适合大样本的线性SVMs快速集成模型

胡文军[1,2] 王 娟[1] 王培良[1] 王士同[2]
(湖州师范学院信息与工程学院 湖州 313000)[1] (江南大学数字媒体学院 无锡 214122)[2]

摘 要 线性SVM具有算法简单、训练和测试速度快等优点,但不能用于解决线性不可分问题。为此,将样本数据集划分为多个集合并分别构造它们的LSVM,然后运用径向基函数的非线性组合来拟合非线性的决策函数,从而解决线性不可分问题。鉴于此,提出了一种适合非线性大样本分类的LSVM快速集成模型FMELSVM。该模型利用径向基函数RBF改善了LSVM的非线性输出能力,同时引进了优化权来提升LSVM的集成效果。UCI数据集的实验结果表明,FMELSVM在处理大样本方面具有较好的性能优势。

关键词 分类,线性SVM,径向基函数,梯度下降法
中图法分类号 TP391.4 **文献标识码** A

Fast Model of Ensembling Linear Support Vector Machines Suitable for Large Datasets

HU Wen-jun[1,2] WANG Juan[1] WANG Pei-liang[1] WANG Shi-tong[2]
(School of Information and Engineering, Huzhou Teachers College, Huzhou 313000, China)[1]
(School of Digital Media, Jiangnan University, Wuxi 214122, China)[2]

Abstract Although the algorithm of linear support vector machine (LSVM) is simple, efficient in training and testing speeds, it can not be applied for nonlinear datasets. For overcoming its drawback, the original training data was splited into several subsets and their LSVMs were respectively constructed. Then, we fit a nonlinear decision function for solving linear inseparation through the combination of the nonlinear radical basis functions (RBFs). Based on this motivation, we developed a new model, called fast model of ensembling LSVMs (FMELSVM), which is suitable for the classification of large datasets. This model improves the nonlinear capabilities of LSVMs using RBF. Meanwhile, the ensembling effects are enhanced by introducing an optimized weight vector. Experimental results on UCI demonstrate that FMELSVM obtains competitive effectiveness for large datasets.

Keywords Classification, Linear SVM, Radical basis function, Gradient descent method

1 引言

针对中大样本的模式分类是模式识别领域的研究热点之一。典型的支持向量机(Support Vector Machine, SVM)[1]分类器获得了很大的改进,如:支持向量数及错分样本数可控的v-SVM[2];考虑数据样本局部信息的平均邻近间隔支持向量机(Average neighbourhood margin SVM, ANMSVM)[3];保持全局和局部结构信息的半监督SVM[4]等。由于大部分真实数据集是线性不可分的,因此核化的SVM(Kernelized SVM, KSVM)得到了广泛研究,如Tsang等人将一类或二类特殊形式的L2-SVM等价于最小包含球(Minimum enclosing ball, MEB)或中心约束MEB(Center constrained MEB, CC-MEB)[5-7],进而利用核心集向量机(Coreset vector machine, CVM)实现大样本训练;Quang-Anh等人提出减少支持向量数的基于聚类方法的Clustering-based SVM[8];从概率密度角度,JooSeuk等人将KSVM推广成L2-Kernel分类器[9]。由于核的引入,增加了KSVM的训练和测试时间。

线性SVM(Linear SVM, LSVM)算法简单,训练和测试速度较快。但LSVM与KSVM一样需要求解二次规划(Quadratic programming, QP)问题,计算复杂度不小于$O(n^2)$,甚至达到$O(n^3)$[5-7]。虽然采用SMO算法[10]可以提高SVM的训练速度,但决策速度仍然与支持向量个数相关,因此它在决策速度方面并没有优势。而LSVM因不能解决线性不可分问题,一直未引起极大的关注。最近,多位学者提出了$O(n)$复杂度的LSVM求解方法,如切平面技巧的SVM-Perf算法[11]、对偶坐标下降法(Dual coordinate descent, DCD)[12]和基于坐标下降或信赖域牛顿法的LIBLinear[13]等,使得LSVM在解决线性可分或高维线性不可分数据方面得到了重大关切。胡等人针对分割超平面(Separating hyperplane, SHP)方法,提出了SHP快速集成(Fast ensemble of

到稿日期:2013-07-02 返修日期:2013-10-19 本文受国家自然科学基金项目(61170122),浙江省自然科学基金项目(LY13F020011, LY12F03008),浙江省科技计划项目(2013C31097),湖州师范学院校级项目(KX24063, KX24058)资助。

胡文军(1977—),男,博士,讲师,主要研究方向为模式识别、人工智能等,E-mail:hoowenjun@yahoo.com.cn;王 娟(1981—),女,硕士,实验师,主要研究方向为智能系统和故障检测等;王培良(1963—),男,硕士,教授,主要研究方向为模式识别、智能控制、过程建模等;王士同(1964—),男,硕士,教授,主要研究方向为模式识别、人工智能、数据挖掘、模糊系统等。

separating hyperplane, FE-SHP)[14]方法，以提高训练速度和解决非线性分类问题。而本文为了解决 LSVM 线性不可分问题，提出了一种适合非线性问题的 LSVM 快速集成模型(Fast model of ensembling LSVMs, FMELSVM)，此模型既考虑了 LSVM 的快速性又体现了 KSVM 的非线性能力。

2 SVM

给定二类数据集 $D=\boldsymbol{X}\times\boldsymbol{Y}=\{(\boldsymbol{x}_i,y_i)\mid 1\leqslant i\leqslant n\}\subset\mathcal{R}^d\times\{\pm 1\}$。

2.1 LSVM 和 KSVM

基于统计学习理论的 SVM 方法是找出最优超平面 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$ 将两类样本分开[1,2]，其数学模型为

$$\begin{aligned}&\min_{\boldsymbol{w},b,\xi_i}\ \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{n}\xi_i\\&\text{s. t. } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1-\xi_i,1\leqslant i\leqslant n\\&\quad\xi_i\geqslant 0,1\leqslant i\leqslant n\end{aligned}\tag{1}$$

其中，C 是惩罚因子，ξ_i 是引入的松弛变量。利用拉格朗日技巧得到对偶形式

$$\begin{aligned}&\max_{\boldsymbol{\alpha}}\ 2\sum_{i=1}^{n}\alpha_i-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j\\&\text{s. t. }\sum_{i=1}^{n}\alpha_i y_i=0,0\leqslant\alpha_i\leqslant C,1\leqslant i\leqslant n\end{aligned}\tag{2}$$

其中，$\boldsymbol{\alpha}=(\alpha_1,\cdots,\alpha_n)^{\mathrm{T}}\geqslant 0$(表示向各个元素均$\geqslant 0$)是拉格朗日乘子向量。求解式(2)得到 LSVM 的分类超平面为

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=\sum_{\boldsymbol{x}_i\in SVs}\alpha_i y_i\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}+b\tag{3}$$

其中，SVs 是支持向量集合，b 是偏移项并可根据 Karush-Kuhn-Tucker (KKT)条件求解得到[1,2]，输出类标签 $y=sign(f(\boldsymbol{x}))$。显然，LSVM 只涉及输入空间以及在此空间中的内积运算，因此算法简单，训练和决策速度较快，但对于非线性数据集，LSVM 很难做到有效分类。

为此，通过 ϕ 将输入空间映射到特征空间，然后利用正定核函数 $k:\mathcal{R}^d\times\mathcal{R}^d\in\mathcal{R}$ 诱导特征空间中的内积形式，即 $k(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$。借助此技巧，LSVM 转变成核化的 KSVM，其数学模型为

$$\begin{aligned}&\min_{\boldsymbol{w},b,\xi_i}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{n}\xi_i\\&\text{s. t. } y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\geqslant 1-\xi_i,1\leqslant i\leqslant n\\&\quad\xi_i\geqslant 0,1\leqslant i\leqslant n\end{aligned}\tag{4}$$

对偶形式为

$$\begin{aligned}&\max_{\boldsymbol{\alpha}}\ 2\sum_{i=1}^{n}\alpha_i-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)\\&\text{s. t. }\sum_{i=1}^{n}\alpha_i y_i=0,0\leqslant\alpha_i\leqslant C,1\leqslant i\leqslant n\end{aligned}\tag{5}$$

KSVM 的分类超平面为

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b=\sum_{\boldsymbol{x}_i\in SVs}\alpha_i y_i k(\boldsymbol{x}_i,\boldsymbol{x})+b\tag{6}$$

2.2 SVM 复杂度分析

不失一般性，选高斯核 $k(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp(-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2/\sigma^2)$ 为核函数，其中 σ^2 是带宽参数。在决策一个未知样本时，根据式(6)可知，KSVM 需要 $|SVs|\times d\times d$ 次加法运算(因为 $\boldsymbol{x}_i-\boldsymbol{x}$ 需要 d 次加法，而 $(\boldsymbol{x}_i-\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}_i-\boldsymbol{x})$ 也需要 d 次加法运算)和 $|SVs|$ 次指数运算；而根据式(3)可知 LSVM 只需要 $|SVs|\times d$ 次加法运算(因为 $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}$ 需要 d 次加法运算)，而实际上 $\boldsymbol{w}$ 已知，只需计算 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$，所以只需要 d 次加法运算，显然在决策速度上 LSVM 优于 KSVM。同理，根据式(5)知 KSVM 对偶问题的二次项矩阵计算需要 $n^2\times d\times d$ 次加法运算和 n^2 次指数运算，而据式(2)知 LSVM 只需要 $n^2\times d$ 次加法运算。LSVM 和 KSVM 均涉及求解 QP 问题，故求解的计算复杂度不小于 $O(n^2)$，甚至达到 $O(n^3)$[5-7]，但 LSVM 可以通过文献[10-12]提出的方法使其计算复杂度降为 $O(n)$。显然，LSVM 在测试和训练速度方面都会明显优于 KSVM，但在非线性能力方面不如 KSVM。为此，设计一种既考虑 LSVM 快速性又能体现 KSVM 非线性能力的 SVM 模型具有一定的意义。

3 FMELSVM

3.1 FMELSVM 模型

为了更好地说明本文算法思路，此节先给出如图 1 所示的 FMELSVM 模型。此模型思想是：先将样本集 D 分成 D_1，…，D_M 共 M 个子集，进而利用 LSVM 对各子集进行训练，根据 LSVM 的输出函数式(3)得到 M 个线性实值输出函数 $f_j(\boldsymbol{x})$；然后进行含参变量 $\boldsymbol{\alpha}$ 权的非线性加权得到非线性的实值输出函数 $f(\boldsymbol{x})$；最后利用尺度函数(Scaling function, SF)将实值函数 $f(\boldsymbol{x})$ 转换成某种概率输出 $p(\boldsymbol{x})$，进而在整个样本集 D 上通过此输出概率和某种优化准则完成权 $\boldsymbol{\alpha}$ 及相关参数的求解。具体过程详见 3.2 节。

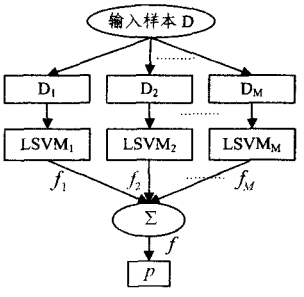

图 1 FMELSVM 模型

3.2 模型求解

含参变量 $\boldsymbol{\alpha}$ 的非线性加权实质上是为了改善 LSVM 集成后的非线性能力，这里选用非线性的径向基 RBF(为简单，本文亦称之为高斯核)为基函数，然后进行加权，即定义如下

$$\lambda_j=\alpha_j\exp(-\|\boldsymbol{x}-\boldsymbol{\mu}_j\|^2/\sigma^2)\tag{7}$$

其中，$\boldsymbol{\mu}_j$ 是 D_j 的中心，σ 是高斯核的带宽参数，$\alpha_j\geqslant 0(1\leqslant j\leqslant M)$是优化参变量即本文所述的优化权。当高斯核输出用于分类时，为了确保输出结果不出现无约束边界，这里合理规定 $\sum_{j=1}^{M}\alpha_j=1$。所以

$$\begin{aligned}f(\boldsymbol{x})&=\sum_{j=1}^{M}\lambda_j f_j(\boldsymbol{x})\\&=\sum_{j=1}^{M}\alpha_j\exp(-\|\boldsymbol{x}-\boldsymbol{\mu}_j\|^2/\sigma^2)(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}+b_j)\\&=\sum_{j=1}^{M}\alpha_j z_j(\boldsymbol{x})\\&=\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{z}(\boldsymbol{x})\end{aligned}\tag{8}$$

其中，

$$\begin{aligned}\boldsymbol{z}(\boldsymbol{x})&=[z_j(\boldsymbol{x})]_{M\times 1}\\&=[\exp(-\|\boldsymbol{x}-\boldsymbol{\mu}_j\|^2/\sigma^2)(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}+b_j)]_{M\times 1}\end{aligned}\tag{9}$$

显然，训练样本 $\boldsymbol{x}$ 经 M 条 LSVM 支路以及非线性高斯核映射成新样本 $\boldsymbol{z}(\boldsymbol{x})$，即 $\exp\circ f:\boldsymbol{x}\in\mathcal{R}^d\mapsto\boldsymbol{z}\in\mathcal{R}^M$，其中符号 $\circ$ 表示函数复合。因此，式(8)给出的是一种非线性关系，基函数 $\exp(-\|\boldsymbol{x}-\boldsymbol{\mu}_j\|^2/\sigma^2)$ 用于改善 LSVM 的非线性能力，而

权系数 α_j 用于优化各 LSVM 的非线性集成能力，以提高测试精度。

对于已知模型中未知参数的估计或求解，普遍采用期望最大算法（Expectation maximization，EM）或梯度下降法等[13]。一般地，需要模型输出转化为某种概率分布 $p(y|\boldsymbol{x},\boldsymbol{\alpha})\in[0,1]$，进而构造似然函数并通过最大似然函数来估计参数。因此，需要将式(8)的 $f(\boldsymbol{x})\in\mathcal{R}$ 通过尺度函数 $SF:\mathcal{R}\to[0,1]$ 估计 $\boldsymbol{x}$ 所属类别的条件概率，即 $SF(f(\boldsymbol{x}))=p(y|\boldsymbol{x})$。Platt 等人提出，通过一个 Sigmoid 函数可以将 SVM(线性或非线性)的实值输出转换成一种后验概率[15]，即

$$p(y=1|\boldsymbol{x})=\frac{1}{1+\exp(Af(\boldsymbol{x})+B)} \tag{10}$$

$$p(y=-1|\boldsymbol{x})=1-\frac{1}{1+\exp(Af(\boldsymbol{x})+B)}=1-p(y=1|\boldsymbol{x}) \tag{11}$$

其中，A 和 B 是待定参数，可通过最小交叉熵获得[15,16]。本文尺度函数也选为 Sigmoid 函数，至此，本文算法演化成求解权向量 $\boldsymbol{\alpha}$ 及参数 A 和 B。为此，构造所有训练样本的交叉熵对数似然函数

$$Q(\boldsymbol{\alpha},A,B)=\sum_{i=1}^{n}t_i\ln p_i+(1-t_i)\ln(1-p_i) \tag{12}$$

其中，$t_i=\frac{1+y_i}{2}$ 和 $p_i=p(y_i=1|\boldsymbol{x}_i)=\frac{1}{1+\exp(Af(\boldsymbol{x}_i)+B)}$。因此

$$\begin{aligned}(\hat{\boldsymbol{\alpha}},\hat{A},\hat{B})&=\underset{\boldsymbol{\alpha}^{\mathrm{T}}1=1,\boldsymbol{\alpha}\in\mathcal{R}^M,A,B\in\mathcal{R}}{\arg\max}\ Q(\alpha,A,B)\\&=\underset{\boldsymbol{\alpha}^{\mathrm{T}}1=1,\boldsymbol{\alpha}\in\mathcal{R}^M,A,B\in\mathcal{R}}{\arg\min}\sum_{i=1}^{n}\{t_i\ln(1+\exp(Af(\boldsymbol{x}_i)+B))+(1-t_i)\ln(1+\exp(Af(\boldsymbol{x}_i)+B))-(1-t_i)\ln\exp(Af(\boldsymbol{x}_i)+B)\}\\&=\underset{\boldsymbol{\alpha}^{\mathrm{T}}1=1,\boldsymbol{\alpha}\in\mathcal{R}^M,A,B\in\mathcal{R}}{\arg\min}\sum_{i=1}^{n}\{\ln(1+\exp(A\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{z}_i+B))-(1-t_i)(A\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{z}_i+B)\}\end{aligned} \tag{13}$$

注意到上式中 $\boldsymbol{\alpha}^{\mathrm{T}}1=1$ 的等式受约束，故不能直接采用 EM 或梯度下降法求解。为此，令 $\alpha_j=\frac{v_j^2}{\|\boldsymbol{v}\|_2^2}$（这里 $\|\cdot\|_2$ 是欧式 2 范数），则

$$\begin{aligned}(\hat{\boldsymbol{v}},\hat{A},\hat{B})&=\underset{v\in\mathcal{R}^M,A,B\in R}{\arg\min}\sum_{i=1}^{n}\{\ln(1+\exp(\frac{A(\boldsymbol{v}^2)^{\mathrm{T}}\boldsymbol{z}_i}{\|\boldsymbol{v}\|_2^2}+B))-(1-t_i)(\frac{A(\boldsymbol{v}^2)^{\mathrm{T}}\boldsymbol{z}_i}{\|\boldsymbol{v}\|_2^2}+B)\}\\&=\underset{v\in\mathcal{R}^M,A,B\in R}{\arg\min}F(\boldsymbol{v},A,B)\end{aligned} \tag{14}$$

其中，$\boldsymbol{v}^p=\underbrace{\boldsymbol{v}\otimes\boldsymbol{v}\otimes\cdots\otimes\boldsymbol{v}}_{p}$，$\otimes$ 是 Hadamard 乘积算子。通过 $F(\boldsymbol{v},A,B)$ 对 A、B 和 $\boldsymbol{v}$ 的梯度可得如下迭代公式：

$$A^{(t_{ab}+1)}\leftarrow A^{(t_{ab})}-\eta_A\sum_{i=1}^{n}\{\{t_i-(1+\exp(\frac{A^{(t_{ab})}(\boldsymbol{v}^{(t_v)2})^{\mathrm{T}}\boldsymbol{z}_i}{\|\boldsymbol{v}^{(t_v)}\|_2^2}+B^{(t_{ab})}))^{-1}\}\frac{(\boldsymbol{v}^{(t_v)2})^{\mathrm{T}}\boldsymbol{z}_i}{\|\boldsymbol{v}^{(t_v)}\|_2^2}\} \tag{15}$$

$$B^{(t_{ab}+1)}\leftarrow B^{(t_{ab})}-\eta_B\sum_{i=1}^{n}\{t_i-(1+\exp(\frac{A^{(t_{ab})}(\boldsymbol{v}^{(t_v)2})^{\mathrm{T}}\boldsymbol{z}_i}{\|\boldsymbol{v}^{(t_v)}\|_2^2}+B^{(t_{ab})}))^{-1}\} \tag{16}$$

$$\boldsymbol{v}^{(t_v+1)}\leftarrow\boldsymbol{v}^{(t_v)}-\eta_v A^{(t_{ab})}\sum_{i=1}^{n}\{t_i-(1+\exp(\frac{A^{(t_{ab})}(\boldsymbol{v}^{(t_v)2})^{\mathrm{T}}\boldsymbol{z}_i}{\|\boldsymbol{v}^{(t_v)}\|_2^2}+B^{(t_{ab})}))^{-1}\}\boldsymbol{z}_i\otimes\frac{\|\boldsymbol{v}^{(t_v)}\|_2^2\boldsymbol{v}^{(t_v)}-\boldsymbol{v}^{(t_v)3}}{\|\boldsymbol{v}^{(t_v)}\|_2^4} \tag{17}$$

其中，$\eta_A>0$，$\eta_B>0$ 和 $\eta_v>0$ 是迭代学习率。

3.3 算法实现

参数 A 和 B 用于拟合 LSVM 的概率输出，而权向量 $\boldsymbol{\alpha}$（或权向量 $\boldsymbol{v}$）用于提升 LSVM 的集成效果，它们的性质是不同的，但迭代中又相互牵连，为了加速算法的迭代收敛，本文将参数 A 和 B 的迭代过程与权向量 $\boldsymbol{\alpha}$ 的迭代过程分开，即先固定某个(些)参数，再迭代另外一个(组)参数，然后反过来迭代，直至收敛，此方法常用于多变量迭代算法中[17]。因此，综合 3.1 节和 3.2 节，FMELSVM 算法归纳如下：

FMELSVM 实现算法

输入：数据集 $D=\mathbf{X}\times\mathbf{Y}=\{(\mathbf{x}_i,y_i)\,|\,1\leqslant i\leqslant n\}\subset\mathcal{R}^d\times\{\pm1\}$，核带宽参数 σ，学习率 η_v，η_A 和 η_B，迭代终止参数 ε_v 和 ε_{ab}；

输出：权向量 $\boldsymbol{\alpha}$ 及参数 A 和 B。

步骤 1　(LSVM 过程)D 分解为 M 个子集 D_j($1\leqslant j\leqslant M$)；利用 LSVM 训练 D_j，得到 M 个决策函数 $f_j(\mathbf{x})$，并计算各个子集的中心点 $\boldsymbol{\mu}_j$；利用式(9)计算 $\mathbf{z}_i$；

步骤 2　(迭代初始化初始)$t_v=0$，$t_{ab}=0$，$A^{(0)}$，$B^{(0)}$，$\mathbf{v}^{(0)}$ 且 $\boldsymbol{\alpha}^{(0)}=\mathbf{v}^{(0)}\otimes\mathbf{v}^{(0)}/\|\mathbf{v}^{(0)}\|_2^2$；

步骤 3　Do while

根据式(15)迭代 $A^{(t_{ab}+1)}$

根据式(16)迭代 $B^{(t_{ab}+1)}$

$t_{ab}=t_{ab}+1$

Until $|A^{(t_{ab})}-A^{(t_{ab}-1)}|+|B^{(t_{ab})}-B^{(t_{ab}-1)}|<\varepsilon_{ab}$；

步骤 4　Do while

根据式(17)迭代 $\mathbf{v}^{(t_v+1)}$；

$\boldsymbol{\alpha}^{(t_v+1)}=\mathbf{v}^{(t_v+1)}\otimes\mathbf{v}^{(t_v+1)}/\|\mathbf{v}^{(t_v+1)}\|_2^2$

$t_v=t_v+1$

Until $\|\boldsymbol{\alpha}^{(t_v)}-\boldsymbol{\alpha}^{(t_v-1)}\|_1<\varepsilon_v$

若步骤 4 仅运行 1 次，则转步骤 5，否则转步骤 3；

步骤 5　$\boldsymbol{\alpha}=\boldsymbol{\alpha}^{(t_v)}$，$A=A^{(t_{ab})}$，$B=B^{(t_{ab})}$。

迭代初值 $\boldsymbol{v}^{(0)}$、$A^{(0)}$ 和 $B^{(0)}$ 会影响到算法效率，为此根据各参数性质给定初值：1)由于 $\alpha_j=v_j^2/\|\boldsymbol{v}\|_2^2$，故 $\boldsymbol{v}^{(0)}$ 可随机产生；2)式(10)表示 $\boldsymbol{x}$ 的+1 类条件概率，必须确保式(10)是单调递增函数，即 $f(\boldsymbol{x})\gg0$ 时 $p(y=1|\boldsymbol{x})\approx1$，显然 $A<0$ 可保证此单调性，故选 $A^{(0)}=0$；3)参数 B 实际上对应 $f(\boldsymbol{x})$ 中的偏移项，而式(10)是 $\boldsymbol{x}$ 的+1 类条件概率，因此可从训练样本的先验概率确定其初值，即 $B^{(0)}=\ln\frac{(N^-+1)}{(N^++1)}$，其中 N^- 和 N^+ 分别是负类和正类的训练样本数。

同时，算法中 M 的取值和样本 D 的划分方式也会影响到模型的分类效果，为此：1)对于 M 的取值，建议取为几到十几，若太大可能会破坏各个子集 D_j 的样本分布结构，而导致各子集结构严重偏离实际样本结构；2)对于样本划分，为了避免划分的子集出现数据不平衡现象和加速获得最优分类界面，建议将样本集 D 的各类平分成 M 个子集。

3.4 复杂度分析

本文算法训练复杂度主要包括两部分：第一部分是利用 LSVM 训练 M 个子集(算法步骤 1)，若采用 QP 求解 LSVM(此时本文算法简记为 $\mathrm{FMELSVM_{QP}}$)，其求解复杂度为 $O(n^3/M^2)$，若采用第 1 节所述方法，以 LIBLinear 为例(此时本

文算法简记为 $FMELSVM_{LIBL}$)，其求解复杂度为 $O(n)$。第二部分是算法的迭代过程(步骤 3 和 4)，当给定 v(参数 A 和 B)后，步骤 3(步骤 4)单次迭代复杂度为 $O(n)$，故步骤 3 和 4 的整体复杂度为 $O((t_{ab}+t_v)n)$。对于整个算法复杂度而言，$FMELSVM_{QP}$ 为 $O(n^3/M^2+(t_{ab}+t_v)n)$，而 $FMELSVM_{LIBL}$ 为 $O((1+t_{ab}+t_v)n)$。显然，本文算法的复杂度明显低于基于 QP 求解的 KSVM 和 L2-Kernel 算法。

关于测试复杂度：式(9)将未知样本 $\boldsymbol{x}$ 映射成 $\boldsymbol{z}(\boldsymbol{x})$，此过程需要 $M\times d+M\times d\times d=M\times d\times(d+1)$ 次加法(因 M 个线性超平面需要 $M\times d$ 次加法，M 个非线性映射需要 $M\times d\times d$ 次加法)和 M 次指数运算；然后通过式(8)计算集成输出 $f(\boldsymbol{x})$ 需要 M 次加法；最后通过式(10)转化概率输出需要 1 次指数运算。整个过程需要 $M\times d\times(d+1)+M\approx M\times d^2$(特别是对于较高维数据时)次加法和 $M+1$ 次指数运算，显然，与训练样本数无关，故本文算法在测试速度方面有较大优势。

4 实验结果与分析

实验环境：CPU 2.6GHz，2G RAM，Intel Core(TM)，XP OS，Matlab 2009a。

利用表 1 中的 UCI 数据集分别从测试精度、训练时间 Training time(单位：s)和分类时间 Testing time(单位：s)等 3 方面比较 $FMELSVM_{QP}$(采用 QP 方法求解 LSVM)、$FMELSVM_{LIBL}$(采用 LIBLinear 方法求解 LSVM)、KSVM 及 L2-Kernel[9]算法的性能，此处安排 QP 方法的 $FMELSVM_{QP}$ 主要是为了比较说明 LIBLinear 方法训练 LSVM 可以获得更低的训练复杂度。表 1 中的 PBRH 是 Pen Based Recognition of Handwritten Digits (数字 0，1 构成)的缩写。L2-Kernel 分类器作为一种新的利用概率差规则的分类器，其优势在于有助于我们分析样本集合的一些核密度估计(Kernel Density Estimate，KDE)性质，在构造时主要考虑到两类样本的分布密度，同时利用累积平方误差(Integrated Squared Error，ISE)准则最优逼近两者密度差实现分类，其实质是将 KSVM 决策函数中正负类样本组合看成各类的分布密度，而最终通过密度差实现优化，它是一个基于概率基础的分类机，故也选用它作为比较对象。关于 L2-Kernel 的详细信息请参考文献[9]。

表 1 UCI 数据集

数据集	样本数	正类样本数	负类样本数	维数
Iris	150	50	100	4
Arrhythmia	420	237	183	278
Breast cancer	699	241	458	9
PBRH	1559	779	780	16
Waveform	3304	1657	1647	21
Waveform noise	3345	1692	1653	40
Landsat satellite	3041	1533	1508	36
Abalone	4177	1407	2770	10

考虑到训练数据的不平衡，采用几何精度 g 进行评价，该方法常用于评价不平衡数据集[19]，即分别统计正负类的分类精度 a^+ 和 a^-，则 $g=\sqrt{a^+\cdot a^-}$，其中 a^+ 和 a^- 分别用下式进行计算。

$$a^+=\frac{\#\ \text{positive samples correctly classified}}{\#\ \text{total positive samples classified}}\times 100\%$$

$$a^-=\frac{\#\ \text{negative samples correctly classified}}{\#\ \text{total negative samples classified}}\times 100\%$$

训练样本和测试样本：首先对样本的各个特征进行了归一—[−1，+1]处理，然后分别从正负类样本中随机抽取 70% 作为训练样本，剩余 30%作为测试样本。

FMELSVM 实验：正负类训练样本平分为 6 组；LSVM 中的惩罚系数 C 从网格{1e−5，1e−4，1e−3，1e−2，1e−1，1e+0，1e+1，1e+2}中寻优；高斯核 $\exp(-\|\boldsymbol{x}-\boldsymbol{\mu}\|^2/\sigma^2)$ 带宽参数 σ^2 从网格{$s^2/8$，$s^2/4$，$s^2/2$，s^2，$2s^2$，$4s^2$，$8s^2$，$16s^2$，$32s^2$，$64s^2$，$128s^2$}中寻优，其中 s 是训练样本的平均 2 范数。为了避免 RBF 出现过大边界，实验数据均在各特征上进行了归一处理，此过程有助于本算法的收敛。

KSVM 实验：本实验采用式(4)的模型进行实验，并在 Matlab 中实现对偶问题式(5)的 QP 求解，惩罚系数 C 从网格{0.02，0.05，0.1，0.2，0.5，1，2，5，10，20，50，100，200，500，1000}中寻优；选高斯核为核函数，带宽参数从网格{$s^2/2048$，$s^2/512$，$s^2/64$，$s^2/32$，$s^2/16$，$s^2/8$，$s^2/4$，$s^2/2$，s^2，$2s^2$，$4s^2$，$8s^2$，$16s^2$}中寻优。

L2-Kernel 实验：选高斯核为核函数，带宽参数从网格{$s^2/128$，$s^2/64$，$s^2/32$，$s^2/16$，$s^2/8$，$s^2/4$，$s^2/2$，s^2，$2s^2$，$4s^2$，$8s^2$，$16s^2$}中寻优；当样本特征维数超过 15 时，L2-Kernel 采用 $L2QP_\eta$ 版本，η 选择方法同文献[9]。

通过 5 重交叉验证选择最优参数，然后每个数据集随机运行 5 次后统计性能指标并以均值和标准差形式输出，表 2—表 4 给出了实验结果。从表 2 可看出，$FMELSVM_{QP}$ 和 $FMELSVM_{LIBL}$ 在精度上优于或等同于 KSVM 和/或 L2-Kernel 算法。对于 Arrhythmia(有 278 个特征)数据集，L2-Kernel 相对较差，这是因 L2-Kernel 算法对于高维样本适应性不强。

表 2 测试精度比较 g(%)

数据集	$FMELSVM_{QP}$	$FMELSVM_{LIBL}$	KSVM	L2-Kernel
Iris	93.63±3.22	95.63±4.34	95.91±3.54	95.23±2.52
Arrhythmia	73.10±1.28	72.09±2.59	72.09±3.94	65.38±4.84
Breast cancer	95.53±1.84	95.17±1.29	97.15±0.82	96.43±0.51
PBRH	100.00±0.00	99.91±0.10	99.87±0.12	99.91±0.10
Waveform	91.40±0.35	91.06±0.57	91.76±0.86	87.91±0.63
Waveform noise	92.54±0.68	91.36±0.73	93.20±0.98	87.59±0.70
Landsat satellite	99.78±0.17	99.84±0.05	93.90±0.08	91.78±0.96
Abalone	79.66±0.93	80.07±1.28	73.13±0.94	78.60±0.75

表 3 训练时间比较(s)

数据集	$FMELSVM_{QP}$	$FMELSVM_{LIBL}$	KSVM	L2-Kernel
Iris	0.6071± 0.1912	3.6766± 2.6884	0.4148± 0.2651	0.4912± 0.2329
Arrhythmia	1.4052± 0.1487	1.0360± 0.5568	11.7603± 0.9682	4.2780± 0.4211
Breast cancer	1.4154± 0.2798	0.5252± 0.1081	140.7271± 15.8481	163.1803± 58.8629
PBRH	10.0885± 0.3166	0.4800± 0.0723	401.2220± 13.0149	443.6479± 40.5153
Waveform	157.2604± 2.3199	3.1443± 0.5942	5245.5628± 28.8632	62.2156± 0.2453
Waveform noise	244.2115± 3.0065	3.1696± 0.3950	5551.6882± 55.8183	69.1285± 0.1757
Landsat satellite	142.6112± 4.2389	8.0633± 0.9919	1890.8793± 19.3153	5236.1527± 274.3196
Abalone	213.3480± 3.9099	8.6566± 1.6294	5806.6646± 37.1739	16712.3206± 715.3591

表 4 测试时间比较(s)

数据集	FMELSVM$_{QP}$	FMELSVM$_{LIBL}$	KSVM	L2-Kernel
Iris	0.0032± 0.0015	0.0033± 0.0017	0.0316± 0.0003	0.0335± 0.0026
Arrhythmia	0.0133± 0.0011	0.0136± 0.0016	0.4884± 0.0018	0.4883± 0.0156
Breast cancer	0.0125± 0.0020	0.0118± 0.0017	0.7129± 0.0045	0.7104± 0.0025
PBRH	0.0255± 0.0013	0.0256± 0.0015	3.6301± 0.0105	3.6274± 0.0090
Waveform	0.0541± 0.0011	0.0540± 0.0012	16.0742± 0.0416	16.5126± 0.0768
Waveform noise	0.0652± 0.0012	0.0580± 0.0017	17.7808± 0.0393	18.2996± 0.0567
Landsat satellite	0.0513± 0.0011	0.0517± 0.0011	14.2593± 0.0370	14.7610± 0.0300
Abalone	0.0662± 0.0015	0.0657± 0.0017	24.8824± 0.0362	39.4698± 27.5485

从表 3 可看出，对于样本数较少的数据集如 Iris，FMELSVM 算法在训练速度上没有任何优势。但对于样本数相对较多的 7 个数据集，本文算法明显快于其他两种算法，其原因是 KSVM 和 L2-Kernel 需要对所有训练样本进行 QP 求解，而 FMELSVM$_{QP}$ 分解为 $M=6$ 个子 QP 完成 LSVM 训练，FMELSVM$_{LIBL}$ 则利用坐标下降和信赖域牛顿法的 LIBLinear 完成 LSVM 训练。从表 3 还可以看出，在所比较的 4 种算法中，FMELSVM$_{LIBL}$ 算法训练速度是最快的。

从表 4 可看出，FMELSVM 明显快于 KSVM 和 L2-Kernel 算法，如对于 Iris 数据集，FMELSVM 比其他两种算法快近 10 倍，而对于 Waveform、Waveform noise 和 Landsat Satellite 数据集，本文算法快于其他算法高出近 250 倍，在 8 个数据集的平均测试速度也高出近 250～300 倍。其原因是本文算法在决策未知样本时与支持向量 SVs 无关，只涉及到 $M=6$ 次非线性运算，而 KSVM 和 L2-Kernel 涉及到 $|SVs|$ 次非线性运算，一般地，$|SVs| \gg M$，故本文算法决策速度快于其他两种算法。

综上所述，FMELSVM 算法在不降低识别精度前提下，训练和测试速度均优于 KSVM 和 L2-Kernel 算法，因此本文算法可以完成对较大样本的训练和测试，特别适用于对实时性要求较高的场合。

结束语 利用基函数改善 LSVM 的非线性能力，并通过优化权提升 LSVM 的集成效果，在此基础上提出了一种适合大样本分类问题的 LSVM 快速集成模型。而求解此模型的几个相关参数则采用梯度下降法实现，同时对参数优化算法的收敛性也给出了证明。最后的 UCI 数据集实验表明，本文所提算法的两种版本 FMELSVM$_{QP}$ 和 FMELSVM$_{LIBL}$ 在测试精度、训练速度和测试速度等方面均获得了较好的效果。实际上，本文是对多个 LSVM 进行集成，然后转化为概率输出从而求解对应的集成模型。那么采用其他集成方式，如先将 LSVM 的输出转化为概率输出，然后再进行集成，或采用其他类型基函数进行非线性映射等，会得到什么效果呢？另外利用梯度下降法求解所提模型中的相关参数，理论上能否保证是全局最优呢？这些问题将是我们近期研究工作的着眼点。

参 考 文 献

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

[2] Schölkopf B, Smola A, Williamson R C, et al. New support vector algorithms[J]. Neural Computation, 2000, 12(5): 1207-1245

[3] 王晓明，王士同. 平均邻近间隔支撑向量机[J]. 智能系统学报，2010，5(4)：313-319

[4] 皋军，王士同，邓赵红. 基于全局和局部保持的半监督支持向量机[J]. 电子学报，2010，38(7)：1626-1633

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

[6] Deng Zhao-hong, Chung Fu-lai, Wang Shi-tong. FRSDE: fast reduced set density estimator using minimal enclosing ball approximation[J]. Pattern Recognition, 2008, 41(4): 1363-1372

[7] Chung Fu-lai, Deng Zhao-hong, Wang Shi-tong. From minimum enclosing ball to fast fuzzy inference system training on large datasets[J]. IEEE Transactions on Fuzzy Systems, 2009, 17(1): 173-184

[8] Tran Q A, Zhang Q L, Li X. Reduce the number of support vectors by using clustering techniques[C]// International Conference on Machine Learning and Cybernetics. Xi'an, China, 2003: 1245-1248

[9] JooSeuk K, Clayton D S. L2 kernel classification [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(10): 1822-1831

[10] Platt J C. Fast training of support vector machines using sequential minimal optimization[M]// Schölkopf B, Burges C J C, Smola A J. Advances in kernel methods. Cambridge, MA: MIT Press, USA, 1999: 185-208

[11] Thorsten J. Training linear SVMs in linear time[C]// Proc 12th ACM International Conference Knowledge Discovery and Data Mining. Philadelphia, PA, 2006: 217-226

[12] Hsieh C J, Chang K W, Lin C J, et al. A dual coordinate descent method for large-scale linear SVM[C]// Proc 25th International Conference Machine Learning. Helsinki, Finland, 2008: 1-12

[13] Fan R E, Chang K W, Hsieh C J, et al. LIBLINEAR: A library for large linear classification[J]. Journal of Machine Learning Research, 2008, 9: 1871-1874

[14] 胡文军，王士同，王娟，等. 非线性分类的分割超平面快速集成方法[J]. 电子信息学报，2012，18(7)：843-860

[15] Platt J C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods[M]// Smola A J, Bartlett P, Schölkopf B, et al. Advances in Large Margin Classifiers. Cambridge: MIT Press, 1999: 61-74

[16] Stefan R. SVM classifier estimation from group probabilities[C]// Proc 27th International Conference Machine Learning. Haifa, Israel, 2010: 911-918

[17] Kovalsky S Z, Cohen G, Hagege R, et al. Decoupled linear estimation of affine geometric deformations and nonlinear intensity transformations of images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(5): 940-946

[18] Banach S. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales[J]. Fundamenta Mathematicae, 1922, 3: 133-181

[19] Tao Jian-wen, Wang Shi-tong, Hu Wen-jun, et al. ρ-Margin kernel learning machine with magnetic field effect for both binary classification and novelty detection[J]. International Journal of Software and Informatics, 2010, 4(3): 305-324