

无偏置支持向量回归优化问题*

丁晓剑¹⁺, 赵银亮²

¹(信息系统工程重点实验室, 江苏 南京 210007)

²(西安交通大学 电子与信息工程学院, 陕西 西安 710049)

Support Vector Regression Optimization Problem without Bias

DING Xiao-Jian¹⁺, ZHAO Yin-Liang²

¹(Science and Technology on Information Systems Engineering Laboratory, Nanjing 210007, China)

²(School of Electrical and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

+ Corresponding author: E-mail: wjswsl@163.com

Ding XJ, Zhao YL. Support vector regression optimization problem without bias. *Journal of Software*, 2012, 23(9): 2336-2346 (in Chinese). <http://www.jos.org.cn/1000-9825/4150.htm>

Abstract: To study the role of bias in support vector regression (SVR), primal and dual optimization formulations of support vector regression optimization problem without bias (NBSVR) are proposed first, and the necessary condition of NBSVR optimization formulation's global optima is presented and sub-optima solution of NBSVR dual problem has been proved for the dual problem of SVR then. An active set algorithm of dual optimization formulation without bias is proposed, and the linear convergence of the proposed algorithm has been proved. The experimental results on 21 benchmark datasets show that in the solution space of dual problem, SVR can only obtain the sub-optimal solution of NBSVR, the root mean square error (RMSE) of NBSVR tends to lower than SVR. The training time of NBSVR is not only less than SVR, but also less sensitive to kernel parameter.

Key words: bias; support vector regression; active set; generalization ability

摘要: 为了研究偏置对支持向量回归(support vector regression, 简称 SVR)问题泛化性能的影响, 首先提出了无偏置 SVR(NBSVR)的优化问题及其对偶问题. 推导出了 NBSVR 优化问题全局最优解的必要条件, 然后证明了 SVR 的对偶问题只能得到 NBSVR 对偶问题的次优解. 同时提出了 NBSVR 的有效集求解算法, 并证明了它是线性收敛的. 基于 21 个标准数据集的实验结果表明, 在对偶问题解空间上, 有偏置支持向量回归算法只能得到无偏置支持向量回归算法的次优解, NBSVR 的均方根误差要低于 SVR. NBSVR 的训练时间不仅低于 SVR, 而且对核参数变化不太敏感.

关键词: 偏置; 支持向量回归; 有效集; 泛化性能

中图法分类号: TP18 **文献标识码:** A

1995 年, Cortes 和 Vapnik 提出了可高效处理分类和回归问题的支持向量网络^[1], 称为支持向量机(support vector machine, 简称 SVM). SVM 的核心思想是最大化分类/回归间隔, 即最小化 $\|w\|$, 使决策超平面的间隔最大

* 基金项目: 国家自然科学基金(61173040)

收稿时间: 2010-08-17; 修改时间: 2011-01-31; 定稿时间: 2011-11-02

化,并通过调节偏置 b 得到最优的决策超平面: $\langle w, x \rangle + b = 0$. 相比传统的分类和回归算法, SVM 的学习性能有了大幅提高,但是,文献[1]在理论上并未指出偏置 b 在决策超平面存在的意义. 另一方面,无偏置的决策超平面在神经网络的算法应用中得到了很好的泛化性能,如文献[2]提出的正则极限学习机(regularized extreme learning machine,简称 RELM)和文献[3]提出的基于极限学习机的优化方法. 所以, b 对 SVM 回归问题泛化性能的影响是一个值得研究的问题.

训练 SVM 的核心问题求解其优化问题,自从 SVM 提出以后,有很多的方法被提出以求解其优化问题. 直接的方法是求解其原始优化问题^[4],更多的间接方法则是求解其优化问题^[5-12],间接的方法又可以分为两种:1) 分解法^[6-8]. 文献[12]证明了 SVM 优化问题可以分解为一系列子问题来求解,分解法的特点是每次迭代所需求解的子问题的变量个数是相同的,其中,训练回归 SVM 的 SMO(sequential minimal optimization)方法^[8]是一种极端分解方法,每次迭代只需求解含有两个工作变量的子优化问题. 文献[13,14]从工作集选择策略角度对 SMO 的回归问题作了改进,使得 SMO 在拟合回归数据时运算速度更快、精度更高. 但是,SMO 在处理回归优化问题时,每次迭代都需要正确更新 b ,以保证下次迭代正常进行,如果每次迭代后两个工作变量更新后的值都在边界约束上时,那么正确的 b 值在一个范围内,无法精确计算出. 文献[15]通过减少对偶间隙有效地缩小了 b 的取值范围,但是仍然需要近似估计 b 的值;2) 有效集(active set)法^[9-11]. 有效集方法通过一系列迭代识别出优化问题的正确有效集,在迭代过程中,有效集的元素个数是递增的. 有效集方法和 SMO 方法的不同在于,每次迭代中 b 值都可以精确计算出来,从而避免由 b 值计算失误而导致误差的累积.

本文使用有效集方法来比较支持向量回归(support vector regression,简称 SVR)问题和无偏置 SVR(no-bias support vector regression,简称 NBSVR)的泛化性能. 首先提出了 NBSVR 原始优化问题及其对偶问题,从理论上给出了 NBSVR 最优解和 SVR 最优解的分析比较,给出了 NBSVR 的有效集求解方法,并与 SVR 进行多个指标的实验比较.

本文第 1 节分析 NBSVR 原始优化问题和对偶优化问题. 第 2 节对 SVR 和 NBSVR 的对偶优化问题进行理论分析. 第 3 节给出有效集算法描述,算法的收敛性分析和初始迭代向量分析. 第 4 节给出 SVR 和 NBSVR 在 21 个标准数据集上的测试比较结果. 第 5 节是结论.

1 NBSVR 优化问题

给定一个含有 N 个样本的训练样本集 $\{x_i, y_i\}_{i=1}^N$, 其中,输入为 d 维向量 $x_i \in R^d$, 输出为 $y_i \in R$. SVR 硬间隔约束条件为

$$|y_i - \langle w \cdot x_i \rangle - b| \leq \varepsilon \quad (1)$$

其中, ε 为用户设定的参数,表示决策函数对所有样本预测的输出值与目标值 y_i 最多允许有 ε 的偏差. 如果公式(1)不满足,SVR 硬间隔优化问题是无解的,这时可引入松弛变量 ξ_i 和 ξ_i^* 放宽约束条件来解决这个问题. 由于 NBSVR 的约束条件没有偏置项, NBSVR 优化问题表示为

$$\left. \begin{aligned} & \text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \\ & \text{subject to } y_i - \langle w \cdot x_i \rangle \leq \varepsilon + \xi_i, \langle w \cdot x_i \rangle - y_i \leq \varepsilon + \xi_i^*, \xi_i \geq 0, \xi_i^* \geq 0 \end{aligned} \right\} \quad (2)$$

构造公式(2)的拉格朗日函数:

$$L_1 := \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) - \sum_{i=1}^N (\eta_i \xi_i + \eta_i^* \xi_i^*) - \sum_{i=1}^N \alpha_i (\varepsilon + \xi_i - y_i + \langle w \cdot x_i \rangle) - \sum_{i=1}^N \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w \cdot x_i \rangle) \quad (3)$$

其中, $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$ 为非负拉格朗日乘子:

$$\alpha_i \geq 0, \alpha_i^* \geq 0, \eta_i \geq 0, \eta_i^* \geq 0 \quad (4)$$

对 L_1 关于向量 w 和变量 ξ_i, ξ_i^* 求偏导数,下列条件都满足时得到公式(3)的最优解:

$$\partial L_1 / \partial w = 0 \Rightarrow w = \sum_{i=1}^N (\alpha_i - \alpha_i^*) x_i \quad (5)$$

$$\partial L_1 / \partial \xi_i = 0 \Rightarrow C = \alpha_i + \eta_i \quad (6)$$

$$\partial L_1 / \partial \xi_i^* = 0 \Rightarrow C = \alpha_i^* + \eta_i^* \quad (7)$$

将公式(5)~公式(7)代入公式(3)可得对偶优化问题:

$$\left. \begin{aligned} & \text{minimize } \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle + \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) - \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ & \text{subject to } 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, \forall i \end{aligned} \right\} \quad (8)$$

通常,训练样本集在原始输入空间中是线性不可分的,需要引入映射函数 ϕ 将 x_i 映射为特征向量 $\phi(x_i)$,以使样本在特征空间中线性可分.从对偶优化问题(8)可以看出,只需计算样本的内积形式就可以求解出 NBSVR 的最优回归超平面,只需找到某核函数 K 使得 $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$,从而避免计算高维向量 $\phi(x_i)$.

由公式(5)可知,NBSVR 的决策函数可表示为

$$h(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x_i, x) \quad (9)$$

其中, $K(x_i, x_j)$ 为满足 Mercer 定理的核函数.

2 理论分析

为了方便推导,公式(8)的目标函数可作如下表示:

$$\begin{aligned} & \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) + \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) - \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) = \\ & \frac{1}{2} \sum_{i,j=1}^N (\alpha_i \alpha_j - \alpha_i \alpha_j^* - \alpha_i^* \alpha_j + \alpha_i^* \alpha_j^*) K(x_i, x_j) - \sum_{i=1}^N (-\varepsilon + y_i) \alpha_i + (-\varepsilon - y_i) \alpha_i^* \end{aligned} \quad (10)$$

引入向量 $\delta \in \mathbb{R}^{2N}$ 和 $q \in \mathbb{R}^{2N}$,矩阵 $H \in \mathbb{R}^{2N \times 2N}$,公式(8)可简化为

$$\left. \begin{aligned} & \text{minimize } f(\delta) = \frac{1}{2} \delta^T H \delta - q^T \delta \\ & \text{subject to } 0 \leq \delta_i \leq C, \forall i \end{aligned} \right\} \quad (11)$$

其中, $\delta = [\alpha; \alpha^*]$, $H = \begin{bmatrix} K & -K^T \\ -K & K \end{bmatrix}$, $q = [-\varepsilon + y; -\varepsilon - y]^T$, $\varepsilon = [\varepsilon, \dots, \varepsilon]^T \in \mathbb{R}^N$, $y = [y_1, \dots, y_N]^T \in \mathbb{R}^N$.

2.1 Karush-Kuhn-Tucker条件

对于二次规划问题,判断向量 δ^* 是否为它的最优解的充分必要条件为 δ^* 是否为公式(11)目标函数的 Karush-Kuhn-Tucker(KKT)点^[16].基于 KKT 理论,构造公式(11)的拉格朗日问题:

$$L_2 := \frac{1}{2} \delta^T H \delta - q^T \delta - \sum_{i=1}^N \rho_i \delta_i - \sum_{i=1}^N \rho_i^* (C - \delta_i) \quad (12)$$

其中, ρ_i 和 ρ_i^* 为非负拉格朗日乘子.令 $g(\delta)$ 为公式(11)的梯度,即 $g(\delta) = H\delta - q$,则有 $g_i(\delta) = [H\delta - q]_i$,其中, $g_i(\delta)$ 为向量 $g(\delta)$ 的第 i 个元素.根据 KKT 理论,公式(12)的最优解必须满足下列条件:

- (1) $\partial L_2 / \partial \delta_i = 0 \Rightarrow g_i(\delta) = \rho_i - \rho_i^*$;
- (2) $0 \leq \delta_i \leq C, \forall i$;
- (3) $\rho_i \geq 0, \forall i$;
- (4) $\rho_i^* \geq 0, \forall i$;
- (5) $\rho_i \delta_i = 0, \forall i$;
- (6) $\rho_i^* (C - \delta_i) = 0, \forall i$.

其中,条件(1)为稳定点条件,条件(2)为原问题(11)的可行性条件,条件(3)和条件(4)为公式(12)的可行性条件,条件(5)和条件(6)为公式(11)和公式(12)之间的互补松弛性条件.由这 6 个条件可对 δ_i 进行如下划分:

- 1) $\delta_i = 0$.由条件(6)可知 $\rho_i^* = 0$,再由条件(3)可得 $g_i(\delta) \geq 0$;

- 2) $\delta_i=C$.由条件(5)可知 $\rho_i=0$,再由条件(4)可得 $g_i(\delta)\leq 0$;
 3) $0<\delta_i<C$.由条件(5)和条件(6)可知 $\rho_i=0$, $\rho_i^*=0$. 可得 $g_i(\delta)=0$.

因此,当且仅当以下条件成立时,公式(12)才会得到最优解:

$$\begin{cases} \delta_i = 0 & \Leftrightarrow g_i(\delta) \geq 0 \\ 0 < \delta_i < C & \Leftrightarrow g_i(\delta) = 0 \\ \delta_i = C & \Leftrightarrow g_i(\delta) \leq 0 \end{cases} \quad (13)$$

2.2 SVR和NBSVR最优解分析

与 NBSVR 对偶问题相比,SVR 的对偶问题具有相同的目标函数,但是约束条件不同:

$$\begin{cases} \text{minimize } f(\delta) = \frac{1}{2} \delta^T H \delta - q^T \delta \\ \text{subject to } \sum_1^N \delta_i = \sum_{N+1}^{2N} \delta_j, 0 \leq \delta_i \leq C, \forall i \end{cases} \quad (14)$$

定义 1. 满足某优化问题约束条件的点称为该优化问题的可行点,由全部可行点组成的集合称为该优化问题的可行域,公式(11)和公式(14)的可行域分别记为 D_1 和 D_2 ,即

$$\begin{aligned} D_1 &= \{\delta \mid 0 \leq \delta_i \leq C, i \in E \cup I\}, \\ D_2 &= \{\delta \mid \sum \delta_i = \sum \delta_j, i \in E, j \in I, 0 \leq \delta_i \leq C, i \in E \cup I\}, \end{aligned}$$

其中, $E=\{1, \dots, N\}$, $I=\{N+1, \dots, 2N\}$.

定义 2. 若存在 $\delta^* \in D_1$,使得对任意的 $\delta \neq \delta^*$, $\delta \in D_1$ 均有 $f(\delta) \geq f(\delta^*)$,则称 δ^* 是 $f(\delta)$ 在可行域 D_1 上的全局最优解.

由于 SVR 和 NBSVR 采用满足 Mercer 定理的函数作为核函数,所以公式(11)和公式(14)的目标函数相应的矩阵 H 为半正定矩阵,它们对应的优化问题为凸二次规划问题,即存在全局最优解.

定理 1. 令目标函数 $f(\delta)$ 在可行域 D_1 的全局最优解为 $f(\delta^1)$, $f(\delta)$ 在可行域 D_2 的全局最优解为 $f(\delta^2)$, 则 $f(\delta^1) \leq f(\delta^2)$.

证明:由定义 1 可知, $D_1 \supseteq D_2$. 如果 $\delta^1 \in D_2$, 则表示公式(11)的可行域和公式(14)的可行域相同;又由于这两个优化问题的目标函数是一致的,此时公式(11)和公式(14)完全一致,必然得到相同的全局最优解,即 $f(\delta^1) = f(\delta^2)$. 如果 $\delta^1 \notin D_2$, 即 $\delta^1 \in D_1 \setminus D_2$, 而 $\delta^2 \in D_2$, 所以 $\delta^2 \neq \delta^1$. 由定义 2 可知, $f(\delta^2) \geq f(\delta^1)$. 综上,有 $f(\delta^1) \leq f(\delta^2)$ 成立. \square

定理 1 表明,在求解相应的对偶问题时,SVR 只能得到 NBSVR 的次优解.从另一方面看,公式(11)是在多维空间 $[0, C]^{2N}$ 中寻找最优解,而公式(14)是在多维空间 $[0, C]^{2N}$ 中沿着超平面 $\sum \delta_i = \sum \delta_j, i \in E, j \in I$ 寻找最优解,显然,公式(11)的解空间要大于公式(14)的解空间.

3 NBSVR 优化问题的有效集方法

框式约束二次规划问题从数学优化角度看已做过很多研究,一般使用内点法和有效集方法.内点法的优点是迭代次数很少,计算复杂度对约束条件数目不敏感,适合处理稀疏矩阵,但是每次迭代的计算代价较大.有效集方法的每次迭代计算代价很小,但是迭代次数很多,与约束条件数目关系很大.

相比一般的框式约束问题,公式(11)存在两种特性:1) 由核矩阵构成的矩阵 H 为稠密矩阵;2) 下界约束都为 0.这两种特性表明,公式(11)更适合用有效集方法求解.由于下界约束为 0,有效集算法可以使每次迭代的计算代价进一步降低,并能降维求解目标优化问题.

3.1 NBSVR 有效集算法

有效集方法是一种迭代求解方法,每次迭代完成时需要判断迭代得到的向量是否满足目标问题的 KKT 条件^[16]:如果满足,则算法得到最优解;否则,继续进行迭代操作.

对公式(11)引入 3 个指标集合 L, U, S , 并定义: $L := \{i \mid \delta_i = 0\}$, $S := \{i \mid 0 < \delta_i < C\}$, $U := \{i \mid \delta_i = C\}$. 因为 $\delta \in R^{2N}$, 所以

$L \cup S \cup U = \{1:2N\}$. 并称 L 和 U 为有效集, S 为工作集. δ_i 为 0 称为下边界约束, δ_i 为 C 称为上边界约束.

根据指标集的划分策略, 公式(11)的目标函数中变量和矩阵可作如下划分: $\delta = [\delta_L, \delta_S, \delta_U]^T$, $q = [q_L, q_S, q_U]^T$, 其中, 向量 δ_L, δ_S 和 δ_U 分别由指标集 L, U, S 标识, 向量 q_L, q_S 和 q_U 分别由集合 L, U, S 标识. 同理可定义矩阵 H 的子矩阵, 如 H_{LL}, H_{LS} 和 H_{LU} , 矩阵的行分别由集合 L, L, L 标识, 矩阵的列分别由集合 L, S, U 标识.

因为 δ_i 值为 0 时对目标函数的函数值计算没有影响, 所以公式(11)可简化为下面的形式:

$$\begin{aligned} & \underset{\delta}{\text{minimize}} \quad \frac{1}{2} \delta_S^T H_{SS} \delta_S + \delta_U^T H_{US} \delta_S - q_S^T \delta_S + \frac{1}{2} \delta_U^T H_{UU} \delta_U - q_U^T \delta_U \\ & \text{subject to} \quad 0 \leq \delta_{S_j} \leq C, \forall j \end{aligned} \quad (15)$$

其中, δ_{S_j} 为 δ_S 的第 j 个元素. 由 δ_S 的定义可知, 公式(14)的约束条件必然满足, 求解公式(15)等价于求解无约束优化问题:

$$\underset{\delta_S}{\text{minimize}} \quad \frac{1}{2} \delta_S^T H_{SS} \delta_S + \delta_U^T H_{US} \delta_S - q_S^T \delta_S + \frac{1}{2} \delta_U^T H_{UU} \delta_U - q_U^T \delta_U \quad (16)$$

由于 δ_U 中元素的值都固定为 C , 所以公式(16)可以看作是关于 δ_S 求最小值.

NBSVR 有效集算法利用 2 层循环来求解目标优化问题. 首先, 外循环判断 δ 中边界约束元素是否都满足公式(13), 如果都满足, 算法终止; 如果不满足, 按照下式选取一个违反 KKT 条件程度最大的指标:

$$\arg \max_i \{ |g_i(\delta)|, g_i(\delta) < 0, \forall i \in L \text{ or } g_i(\delta) > 0, \forall i \in U \} \quad (17)$$

根据公式(17)选择的指标 i , 将对应的元素 δ_i 加入 δ_S 中, 使得 $\delta_S = [\delta_S; \delta_i]$; 再更新相应的指标集 S , 使得 $S = S + \{i\}$. 如果 $\delta_i = C$, 需从 δ_U 中删除一个元素, 再更新相应的指标集 U , 使得 $U = U - \{i\}$. 然后进入内循环. 给定初始可行迭代向量 δ^k ($\delta^k \in [0, C]^{2N}$), 初始有效集 L^k, U^k , 工作集 S^k , 根据 δ^k, L^k, U^k 和 S^k 给出子向量 δ_S^k 和 δ_U^k , NBSVR 有效集外循环算法可描述如下:

算法 1. NBSVR 有效集外循环算法.

第 1 步: 判断 δ^k 是否满足公式(13)的约束, 如果满足, 算法终止; 如果不满足, 执行下列操作.

第 2 步: 根据公式(17)选择一个 δ_i^k , 令 $\delta_S^k = [\delta_S^k; \delta_i^k]$, $S^k = S^k + \{i\}$.

第 3 步: 如果 $\delta_i^k = 0$, 令 $L^k = L^k - \{i\}$; 否则, 令 $U^k = U^k - \{i\}$, 将 $\delta_{U_i}^k$ 从 δ_U^k 中删除.

第 4 步: 跳转到内循环算法(算法 2).

内循环的目标是求解公式(16), 使得最优解 δ_S 满足上下界约束. 由于 H 是对称半正定矩阵, 它的子矩阵 H_{SS} 也是对称半正定的, 因此, 公式(15)关于 δ_S 存在全局最优解. 公式(16)的最优解可通过式(18)直接求出

$$\delta_S^{new} = -(H_{SS})^{-1} (H_{US}^T \delta_U - q_S)^T \quad (18)$$

如果 δ_S^{new} 中的元素都在约束界内, 即 $\delta_{S_j}^{new} \in (0, C), \forall j$, 则终止内循环, 进行外循环操作; 否则, 将 δ_S 的变化值(记作 $d = \delta_S^{new} - \delta_S$)乘以步长 sig , 设使得 δ_S 中只有一个元素 δ_{S_j} 在界上. sig 通过式(19)计算:

$$sig = \min \left\{ \min_{l, d_l < 0} \{-\delta_{S_l} / d_l\}, \min_{l, d_l > 0} \{C - \delta_{S_l} / d_l\} \right\} \quad (19)$$

如果 δ_S^{new} 越界, 先计算 $sig_1 = \min_{l, d_l < 0} \{-\delta_{S_l} / d_l\}$ 和 $sig_2 = \min_{l, d_l > 0} \{C - \delta_{S_l} / d_l\}$, 分别找出使 δ_S^{new} 上界和下界元素满足边界的最短步长 sig_1 和 sig_2 , 再计算 $sig = \min\{sig_1, sig_2\}$, $\delta_S = \delta_S + sig \cdot d$, 使得 δ_S 只有一个元素在边界上, 其余元素都回缩到边界内. 找出 δ_S 在边界上的元素 δ_{S_j} , 将 δ_{S_j} 从 δ_S 中删除, 并将相应的指标 i 从集合 S 中删除, 即 $S = S - \{i\}$. 如果 $\delta_{S_j} = 0$, 需将相应的 i 加入到集合 L 中, 即 $L = L + \{i\}$; 如果 $\delta_{S_j} = C$, 需将相应的指标 i 加入到集合 U 中, 即 $U = U + \{i\}$. 此外, 还需将边界约束加入到 δ_U 中, 即 $\delta_U = [\delta_U; C]$. 然后, 继续使用 δ_S 和 δ_U 求解公式(16), 直到求出的解 δ_S^{new} 所有元素都在约束界内. 由上所述, NBSVR 有效集内循环算法可描述如下:

算法 2. NBSVR 有效集内循环算法.

第 1 步: 根据公式(18)计算得到 δ_S^{new} .

第 2 步: 判断 δ_S^{new} 中的所有元素是否都满足 $\delta_{S_j}^{new} \in (0, C)$: 如果满足, 执行第 3 步; 如果不满足, 执行第 5 步.

第 3 步: $\delta_S^{k+1} = \delta_S^{new}, \delta_U^{k+1} = \delta_U^k, S^{k+1} = S^k, L^{k+1} = L^k, U^{k+1} = U^k$,
 令 $\delta^{k+1} = [0, \dots, 0]^T \in R^{2N}, \delta_S^{k+1}(S^{k+1}) = \delta_S^{k+1}, \delta_U^{k+1}(U^{k+1}) = \delta_U^{k+1}, k=k+1$.

第 4 步:跳转到外循环算法(算法 1).

第 5 步:令 $d^k = \delta_S^{new} - \delta_S^k$,根据公式(19)计算 sig .

第 6 步:令 $\delta_S^{k+1} = \delta_S + sig \cdot d^k$,找出 δ_S^{k+1} 在边界上的元素 $\delta_{S_j}^{k+1}$.将 $\delta_{S_j}^{k+1}$ 从 δ_S^{k+1} 中删除.

第 7 步:令 $S^k = S^k - \{i\}$,其中, i 为 S^k 中的第 j 个元素.

第 8 步:如果 $\delta_{S_j}^{k+1} = 0$,令 $L^{k+1} = L^k + \{i\}, \delta_U^{k+1} = \delta_U^k, U^{k+1} = U^k + \{i\}$;

否则,令 $U^{k+1} = U^k + \{i\}, \delta_U^{k+1} = [\delta_U^k; C], L^{k+1} = L^k, k=k+1$.

第 9 步:转第 1 步.

3.2 收敛性分析

本节主要对 NBSVR 有效集算法的收敛性进行分析.令 $A=L \cup U$ 为公式(11)的有效集, δ^* 为公式(11)的最优解, A^* 为公式(11)得到最优解 δ^* 时对应的有效集. δ^k 为 NBSVR 有效集算法第 k 次迭代得到的解, $p(\delta^k)$ 为公式(11)第 k 次迭代得到的函数值.

定理 2. 令 $\{A^k\}$ 为一组有效集序列, $\{\delta^k\}$ 为一组 N 维向量序列, 则 NBSVR 有效集算法在有限步内线性收敛到 δ^* .

证明:由于 $0 \leq |A| \leq N$, 所以由 $1 \sim N$ 之间所有整数构成的可能的子列 $\{A^k\}$ 必定是有限的. 由致密性定理可知, 任意初始有效集 A^k 必然能在有限次迭代中得到 A^* , 从而算法能够在有限步内收敛到 δ^* . 此外, 计算可以得到由内循环选择的步长满足 $0 < sig \leq 1$, 公式(18)计算得到的解必然保证公式(16)的无约束优化问题单调下降; 又因为求解公式(16)等价于求解公式(11), 所以第 $K+1$ 迭代得到的解 δ^{k+1} 必然满足 $p(\delta^{k+1}) < p(\delta^k)$. 即 NBSVR 有效集算法是单调下降的, 所以 NBSVR 有效集算法有限步内线性收敛到 δ^* . \square

3.3 初始迭代向量分析

由 NBSVR 有效集算法可知, 每次迭代的计算代价与工作集对应的变量个数有较大的关系. 变量个数越多, 计算代价也就越大, 所以初始迭代向量的工作变量一般设置比较少. 由第 3.1 节可知, 集合 U 对应的元素为边界支持向量(bound support vector, 简称 BSV), 集合 S 对应的元素为非边界支持向量(non-bound support vector, 简称 NBSV). 不同数据集的边界支持向量在优化问题所有变量中的比例是不同的. 对边界支持向量比例较大的数据集, 初始有效集变量都设为上界约束的变量可以有效改善算法的性能. 对边界支持向量比例较小的数据集, 初始有效集变量都设为下界约束的变量可以有效改善算法的性能.

4 实验与性能分析

本节通过实验分析 NBSVR 和 SVR 的泛化性能. NBSVR 的二次规划问题采用提出的 NBSVR 有效集算法求解, SVR 的二次规划优化问题采用文献[9]提出的有效集方法求解(下文两种算法简称为 SVR 算法和 NBSVR 算法). 两种优化问题的核函数均选择高斯核函数: $K(u, v) = \exp(-\|u - v\|^2 / 2\gamma^2)$. 所有算法都是基于 Pentium 4, 2.53GHZ CPU, MATLAB2007 环境下实现. 实验用到的数据来自 UCI 数据库^[17]和 StatLib 数据库^[18], 数据的输入和输出都归一化到 $[0, 1]$ 之间, 数据的具体描述见表 1. 其中, 编号 1~14 的数据为 Statlib 数据集, 编号 15~21 的数据为 UCI 数据集.

有些数据的属性是由字符或字符串组成, 需要作相关预处理将其变为数字属性. Abalone 数据有一个属性 'sex' 是由 M, F 和 I 组成, 分别用 1, 2, 3 表示 M, F, I. Cloud 数据有两个属性 'seeded' 和 'season' 分别由字符和字符串组成, 需要将字符串转化为二进制表示形式. 如, 'season' 性为 AUTUMN, WINTER, SPRING 和 SUMMER, 则 AUTUMN 可表示为 1, 0, 0, 0, WINTER 表示为 0, 1, 0, 0, SPRING 表示为 0, 0, 1, 0, SUMMER 表示为 0, 0, 0, 1. 经过预处理后, Cloud 数据属性由 6 个增加到 9 个, Machine_cpu 数据属性由 7 个增加到 36 个, Servo 数据属性由 4 个增加

到 19 个.

Table 1 Benchmark data sets

表 1 标准数据集

编号	数据集	训练样本数	测试样本数	属性数
1	Babyfat	130	122	14
2	Balloon	1 000	1 001	2
3	Basketball	50	46	4
4	Bolts	20	20	7
5	Cleveland	150	153	13
6	Elusage	25	20	2
7	Fruitfly	60	65	4
8	Lowbwt	100	89	9
9	Mbgrade	30	31	2
10	Pollution	30	30	15
11	Quake	1 000	1 178	3
12	Sensory	300	276	11
13	Strike	300	325	6
14	Space-ga	1 500	1 607	6
15	Abalone	2 000	2 177	8
16	Autoprice	80	79	15
17	Cloud	50	58	9
18	Housing	250	256	13
19	Machine_cpu	100	109	36
20	Pyrim	40	34	27
21	Servo	80	87	19

4.1 用户参数选取

SVR 和 NBSVR 优化问题的 ε 设为 0.01, 由于 SVR 和 NBSVR 优化问题算法性能受代价参数 C 和核参数 γ 影响较大, 需要选择合适的参数值以使算法的泛化性能最好. 参照文献[19]中的参数选择方法, 对于每个数据集选取 15 个 C 值和 15 个 γ 值, 共有 225 种参数对, 并选择性能最好的组合 (C, γ) . 15 个 C 值分别取 $10^{-3}, 10^{-2}, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 10^2, 10^3$ 和 10^4 . 15 个 γ 值分别取 $10^{-3}, 10^{-2}, 0.1, 0.2, 0.4, 0.8, 1, 2, 5, 10, 20, 50, 10^2, 10^3$ 和 10^4 . 令 SVR 和 NBSVR 有效集算法初始迭代向量 $\delta^0 = [\delta_1^0, \dots, 0]^T \in R^{2N}$, 并设 $\delta_1^0 = C/2$, 即 δ^0 第 1 个元素值为 $C/2$, 其余的值为 0, 则对应的有效集和工作集分别为 $L^0 = \{2:2N\}, L^0 = \emptyset, S = \{1\}$.

4.2 算法性能比较

在此, 利用训练样本的训练时间(time)、测试样本的均方根误差(root mean square error, 简称 RMSE)、迭代次数(iteration)、NBSV 和 BSV 作为算法的评价指标. 表 2 包含了 Time 和 RMSE 的指标测试结果(两种算法的最优参数). 表 3 包含了 Iterations, BSV 和 BSV 这 3 个指标的测试结果.

Table 2 Training time and RMSE comparison of algorithms

表 2 算法训练时间和均方根误差比较

数据集	SVR			NBSVR		
	(C, γ)	Time (s)	RMSE	(C, γ)	Time (s)	RMSE
Babyfat	$(10^3, 20)$	0.093 1	0.010 5	$(10^3, 20)$	0.070 8	0.010 5
Balloon	$(2, 0.8)$	3.992 0	0.040 8	$(10, 1)$	3.879 5	0.039 4
Basketball	$(0.5, 0.8)$	0.066 8	0.133 3	$(10^{-2}, 1)$	0.040 3	0.133 2
Bolts	$(50, 5)$	0.030 3	0.114 1	$(50, 5)$	0.021 8	0.113 9
Cleveland	$(0.5, 2)$	0.161 2	0.227 4	$(2, 5)$	0.149 0	0.227 2
Elusage	$(50, 2)$	0.027 0	0.117 0	$(10^4, 5)$	0.017 3	0.115 3
Fruitfly	$(0.2, 1)$	0.078 6	0.181 6	$(50, 20)$	0.058 2	0.182 0
Lowbwt	$(0.2, 0.4)$	0.138 5	0.236 7	$(20, 1)$	0.133 1	0.126 8
Mbgrade	$(10^2, 50)$	0.036 2	0.194 9	$(10^2, 50)$	0.028 7	0.194 9
Pollution	$(0.5, 2)$	0.035 0	0.141 0	$(0.5, 1)$	0.023 3	0.119 2
Quake	$(0.5, 10^{-3})$	105.52	0.170 7	$(50, 1)$	11.582	0.175 8
Sensory	$(0.1, 10^2)$	0.448 3	0.146 3	$(0.1, 10^2)$	0.316 0	0.146 3
Strike	$(0.1, 10^{-2})$	1.776 7	0.060 2	$(2, 50)$	0.359 1	0.061 5
Space-ga	$(10^4, 20)$	33.448	0.039 0	$(10^4, 20)$	32.234	0.038 8
Abalone	$(20, 10^{-2})$	849.591	0.103 1	$(0.05, 2)$	79.021 9	0.086 2

Table 2 Training time and RMSE comparison of algorithms (Continued)**表 2** 算法训练时间和均方根误差比较(续)

数据集	SVR			NBSVR		
	(C, γ)	Time (s)	RMSE	(C, γ)	Time (s)	RMSE
Autoprice	(2,2)	0.116 7	0.065 5	(1,2)	0.081 0	0.064 0
Cloud	(10 ⁴ ,50)	0.081 3	0.081 5	(10 ⁴ ,50)	0.060 3	0.081 4
Housing	(5,10)	0.256 4	0.131 4	(2,2)	0.251 6	0.130 5
Machine_cpu	(10 ⁴ ,10)	0.066 2	0.030 2	(10 ⁴ ,10)	0.064 2	0.031 2
Pyrim	(2,2)	0.048 5	0.081 9	(2,2)	0.033 7	0.076 1
Servo	(2,2)	0.102 7	0.082 5	(2,2)	0.084 9	0.082 5

Table 3 Support vectors and iteration times comparison of algorithms**表 3** 算法支持向量和迭代次数比较

数据集	SVR			NBSVR		
	NBSV	BSV	Iterations	NBSV	BSV	Iterations
Babyfat	18	4	74	14	4	68
Balloon	6	104	280	6	97	276
Baskball	8	38	92	1	44	89
Bolts	10	7	46	10	7	46
Cleveland	21	122	295	11	135	291
Elusage	4	14	44	6	13	40
Fruitfly	6	53	114	4	53	116
Lowbwt	44	47	146	49	40	203
Mbgrade	1	28	61	1	28	61
Pollution	4	20	54	8	17	48
Quake	964	32	1 036	20	860	2 148
Sensory	4	261	536	2	256	520
Strike	239	16	281	2	221	448
Space-ga	46	1 065	2 866	43	1 063	2 965
Abalone	1 379	0	1 381	5	1 736	3 485
Autoprice	19	51	149	15	52	141
Cloud	10	31	122	10	31	122
Housing	5	214	455	27	183	435
Machine_cpu	30	0	56	32	0	60
Pyrim	33	6	55	26	9	50
Servo	63	5	81	65	5	85

从表 2 可以看出:在 21 个数据集的 RMSE 指标测试中,NBSVR 在 13 个数据集上的性能较好,SVR 在 4 个数据集上的性能较好,另外 4 个数据集两种算法 RMSE 值相同;在训练时间上,NBSVR 全部优于 SVR.在 Quake, Strike 和 Abalone 3 个数据集上,SVR 的训练时间比 NBSVR 要少 5 倍~10 倍.这是由于两种算法的最佳参数对中的核参数的值很小,从而导致核矩阵近似奇异化,在训练样本较大时训练时间会变得很长.

从表 3 可以看出,在 Quake, Strike 和 Abalone 这 3 个数据集上,SVR 算法的 NBSV 数目非常多.从第 3.3 节的分析可知,在 NBSV 数目很多的情况下,算法计算代价是相当大的,所以,由 SVR 算法训练的时间会很长.在其他数据集上,结合表 2 的分析,如果数据集的 NBSV 相似,迭代次数少的算法训练时间也会较少.

4.3 对偶问题最优解比较

本节对 SVR 和 NBSVR 求解对偶问题得到最优解时对应的函数值进行比较.为了公平比较,SVR 和 NBSVR 的参数全部设为 SVR 的最佳参数对(见表 2).每次将 21 个数据集随机打乱,并按顺序选取表 1 对应的训练样本数目,再用两个算法训练比较.该测试分为 3 次,记为测试 1、测试 2、测试 3.测试结果见表 4.

从表 4 可以看出:在 21 个数据集的 4 次测试中,NBSVR 总是会得到比 SVR 更小的对偶问题函数值;从对偶问题角度看,含有约束 $\sum_1^N \delta_i = \sum_{N+1}^{2N} \delta_j$ 的优化问题只能得到次优解.由此可见,SVR 对偶优化问题中约束条件 $\sum_1^N \delta_i = \sum_{N+1}^{2N} \delta_j$ 会限制算法得到更优的解.

Table 4 Global function value comparison of dual problems
表 4 对偶问题最优函数值比较

数据集	测试 1		测试 2		测试 3	
	SVR	NBSVR	SVR	NBSVR	SVR	NBSVR
Babyfat	-6.4850e+002	-6.4932e+002	7.9485e+002	7.9685e+002	-7.9985e+002	-8.0082e+002
Balloon	-1.7475e+000	-2.4337e+000	-2.0411e+000	-2.6888e+000	-1.7127e+000	-2.4245e+000
Basketball	-1.8781e+000	-2.0151e+000	-2.2336e+000	-2.5382e+000	-1.7291e+000	-1.8732e+000
Bolts	-5.5994e+001	-5.6827e+001	-5.6382e+001	-6.0978e+001	-5.9524e+001	-6.4645e+001
Cleveland	-1.0365e+001	-1.0460e+001	-1.0778e+001	-1.0852e+001	-1.1005e+000	-1.1118e+001
Elusage	-5.5204e+001	-5.6124e+001	-6.9875e+001	-7.3740e+001	-7.8588e+001	-8.4179e+001
Fruitfly	-1.7450e+000	-1.8056e+000	-1.3591e+000	-1.4188e+000	-1.2818e+000	-1.3528e+000
Lowbwt	-1.1754e+000	-3.2825e+000	-1.1271e+000	-3.2996e+000	-1.0143e+000	-3.3146e+000
Mbgrade	-5.9052e+002	-5.9067e+002	-4.7037e+002	-4.7055e+002	-4.3877e+002	-4.3888e+002
Pollution	-1.0768e+000	-1.1749e+000	-1.5319e+000	-1.5927e+000	-1.3206e+000	-1.4533e+000
Quake	-1.5640e+001	-2.6788e+001	-1.4496e+001	-2.5541e+001	-1.5128e+001	-2.5994e+001
Sensory	-3.8698e+000	-3.9966e+000	-3.5719e+000	-3.7019e+000	-3.6359e+000	-3.7659e+000
Strike	-3.7714e-001	-5.3708e-001	-4.5832e-001	-6.2830e-001	-5.2497e-001	-7.2240e-001
Space-ga	-2.7019e+005	-2.7079e+005	-2.8387e+005	-2.8433e+005	-2.6965e+005	-2.7021e+005
Abalone	-1.1127e+001	-1.0514e+002	-9.8542e+000	-1.0222e+002	-1.0527e+001	-1.0362e+002
Autoprice	-7.5856e+000	-7.9427e+000	-6.4505e+000	-6.7247e+000	-6.7433e+000	-7.1462e+000
Cloud	-1.0987e+004	-1.1151e+004	-1.4585e+004	-1.4772e+004	-1.1184e+004	-1.1388e+004
Housing	-9.5502e+001	-9.5683e+001	-9.5476e+001	-9.5610e+001	-9.0453e+001	-9.0563e+001
Machine_cpu	-1.2039e+002	-1.4444e+002	-5.0295e+002	-6.2580e+002	-3.9730e+002	-5.3445e+002
Pyrim	-1.8578e+000	-1.9494e+000	-1.7049e+000	-1.8807e+000	-2.0595e+000	-2.1273e+000
Servo	-3.5996e+000	-3.6207e+000	-3.6963e+000	-3.7206e+000	-2.2815e+000	-2.3011e+000

4.4 参数选择对训练时间的影响

本节分析代价参数 C 和核参数 γ 选择对算法训练时间的影响,数据集选用 Strike,图 1 为 SVR 和 NBSVR 算法的测试结果.可以看出:当 γ 固定时,SVR 和 NBSVR 训练时间对 C 值都不太敏感;当 C 固定时,SVR 训练时间对 γ 值比 NBSVR 敏感的多,尤其当 $\gamma < 0.1$ 时,SVR 训练时间会变得很长.当数据集样本数目很大时,SVR 寻找最优参数将会花费大量的时间.

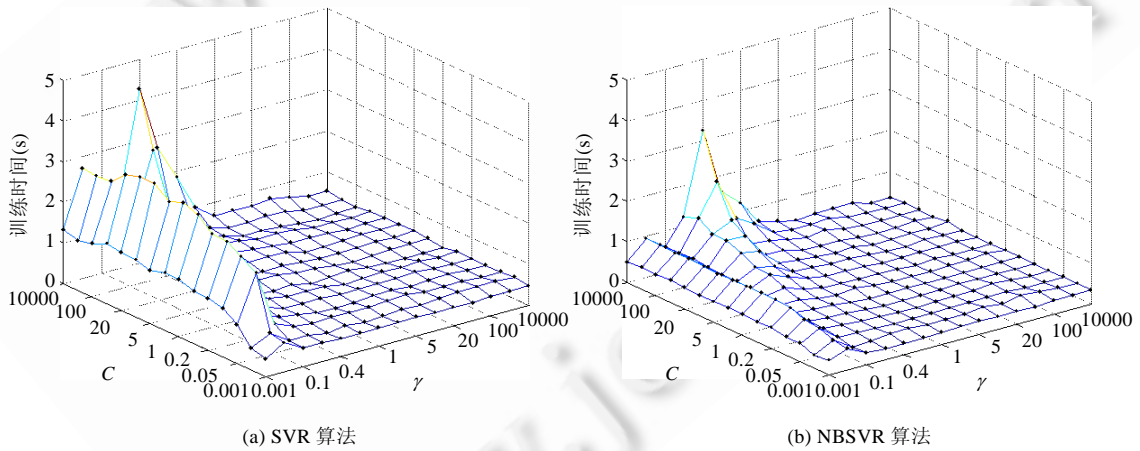


Fig.1 Relationship of parameters selection on Strike data set

图 1 Strike 数据集两个算法参数选择与训练时间的关系

5 结 论

本文研究并给出了 NBSVR 原始优化问题和对偶优化问题,提出了 NBSVR 对偶优化问题的有效集求解方法,并给出了收敛性分析.在实验中,利用多个标准数据集对 SVR 和 NBSVR 算法性能进行了不同指标的分析,

NBSVR 算法对于半数以上数据集的均方根误差低于 SVR,所有数据集的训练时间 NBSVR 均低于或类似于 SVR.对偶问题最优解测试表明,在相同参数设置下,NBSVR 均能得到比 SVR 更优的解.选取不同参数值对时,NBSVR 的训练时间变化较为平缓.

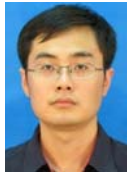
由此可见,无偏置的支持向量回归算法要比有偏置支持向量回归算法的泛化性能更好,训练时间更短,选择合适的参数更便捷.将来的工作包括将无偏置的支持向量回归应用现实生活的真实数据上,以验证它的高效性.

References:

- [1] Cortes C, Vapnik V. Support vector networks. *Machine Learning*, 1995,20(3):273–297. [doi: 10.1023/A:1022627411411]
- [2] Deng WY, Zheng QH, Chen L, Xu XB. Research on extreme learning of neural networks. *Chinese Journal of Computers*, 2010, 33(2):279–287 (in Chinese with English abstract).
- [3] Huang GB, Ding XJ, Zhou HM. Optimization method based extreme learning machine for classification. *Neurocomputing*, 2010, 74(1-3):155–163. [doi: 10.1016/j.neucom.2010.02.019]
- [4] Lee YJ, Mangasarian OL. SSVM: A smooth support vector machine for classification. *Computational Optimization and Applications*, 2001,20(1):5–22. [doi: 10.1023/A:1011215321374]
- [5] Mangasarian OL, Musicant DR. Lagrangian support vector machines. *Journal of Machine Learning Research*, 2001,1(3):161–177. [doi: 10.1162/15324430152748218]
- [6] Osuna E, Freund R, Girosit F. Training support vector machines: An application to face detection. In: Rougeaux S, Kuniyoshi Y, eds. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition: Puerto Rico*: IEEE Press, 1997. 130–136. [doi: 10.1109/CVPR.1997.609310]
- [7] Joachims T. Making large-scale SVM learning practical. In: Scholkopf B, Burges C, Smola A, eds. *Proc. of the Advances in Kernel Methods-Support Vector Learning*. Cambridge: MIT Press, 1999. 169–184.
- [8] Smola AJ, Scholkopf B. A tutorial on support vector regression. Technical Report, TR 1998-030, London: Royal Holloway College, 1998. [doi: 10.1023/B:STCO.0000035301.49549.88]
- [9] Scheinberg K. An efficient implementation of an active set method for SVMs. *Journal of Machine Learning Research*, 2006,7(10): 2237–2257.
- [10] Vishwanathan SVN, Smola AJ, Murty MN. SimpleSVM. In: Fawcett T, Mishra N, eds. *Proc. of the Int'l Conf. on Machine Learning*: Washington: AAAI Press, 2003. 760–767.
- [11] Mangasarian OL, Musicant DR. Active support vector machine classification. In: Leen TK, Dietterich TG, Tresp V, eds. *Proc. of the Advances in Neural Information Processing Systems*. Denver: MIT Press, 2001. 138–144.
- [12] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines. In Principe J, Giles L, Morgan N, Wilson E, eds. *Proc. of the Neural Networks for Signal Processing VII*. Amelia Island: IEEE Press, 1997. 276–285. [doi: 10.1109/NNSP.1997.622408]
- [13] Zhang HR, Han ZZ. An improved sequential minimal optimization learning algorithm for regression support vector machine. *Journal of Software*, 2003,14(12):2006–2013 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/2006.htm>
- [14] Zhu QD, Zhang Z, Xing ZY. Improved SMO learning method of support vector machine. *Journal of Harbin Engineering University*, 2007,28(2):183–188 (in Chinese with English abstract).
- [15] Shevade SK, Keerthi SS, Bhattacharyya C, Murthy KRK. Improvements to the SMO algorithm for SVM regression. *IEEE Trans. on Neural Networks*, 2000,11(5):1188–1193. [doi: 10.1109/72.870050]
- [16] Fletcher R. *Practical Methods of Optimization, Constrained Optimization Vol.2*. Chichester-New York, Brisbane-Toronto: John Wiley & Sons, 1981.
- [17] Blake CK, Merz CJ. UCI repository of machine learning databases. <http://archive.ics.uci.edu/ml/>
- [18] Mike M. Statistical datasets. <http://lib.stat.cmu.edu/datasets/>
- [19] Ghanty P, Paul S, Pal NR. NEUROSVM: An architecture to reduce the effect of the choice of kernel on the performance of SVM. *Journal of Machine Learning Research*, 2009,10(3):591–622. [doi: 10.1145/1577069.1577090]

附中文参考文献:

- [2] 邓万宇,郑庆华,陈琳,许学斌.神经网络极速学习方法研究.计算机学报,2010,33(2):279-287.
- [13] 张浩然,韩正之.回归支持向量机的改进序列最小优化学习算法.软件学报,2003,14(12):2006-2013. <http://www.jos.org.cn/1000-9825/14/2006.htm>
- [14] 朱齐丹,张智,邢卓异.支持向量机改进序列最小优化学习算法.哈尔滨工程大学学报,2007,28(2):183-188.



丁晓剑(1982—),男,江苏沐阳人,博士,工程师,主要研究领域为神经网络,机器学习算法.



赵银亮(1960—),男,博士,教授,博士生导师,主要研究领域为语言及编译系统,并行计算,机器学习.

www.jos.org.cn

www.jos.org.cn