文章编号: 1001-0920 (2013) 07-1065-08

基于最小包含球的非静态大数据集的快速分类算法

史荧中[1,2], 王士同[1], 王 骏[1], 邓赵红[1]

(1. 江南大学 数字媒体学院，江苏 无锡 214122；2. 无锡职业技术学院 物联网技术学院，江苏 无锡 214121)

摘 要: 对于小规模的非静态数据, 最近提出的时间自适应支持向量机 (TA-SVM) 方法表现出良好的性能, 它从兼顾局部优化和全局优化的角度同时求解多个子分类器的特性. 但对于大数据集, 较高的计算代价限制了它的实用性. 针对此不足, 结合核心向量机 (CVM) 理论提出了针对非静态大数据集的新颖分类方法, 即基于中心约束最小包含球 (CCMEB) 的 TA-CVM, 简称 CCTA-CVM. 该方法具有渐近线性时间复杂度的优点, 同时继承了 TA-SVM 的良好性能. 最后通过实验验证了所提出方法的有效性.

关键词: 数据漂移；非静态大数据集；最小包含球；线性时间复杂度

中图分类号: TP181 **文献标志码:** A

Fast classification for nonstationary large scale data sets using minimal enclosing ball

SHI Ying-zhong[1,2], *WANG Shi-tong*[1], *WANG Jun*[1], *DENG Zhao-hong*[1]

(1. School of Digital Media，Jiangnan University，Wuxi 214122，China；2. School of Internet of Things，Wuxi Institute of Technology，Wuxi 214121，China. Correspondent：SHI Ying-zhong，E-mail：shiyz@wxit.edu.cn)

Abstract: For small scale nonstationary data sets, the recently-proposed classifier time adaptive support vector machine(TA-SVM) exhibits its good performance with the distinctive characteristic of simultaneously solving several subclassifiers locally and globally. However, for large scale data sets, its high computational cost severely weakens its usefulness. In order to overcome this shortcoming, a novel classifier named center-constrained minimal enclosing ball(CCMEB) based time adaptive core vector machine(CCTA-CVM for brevity) for large nonstationary datasets is proposed by using core vector machine(CVM) theory. This classifier has the merit of asymptotic linear time complexity and inherits the good performance of TA-SVM. Experimental results show the effectiveness of the proposed classifier.

Key words: data drift; large nonstationary datasets; minimal enclosing ball; linear time complexity

0 引 言

在某些应用场合, 如大型零售超市、电子商务网站、电信领域服务商或者科研机构等, 每天都会产生大量的数据, 而且数据特征是随着时间而不断变化的, 数据的分类模型也会随着时间和地点的不同而变化. 针对这类不断变化的分类模型问题, Helmbold 等[1]于 1994 年提出了漂移概念 (drifting concepts), Widmer 等[2]于 1996 年使用了概念漂移 (concept drift), 二者略有不同. 前者主要针对缓慢变化型数据, 相近的研究有文献 [3-8]; 后者主要研究因数据特征突然变化而引起的模型变化问题, 相似的研究有文献 [6, 9-13]. 本文只针对缓慢变化型数据研究其分类模型的变化. 以往针对漂移概念所做的工作有些是从概率角度展开研究的[3,5], 更多的是基于“局部分类器”[2,4,9,14-16]的思路, 即从总体数据中依次选取一定区间 (称为滑动窗) 内的数据先分别求出各子分类器, 再进行某种组合. 它们的差异主要在于滑动窗的长度如何选取、滑动窗内各数据的权重和子分类器的集成等.

2011 年, Grinblat 等[17]利用支持向量机之间的耦合来求解非静态数据的分类问题. 他们使用时间自适应支持向量机 (TA-SVM) 方法求解渐变的子分类器, 其核心思想是求解一个子分类器序列, 子分类器不仅使相应时间段内的数据达到分类最优, 还从全局的角度考虑了与其他子分类器之间的关系. TA-SVM 在局部优化与全局优化之间保持了很好的平衡性, 因此在各种场景的对比实验中都优于以往的算法.

收稿日期: 2012-02-24；**修回日期:** 2012-10-17.

基金项目: 国家自然科学基金项目(60903100, 61170122)；江苏省自然科学基金项目(BK2011003).

作者简介: 史荧中(1970−), 男, 博士生, 从事模式识别、数据挖掘的研究；王士同(1964−), 男, 教授, 博士生导师, 从事模式识别、数据挖掘、模糊神经网络等研究.

参照文献[17], TA-SVM方法原始问题的对偶问题等价于在核空间上的另一个SVM的对偶, 故该方法的算法时间复杂度等同于常见的SVM的复杂度, 一般情况下为$O(n^3)$. 为了取得较好的计算效率, 亦可采用序列最小化[18](SMO)方法来求解SVM所对应的二次规划问题, 使其复杂度降为$O(n^{2.3})$. 但是, 即便如此, TA-SVM方法在面对大数据集时仍存在很大的局限性, 为此, Grinblat等[17]对于求解大数据集的问题, 给出了如下建议: 抛弃部分早期的数据, 只选用近期的数据来求解问题. 从本质上讲, 这时的TA-SVM方法又回到了类似于滑动窗算法的局部求解模式, 并且难以面对子数据集本身就是大数据集的问题. 如何找出一种新的方法, 既能保持TA-SVM具有良好的分类性能, 同时又适用于各种情形的大样本数据集, 正是本文的出发点. 本文首先给出时间自适应核心向量机(TA-CVM)方法, 该方法吸收了TA-SVM的优点, 综合考虑了子分类器的局部优化与全局优化; 然后, 结合Tsang等[19-20]提出的中心约束最小包含球(CCMEB)理论给出TA-CVM方法的快速求解算法CCTA-CVM, CCTA-CVM算法的最大优点是其渐近时间复杂度与样本容量N呈线性关系; 最后通过实验结果表明了所提出方法的有效性.

1 TA-SVM原理与方法

TA-SVM方法针对的是分类模型缓慢变化的二类分类问题. 设数据集$\{(x_i, y_i)|i = 1, 2, \cdots, n\}$中含有$n$个样本点, 由$m$个按时间顺序采集的子数据集$p_\mu$组成, $\mu \in \{1, 2, \cdots, m\}$. 第$\mu$个子数据集$p_\mu$中的数据点所对应的下标记为$\{i : \mu(i) = \mu\}$. P为$m \times n$矩阵, 用于标识第j个点是否属于第μ个子数据集, $P_{\mu j} = 1$当且仅当$j \in p_\mu$, 否则取值为0. 若每个子数据集都存在相应的子分类器, 第μ个子分类器记为(w_μ, b_μ), 则用$(w_{\mu(i)}, b_{\mu(i)})$表示第i个点所对应的子分类器.

1.1 TA-SVM基本原理

TA-SVM不仅考虑各个子分类器的优化, 而且考虑到随着时间的变化各子分类器的变化应该比较平稳, 因此将各子分类器之间的差异进行了约束. 其思想可由下式表示:

$$\min \sum_{\mu=1}^{m} \mathrm{Risk}(f_\mu) + \gamma \sum_{\mu=1}^{m-1} d(f_{\mu+1}, f_\mu). \tag{1}$$

其中: $\mathrm{Risk}(f_\mu)$表示第μ个子分类器f_μ的结构风险; $d(f_{\mu+1}, f_\mu) = \|w_\mu - w_v\|^2 + (b_\mu - b_v)^2$为相邻两个子分类器之间的差异, $\min \sum_{\mu=1}^{m-1} d(f_{\mu+1}, f_\mu)$为全局优化项; γ为对局部优化与全局优化进行权衡的因子. TA-SVM的基本原理如图1所示.

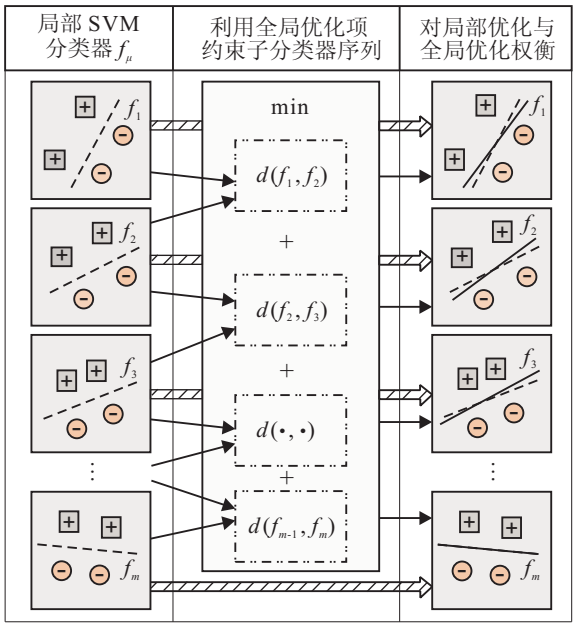


图 1 TA-SVM方法基本原理

图1中以由上至下的正方形框以及内部数据来模拟缓慢变化的场景. 左侧各虚线表示各个局部子分类器; 右侧各实线表示通过全局优化对局部优化进行权衡矫正后的子分类器序列; 中间实线框表示对子分类器序列进行约束处理, 以使序列的变化较平稳.

1.2 TA-SVM方法及其对偶

基于式(1)的思想, Grinblat等给出了TA-SVM方法的目标函数形式如下:

$$\min_{w_\mu, b_\mu} \frac{1}{2m} \sum_{\mu=1}^{m} \Big(\|w_\mu\|^2 + \frac{\gamma}{2} \sum_{v=1}^{m} Q_{\mu v} (\|w_\mu - w_v\|^2 + (b_\mu - b_v)^2) \Big) + C \sum_{i=1}^{n} \xi_i;$$

$$\text{s.t. } y_i (w_{\mu(i)}^{\mathrm{T}} \varphi(x_i) + b_{\mu(i)}) \geqslant \xi_i,\ \xi_i \geqslant 0. \tag{2}$$

其中: 矩阵Q为指示各子分类器间相关性的$m \times m$矩阵, $Q_{\mu\nu} = 1$当且仅当$|\mu - \nu| = 1$, 否则值为0; C为错分惩罚因子. 通过对式(2)进行推导[17], 可得到其对偶形式并简化如下:

$$\max_{\alpha} -\frac{1}{2} \alpha^{\mathrm{T}} G \alpha + \sum_i \alpha_i;$$

$$\text{s.t. } 0 \leqslant \alpha \leqslant C,\ \sum_i \alpha_i y_i = 0. \tag{3}$$

其中: $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)^{\mathrm{T}}$, α_i为拉格朗日算子, 而

$$G = (P^{\mathrm{T}} M^{-1} P) \otimes K \otimes Y + (P^{\mathrm{T}} (M - I/m)^{+} P) \otimes Y.$$

式中

$$M = [M_{\mu v}]_{n \times n},$$

$$M_{\mu v} = \begin{cases} \Big(1 + \gamma \sum\limits_k Q_{\mu k}\Big) \Big/ m, & \mu = v; \\ -\gamma Q_{\mu v}/m, & \mu \neq v. \end{cases}$$

$K=[K_{ij}]_{n\times n}, K_{ij}=\varphi^{\mathrm{T}}(x_i)\varphi(x_j), \varphi$ 为核函数. $Y=[Y_{ij}]_{n\times n}, Y_{ij}=y_i\cdot y_j$. 记号 $\otimes$ 表示两个矩阵之间的哈达马积, 即 $(K\otimes Y)_{ij}=K_{ij}\cdot Y_{ij}$. 矩阵 $M-I/m$ 为奇异矩阵, 其并不可逆, 因此用 $(M-I/m)^{+}$ 表示矩阵 $M-I/m$ 的伪逆.

2　TA-CVM 方法

本质上, TA-CVM 方法是对随时间而缓慢变化的数据集求解一个子分类器序列. 本文按文献 [19-20] 的方法对式 (2) 进行稍许改变, 并在考虑局部优化与全局优化的同时又引入了子分类器的分类间隔 ρ. 通过推导可以得到更简洁的对偶形式, 同时避免了 TA-SVM 方法中出现的伪逆问题. 下面给出具体的 TA-CVM 方法的目标函数:

$$\min_{w_\mu,b_\mu}\frac{1}{2m}\sum_{\mu=1}^{m}\Big(\|w_\mu\|^2+b_\mu^2+\frac{\gamma}{2}\sum_{v=1}^{m}Q_{\mu v}(\|w_\mu-w_v\|^2+(b_\mu-b_v)^2)\Big)-\rho+\frac{C}{2}\sum_{i=1}^{n}\xi_i^2;$$

$$\text{s.t. } y_i(w_{\mu(i)}^{\mathrm{T}}\varphi(x_i)+b_{\mu(i)})\geqslant\rho-\xi_i. \tag{4}$$

其相应的拉格朗日函数为

$$L=\frac{1}{2m}\sum_{\mu=1}^{m}\Big(\|w_\mu\|^2+b_\mu^2+\frac{\gamma}{2}\sum_{v=1}^{m}Q_{\mu v}(\|w_\mu-w_v\|^2+(b_\mu-b_v)^2)\Big)-\rho+\frac{C}{2}\sum_{i=1}^{n}\xi_i^2-\sum_{i=1}^{n}\alpha_i(y_i(w_{\mu(i)}^{\mathrm{T}}\varphi(x_i)+b_{\mu(i)})-\rho+\xi_i). \tag{5}$$

由 KKT 条件, L 取得极值时, 有

$$\frac{\partial L}{\partial\rho}=0,\ \frac{\partial L}{\partial\xi_i}=0,\ \frac{\partial L}{\partial w_u}=0,\ \frac{\partial L}{\partial b_u}=0;$$

$$\frac{\partial L}{\partial\rho}=0=-1+\sum_{i=1}^{n}\alpha_i\Rightarrow\sum_{i=1}^{n}\alpha_i=1;$$

$$\frac{\partial L}{\partial\xi_i}=0=C\xi_i-\alpha_i\Rightarrow\xi_i=\alpha_i/C.$$

记 I 为单位矩阵, 并记

$$W=\frac{1}{2m}\sum_{\mu=1}^{m}\|w_\mu\|^2+\frac{\gamma}{4m}\sum_{\mu=1}^{m}\sum_{v=1}^{m}Q_{\mu v}\|w_\mu-w_v\|^2-\sum_{i=1}^{n}\alpha_i y_i w_{\mu(i)}^{\mathrm{T}}\varphi(x_i),$$

$$B=\frac{1}{2m}\sum_{\mu=1}^{m}b_\mu^2+\frac{\gamma}{4m}\sum_{\mu=1}^{m}\sum_{v=1}^{m}Q_{\mu v}(b_\mu-b_v)^2-\sum_{i=1}^{n}\alpha_i y_i b_{\mu(i)}.$$

再将 ξ_i 和 α_i 的条件代入式 (5), 可以得到

$$L=W+B-\frac{1}{2}\alpha^{\mathrm{T}}(I/C)\alpha. \tag{6}$$

由

$$\frac{\partial L}{\partial w_\mu}=0=\frac{1}{m}w_\mu+\frac{\gamma}{m}\sum_{v}Q_{\mu v}(w_\mu-w_v)-\sum_{j\in p_\mu}\alpha_j y_j\varphi(x_j),$$

可得

$$\frac{1}{m}\Big(w_\mu+\gamma\sum_{v}Q_{\mu v}(w_\mu-w_v)\Big)=\sum_{j\in p_\mu}\alpha_j y_j\varphi(x_j).$$

这里使用式 (3) 中已定义的可逆矩阵 M, 则有

$$w_\mu=\sum_{j}M_{\mu\mu(j)}^{-1}\alpha_j y_j\varphi(x_j).$$

因为 $(P^{\mathrm{T}}M^{-2}P)_{ij}=\sum_{\mu}M_{\mu\mu(i)}^{-1}M_{\mu\mu(j)}^{-1}$, 所以

$$\sum_{\mu=1}^{m}\|w_\mu\|^2=\sum_{\mu}\sum_{ij}M_{\mu\mu(i)}^{-1}M_{\mu\mu(j)}^{-1}\alpha_i\alpha_j y_i y_j\varphi^{\mathrm{T}}(x_i)\varphi(x_j)=\sum_{ij}(P^{\mathrm{T}}M^{-2}P)_{ij}\alpha_i\alpha_j y_i y_j\varphi^{\mathrm{T}}(x_i)\varphi(x_j)=\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-2}P)\otimes K\otimes Y)\alpha, \tag{7}$$

$$\sum_{\mu=1}^{m}\sum_{v=1}^{m}Q_{\mu v}\|w_\mu-w_v\|^2=\sum_{\mu v}Q_{\mu v}\sum_{ij}(M_{\mu\mu(i)}^{-1}-M_{vv(i)}^{-1})\times(M_{\mu\mu(j)}^{-1}-M_{vvj}^{-1})\alpha_i\alpha_j y_i y_j\varphi^{\mathrm{T}}(x_i)\varphi(x_j)=\sum_{ij}2\Big(\sum_{\mu}M_{\mu\mu(i)}^{-1}M_{\mu\mu(j)}^{-1}D_{\mu\mu}-\sum_{\mu v}M_{\mu\mu(i)}^{-1}M_{v\mu(j)}^{-1}Q_{\mu v}\Big)\alpha_i\alpha_j y_i y_j\varphi^{\mathrm{T}}(x_i)\varphi(x_j)=2\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}(D-Q)M^{-1}P)\otimes K\otimes Y)\alpha. \tag{8}$$

其中对角矩阵 D 为

$$D_{\mu v}=\begin{cases}\sum\limits_{k}Q_{\mu k},\ \mu=v;\\ 0,\ \mu\neq v.\end{cases}$$

将式(7) 和 (8) 代入 (6), 可得到

$$L=\frac{1}{2m}\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-2}P)\otimes K\otimes Y)\alpha+\frac{\gamma}{2m}\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}(D-Q)M^{-1}P)\otimes K\otimes Y)\alpha-\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}P)\otimes K\otimes Y)\alpha+B-\frac{1}{2}\alpha^{\mathrm{T}}(I/C)\alpha=-\frac{1}{2}\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}P)\otimes K\otimes Y)\alpha+B-\frac{1}{2}\alpha^{\mathrm{T}}(I/C)\alpha. \tag{9}$$

下面对 b_μ 进行求解:

$$\frac{\partial L}{\partial b_\mu}=0=\frac{1}{m}b_\mu+\frac{\gamma}{m}\sum_{v}Q_{\mu v}(b_\mu-b_v)-\sum_{j\in p_\mu}\alpha_j y_j.$$

类似于 w_μ 的推导, 可得

$$b_\mu = \sum_j M^{-1}_{\mu\mu(j)}\alpha_j y_j,$$

$$\sum_{\mu=1}^{m} b_\mu^2 = \alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-2}P)\otimes Y)\alpha; \tag{10}$$

$$\sum_{\mu=1}^{m}\sum_{v=1}^{m} Q_{\mu v}(b_\mu - b_v)^2 =$$

$$2\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}(D-Q)M^{-1}P)\otimes Y)\alpha. \tag{11}$$

将式 (10) 和 (11) 代入 (9), 得

$$L = -\frac{1}{2}\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}P)\otimes K\otimes Y)\alpha - \frac{1}{2}\alpha^{\mathrm{T}}((P^{\mathrm{T}}M^{-1}P)\otimes Y)\alpha - \frac{1}{2}\alpha^{\mathrm{T}}(I/C)\alpha. \tag{12}$$

由式 (12) 以及推导过程中 α 的条件, 即可得到

$$\max_{\alpha}\ -\frac{1}{2}\alpha^{\mathrm{T}}H\alpha;$$
$$\text{s.t. } \alpha \geqslant 0,\ \alpha^{\mathrm{T}}\mathbf{1} = 1. \tag{13}$$

其中 $H = (P^{\mathrm{T}}M^{-1}P)\otimes(K + \mathbf{1}_{n\times n})\otimes Y + I/C$.

3 MEB 问题与 TA-CVM 方法的快速算法

3.1 最小包含球 (MEB) 问题

核化最小包含球 (MEB) 问题可以表示为二次规划问题的矩阵形式, 即

$$\max_{\alpha}\ \alpha^{\mathrm{T}}\mathrm{diag}(K) - \alpha^{\mathrm{T}}K\alpha;$$
$$\text{s.t. } \alpha \geqslant 0,\ \alpha^{\mathrm{T}}\mathbf{1} = 1. \tag{14}$$

其中: $K_{n\times n} = [k(x_i, x_j)] = [\varphi^{\mathrm{T}}(x_i)\varphi(x_j)]$ 为核矩阵, $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]^{\mathrm{T}}$ 为 Lagrange 乘子. 通过求解式 (14), 即可得到 MEB 的中心 c 和半径 R, 即

$$c = \sum_{i=1}^{n}\alpha_i\phi(x_i),$$
$$R = \sqrt{\alpha^{\mathrm{T}}\mathrm{diag}(K) - \alpha^{\mathrm{T}}K\alpha}. \tag{15}$$

考察核函数中的特殊情形, 即核矩阵对角元素恒为常数

$$k(x_i, x_i) = k,\ \forall i. \tag{16}$$

Tsang 等[20]指出, 形如式 (14) 且满足式 (16) 的二次规划问题均等价于求解 MEB 问题. 他们进一步指出, 硬边界 SVDD问题、软边界单类及二类的 L_2-SVM 问题都可以转化为核化 MEB 问题. 在此基础上, 利用最小包含球理论[21]的核心集 (Core-set) 技术开发了核心向量机 (CVM) 算法[20], CVM 算法对于处理大规模数据集发挥了惊人的效率.

Tsang 等[19]对 CVM 进行了扩展, 提出了通用核心向量机 (GCVM). 对于形如式 (14) 的二次规划问题, 即使不满足式 (16), 也可以使用核心集方法进行快速求解, 称为中心约束最小包含球算法 (CCMEB 算法).

在 CCMEB 中, 给核空间中任意样本点 $\phi(x_i)$ 附加一维新特征 $\delta_i \in R$, 形成新特征空间中的新样本

$$\tilde{\phi}(x_i) = \begin{bmatrix}\phi(x_i)\\ \delta_i\end{bmatrix},$$

使其满足 MEB 问题的式 (16) 条件, 然后求解在新特征空间中的最小包含球. 对该最小包含球增加一个约束条件, 即最小包含球中增加的特征维对应的中心固定在原点, 亦即 CCMEB 的中心为 $\begin{bmatrix}c\\0\end{bmatrix}$, 这里 c 为原特征空间中的最小包含球球心. CCMEB 问题的标准形式如下:

$$\max\ \alpha^{\mathrm{T}}(\mathrm{diag}(K) + \Delta) - \alpha^{\mathrm{T}}K\alpha;$$
$$\text{s.t. } \alpha \geqslant 0,\ \alpha^{\mathrm{T}}\mathbf{1} = 1. \tag{17}$$

其中: $\Delta = (\delta_1^2, \delta_2^2, \cdots, \delta_n^2)^{\mathrm{T}} \geqslant 0$ 为用户定义的值, 目的是使式 (17) 目标函数中 α 的一次项系数为常数. 为了保证 Δ 的非负性, 在式 (17) 中增加了新项 $-\eta\alpha^{\mathrm{T}}\mathbf{1}, \eta \in R$ 为常数. 因为 $-\eta\alpha^{\mathrm{T}}\mathbf{1} = -\eta$, 增加一个常量不改变问题的求解, 所以式 (17) 可以改写成

$$\max\ \alpha^{\mathrm{T}}(\mathrm{diag}(K) + \Delta - \eta\mathbf{1}) - \alpha^{\mathrm{T}}K\alpha;$$
$$\text{s.t. } \alpha \geqslant 0,\ \alpha^{\mathrm{T}}\mathbf{1} = 1. \tag{18}$$

3.2 TA-CVM 的快速算法

3.2.1 CCTA-CVM 算法描述

求解 TA-CVM 问题的复杂度, 包括核矩阵 H 的计算以及式 (13) 的求解. 式 (13) 是一个普通的二次规划问题, 其求解时间复杂度为 $O(n^{2.3}) \sim O(n^3)$, 对于大数据集而言应是相当可观的计算开销. 幸运的是, 仔细观察式 (13), 它可转化为形如式 (18) 的 CCMEB 问题. 因此, 原本用二次规划形式求解 TA-CVM 方法便可转化为对 CCMEB 问题的求解, 且 CCMEB 问题的求解过程中只需要计算核心集之外的点到核心集的距离, 而无需计算所有点之间的相互距离, 因此不必预先计算核矩阵 H, 从而使得 TA-CVM 方法能用 CCMEB 的快速算法求解. 将式 (13) 等价地改写为

$$\max\ \alpha^{\mathrm{T}}(\mathrm{diag}H) + \Delta - \eta\mathbf{1}) - \alpha^{\mathrm{T}}H\alpha;$$
$$\text{s.t. } \alpha \geqslant 0,\ \alpha^{\mathrm{T}}\mathbf{1} = 1. \tag{19}$$

这是一个 CCMEB 问题, 其中 $\Delta = -\mathrm{diag}(H) + \eta\mathbf{1}$, 通过调节常数 η 的值使 $\Delta \geqslant 0$. 因为求解式 (19) 时 Δ 和 η 需要确定, 所以需预先计算核矩阵 H 的对角线元素的值. 但是, 如果采用高斯核, 则由 H 的形式可知, 只需计算矩阵 M 的逆即可预先估算 Δ 和 η 的值.

下面给出 TA-CVM 方法的快速求解算法, 记为 CCTA-CVM 算法.

算法 1 CCTA-CVM 算法.

输入: 大数据集 S, 核心集逼近精度 ε 以及 η 和 Δ 等参数;

输出: 核心集 S_t, 权重系数 α.

Step 1: 初始化核心集 S_0, 最小包含球中心 c, 半径 R, 迭代次数 $t=1$;

Step 2: 若所有点都被球 $B(c_0,(1+\varepsilon)R_t)$ 包围, 则转 Setp 7;

Setp 3: 找到离 c_t 最远的点 x 加入核心集, 使得 $S_{t+1}=S_t\bigcup x$;

Step 4: 求解新的核心集 S_t, 得到球心 c_{t+1} 和半径 R_{t+1}, 权重系数 α;

Step 5: 计算新的球心到其他各点的距离;

Step 6: $t=t+1$, 转 Step 2;

Setp 7: 终止训练, 返回核心 S_t, 权重系数 α.

CCTA-CVM 实现说明:

1) 在 Step 1 中, 虽然可以任选 $x\in S$ 来初始化核心集 $S_0=\{x\}$, 但好的初始化能提高算法的性能[22-23]. 可以从大数据集 S 中选取子集 $S_{\rm sub}$, 任取 $S_{\rm sub}$ 中的一点 x_0, 找出 $S_{\rm sub}$ 中离 x_0 最远的一点 x_1, 再从 $S_{\rm sub}$ 中找出离 x_1 最远的一点 x_2. 核心集被初始化为 $S_0=\{x_1,x_2\}$.

2) 在 Step 2 和 Step 5 中, 为了判断点 x_k 是否位于 $B(c_t,(1+\varepsilon)R_t)$ 球体内, 需要计算在特征空间中 x_k 到球心 c_t 的距离. 记 $c_t{}'=\sum\limits_{x_i\in S_t}\alpha_i\phi(x_i)$, 则有

$$c_t=\begin{bmatrix}c_t{}'\\0\end{bmatrix},\ \tilde{\phi}(x_k)=\begin{bmatrix}\phi(x_k)\\\delta_k\end{bmatrix},$$

$$\|\tilde{\phi}(x_k)-c_t\|^2=\|\phi(x_k)-c_t{}'\|^2+\delta_k^2=$$

$$\sum_{x_i,x_j\in S_t}\alpha_i\alpha_jH(x_i,x_j)-$$

$$2\sum_{x_i\in S_t}\alpha_iH(x_i,x_k)+H(x_k,x_k)+\delta_k^2.$$

这里, 因为 $\Delta=-\mathrm{diag}(H)+\eta=(\delta_1,\delta_2,\cdots,\delta_n)^{\rm T}$, 所以 $\delta_k=-H(x_k,x_k)+\eta$.

3) 作者在实践中发现, 若 η 取值较大, 则 CCMEB 算法的收敛速度非常快, 所得到的核心集数量也较少, 但实验精度也随之降低. 对于 TA-CVM 方法而言, 如果选用高斯核作为核函数, 则可以根据参数 γ 的值预先估计 η 的合理取值.

3.2.2　CCTA-CVM 的时间复杂度

CCTA-CVM 算法是基于 MEB 近似算法的一个特例, 因此在衡量系统开销时, 有关 MEB 核心集的结论适用于 CCTA-CVM 算法. 本文根据文献 [19-21, 23-24], 直接给出如下相似的性质.

性质 1　对于给定的 MEB 近似误差 ε, 由 CCTA-CVM 算法求得的核心集数量的上界为 $O(1/\varepsilon)$, 算法迭代次数的上界为 $O(1/\varepsilon)$.

性质 2　对于给定的误差 ε, CCTA-CVM 算法的时间复杂度的上界为 $O(N/\varepsilon^2+1/\varepsilon^4)$.

性质 1 给出了在最坏情况下的算法叠代次数; 性质 2 给出了在最坏情况下的算法运行时间, 与数据集的容量 N 呈线性关系. 事实上, 作者在实验中发现, 在面对大数据集时, 算法实际叠代次数及运行时间远低于理论最坏值, 这也说明了 CCTA-CVM 算法对大数据集的处理是非常有效的.

4　实验结果与分析

下面对本文所提出的 TA-CVM 方法及其快速算法进行实验验证. 首先, 考察 TA-CVM 方法中引入全局优化后对求解子分类器的影响; 然后, 以 TA-SVM 方法为参照, 评估 TA-CVM 方法的分类精度以及在处理大样本数据集时的时间性能.

4.1　实验设置

实验环境: 操作系统为 Windows XP, 处理器为奔腾 1.87 GHz, 内存为 2 GB, 主要软件 Matlab R2010a.

4.1.1　所用方法

表 1 列出了实验中的各种算法及主要参数. 其中: 参数 C 为错分惩罚因子, γ 为权衡因子, 下面将重点考虑这两个参数; σ 为核参数, 由于采用高斯核 ($\sigma=1$) 后, TA-SVM 和 TA-CVM 均有较好的分类性能, 且更易于估计 CCMEB 中的参数, 以下实验中均选用高斯核 ($\sigma=1$); CCTA-CVM 中的参数 Δ 和 η 可由矩阵 M 主对角线的值直接求得, 不需展开讨论; 参数 ε 仅指 MEB 近似精度, 求解二次规划问题时的精度参数都取默认值 $\varepsilon'=$ 1e-6. 本文使用 CMU 提供的工具包中的 iqph 函数 (http://www.cs.cmu.edu/~ggordon/iqph.m) 求解二次规划问题, 因为其在执行过程中的性能优于 Matlab 自带函数的性能.

表 1　实验所用各种算法及主要参数

算法	所用数学模型及求解方法	主要参数
TA-SVM	基于式 (3) 求解二次规划	C,γ,σ
TA-CVM	基于式 (13) 求解二次规划	C,γ,σ
CCTA-CVM	基于式 (19) 用快速算法求解	$C,\gamma,\sigma,\varepsilon,\Delta,\eta$

4.1.2　实验所用数据及参数设置

为了能够客观地评估 TA-CVM 方法的性能, 本文参照了 TA-SVM 方法的实验设置.

将 TA-SVM[17] 中实验所用的高斯漂移数据集记为 DS1, 这是一个二类数据集, 每个类中数据的特征都在缓慢地变化. 第 4.2 节中的图 2(a) 显示了 DS1 中部分数据生成的过程. DS1 中的数据由下式生成:

$$x_i=\left\{\frac{2i\pi}{n}-\pi+0.2y_i+\varepsilon_1,\sin\left(\frac{2i\pi}{n}-\pi+0.2y_i\right)+\varepsilon_2\right\}. \tag{20}$$

其中: $i=1,2,\cdots,n;\varepsilon_{1,2}$ 服从于均值为0、方差为0.1的正态分布; y_i 为 ±1 的随机序列, 并保持正类负类的个数总体均衡.

将TA-SVM[17]中的旋转超平面记为DS2, 这是缓慢漂移概念研究中最常用的数据集之一 (http://pages.bangor.ac.uk/~mas00a/EPSRC_simulation_framework/changing_environments_stage1a.htm). 分类超平面绕原点不停旋转, 正类负类样本点随时随机生成.

参照TA-SVM[17]的设置, 首先对DS1取 $n=500$, 生成1组训练集, 再分别生成100组验证集和测试集. 实验中先利用验证集优化系统参数, 再使用测试集评估算法的分类能力, 然后对DS2取 $n=360$, 分别独立生成1组训练集、100组验证集、100组测试集. 数据集DS1和DS2用于评估算法的分类精度. 在优化系统参数时, 本文采用网格遍历方式搜索最优参数组合.

增加高斯漂移数据集和旋转超平面的数据采样量 n, 分别得到大样本数据集DS3和DS4, 用于评估算法的时间性能. 有关DS3、DS4的细节详见第4.4节.

4.2 TA-CVM中全局优化项对子分类器的影响

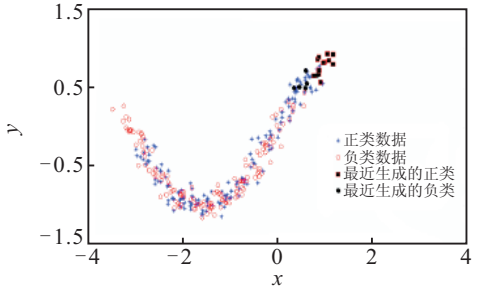
为了考察TA-CVM方法中引入全局优化项后, 能否对子分类器序列起到矫正作用, 本文在数据集DS1上进行了实验. 在TA-CVM方法中引入全局优化项后对求解子分类器序列的影响如图2所示. 其中: 图2(a)为数据集DS1中样本点的生成过程, 图中只显示了前320个点, 深色点为 i 从301到320时生成的点, 即最新生成的20个点; 图2(b)显示了用普通SVM方法求解的子分类器, 将DS1训练集按生成的顺序划分为50个子数据集, 每个子数据集内的数据量为10, 再用常规SVM方法分别求解子分类器; 图2(c)显示了TA-CVM方法求解的子分类器序列, 将DS1训练集划分为500个子数据集, 用本文的TA-CVM方法求得含500个子分类器的序列, 图中画出了 $\mu=5+10\times i(i=0,1,\cdots,49)$ 时的子分类器; 图2(d)中同样使用TA-CVM方法将数据集划分成100个子数据集, 每个子数据集内有5个点, 对求得的100个子分类器相间地画出其中的50个.

由图2可观察得到如下结果.

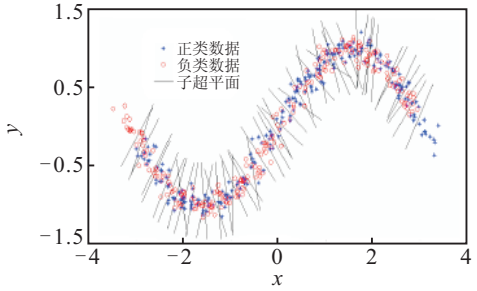
1) 在图2(b)中可以直观地看到, 虽然每个子数据集中的数据相对较多, 但由于算法只求得了局部最优解, 并未将子分类器之间的相关性纳入考虑, 部分相邻子分类器之间的差异较大.

2) 在图2(c)中可以看到, 在每个子数据集中只有1个数据在极端情况下仍能通过相邻子分类器间的相关性求得子分类器序列, 显示出TA-CVM中对相邻子分类器的差异进行约束是有效的.

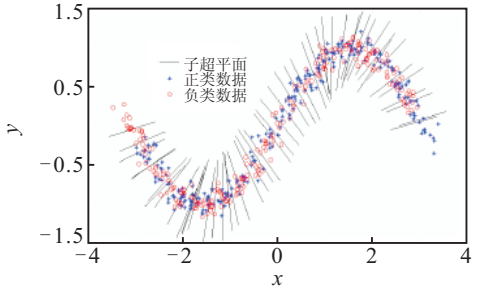
3) 从图2(d)中可以看到, 由于TA-CVM考虑了全局优化, 子分类器序列的变化非常平稳, 得到了预期的良好效果.



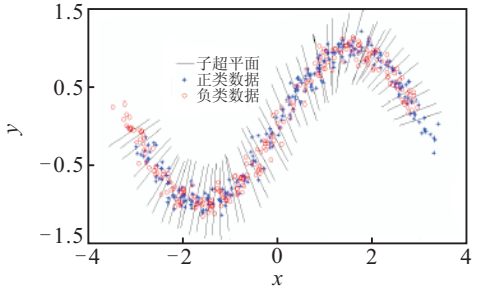
(a) 非静态数据集的生成过程



(b) 使用普通SVM方法求解各子分类器 (每个子数据集含10个点)



(c) 使用TA-CVM求解子分类器序列 (每个子数据集含1个点)



(d) 使用TA-CVM求解子分类器序列 (每个子数据集含5个点)

图2 TA-CVM中全局优化项的作用

综上所述, TA-CVM中引入全局优化项后对子分类器序列的约束和矫正作用非常有效.

4.3 在DS1和DS2上的实验结果及分析

下面对TA-SVM、TA-CVM、CCTA-CVM的分类精度进行评估, 实验在DS1 (划分成50个子数据集) 和DS2 (划分成360个子数据集) 上进行. 根据第4.1.2节所述的参数设置, 利用验证集优化参数组合, 再以此参数对训练集进行训练. 对每个算法都进行100次

实验, 求得平均准确率. CCTA-CVM 算法中的 MEB 近似精度 ε 也与算法准确率相关, 本文在实验中对 ε =1e-4、ε =1e-5、ε =1e-6 共 3 种情况进行了测试.

在 DS1、DS2 上最后得到的平均准确率和标准差分别如表 2 和表 3 所示.

表 2　3 种算法在 DS1 上的平均准确率

所用方法	主要参数值			准确率/%
	C	γ	ε	
TA-SVM	1	$2^{6.59\pm1.33}$	–	97.74 ± 0.21
TA-CVM	10	$2^{10.81\pm0.93}$	–	97.91 ± 0.15
CCTA-CVM	10	$2^{10.47\pm0.93}$	1e-4	97.90 ± 0.16
CCTA-CVM	10	$2^{10.55\pm0.90}$	1e-5	**97.91±0.14**
CCTA-CVM	10	$2^{10.67\pm0.75}$	1e-6	97.91 ± 0.14

表 3　3 种算法在 DS2 上的平均准确率

所用方法	主要参数值			准确率/%
	C	γ	ε	
TA-SVM	1	$2^{12.57\pm1.08}$	–	94.17 ± 0.78
TA-CVM	1	$2^{13.39\pm0.79}$	–	**94.89±0.66**
CCTA-CVM	1	$2^{13.73\pm1.17}$	1e-4	94.70 ± 0.67
CCTA-CVM	1	$2^{13.81\pm1.16}$	1e-5	94.71 ± 0.65
CCTA-CVM	1	$2^{13.77\pm1.13}$	1e-6	94.72 ± 0.67

从表 2 和表 3 的两个对比实验的结果中可以看出, TA-CVM 算法的准确率与 TA-SVM 方法相当, 而结合 MEB 近似算法进行求解的 CCTA-CVM 算法的准确率并没有明显损失.

另外, 从实验中可以看出, 当 ε = 1e-5 时, 已经能够取得较理想的效果, 因此在下文的算法时间性能对比实验中, MEB 的近似程度都取 ε =1e-5.

4.4　在 DS3 和 DS4 上的实验结果及分析

下面在数据集 DS3 和 DS4 上对所提出的方法 CCTA-CVM 进行求解速度评估. 由于 SMO 快速算法常被用于求解样本量较多的 SVM 问题, 本文将 TA-SVM (用 SMO 方法求解) 作为实验参照, 同时将 TA-CVM (iqph 方法求解) 作为基准.

数据集 DS3 为第 4.1.2 节介绍的高斯漂移数据集, 选取不断增加的数据采样量. 由 $m = 50$ 个子集组成, 每个子集中的数据量分别为 $5, 10, \cdots, 200$.

数据集 DS4 为第 4.1.2 节介绍的旋转超平面, 同样选取不断增加的数据采样量. 数据集由 $m = 360$ 个子集组成, 每个子集中的数据量选取不断增加的值, 分别为 $1, 2, 5, 10, 15, 20, 25, 50, 75$.

本文对上述 3 种求解法进行对比, 针对不同的数据量对每种算法进行了 10 次求解实验, 得到其平均求解时间, 分别如表 4 (表 4 中“–”表示运行该方法时, 本实验环境中 Matlab 内存溢出, 算法无法继续执行) 和图 3 所示. 表 4 为在 DS3 上各种算法对不同数据量的 10 次求解的平均时间. 图 3 为在 DS4 上各种算法对不同数据量的平均求解时间.

表 4　数据集 DS3 上各方法的求解时间　s

样本容量	TA-CVM	TA-SVM (SMO)	CCTA-CVM
250	**0.254 8**	0.803 2	3.434 4
500	**1.582 8**	2.229 8	5.374 8
750	7.849 9	**6.959 4**	7.479 1
1 000	21.328 1	12.048 3	**8.869 7**
1 500	75.881 2	21.687 4	**14.484 4**
2 500	344.015 6	52.301 6	**24.195 6**
3 750	–	120.901 6	**47.626 4**
5 000	–	–	**62.565 5**
7 500	–	–	**125.254 6**
10 000	–	–	**189.590 5**

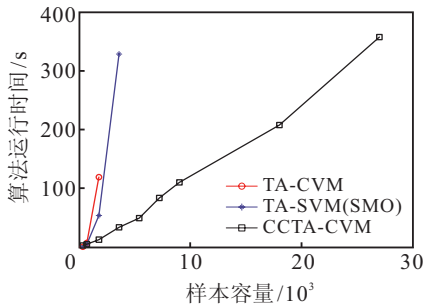


图 3　各方法在 DS4 上的求解时间

观察表 4 和图 3 可以得到如下结论:

1) 从表 4 可看出, 当数据量不超过 1 000 时, TA-SVM (用 SMO 求解) 方法在问题求解时是可行的; 当样本量达到 10^3 的数量级时, 本文方法体现出明显的求解速度的优势.

2) 从图 3 和表 4 可以看出, 在本文 Matlab 实验环境中, 用 SMO 方法求解问题时处理能力不超过 5 000 个数据量, 而本文方法并无这个限制.

3) 从图 3 可以看出, SMO 方法的求解时间接近 $O(n^{2.3})$, 而随着样本量逐渐增加, 本文方法的求解时间上升较缓慢, 其渐近时间复杂度与样本容量 N 呈近似线性关系, 这也从侧面印证了 CVM 理论的正确性.

综上所述, 本文所提出的 CCTA-CVM 方法在处理大样本数据集时, 与 TA-SVM (用 SMO 求解) 方法相比具有明显的速度优势, 从而适用于大样本数据集.

5　结　论

本文针对非静态大数据集的分类问题, 首先提出了 TA-CVM 方法, 并结合近似最小包含球理论设计了本文的中心算法 CCTA-CVM. 本文方法吸收了 TA-SVM 兼顾局部优化和全局优化的优点, 而 CCTA-CVM 算法良好的时间性能又能使得在面对大数据集时仍能获得相对快速的决策. 实验中的准确率及快速性验证了本文方法的有效性. 当然, TA-CVM 仍然需在几个方面进一步研究, 以便适用于更多的应用场合. 如在极端情况 (子数据集个数极多) 下对对称三对角大矩阵 M 的快速求逆问题, 对各子数据集之间的采样时间间隔不固定、各子数据集的数据量不等的情况如何进行优化耦合问题等, 将是下一步的研究重点.

参考文献(References)

[1] Helmbold D P, Long P M. Tracking drifting concepts by minimizing disagreements[J]. Machine Learning, 1994, 14(1): 27-45.

[2] Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts[J]. Machine Learning, 1996, 23(1): 69-101.

[3] Bartlett P L, Ben-Dabid S, Kulkarni S R. Learning changing concepts by exploiting the structure of change[J]. Machine Learning, 2000, 41(2): 153-174.

[4] Bartlett P L. Learning with a slowly changing distribution[C]. Proc the 5th Annual Workshop Computational Learning Theory. Pittsburgh, 1992: 243-252.

[5] Barve R D, Long P M. On the complexity of learning from drifting distributions[C]. Proc the 9th Annual Workshop Computational Learning Theory. San Mateo, 1996: 170-193.

[6] 欧阳震诤. 不平稳数据流的分类技术研究[D]. 长沙: 国防科技大学计算机科学与技术系, 2009.
(Ouyang Z Z. Classification techniques in mining unsteady data streams[D]. Changsha: Department of Computer Science and Technology, National University of Defense, 2009.)

[7] Alippi C, Boracchi G, Roveri M. Just in time classifiers: Managing the slow drift case[C]. Proc Int Joint Conf Neural Network. Atlanta, 2009: 114-120.

[8] Freund Y, Mansour Y. Learning under persistent drift[C]. Proc the 3rd European Conf on Computational Learning Theory. London, 1997: 109-118.

[9] Salganicoff M. Tolerating concept and sampling shift in lazy learning using prediction error context switching[J]. Artificial Intelligence Review, 1997, 11(1): 133-155.

[10] Street W N, Kim Y. A streaming ensemble algorithm(SEA) for large-scale classification[C]. Proc the 7th ACM SIGKDD Int Conf Knowledge Discovery Data Mining. San Francisco, 2001: 377-382.

[11] Bach S H, Maloof M A. Paired learners for concept drift[C]. Proc IEEE Int Conf Data Mining. Los Alamitos, 2008: 23-32.

[12] Alippi C, Roveri M. Just-in-time adaptive classifiers — Part I: Detecting nonstationary changes[J]. IEEE Trans on Neural Network, 2008, 19(7): 1145-1153.

[13] Alippi C, Roveri M. Just-in-time adaptive classifiers — Part II: Designing the classifier[J]. IEEE Trans on Neural Network, 2008, 19(12): 2053-2064.

[14] Klinkenberg R, Joachims T. Detecting concept drift with support vector machines[C]. Proc the 17th Int Conf Machine Learning. San Mateo, 2000: 487-494.

[15] Klinkenberg R, Renz I. Adaptive information filtering: Learning in the presence of concept drifts[C]. Workshop Notes of the ICML/AAAI Workshop Learning for Text Categorization. Menlo Park: AAAI Press, 1998: 33-40.

[16] Lanquillon C. Enhancing test classification to improve information filtering[D]. Magdeburg: Department of Computer Science, University of Magdeburg, 2001.

[17] Grinblat G L, Uzal L C, Ceccatto H A, et al. Solving nonstationary classification problems with coupled support vector machines[J]. IEEE Trans on Neural Network, 2011, 22(1): 37-51.

[18] Platt J. Fast training of support vector machines using sequential minimal optimization[C]. Advances in Kernel Methods-Support Vector Learning. Cambridge: MIT Press, 2000: 185-208.

[19] Tsang I, Kwok J, Zurada J. Generalized core vector machines[J]. IEEE Trans on Neural Networks, 2006, 17(5): 1126-1139.

[20] Tsang I, Kwok J, Cheung P. Core vector machines: Fast SVM training on very large data sets[J]. J of Machine Learning Research, 2005, 6(1): 363-392.

[21] Badoiu M, Clarkson K L. Optimal core sets for balls[J]. Computational Geometry, 2002, 40(1): 14-22.

[22] Schokopf B, Platt J, Shawe-Taylor J, et al. Estimating the support of a high-dimensional distribution[J]. Neural Computation, 2001, 13(7): 1443-1471.

[23] Zhaohong Deng, Fu-Lai Chung, ShitongWang. FRSDE: Fast reduced set density estimator using minimal enclosing ball approximation[J]. Pattern Recognition, 2008, 41(4): 1363-1372.

[24] Kumar P, Mitchell J S B, Yildirim A. Approximate minimum enclosing balls in high dimensions using core-sets[J]. ACM J of Experimental Algorithmics, 2003, 8(1): 1-29.