

偏置 b 对支持向量机分类问题泛化性能的影响

丁晓剑¹ 赵银亮¹

摘要 Poggio 指出支持向量机 (Support vector machine, SVM) 中偏置 b 项是为了保证核函数的正定性, 当使用的核函数为正定核时, b 就不需要存在. 为了验证 b 对 SVM 分类问题泛化性能的影响, 研究了无 b SVM 的优化问题并给出了相应的有效集求解算法. 通过 XOR 分类问题的实验研究得出约束条件 $\sum_1^N y_i \alpha_i = 0$ 会影响 SVM 得到最佳分类超平面. 实验中的基准数据集包括了中小数据集、大规模数据集、高维数据集和多类分类数据集, 并使用高斯正定核和多项式正定核作为核函数. 基于 26 个标准数据集的实验表明无 b SVM 在分类问题中的计算代价要低于 SVM, 泛化性能要好于 SVM. 参数敏感性测试表明无 b SVM 对代价参数变化不太敏感, 这使得无 b SVM 能在较少的参数值对中得到最佳测试精度.

关键词 偏置, 支持向量机, 泛化性能, 有效集

DOI 10.3724/SP.J.1004.2011.01105

Influence of Bias b on Generalization Ability of SVM for Classification

DING Xiao-Jian¹ ZHAO Yin-Liang¹

Abstract It has been pointed out by Poggio that the b term in support vector machine (SVM) is to guarantee the positive definiteness of kernel and b is not needed if the used kernel is positive definite. To testify the role of b in the generalization ability of SVM for classification, optimization formulation of SVM without b is analyzed and the corresponding active set solution algorithm is proposed. By experiments on XOR classification problem, it can be concluded that SVM would fail to reach the optimum classification hyperplane due to the existence of constraint condition $\sum_1^N y_i \alpha_i = 0$. Small to medium data sets, large data sets, high-dimension data sets and mutli-class classification data sets are employed in the simulations as well as the Gaussian positive definite kernel and polynomial positive definite kernel are used. The experimental results on 26 benchmark data sets show that the computational cost of SVM without b is over that of SVM and the generalization performance is over SVM. SVM without b is less sensitive to cost parameter, and this makes SVM without b reaches the optimal testing rate with less parameters pair values.

Key words Bias, support vector machine (SVM), generalization ability, active set

作为支持向量机 (Support vector machine, SVM) 的雏形, Cortes 和 Vapnik^[1] 在神经网络的基础上首先提出了基于两类分类问题的支持向量网络 (Support vector networks) 的概念, 并指出需要调节向量 w 和变量 b 才能得到最优分类超平面. 然而文献 [1] 并未提及 b 存在的实际意义以及 SVM 的优化问题在理论上是否需要 b .

Poggio 等^[2] 从 SVM 优化问题中的核函数的正定性角度研究得出, 如果核函数是条件正定的 (Conditionally positive definite), SVM 优化问题需要 b 的存在; 如果核函数是正定的 (Positive definite), SVM 优化问题中是不需要 b 的. 文献 [3] 从数学学习理论研究得出多项式核 (Polynomial kernel) 在所有训练样本的输入都相异的情况下是正定的. 文献 [4-5] 也同样指出使用高斯核 (Gaussian kernel)

和多项式核的 SVM 优化问题不需要 b . 上述文献仅从理论上分析了各种核函数对于 SVM 优化问题是否需要 b , 并没有研究在具体算法上 b 对于 SVM 优化问题泛化性能的影响. 不过基于统计学习理论思想的类似于 SVM 的学习器^[6-8] 和神经网络学习方法^[9-11] 已经成功地应用了无 b 的情况, 如 Huang 等^[6] 提出的基于极限学习机 (Extreme learning machine) 的分类优化问题, Liu 等^[7] 提出的基于非线性系统识别的选择递归核学习 (Selective recursive kernel learning, SRKL) 优化问题.

为了研究无 b SVM 分类问题在高斯核和多项式核上的泛化性能, 本文分析了无 b SVM 分类原始优化问题和对偶优化问题, 提出了无 b SVM 分类对偶优化问题的一种有效集求解方法, 并与 SVM 分类优化问题的有效集方法在 26 个标准数据集上进行多个指标的比较.

1 无 b SVM 优化问题

1.1 SVM 优化问题

给定一个含有 N 个样本的训练样本集 $\{\mathbf{x}_i, y_i\}_{i=1}^N$, 其中输入为 d 维向量 $\mathbf{x}_i \in \mathbf{R}^d$, 输出

收稿日期 2010-11-09 录用日期 2011-01-25
Manuscript received November 9, 2010; accepted January 25, 2011

国家高技术研究发展计划 (863 计划) (2008AA01Z136) 资助
Supported by National High Technology Research and Development Program of China (863 Program) (2008AA01Z136)

1. 西安交通大学电子与信息工程学院 西安 710049
1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049

为 $y_i \in \mathbf{R}$. 训练 SVM 分类问题等价于求解如下对偶优化问题:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \dots, N \end{aligned} \quad (1)$$

根据 KKT (Karush-Kuhn-Tucker) 定理^[12] 和拉格朗日乘子理论, 求解式 (1) 等价于求解下面的对偶优化问题:

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \end{aligned} \quad (2)$$

其中每个拉格朗日乘子 α_i 对应着一个训练样本 (\mathbf{x}_i, y_i) . 通常训练样本在输入空间中是线性不可分的, 需要引入映射函数 $\phi: \mathbf{R}^d \mapsto \mathbf{H}$ 将 \mathbf{x}_i 映射到高维欧氏空间 \mathbf{H} 中. 由于在对偶优化问题 (2) 中只需计算样本在高维空间中的内积形式 $\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$, 在此只需找满足 $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ 的核函数 K 以代替 ϕ 的计算.

1.2 无 b SVM 优化问题

对于无 b 的 SVM 优化问题, 由于约束条件没有 b 项, 优化问题可表示如下:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i \langle \mathbf{w} \cdot \mathbf{x}_i \rangle \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \dots, N \end{aligned} \quad (3)$$

构造式 (3) 的拉格朗日函数:

$$\begin{aligned} L_1 = \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \\ & \sum_{i=1}^N \alpha_i (y_i \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i \end{aligned} \quad (4)$$

其中, α_i, β_i 为非负拉格朗日乘子.

对 L_1 关于向量 \mathbf{w} 和变量 ξ_i 求偏导数, 下列条件都满足时得到式 (3) 的最优解:

$$\frac{\partial L_1}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (5)$$

$$\frac{\partial L_1}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \beta_i \quad (6)$$

将式 (5) 和式 (6) 代入式 (3) 可得无 b SVM 的对偶优化问题:

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \end{aligned} \quad (7)$$

同式 (2), 利用 $K(\mathbf{x}_i, \mathbf{x}_j)$ 来代替 $\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ 的计算. 与式 (2) 相比, 无 b SVM 的对偶优化问题缺少一个约束条件 $\sum_{i=1}^N y_i \alpha_i = 0$, 从对偶优化问题的解空间看, 式 (7) 的搜索空间比式 (2) 要大, 直观上式 (2) 的解空间是包含于式 (7) 的.

由式 (5) 可知, NBSVR 的决策函数可表示为 $\text{sgn}(f(\mathbf{x}))$, 其中

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \quad (8)$$

其中 $K(\mathbf{x}_i, \mathbf{x})$ 为满足 Mercer 定理的核函数.

式 (7) 可以简化为

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^T H \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \end{aligned} \quad (9)$$

其中, $H = K \mathbf{y} \mathbf{y}^T \in \mathbf{R}^{N \times N}$, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]^T \in \mathbf{R}^N$, $\mathbf{y} = [y_1, \dots, y_N]^T \in \mathbf{R}^N$, $\mathbf{e} = [1, \dots, 1]^T \in \mathbf{R}^N$.

2 无 b SVM 有效集算法

2.1 KKT 条件

无 b SVM 的有效集算法是为了求解式 (9). 为了推导出有效集算法的停止条件, 有必要给出式 (9) 的最优求解条件. 构造式 (9) 的拉格朗日函数:

$$L_2 = \frac{1}{2} \boldsymbol{\alpha}^T H \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} - \sum_{i=1}^N \delta_i \alpha_i - \sum_{i=1}^N \mu_i (C - \alpha_i) \quad (10)$$

其中, δ_i 和 μ_i 为非负拉格朗日乘子. 令 $\mathbf{g}(\boldsymbol{\alpha})$ 为式 (9) 的梯度, 即 $\mathbf{g}(\boldsymbol{\alpha}) = H \boldsymbol{\alpha} - \mathbf{e}$, 则有 $g_i(\boldsymbol{\alpha}) = [H \boldsymbol{\alpha} - \mathbf{e}]_i$, 其中 $g_i(\boldsymbol{\alpha})$ 为向量 $\mathbf{g}(\boldsymbol{\alpha})$ 的第 i 个元素. 式 (10) 的最优条件满足:

$$\frac{\partial L_2}{\partial \alpha_i} = 0 \Rightarrow g_i(\boldsymbol{\alpha}) - \delta_i + \mu_i = 0 \quad (11)$$

根据 KKT 理论, 式 (9) 的 KKT 可行性条件应该为:

$$\begin{aligned} \text{初始可行性:} \quad & 0 \leq \alpha_i \leq C, \quad \forall i \end{aligned} \quad (12)$$

对偶可行性:

$$\delta_i \geq 0, \mu_i \geq 0, \forall i \tag{13}$$

对偶与原问题的补条件:

$$\delta_i \alpha_i = 0, \forall i \tag{14}$$

$$\mu_i(C - \alpha_i) = 0, \forall i \tag{15}$$

由以上 3 种条件分析, 如果 $\alpha_i = 0$, 根据式 (15) 有 $\mu_i = 0$, 再由式 (11) 和式 (13) 可知, $g_i(\boldsymbol{\alpha}) = \delta_i \geq 0$. 如果 $\alpha_i = C$, 根据式 (14) 有 $\delta_i = 0$, 再由式 (11) 和式 (13) 可知, $g_i(\boldsymbol{\alpha}) = -\mu_i \leq 0$. 如果 $0 < \alpha_i < C$, 根据式 (14) 和式 (15) 有 $\delta_i = \mu_i = 0$, 再由式 (11) 可知, $g_i(\boldsymbol{\alpha}) = 0$.

综上所述, 式 (9) 的最优解必须满足下列条件:

$$\begin{aligned} \alpha_i = 0 & \Leftrightarrow g_i(\boldsymbol{\alpha}) \geq 0 \\ \alpha_i = C & \Leftrightarrow g_i(\boldsymbol{\alpha}) \leq 0 \\ 0 < \alpha_i < C & \Leftrightarrow g_i(\boldsymbol{\alpha}) = 0 \end{aligned} \tag{16}$$

有效集算法不断迭代以保持向量 $\boldsymbol{\alpha}$ 的可行性 ($0 \leq \alpha_i \leq C$), 直到式 (16) 全部满足为止.

2.2 有效集算法

定义指标集 $I = \{1, \dots, N\}$, 将 I 划分为 3 个子集 I_0, I_s, I_c : $\forall i \in I_0, \alpha_i = 0, \forall i \in I_c, \alpha_i = C$ 和 $\forall i \in I_s, 0 < \alpha_i < C$. 显而易见有 $I_0 \cup I_s \cup I_c = I$ 和 $I_0 \cap I_s = I_s \cap I_c = I_0 \cap I_c = \emptyset$. 称 I_0 和 I_c 为有效集, I_s 为工作集. 基于该划分策略, 式 (9) 中的向量和矩阵可表示如下:

$$\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\alpha}_s \\ \boldsymbol{\alpha}_c \end{pmatrix}, \boldsymbol{e} = \begin{pmatrix} \boldsymbol{e}_0 \\ \boldsymbol{e}_s \\ \boldsymbol{e}_c \end{pmatrix} \tag{17}$$

$$H = \begin{bmatrix} H_{00} & H_{0s} & H_{0c} \\ H_{s0} & H_{ss} & H_{sc} \\ H_{c0} & H_{cs} & H_{cc} \end{bmatrix}$$

其中, $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_s$ 和 $\boldsymbol{\alpha}_c$ 为 $\boldsymbol{\alpha}$ 的子向量, 由集合 I_0, I_s, I_c 标识, 即 $\boldsymbol{\alpha}_0$ 的所有元素由 α_i 组成, 其中 $i \in I_0, \boldsymbol{\alpha}_s$ 的所有元素由 α_j 组成, 其中 $j \in I_s, \boldsymbol{\alpha}_c$ 的所有元素由 α_k 组成, 其中 $k \in I_c$. $\boldsymbol{e}_0, \boldsymbol{e}_s$ 和 \boldsymbol{e}_c 为 \boldsymbol{e} 的子向量, 由集合 I_0, I_s, I_c 标识. H_{00}, H_{0s} 和 H_{0c} 等矩阵为 H 的子矩阵, 矩阵的行分别由集合 I_0, I_0, I_0 标识, 矩阵的列分别由集合 I_0, I_s, I_c 标识, 即 H_{00} 的所有元素都由 H_{ii} 组成, 其中 $i \in I_0, j \in I_0, H_{0s}$ 和 H_{0c} 同理可得.

当 $i \in I_0$ 时 α_i 不影响式 (9) 中目标函数的计算, 式 (9) 可以简化为

$$\begin{aligned} \min_{\boldsymbol{\alpha}_s} & \frac{1}{2} \boldsymbol{\alpha}_s^T H_{ss} \boldsymbol{\alpha}_s + \boldsymbol{\alpha}_c^T H_{cs} \boldsymbol{\alpha}_s - \\ & \boldsymbol{e}_s^T \boldsymbol{\alpha}_s + \frac{1}{2} \boldsymbol{\alpha}_c^T H_{cc} \boldsymbol{\alpha}_c - \boldsymbol{e}_c^T \boldsymbol{\alpha}_c \end{aligned} \tag{18}$$

s.t. $0 \leq \alpha_{sj} \leq C, \forall j$

其中, α_{sj} 为 $\boldsymbol{\alpha}_s$ 的第 j 个元素, 由 $\boldsymbol{\alpha}_s$ 的定义可知式 (18) 的约束条件必然满足, 将式 (18) 看成是关于变量求最小, 则求解式 (18) 等价于求解下面无约束优化问题:

$$\min_{\boldsymbol{\alpha}_s} \frac{1}{2} \boldsymbol{\alpha}_s^T H_{ss} \boldsymbol{\alpha}_s + \boldsymbol{\alpha}_c^T H_{cs} \boldsymbol{\alpha}_s - \boldsymbol{e}_s^T \boldsymbol{\alpha}_s \tag{19}$$

有效集算法在每次迭代中先判断 $\boldsymbol{\alpha}$ 中元素是否满足式 (16) 的最优解条件, 定义集合 S :

$$\begin{aligned} S = & \{i : \alpha_i = 0 \text{ and } g_i(\boldsymbol{\alpha}) < 0\} \cup \\ & \{i : \alpha_i = C \text{ and } g_i(\boldsymbol{\alpha}) > 0\} \end{aligned} \tag{20}$$

如果集合 $S = \emptyset$, 则有效集算法停止迭代, 否则从 S 中选择一个元素 i 并将相应的分量 α_i 为加入 $\boldsymbol{\alpha}_s$ 重新优化, 选取指标 i 的策略按照下式:

$$\begin{aligned} i = \arg \max_i & \{ |g_i(\boldsymbol{\alpha})|, |g_i(\boldsymbol{\alpha})| < 0, \\ & \forall i \in I_0 \text{ or } g_i(\boldsymbol{\alpha}) > 0, \forall i \in I_c \} \end{aligned} \tag{21}$$

由于有效集算法的迭代向量需要有一个初始值, 在此设 $\boldsymbol{\alpha}^0 = [0, \dots, 0]^T \in \mathbf{R}^N, I_0^0 = \{1, \dots, N\}, I_s^0 = I_c^0 = \emptyset$, 则有效集算法具体步骤如下:

设 $k = 0$, 给定初始迭代向量 $\boldsymbol{\alpha}^k$, 集合 I_0^k, I_s^k 和 I_c^k , 根据式 (20) 计算集合 S^k , 并根据划分策略给出子向量 $\boldsymbol{\alpha}_s^k$ 和 $\boldsymbol{\alpha}_c^k$.

步骤 1. 若 $S^k = \emptyset$, 算法终止, 否则, 转向步骤 2.

步骤 2. 根据式 (21) 选择一个指标 i , 设 $\boldsymbol{\alpha}_s^k = [\boldsymbol{\alpha}_s^k; \alpha_i^k], I_s^k = I_s^k + \{i\}$. 若 $\alpha_i^k = 0$, 则设 $I_0^k = I_0^k - \{i\}$, 否则, 设 $I_c^k = I_c^k - \{i\}$, 将 α_{si}^k 从 $\boldsymbol{\alpha}^k$ 中删除. 转到步骤 3.

步骤 3. 求解式 (19), 得到解 $\boldsymbol{\alpha}_s^{\text{new}}$. 若 $\boldsymbol{\alpha}_s^{\text{new}} \in (0, C)$, 转到步骤 4, 否则, 转到步骤 5.

步骤 4. 设 $\boldsymbol{\alpha}_s^{k+1} = \boldsymbol{\alpha}_s^{\text{new}}, \boldsymbol{\alpha}_c^{k+1} = \boldsymbol{\alpha}_c^k, I_s^{k+1} = I_s^k, I_0^{k+1} = I_0^k, I_c^{k+1} = I_c^k$, 设 $\boldsymbol{\alpha}^{k+1} = [0, \dots, 0]^T \in \mathbf{R}^N, \boldsymbol{\alpha}^{k+1}(I_s^{k+1}) = \boldsymbol{\alpha}_s^{k+1}, \boldsymbol{\alpha}^{k+1}(I_c^{k+1}) = \boldsymbol{\alpha}_c^{k+1}$, 根据式 (20) 更新 S^k , 设 $S^{k+1} = S^k, k = k + 1$, 然后, 转到步骤 1.

步骤 5. 设 $\boldsymbol{d}^k = \boldsymbol{\alpha}_s^{\text{new}} - \boldsymbol{\alpha}_s^k$, 沿着下降方向 \boldsymbol{d}^k 计算合适的步长 λ^k , 设 $\boldsymbol{\alpha}_s^{k+1} = \boldsymbol{\alpha}_s^k + \lambda^k \boldsymbol{d}^k$, 使得对某 j

$\in I_s^k$ 有 $\alpha_{s_j}^{k+1} = 0$ 或者 $\alpha_{s_j}^{k+1} = C$. 将 $\alpha_{s_j}^{k+1}$ 从 α_s^{k+1} 中删除, 设 $I_s^{k+1} = I_s^k \setminus \{i_s\}$, 其中, i_s 为 I_s^k 中的第 j 个元素. 若 $\alpha_{s_j}^{k+1} = 0$, 设 $I_0^{k+1} = I_0^k \setminus \{i_0\}$, 其中 i_0 为 I_0^k 中的第 j 个元素. $\alpha_c^{k+1} = \alpha_c^k$, $I_c^{k+1} = I_c^k \setminus \{i_c\}$, 其中 i_c 为 I_c^k 中的第 j 个元素, 否则, $I_c^{k+1} = I_c^k \cup \{i_c\}$, 其中, i_c 为 I_c^k 中的第 j 个元素, $\alpha_c^{k+1} = [\alpha_c^k; C]$, $I_0^{k+1} = I_0^k$. 令 $k = k + 1$, 转到步骤 3.

算法中的步骤 3 主要是求解式 (19), 由于在本文中采用高斯核和多项式核作为核函数, 在前文分析过, H 是对称正定矩阵, 它的子矩阵 H_{ss} 也是对称正定的, 因此式 (19) 关于 α_s 存在全局最优解. 式 (19) 的最优解可通过下式直接求出:

$$\alpha_s^{\text{new}} = -(H_{ss})^{-1}(H_{cs}^T \alpha_c - e_s)^T \quad (22)$$

由于 α_s^{new} 可能越界, 令 $d^k = \alpha_s^{\text{new}} - \alpha_s^k$, 算法将计算最大步长 $\lambda^k \in [0, 1]$, 使得 $\alpha_s^k + \lambda^k d^k$ 刚好不越界, 即只有一个分量在界上, 其余分量都在界内. 图 1 给出了步长调整的示意图, 其中:

$$\begin{aligned} \lambda^k &= \min\{1, \bar{\lambda}^k\}, \quad \bar{\lambda}^k = \min\{\lambda_1^k, \lambda_2^k\}, \\ \lambda_1^k &= \min_{\{j \in I_s^k | d_j^k < 0\}} \frac{-\alpha_{s_j}^k}{d_j^k}, \quad \lambda_2^k = \min_{\{j \in I_s^k | d_j^k > 0\}} \frac{C - \alpha_{s_j}^k}{d_j^k} \end{aligned} \quad (23)$$

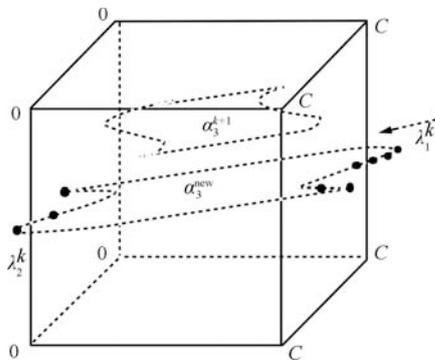


图 1 步长调整方法

Fig. 1 Step size adjusting method

2.3 优化问题解空间分析

无 b SVM 与 SVM 具有相同的目标优化问题, 但是约束条件不同, SVM 除了要满足约束条件 $0 \leq \alpha_i \leq C, \forall i$ 外, 还需满足约束条件 $\sum_1^N y_i \alpha_i = 0$. 无 b SVM 在高维空间 $[0, C]^N$ 中寻找式 (7) 的最优解, 而 SVM 是在高维空间 $[0, C]^N$ 中沿着超平面 $\sum_1^N y_i \alpha_i = 0$ 寻找最优解, 很显然 SVM 只能得到无 b SVM 的次优解. 而且该超平面只与训练样本的输出 y_i 有关, 而与训练样本的输入 x_i 无关, 这意味着不管训练样本在特征空间的位置怎么改变 (输出向量不变), SVM 分类器求解优化问题的约束条件 $\sum_1^N y_i \alpha_i = 0$ 一直都不改变. 为了验证该约束条件的合理性, 将在文后实验中用 XOR 分类问题进

行讨论.

3 基准数据集测试

为了研究 b 对 SVM 分类性能的影响, 本文采用文献 [13] 的算法 AS (Active set) 与本文提出的无 b SVM 有效集算法 NBAS (No bias active set) 进行全面的比较. 由于 AS 算法没有具体说明当最优解 KKT 条件不满足的时候如何选取变量进行迭代, 为了公平地比较, AS 算法采用式 (21) 来选取不满足 KKT 条件的变量进行下一步迭代. 所有算法都是基于 Pentium 4, 2.53 GHZ, Matlab 2007 环境下实现, 实验数据集分别来自 UCI 数据库^[14]、Statlog 数据库^[15] 和 Kent ridge 生物医学数据库^[16]. 实验数据从大小上可分为中小规模数据集和大规模数据集 (Adult 数据集由 Platt 划分为 9 个子集^[17]), 从属性特征上可分为中低维数据集和高维数据集, 从类别上可分为二类分类问题和多类分类问题. 数据集具体描述见表 1, 所有样本的输入都归一化到 $[0, 1]$ 之间, 已分表示给定数据集是否事前已经分为训练样本和测试样本.

3.1 参数设定

对于 AS 算法和 NBAS 算法性能比较使用高斯核 ($K(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|^2 / 2\gamma^2)$) 和多项式核 ($K(\mathbf{u}, \mathbf{v}) = (1 + \mathbf{u}^T \mathbf{v})^P$) 作为核函数, 对于高斯核, 需要合适的核参数 γ 和代价参数 C 才能得到尽量好的分类性能. 参照文献 [18] 中的参数选择方法, 对高斯核, 对于每个数据集选取 15 个 γ 值和 20 个 C 值, 共有 300 种参数对, 并选择性能最好的组合 (C, γ) . 20 个 C 值分别取: 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 500, 1000, 2000, 4000, 8000, 10000 和 100000. 15 个 γ 值分别取: 0.001, 0.01, 0.1, 0.2, 0.4, 0.8, 1, 2, 5, 10, 20, 50, 100, 1000 和 10000. 对多项式核, 对于每个数据集选取 15 个 P 值和 20 个 C 值, 共有 300 种参数对, 并选择性能最好的组合 (C, P) . 20 个 C 值分别取: 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 500, 1000, 2000, 4000, 8000, 10000 和 100000. 15 个 P 值分别取: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15. 对于每个数据集采用 50 次测试, 每次测试随机产生表 1 描述的训练样本数目和测试样本数目. 实验指标包括平均训练时间 (Average training time, ATT)、平均测试精度 (Average training accuracy, ATA) 和标准方差 (Deviation, DEV).

3.2 XOR 分类问题测试

XOR 问题是 Hurwitz^[19] 提出的在二维空间的非线性分类问题, 本文用三元 XOR 来验证第 2.3 节

的论断, 即无 b SVM 解向量 α 满足约束 $\sum_1^N y_i \alpha_i = 0$ 时, SVM 同时得到最优解, 无 b SVM 解向量不满足该约束时, SVM 只能得到次优解. 本节的实验将说明下面的现象: 在一定情况下, 无 b SVM 的最优分类超平面要比 SVM 分类器的最优分类超平面更合理, 在这种情况下无 b SVM 的对偶解向量是不满足约束 $\sum_1^N y_i \alpha_i = 0$ 的.

三元 XOR 问题对应的训练样本为: $\mathbf{x} = [-1, -1; 1, -1; -1, 1]$, $\mathbf{y} = [1, -1, -1]^T$.

表 1 标准数据集

Table 1 Benchmark data sets

数据集	属性数	训练样本数	测试样本数	类别数	已分
Heart	13	70	200	2	否
Pwlinear	10	100	100	2	否
Sonar	60	100	158	2	否
Monk's Problem 1	6	124	432	2	是
Liver-disorders	6	200	145	2	否
Ionosphere	34	200	251	2	否
Breast-cancer	10	300	383	2	否
Australian	14	300	390	2	否
Pimadata	8	400	368	2	否
Creat	44	300	390	2	否
Leukemia	7129	38	34	2	是
Colon	2000	30	32	2	是
Iris	4	100	50	3	否
Segment	19	1000	1320	7	否
Satimage	36	4435	2000	6	是
Wine	13	100	78	3	否
Glass	9	100	114	6	否

图 2 和图 3 为经过调节参数的两个分类器的最优分类超平面, SVM 和无 b SVM 的最优参数对都为 $(2, 1)$. 可以看出, 对于同分布的训练样本和测试样本, 由无 b SVM 分类器得到的分类超平面明显更合理. 注意 SVM 在调整参数时, 只会放大或缩小图 2 所示的不规则圆形大小, 而不会改变其形状. SVM 的分类超平面只对测试样本的 2 维输入都处于 $[-1, 1]$ 时预测的值比较合理, 而当扩大测试样本范围时, 由图 2 得到的超平面很可能将与正类同分布的样本预测为负类. 此时 SVM 分类超平面对应的解为 $\alpha = [1.6148, 0.8074, 0.8074]^T$, 满足约束 $\sum_1^N y_i \alpha_i = 0$. 无 b SVM 超平面对应的解为 $\alpha = [1.3130, 1.1565, 1.1565]^T$, 不满足约束 $\sum_1^N y_i \alpha_i = 0$.

3.3 中小规模数据集性能测试

由表 2 可以看出, 10 个数据集在高斯核函数比较中, NBAS 算法在其中 9 个数据集上的平均测试

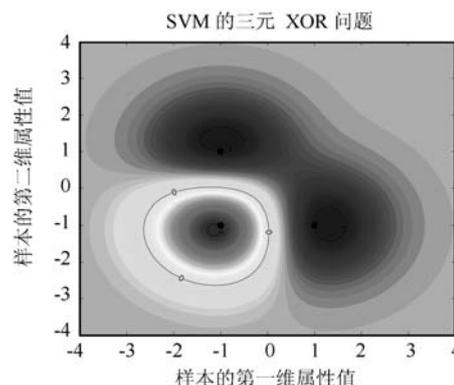


图 2 SVM 对三元 XOR 问题分类

Fig. 2 Three points XOR problem classification of SVM

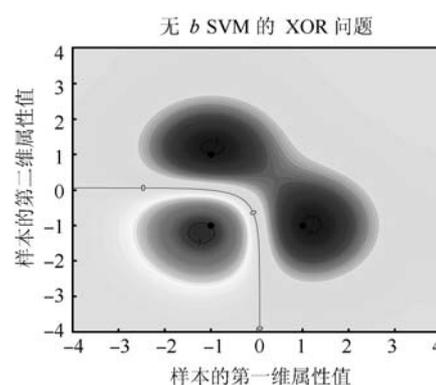


图 3 无 b SVM 对三元 XOR 问题分类

Fig. 3 Three points XOR problem classification of SVM without b

精度要高于 AS 算法, AS 算法则在 Monk's Problem 1 数据集上的平均测试精度稍高. 由表 3 可以看出, 在多项式核函数的比较中, 在 Monk's Problem 1 数据集上两个算法得到相同的平均测试精度, NBAS 算法在其中 7 个数据集上的平均测试精度要好, 而 AS 算法在另外 2 个数据上的平均测试精度要高. NBAS 算法在高斯核和多项式核上 10 个数据集的平均训练时间都要低于 AS 算法.

3.4 大数据集性能测试

在本节实验中, 将使用由 Platt 创建的 9 个 Adult 数据子集来测试, 名称分别为 A1a 到 A9a. 由于在多项式核的参数选取时, 两个算法经常遇到核矩阵奇异化, 以至于得不到正确的解, 导致无法选取最优参数对. 所以本节测试选取高斯核进行两个算法性能的比较. 由表 4 可以看出, NBAS 算法在 3 个子集上的平均测试精度要高于 AS 算法, 在其他 6 个子集上两个算法的平均测试精度相同. 而在平均训练时间指标上, NBAS 在全部数据集上都少于 AS.

表 2 中小规模数据集在高斯核上的性能比较

Table 2 Gaussian kernel performance comparison on small to medium data sets

数据集	AS (高斯核)			NBAS (高斯核)		
	ATT (s)	ATA (%)	DEV (%)	ATT(s)	ATA (%)	DEV (%)
Heart	0.0382	76.00 (10 ⁴ , 5)	3.85	0.0221	76.26 (10 ⁴ , 5)	2.81
Pwlinear	0.0357	84.35 (10 ⁴ , 10 ³)	3.28	0.0167	85.26 (10 ⁴ , 10 ²)	2.32
Sonar	0.0412	83.33 (20, 1)	3.55	0.0239	83.72 (20, 1)	3.54
Monk's problem 1	0.1212	95.37 (10, 1)	0	0.0821	95.14 (10, 1)	0
Liver-disorders	0.0972	68.24 (10, 2)	4.58	0.0839	71.56 (10, 2)	2.93
Ionosphere	0.0357	88.50 (8000, 20)	2.53	0.0161	88.82 (8000, 20)	2.11
Breast-cancer	0.1118	94.20 (5, 50)	0.87	0.0826	95.19 (5, 50)	0.75
Australian	0.3211	67.53 (1, 1)	1.69	0.1408	67.66 (2, 2)	1.74
Pimadata	0.2049	76.53 (2000, 10 ²)	1.99	0.2501	76.93 (2000, 10 ²)	1.40
Creat	0.2026	80.55 (10 ⁵ , 20)	1.79	0.1700	81.88 (10 ⁵ , 10)	1.82

表 3 中小规模数据集在多项式核上的性能比较

Table 3 Polynomial kernel performance comparison on small to medium data sets

数据集	AS (多项式核)			NBAS (多项式核)		
	ATT (s)	ATA (%)	DEV (%)	ATT (s)	ATA (%)	DEV (%)
Heart	0.0273	75.58 (20, 2)	2.62	0.0213	75.79 (20, 2)	3.05
Pwlinear	0.0290	83.92 (10 ⁻² , 1)	2.84	0.0216	84.56 (10 ⁻² , 1)	3.24
Sonar	0.0212	79.31 (1, 6)	4.18	0.0188	79.54 (1, 6)	3.95
Monk's problem 1	0.1214	95.60 (1, 7)	0	0.0826	95.60 (1, 7)	0
Liver-disorders	0.1251	68.24 (0.05, 6)	3.24	0.1073	69.42 (0.05, 6)	2.81
Ionosphere	0.1367	89.77 (0.5, 1)	2.44	0.0218	89.15 (1, 4)	1.91
Breast-cancer	0.0475	94.49 (1, 8)	1.07	0.0616	95.99 (10 ⁻³ , 2)	0.59
Australian	0.1732	67.48 (2, 2)	1.73	0.1470	67.85 (2, 2)	1.74
Pimadata	0.2722	76.79 (10, 1)	1.76	0.1922	76.07 (10 ⁻² , 2)	1.82
Creat	0.2232	80.70 (2000, 1)	1.56	0.3237	81.11 (10 ⁵ , 1)	1.65

表 4 大数据集在高斯核上的性能测试

Table 4 Gaussian kernel performance comparison on large data sets

Adult	AS (高斯核)		NBAS (高斯核)	
	ATT (s)	ATA (%)	ATT (s)	ATA (%)
A1a (1 605, 30 956)	7.521	84.38	6.732	84.38
A2a (2 265, 30 296)	19.71	84.58	18.47	84.59
A3a (3 185, 29 376)	46.60	84.51	43.43	84.51
A4a (4 781, 27 780)	207.5	84.58	186.6	84.59
A5a (6 414, 26 147)	462.9	84.40	445.6	84.40
A6a (11 220, 21 341)	1 676	84.62	1 636	84.65
A7a (16 100, 16 461)	3 273	84.82	3 096	84.82
A8a (22 696, 9 865)	7 916	85.21	7 132	85.21
A9a (32 561, 16 281)	17 939	85.04	13 454	85.04

3.5 高维数据集性能测试

本节将选取 2 组基因表达数据集 Leukemia 和 Colon 进行测试. 白血病数据集 Leukemia 包含了 72 个样本, 每个样本含有 7129 条基因, 其中 47 个样本为急性淋巴白血病 (ALL), 25 个样本为急性骨髓白血病 (AML). 根据文献 [20], 将前 38 个样本作为训练样本 (27ALL, 11AML), 后 34 个样本作为测试样本 (20ALL, 14AML).

结肠癌数据集 Colon 包含了 62 个样本, 每个样本含有 2000 条基因, 其中 22 个样本为正常样本 (Normal), 40 个样本为结肠癌样本 (Tumor). 在此, 将前 30 个样本作为训练样本, 后 32 个样本作为测试样本. 由表 5 和表 6 可以看出, NBAS 算法在高斯核上的测试精度要优于 AS, 在多项式核上泛化性能比较差. 在训练时间上可以看出两个算法使用高斯核时受基因数目影响较大, 当基因数较多时, 训练时间相对较长. 而在使用多项式核时, 两个算法的训练时间相对高斯核要少很多.

3.6 多类数据集性能测试

由于 SVM 直接求解多类分类问题比较困难, 本节测试利用 one-against-one (OAO) 和 one-against-all (OAA) 方法^[21] 将多类分类问题转化为两类问题求解. 由表 7 和表 8 可见, 在 OAO 方法中, 在 4 个数据集上 NBAS 算法平均测试精度要高, 在 OAA 方法中, NBAS 算法在全部 5 个数据集上的平均测试精度都要高于 AS 算法.

3.7 参数敏感性测试

本节主要分析核参数和代价参数对 2 种算法训练精度和测试精度的影响, 实验数据选取 Pwlinear 数据集. 由于两种算法的多项式核矩阵在很多参数对上易陷入奇异化, 在本实验中核函数选择高斯核.

从图 4 和图 5 可以看出, AS 算法对代价参数 C 和核参数 γ 都比较敏感, 而 NBAS 算法的测试精度只对核参数 γ 比较敏感, 而对代价参数 C 没有 AS 算法敏感. 这是一个非常重要的性质, 虽然在本文用

表 5 基因数据在高斯核上的测试

Table 5 Gaussian kernel test on gene data sets

数据集	AS (高斯核)			NBAS (高斯核)		
	(C, γ)	ATT (s)	ATA (%)	(C, γ)	ATT (s)	ATA (%)
Leukemia	$(10^3, 10^3)$	3.2674	82.35	$(10^3, 10^3)$	3.0627	82.35
Colon	$(10^3, 10^3)$	0.2391	81.25	$(5, 10^2)$	0.2230	84.38

表 6 基因数据在多项式核上的测试

Table 6 Polynomial kernel test on gene data sets

数据集	AS (多项式核)			NBAS (多项式核)		
	(C, P)	ATT (s)	ATA (%)	(C, P)	ATT (s)	ATA (%)
Leukemia	$(10^{-3}, 1)$	0.0241	82.35	$(10^{-3}, 1)$	0.0119	79.41
Colon	$(10^{-3}, 1)$	0.0137	81.25	$(10^{-2}, 1)$	0.0049	78.13

表 7 多类数据集 OAO 方法比较

Table 7 OAO method comparison on multi-class data sets

数据集	OAO							
	AS (高斯核)				NBAS (高斯核)			
	(C, γ)	ATT (s)	ATA (%)	DEV (%)	(C, γ)	ATT (s)	ATA (%)	DEV (%)
Iris	$(10, 0.8)$	0.0419	95.56	2.34	$(10, 1)$	0.0210	95.76	2.21
Segment	$(10^3, 0.8)$	0.9176	96.11	0.51	$(500, 1)$	0.6654	96.45	0.56
Satimage	$(10, 2)$	368.3	91.90	0	$(1, 1)$	335.6	92.00	0
Wine	$(1, 1)$	0.0580	98.08	1.18	$(10^2, 1)$	0.0229	97.46	1.44
Glass	$(10^2, 2)$	0.1894	65.96	3.65	$(10^2, 1)$	0.0729	67.37	4.18

表 8 多类数据集 OAA 方法比较

Table 8 OAA method comparison on multi-class data sets

数据集	OAA							
	(C, γ)	AS (高斯核)			NBAS (高斯核)			
		ATT (s)	ATA (%)	DEV (%)	(C, γ)	ATT (s)	ATA (%)	DEV (%)
Iris	(10, 0.8)	0.0529	95.52	2.70	(10, 1)	0.0341	95.60	2.53
Segment	$(10^2, 0.8)$	4.5028	96.03	0.73	$(10^2, 1)$	4.6252	96.45	0.47
Satimage	(1, 2)	650.1	91.35	0	(1, 1)	520.3	92.00	0
Wine	(1, 0.8)	0.0884	97.77	1.47	(1, 1)	0.0520	97.82	1.52
Glass	(10, 1)	0.1429	65.26	3.50	(10, 1)	0.1007	66.53	3.30

了 300 个参数对来寻找最佳测试精度, 但是该测试精度很可能不是实际上最好的测试精度。

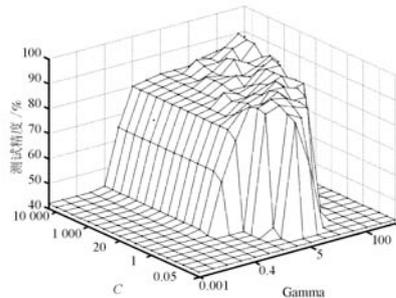


图 4 AS 算法在 Pwlinear 上的测试精度

Fig. 4 Testing accuracy of AS algorithm on Pwlinear

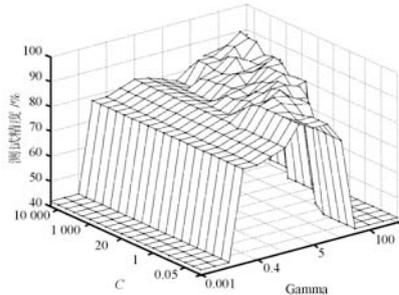


图 5 NBAS 算法在 Pwlinear 上的测试精度

Fig. 5 Testing accuracy of NBAS algorithm on Pwlinear

如需寻找更好的测试精度, 需要更多的参数对来进行测试. 而选取更多的参数对必然会增加计算代价, 尤其是在大数据集应用时. 在核参数选定的情况下, 由于 NBAS 对参数 C 不敏感, 假定 NBAS 在 C 为 0.2 和 0.3 的时候测试精度都能达到 90%, AS 在 C 为 0.2 的时候测试精度为 89.5%, 在 C 为 0.3 的时候测试精度为 90%. 由于我们选取 C 的可能值对中只有 0.2, 这就造成 AS 的最优测试精度只能达到 89.5%. 由此可见, 算法对参数不敏感的特性可以使算法有更大的几率找到实际最佳测试精度。

在训练精度上 (图 6 和图 7), NBAS 算法在所有参数对上都是 100%, 即无论参数选择多少, 算

法都能将训练样本分类正确. 由 Vapnik 理论^[1] 可知, SVM 可以看作是一种特殊的单隐藏层前馈神经网络 (Single hidden layer feedforward neural network, SLFN), 利用核函数代替了神经网络中的激活函数. 文献 [22–23] 指出, SLFN 的学习能力只和隐藏层节点的数目有关, 和输入层的权值无关, 并且最多只需一个隐藏层节点就可以完全拟合 N 个训练样本. 核函数的思想是将输入样本 x 映射到高维空间 H 中, H 的维数一般远大于 N ^[24], 从理论上无 b SVM 应该在任何参数选择下都能将训练样本完全分类正确, 而图 7 正好验证了这个性质。

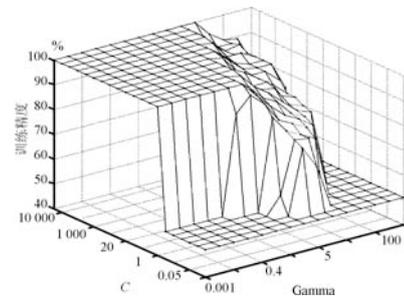


图 6 AS 算法在 Pwlinear 上的训练精度

Fig. 6 Training accuracy of AS algorithm on Pwlinear

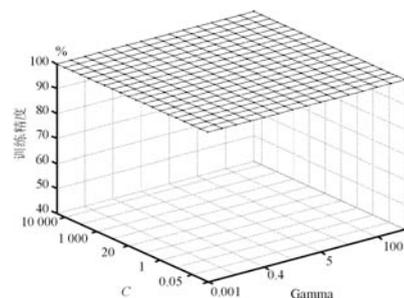


图 7 NBAS 算法在 Pwlinear 上的训练精度

Fig. 7 Training accuracy of NBAS algorithm on Pwlinear

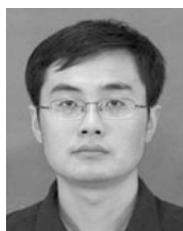
4 结论

本文主要讨论和分析了 b 对 SVM 分类问题的泛化性能的影响. 给出了无 b SVM 的优化问题并提出了相应的有效集算法, 与同目标 SVM 优化问题相比 (优化问题相同, 约束条件不同), 在解空间上无

b SVM 能够得到比 SVM 更优的解. 在实验中验证了无 b SVM 网络结构和 SLFN 是一致的, 在高维空间上能够将训练样本完全分类正确. 在测试样本性能分析中, 无 b SVM 的测试精度普遍要高于 SVM, 而且在所有数据集上的计算代价都要低于 SVM. 由理论分析和实验可见, 在常用的高斯核和多项式核上, 无 b SVM 分类器具有更好的泛化性能和减少训练时的计算代价. 由于 SVM 分类问题现已大量应用在主动学习和半监督学习等领域, 本文将来的研究工作将验证无 b SVM 在这些相关领域的有效性. 此外, 求解算法是优化问题的核心, 而无 b SVM 优化问题是一个较新的课题, 本文后续的工作将针对无 b SVM 优化问题, 研究出更高效的求解算法.

References

- Cortes C, Vapnik V. Support vector networks. *Machine Learning*, 1995, **20**(3): 273–297
- Poggio T, Mukherjee S, Rifkin R, Verri A. Technical Report A.I.Memo.No.2001-011, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA, 2001
- Lu F Y, Sun H W. Positive definite dot product kernels in learning theory. *Advances in Computational Mathematics*, 2005, **22**(2): 181–198
- Evgeniou T, Pontil M, Poggio T. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 2000, **13**(1): 1–50
- Kecman V, Vogt M, Huang T M. On the equality of kernel AdaTron and sequential minimal optimization in classification and regression tasks and alike algorithms for kernel machines. In: Proceedings of the 11th European Symposium on Artificial Neural Networks. Bruges, Belgium: IEEE, 2003. 215–222
- Huang G B, Ding X J, Zhou H M. Optimization method based extreme learning machine for classification. *Neurocomputing*, 2010, **74**(1–3): 155–163
- Liu Y, Wang H Q, Yu J, Li P. Selective recursive kernel learning for online identification of nonlinear systems with NARX form. *Journal of Process Control*, 2010, **20**(2): 181–194
- Ding Xiao-Jian, Zhao Yin-Liang. Sequential minimal optimization method based on optimization ELM. *Journal of Xi'an Jiaotong University*, 2011, **45**(6): 7–12 (丁晓剑, 赵银亮. 优化极限学习机的序列最小优化方法. 西安交通大学学报, 2011, **45**(6): 7–12)
- Yang Wu-Yi, Liang Wei, Xin Le, Zhang Shu-Wu. Subspace semi-supervised fisher discriminant analysis. *Acta Automatica Sinica*, 2009, **35**(12): 1513–1519
- Zhang Yu-Nong, Chen Zeng-Hai, Chen Ke. Convergence properties analysis of gradient neural network for solving online linear equations. *Acta Automatica Sinica*, 2009, **35**(8): 1136–1139
- Qiao Jun-Fei, Han Hong-Gui. Optimal structure design for RBFNN structure. *Acta Automatica Sinica*, 2010, **36**(6): 865–872 (乔俊飞, 韩红桂. RBF 神经网络的结构动态优化设计. 自动化学报, 2010, **36**(6): 865–872)
- Fletcher R. *Practical Methods of Optimization (Volume 2): Constrained Optimization*. New York: John Wiley and Sons, 1981
- Scheinberg K. An efficient implementation of an active set method for SVMs. *The Journal of Machine Learning Research*, 2006, **7**(10): 2237–2257
- Frank A, Asuncion A. UCI machine learning repository [Online], available: <http://www.ics.uci.edu/~mllearn/>, November 12, 2009
- Srinivasan A. UCI machine learning repository: statlog (landsat satellite) data set [Online], available: [http://archive.ics.uci.edu/ml/datasets/Statlog+\(Landsat+Satellite\)](http://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)), July 1, 2010
- Li J Y, Liu H Q. Kent ridge bio-medical dataset [Online], available: <http://datam.i2r.a-star.edu.sg/datasets/krbd/>, August 15, 2010
- Platt J C. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*. Cambridge: The MIT Press, 1999. 185–208
- Ghanty P, Paul S, Pal N R. NEUROSVN: an architecture to reduce the effect of the choice of kernel on the performance of SVM. *The Journal of Machine Learning Research*, 2009, **10**(3): 591–622
- Ling S H, Leung F H F, Lam H K, Lee Y S, Tam P K S. A novel genetic-algorithm-based neural network for short-term load forecasting. *IEEE Transactions on Industrial Electronics*, 2003, **50**(4): 793–799
- Golub T R, Slonim D K, Tamayo P, Huard C, Gaasenbeek M, Mesirov J P. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 1999, **286**(5439): 531–537
- Hsu C W, Lin C J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 2002, **13**(2): 415–425
- Huang G B, Babri H A. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, 1998, **9**(1): 224–229
- Huang G B. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 2003, **14**(2): 274–281
- Vapnik V N. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995



丁晓剑 西安交通大学电子与信息工程学院博士研究生. 主要研究方向为神经网络和机器学习. 本文通信作者.

E-mail: xjdj@stu.xjtu.edu.cn

(DING Xiao-Jian Ph.D. candidate at the School of Electronics and Information Engineering, Xi'an Jiaotong University. His research interest covers neural networks and machine learning. Corresponding author of this paper.)



赵银亮 西安交通大学电子与信息工程学院教授. 主要研究方向为语言及编译系统、并行计算与机器学习.

E-mail: zhaoy@mail.xjtu.edu.cn

(ZHAO Yin-Liang Professor at the School of Electronics and Information Engineering, Xi'an Jiaotong University. His research interest covers language design, compiler systems, parallel computing, and machine learning.)