基于中心加权的局部核向量机算法

李 琳，伍少梅，唐宁九

(四川大学计算机学院 成都 610065)

【摘要】为了解决大规模非线性分类中局部学习的不平衡性问题，提出一种改进的局部支持向量机算法，在高维特征空间中聚类后，为每一个簇构造局部非线性支持向量机。为了克服簇内样本的分布不均衡问题，根据闭合超平面不规则边界的几何特点，经过梯度下降寻找稳定均衡向量，以此构造簇几何中心；再结合簇密度中心共同约束类心形成双重加权中心。然后通过求解加权最小闭球问题实现对大规模样本向量的分类。对照实验显示，除了个别数据集以外，改进的算法在训练时间、测试时间以及测试精度等方面都比另外两种分类算法表现更佳。

关 键 词 双中心; 超曲面; 局部支持向量机; 最小闭球; 稳定均衡向量

中图分类号 TP181 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.04.025

Center-Weighted and Localized Core Vector Machine Algorithm

LI Lin, WU Shao-mei, and TANG Ning-jiu

(School of Computer Science, Sichuan University Chengdu 610065)

Abstract An improved algorithm for localized support vector machine is proposed to resolve the imbalance of local learning problem in nonlinear classifications on large data sets. The algorithm uses the supervised clustering algorithm for clustering in a feature space of high dimension and then constructs local nonlinear support vector machines for each cluster. According to the geometric feature of irregular borders of enclosing sphere, the geometric center for a stable equilibrium point is constructed and a dual-weighted center of two relevant weights is formed through calculating density center of the cluster. At last, the classification of large data set is carried out by solving the problem of weighted minimum enclosing ball. Compared with the other two algorithms of controlled group, the proposed algorithm shows shorter training time and testing time as well as higher testing precision except for some individual data sets.

Key words double centers; hypersurface; localized support vector machine; minimum enclosing ball; stable equilibrium point

支持向量机(support vector machine, SVM)是20世纪90年代中期出现的机器学习技术，具有强泛化能力，在许多应用领域都表现出精准的计算能力，尤其是处理高维数据时[1]。目前，大规模分类问题是SVM研究领域中的一个重要问题，它在网页分类、文本分类、脱机手写体汉字识别、生物信息处理等领域都具有非常广泛的应用。

近几年来，一些针对大规模分类问题的算法层出不穷[2]。例如在线性分类领域，基于L_1-范SVM，文献[3-5]分别提出了随机梯度下降算法，文献[6-7]提出了捆集法，文献[7]提出了割平面算法。文献[8]提出基于k近邻的局部化SVM算法，该算法用k个近邻构造SVM分类器，比较适合大规模数据分类，但其局部学习问题常常会表现出不平衡性[8]。文献[9]提出的基于有监督聚类的局部SVM算法改进了上述问题，但当样本数量较大时会耗时较多。在非线性领域，文献[10]曾提出基于矩阵低秩近似的内点算法；文献[11]在决策树分解的基础上提出SVM分类算法(decision tree support vector machine, DTSVM)。文献[12]在分解算法的基础上提出一种核向量机(core vector machine, CVM)算法，通过解最小闭球(minimum enclosing ball, MEB)问题解决SVM问题，从而避免数据频繁进出工作向量集合，有效降低算法复杂度。然而，该算法不能很好地应对数据分布的不平衡性[13]。

本文提出一种改进的局部化支持向量机，即中心加权的局部核向量机算法(center-weighted and localized core vector machine, CLCVM)，将有监督聚

收稿日期：2013－09－29； 修回日期：2014－02－24

作者简介：李琳(1972－)，女，博士，主要从事网络舆情分析和数据挖掘方面的研究.

类和求解最小闭球有机地结合，并融入双重中心加权的思想，即通过权衡几何中心和密度中心对类心的双重贡献来调整分类面位置，从而有效地解决当数据分布不平衡时分类误差大的问题。为了验证算法的有效性，论文进行了对照实验，在样本训练时间、测试时间以及测试精准度等方面将CLCVM同文献[12]的CVM以及文献[11]的DTSVM进行比较，并对实验结果加以分析。

1　支持向量机

SVM是先把原始输入数据经过非线性变换映射到高维空间中，然后求取最优线性分类面[14]。

1.1　线性SVM：线性可分的情况

令训练集合[1]$D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$，其中$\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{ir})$是一个$r$维多输入向量，属于实数空间$X\subseteq R^r$，$y_i$是它的类标记(输出值)，并且$y_i\in\{1,-1\}$。1表示正类，–1表示负类。$i=1,2,\cdots,n$。为了构造一个分类器，SVM寻找一个线性函数：

$$f(x)=<\boldsymbol{\omega}\cdot\boldsymbol{x}>+b \tag{1}$$

如果$f(x_i)>0$，那么$\boldsymbol{x}_i$被赋予正类；否则被赋予负类：

$$y_i=\begin{cases}1 & \text{当}<\boldsymbol{\omega}\cdot\boldsymbol{x}>+b\geqslant 0\\ -1 & \text{当}<\boldsymbol{\omega}\cdot\boldsymbol{x}>+b<0\end{cases} \tag{2}$$

$f(x)$是一个实值函数f：$x\subseteq R^r\to R$。$\boldsymbol{\omega}=(\omega_1,\omega_2,\cdots,\omega_r)\in R^r$是垂直于超平面的法向量；$b\subseteq R$被称为偏置。$<\boldsymbol{\omega}\cdot\boldsymbol{x}>$表示$\boldsymbol{\omega}$和$x$的点积。

定义 1(线性SVM：线性可分的情况)　给定一个线性可分的训练样本集合D，学习问题就是解决下列约束最小化问题：

$$\min\frac{\langle\boldsymbol{\omega}\cdot\boldsymbol{\omega}\rangle}{2}$$

$$\text{s.t.}\quad y_i(\langle\boldsymbol{\omega}\cdot\boldsymbol{x}_i\rangle+b)\geqslant 1\qquad i=1,2,\cdots,n \tag{3}$$

解决式(3)可以得到$\boldsymbol{\omega}$和b的解，进而得到具有最大边距$2/\|\boldsymbol{\omega}\|$的超平面$<\boldsymbol{\omega}\cdot\boldsymbol{x}_i>+b=0$。由于目标函数是二次和凸的，并且约束在和上是线性的，因此可用标准拉格朗日乘子方法来解决，相应导致对偶问题，具体请参阅文献[1]。

1.2　线性SVM：线性不可分的情况

数据线性可分的情况只是一种理想情况。实际上，由于例外的存在和噪声的污染，可能使样本分类结果产生误差。要想提高分类性能，就必须让SVM适应样本噪声的存在。但是线性可分的SVM是无法从嘈杂的数据中找到解的，因为约束是不能满足的。为此引入松弛变量$\xi_i\geqslant 0$处理不可分的样本点。于是构造最优超平面(广义最优超平面)的问题转换为求解下列优化问题[15]：

$$\min\frac{<\boldsymbol{\omega}\cdot\boldsymbol{\omega}>}{2}+C\left(\sum_{i=1}^{n}\xi_i\right)^k$$

$$\text{s.t. } y_i\ (<\boldsymbol{\omega}\cdot\boldsymbol{x}>+b)\geqslant 1-\xi_i\quad \xi_i\geqslant 0,\ i=1,2,\cdots,n \tag{4}$$

其中$C\geqslant 0$是惩罚分类错误的参数，用于调整置信范围和经验误差之间的均衡。式(4)得到的优化问题仍然是一个凸优化的问题。经常使用$k=1$，这样在对偶问题中ξ_i和它的拉格朗日算符都不会出现。下文只讨论$k=1$的情况。

1.3　非线性SVM

超平面的分类能力毕竟是有限的，为此需要考虑分离曲面。令$\boldsymbol{x}$为输入空间的向量，则通过非线性映射函数φ，把$\boldsymbol{x}$映射到高维特征空间F后，再求最优分类超平面。这样，训练数据$\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$就变成了$\{(\phi(\boldsymbol{x}_1),y_1),(\phi(\boldsymbol{x}_2),y_2),\cdots,(\phi(\boldsymbol{x}_n),y_n)\}$[1]。于是式(4)的优化问题就变成了：

$$\min\frac{<\boldsymbol{\omega}\cdot\boldsymbol{\omega}>}{2}+C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } y_i(<\boldsymbol{\omega}\cdot\phi(x_i)>+b)\geqslant 1-\xi_i\quad \xi_i\geqslant 0,\ i=1,2,\cdots,n \tag{5}$$

对应的对偶问题是：

$$\max L_{\mathrm{D}}=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j<\phi(\boldsymbol{x}_i)\cdot\phi(\boldsymbol{x}_j)>$$

$$\text{s.t.}\quad \sum_{i=1}^{n}\alpha_i y_i=0\qquad 0\leqslant\alpha_i\leqslant C,\ i=1,2,\cdots,n \tag{6}$$

式中，L_{D}是对偶变量；$\alpha_i>0$是拉格朗日乘子。最终的分类决策准则为：

$$\sum_{i=1}^{n}y_i\alpha_i<\boldsymbol{\phi}(\boldsymbol{x}_i)\cdot\boldsymbol{\phi}(\boldsymbol{x})>+b \tag{7}$$

直接将输入数据变换到特征空间然后用线性SVM分类会有一个潜在问题，那就是维灾难。一些有用的变换特征空间的维数可能非常巨大，即使在输入空间中的属性数量并不多。这使得计算很难进行。虽然如此，式(6)和式(7)在特征空间F中只需要计算点积$<\boldsymbol{\phi}(\boldsymbol{x})\cdot\boldsymbol{\phi}(\boldsymbol{z})>$[1]，并不需要直接计算映射后的向量$\boldsymbol{\phi}(\boldsymbol{x})$。在SVM中，可以用核函数解决此点积计算，记为$K$：

$$K(\boldsymbol{x},\boldsymbol{z})=<\boldsymbol{\phi}(\boldsymbol{x})\cdot\boldsymbol{\phi}(\boldsymbol{z})> \tag{8}$$

这样就可以在特征空间中直接计算$\boldsymbol{x}$和$\boldsymbol{z}$的点积(如式(9))。常用的核函数包括多项式式(9)、高斯径向基核式(10)等[9]：

$$K(\boldsymbol{x},\boldsymbol{z})=(<\boldsymbol{x}\cdot\boldsymbol{z}>+\theta)^d \tag{9}$$

$$K(\boldsymbol{x},\boldsymbol{z})=\mathrm{e}^{-\|\boldsymbol{x}-\boldsymbol{z}\|^2/2\sigma} \tag{10}$$

其中，度数$d\in N$，参数$\theta\in R$，$\sigma>0$。

2 改进的局部化支持向量机

局部支持向量机在本质上属于一种分解算法，是解决大规模样本下SVM训练问题的一类有效方法。它将一个大的二次规划问题[9]分解为一系列规模较小的二次规划问题。在每次迭代中利用传统优化算法求解一个子二次规划问题[9]。在实际情况中，样本的分布情况往往很复杂，有些是相互交织在一起的，因此直接利用输入空间的线性判别函数难以得到较高的分类精度，这就需要在聚类后构造局部SVM时，用“超曲面”代替“超平面”，使用“最小闭球”的思想来解决问题。文献[8]验证了求解最小闭球问题等价于求解L_2范支持向量机。

2.1 有监督聚类算法

参照文献[8-9]，设$\boldsymbol{\Sigma}=(\boldsymbol{\sigma}(x_1),\boldsymbol{\sigma}(x_2),\cdots,\boldsymbol{\sigma}(x_n))\in R^{l\times n}$是训练样本和测试样本的相似度矩阵，其中$l$和$n$分别是训练样本和测试样本的个数，$\boldsymbol{\sigma}(x_i)$是第$i$个测试样本和训练样本之间的相似度向量，$Z_{ij}$表示第$i$个训练样本属于第$j$个簇的隶属度，$x_i$表示相似度矩阵$\boldsymbol{\Sigma}$的第$i$行，$c_j$表示第$j$个簇的中心。然后给出聚类的数学模型[8]：

$$\begin{aligned}&\min\sum_{j=1}^{k}\sum_{i=1}^{l}Z_{i,j}\left\|x_j-c_j\right\|^2+R\sum_{j=1}^{k}\left|\sum_{i=1}^{l}Z_{i,j}y_i\right|\\&\text{s.t.}\quad 0\leqslant Z_{i,j}\leqslant 1,\quad i=1,2,\cdots,l\\&\sum_{j=1}^{k}Z_{i,j}=1,\quad j=1,2,\cdots,k\end{aligned}\tag{11}$$

式(11)中目标函数的第二项确保了每个簇中正负类的数目是平衡的。在引入松弛变量t_j的基础上，优化问题式(11)可以转化为下列优化问题[8]：

$$\begin{aligned}&\min_{Z,t}\sum_{j=1}^{k}\sum_{i=1}^{l}Z_{i,j}\left\|x_j-c_j\right\|^2+R\sum_{j=1}^{k}t_j\\&\text{s.t.}\quad -t_j\leqslant Z_{i,j}y_i\leqslant t_j,\ i=1,2,\cdots,l\\&t_j\leqslant 0,\ 0\leqslant Z_{i,j}\leqslant 1,\ j=1,2,\cdots,k\\&\sum_{j=1}^{k}Z_{i,j}=1\end{aligned}\tag{12}$$

优化问题式(12)可用文献[16]的线性规划算法求解。

2.2 簇中心加权思想

鉴于簇边界的不规则性，从第k簇的向量集合(包含簇边界上的虚拟向量)中划分出具有一定规模的子集$\{x_1,x_2,\cdots,x_n\}$构造一个凸集，且该凸集能够包含第k簇的内部所有向量。由于支持向量是指那些在间隔区超平面上的训练样本点，它们到被考虑为闭球结构的SVM的球心距离一样，而包含在闭球内部的向量点与球心的距离总是小于R。当$x_1=x_2$时，$K(x,x)=1$。数据样本点x距离超球体中心c的距离可表示如下：

$$\begin{aligned}&R^2(x)=\left\|\phi(x)-c\right\|^2=\\&K(x,x)-2\sum_{i=1}^{n}\beta_iK(x_i,x)+\sum_{i,j=1}^{n}\beta_i\beta_jK(x_i,x)=\\&1-2\sum_{i=1}^{n}\beta_iK(x_i,x)+\sum_{i,j=1}^{n}\beta_i\beta_jK(x_i,x_j)\end{aligned}\tag{13}$$

其中，$\beta>0$。如果式(13)二次可微，说明能够以超平面上的向量x为起点，找到距离闭球球心最近的向量。然后采用文献[17]的梯度下降算法寻求全局最优解，迭代的终止条件是梯度向量的幅值小于等于一个预先设置的阈值即可，由此找到满足条件的样本点x，此即为均衡向量，如果该向量对应的Jacobian矩阵：

$$\boldsymbol{J}_R(\bar{x})=\nabla^2R^2(\bar{x})\tag{14}$$

该矩阵中的数据样本特征值都大于零[18]，那么就把它视为稳定均衡向量。于是可以在一个闭合超平面中划分出一个子域。

定义 2(几何中心)从不同的数据样本点开始，都可以在局部区域以内找到相应的稳定均衡向量，此即为该簇的几何中心(geometric center)，为了方便描述，用c_g表示。第k个簇的几何中心概括其含义为：

$$c_g(C_k)=\left\{c\,\middle|\,c\in R^d,\ \lim_{x_i\in C_k,C_k\in C}x_i(t)=c\right\}\tag{15}$$

式中，C_k是聚类空间C的第k个簇，i=1,2,…,n；d是向量的维数。

定义 3(密度中心)每个聚类空间子域的所有样本向量的平均值即为该子域的密度中心(density center)，为了方便描述，用c_d表示，于是，第k个簇的密度中心概括其含义为：

$$c_d(C_k)=\frac{1}{m}\sum_{j=1}^{m}\boldsymbol{x}_{kj}\tag{16}$$

式中，C_k是聚类空间C的第k个簇；$\boldsymbol{x}_{kj}$是第k簇的第j个样本向量；j=1,2,…,m；m是第k簇的向量数目。

当第k簇的向量规模不断壮大时，第k簇密度中心逐渐靠近该簇的最稠密分布区的中心。随着样本分布稠密程度的增加，几何中心与密度中心逐渐趋于一致，它们两者之间的距离可以描绘出第k簇内部样本向量的分布特征。当样本向量呈现近似规范的球形分布时，几何中心与密度中心将趋于重合；而当两者之间的距离越大，说明样本数据的分布越稀疏[18]。

由于双中心共同约束着第k个簇的类心位置，本文分别用不同的权值来描述两个中心对类心的约束作用。其中，几何中心对类心c_k的贡献度记为$w_{c_g}(c_k)$，而密度中心对类心c_k的贡献度记为$w_{c_d}(c_k)$。为了增强算法对簇边界不规则几何特点的自适应能力，本文约定几何中心对类心的贡献大于密度中心的贡献[18]。经过约束后的第k个簇的加权类心记为c'_k。

2.3　加权最小闭球问题

定义 4(最小闭球)　设点集$S=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l\}$，其中$\boldsymbol{x}_i \in R^d$。$S$的最小闭球(用MEB($S$)表示)是指包含$S$中所有点的最小球[8,12]。

定义 5((1+ε)近似)　设中心c'_k、半径为R的球记作$B(c'_k, R)$。如果$R \leqslant r_{\mathrm{MEB}(S)}$，$S \subset B(c'_k, (1+\varepsilon)R)$，且$\varepsilon<0$，则称$B(c'_k, (1+\varepsilon)R)$是MEB($S$)的(1+$\varepsilon$)近似[8,12]。

定义 6(核集)　如果$Q \subseteq S$，$B(c'_k, r) = \mathrm{MEB}(Q)$，$S \subset B(c'_k, (1+\varepsilon)r)$，则称点集$Q$为点集$S$的核集[8,12]。

最小闭球问题等价于硬间隔支持向量域描述问题[8,19]，其对应的原始优化问题为：

$$\min_{c_g, c_d, R} R^2$$

$$\text{s.t.}\quad w_{c_g}(C_k)\left\|c_g - \varphi(\boldsymbol{x}_i)\right\|^2 + w_{c_d}(C_k)\left\|c_d - \varphi(\boldsymbol{x}_i)\right\|^2 \leqslant R^2$$

$$i = 1,2,\cdots,l,\quad w_{c_g}(C_k) + w_{c_d}(C_k) = 1$$

$$\varphi(\boldsymbol{x}_i) = w_{c_g}(C_k) \times \Psi(\boldsymbol{x}_i) + w_{c_d}(C_k) \times \phi(\boldsymbol{x}_i) \quad (17)$$

其中，ψ是向量$\boldsymbol{x}$从输入空间X到另一个空间G的几何特征映射[20]。原始优化问题的对偶问题的分量形式为：

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{s.t.}\quad \alpha_i \geqslant 0,\quad i=1,2,\cdots,l,\quad \sum_{i=1}^{l} \alpha_i = 1 \quad (18)$$

上述优化问题的矩阵形式为：

$$\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^{\mathrm{T}} \mathrm{diag}(K) - \boldsymbol{\alpha}^{\mathrm{T}} K \boldsymbol{\alpha}$$

$$\text{s.t.}\quad \boldsymbol{\alpha} \geqslant 0,\quad \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{e} = 1 \quad (19)$$

式中，$\boldsymbol{\alpha}=\alpha_1, \alpha_2, \cdots, \alpha_l$，$\boldsymbol{e}=(1, 1, \cdots, 1)$。

加权中心c'_k、半径R和Lagrange乘子$\boldsymbol{\alpha}$之间有如下关系[8]：

$$c'_k = \sum_{i=1}^{l} \alpha_i \varphi(\boldsymbol{x}_i) \quad (20)$$

$$R = \left(\sum_{i=1}^{l} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)^{\frac{1}{2}} \quad (21)$$

对于特殊核函数$K(\boldsymbol{x}, \boldsymbol{x}) = k$，优化问题式(19)简化为：

$$\max_{\alpha} -\boldsymbol{\alpha}^{\mathrm{T}} K \boldsymbol{\alpha}$$

$$\text{s.t.}\quad \boldsymbol{\alpha} \geqslant 0,\quad \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{e} = 1 \quad (22)$$

2.4　加权核向量机算法

1) 初始化S_0, c_{g0}, c_{d0}, R_0。

2) 如果没有样本点落在球$B(c'_k, (1+\varepsilon)R_k)$的外边，则停止计算[8,12]。

3) 寻找距离加权中心c'_k最远的点z，设$S_{k+1}=S_k \cup \{z\}$。

4) 解对偶问题式(22)得到最优解$\boldsymbol{\alpha}$，根据式(20)和式(21)刷新MEB模型的加权中心和半径，设置$c'_{k+1} = c'_{\mathrm{MEB}(S_{k+1})}$，$R_{K+1} = r_{\mathrm{MEB}(S_{k+1})}$。

5) $k=k+1$，跳转至步骤2)。

在算法步骤1)中，合理的初始化能够提高算法的收敛速度，因此本文采用文献[l3]中的初始化方法，令z是S中任一点，z_m是S中离z最远的点，z_n是S中离z_m最远的点。在分类问题中，要求点z_m、z_n分别属于不同的类。然后，令初始化的核心集为$S_0=\{z_m, z_n\}$，对应系数为$\alpha_m=\alpha_n = 0.5$，半径为$R_0=0.5(2(k(x, x)+2)+2k(x_m, x_n))^{0.5}$

在算法步骤3)中，通过式(23)实现寻找距离加权中心c'_k最远的点z[8]：

$$\begin{aligned}
&\arg \max_{z_m \notin B(C_k, (1+\varepsilon)R_k)} \left\|c'_k - \tilde{\Psi}(z_m)\right\|^2 = \\
&\arg \max_{z_m \notin B(C_k, (1+\varepsilon)R_k)} \sum_{Z_j \in S_k} \alpha_i \tilde{K}(z_j, z_m) = \\
&\arg \max_{z_m \notin B(C_k, (1+\varepsilon)R_k)} \sum_{Z_j \in S_k} \alpha_i K(z_j, z_m) = \\
&\arg \max_{z_m \notin B(C_k, (1+\varepsilon)R_k)} \boldsymbol{\omega}^{\mathrm{T}} \Psi(x_m) = \\
&\arg \max_{z_m \notin B(C_k, (1+\varepsilon)R_k)} (\tilde{K}\boldsymbol{\alpha})_m
\end{aligned} \quad (23)$$

3　实验结果与分析

本文的实验运行环境采用Intel Core i7-3770 3.4 GHz，4 GB内存， Windows 7操作系统。所有算法均采用Java编程实现。

为了验证CLCVM算法的有效性，论文通过对照实验把它和文献[12]提出的CVM算法以及文献[11]提出的DTSVM算法进行比较。实验采用的数据一部分来自复旦大学计算机信息与技术系国际数据库中心自然语言处理小组搜集的文本分类语料库[21]。该语料库包含20个类别，其中训练语料和测试语料的文档数分别为9 804篇和9 833篇。每个类别的文档数分布不均匀，少的类别有十几个文档而多的可达几千个，因此实验还另外选取了Sohu新闻语料库[22]中的部分数据作为补充。该语料库来源于Sohu新闻网站保存的大量经过编辑手工整理与分类的新闻语料

与对应的分类信息。其分类体系包括几十个分类节点，网页规模约为十万篇文档。本文在综合了两个语料库的基础上精心挑选了7个数据集，如表1所示。

由于高维的特征空间为分类算法带来极高的计算复杂度和空间复杂度，且影响算法的可扩展性，所以只有对特征空间进行有效的降维，才能提高文本分类的效率和效果。实验中，本文对于维数较大的数据集采用主成分分析(PCA)方法进行降维处理。然后使用文献[23]主页上的归一化软件对所有数据集进行归一化处理。针对上面的所有数据集，分别把它们分成训练集、验证集和测试集。在实验中，算法的核函数均选用高斯径向基核函数。借鉴文献[8]，本文对SVM的超参数$\pi=(C,\sigma)$进行了网格剖分，其中$C\in\{10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}\}$，$\sigma\in\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$，从而整个寻优空间$\Omega$有63个候选值。设定SVM超参数寻优规模$k$=5。设置分支因子$B$的寻优空间$B$={4,5}，局部规模阈值初始值$r$=1 500。另外设定双中心对类心的贡献度权值分别为$w_{cg}(c_k)$=0.6和$w_{cd}(c_k)$=0.4。

表1 实验中所用的数据集

数据集	类别数	维数	训练样本数	验证样本数	测试样本数	总样本数
体 育	3	12	13 220	3 240	3 440	19 900
财 经	10	23	37 753	7 716	8 858	54 327
IT	8	21	56 080	11 500	11 970	79 550
健 康	5	13	28 192	5 396	6 282	39 870
旅 游	4	20	57 102	12 084	23 253	92 439
教 育	5	15	21 068	4 562	5 260	30 890
文 化	10	32	124 980	32 842	34 358	192 180

表2 3个算法在数据集上的训练时间 s

数据集	体育	财经	IT	健康	旅游	教育	文化
CLCVM	0.85	18.89	18.54	3.69	35.41	4.72	58.86
CVM	1.18	32.73	63.82	4.53.	127.30	1.97	156.02
DTSVM	0.87	19.91	18.75	2.42	35.43	4.69	59.07

表3 3个算法在数据集上的测试时间 s

数据集	体育	财经	IT	健康	旅游	教育	文化
CLCVM	0.06	0.40	5.49	0.81	19.91	0.61	18.57
CVM	0.18	0.60	36.42	1.53.	62.54	0.56	86.72
DTSVM	0.07	0.41	5.73	0.82	17.83	0.59	18.62

表4 3个算法在数据集上的测试精度 %

数据集	体育	财经	IT	健康	旅游	教育	文化
CLCVM	97.10	94.38	97.93	96.58	89.46	90.82	93.16
CVM	96.43	93.74	97.12	95.47.	87.61	91.40	92.70
DTSVM	96.62	93.40	97.81	95.74	87.36	91.57	92.46

表2为3个算法在数据集上的训练时间。从表2可以看出，即便是面对中等以上规模的数据集，除了数据集“健康”和“教育”之外，CLCVM算法比CVM和DTSVM消耗的时间更少。尤其是对于规模较大的“IT”“旅游”以及“文化”这3个数据集，CLCVM训练样本所用的时间最短。对于不同规模的数据集，CLCVM的加权方式对其产生的影响是不同的，即使规模较大训练样本的时间也没有发生显著的增加。在实验中，当局部规模的阈值超过训练样本数时，DTSVM算法仅仅扫描一遍数据集，建立一颗只有一个节点的决策树[8]，因此其训练消耗时间也较少。

表3为3个算法在数据集上的测试时间。从表3中各算法所用的测试时间看，CLCVM算法除了数据集“教育”和“旅游”之外，也表现得比其他两种算法更快。相比而言，CLCVM对训练集的随机访问的依赖性最小，它可以不需要将训练集全部加载到内存，所处理的数据量不会受到内存的限制，因而无论是处理中等规模的数据集还是大规模的数据集时，它都能表现出较稳定的性能和较快的速度。

表4为3个算法在数据集上的测试精度。在表4中，3种算法都表现出较好的测试精度。其中，CLCVM算法除了“教育”以外，在其他几个数据集上表现出比CVM和DTSVM算法更高的精确度。实验充分说明了双重中心加权的思想能够自适应地通过权值的调整来改善样本分布不均衡的数据集的分类效果，至于个别数据集分类性能略逊于对照组，主要是因为其包含的噪声较大且极不均匀所致，由此预示了算法下一步将在抗噪性能方面做进一步扩展。

4 结 论

为了解决大规模非线性分类问题，本文提出了中心加权的局部核向量机算法。该方法考虑了样本的分布形状，具有一定的几何解释，求解过程简单直观，比较适合大规模分类问题。虽然如此，本文在研究分类问题时所涉及的所有样本(包括训练样本和测试样本)都是静止的，在整个学习的过程中没有发生变化。但在实际应用中，样本可能是随着时间的变化而增加的，由此产生的增量学习问题[24]也将是论文下一步要考虑的内容之一。

参 考 文 献

[1] 周志华, 王珏. 机器学习及其应用2007[M]. 北京: 清华大学出版社, 2007.
ZHOU Zhi-hua, WANG Yu. Machine learning and applications 2007[M]. Beijing: Tsinghua University Press, 2007.

[2] VISHWANATHAN S V N, SMOLA A J, MURTY M N. Simple SVM[C]//Proceedings of the Twentieth International Conference on Machine Learning. Washington, D C: [s.n.], 2003: 760-767.

[3] ZHANG T. Solving large scale linear prediction problems using stochastic gradient descent algorithms[C]// Proceedings of the Twentieth-First International Conference on Machine Learning. New York: [s.n.], 2004: 919-926.

[4] BOTTON L. Stochastic gradient descent[EB/OL]. [2013-05-21]. http://blog.sina.com.cn/s/blog_5033f3b40101cdmn.htm.

[5] SHALEV S S, SINGER Y, SREBRO N. Pegasos: Primal estimated sub-gradient solver for SVM[C]//International Conference on Machine Learning (ICML). [S.l.]: [s.n.], 2007: 807-814.

[6] TEO C H, VISHWANATHAN S V N, SMOLA A J, et al. Bundle methods for regularized risk minimization[J]. Journal of Machine Learning Research, 2010(11): 311-365.

[7] JOACHIMS T. Training linear SVMs in linear time[C]//Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.]: [s.n.], 2006: 217-226.

[8] 杨晓伟, 郝志峰. 支持向量机的算法设计与分析[M]. 北京: 科学出版社, 2013.
YANG Xiao-wei, HAO Zhi-feng. Algorithm design and analysis on support vector machine[M]. Beijing: Science Press, 2013.

[9] CHENG H B, TAN P N, JIN R. Efficient algorithm for localized support vector machine[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(4): 537-549.

[10] FINE S, SCHEINBERG K. Efficient SVM training using low-rank kernel representations[J]. Journal of Machine Learning Research, 2001(2): 243-264.

[11] CHANG F, GUO C Y, LIN X R, et al. Tree decomposition for large-scale SVM problems[J]. Journal of Machine Learning Research, 2010(11): 2935-2972.

[12] TSANG I W H, KWOK J T Y, CHEUNG P M. Core vector machines: fast SVM training on very large data sets[J]. Journal of Machine Learning Research, 2005(6): 363-392.

[13] 应维云, 覃正, 赵宇, 等. SVM方法及其在客户流失预测中的应用研究[J]. 系统工程理论与实践, 2007(7): 105-110.
YING Wei-yun, QIN Zheng, ZHAO Yu, et al. Support vectormachine and its application in customer churn prediction[J]. Systems Engineering Theory & Practice. 2007(7): 105-110.

[14] 潘浪, 单明霞. 支持向量机在资源评价中的应用研究[J]. 长江大学学报(自然科学版)理工卷，2009, 6(4): 192-194.
PAN Lang, SHAN Ming-xia. Application of SVM in resource evaluation[J]. Journal of Yangtze University (Natural Science Edition) Sci & Eng V, 2009, 6(4): 192-194.

[15] 杜京义, 侯媛彬. 基于核方法的故障诊断理论及其方法的研究[M]. 北京: 北京大学出版社, 2010.
DU Jing-yi, HOU Yuan-bin. Research on fault diagnosis theory and methods based on kernel methods[M]. Beijing: Peking University Press, 2010.

[16] 尹传环, 牟少敏, 田盛丰, 等. 局部支持向量机的研究进展[J]. 计算机科学, 2012, 39(1): 170-174.
YIN Chuan-huan, MOU Shao-min, TIAN Sheng-feng, et al. Survey of recent trends in local support vector machine[J]. Computer Science. 2012, 39(1): 170-174.

[17]徐昕. 增强学习与近似动态规划[M]. 北京: 科学出版社, 2010.
XU Xin. Enhanced learning and approximate dynamic programming[M]. Beijing: Science Press, 2010.

[18]平源. 基于支持向量机的聚类及文本分类研究[D]. 北京: 北京邮电大学, 2012.
PING Yuan. Research on clustering and text categorization based on support vector machine[D]. Beijing: Beijing University of Posts and Telecommunications, 2012.

[19] TAX D M J, DUIN R P W. Support vector domain description[J]. Pattern Recognition Letters, 2010, 20(14): 1191- 1199.

[20] 崔晨旸, 石教英, 王东辉. 几何特征映射下的3维模型相似性匹配研究[J]. 中国图像图形学报, 2006(5): 46-49.
CUI Chen-yang, SHI Jiao-ying, WANG Dong-hui. 3D model similarity measurement based on geometric feature map[J]. Journal of Image and Graphics, 2006(5): 46-49.

[21] 复旦大学计算机信息与技术系国际数据库中心自然语言处理小组. 文本分类语料库[DB/OL]. [2013-05-21]. http://www. nlp.org. cn/categories /default.php? cat_id = 16.
Natural Language Processing Group of International Database Center, Department of Computer Information and Technology, Fudan University. Corpus of text classification[DB/OL]. [2013-05-21]. http://www. nlp.org. cn/categories /default.php? cat_id = 16.

[22] 搜狗实验室. 搜狐文本分类语料库[DB/OL]. [2013-05-21]. http://www.sogou.com/labs/dl/c.html.
Sogou Lab. Corpus of text classification[DB/OL]. [2013-05-21]. http:// www.sogou.com/labs/dl/c.html.

[23] CHANG C C, LIN C J. LIBSVM: a library for support vector machines[DB/OL]. [2013-05-21]. http://www.csie.ntu.edu.tw/ ~ cjlin/libsvm.

[24] 刘振丙. 基于尺度化凸壳的最大间隔分类方法研究[D]. 武汉: 华中科技大学, 2010.
LIU Zhen-bing. The study of maximal-margin classification approaches based on sealed convex hulls[D]. Wuhan: Huazhong University of Science and Technology, 2010.

编 辑 税 红