{D}_i^+=\{(\bar{u}_c,\bar{y}_c)\mid\forall(u_c,y_c)\in D_s \text{ if } (u_c,y_c)\in J((x_i,y_i))\} \tag{22}$$

$\bar{u}_c = x_i$ and $\bar{y}_c = 1\}$, $i = 1, \cdots, n$, $\Omega((x_i, y_i)) = \{(u_i, y_i) \mid u_i = x_i\} \cup J((x_i, y_i))$, $i = 1, \cdots, n$, 根据以上定义可知 $\forall (x_i, y_i) \in D$, 如果 $y_i = 1$, 则 $|\Omega((x_i, y_i))| = N + 1$; 如果 $y_i = -1$, 则 $|\Omega((x_i, y_i))| = 1$. 定义

$$\bar{D}_o = \{(\bar{u}_i, \bar{y}_i) \mid \forall (x_i, y_i) \in D, \bar{u}_i = x_i \text{ and } \bar{y}_i = y_i\} \quad (23)$$

$$\bar{D} = (\bigcup_{i=1}^{n} \bar{D}_i^+) \cup \bar{D}_o$$

和数据集 $\bar{D}$ 相关的 *SVM* 对偶问题为:

$$\min_{\bar{\alpha}} \quad \bar{W}(\bar{\alpha}) = \frac{1}{2}\bar{\alpha}^{\mathrm{T}}\bar{Q}\bar{\alpha} - e^{\mathrm{T}}\bar{\alpha} \quad (24)$$

$$\text{subject to} \quad 0 \leq \bar{\alpha}_i \leq C, i = 1, \cdots, |\bar{D}|$$

$$\bar{y}^{\mathrm{T}}\bar{\alpha} = 0$$

由于 $\bar{D}$ 中包含很多重复样本,所以称式(24)为重复支持向量机(Duplicate Support Vector Machine,简称 DSVM)[3]. 根据以上定义可推导出:

定理 1　如果 BSV M 的参数 $\eta = N + 1$, $\bar{\alpha}^*$ 为 DSVM 的最优解, α^* 为 BSVM 的最优解,则 $\bar{W}(\bar{\alpha}^*) = W(\alpha^*)$. 并且对于任意的 $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]$,如果满足 $\{\alpha_i = \sum_{(u_c, y_c) \in \Omega((x_i, y_i))} \bar{\alpha}_c^*, i = 1, \cdots, n\}$,则 α 为 BSVM 的最优解. 反之,对于任意的 $\bar{\alpha} = [\bar{\alpha}_1, \bar{\alpha}_2, \cdots, \bar{\alpha}_{(N+1)n+m}]$,如果满足 $\{\sum_{(u_c, y_c) \in \Omega((x_i, y_i))} \bar{\alpha}_c = \alpha_i^*, i = 1, \cdots, n\}$,则 $\bar{\alpha}$ 为 DSVM 的最优解.

证明　对于任意的 α,如果 $\{\alpha_i = \sum_{(u_c, y_c) \in \Omega((x_i, y_i))} \bar{\alpha}_c^*, i = 1, \cdots, n\}$,又因 $\eta = N + 1$,则 α 满足式(5)的约束条件,即 α 为 BSVM 的可行解. 对于任意的 $\bar{\alpha}$,如果 $\{\sum_{(u_c, y_c) \in \Omega((x_i, y_i))} \bar{\alpha}_c = \alpha_i^* / |\Omega((x_i, y_i))|, \forall (u_c, y_c) \in \Omega((x_i, y_i)), i = 1, \cdots, n\}$,则 $\bar{\alpha}$ 满足式(24)的约束条件,即 $\bar{\alpha}$ 为 DSVM 的可行解. 根据 D 和 $\bar{D}$ 的定义可以推出:

$$\frac{1}{2}\sum_{i,j=1}^{n+m} \alpha_i^* \alpha_j^* y_i y_j k(x_i, x_j) - \sum_{i=1}^{n+m} \alpha_i^* = \frac{1}{2}\sum_{p,q=1}^{(n+1)N+m} \bar{\alpha}_p \bar{\alpha}_q \bar{y}_p \bar{y}_q k(\bar{u}_p, \bar{u}_q) - \sum_{p=1}^{(n+1)N+m} \bar{\alpha}_p \quad (25)$$

这意味着 $W(\alpha^*) = \bar{W}(\bar{\alpha})$,同理可证 $W(\alpha) = \bar{W}(\bar{\alpha}^*)$. 由于 α^*, $\bar{\alpha}^*$ 分别为 BSVM 和 DSVM 的最优解,因此 $W(\alpha^*) \leq W(\alpha)$, $\bar{W}(\bar{\alpha}^*) \leq \bar{W}(\bar{\alpha})$,综合以上式子可以推出 $\bar{W}(\bar{\alpha}^*) = W(\alpha^*)$. 又因 $\bar{W}(\bar{\alpha}) = W(\alpha^*) = \bar{W}(\bar{\alpha}^*)$,因此可推出 $\bar{\alpha}$ 为 DSVM 的最优解,类似可推出 α 为 BSVM 的最优解.

根据定理 1 可知,求解 $\eta = N + 1$ 时的 BSVM 所对应的二次优化问题等价于求解 DSVM 所对应的二次优化问题,因此,衡量 SMOTE_SVM 和 BSVM 的差别就可转化为比较 SMOTE_SVM 和 DSVM 对应二次优化问题的差别. 需要指出的是,式(5)中的参数 η 的设置目前缺乏理论指导,一般启发式地设为 m/n,即负类与正类样本数量的比值. 定理 1 中的前提条件 $\eta = N + 1$ 正是这个比值的反映. 根据 $\tilde{D}$ 和 $\bar{D}$ 的定义可知,SMOTE_SVM 和 DSVM 具有相同的数据规模,比较式(21),(24)可以看出,二者具有相同的二次优化模型,区别仅在于 Hessian 矩阵的不同,因此,比较二者的差别可以看作当 Hessian 矩阵从 $\bar{Q}$ 转化为 $\tilde{Q}$ 时,该二次优化模型的稳定性.

定理 2　如果 $\tilde{Q}$ 是正定矩阵并且 $\varepsilon = \|\tilde{Q} - \bar{Q}\|$, $\tilde{\alpha}^*$ 和 $\bar{\alpha}^*$ 分别为式(21),(24)的最优解,则

$$\|\bar{\alpha}^* - \tilde{\alpha}^*\| \leq \frac{\bar{r}C\varepsilon}{\chi}, \tilde{W}(\bar{\alpha}^*) - \tilde{W}(\tilde{\alpha}^*) \leq \frac{(\bar{r}^2 + \tilde{r}^2)C^2\varepsilon}{2} \quad (26)$$

其中 χ 表示矩阵 $\tilde{Q}$ 的最小特征值,$\tilde{r}$ 和 $\bar{r}$ 分别表示对应于式(21),(24)的支持向量数[3].

证明　由于式(21),(24)具有相同的可行域,并且 $\tilde{\alpha}^*$ 和 $\bar{\alpha}^*$ 分别为式(21),(24)的最优解,可以推出:

$$(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}} \nabla \tilde{W}(\tilde{\alpha}^*) \geq 0, (\tilde{\alpha}^* - \bar{\alpha}^*)^{\mathrm{T}} \nabla \bar{W}(\bar{\alpha}^*) \geq 0 \quad (27)$$

其中 $\nabla \bar{W}(\bar{\alpha}) = \bar{Q}\alpha - e$, $\nabla \tilde{W}(\tilde{\alpha}) = \tilde{Q}\alpha - e$ 表示梯度. 相加式(27)的两个不等式并做适当调整可得:

$$\begin{aligned}&(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\nabla \bar{W}(\bar{\alpha}^*) - \nabla \tilde{W}(\tilde{\alpha}^*)) \leq 0\\ &\Rightarrow (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\nabla \bar{W}(\bar{\alpha}^*) - \nabla \tilde{W}(\tilde{\alpha}^*)) \leq (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\nabla \bar{W}(\tilde{\alpha}^*) - \nabla \bar{W}(\tilde{\alpha}^*))\\ &\Rightarrow (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\nabla \bar{W}(\bar{\alpha}^*) - \nabla \bar{W}(\tilde{\alpha}^*)) \leq (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\nabla \tilde{W}(\tilde{\alpha}^*) - \nabla \bar{W}(\tilde{\alpha}^*)) \quad (28)\end{aligned}$$

代入相应的梯度表达式可以推出

$$(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}\bar{Q}(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}} \leq (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\tilde{Q} - \bar{Q})\tilde{\alpha}^* \quad (29)$$

对式(29)的左右两边同时加上 $(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\tilde{Q} - \bar{Q})(\bar{\alpha}^* - \tilde{\alpha}^*)$ 可得

$$(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}\tilde{Q}(\bar{\alpha}^* - \tilde{\alpha}^*) \leq (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\tilde{Q} - \bar{Q})\bar{\alpha}^* \quad (30)$$

如果 χ 为矩阵 $\tilde{Q}$ 的最小特征值,可以推出

$$\chi\|\bar{\alpha}^* - \tilde{\alpha}^*\|^2 \leq (\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}\tilde{Q}(\bar{\alpha}^* - \tilde{\alpha}^*) \quad (31)$$

$(\bar{\alpha}^* - \tilde{\alpha}^*)^{\mathrm{T}}(\tilde{Q} - \bar{Q})\bar{\alpha}^* \leq \|\bar{\alpha}^* - \tilde{\alpha}^*\|\|\tilde{Q} - \bar{Q}\|\|\bar{\alpha}^*\|$

又因为 $\|\bar{\alpha}^*\| \leq \bar{r}C$,根据式(30),(31)可以推出 $\|\bar{\alpha}^* - \tilde{\alpha}^*\| \leq \frac{\bar{r}C\varepsilon}{\chi}$. 由于 $\tilde{\alpha}^*$ 是式(21)的最优解,所以 $\tilde{W}(\bar{\alpha}^*) - \tilde{W}(\tilde{\alpha}^*) > 0$,同时:

$$\begin{aligned}\tilde{W}(\bar{\alpha}^*) - \tilde{W}(\tilde{\alpha}^*) &= \frac{1}{2}(\bar{\alpha}^*)^{\mathrm{T}}(\tilde{Q} - \bar{Q})\bar{\alpha}^* + \bar{W}(\bar{\alpha}^*) - \tilde{W}(\tilde{\alpha}^*)\\ &\leq \frac{1}{2}(\bar{\alpha}^*)^{\mathrm{T}}(\tilde{Q} - \bar{Q})\bar{\alpha}^* + \bar{W}(\tilde{\alpha}^*) - \tilde{W}(\tilde{\alpha}^*)\end{aligned}$$

$$= \frac{1}{2}(\bar{\boldsymbol{\alpha}}^*)^{\mathrm{T}}(\tilde{\mathrm{Q}} - \bar{\mathrm{Q}})\bar{\boldsymbol{\alpha}}^* - \frac{1}{2}(\tilde{\boldsymbol{\alpha}}^*)^{\mathrm{T}}(\tilde{\mathrm{Q}} - \bar{\mathrm{Q}})\tilde{\boldsymbol{\alpha}}^*$$

$$\leq \frac{1}{2}\|\tilde{\mathrm{Q}} - \bar{\mathrm{Q}}\|\|\bar{\boldsymbol{\alpha}}^*\|^2 + \frac{1}{2}\|\tilde{\mathrm{Q}} - \bar{\mathrm{Q}}\|\|\tilde{\boldsymbol{\alpha}}^*\|^2$$

$$\leq \frac{(\bar{r}^2 + \tilde{r}^2)C^2\varepsilon}{2}$$

4 实验结果及分析

采用 *VC* + + 6.0 和 Matlab 7.0 实现了所提出的基于 KSMOTE 的分类算法,并将它和 SVM,Biased SVM,SMOTE 算法进行比较,其中 SMOTE 算法表示先用 SMOTE 合成样本,再用标准 SVM 进行训练. LIBSVM 2.88 被选取作为标准 SVM 训练算法. 实验所用数据集是研究非平衡数据分类问题常用的六个公开数据集,它们来自 UCI 机器学习数据库,分别是 Abalone,Car,Glass,Phoneme,Pima,Segmentation 数据集. 这六个数据集详细描述如表 1 所示(其中不平衡率表示负类样本和正类样本数量的比值).

本实验采用交叉验证方法,将上述六个数据集每个平均分为四份,并使每份子集保持和总体样本相同的不平衡率,每次实验选取其中三份作为训练集,余下一份作为测试集,将四次实验结果的平均值作为该算法的最终结果. 所有实验皆采用高斯核函数,LIBSVM 的参数 *C* 和高斯核函数参数的选取是在实验数据集一随机抽取的较小子集上采用 10 次交叉测试所得结果的最优值.

表 1 实验数据集描述

数据集	样本总数	正类样本数	负类样本数	不平衡率
Abalone	4,177	32	4,145	129.53
Car	1,728	69	1,659	24.04
Glass	214	29	185	6.38
Phoneme	5,404	1,586	3,818	2.41
Pima	768	268	500	1.87
Segmentation	210	30	180	6

4.1 技术指标

表 2 为二类问题的混淆矩阵,*TP* 表示预测正确的正例数目,*TN* 表示预测正确的负例数目,*FP* 表示预测错误的负例数目,*FN* 表示预测错误的正例数目.

表 2 二类问题的混淆矩阵

	预测正类	预测负类
实际正类	*TP*	*FN*
实际负类	*FP*	*TN*

在非平衡数据集分类的实际应用中,人们往往更关心小类别的分类效果,因此绝大部分相关文献[2][5,6]定义:

$$g = \sqrt{acc^+ \cdot acc^-} \tag{33}$$

作为衡量算法效果的技术指标,其中:

$$acc^+ = \frac{TP}{TP + FN},\ acc^- = \frac{TN}{TN + FP} \tag{34}$$

根据定义可以看出,acc^+,acc^- 分别表示正负类样本的分类准确率,*g* 值与 acc^+,acc^- 的关系是非线性的,acc^+(acc^-)的值越小,则 *g* 值就越小,这就意味着少数类样本错分得越多,错分的代价就越大.

4.2 结果及分析

实验结果如表 3 所示,对于 SMOTE 和 KSMOTE 而言,括号内的值表示对应数据集正类样本的上采样率,即 3.1 节算法参数 *N* 的值.

表 3 不同算法的 g 值

数据集	SVM	Biased SVM	SMOTE(*N*)	KSMOTE(*N*)
Abalone	0	0.8137	0 (10)	0 (10)
Car	0	0.3227	0.9884 (5)	0.9875 (5)
Glass	0.8658	0.8814	0.9236 (1)	0.9328 (1)
Phoneme	0.8276	0.8312	0.8347 (1)	0.8543 (1)
Pima	0.7119	0.7326	0.7456 (1)	0.7833 (1)
Segmentation	0.9184	0.9366	0.9773 (1)	0.9865 (1)
平均值	0.5540	0.7530	0.7449	0.7574

从表 3 可以看出,在六个数据集上,SVM 取得最小的平均 *g* 值,Biased SVM 的平均 *g* 值略高于 SMOTE,然而,这是因为 SMOTE 对应于 Abalone 这个深度不平衡数据集的 *g* 值为零,从而影响了相应的平均值. 从总体上看,在除 Abalone 以外的其它五个数据集上,对应于 SMOTE 算法的 *g* 值都要远高于 Biased SVM,这说明对于非平衡数据集,SMOTE. SVM 类方法总体上确实要优于 Biased SVM. 从表 3 还可看出,KSMOTE 算法取得最高的平均 *g* 值,在相同 *N* 值情况下,除了在 Car 数据集上,对应于 KSMOTE 的 *g* 值略低于 SMOTE,在 Abalone 数据集上,两算法取得相同 *g* 值外,在其它四个数据集上,KSMOTE 算法对应的 *g* 值都要高于 SMOTE 算法.

表 3 显示的是 KSMOTE 和 SMOTE 算法在 N 的一种取值情况下的分类效果比较,当 *N* 取不同值时,KSMOTE 算法的分类效果是否仍然高于 SMOTE 算法?图 1 显示的是,当 *N* 取不同值时,在每种取值情况下,对应于 KSMOTE 和 SMOTE 算法的分类效果比较(本实验在 Car,Glass,Segmentation 三个数据集上进行,其中 X 轴对应 *N* 的取值,Y 轴对应 *g* 值). 从图 1 可以看出,在三个数据集上,对应于 *N* 的不同取值,KSMOTE 算法的总体分类效果要好于 SMOTE 算法,这进一步证实了,工作于特征空间的 KSMOTE 算法所构造的样本的质量,要高于工作于输入空间的 SMOTE 算法所合成的样本的质量,因此能够更好地处理 SVM 在非平衡数据集上的分类问题,从而取得更好的分类效果.

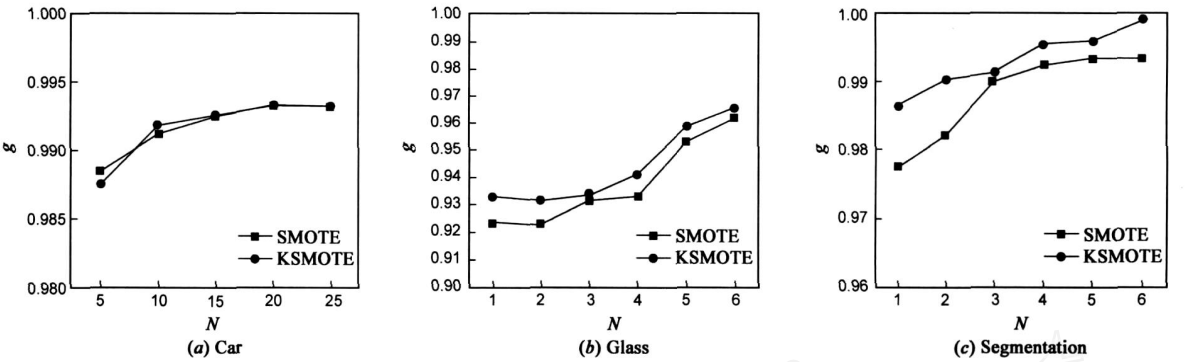

图1　基于SMOTE和KSMOTE的分类算法在Car, Glass和Segmentation三个数据集上，随着N的不同取值的分类效果比较

5　结论

针对 SMOTE_SVM 类型算法存在的问题,提出一种新颖的 SMOTE_SVM 类型算法:基于 KSMOTE 的分类算法,该算法工作于特征空间,从而解决在不同空间处理训练样本所带来的不一致问题.实验结果表明,该算法所构造的少数类样本具有更高的数据质量,因此能够更为有效地解决数据失衡问题,在非平衡数据集上取得更好的分类效果.另一方面,本文从理论角度分析了 SMOTE_SVM 类型算法和 SVM 处理非平衡数据集的标准算法 Biased SVM 所对应二次优化问题之间的差别,这对进一步研究 SMOTE_SVM 类型算法在非平衡数据集上的分类问题,具有积极意义.在接下来的工作中,将进一步完善所提出的算法,使之适用于任意核函数.

参考文献:

[1] Veropoulos K., Campbell C. and Cristianini N. Controlling the Sensitivity of Support Vector Machines[A]. Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 1999)[C]. Stockholm, Sweden: IJCAI Press, 1999: 55 - 60.

[2] R. Akbani, S. Kwek and N. Japkowicz. Applying Support Vector Machines to Imbalanced Datasets[A]. Proceedings of the 15th European Conference on Machine Learning (ECML 2004)[C]. Italy: Springer Press, 2004. 39 - 50.

[3] Yuan J., Li J., and Zhang B. Learning Concepts from Large Scale Imbalanced Data Sets using Support Cluster Machines[A]. Proceedings of the 14th annul ACM International Conference on Multimedia[C]. Santa Barbara: ACM Press, 2006. 441 - 450.

[4] P. Kang and S. Cho. EUS SVMs: Ensemble of Under - Sampled SVMs for Data Imbalance Problems[A]. Proceedings of the 13th International Conference on Neural Information Processing (ICONIP 2006)[C]. Hong Kong: Springer Press, 2006: 837 - 846.

[5] 李鹏,汪晓龙,刘远超,王宝勋. 一种基于混合策略的失衡数据集分类方法[J]. 电子学报, 2007, 35(11): 2161 - 2165.
Li Peng, Wang Xiao-long, Liu Yuan-chao, Wang Bao-xun. A classification method for imbalance data set based on hybrid strategy[J]. Acta Electronica Sinica, 2007, 35(11): 2161 - 2165. (in Chinese)

[6] T Imam, K M Ting, J Kamruzzaman. z - SVM: An SVM for Improved Classification of Imbalanced Data [A]. Proceedings of the 19th Australian Joint Conference on Artifical Intelligence (AJCAI 2006)[C]. Hobart, Australia: Springer Press, 2006. 264 - 273.

[7] Chawla N V, Bowyer K W, Hall L O, Kegelmeyer W. P. Smote: Synthetic Minority Over-sampling Technique[J]. Journal of Artificial Intelligence Research. (JAIR), 2002, 16: 321 - 357.

[8] Y. Liu, A. An, X. Huang. Boosting prediction accuracy on imbalanced datasets with SVM ensembles[A]. Proceedings of the 10th Pacific - Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006)[C]. Singapore: Springer Press, 2006: 107 - 118.

[9] J T Kwok, I W Tsang. The Pre-image Problem in Kernel Methods [J]. IEEE Transactions on Neural Networks, 2004, 15(6): 1517 - 1525.

[10] J C Gower. Adding a Point to Vector Diagrams In Multivariate Analysis [J]. Biometrika, 1968, 55(3): 582 - 585.

作者简介:

曾志强　男,1971 年出生于福建厦门,博士,讲师.研究方向:统计学习理论、模式识别.
E-mail: lbxzzq@163.com

吴　群　男,1978 年出生于江西临川,博士,助理研究员.研究方向:模式识别与人工智能.
E-mail: wuq@zju.edu.cn