HSMC-SVM 的二次逼近快速训练算法

徐　图　　罗　瑜　　何大可
(西南交通大学信息科学与技术学院　成都　610031)

摘　要：HSMC-SVM 是一种直接型高速多类支持向量机，适合用于类别较多的分类场合，但由于 SMO 算法采用经验方法选择工作集，使得在用 SMO 算法训练 HSMC-SVM 时，收敛速度较慢。为提高 HSMC-SVM 的收敛速度，该文提出用基于二次逼近的可行方向法来训练 HSMC-SVM，并使用了样本缩减策略。实验表明，这种方法可以有效提高 HSMC-SVM 的收敛速度，其收敛速度已经超过了基于 libsvm 的组合多类支持向量机，完全可以用于分类类别多、样本数量大的分类场合。

关键词：超球体多类支持向量机；SMO 训练算法；工作集选择；二次逼近
中图分类号：TP18　　**文献标识码**：A　　**文章编号**：1009-5896(2008)11-2746-04

Training Algorithm of HSMC-SVM Based on Second Order Approximation

Xu Tu　　Luo Yu　　He Da-ke
(*School of Information Science and Technology*, *Southwest Jiaotong University*, *Chengdu*　610031, *China*)

Abstract: HSMC-SVM is a kind of high-speed multi-class SVM with direct mode, and it is appropriate for the situation having lots of categories. Because working set selection of SMO algorithm is based on experience, HSMC-SVM would converge slowly trained with SMO. For accelerating the convergence process of HSMC-SVM, a new approach of working set selection based on second order approximation is proposed. At the same time, shrinking strategy is used too. The numeric experiments show that these measures can speed up the convergence process of HSMC-SVM efficiently. The convergence process of HSMC-SVM is even shorter than these composed multi-class SVMs trained with libsvm. Hence, HSMV-SVM based on second order approximation is very appropriate for the situation that classification category is more and the number of training samples is large.

Key words: Hyper-Sphere Multi-Class SVM(HSMC-SVM); Sequential Minimization Optimization(SMO) training algorithm; Working set selection; Second Order Approximation(SOA)

1　引言

目前经典的多类 SVM 大多是将标准 SVM 进行组合获得的，如 1-v-r 和 1-v-1 等，即是将多分类问题转化为二分类问题，然后用标准 SVM 进行分类，在测试阶段，依据某种判别算法，确定某样本的属类。这种组合方式的多类 SVM，每一类样本都会参加多个 SVM 的训练，因此，当类别比较多时，训练效率低下。2003 年，Zhu 提出了一种基于球结构的多类分类器[1]，其思想是为每类样本建立一个超球体，使得在超球体半径尽可能小的情况下，包含的样本尽可能多。每一类样本只参与一个超球体 SVM 的训练，对于多类样本，就在空间中产生多个超球体，从而形成像肥皂泡一样的分类结构，并使用相应策略进行判决。由于每类样本只参加一个超球体的训练，在处理类别较多的情形，训练效率较高。Xu 将这种多类分类器命名为超球体支持向量机(Hyper-Sphere Multi-Class SVM, HSMC-SVM)，并提出 HSMC-SVM 的 SMO 训练算法[2]，解决了 HSMC-SVM 的训练问题。

SMO 算法是由 Platt 提出的用于标准 SVM 的训练算法[3]，其思想是对乘子两两地进行优化，直到所有乘子满足 Karush-Kuhn-Tucker(KKT)条件为止，使得 SMO 的总体收敛速度较快。但在 SMO 算法中，工作集的选择是基于经验的。它总是把外循环中的首先违反 KKT 条件的乘子做为第 1 个乘子，使得算法的收敛速度依样本的排列顺序而不同。这种基于经验的工作集选择，每次选择出的并非最大违反对，因此使算法的总体收敛速度受到了很大影响。为了提高 SMO 算法的收敛速度，Keerthi 等人提出了最大违反对(the worst violating pair)的概念[4]，直接以最大违反对作为工作集，并证明了这样进行工作集选择可使 SMO 算法有限终止[5]。Joachims 利用 Zoutendijk 可行方向法[6]来求解 SVM 的 QP 问题，并利用目标函数在可行方向上的一阶泰勒展开式，得到了基于一次逼近的工作集选择法[7]，实验证明，这样选择的工作集克服了基于经验的选择法的盲目性，可以快速确定最大违反对，有效提高了 SMO 算法的收敛速度。在一次逼

2007-05-14 收到，2007-11-05 改回

近的基础上，Fan 以目标函数在可行方向上的二阶泰勒展开式构造可行方向法，提出了二次逼近(Second Order Approximation, SOA)的工作集选择法[8]。由于“二次逼近”比“一次逼近”的误差小，故利用“二次逼近”方法可以进一步提高收敛速度。为了加快 HSMC-SVM 的训练速度，将“二次逼近”的思想运用于 HSMC-SVM。实验表明，这样进一步提高了 HSMC-SVM 的训练算法，使得 HSMC-SVM 成为一种高速的直接型多分类器。

2　KKT 条件的判决标准

根据最优化理论，满足目标函数的 KKT 条件的点为原问题的解。因此，首先给出满足 KKT 条件的判决标准，作为迭代结束的停止条件。

HSMC-SVM 是由一组超球体构成的。确定某个超球体，需要求解的 QP 问题是:

$$\left.\begin{aligned}&\min\ f(\boldsymbol{\alpha})=\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\alpha}-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{\alpha}\\&\text{s.t.}\quad 0\le\alpha_i\le C\\&\qquad\quad\sum\alpha_i=1\end{aligned}\right\}\qquad(1)$$

其中 $\boldsymbol{Q}$ 为 $l\times l$ 矩阵，$Q_{ij}=K(x_i,x_j)$，$\boldsymbol{\alpha}$ 和 $\boldsymbol{P}$ 为 l 维列向量，$P_i=K(x_i,x_i)$，$K(x,y)$为核函数，$\boldsymbol{e}$ 为 l 维单位列向量，l 为样本数量。

对式(1)取 Lagrange 函数:

$$L(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\mu})=\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\alpha}-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{T}}(C\boldsymbol{e}-\boldsymbol{\alpha})+Z(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}-1)\qquad(2)$$

令 $\partial L/\partial\boldsymbol{\alpha}=2\boldsymbol{Q}\boldsymbol{\alpha}-\boldsymbol{P}-\boldsymbol{\lambda}+\boldsymbol{\mu}+Z\boldsymbol{e}=0$，又 $\nabla f(\boldsymbol{\alpha})=2\boldsymbol{Q}\boldsymbol{\alpha}-\boldsymbol{P}$，则有

(1)$\nabla f(\boldsymbol{\alpha})_i+Z=\lambda_i-\mu_i$, (2) $\lambda_i\alpha_i=0$, (3) $\mu_i(C-\alpha_i)=0$,

(4) $\lambda_i\ge 0,\mu_i\ge 0,\qquad i=1,\cdots,l$

当 $\alpha_i<C$，由(3)，(4)得 $\mu_i=0,\lambda_i\ge 0$，再由(1)得 $\nabla f(\boldsymbol{\alpha})_i+Z\ge 0$；

当 $\alpha_i>0$，由(2)，(4)得 $\mu_i\ge 0,\lambda_i=0$，再由(1)得 $\nabla f(\boldsymbol{\alpha})_i+Z\le 0$。

即 $\alpha_i<C\Rightarrow-\nabla f(\boldsymbol{\alpha})_i\le Z$；$\alpha_i>0\Rightarrow-\nabla f(\boldsymbol{\alpha})_i\ge Z$。于是，可以定义两个下标集:

$I_{\mathrm{low}}(\boldsymbol{\alpha})=\{t\mid\alpha_t<C\}$，$I_{\mathrm{up}}(\boldsymbol{\alpha})=\{t\mid\alpha_t>0\}$。

若 $i\in I_{\mathrm{low}}(\boldsymbol{\alpha})$，$j\in I_{\mathrm{up}}(\boldsymbol{\alpha})$，则 $-\nabla f(\boldsymbol{\alpha})_i\le-\nabla f(\boldsymbol{\alpha})_j$，则表明 α_i，α_j 满足式(1)的 KKT 条件，否则，α_i，α_j 就是一对违反对。

令 $m(\boldsymbol{\alpha})=\arg\max\{-\nabla f(\boldsymbol{\alpha})_t\mid t\in I_{\mathrm{low}}(\boldsymbol{\alpha})\}$，$M(\boldsymbol{\alpha})=\arg\min\{-\nabla f(\boldsymbol{\alpha})_t\mid t\in I_{\mathrm{up}}(\boldsymbol{\alpha})\}$。

当 $m(\boldsymbol{\alpha})\le M(\boldsymbol{\alpha})$，表明 $\boldsymbol{\alpha}$ 满足 KKT 条件，是一个可行解。

3　二次逼近工作集选择

根据 Zoutendijk 可行方向法，确定可行方向 $\boldsymbol{d}$，选择工作集 α_i 和 α_j，使得 $f(\boldsymbol{\alpha}^k)$ 在 $\boldsymbol{d}$ 上的目标值下降最大，k 为迭代次数。即在 $k+1$ 次迭代时，$\boldsymbol{\alpha}^k$ 由 $\boldsymbol{\alpha}^k+\boldsymbol{d}$ 代替，$f(\boldsymbol{\alpha}^k)$ 就变为 $f(\boldsymbol{\alpha}^k+\boldsymbol{d})$，$f(\boldsymbol{\alpha}^k+\boldsymbol{d})$ 取得最大的下降。将 $f(\boldsymbol{\alpha}^k+\boldsymbol{d})$ 在 $\boldsymbol{\alpha}^k$ 处进行二阶泰勒展开并整理得:

$$f(\boldsymbol{\alpha}^k+\boldsymbol{d})-f(\boldsymbol{\alpha}^k)\approx\nabla f(\boldsymbol{\alpha}^k)_B^{\mathrm{T}}\boldsymbol{d}_B+\frac{1}{2}\boldsymbol{d}_B^{\mathrm{T}}\nabla^2 f(\boldsymbol{\alpha}^k)_B\boldsymbol{d}_B\qquad(3)$$

下标 B 表示工作集，除了工作集中的乘子外，所有乘子均属于非工作集，下文中用 N 表示非工作集。为使 $f(\boldsymbol{\alpha}^k)$ 在 $\boldsymbol{d}$ 上下降最大，即需求解下列最优化问题:

$$\{i,j\}=\arg\min_{B:|B|=2}\ \mathrm{Sub}(B)<0$$

其中 Sub(B)定义为

$$\left.\begin{aligned}&\min\ \nabla f(\boldsymbol{\alpha}^k)_B^{\mathrm{T}}\boldsymbol{d}_B+\frac{1}{2}\boldsymbol{d}_B^{\mathrm{T}}\nabla^2 f(\boldsymbol{\alpha}^k)_B\boldsymbol{d}_B\\&\text{s.t.}\quad \boldsymbol{e}^{\mathrm{T}}\boldsymbol{d}_B=0\\&\qquad\quad d_t\ge 0,\quad \alpha_t^k=0,\quad t\in B\\&\qquad\quad d_t\le 0,\quad \alpha_t^k=C,\quad t\in B\end{aligned}\right\}\qquad(4)$$

约束条件由下式得到:

(1) $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}_B=1-\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}_N$，$\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\alpha}_B+\boldsymbol{d}_B)=1-\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}_N$

$\Rightarrow\boldsymbol{e}^{\mathrm{T}}\boldsymbol{d}_B=0$;

(2) $\alpha_t^k=0,\quad-\alpha_t^k\le d_t\le C-\alpha_t^k\Rightarrow d_t\ge 0$;

(3) $\alpha_t^k=C,\quad-\alpha_t^k\le d_t\le C-\alpha_t^k\Rightarrow d_t\le 0$。

又由 $0\le\alpha_i+d_i\le C$，$0\le\alpha_j+d_j\le C$，$d_i+d_j=0\Rightarrow L\le d_j\le H$，其中 $L=\max\{-\alpha_j,\alpha_i-C\}$，$H=\min\{C-\alpha_j,\alpha_i\}$。

将式(4)展开得

$$\begin{aligned}\mathrm{Sub}(B)&=\nabla f(\boldsymbol{\alpha}^k)_B^{\mathrm{T}}\boldsymbol{d}_B+\frac{1}{2}\boldsymbol{d}_B^{\mathrm{T}}\nabla^2 f(\boldsymbol{\alpha}^k)_B\boldsymbol{d}_B\\&=\nabla f(\boldsymbol{\alpha}^k)_i d_i+\nabla f(\boldsymbol{\alpha}^k)_j d_j+\frac{1}{2}d_i\nabla f^2(\boldsymbol{\alpha}^k)_{ii}d_i\\&\quad+d_i\nabla f^2(\boldsymbol{\alpha}^k)_{ij}d_j+\frac{1}{2}d_j\nabla f^2(\boldsymbol{\alpha}^k)_{jj}d_j\end{aligned}$$

将 $d_i=-d_j$ 代入上式并化简，得

$$\mathrm{Sub}(B)=p_{ij}d_j+\frac{1}{2}\eta_{ij}d_j^2\qquad(5)$$

其中 $p_{ij}=-\nabla f(\boldsymbol{\alpha}^k)_i+\nabla f(\boldsymbol{\alpha}^k)_j$，$\eta_{ij}=k_{ii}-2k_{ij}+k_{jj}$，$k_{ij}$ 表示核函数 $k(x_i,x_j)$。

若核函数满足 Mercer 条件，那么当 $i\ne j$ 时，$p_{ij}>0$，$\eta_{ij}>0$。当 $d_j=-p_{ij}/\eta_{ij}<0$ 时，Sub(B)取得最小值，此时 Sub(B)$=-p_{ij}^2/(2\eta_{ij})$。

综上所述，“二次逼近”算法可描述如下:

(1)取 $i\in\arg\max\limits_t\left\{-\nabla f(\boldsymbol{\alpha}^k)_t\mid t\in I_{\mathrm{low}}(\boldsymbol{\alpha}^k)\right\}$；

(2)取 $j\in\arg\min\limits_t\left\{-\frac{p_{ij}^2}{2\eta_{ij}}\mid t\in I_{\mathrm{up}}(\boldsymbol{\alpha}^k),\ -\nabla f(\boldsymbol{\alpha}^k)_t<-\nabla f(\boldsymbol{\alpha}^k)_i\right\}$；

(3)于是，工作集为 $B=\{i,j\}$，工作集选择结束；若没有发现违反对，迭代过程结束。在判别 $m(\boldsymbol{\alpha})\le M(\boldsymbol{\alpha})$ 时，不必

要求完全满足，可以考虑在一定精度 $e \geqslant 0$ 下满足即可，即只要满足 $m(\boldsymbol{\alpha}) \le M(\boldsymbol{\alpha}) + e$ 即可。

4 缩减策略

在迭代过程中，已经满足 KKT 条件的点，因不再参与优化，将之缩减，可以提高搜索速度，从而加速迭代过程。其缩减策略可描述如下：

当 $i \in I_{\text{low}}$ 时，$-\nabla f(\boldsymbol{\alpha}^k)_i < M(\boldsymbol{\alpha}^k)$ 的样本或当 $i \in I_{\text{up}}$ 时，$-\nabla f(\boldsymbol{\alpha}^k)_j > m(\boldsymbol{\alpha}^k)$ 的样本，均可进行缩减。缩减集 $S = \{t \mid -\nabla f(\boldsymbol{\alpha}^k)_t > m(\boldsymbol{\alpha}^k), \alpha_t^k > 0\} \cup \{t \mid -\nabla f(\boldsymbol{\alpha}^k)_t < M(\boldsymbol{\alpha}^k), \alpha_t^k < C\}$。当迭代次数为 $\min(l,1000)$时，进行一次缩减运算，l 为某类的样本总数。

5 “二次逼近”的收敛性分析

文献[2]指出，高斯核函数最适合用于 HSMC-SVM。因此，此处仅考虑在半正定核条件下的“二次逼近”的收敛性分析，即式(5)中的 $\eta_{ij} > 0$ 。

由式(5)知，

$$f(\boldsymbol{\alpha}^{k+1}) - f(\boldsymbol{\alpha}^k) = p_{ij} d_j + \frac{1}{2} \eta_{ij} d_j^2 \tag{6}$$

当 $d_j = -p_{ij}/\eta_{ij}$ 时，式(6)取最小值，即 $d_j = \max(-p_{ij}/\eta_{ij}, L) < 0$，$L$ 是 d_j 的下界，此时，有 $p_{ij} + \eta_{ij} d_j \ge 0$ 和 $\eta_{ij} > 0$，则由式(6)可得：

$$\begin{aligned} f(\alpha^{k+1}) - f(\alpha^k) &= p_{ij} d_j + \frac{1}{2} \eta_{ij} d_j^2 \\ &= (p_{ij} + \eta_{ij} d_j) d_j - \frac{\eta_{ij}}{2} d_j^2 \\ &\le -\frac{\eta_{ij}}{2} d_j^2 \le 0 \end{aligned} \tag{7}$$

根据文献[1]可知 $f(\boldsymbol{\alpha}^k)$ 为正数序列，即对所有 $k \geqslant 0$，均有 $f(\boldsymbol{\alpha}^k) \geqslant 0$，由式(7)可知 $f(\boldsymbol{\alpha}^k)$ 是递减序列；又因为 $f(\boldsymbol{\alpha})$ 是一个凸二次规划问题，必存在最小值 $f(\overline{\boldsymbol{\alpha}})$，$f(\boldsymbol{\alpha}^k)$ 将在有限步迭代后收敛到 $f(\overline{\boldsymbol{\alpha}})$。故“二次逼近”工作集选择法使 $f(\boldsymbol{\alpha}^k)$ 的迭代过程收敛。

6 数值实验

为了验证“二次逼近”工作集选择法的效率，此处进行了 2 组数值实验。第 1 组实验是“二次逼近”法与 SMO 算法间的比较；第 2 组实验是在“二次逼近”法下训练的 HSMC-SVM 与用 libsvm 训练的多类支持向量机在速度和精度方面的比较。实验环境为 Inter PentiumIV 2.6G，512M RAM，Windows 系统，编译环境为 C++ Builder 6.0。

在第 1 组实验中，从 UCI 数据库中选择 optical，splice，pima，segment，vehicle 数据集[9,10]，这些数据集的基本信息在表 1 中列出。

在上述数据集上分别用文献[2]提出的针对 HSMC-SVM 的 SMO 算法和本文中的“二阶逼近”(Second Order Approximation, SOA)算法进行训练，并进行 10-折交叉验证，分别用不同的核参数，取最好结果。从表 1 可以看出，对于同一个数据集，在相同的训练精度 e 下，SOA 的收敛速度大大快于 SMO 的收敛速度，基本上达到 2~6 倍，而且，训练精度也略优于 SMO 算法。在 Splice 数据集上，为取得较高精度，两种训练算法的核参数略有不同。

为了将用 SOA 训练的 HSMC-SVM 与用 libsvm 训练的多类支持向量机在速度和精度上进行比较，本文做了第 2 组实验。第 2 组实验使用 UCI 数据库中的 Pen-based Handwritten Digits 数据集[9]，其中有 10 个类别，包含 7494 个训练样本和 3498 个测试样本，样本属性为 16。在实验中，使用了 libsvm 的训练部分的代码，分别构建 1-v-1 和 1-v-r 多类分类器，并分别在精度 e=0.1，e=0.05 和 e=0.01 下进行训练和测试，实验结果如表 2 所示，其中训练时间和测试时间为使用相同参数进行 5 次训练后的平均值。

表 1 SMO 和 SOA 训练算法比较

数据集	样本数	类别	属性	训练算法	核参数	训练时间(s)	检测率(%)
optical	3823	10	64	SMO	e=0.1，σ =19	58.649	90.15
				SOA	e=0.1，σ =19	24.713	95.38
splice	1000	2	60	SMO	e=0.01，σ =30	31.396	80.32
				SOA	e=0.01，σ =40.8	7.276	80.55
pima	768	2	8	SMO	e=0.01，σ =21	7.206	82.38
				SOA	e=0.01，σ =21	1.255	83.64
segment	2310	7	19	SMO	e=0.01，σ =0.8	18.87	85.59
				SOA	e=0.01，σ =0.8	6.12	87.31
vehicle	846	4	18	SMO	e=0.1，σ =0.3	5.711	79.82
				SOA	e=0.1，σ =0.3	1.766	81.32

表 2　在 Pen-based Handwritten Digits 数据集上的实验结果

精度	分类法	核参数	训练时间(s)	测试时间(s)	检测率(%)
e=0.1	SOA	σ =41	27.891	11.813	92.62
	1-v-1	σ =41	45.849	18.203	98.08
	1-v- r	σ =41	59.505	9.754	97.65
e=0.05	SOA	σ =43	33.062	10	93.71
	1-v-1	σ =43	45.198	18.593	98.08
	1-v- r	σ =43	60.875	10.281	97.65
e=0.01	SOA	σ =44.5	44.672	5.625	95.45
	1-v-1	σ =44.5	44.87	18.719	98.11
	1-v- r	σ =44.5	63.5	10.948	97.65

在第 2 组实验中，固定 HSMC-SVM 的惩罚参数 C=0.8，对于 libsvm，取 C=100，缓存设为 40M。经过多次实验，选取合适的 σ 值以获得最佳分类效果，为与 libsvm 的分类效果进行对比，将 libsvm 的 σ 值设为与 SOA 相同。在两组实验中，均取高斯核函数：$K(x,y)=\exp(-\|x-y\|^2/2\sigma^2)$。

从表 2 的结果来看，使用 SOA 训练的 HSMC-SVM，在训练速度上已超过了用 libsvm 训练的 1-v-1 和 1-v-r 分类器。不过，使用 HSMC-SVM 进行分类，其分类精度比 1-v-1 和 1-v-r 略低，这是 HSMC-SVM 的不足之处。但在训练样本的类别数较多，样本数量较大，训练速度要求较高的情况下，可以采用 HSMC-SVM 来进行分类，效果较好。

7　结束语

HSMC-SVM 在多类样本间形成多球体分类结构，是一种直接型高速多类分类器，具有训练和测试效率较高的优点，但由于 SMO 算法使用经验方法来确定工作集，使其收敛速度仍然较慢，本文利用“二次逼近”法来确定工作集，并使用了样本缩减策略，使得 HSMC-SVM 收敛速度明显提高。实验表明，基于 SOA 的 HSMC-SVM 在类别较多的情况下，训练速度已经超过了目前成熟的组合多分类器，成为一种新型高速多类分类器。

参 考 文 献

[1] 朱美琳, 刘向东, 陈世福. 用球结构解决多分类问题. 南京大学学报(自然科学版), 2003, 39(2): 153–158.
Zhu Mei-lin, Liu Xiang-dong, and Chen Shi-fu. Solving the problem of multi-class pattern recognition with sphere-structured support vector machines [J]. *Journal of Nanjing University* (*Natural Science*), 2003, 39(2): 153–158.

[2] Xu Tu, He Dake, and Luo Yu. A new orientation for multi-class SVM. Proceeding of 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. Qingdao, China. 30 July to 1 August 2007: 899–904.

[3] Platt J. Fast training of support vector machines using sequential minimal optimization. In: Schölkopf B, Burges C, Smola A, eds. Advances in Kernel Methods-Support Vector Learning. Cambridge, MA: MIT Press, 1999: 185–208.

[4] Keerthi S, Shevade S, Bhattcharyya C, and Murthy K. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 2001, 13(3): 637–649.

[5] Keerthi S and Gilbert E. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 2002, 46(1/3): 351–360.

[6] 薛毅. 最优化原理与方法. 北京: 北京工业大学出版社, 2001: 291–302.
Xue Yi. Optimization Theories and Methods. Beijing: Beijing Industry University Publishing House, 2001: 291–302.

[7] Joachims T. Making large-scale support vector machine learning practical. In: Schölkopf B, Burges C, Smola A, eds. Advances in Kernel Methods-Support Vector Learning. Cambridge, MA: MIT Press, 1999: 169–184.

[8] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training support vector machines. *Machine Learning Research*, 2005 (6): 1889–1918.

[9] http://www.ics.uci.edu/~mlearn/MLRepository.html

[10] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

徐　图：　男，1972 年生，博士生，研究方向为机器学习、智能网络安全.

罗　瑜：　男，1977 年生，博士生，研究方向为并行计算及模式识别.

何大可：　男，1944 年生，教授，博士生导师，研究方向为密码学、信息安全、并行计算.