

文章编号: 1001-0920(2009)11-1609-06

## 回归型支持向量机的调节熵函数法

吴青<sup>1,2</sup>, 刘三阳<sup>1</sup>, 张乐友<sup>1</sup>

(1. 西安电子科技大学 数学科学系, 西安 710071; 2. 西安邮电学院 信息与控制系, 西安 710061)

**摘要:** 基于最优化理论中的 KKT 互补条件建立支持向量回归机的无约束不可微优化模型, 并给出了一种有效的光滑近似解法——调节熵函数方法. 该方法不需参数取值很大便可逼近问题的最优解, 从而避免了一般熵函数法为了逼近精确解, 参数取得过大而导致数值的溢出现象, 为求解支持向量回归机提供了一条新途径. 数值实验结果表明, 回归型支持向量机的调节熵函数法改善了支持向量机的回归性能和效率.

**关键词:** 支持向量机; 回归;  $\epsilon$ -不敏感损失函数; 调节熵函数; Newton 算法

**中图分类号:** TP181

**文献标识码:** A

## Adjustable entropy function method for support vector regression

WU Qing<sup>1,2</sup>, LIU San-yang<sup>1</sup>, ZHANG Le-you<sup>1</sup>

(1. Department of Mathematical Sciences, Xidian University, Xi'an 710071, China; 2. Department of Information and Control, Xi'an Institute of Posts and Telecommunications, Xi'an 710061, China. Correspondent: WU Qing, E-mail: xidianwq@yahoo.com.cn)

**Abstract:** Based on Karush-Kuhn-Tucker(KKT) complementary condition in optimization theory, unconstrained non-differential optimization models for support vector regression are proposed, and an adjustable entropy function method is given. This method can find an optimal solution with a relatively small parameter. It avoids the numerical overflow in the entropy function methods available. It is a new approach to solve support vector regression. Numerical results show that the new approach improves the regression performance and increases the learning efficiency.

**Key words:** Support vector machine; Regression;  $\epsilon$ -insensitive loss function; Adjustable entropy function; Newton algorithm

### 1 引言

近年来,建立在统计学习理论基础上的支持向量机(SVM)方法,已成为数据挖掘和机器学习领域研究的一个热点. SVM 作为一种通用有效的学习机,已成功地应用于模式分类、回归分析和密度估计等问题<sup>[1,2]</sup>. 支持向量回归对离散数据的拟合具有很好的效果. 目前,对于支持向量回归机(SVR)<sup>[3,4]</sup>模型的算法机理已有较为明确的认识,但算法的设计和实现依然比较困难,而且收敛速度较慢.

将光滑函数用于 SVR,可使原来不可微的模型变成可微模型,从而可以采用快速的求解算法,降低 SVM 的计算复杂性. 2005 年, Lee 等提出了光滑的 SVR 模型  $\epsilon$ -SSVR<sup>[5]</sup>, 结果表明  $\epsilon$ -SSVR 的效果好于 LIBSVM<sup>[6]</sup> 和 SVM<sup>high</sup><sup>[7]</sup> 算法. 2008 年,熊金志等运用插值函数和复合函数的方法提出一类新的光滑函

数<sup>[8]</sup>,回归性能有了进一步的改善. 采用光滑化技巧解决 SVR 的研究尚有很大的发展空间,有待于进一步深入地研究.

极大熵方法是近几年出现的一种新的数值优化方法,可以有效地处理极大极小问题和多约束非线性规划等优化问题. 它具有收敛速度快、数值稳定性好、易于计算机实现等优点,是求解大型多约束优化问题和某些不可微优化问题的有效方法. 文献[9-12]中的数值例子表明,该方法具有一定的应用价值,但进一步的分析表明它存在严重的不足<sup>[11]</sup>,只有当参数  $p$  趋于无穷大时,才能逼近问题的精确解,然而当  $p$  取较大值时,又容易产生数值的溢出现象. [13,14]分别采用传统的熵函数和分段熵函数法训练 SVM 的分类问题,虽然避免了数值的溢出现象,但只有参数  $p$  取值很大时才能取得满意的结

收稿日期: 2008-10-08; 修回日期: 2009-03-21.

基金项目: 国家自然科学基金项目(60574075,60674108).

作者简介: 吴青(1975—),女,山东临沂人,博士生,从事模式识别、机器学习和最优化理论及其应用的研究; 刘三阳(1959—),陕西临潼人,教授,博士生导师,从事最优化理论方法、数据挖掘、支持向量机等研究.

果.[15]提出将调节熵函数法应用于支持向量机的分类问题,使得当参数  $p$  的取值不太大时,便可以逼近问题的最优解.

本文给出了带调节因子的熵函数法,并将其应用于 SVR 问题,克服了极大熵函数法训练 SVR 所造成的数值溢出的现象,不需要参数  $p$  取值很大,便可以逼近问题的精确解. 本文将调节熵函数应用于 SVR,从而为求解 SVM 的回归问题提供了一种新途径. 数值实验结果表明,AEF-SVR 具有较好的回归性能和效率,并成功地避免了数值的溢出现象.

## 2 支持向量机

考虑回归问题,给定训练样本  $\{x_i, y_i\}, i = 1, 2, \dots, l$ . 其中:  $x_i \in R^k, y_i \in R, x_i$  为  $k$  维输入样本,  $l$  为样本的总个数. SVR 的学习问题就是一个二次规划问题,通常采用 Vapnik 的  $\epsilon$ -不敏感损失函数,即指定容许误差  $\epsilon$ . 若样本  $x_i$  误差为  $\xi_i$ , 则当  $|\xi_i| \leq \epsilon$  时,不计损失;否则,损失计为  $|\xi_i| - \epsilon$ . 回归函数可表示为

$$p(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(x, x_i) + b.$$

其中:  $\alpha$  和  $\alpha^*$  均为待求的  $k$  维向量;  $k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$  称为核函数,  $\varphi(x)$  为从样本空间到高维特征空间的映射函数,核函数表示为两个  $\varphi(\cdot)$  的点积. SVR 需求解如下二次规划问题(P1):

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*); \\ \text{s. t.} \quad & y_i - (w \cdot \varphi(x_i)) - b \leq \epsilon + \xi_i, \\ & (w \cdot \varphi(x_i)) + b - y_i \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, l. \end{aligned}$$

其中:  $(w \cdot \varphi(x_i))$  表示内积;参数  $C$  为常数,表示对超出误差  $\epsilon$  样本的惩罚程度.

模型(P1)的对偶形式为(P2)

$$\begin{aligned} \min_{\alpha, \alpha^*} \quad & \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(x_i, x_j) - \\ & \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) + \epsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*); \\ \text{s. t.} \quad & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \\ & \alpha_i, \alpha_i^* \in [0, C], \quad i = 1, 2, \dots, l. \end{aligned}$$

其中

$$\begin{aligned} \alpha &= (\alpha_1, \alpha_2, \dots, \alpha_l)^\top, \\ \alpha^* &= (\alpha_1^*, \alpha_2^*, \dots, \alpha_l^*)^\top. \end{aligned}$$

求得  $\alpha$  及  $\alpha^*$  后,可得回归函数  $p(x)$ . 令

$$\begin{aligned} f(\alpha, \alpha^*) &\triangleq \\ & \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_j - \alpha_j^*) k(x_i, x_j) (\alpha_i - \alpha_i^*), \end{aligned}$$

$$\begin{aligned} g_i(\alpha, \alpha^*, b) &\triangleq \\ & y_i - \left( \sum_{j=1}^l (\alpha_j - \alpha_j^*) k(x_i, x_j) + b \right), \\ & i = 1, 2, \dots, l. \end{aligned} \tag{1}$$

则模型(P1)等价于模型(P3)

$$\begin{aligned} \min_{\alpha, \alpha^*} \quad & f(\alpha, \alpha^*) + C \sum_{i=1}^l (\xi_i + \xi_i^*); \\ \text{s. t.} \quad & g_i(\alpha, \alpha^*, b) \leq \epsilon + \xi_i, \\ & -g_i(\alpha, \alpha^*, b) \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, l. \end{aligned}$$

对于优化问题(P3),引入 Lagrange 乘子  $\beta, \beta^*, \gamma, \gamma^*$ , 其中

$$\begin{aligned} \beta &= (\beta_1, \beta_2, \dots, \beta_l)^\top, \\ \beta^* &= (\beta_1^*, \beta_2^*, \dots, \beta_l^*)^\top, \\ \gamma &= (\gamma_1, \gamma_2, \dots, \gamma_l)^\top, \\ \gamma^* &= (\gamma_1^*, \gamma_2^*, \dots, \gamma_l^*)^\top. \end{aligned}$$

建立 Lagrange 函数

$$\begin{aligned} L(\alpha, \alpha^*, b, \beta, \beta^*, \gamma, \gamma^*) &= \\ & f(\alpha, \alpha^*) + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \\ & \sum_{i=1}^l \beta_i (\epsilon + \xi_i - g_i(\alpha, \alpha^*, b)) - \\ & \sum_{i=1}^l \gamma_i \xi_i - \sum_{i=1}^l \beta_i^* (\epsilon + \xi_i^* + \\ & g_i(\alpha, \alpha^*, b)) - \sum_{i=1}^l \gamma_i^* \xi_i^*. \end{aligned}$$

利用 KKT 互补条件,有

$$\beta_i (\epsilon + \xi_i - g_i(\alpha, \alpha^*, b)) = 0, \tag{2}$$

$$\beta_i^* (\epsilon + \xi_i^* + g_i(\alpha, \alpha^*, b)) = 0, \tag{3}$$

$$(C - \beta_i - \gamma_i) \xi_i = 0, \tag{4}$$

$$(C - \beta_i^* - \gamma_i^*) \xi_i^* = 0. \tag{5}$$

由式(2) ~ (5)可知

$$\begin{aligned} \xi_i &= \max\{0, g_i(\alpha, \alpha^*, b) - \epsilon\}, \\ \xi_i^* &= \max\{0, -g_i(\alpha, \alpha^*, b) - \epsilon\}. \end{aligned}$$

因此,模型(P1)可等价于无约束优化模型(P4)

$$\begin{aligned} \min_{\alpha, \alpha^*, b} \quad & f(\alpha, \alpha^*) + C \sum_{i=1}^l [\max\{0, g_i(\alpha, \alpha^*, b) - \\ & \epsilon\} + \max\{0, -g_i(\alpha, \alpha^*, b) - \epsilon\}]. \end{aligned}$$

引入函数

$$\begin{aligned} r_i(\alpha, \alpha^*, b) &= g_i(\alpha, \alpha^*, b) - \epsilon, \\ r_{i+1}(\alpha, \alpha^*, b) &= -g_i(\alpha, \alpha^*, b) - \epsilon, \\ & i = 1, 2, \dots, l. \end{aligned}$$

则无约束不可微优化模型(P4)可等价于如下含有  $2l + 1$  个变量的无约束不可微规划(P5)

$$\min_{\alpha, \alpha^*, b} f(\alpha, \alpha^*) + C \sum_{i=1}^{2l} \max\{0, r_i(\alpha, \alpha^*, b)\}.$$

令

$$\begin{aligned} \|\xi\|_1 &= \sum_{i=1}^{2l} \max\{0, r_i(\alpha, \alpha^*, b)\}, \\ \|\xi\|_\infty &= \max\{0, r_1(\alpha, \alpha^*, b), r_2(\alpha, \alpha^*, b), \dots, r_{2l}(\alpha, \alpha^*, b)\}, \end{aligned}$$

则由有限维线性空间上的向量范数等价性定理, 可得如下不等式:

$$\frac{1}{2l} \|\xi\|_1 \leq \|\xi\|_\infty \leq \|\xi\|_1. \quad (6)$$

因此, 若用  $\|\xi\|_\infty$  代替式(6)中  $\|\xi\|_1$ , 则可得无约束优化模型(P6)

$$\begin{aligned} \min_{\alpha, \alpha^*, b} \Gamma(\alpha, \alpha^*, b) = \\ f(\alpha, \alpha^*) + C \max\{0, r_1(\alpha, \alpha^*, b), \\ r_2(\alpha, \alpha^*, b), \dots, r_{2l}(\alpha, \alpha^*, b)\}. \end{aligned}$$

显然, 无约束优化模型(P6)在形式上更加简洁. (P6)的目标函数是凸的, 但不可微, 当  $k$  较小时, 可用直接搜索法求解, 如 Nelder-Mead 单纯形法、Powell 方向加速法. 当  $k$  较大时, 直接搜索法效率低, 需采用其他方法(熵方法是处理这类极大值函数不可微优化的一种有效技巧). 下面给出优化问题(P6)的一种光滑函数的解法.

### 3 调节熵函数法

对于极大极小问题  $\min_{x \in R^k} \max_{1 \leq i \leq m} f_i(x)$  (这里  $f_i(x)$  是  $R^k$  中连续可微的函数,  $m \geq 2$  是正整数), 利用极大熵原理推导出的可微函数

$$F_p(x) = \frac{1}{p} \ln\left(\sum_{i=1}^m \exp(pf_i(x))\right),$$

常称为极大熵函数. 用该可微函数来逼近最大值函数  $F(x) = \max_{1 \leq i \leq m} f_i(x)$ , 只有当参数  $p$  趋于无穷大时,  $F_p(x)$  的解才趋于  $F(x)$  的精确解, 所以当精度要求很高时,  $p$  必须取得很大. 由文献[9]知, 这样容易产生数值的溢出现象. 为了克服  $F_p(x)$  的不足, 本文提出了带调节因子的熵函数法来训练 SVR, 即 AEF-SVR 方法.

定义调节熵函数

$$F_{(p, \mu)}(x) = \frac{1}{p} \ln\left(\sum_{i=1}^m \mu_i \exp(pf_i(x))\right). \quad (7)$$

其中:  $p$  和  $\mu$  均为参数,  $p > 0, \mu \in R^m, \mu_i \geq 0$  且  $\sum_{i=1}^m \mu_i = 1$ .

通过引入调节熵函数, 问题(P6)的求解可转化为如下的无约束优化问题(P7)

$$\begin{aligned} \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) = \\ f(\alpha, \alpha^*) + \frac{C}{p} \ln\left(\sum_{i=0}^{2l} \mu_i \exp(pr_i(\alpha, \alpha^*, b))\right). \end{aligned} \quad (8)$$

其中:  $r_0(\alpha, \alpha^*, b) = 0, p > 0, \mu \in \Delta = \{\mu \in R^{2l+1}, \mu_i \geq 0, i = 0, 1, \dots, 2l, \sum_{i=0}^{2l} \mu_i = 1\}$  为参数. 由以上分析, 优化问题(P1)等价于无约束优化问题(P7), 容易证明  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  是任意阶可微的凸函数. 因此, 优化问题(P7)的任一局部最优解都是全局最优解.

定义  $R(\alpha, \alpha^*, b) = \max_{0 \leq i \leq 2l} r_i(\alpha, \alpha^*, b), I(\alpha, \alpha^*, b) = \{i \mid r_i(\alpha, \alpha^*, b) = R(\alpha, \alpha^*, b), i = 0, 1, \dots, 2l\}, \Delta_I = \{\mu \in \Delta, \text{对 } i \notin I(\alpha, \alpha^*, b), \mu_i = 0\}$ , 由极大熵函数的逼近性质可得如下定理:

**定理 1** 设  $\mu \in \Delta$  且  $\mu \neq 0$ , 则  $C \ln \mu_{\max} / p \leq \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) - \Gamma(\alpha, \alpha^*, b) \leq 0$ , 其中  $\mu_{\max} = \max\{\mu_i, i \in I(\alpha, \alpha^*, b)\}$ , 且当  $p \rightarrow \infty$  时,  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  一致收敛到  $\Gamma(\alpha, \alpha^*, b)$ .

**证明** 显然有

$$\begin{aligned} \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) \leq \\ f(\alpha, \alpha^*) + \frac{C}{p} \ln\left(\sum_{i=0}^{2l} \mu_i \exp(pR(\alpha, \alpha^*, b))\right) = \\ f(\alpha, \alpha^*) + CR(\alpha, \alpha^*, b) = \Gamma(\alpha, \alpha^*, b). \end{aligned}$$

因为对于  $i \in I(\alpha, \alpha^*, b)$ , 有  $r_i(\alpha, \alpha^*, b) = R(\alpha, \alpha^*, b)$ , 则

$$\begin{aligned} \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) \geq \\ f(\alpha, \alpha^*) + \frac{C}{p} \ln(\mu_i \exp(pR(\alpha, \alpha^*, b))) = \\ f(\alpha, \alpha^*) + CR(\alpha, \alpha^*, b) + \frac{C}{p} \ln \mu_i. \end{aligned}$$

于是有

$$\Gamma_{(p, \mu)}(\alpha, \alpha^*, b) \geq \Gamma(\alpha, \alpha^*, b) + \frac{C}{p} \ln \mu_{\max}.$$

所以, 当  $\mu \in \Delta, \mu \neq 0$  时

$$C \ln \mu_{\max} / p \leq$$

$$\Gamma_{(p, \mu)}(\alpha, \alpha^*, b) - \Gamma(\alpha, \alpha^*, b) \leq 0,$$

当  $p \rightarrow \infty$  时,  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  一致收敛于  $\Gamma(\alpha, \alpha^*, b)$ .

**定理 2**<sup>[11]</sup> 对于  $(\alpha, \alpha^*, b) \in R^l \times R^l \times R, p \in R^+, \text{有 } \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) = \Gamma(\alpha, \alpha^*, b)$ , 当且仅当  $\mu \in \Delta_I$ .

以上 2 个定理表明, 不仅可以调节  $p$ , 而且可以通过调节  $\mu$ , 使  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  更快地逼近  $\Gamma(\alpha, \alpha^*, b)$ . 如果一直调节  $p$  使  $p$  充分大, 则会出现  $\exp(pr_i)$  溢出的数值计算问题. 所以调节熵函数法在取得充分大的  $p$  后, 便不再增大  $p$  的值, 而是调节  $\mu$ , 使得  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  收敛到  $\Gamma(\alpha, \alpha^*, b)$ , 从而避免数值的溢出问题. 下面给出  $\mu$  的调节方法.

设  $p > 0, \mu \in \Delta$  是当前的迭代点值, 即对于  $p > \Delta, \mu \in \Delta, (\alpha, \alpha^*, b)$  是  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  的极小解, 则有  $\nabla \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) = 0$ , 即

$$\begin{aligned} & \nabla \Gamma_{(p, \mu)}(\alpha, \alpha^*, b) = \\ & \begin{bmatrix} \mathbf{K}(\alpha - \alpha^*) \\ -\mathbf{K}(\alpha - \alpha^*) \\ 0 \end{bmatrix} + \\ & C \begin{bmatrix} \sum_{i=0}^l \mu_i \exp(pr_i) - \sum_{i=l+1}^{2l} \mu_i \exp(pr_i) \\ \sum_{j=0}^{2l} \mu_j \exp(pr_j) \end{bmatrix} \begin{bmatrix} -\mathbf{K}e_i \\ \mathbf{K}e_i \\ -1 \end{bmatrix} = 0. \end{aligned} \tag{9}$$

其中: 核矩阵  $\mathbf{K} = (k(x_i, x_j))_{i,j=1}^l$  为对称矩阵,  $e_i$  表示第  $i$  个分量为 1 的  $l$  维单位向量. 选取

$$\sigma_i = \frac{\mu_i \exp(pr_i)}{\sum_{j=0}^{2l} \mu_j \exp(pr_j)}, \quad i = 0, 1, \dots, 2l, \tag{10}$$

则  $\sigma_i \in \Delta$  且满足

$$\begin{aligned} & \begin{bmatrix} \mathbf{K}(\alpha - \alpha^*) \\ -\mathbf{K}(\alpha - \alpha^*) \\ 0 \end{bmatrix} + C \left( \sum_{i=0}^l \sigma_i - \sum_{i=l+1}^{2l} \sigma_i \right) \begin{bmatrix} -\mathbf{K}e_i \\ \mathbf{K}e_i \\ -1 \end{bmatrix} = 0, \end{aligned} \tag{11}$$

所以式(10)定义的  $\sigma_i$  可作为下一个调解因子的值,  $\Gamma_{(p, \mu)}(\alpha, \alpha^*, b)$  的 Hessian 矩阵也容易求出.

### 4 支持向量机的调节熵函数法

Newton 下降法具有二次收敛性. 下面给出二次无约束规划问题(P7)的 Newton 算法.

**算法 1** 输入  $\{(x_i, y_i)\}_{i=1}^l$ , 初值  $z_0 = (\alpha_0, \alpha_0^*, b_0)$ ,  $\epsilon_1 > 0, \epsilon_2 > 0, p_0 = 1, \mu_i^0 = 1/(l+1), i = 0, 1, \dots, 2l, k = 0$ .

Step1: 计算  $g_k = \nabla \Gamma_{(p_k, \mu_k)}(z_k)$ , 若  $\|g_k\| \leq \epsilon_1$ , 则转 Step4.

Step2: 计算

$$\mathbf{H}_k = \nabla^2 \Gamma_{(p_k, \mu_k)}(z_k), \mathbf{d}_k = -\mathbf{H}_k^{-1} g_k.$$

Step3: 一维搜索  $\min_{\lambda > 0} \Gamma_{(p_k, \mu_k)}(z_k + \lambda \mathbf{d}_k)$ , 得  $\lambda_k$ ,

$z_{k+1} = z_k + \lambda_k \mathbf{d}_k, k = k + 1$ , 转 Step1.

Step4:  $z^*(p_k) = z_k$ , 若满足  $\|z^*(p_k) - z^*(p_{k-1})\| > \epsilon_2$ , 则计算

$$\begin{aligned} & p_{k+1} = 10p_k, \\ & \mu_i^{k+1} = \frac{\mu_i^k \exp(p_k r_i(z_k))}{\sum_{j=0}^{2l} \mu_j^k \exp(p_k r_j(z_k))} \end{aligned}$$

$$i = 0, 1, \dots, 2l, k = k + 1,$$

转 Step1; 否则转 Step5.

Step5: 算法结束, 输出最优解  $(\hat{\alpha}, \hat{\alpha}^*, \hat{b}) = z_k$ .

由于 Newton 算法是二次收敛的, 对于较小的

$p$ , 只需很少几部迭代便满足了  $\epsilon_1$  限制. 整个算法具有二次收敛性.

传统的 SVR 训练算法需要解决大规模样本条件下的凸二次规划问题, 需要处理  $l \times l$  阶矩阵, 运算量和所需内存都比较大. AEF-SVR 在求解过程中, 除必须存储原始数据外, 只涉及到  $2k + 1$  阶方阵的存储和求逆, 对于回归问题, 一般有  $k \ll l$ . 相比而言, 对于大规模的回归问题, 该算法的存储量和计算量都较小.

### 5 数值实验

为了充分比较 AEF-SVR, MEF-SVR 和传统  $\epsilon$ -SVR 的性能差异, 分别将这 3 种算法应用于 2 个人工数据集和 3 个实际数据集回归问题. 采用高斯核函数

$$\begin{aligned} & k(x_i, x_j) = \exp(-\theta \|x_i - x_j\|^2), \\ & i, j = 1, 2, \dots, l, \end{aligned}$$

其中  $\theta$  为高斯核的宽度参数. 实验均在 P-IV 机上完成, 程序采用 Matlab7.0 编写. 以二范数相对误差 (SRE) 作为性能指标来衡量不同方法对训练数据的拟合效果以及对测试数据的预测效果.

$$\text{SRE} = \frac{\|y - \hat{y}\|_2}{\|y\|_2}. \tag{12}$$

其中:  $y$  为实际观测值,  $\hat{y}$  为预测值.

#### 实验 1 人工数据集回归问题

sinc 函数

$$f(x) = 0.5 \cdot \frac{\sin(\frac{30}{\pi}x)}{\frac{30}{\pi}x} + \rho$$

是例证 SVR 的流行选择. 其中  $\rho$  是附加的高斯白噪声, 均值为 0, 方差  $\sigma^2 = 0.08$ . 在区间  $[-1, 1]$  上等间隔产生 101 个点作为训练样本, 观测值由 sinc 函数产生. 用随机产生的 1000 个无噪声样本度量误差, 取 10 次实验的平均值作为测试误差. 设  $\epsilon = 0.02, \theta = 5.56, C = 30$ . 在 MEF-SVR 和 AEF-SVR 中, 分别取 10000 和 10.

实验结果表明, 与  $\epsilon$ -SVR 相比, MEF-SVR 和 AEF-SVR 的相对误差较小, 且二者相差不大, MEF-SVR 和 AEF-SVR 的 CPU 用时也比传统的  $\epsilon$ -SVR 短. 但是, AEF-SVR 算法中参数  $p$  的取值远远小于 MEF-SVR 中  $p$  的取值.

图 1 给出了 AEF-SVR 的回归曲线. 其中: “+” 表示训练样本点, 中间一条实线是 AEF-SVR 产生的回归曲线, 上下两条点线到回归曲线垂直方向的间距为  $\epsilon$ , 从而上下两条点线形成了  $\epsilon$ -管道, 该管道套住了训练集中的所有样本点. 具体实验结果见表 1.

表 1 3 种算法的实验结果比较

算 法	$(\theta, C, \epsilon, \rho)$	训练误差	测试误差	CPU 时间 /s
AEF-SVR	(5.56, 30, 0.02, 10)	0.0533	0.0492	0.036
MEF-SVR	(5.56, 30, 0.02, 10 000)	0.0527	0.0499	0.031
$\epsilon$ -SVR	(5.56, 30, 0.02, \)	0.0611	0.0570	0.049

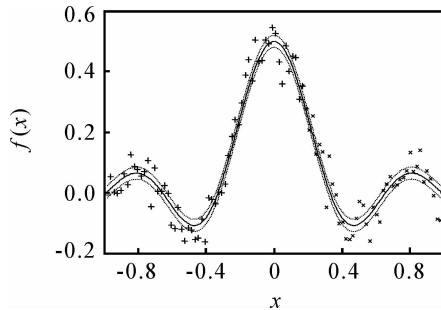


图 1 调节熵函数支持向量回归机产生的回归曲线

实验 2 实际数据集回归问题

波士顿住房问题<sup>[16]</sup>是检验算法回归性能的基础数据集,该实验分别测试了 AEF-SVR 在波士顿住房问题、Kin-fh 数据集<sup>[17]</sup>和 Comp-activ 数据集<sup>[18]</sup>上的性能,并与 MEF-SVR 和  $\epsilon$ -SVR 分别进行了比较.对于波士顿住房问题,试验中随机选取 450 个样本作为训练样本,其余的作为测试样本;对于 Kin-fh 数据集,从整个数据集中随机选择 7373 个样本作为训练样本,其余的作为测试样本;对于 Comp-activ 数据集,随机选取 7373 个样本作为训练样本,其余的作为测试样本.实验中对所有的样本数据都进行了归一化处理,取 10 次实验的平均值作为最终结果.表 2 给出了所使用的回归基准问题的数

据特征;表 3 为 AEF-SVR,MEF-SVR 和  $\epsilon$ -SVR 在 3 个回归基准问题上的实验结果,包括回归问题的均方误差(MSE)和训练时间(CPU)比较.

表 2 基准问题的数据特征

数据集	训练样本 / 个	测试样本 / 个	特征数
Boston Housing	450	56	13
Kin-fh	7373	819	32
Comp-activ	7373	819	21

AEF-SVR 算法因为可以同时调节参数  $\rho$  和  $\mu$ ,从而  $\rho$  不需取得很大便可以得到较好的实验结果.在以上 3 个试验中, $\rho$  分别取 10,10,12. MEF-SVR 采用传统的极大熵函数法训练 SVR,可以得到很好的回归效果,但  $\rho$  的取值都较大.在以上 3 个实验中, $\rho$  分别取 5000,5000,100000,3 种算法均具有很高的拟合能力,而 AEF-SVR 和 MEF-SVR 的回归性能和训练速度均优于  $\epsilon$ -SVR. MEF-SVR 算法虽然避免了数值的溢出现象,训练时间略短于 AEF-SVR,但参数  $\rho$  的取值比较大,均不小于 5000,甚至达到  $10^5$  才取得了表 3 所示的结果. AEF-SVR 在  $\rho$  的取值很小、训练时间较短的前提下,取得了比较好的回归性能.

表 3 基准问题上的对比实验结果

数据集	算 法	参数(C, $\epsilon$ , $\rho$ )	训练误差	测试误差	CPU 时间 /s
Boston	AEF-SVR	(50, 0.1, 10)	0.2034	0.1956	0.045
	MEF-SVR	(50, 0.1, 5000)	0.2040	0.1952	0.040
	$\epsilon$ -SVR	(50, 0.1, \)	0.2065	0.1988	0.99
Kin-fh	AEF-SVR	(10, 0.1, 10)	0.1346	0.1340	2.332
	MEF-SVR	(10, 0.1, 5000)	0.1323	0.1354	2.207
	$\epsilon$ -SVR	(10, 0.1, \)	0.1346	0.1359	12.310
Comp-activ	AEF-SVR	(60, 0.2, 12)	0.028	0.033	1.664
	MEF-SVR	(60, 0.2, 100000)	0.031	0.033	1.525
	$\epsilon$ -SVR	(60, 0.2, \)	0.037	0.034	8.652

6 结 论

本文在  $\epsilon$ -SVR 约束优化模型的基础上,依据最优理论建立了无约束模型.为了解决目标函数不可微问题给求解带来的困难,引入了调节熵函数这一光滑化技术,将上述问题转化为可微的无约束优化问题,进而给出一种简单易行且二次收敛的 Newton 算法,为求解 SVR 提供了一种新思路.

SVR 的调节熵函数法可以同时调节参数  $\rho$  和  $\mu$ ,以任意精度逼近原问题的解,避免了数值的溢出现象,节约了学习过程的存储空间,在保证回归性能不减(或稍有增加)的前提下,提高了学习效率.

参考文献(References)

[1] Vapnik V N. The nature of statistical learning theory [M]. New York: Springer-Verlag, 2000.

- [2] Vapnik V N, Golowich S, Smola A. Support vector method for function approximation, regression estimation, and signal processing [C]. *Advances in Neural Information Processing Systems*. Cambridge: MIT Press, 1996, 9: 281-287.
- [3] Smola A J, Schölkopf B. A tutorial on support vector regression [J]. *Statistics and Computing*, 2004, 14(3): 199-222.
- [4] Shevade S K, Keerthi S S, Bhattacharyya C, et al. Improvements to SMO algorithm for regression [J]. *IEEE Trans on Neural Networks*, 2000, 11(5): 1188-1193.
- [5] Lee Y J, Hsieh W F, Huang C F.  $\epsilon$ -SSVR: A smooth support vector machine for  $\epsilon$ -insensitive regression [J]. *IEEE Trans on Knowledge and Data Engineering*, 2005, 17(5): 678-685.
- [6] Chang C C, Lin C J. LIBSVM: A library for support vector machine [DB/OL]. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 2001-11-27.
- [7] Joachims T. SVM<sup>light</sup> [DB/OL]. <http://svmlight.joachims.org>. 2002-05-13.
- [8] 熊金志, 胡金莲, 袁华强, 等. 支持向量回归机的光滑函数研究 [J]. *模式识别与人工智能*, 2008, 21(2): 273-279.  
(Xiong J Z, Hu J L, Yuan H Q, et al. Smoothing functions for support vector regressions [J]. *Pattern Recognition and Artificial Intelligence*, 2008, 21(2): 273-279.)
- [9] 李兴斯. 一类不可微优化问题的有效解法 [J]. *中国科学(A)*, 1994, 24(4): 371-377.  
(Li X S. An efficient method for a sort of non-differential optimization problems [J]. *Science in China (A)*, 1994, 24(4): 371-377.)
- [10] 李兴斯. 非线性极大极小问题的一个有效解法 [J]. *科学通报*, 1991, 36(19): 1448-1451.  
(Li X S. An efficient method for nonlinear minmax problems [J]. *Chinese Science Bulletin*, 1991, 36(19): 1448-1451.)
- [11] 杨庆之, 杨德庄, 张敏洪. 调节熵函数 [J]. *计算数学*, 2001, 23(1): 81-86.  
(Yang Q Z, Yang D Z, Zhang M H. Adjustable entropy function method [J]. *Mathematica Numerica Sinica*, 2001, 23(1): 81-86.)
- [12] Zhang L W, Tang H W. A maximum entropy algorithm with parameters for solving minimax problem [J]. *Archives of Control Sciences*, 1997, 6(1/2): 47-59.
- [13] 郭崇慧, 孙建涛, 陆玉昌. 广义支持向量机优化问题的极大熵方法 [J]. *系统工程理论与实践*, 2005, 25(6): 27-32.  
(Guo C H, Sun J T, Lu Y C. Maximum entropy approach for generalized support vector machine optimization problems [J]. *Systems Engineering-Theory & Practice*, 2005, 25(6): 27-32.)
- [14] 周水生, 周利华. 训练支持向量机的低维 Newton 算法 [J]. *系统工程与电子技术*, 2004, 26(9): 1315-1318.  
(Zhou S S, Zhou L H. Lower dimension Newton algorithm for training the support vector machines [J]. *Systems Engineering and Electronics*, 2004, 26(9): 1315-1318.)
- [15] Wu Q, Liu S Y, Zhang L Y. Adjustable entropy function method for support vector machines [J]. *J of Systems Engineering and Electronics*, 2008, 19(5): 1029-1034.
- [16] Murphy P M, Aha D W. UCI machine learning repository [DB/OL]. [www.ics.uci.edu/~mlearn/MLRepository.html](http://www.ics.uci.edu/~mlearn/MLRepository.html). 1996-10-10
- [17] Delve. Data for evaluating learning in valid experiments, Kin-family dataset [DB/OL]. <http://www.cs.toronto.edu/~delve/data/kin/desc.html>. 1996-10-05.
- [18] Delve. Data for evaluating learning in valid experiments, comp-activ dataset [DB/OL]. <http://www.cs.toronto.edu/~delve/data/comp-activ/desc.html>. 1996-10-08.

~~~~~

(上接第 1608 页)

- [52] Zhang Y, Tian Y-P. Consentability and protocol design of multi-agent systems with stochastic switching topology [J]. *Automatica*, 2009, 45(5): 1195-1201.
- [53] Hatano Y, Mesbahi M. Agreement over random networks [J]. *IEEE Trans on Automatic Control*, 2005, 50(11): 1867-1872.
- [54] Fagnani F, Zampieri S. Average consensus with packet drop communication [J]. *SIAM J on Control and Optimization*, 2009, 48(1): 102-133.
- [55] Mehyar M, Spanos D, Pongsajapan J, et al. Asynchronous distributed averaging on communication networks [J]. *IEEE/ACM Trans on Networking*, 2007, 15(3): 512-520.