基于粗糙集的多类 CVM

牛 罡, 商 琳
(南京大学软件新技术国家重点实验室, 江苏南京 210093)

摘 要: 标准的 SVM 对于训练集具有 $O(\ell^3)$ 的时间复杂度和 $O(\ell^2)$ 的空间复杂度, 2005 年提出的 CVM 具有线性的时间复杂度和与训练集大小无关的空间复杂度. 本文结合粗糙集和 CVM, 提出了一种新的多类分类 RSCVM 方法, 该方法对二类 CVM 定义上近似和下近似, 然后扩展到多类情形. 本文最后给出在真实世界数据集上的实验结果及其分析, 显示 RSCVM 方法具有快速和产生较少支持向量的优点.

关键词: 粗糙集; 上近似; 下近似; CVM (core vector machine)
中图分类号: TP181 文献标识码: A 文章编号: 0372-2112 (2008) 12A-055-05

Rough Set Based Multi-Class Core Vector Machine

NIU Gang, SHANG Lin
(National Laboratory for Novel Software Technology of Nanjing University, Nanjing, Jiangsu 210093, China)

Abstract: While the standard SVM has $O(l^3)$ time complexity and $O(l^2)$ space complexity with the size of the training set, the CVM proposed in 2005 has linear time complexity, and the space complexity of CVM is independent of the training set's size. In this paper we proposed a novel method called RSCVM. We first defined the upper and lower approximation of a binary CVM, then extended the definition to the multi-class situation. Hence RSCVM combined the CVM and rough set theory. We also gave some experiment results on several real-world data sets that illustrated RSCVM's merit of faster speed and less support-vectors.

Key words: rough set; upper approximation; lower approximation; CVM (core vector machine)

1 引言

关于线性判别分析的最早理论在 20 世纪 30 年代即由 Fisher 等人创立, 在 1956 年 Rosenblatt 提出了具有深远影响的感知器学习算法[12]. 作为最早的线性学习器之一, 感知器只能学习线性可分的数据集, 为克服此限制, SVM 应运而生[13]. SVM 需要指定一个核函数, 该函数计算原始输入数据在隐式定义的核诱导特征空间中的内积. 一旦确定了核函数, 无论训练或测试 SVM 都只需要用到输入数据在核函数下的值, 这被称为核技巧. 另外, 与感知器不同, SVM 试图在核诱导特征空间中通过最小化边界分布的泛化误差建立分离超平面. 自从 Vapnik 创立统计学习理论以来, SVM 以其坚实的理论基础和良好的实用性能而得到了广泛的关注.

训练 SVM 是一个二次规划问题. 设训练数据集含有 l 个样本, 传统的 QP 求解方法的时间复杂度为 $O(\ell^3)$, 空间复杂度至少为 $O(\ell^2)$, 这使得 SVM 不能运行在大型的数据集上, 而现实世界中的数据集往往都是庞大的. 对此, 人们提出了一些改进方法, 最常用和最成熟的是 SMO 方法[4]. 缓存核矩阵等技术可以加速 SMO 方法, 使得 SMO 在实践中的时间复杂度为 $O(\ell)$ 和 $O(\ell^3)$ 之间[4], 但这只是经验的, 并没有理论的证明. 在 2005 年 Tsang 等人提出了 CVM[1]. CVM 把 QP 求解转化为最小闭球(minimum enclosing ball)问题, 并使用一个逼近率为 $(1+\varepsilon)$ 的近似算法[5], 得到原 QP 的一个 $(1+\varepsilon)^2$ 近似, 可以在理论上证明它的时间复杂度为 $O(\ell)$, 空间复杂度为 $O(1)$[1]. 作为 CVM 的后续工作, Tsang 等人又在 2007 年提出了 BVM[3]. CVM 和 BVM 无论在理论上还是实践上都具有巨大的优越性.

另一方面, 现实世界中的分类一般不只涉及两类问题, 人们普遍认为多类分类问题比二类分类更难处理. 使用 SVM 的多类分类方法有常见的 1-v-r (one-versus-rest), 1-v-1 (one-versus-one)[6], 以及 Platt 等人提出的 DAGSVM[7]. Lingras 等人于 2007 年提出了一种基于粗糙集的 1-v-1 和 1-v-r 方法[2], 通过对二类分类 SVM 定义上下近似, 把粗糙集理论与支持向量机结合起来解决多类

收稿日期: 2008-05-07; 修回日期: 2008-09-26
基金项目: 国家自然科学基金(No. 60503022)

分类问题. Lingras 等人的工作为粗糙集理论与统计学习理论的结合提供了一个崭新的视角, 具有重要意义. 另一方面求解上下近似需要很多额外的计算开销, 使得该方法只能应用于小规模人工数据集.

本文基于 Lingras 等人的基本思想, 对于二类 SVM 在线性可分时的上下近似描述, 我们给出了一个新的形式化定义, 并在保持一致性的条件下对该定义进行扩充, 使得该定义适合于非线性可分的情形, 并进一步通过在核诱导特征空间中对训练集进行空间划分, 给出多类的上下近似定义, 提出了基于粗糙集的多类分类 CVM, 即 RSCVM 方法.

2 Core Vector Machines

本节简要介绍 CVM[1] 及其改进 BVM[3].

2.1 CVM 分类器

给定训练集 $S_{train}=\{(\boldsymbol{x}_i, y_i)\mid \boldsymbol{x}_i\in\mathbf{R}^d, y_i\in\{+1,-1\}\}_{i=1}^{\ell}$ 和核函数 k, 设 φ 是与 k 对应的特征映射, 令 $S=\{(\varphi(\boldsymbol{x}_i), y_i)\}_{i=1}^{\ell}$, 则 S 的最小闭球定义为包含 S 中所有点的最小球. 形式化的定义为, S 的 MEB 为 $B(\boldsymbol{c}^*, R^*)$, 其中球心 $\boldsymbol{c}^*$ 和半径 R^* 满足

$$\begin{aligned}&\min_{\boldsymbol{c},R} R^2\\ &\text{s.t. } \|\boldsymbol{c}-\varphi(\boldsymbol{x}_i)\|^2\leqslant R^2,\ \forall i\end{aligned}\tag{1}$$

对偶形式为

$$\begin{aligned}&\max_{\alpha_i}\sum_{i=1}^{\ell}\alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}_i)-\sum_{i,j=1}^{\ell}\alpha_i\alpha_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)\\ &\text{s.t. }\sum_{i=1}^{\ell}\alpha_i=1;\ \alpha_i\geqslant 0,\ \forall i\end{aligned}\tag{2}$$

如果核函数满足

$$k(\boldsymbol{x},\boldsymbol{x})=\kappa,\quad \kappa\text{为一常量}\tag{3}$$

则有[1]

$$\begin{aligned}&\max_{\alpha_i}-\sum_{i,j=1}^{\ell}\alpha_i\alpha_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)\\ &\text{s.t. }\sum_{i=1}^{\ell}\alpha_i=1;\ \alpha_i\geqslant 0,\ \forall i\end{aligned}\tag{4}$$

另一方面, 考虑 L2-SVM, 当核函数满足式(3)时, 它的对偶形式为[1]

$$\begin{aligned}&\max_{\alpha_i}-\sum_{i,j=1}^{\ell}\alpha_i\alpha_j\left(y_iy_jk(\boldsymbol{x}_i,\boldsymbol{x}_j)+y_iy_j+\frac{\delta_{ij}}{C}\right)\\ &\text{s.t. }\sum_{i=1}^{\ell}\alpha_i=1;\ \alpha_i\geqslant 0,\ \forall i\end{aligned}\tag{5}$$

记 $\tilde{k}(\boldsymbol{z}_i,\boldsymbol{z}_j)=y_iy_jk(\boldsymbol{x}_i,\boldsymbol{x}_j)+y_iy_j+\frac{\delta_{ij}}{C}$,

则 $\tilde{k}(\boldsymbol{z},\boldsymbol{z})=\kappa+1+\frac{1}{C}=\tilde{\kappa}$, 上式转化为

$$\begin{aligned}&\max_{\alpha_i}-\sum_{i,j=1}^{\ell}\alpha_i\alpha_j\tilde{k}(\boldsymbol{z}_i,\boldsymbol{z}_j)\\ &\text{s.t. }\sum_{i=1}^{\ell}\alpha_i=1;\ \alpha_i\geqslant 0,\ \forall i\end{aligned}\tag{6}$$

这是一个最小闭球问题. 事实上, 只要核函数 k 满足式(3), L2-SVM 可以与最小闭球问题相互转化[1]. 图 1 是 CVM 学习算法[3].

算法 1: CVM Algorithm
1: 初始化 S_0, $\boldsymbol{c}_0$, R_0 并置 $t=0$.
2: 如果 $\forall z$, $\varphi(z)\in B(\boldsymbol{c}_t, (1+\varepsilon)R_t)$, 则算法停止. 否则, 设 $\varphi(z_t)$ 是一个这样的点, 令 $S_{t+1}=S_t\cup\varphi(z_t)$.
3: 寻找 MEB(S_{t+1}).
4: $t=t+1$ 并转至 2.

图 1 CVM 算法

2.2 BVM 分类器

给定 ε, 不使用随机化加速[8]的 CVM 至多在第 $2/\varepsilon$ 次迭代时停止, 时间复杂度为 $O(\varepsilon^{-2}\ell+\varepsilon^{-4})$[1]; 当使用随机化加速[8]时, 时间复杂度为 $O(\varepsilon^{-8})$, 空间复杂度为 $O(\varepsilon^{-4})$, 它们只依赖于逼近率 ε 而与训练集大小 ℓ 无关[1].

作为 CVM 的改进, BVM(Ball Vector Machine) 仅寻找给定点集的一个具有半径 $r=\sqrt{\kappa}$ 的闭球, 而非最小闭球. 每次迭代只需计算球心, 而不用计算半径, 而且 $\boldsymbol{c}_t$ 一定是 S_{t+1} 内元素的线性组合. 标准 BVM 学习算法至多在第 $2/\varepsilon$ 次迭代时停止, 时间复杂度为 $O(\varepsilon^{-4})$, 空间复杂度为 $O(\varepsilon^{-2})$[3].

3 RSCVM: 基于粗糙集的多类 CVM

假设有一个已经训练完毕的二类 SVM, $f(\boldsymbol{x})=\langle\boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b$, 现在来定义这个 SVM 的上近似和下近似.

3.1 二类线性可分情形

假设在核诱导特征空间(kernel-induced feature space)中数据线性可分. 因为在实际应用中 SVM 总是以对偶变量和支持向量的对偶形式被存储的, $\varphi(\boldsymbol{x})$ 在训练和测试中都不会出现, 这意味着我们在分析时总是可以提升核诱导特征空间的维数从而提高 SVM 的 VC 维, 使得该条件满足. 线性可分时的 SVM 寻找最大分类边界, 在两个边界超平面之间没有训练数据. 我们可以把 SVM 分类器看做一个数据描述的工具, 把 SVM 的边界(margin)想象成描述两个数据类的边界(boundary)[14]. 这使得我们可以用如下方法构造粗糙集的上下近似. 对于分类函数 $f(\boldsymbol{x})=\langle\boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b$ 中的常量, 选择 b_1, 使得训练集的所有数据都能正确分类并且存在一个目标类的样本点位于分离超平面上; 选择 b_2, 使得训练集的所有数据都能正确分类并且存在一个对照类的样本点位于分离超平面上[2]. 形式化的定义是:

$$\begin{aligned}&b_1: y(\langle\boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b_1)\geqslant 0,\ \forall(\varphi(\boldsymbol{x}),y)\in S;\\ &\quad\exists(\varphi(\boldsymbol{x})+1)\in S,\langle\boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b_1=0,\\ &b_2: y(\langle\boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b_2)\geqslant 0,\ \forall(\varphi(\boldsymbol{x}),y)\in S;\\ &\quad\exists(\varphi(\boldsymbol{x}),-1)\in S,\langle\boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b_2=0\end{aligned}\tag{7}$$

很容易看出, b_1 和 b_2 对应着 SVM 的边界 $b \pm \|w\|^{-2}$. 常量 b_1 的作用是让分类函数在核诱导特征空间中对应的超平面向着目标类的边界移动, 而 b_2 的作用则是让超平面向着对照类的方向移动. 在测试阶段, 如果 $\langle w, \varphi(x)\rangle + b_1 \geqslant 0$, 我们说 x 确定的属于目标类, 输出 $y = +1$; 如果 $\langle w, \varphi(x)\rangle + b_2 \leqslant 0$, 我们说 x 确定的属于对照类, 输出 $y = -1$; 否则 x 的分类不确定[2].

3.2 二类非线性可分情形

为了继续分析, 我们把式(7)改写成:

$$b_1 = \arg\min_b \langle w, \varphi(x)\rangle + b \geqslant 0,\ \forall(\varphi(x), +1) \in S;$$
$$b_2 = \arg\max_b \langle w, \varphi(x)\rangle + b \leqslant 0,\ \forall(\varphi(x), -1) \in S. \tag{8}$$

当数据在核诱导特征空间中线性可分时, 式(8)与式(7)等价; 当数据非线性可分时, 式(7)中定义的 b_1, b_2 不存在. 所以定义式(8)与式(7)等价的条件是在式(8)中 $b_1 \leqslant b_2$, 此时数据线性可分.

如果训练集在选定的 $\varphi(x)$ 下不是线性可分的, 我们给出如下定义:

$$u = \arg\max_b \langle w, \varphi(x)\rangle + b \geqslant 0,\ \forall(\varphi(x), -1) \in S;$$
$$v = \arg\min_b \langle w, \varphi(x)\rangle + b \geqslant 0,\ \forall(\varphi(x), +1) \in S;$$
$$b_1 = \min(u, v), \quad b_2 = \max(u, v). \tag{9}$$

式(9)就是在式(8)的基础上强制要求 $b_1 \leqslant b_2$ 得到的.

我们指出, 通过定义式(9), 研究非线性可分情形是有意义的. 虽然一个核函数 K 可以与多个特征映射 $\varphi(x)$ 对应, 但核函数的 VC 维等于维数最低的特征空间的维数加 1(如果 VC 维有限), 所以在核函数固定的情况下, 变化 $\varphi(x)$ 不会影响分类器的性质; 其次, 与高斯径向基核不同(即使高斯核也可选择径向基函数的宽度使得 VC 维有限), 多项式核没有无限的支持集, 所以 VC 维总是有限的, 意味着不能为了要求训练集在核诱导特征空间中线性可分而无限提高多项式核的次数, 否则很容易造成过拟合.

通过 $y_i(\langle w, \varphi(x)\rangle + b) \geqslant 1 - \xi_i$ 引入松弛变量 ξ_i, 令 $\rho_1 = b - \|w\|^{-2}$, $\rho_2 = b + \|w\|^{-2}$ 此时 b_1, b_2 和 ρ_1, ρ_2 是不同的. 因为数据在核诱导特征空间中非线性可分, 所以在目标类和对照类里必然分别存在至少一个 (x_i, y_i) 使得 $\xi_i > 1$. 如果训练集所有的样本对应的松弛变量 $\xi_i < 2$, 那么 $\rho_1 < b_1 < b_2 < \rho_2$; 如果目标类和对照类分别存在至少一个 (x_i, y_i) 使得 $\xi_i \geqslant 2$, 那么 $b_1 \leqslant \rho_1 < \rho_2 \leqslant b_2$; 如果只在目标类(对照类)中存在一个 (x_i, y_i) 使得 $\xi_i \geqslant 2$, 那么 $b_1 \leqslant \rho_1 < b_2 < \rho_2$ ($\rho_1 < b_1 < \rho_2 \leqslant b_2$).

与式(9)相对应的目标类下近似 $\underline{A}^+$, 对照类下近似 $\underline{A}^-$, 以及目标类上近似 $\overline{A}^+$, 对照类上近似 $\overline{A}^-$ 可定义如下:

$$\begin{aligned}
\underline{A}^+ &= \left\{ x \mid x \in \mathbf{R}^d, \langle w, \varphi(x)\rangle + b_1 \geqslant 0 \right\},\\
\overline{A}^+ &= \left\{ x \mid x \in \mathbf{R}^d, \langle w, \varphi(x)\rangle + b_2 \geqslant 0 \right\},\\
\underline{A}^- &= \left\{ x \mid x \in \mathbf{R}^d, \langle w, \varphi(x)\rangle + b_2 \leqslant 0 \right\},\\
\overline{A}^- &= \left\{ x \mid x \in \mathbf{R}^d, \langle w, \varphi(x)\rangle + b_1 \leqslant 0 \right\}.
\end{aligned} \tag{10}$$

3.3 扩展至多类情形

下面考虑多类情形. 事实上, 基于式(9)中定义的常量 b_1, b_2 把定义式(10)扩展到多类情形有许多种方法, 我们在这里给出一种. 设训练集为

$$S_{train} = \left\{ (x_i, y_i) \mid x_i \in \mathbf{R}^d, y_i \in Y \right\}_{i=1}^{\ell}$$

其中 $Y = \left\{ l_i \right\}_{i=1}^{n}$ 为类标号的集合. 令 $S^i = \{(x, y) \mid (x, y) \in S_{train}, y = l_i\}$, 不失一般性, 我们假设

$$\#S^i \geqslant \#S^j,\ \forall i < j \tag{11}$$

对某类来说, 其样本点个数不少于排在它之后的类. 设对 $1 \leqslant i < n$, cvm_i 是以 S^i 作为目标类, 以所有编号在 i 之后类的并集作为对照类训练得到的 CVM. 形式化的定义为 cvm_i 的训练集为 $S^i_{train} = \{(x, +1) \mid (x, y) \in S^i\} \cup \{(x, -1) \mid \exists j > i, (x, y) \in S^j\}$. 用 cvm_i 的 w_i 代入式(9)可得 b^i_1, b^i_2, 则第 i 类的上下近似可定义为

$$\begin{aligned}
\underline{A}(l_i) &= \left\{ x \mid x \in \mathbf{R}^d, \langle w_i, \varphi(x)\rangle + b^i_1 \geqslant 0 \right\} - \bigcup_{j<i} \underline{A}(l_j),\\
\overline{A}(l_i) &= \left\{ x \mid x \in \mathbf{R}^d, \langle w_i, \varphi(x)\rangle + b^i_2 \geqslant 0 \right\} - \bigcup_{j<i} \underline{A}(l_j).
\end{aligned} \tag{12}$$

当 $i = n$ 时, 令 $S^n_{train} = \{(x, -y) \mid (x, y) \in S^{n-1}_{train}\}$, 则这个解可以直接从 cvm_{n-1} 得到, 故只需训练 $n-1$ 个 CVM, 而传统的 1-v-r 方法要训练 n 个. 图 2 描述了 RSCVM 的训练算法.

算法 2: RSCVM Training Algorithm
1. 对训练集重排序, 使之满足式(11).
2. 对 $i = 1, 2, \cdots, n-1$, 生成 S^i_{train}, 使用算法 1 得到 cvm_i, 求解满足式(9)的 b^i_1 b^i_2.
3. 从 cvm_{n-1} 中得到 b^n_1, b^n_2.

图 2　RSCVM 训练算法

该方法的一个显著特点是训练过程中参加训练的类别逐渐减少, 训练集中的样本也随之减少. 因为类别已经按照它包含的样本点数量多少进行了排序, 第一步完成时样本最多的类别将不再参加今后的训练, 第二步完成时样本次多的类别也将不再参加今后的训练, 依此类推. 考虑最差的情况, 即每个类别都包含相同个数的样本, 此时第一次训练集的大小为 ℓ, 第二次为 $\ell(1 - \frac{1}{n})$, 第三次为 $\ell(1 - \frac{2}{n})$, 依此类推, 最终训练集的总大小为 $\ell(n+1)/2$, 而传统的 1-v-r 方法的训练集总大小为 ℓn, 传统的 1-v-1 方法的训练集总大小为 $2\ell n$.

RSCVM 的测试算法不同于 1-v-1 的投票和 1-v-r 的取最大值, 定义

$$f_1^i(\boldsymbol{x}) = \langle \boldsymbol{w}_i, \varphi(\boldsymbol{x}) \rangle + b_1^i, \quad (13)$$
$$f_2^i(\boldsymbol{x}) = \langle \boldsymbol{w}_i, \varphi(\boldsymbol{x}) \rangle + b_2^i$$

RSCVM 的测试算法为

算法 3: RSCVM Testing Algorithm
1. 置 $i = 1$.
2. 如果 $f_1^i(\boldsymbol{x}) \geqslant 0 (f_2^i(\boldsymbol{x}) > 0)$, 则分类为 l_i, 算法停止.
3. 如果 $i = n$, 则分类失败, 算法停止.
4. $i = i + 1$ 并转至 2.

图 3 RSCVM 测试算法

如果 l_{n-1} 和 l_n 线性可分, 则测试时较容易出现失败. 另外, 使用 $f_1^i(\boldsymbol{x})$ 比使用 $f_2^i(\boldsymbol{x})$ 更容易出现失败. 当使用 $f_1^i(\boldsymbol{x})$ 失败时, 如果 $\boldsymbol{x}$ 在某个类的上近似之内, 则可以分类为 $\boldsymbol{x}$ 在其上近似内的所有类中先验概率最大的那个类, 即 $l_i: f_2^i(\boldsymbol{x}) \geqslant 0; f_2^i(\boldsymbol{x}) < 0, \forall j < i$. 事实上, 简单的使用 $f_1^i(\boldsymbol{x})$ 或者 $f_2^i(\boldsymbol{x})$ 都会导致精度的下降, 原因是这两个函数都不是最优的分离超平面, 但它们的联合表示了有关分离超平面的粗糙信息, 所以仅使用其中之一会导致丢失一部分信息.

4 实验

4. 实验方法

实验中用到了三个分类器, 其中 LibSVM[9] 来自 http://www.csie.ntu.edu.tw/~cjlin/libsvm/[10], CVM 和 BVM 来自 http://www.cs.ust.hk/~ivor/cvm.html[11].

4 个数据集来自 http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html. 具体见表 1.

表 1 实验数据集

data sets	# classes	# attributes	# training set	# testing set
satimage	6	36	4435	2000
letter	26	16	10500	5000
sector	105	55197	6412	3207
acoustic	3	50	78823	19705

LibSVM 使用默认的 C-SVC, RSCVM 和 RSBVM 使用 f_2^i 作为判定函数. 所有分类器的核函数全部选用径向基函数核.

实验平台为 Intel 奔腾 E2160(双核 1.8G), 1G 内存, 运行环境为 Window XP 和 VS2008.

实验的主要目的: 记录精确度, 支持向量个数, 训练时间和测试时间. 通过对比找出 RSCVM 方法的优点和不足.

4.2 实验结果

表 2 至表 5 中, accuracy 代表精确度, # sv 代表支持向量个数, training 和 testing 分别代表训练时间和测试时间, 单位为秒.

我们在 satimage 数据集上收集了 5 组数据, 其中代表性的两组数据见表 2.

在 letter 数据集上的实验数据见表 3.

表 2 satimage 数据集的实验结果

	LibSVM		CVM($\varepsilon = 10^{-4}$)			BVM($\varepsilon = 10^{-4}$)			CVM($\varepsilon = 10^{-6}$)			BVM($\varepsilon = 10^{-6}$)		
	1-v-1	1-v-r	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs
accuracy(%)	90.65	90.75	85.5	84.4	75.3	85.1	85.8	83.9	90.75	90.55	84.2	90.65	90.5	85.2
# sv	1128	1295	789	818	628	1549	1612	1302	1436	1625	1425	1652	1916	1641
training(s)	1.59	7.91	0.88	1.39	1.43	2.20	4.11	3.32	2.31	8.49	5.28	7.69	11.69	8.48
testing(s)	0.67	0.84	0.55	0.58	0.46	0.95	1.05	0.81	0.85	1.09	0.91	1.00	1.31	1.04

表 3 letter 数据集的实验结果

	LibSVM		CVM($\varepsilon = 10^{-4}$)			BVM($\varepsilon = 10^{-4}$)			CVM($\varepsilon = 10^{-6}$)			BVM($\varepsilon = 10^{-6}$)		
	1-v-1	1-v-r	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs
accuracy(%)	96.58	96.18	96.78	95.38	91.82	96.38	95.7	93.04	96.86	96.0	91.9	96.38	95.9	92.66
# sv	4936	4114	5069	3655	3206	6112	5807	4882	5321	4653	3789	6125	6122	5072
training(s)	8.71	39.19	9.31	7.42	12.78	16.34	23.29	27.46	11.13	23.23	20.46	29.72	37.14	35.75
testing(s)	6.24	5.92	6.43	5.30	3.46	7.81	8.20	5.89	6.72	6.81	4.53	7.69	8.63	6.09

表 4 sector 数据集的实验结果

	LibSVM		CVM($\varepsilon = 10^{-3}$)			BVM($\varepsilon = 10^{-3}$)		
	1-v-1	1-v-r	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs
accuracy(%)	89.17	93.79	89.00	80.38	77.20	86.62	87.74	81.70
# sv	6270	6401	6215	6314	6134	6344	6222	6160
training(s)	229.86	3049.63	220.80	198.09	317.03	149.94	146.09	225.26
testing(s)	62.16	55.78	59.45	54.83	50.78	60.03	55.13	50.37

表 5 acoustic 数据集的实验结果

	LibSVM	CVM($\varepsilon = 10^{-5}$)			BVM($\varepsilon = 10^{-5}$)		
	1-v-1	1-v-1	1-v-r	rs	1-v-1	1-v-r	rs
accuracy(%)	71.37	61.06	56.15	50.04	60.85	60.88	50.10
# sv	37474	6217	6602	4311	9945	10713	6968
training(s)	13419	231.2	278.74	311.9	664.4	935.8	765.1
testing(s)	226.89	40.94	53.95	30.07	84.97	86.40	52.98

在 sector 数据集上的实验数据见表 4. 通过减小 ε 可以提高精度, 但需要较长时间.

在 acoustic 数据集上的实验数据见表 5. 这里没有 1-v-r 方法的 SVM 是因为其训练时间太长. 在 ε 较大时 CVM 和 BVM 的精度与 SVM 相差较大, 但是它们的训练时间远小于 SVM. 可以说, 这是由数据集本身的性质造成的.

4.3　对于实验的分析和讨论

从上面的数据我们看到, 通常 SVM 的精度要高于 CVM 与 BVM, 其代价是当训练集尺寸较大时要付出数倍的计算时间开销. 与 CVM 和 BVM 的 1-v-1 和 1-v-r 方法相比, RSCVM 通常产生更少的支持向量, 具有更短的测试时间, 但其精度与 1-v-1 和 1-v-r 方法相比较差. 从现有数据来看, RSCVM 方法在较大的训练集上的表现不如在中小规模的训练集上好.

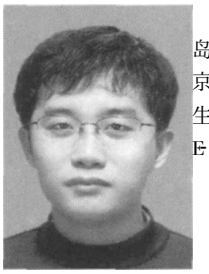
传统的非粗糙集方法对于噪声是不敏感的, 但是基于粗糙集的 RSCVM 对于数据中的噪声较为敏感. 一种解决方案是使用迭代计算上下近似, 但是这样效率很低[2]. 另一方面, 按照式(9) 计算上下近似需要进行 $\#\ SV^i . \#\ SV^i_{train}$. 次核函数估值, 由于支持向量的稀疏性, 使得 $\#\ SV^i$ 远小于 $\#\ SV^i_{train}$. 实验表明, 在支持向量较少时, 求解上下近似所用的时间甚至远远超过了训练 CVM 本身. 所以如何高效的估算上下近似是一个悬而未决的问题, 我们认为快速准确的估算上下近似是目前基于粗糙集理论的支持向量学习器的相关研究中最重要的问题.

最后, 如果简单的使用 f^i_1 或 f^i_2, 其精度不会很高. f^i_1 倾向于将边界上的点分入先验概率小的类, 而 f^i_2 倾向于将边界上的点分入先验概率大的类. 实验表明, f^i_1 的效果总是不如 f^i_2 的. 为了提高预测精度, 在进行测试时可以使用更加复杂的方法, 比如按照先验概率加权的投票, 或者使用领域知识, 这是和具体问题相关的, 此时上下近似提供了比非粗糙集模型更加丰富的信息. 如何更好的使用建立的粗糙集模型, 也是一个重要的问题.

5　结束语

本文提出了一种适用于多类分类问题的新方法. 本文首先使用粗糙集的观点理解最大边界分类器, 然后给出了一个用于二类支持向量机(包括 SVM, CVM 和 BVM) 的上下近似的一致定义, 最后通过在核诱导特征空间中对训练集进行空间划分, 给出多类的上下近似定义, 提出基于粗糙集的多类分类 CVM, 即 RSCVM 方法. 该方法是粗糙集理论与最新的 SVM 求解方法 CVM 结合的产物, 在理论上具有相当程度的创新; 通过对比实验, 发现该方法有产生更少的支持向量等优点, 虽存在精度下降的缺点, 仍具有一定的现实意义. 这种结合是一个新的尝试, 我们认为 RSCVM 方法值得进一步研究, 而且它的性能可以改进.

作者简介:



牛　罡　男, 1984 年 6 月出生于河北秦皇岛. 2007 年毕业于东南大学数学系, 同年进入南京大学计算机科学与技术系, 现为在读硕士研究生, 从事统计机器学习、粗糙集方向的有关研究. E-mail: niugang@ai.nju.edu.cn

商　琳　女, 1973 年 7 月出生于甘肃兰州. 现为南京大学计算机科学与技术系副教授, 从事数据挖掘、机器学习、粗糙集等方向的研究. E-mail: shanglin@nju.edu.cn

参考文献:

[1] I W Tsang, J T Kwok, P M Cheung. Core vector machines: Fast SVM training on very large data sets[J]. Journal of Machine Learning Research, 2005, 6: 363–392.

[2] P Lingras, C Butz. Rough set based 1-v-1 and 1-v-r approaches to support vector machine multi-classification[J]. Information Sciences, 2007, 177(18): 3782–3798.

[3] I W Tsang, A Kocsor, J T Kwok. Simpler core vector machines with enclosing balls[A]. Proceedings of the 24th International Conference on Machine Learning[C]. New York, USA: ACM, 2007. 911–918.

[4] J C Platt. Fast training of support vector machines using sequential minimal optimization[A]. Advances in Kernel Methods Support Vector Learning[C]. Cambridge, USA: MIT Press, 1999. 185–208.

[5] M Bădoiu, K L Clarkson. Optimal core sets for balls[A]. Proceedings of DIMACS Workshop on Computational Geometry[C]. Piscataway, USA, 2002.

[6] V N Vapnik. Statistical Learning Theory[M]. USA & Canada: John Wiley & Sons, 1998.

[7] J C Platt, N Cristianini, J Shawe-Taylor. Large margin DAG's for multiclass classification[A]. Advances in Neural Information Processing Systems[C]. Cambridge, USA: MIT Press, 2000, vol. 12, 547–553.

(下转第 49 页)

ing Research. 2004, 5(11): 1435- 1455.

[16] Schölkopf B, Smola A J. Learning with Kernels: Support Vector Machines: Regularization, Optimization and Beyond[M]. Cambridge, Massachusetts: MIT Press, 2002.

[17] Pólik I, Terlaky T, et al. SeDuMi: a package for conic optimization[EB/OL]. http://imre.polik.googlepages.com/SeDuMi_IMA_poster.pdf

[18] Labit Y, Peaucelle D, et al. SeDuMi Interface 1.02: a tool for solving LMI problems with SeDuMi[A]. in Proc IEEE CACSD[C]. Glasgow, UK, 2002. 272- 277.

[19] Antoniou A, Lu W S. Practical optimization: algorithms and engineering applications[M]. Springer, 2007.

作者简介:

邱德红　男, 1971 年生于湖南永州. 博士, 副教授, 机器学习专业委员会委员. 研究方向为机器学习与数据挖掘、软件智能、新媒体技术.
E-mail: qiudehong@163.com

潘昕昕　男, 1981 年生于湖北省沙市. 硕士研究生, 研究方向为数据挖掘与软件智能.

(上接第 59 页)

[8] A Smola, B Schölkopf. Sparse greedy matrix approximation for machine learning[A]. Proceedings of the 17th International Conference on Machine Learning[C]. San Francisco, USA: Morgan Kaufmann, 2000. 911- 918.

[9] R E Fan, P H. Chen, C J Lin. Working set selection using the second order information for training SVM[J]. Journal of Machine Learning Research, 2005, 6: 1889- 1918.

[10] C C Chang, C J Lin. LIBSVM -- A Library for Support Vector Machines(version 2.85)[OL]. http://www.csie.ntu.edu.tw/cjlin/libsvm/, 2007-12-26.

[11] I W Tsang. LibCVM Toolkit(version 2.02 alpha)[OL]. http://www.cs.ust.hk/ivor/cvm.html, 2008-03-05.

[12] F Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological Review, 1958, 65(6): 386- 408.

[13] C Cortes, V Vapnik. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273- 297.

[14] P Lingras, C Butz. Interval set classifiers using support vector machines[A]. Proceedings of International Conference of the North American Fuzzy Information Processing Society[C]. Banff, Canada: IEEE, 2004, No. 23, 707- 710.