

## 用于回归估计的支持向量机方法

杜树新, 吴铁军

(浙江大学工业控制技术国家重点实验室, 浙江大学智能系统与决策研究所, 浙江杭州 310027)



**摘 要:** 用于回归估计的支持向量机方法以可控制的精度逼近非线性函数, 具有全局最优、良好泛化能力等优越性能, 得到广泛的研究。描述了该方法的基本思想, 着重讨论了  $n$ -SVM、最小二乘 SVM、加权 SVM、线性 SVM 等支持向量机的新方法, 降低训练时间和减少计算复杂性的分解法、SMO 及增量学习算法。在非线形系统参数辨识、预测预报、建模与控制研究中, 支持向量机是很有发展前途的研究方法。

**关键词:** 支持向量机; 回归估计; 预测预报; 建模与控制

文章编号: 1004-731X (2003) 11-1580-06

中图分类号: TP181

文献标识码: A

## Support Vector Machines for Regression

DU Shu-xin, WU Tie-jun

(National Key Lab of Industrial Control Technology, Institute of Intelligent Systems and Decision Making, Zhejiang University, Hangzhou 310027, China)

**Abstract:** Support Vector Machine (SVM) for regression has recently attracted growing research interest due to its obvious advantage such as nonlinear function approximation with arbitrary accuracy, and good generalization ability, unique and globally optimal solutions. An overview of the basic ideas underlying SVM for regression is given in this paper. In particular, new methods such as  $n$ -SVM, LS-SVM, weighted SVM and linear SVM, and optimization algorithms including decomposition method and SMO and incremental learning with fast computational speed and ease of implementation are concentrated as well. SVM for regression is an important and promising new direction in the area of nonlinear parameter identification, forecast, modeling and control.

**Keywords:** support vector machine; regression; forecast; modeling and control

## 引 言

从观测数据中学习归纳出系统规律, 并利用这些规律对未来数据或无法观测到的数据进行预测, 一直是智能系统研究的重点。这类数据驱动 (Data-Driven) 的研究方法中, 神经网络方法是最为广泛的, 通过学习, 不断修正神经网络中神经元的各权重系数, 使训练误差充分小。尽管神经网络方法可以得到小的训练误差, 但对于未经训练的新数据, 其泛化能力较差, 存在过学习问题。神经网络学习方法不成功原因在于它是基于经验风险最小化准则 (Empirical Risk Minimization, ERM)。根据统计学习理论<sup>[1][2]</sup>, 为了控制泛化能力, 需要控制两个因素, 即经验风险值和置信范围值。ERM 准则只强调了经验风险最小 (训练误差), 没有最小化置信范围值, 因此基于 ERM 准则的学习方法, 其泛化能力较差。要最大化泛化能力, 不仅需要最小化经验风险, 而且应最小化置信范围值, 这是 Vapnik 和 Chevenonkis 提出的结构风险最小化准则 (Structural Risk Minimization, SRM) 的基本思想。Vapnik 提出的支持向量机 (Support Vector Machine, SVM) 就是这种思想的具体实现, 它不同于神经网络等传统

方法以训练误差最小化作为优化目标, 而是以训练误差作为优化问题的约束条件, 以置信范围值最小化作为优化目标, 因此, SVM 的泛化能力要明显优越于神经网络等传统学习方法。另外, SVM 的求解最后转化成二次规划问题的求解, 因此, SVM 的解是唯一的、也是全局最优的。正是上述两大优点, 使 SVM 一经提出就得到了广泛的重视。

SVM 方法最早是针对模式识别问题提出来的, 随着 Vapnik 对  $e$  不敏感损失函数的引入, SVM 已推广到非线性系统的回归估计, 并展现了极好的学习性能。支持向量机方法在非线形系统辨识、预测预报、建模与控制的潜在广泛应用, 使得对其研究显得非常重要。本文描述了用于回归估计的支持向量机基本方法, 并对目前的主要研究内容进行了回顾。

## 1 回归估计的支持向量机方法

由统计学习理论知<sup>[1][2]</sup>, 对于回归估计  $f$ , 实际风险  $R_s(f)$  和经验风险  $R_{emp}(f)$  之间以至少  $1-h$  的概率 ( $h>0$ ) 满足

$$R_s(f) \leq R_{emp}(f) + \sqrt{\left(\frac{h \ln(2l/h+1) - \ln(l/4)}{l}\right)} \quad (1)$$

其中  $h$  为 VC 维,  $l$  为样本数。即实际风险受限于经验风险和置信范围 2 部分, 置信范围与学习机器的 VC 维及训练样本数有关。机器学习过程不仅要使经验风险最小, 还要使 VC 维最小 (以最小化置信范围值), 这样才能取得较小的实际风险, 对未来数据具有好的泛化能力, 这是结构风险最小化准则的基本思想。

收稿日期: 2002-10-15

修回日期: 2003-01-03

作者简介: 杜树新(1967-), 男, 浙江东阳人, 博士, 副研究员, 研究方向为智能系统、系统建模与控制等; 吴铁军(1950-), 男, 浙江杭州人, 博士, 教授, 博导, 研究方向为智能控制理论及应用等。

训练样本集假定为  $\{(x_i, y_i), i=1, 2, \dots, l\}$ , 其中  $x_i \in R^N$  为输入,  $y_i \in R$  为对应的目标值,  $l$  为样本数。定义  $\epsilon$  不敏感损失函数为

$$|y - f(x)|_{\epsilon} = \begin{cases} 0 & |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & |y - f(x)| > \epsilon \end{cases} \quad (2)$$

其中  $f(x)$  为通过对样本集的学习而构造的回归估计函数,  $y$  为  $x$  对应的目标值,  $\epsilon > 0$  为与函数估计精度直接相关的设计参数, 该  $\epsilon$  不敏感损失函数形象地比喻为  $\epsilon$  管道。学习的目的是构造  $f(x)$ , 使与目标值之间的距离小于  $\epsilon$ , 同时函数的 VC 维最小, 这样对于未知样本  $x$ , 可最优地估计出对应的目标值。

下面针对训练样本集为线性、非线性两种情况分别讨论回归估计的支持向量机方法。

### 1.1 线性情况

样本数据集为线性时, 假定  $f(x)$  为如下形式

$$f(x) = w \cdot x + b \quad (3)$$

其中  $w \cdot x$  表示向量  $w \in R^N$  与  $x \in R^N$  的内积,  $b \in R$ 。最优问题为

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i - w \cdot x_i - b \leq \epsilon \\ & w \cdot x_i + b - y_i \leq \epsilon, i=1, 2, \dots, l \end{aligned} \quad (4)$$

对线性数据集, VC 维满足<sup>[2]</sup>

$$h \leq \|w\|^2 r^2 + 1 \quad (5)$$

其中  $r$  为包络训练样本数据的最小球半径, 因此, (4) 式的最优化问题中, 最小化  $\frac{1}{2} \|w\|^2$  意味着最小化 VC 维, 同时训练误差作为最优化问题的约束条件, 因此, (4) 式的最优化问题体现了 SRM 的思想, 由此得到的回归估计函数具有较好的泛化能力。约束条件不可实现时, 引入松弛变量  $\mathbf{x}_i, \mathbf{x}_i^*$ , 这样 (4) 式写为

$$\begin{aligned} \min_{w, b, \mathbf{x}} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\mathbf{x}_i + \mathbf{x}_i^*) \\ \text{s.t.} \quad & y_i - w \cdot x_i - b \leq \epsilon + \mathbf{x}_i \\ & w \cdot x_i + b - y_i \leq \epsilon + \mathbf{x}_i^* \\ & \mathbf{x}_i \geq 0 \\ & \mathbf{x}_i^* \geq 0, i=1, 2, \dots, l \end{aligned} \quad (6)$$

其中  $C > 0$  为惩罚系数,  $C$  越大表示对超出  $\epsilon$  管道数据点的惩罚越大。采用拉格朗日乘子法求解这个具有线性不等式约束的二次规划问题, 即

$$\begin{aligned} \max_{a, a^*, b} \quad & \min_{w, b} \{L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\mathbf{x}_i + \mathbf{x}_i^*) - \sum_{i=1}^l a_i (\epsilon + \mathbf{x}_i - y_i + w \cdot x_i + b) \\ & - \sum_{i=1}^l a_i^* (\epsilon + \mathbf{x}_i^* + y_i - w \cdot x_i - b) - \sum_{i=1}^l (b \mathbf{x}_i + b^* \mathbf{x}_i^*)\} \end{aligned} \quad (7)$$

其中  $a_i, a_i^*, b, b^* \geq 0, i=1, 2, \dots, l$  为拉格朗日乘子。由此

$$\frac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_{i=1}^l (a_i - a_i^*) x_i \quad (8)$$

$$\frac{\partial L_p}{\partial b} = 0 \rightarrow \sum_{i=1}^l (a_i - a_i^*) = 0 \quad (9)$$

$$\frac{\partial L_p}{\partial \mathbf{x}_i} = 0 \rightarrow C - a_i - b_i = 0 \quad (10)$$

$$\frac{\partial L_p}{\partial \mathbf{x}_i^*} = 0 \rightarrow C - a_i^* - b_i^* = 0 \quad (11)$$

将式 (8) ~ 式 (11) 代入式 (7), 得到对偶最优化问题

$$\begin{aligned} \max_{a, a^*} \quad & \{L_D = -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (a_i - a_i^*) (a_j - a_j^*) x_i \cdot x_j \\ & - \epsilon \sum_{i=1}^l (a_i + a_i^*) + \sum_{i=1}^l y_i (a_i - a_i^*)\} \\ \text{s.t.} \quad & \sum_{i=1}^l (a_i - a_i^*) = 0 \\ & 0 \leq a_i \leq C \\ & 0 \leq a_i^* \leq C \end{aligned} \quad (12)$$

根据最优化的充要条件 (KKT 条件) 知, 在最优点, 拉格朗日乘子与约束的乘积为 0, 即

$$a_i (\epsilon + \mathbf{x}_i - y_i + w \cdot x_i + b) = 0 \quad (13)$$

$$a_i^* (\epsilon + \mathbf{x}_i^* + y_i - w \cdot x_i - b) = 0 \quad (14)$$

$$b \mathbf{x}_i = 0 \rightarrow (C - a_i) \mathbf{x}_i = 0 \quad (15)$$

$$b^* \mathbf{x}_i^* = 0 \rightarrow (C - a_i^*) \mathbf{x}_i^* = 0 \quad (16)$$

由式 (13) 和式 (14) 可得到 (反证法)

$$a_i \times a_i^* = 0 \quad (17)$$

该式说明如果  $a_i$  不为 0, 则  $a_i^*$  必为 0, 反之亦然, 因此最优化计算得到的  $a_i$  和  $a_i^*$  中, 取值必然是以下 5 种情形之一:

$$\begin{aligned} a_i = 0, a_i^* = 0; \quad & 0 < a_i < C, a_i^* = 0; \quad a_i = 0, 0 < a_i^* < C; \\ a_i = C, a_i^* = 0; \quad & a_i = 0, a_i^* = C. \end{aligned}$$

~ 所对应的  $x_i$  称之为支持向量 (Support Vector, SV),  $a_i - a_i^*$  称为支持值。由式 (8) 知, 非支持向量 ( $a_i = 0, a_i^* = 0$  所对应的  $x_i$ ) 对  $w$  没有贡献, 只有支持向量对  $w$  有贡献, 即对估计函数  $f(x)$  有贡献, 支持向量由此得名, 对应的学习方法称为支持向量机。在支持向量中, 和 对应的  $x_i$  称为边界支持向量 (Boundary Support Vector, BSV), 是超出  $\epsilon$  管道之外的数据点, 和 对应的  $x_i$  称为标准支持向量 (Normal Support Vector, NSV), 是落在  $\epsilon$  管道上的数据点。因此,  $\epsilon$  越大, 支持向量数越少, 但函数估计精度越低。

对于标准支持向量, 如果  $0 < a_i < C$  ( $a_i^* = 0$ ), 由 (15) 式知,  $\mathbf{x}_i = 0$ , 则由 (13) 式可得到

$$w \cdot x_i + b - y_i + \epsilon = 0 \quad (18)$$

这样可计算估计函数中的参数  $b$  为

$$\begin{aligned} b &= y_i - \sum_{j=1}^l (a_j - a_j^*) x_j \cdot x_i - \epsilon \\ &= y_i - \sum_{x_j \in SV} (a_j - a_j^*) x_j \cdot x_i - \epsilon \end{aligned} \quad (19)$$

同样, 对于满足  $0 < a_i^* < C$  ( $a_i = 0$ ) 的标准支持向量, 有

$$b = y_i - \sum_{x_j \in SV} (a_j - a_j^*) x_j \cdot x_i + \epsilon \quad (20)$$

为了计算可靠, 一般对所有标准支持向量分别计算  $b$  的值, 然后求平均值, 即

$$\begin{aligned} b &= \frac{1}{N_{NSV}} \{ \sum_{0 < a_i < C} [y_i - \sum_{x_j \in SV} (a_j - a_j^*) x_j \cdot x_i - \epsilon] \\ &+ \sum_{0 < a_i^* < C} [y_i - \sum_{x_j \in SV} (a_j - a_j^*) x_j \cdot x_i + \epsilon] \} \end{aligned} \quad (21)$$

式中  $N_{NSV}$  为标准支持向量数量。这样由 (8) 式和 (21) 式计算回归估计函数为

$$f(x) = \sum_{x_i \in SV} \mathbf{a}_i - \mathbf{a}_i^* x_i x + b \quad (22)$$

## 1.2 非线性情况

对于训练集为非线性情况,通过某一非线性函数  $f(\cdot)$  将训练集数据  $x$  映射到一个高维线性特征空间,在这个维数可能为无穷大的线性空间中构造回归估计函数,因此,在非线形情况,估计函数  $f(x)$  为如下形式

$$f(x) = w \cdot f(x) + b \quad (23)$$

其中  $w$  的维数为特征空间维数(可能为无穷维)。最优化问题为

$$\begin{aligned} \min_{w, b, x} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (x_i + x_i^*) \\ \text{s.t.} \quad & y_i - w \cdot f(x_i) - b \leq e + x_i \\ & w \cdot f(x_i) + b - y_i \leq e + x_i^* \\ & x_i \geq 0 \\ & x_i^* \geq 0, i = 1, 2, \dots, l \end{aligned} \quad (24)$$

类似线性情况,得到对偶最优化问题

$$\begin{aligned} \max_{\mathbf{a}, \mathbf{a}^*} \{ L_D = & -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\mathbf{a}_i - \mathbf{a}_i^*)(\mathbf{a}_j - \mathbf{a}_j^*) K(x_i, x_j) \\ & - e \sum_{i=1}^l (\mathbf{a}_i + \mathbf{a}_i^*) + \sum_{i=1}^l y_i (\mathbf{a}_i - \mathbf{a}_i^*) \} \\ \text{s.t.} \quad & \sum_{i=1}^l (\mathbf{a}_i - \mathbf{a}_i^*) = 0 \\ & 0 \leq \mathbf{a}_i \leq C \\ & 0 \leq \mathbf{a}_i^* \leq C \end{aligned} \quad (25)$$

其中  $K(x_i, x_j) = f(x_i) \cdot f(x_j)$  称为核函数,回归估计函数为

$$f(x) = \sum_{x_i \in SV} (\mathbf{a}_i - \mathbf{a}_i^*) K(x_i, x) + b \quad (26)$$

其中  $b$  按如下计算

$$\begin{aligned} b = & \frac{1}{N_{NSV}} \left\{ \sum_{0 < \mathbf{a}_i < C} [y_i - \sum_{x_j \in SV} (\mathbf{a}_j - \mathbf{a}_j^*) K(x_j, x_i) - e] \right. \\ & \left. + \sum_{0 < \mathbf{a}_i^* < C} [y_i - \sum_{x_j \in SV} (\mathbf{a}_j - \mathbf{a}_j^*) K(x_j, x_i) + e] \right\} \end{aligned} \quad (27)$$

由 (26) 式和 (27) 式知,尽管通过非线性函数将样本数据映射到具有高维甚至为无穷维的特征空间,但在计算回归估计函数时并不需要显式计算该非线性函数,而只需计算核函数,从而避免高维特征空间引起的维数灾难问题。核函数的选择必须满足 Mercer 条件,常见的核函数有线性函数  $K(x_i, x) = x_i \cdot x$ 、多项式函数  $K(x_i, x) = (x_i \cdot x + 1)^d$ 、径向基函数  $K(x_i, x) = \exp(-\|x - x_i\|^2 / s^2)$ 、多层感知器函数  $K(x_i, x) = \tanh(kx_i \cdot x + q)$ 。

## 1.3 关于支持向量机方法的说明

(1) (25) 式的对偶二次规划问题是一个具有  $2l$  个优化变量,  $1$  个线性等式约束,  $4l$  个线性不等式约束的二次规划问题,因此,求解规模与样本数量有关,而与  $x$  的维数没有关系,这是 SVM 的重要特征之一。利用这一特征,在进行

回归估计时,一般不需要进行特征提取等一些预处理。对于这种二次规划问题可以得到唯一的、全局最优解,因此 SVM 不存在局部最优问题。

(2) 最优化计算得到的  $\mathbf{a}_i$  和  $\mathbf{a}_i^*$  中,多数满足  $\mathbf{a}_i = 0$ ,  $\mathbf{a}_i^* = 0$  (即  $\mathbf{a}_i - \mathbf{a}_i^* = 0$ )。在样本数据中非支持向量占多数,因此,只需少量样本(支持向量)就可构成回归估计函数,使得有用的样本数据大大压缩。解的稀疏性是 SVM 的重要特征之一,使得 SVM 在实际问题求解中具有很强的实用性。通过改变  $e$  值,可以控制 SVM 解的稀疏性。

(3) 采用支持向量机求解回归估计问题时,需选择一核函数。尽管只要满足 Mercer 条件的函数在理论上都可选为核函数,但对于特定问题,由不同的核函数得到的回归估计也会有很大不同,不存在性能优越于其他核函数的核函数。因此,针对某一特定问题,如何选择核函数是至关重要的。另外,即使选择了某类核函数,其相应的参数(如多项式函数的阶次、径向基参数  $s$ )也有如何选择的问题。尽管许多研究者已认识到核函数及其参数选择的重要性,并展开了研究,但并没有有效的方法。

(4) 对于设计参数  $C$  和  $e$  的选择,存在同样问题。 $C$  取得小,则对样本数据中超出  $e$  管道的样本惩罚就小,使训练误差变大,系统的泛化能力变差。 $C$  取得大,相应地  $\frac{1}{2} \|w\|^2$  (即 VC 维)的权重就少,系统的泛化能力变差。同样,  $e$  选小,回归估计精度高,但支持向量数量增多,  $e$  选大,回归估计精度降低,支持向量数量少, SVM 的稀疏性大。在实际问题求解时,如何合适选取这些参数,目前没有有效的方法,一般通过交叉验证法选取适当的参数。

## 2 支持向量机研究内容

### 2.1 支持向量机新方法的研究

#### 2.1.1 $n$ -SVM

在第 2 节介绍的标准 SVM 中,通过参数  $e$  控制回归估计的精度,但  $e$  取多少才能达到所期望的估计精度是不明确的,因此尽管想实现高精度估计,但具体  $e$  应选多少是难以把握的。在 Schölkopf 和 Smola<sup>[3]</sup>提出的  $n$ -SVM 方法中,引入反映超出  $e$  管道之外样本数据点(即边界支持向量数量)和支持向量数的新参数  $n$ ,从而简化 SVM 的参数调节。在  $n$ -SVM 中,优化目标为

$$\begin{aligned} \min_{w, b, x} \quad & \frac{1}{2} \|w\|^2 + C(n\mathbf{e} + \frac{1}{l} \sum_{i=1}^l (x_i + x_i^*)) \\ \text{s.t.} \quad & y_i - w \cdot f(x_i) - b \leq \mathbf{e} + x_i \\ & w \cdot f(x_i) + b - y_i \leq \mathbf{e} + x_i^* \\ & x_i \geq 0 \\ & x_i^* \geq 0, i = 1, 2, \dots, l \\ & \mathbf{e} > 0 \end{aligned} \quad (28)$$

得到对偶最优化问题

$$\begin{aligned} \max_{\mathbf{a}, \mathbf{a}^*} \{ L_D = -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\mathbf{a}_i - \mathbf{a}_i^*)(\mathbf{a}_j - \mathbf{a}_j^*) K(\mathbf{x}_i, \mathbf{x}_j) \\ + \sum_{i=1}^l y_i (\mathbf{a}_i - \mathbf{a}_i^*) \} \\ \text{s.t.} \quad \sum_{i=1}^l (\mathbf{a}_i - \mathbf{a}_i^*) = 0 \\ 0 \leq \mathbf{a}_i \leq C/l \\ 0 \leq \mathbf{a}_i^* \leq C/l \\ \sum_{i=1}^l (\mathbf{a}_i + \mathbf{a}_i^*) \leq Cn \end{aligned} \quad (29)$$

由 KKT 条件, 在最优点满足

$$\sum_{i=1}^l (\mathbf{a}_i + \mathbf{a}_i^*) = Cn \quad (30)$$

对于边界支持向量 BSV,  $\mathbf{a}_i = C/l$  且  $\mathbf{a}_i^* = 0$  (或  $\mathbf{a}_i^* = C/l$  且  $\mathbf{a}_i = 0$ ), 因此对于  $N_{BSV}$  个边界支持向量, 有  $N_{BSV} C/l \leq \sum_{i=1}^l (\mathbf{a}_i + \mathbf{a}_i^*) = Cn$ , 即  $n \geq N_{BSV}/l$ ; 而对于支持向量,  $\mathbf{a}_i \leq C/l$  且  $\mathbf{a}_i^* = 0$  (或  $\mathbf{a}_i^* \leq C/l$  且  $\mathbf{a}_i = 0$ ), 因此对于  $N_{SV}$  个支持向量, 有  $\sum_{i=1}^l (\mathbf{a}_i + \mathbf{a}_i^*) = \sum_{k=1}^{N_{SV}} (\mathbf{a}_k + \mathbf{a}_k^*) \leq N_{SV} C/l$ , 因此有  $Cn \leq N_{SV} C/l$ , 即  $N_{SV}/l \geq n$ , 因此有

$$N_{BSV}/l \leq n \leq N_{SV}/l \quad (31)$$

由上式知,  $l n$  表示 BSV 数量上限和 SV 数量的下限。由于在优化求解过程不需要  $\mathbf{e}$  的值, 因此不需要预先规定  $\mathbf{e}$  取多少值。

### 2.1.2 LS-SVM

在 Suykens<sup>[4]</sup>提出的最小二乘支持向量机 (LS-SVM) 中, 优化指标采用了平方项, 从而将不等式约束转变成等式约束。最优化问题为

$$\begin{aligned} \min_{w, b, \mathbf{x}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{2} \mathbf{g}^T \sum_{i=1}^l \mathbf{x}_i^2 \\ \text{s.t.} \quad y_i = \mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i) + b + \mathbf{x}_i, \quad i = 1, 2, \dots, l \end{aligned} \quad (32)$$

可得到线性方程组

$$\begin{bmatrix} 0 & \mathbf{e}^T \\ \mathbf{e} & \mathbf{Q} + \mathbf{g}^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \quad (33)$$

其中  $\mathbf{e} \in R^l$  为元素为 1 的向量,  $\mathbf{I} \in R^{l \times l}$  为单位阵,  $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_l]^T \in R^l$ ,  $\mathbf{y} = [y_1, y_2, \dots, y_l]^T \in R^l$ ,  $\mathbf{Q} = [q_{ij}]_{l \times l}$ ,  $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ 。因此在 LS-SVM, 将二次规划问题转变成线性方程组的求解, 另外, 与标准 SVM 相比减少了一个调整参数, 减少了  $l$  个优化变量, 从而简化了计算复杂性。但是 LS-SVM 没有稀疏性这一特性。

### 2.1.3 W-SVM

第 1 节描述的标准 SVM 方法中, 引入惩罚系数  $C$  实行对超出  $\mathbf{e}$  管道数据点的惩罚。在实际应用中, 某些问题如用于股市预测、期货预测的时间序列预测问题, 近期数据的重要性要大于早期数据的重要性, 也就是说某些重要样本数据点要求小的训练误差, 有些样本数据点对误差的要求不是很高。因此, 在优化问题描述时, 对每个样本点应采用不同的惩罚系数  $C$ , 或对于每个样本数据点应采用大小不同的  $\mathbf{e}$  管道, 以得到更准确的回归估计, 这种支持向量机我们称之为

加权支持向量机 (W-SVM)。加权支持向量机可以通过对惩罚系数  $C$  的加权实现, 也可以通过对  $\mathbf{e}$  加权实现。通过对参数  $C$  加权实现时, 其最优化问题为

$$\begin{aligned} \min_{w, b, \mathbf{x}, \mathbf{x}^*} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l s_i (\mathbf{x}_i + \mathbf{x}_i^*) \\ \text{s.t.} \quad y_i - \mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i) - b \leq \mathbf{e} + \mathbf{x}_i \\ \mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i) + b - y_i \leq \mathbf{e} + \mathbf{x}_i^* \\ \mathbf{x}_i \geq 0 \\ \mathbf{x}_i^* \geq 0, \quad i = 1, 2, \dots, l \end{aligned} \quad (34)$$

其中  $s_i$  为加权系数。得到对偶最优化问题

$$\begin{aligned} \max_{\mathbf{a}, \mathbf{a}^*} \{ L_D = -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\mathbf{a}_i - \mathbf{a}_i^*)(\mathbf{a}_j - \mathbf{a}_j^*) K(\mathbf{x}_i, \mathbf{x}_j) \\ - \mathbf{e} \sum_{i=1}^l (\mathbf{a}_i + \mathbf{a}_i^*) + \sum_{i=1}^l y_i (\mathbf{a}_i - \mathbf{a}_i^*) \} \\ \text{s.t.} \quad \sum_{i=1}^l (\mathbf{a}_i - \mathbf{a}_i^*) = 0 \\ 0 \leq \mathbf{a}_i \leq C s_i \\ 0 \leq \mathbf{a}_i^* \leq C s_i \end{aligned} \quad (35)$$

W-SVM 需要合理选择加权系数  $s_i$ , 文献[5]认为在样本集中第 1 个样本的重要性最低, 最后一个样本的重要性最高, 再采用线性插值方法得到其他样本点的加权系数。文献[6]针对金融时间序列预测问题建议采用线性加权

$$s_i = \frac{i}{l(l+1)/2} \quad (36)$$

和指数加权

$$s_i = \frac{1}{1 + \exp(p - 2pi/l)} \quad (37)$$

其中  $p$  为控制衰减速率的参数。

通过对  $\mathbf{e}$  加权实现 W-SVM 时, 其优化问题为

$$\begin{aligned} \min_{w, b, \mathbf{x}, \mathbf{x}^*} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\mathbf{x}_i + \mathbf{x}_i^*) \\ \text{s.t.} \quad y_i - \mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i) - b \leq \mathbf{e}_i + \mathbf{x}_i \\ \mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i) + b - y_i \leq \mathbf{e}_i + \mathbf{x}_i^* \\ \mathbf{x}_i \geq 0 \\ \mathbf{x}_i^* \geq 0, \quad i = 1, 2, \dots, l \end{aligned} \quad (38)$$

得到对偶最优化问题为

$$\begin{aligned} \max_{\mathbf{a}, \mathbf{a}^*} \{ L_D = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\mathbf{a}_i - \mathbf{a}_i^*)(\mathbf{a}_j - \mathbf{a}_j^*) K(\mathbf{x}_i, \mathbf{x}_j) \\ - \sum_{i=1}^l \mathbf{e}_i (\mathbf{a}_i + \mathbf{a}_i^*) + \sum_{i=1}^l y_i (\mathbf{a}_i - \mathbf{a}_i^*) \} \\ \text{s.t.} \quad \sum_{i=1}^l (\mathbf{a}_i - \mathbf{a}_i^*) = 0 \\ 0 \leq \mathbf{a}_i \leq C \\ 0 \leq \mathbf{a}_i^* \leq C \end{aligned} \quad (39)$$

在文献[7]中,  $\mathbf{e}_i$  按如下取值

$$\mathbf{e}_i = \mathbf{e} \frac{1 + \exp(p - 2pi/l)}{2} \quad (40)$$

### 2.1.4 基于线性规划的 SVM

在标准 SVM 中, 优化目标是一个  $L_2$  范数, 也正因为  $L_2$  范数, 使最优化问题转化成二次规划问题。Kecman 和

Hadzic<sup>[8]</sup>提出用  $L_1$  范数替代  $L_2$  范数,从而将最优化问题转化成线性规划问题。回归估计函数定义为  $f(x)=Gw$ , 其中  $G$  为核函数,  $w \in R^l$ , 这样最优化问题为

$$\min \|w\| \quad (41)$$

$$s.t. \|Gw - y\|_{\infty} \leq e$$

定义  $w_i = w_i^+ - w_i^-$ ,  $|w_i| = w_i^+ + w_i^-$ ,  $w_i^+ \geq 0$ ,

$w_i^- \geq 0$ ,  $i=1,2,\dots,l$   $w^+ = [w_1^+, w_2^+, \dots, w_l^+]^T$ ,  $w^- = [w_1^-, \dots, w_l^-]^T$   
则上述优化问题转化成如下的线性规划问题

$$\min \sum_i (w_i^+ + w_i^-) \quad (42)$$

$$s.t. G(w^+ - w^-) \leq y + e e$$

$$G(w^+ - w^-) \leq -y + e e$$

$$w_i^+ \geq 0, w_i^- \geq 0, i=1,2,\dots,l$$

线性 SVM 的一个突出问题是线性规划的约束数与  $w$  的维数有关,这样对于非线性样本集,实际上是无法求解的。

## 2.2 大样本数的优化算法研究

由第 2 节描述可知,采用支持向量机方法求解回归估计问题,本质上是一个二次规划问题。对于  $l$  个样本数据,这个二次规划包括了  $2l$  个优化变量,1 个线性等式约束,  $4l$  个线性不等式约束,同时还涉及到  $l \times l$  维核函数矩阵的计算及矩阵与向量的相乘运算,因此求解规模与样本数量有关。样本数非常小时(2~4 个样本),可以采用解析方法计算得到解析解。但随着样本数的增加,已无法采用解析方法求解二次规划问题,需采用诸如牛顿法、共轭梯度法等数值计算方法。对于大样本数,尤其是时间序列问题,它的样本数不断增加,采用常规的优化算法及软件已无法实现二次规划问题的求解。运行时间及计算内存是大样本数求解 SVM 的主要瓶颈。针对大样本数据如何减少二次规划求解过程的计算时间和内存一直是 SVM 的研究热点,目前主要有以下几种方法。

### 2.2.1 分解法(Decomposition Method)

分解法是将大规模的二次规划问题转化成一系列小规模二次规划问题。分解法的基本思想是将大样本数据集分成 2 个子集,即大小固定为  $q$  的工作集  $B$  和非工作集  $N$ ,  $B \cap N = \emptyset$ ,  $B \cup N = \{1,2,\dots,l\}$ , 工作集对应的  $a, a^*$  作为优化变量(记为  $a_B, a_B^*$ ),而非工作集的  $a, a^*$ (记为  $a_N, a_N^*$ )固定不变。这样,大规模的二次规划问题转化成只有  $2q$  个优化变量、1 个线性约束、 $4q$  个线性不等式约束的小规模规划问题

$$\max_{a_B} \{L_D = -\frac{1}{2} \sum_{i,j \in B} (a_i - a_j)(a_i - a_j) K(x_i, x_j) + \sum_{i \in B} (a_i - a_i^*)(y_i - \sum_{j \in N} (a_j - a_j^*) K(x_i, x_j)) - e \sum_{i \in B} (a_i + a_i^*)\} \quad (43)$$

$$s.t. \sum_{i \in B} (a_i - a_i^*) = -\sum_{i \in N} (a_i - a_i^*)$$

$$0 \leq a \leq C$$

$$0 \leq a_i^* \leq C, i \in B$$

工作集的选择是再优化过程,需要多次迭代才会得到最优的  $a$  和  $a^*$ 。分解法基本步骤为

初始化。满足(17)式的  $a^1$ 、 $a^{*1}$  为初始值,  $k=1$  ( $k$  表示第  $k$  次迭代);

如果  $a^k$ 、 $a^{*k}$  满足 KKT 条件(即  $a^k$  为最优值), 结束。否则 选一大大小为  $q$  的工作集  $B \subset \{1,2,\dots,l\}$ , 余下的属于  $N$ ; 求解优化问题式(43), 计算得到  $a_B^k$ ;

$a_B^{k+1} = a_B^k, a_B^{*k+1} = a_B^{*k}, a_N^{k+1} = a_N^k, a_N^{*k+1} = a_N^{*k}, k=k+1$ , 返回。

每次迭代时,工作集  $B$  是重新选择的。分解法的关键问题在每次迭代过程中如何选择工作集  $B$ 。Osuna<sup>[9]</sup>等根据 KKT 条件背离程度选择工作集,而 Loskov<sup>[10]</sup>、Collobert<sup>[11]</sup>采用可行方向法选择工作集, Loskov<sup>[12]</sup>提出最大不一致性(Maximal Inconsistency)方法确定工作集  $B$ 。

由于分解法中的工作集大小  $q$  固定不变,因此子问题的求解不会随着样本数的增加而增加,是固定不变的。

### 2.2.2 顺序优化算法(Sequential Minimal Optimization, SMO)

SMO 算法是分解法的特例,它是将工作集大小限定为 2, 这样在进行子问题求解时直接由解析法计算最优值  $a_B, a_B^*$ , 而不需要采用数值优化算法进行计算。SMO 是由 Platt<sup>[13]</sup>最早提出并应用于模式识别中的支持向量机求解, Smola 和 Schölkopf<sup>[14]</sup>针对回归估计提出了 SMO 算法。许多研究者对该 SMO 算法进行了修正<sup>[15]</sup>。在这里介绍一种简单而又可行的 SMO 方法<sup>[16]</sup>。令

$$I_i = a_i - a_i^*, |I_i| = a_i + a_i^*, -C \leq I_i \leq C \quad (44)$$

通过  $I_i$  的引入,将  $2l$  个优化变量减少到  $l$  个优化变量,这样回归函数及最优化问题分别为

$$f(x) = \sum_{i=1}^l I_i K(x_i, x) + b \quad (45)$$

$$\min \{L_D = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l I_i I_j k_{ij} + e \sum_{i=1}^l |I_i| - \sum_{i=1}^l I_i y_i\}$$

$$s.t. \sum_{i=1}^l I_i = 0 \quad (46)$$

$$-C \leq I_i \leq C, i=1,2,\dots,l$$

其中  $k_{ij} = K(x_i, x_j)$ , 且核函数对称, 即  $k_{ij} = k_{ji}$ 。下标  $u, v$  表示工作集中的 2 个样本数据, 则  $I_u, I_v$  即为优化变量, 其他  $I_i$  固定为常数, 这样上式的最优化问题成为

$$L_D = e |I_u| + e |I_v| - I_u y_u - I_v y_v + \frac{1}{2} I_u^2 k_{uu} + \frac{1}{2} I_v^2 k_{vv} + I_u I_v k_{uv} + I_u z_u^- + I_v z_v^- + L_C \quad (47)$$

其中  $L_C$  为与非工作集相关的常数,

$$z_i^- = \sum_{j=1, j \neq u, v}^l I_j^- k_{ij} = f_i^- - I_u^- k_{ui} - I_v^- k_{vi} - b^- \quad (48)$$

$$f_i^- = f(x_i, I^-, b)$$

上标  $-$  表示上一次迭代过程计算得到的数据。由于  $\sum_{i=1}^l I_i = 0$ ,

因此  $I_u, I_v$  两者之和不变, 即  $I_u + I_v = I_u^- + I_v^- = s^-$ , 因此有

$$I_u = s^- - I_v \quad (49)$$

则最优化问题为

$$L_D = \mathbf{e} |s^- - \mathbf{I}_v| + \mathbf{e} |\mathbf{I}_v| - (s^- - \mathbf{I}_v) y_u - \mathbf{I}_v y_v + \frac{1}{2} (s^- - \mathbf{I}_v)^2 k_{uu} + \frac{1}{2} \mathbf{I}_v^2 k_{vv} + (s^- - \mathbf{I}_v) \mathbf{I}_v k_{uv} + (s^- - \mathbf{I}_v) z_u + \mathbf{I}_v z_v + L_C \quad (50)$$

则

$$\frac{\partial L_D}{\partial \mathbf{I}_v} = 0 \rightarrow \mathbf{e} (\text{sgn}(\mathbf{I}_v) - \text{sgn}(s^- - \mathbf{I}_v)) + y_u - y_v + (\mathbf{I}_v - s^-) k_{uu} + \mathbf{I}_v k_{vv} + (s^- - 2\mathbf{I}_v) k_{uv} - z_u + z_v = 0 \quad (51)$$

推导过程中使用了  $\frac{d|x|}{dx} = \text{sgn}(x)$ 。由(51)式可得到

$$\begin{aligned} \mathbf{I}_v (k_{vv} + k_{uu} - 2k_{uv}) &= y_v - y_u \\ &+ \mathbf{e} (\text{sgn}(\mathbf{I}_u) - \text{sgn}(\mathbf{I}_v)) + s^- (k_{uu} - k_{uv}) + z_u - z_v \\ &= y_v - y_u + f_u^- - f_v^- + \mathbf{e} (\text{sgn}(\mathbf{I}_u) - \text{sgn}(\mathbf{I}_v)) + \mathbf{I}_u k_{uu} \\ &- \mathbf{I}_v k_{uv} + \mathbf{I}_v k_{uu} - \mathbf{I}_v k_{uv} - \mathbf{I}_u k_{uu} - \mathbf{I}_v k_{uv} - b^- + \mathbf{I}_u k_{uv} + \mathbf{I}_v k_{vv} + b^- \\ &= y_v - y_u + f_u^- - f_v^- + \mathbf{e} (\text{sgn}(\mathbf{I}_u) - \text{sgn}(\mathbf{I}_v)) + \mathbf{I}_v (k_{vv} + k_{uu} - 2k_{uv}) \end{aligned} \quad (52)$$

从而得到迭代算法

$$\mathbf{I}_v = \mathbf{I}_v^- + \frac{1}{t} (y_v - y_u + f_u^- - f_v^- + \mathbf{e} (\text{sgn}(\mathbf{I}_u) - \text{sgn}(\mathbf{I}_v))) \quad (53)$$

其中  $t = k_{vv} + k_{uu} - 2k_{uv}$ 。上式中由于等式右边包含了  $\text{sgn}(\mathbf{I}_u)$  和  $\text{sgn}(\mathbf{I}_v)$ , 因此是一个递推求解过程。参数  $b$  根据  $f_u = y_u$  计算, 即

$$\begin{aligned} y_u &= f_u = \sum_{i=1}^l \mathbf{I}_i k_{iu} + b \\ &= \mathbf{I}_u k_{uu} + \mathbf{I}_v k_{uv} + \sum_{j=1, j \neq u, v}^l \mathbf{I}_j k_{ju} + b \\ &= \mathbf{I}_u k_{uu} + \mathbf{I}_v k_{uv} + f_u^- - \mathbf{I}_u k_{uu} - \mathbf{I}_v k_{uv} - b^- + b \end{aligned} \quad (54)$$

因此有

$$b = y_u - f_u^- + (\mathbf{I}_u - \mathbf{I}_u) k_{uu} + (\mathbf{I}_v - \mathbf{I}_v) k_{uv} + b^- \quad (55)$$

同样根据  $f_v = y_v$  可计算  $b$  为

$$b = y_v - f_v^- + (\mathbf{I}_u - \mathbf{I}_u) k_{uv} + (\mathbf{I}_v - \mathbf{I}_v) k_{vv} + b^- \quad (56)$$

取两者的平均即为  $b$  的值。在最优点, 两者的值是相等的。

SMO 的 KKT 条件为

$$\begin{cases} |y_i - f_i| \leq \mathbf{e} & \mathbf{I}_i = 0 \\ |y_i - f_i| = \mathbf{e} & -C < \mathbf{I}_i < C \text{ 且 } \mathbf{I}_i \neq 0 \\ |y_i - f_i| \geq \mathbf{e} & |\mathbf{I}_i| = C \end{cases} \quad (57)$$

该条件可作为程序终止的条件, 也就是说如果求解得到的  $\mathbf{I}_i$  都满足上述条件, 则达到了最优点。

### 2.2.3 增量学习法 (Incremental Learning)

由支持向量机方法得到的回归估计只与支持向量有关, 与其他样本数据无关, 也就是说, 如果只取支持向量作为训练样本, 得到的回归估计与所有样本作为训练样本得到的回归估计是一致的, 这样可采用增量学习方法进行大样本数据的优化计算<sup>[17]</sup>。在增量学习过程中, 将大样本数据集分成若干个小规模的样本集, 按顺序逐步对各个样本子集进行增量学习, 在对每个样本子集增量学习时, 只需要根据上一个样本子集得到的支持向量以及当前的样本子集进行新的最优化计算, 这种增量学习方法在支持向量数目非常多时, 与

大样本数据优化计算一样, 求解困难。

## 3 结论

基于支持向量机方法的回归估计以可控制的精度逼近任一非线性函数, 同时具有全局最优、良好的泛化能力等优越性能, 因此支持向量机的应用非常广泛。目前支持向量机主要应用在金融时间序列预测(如股票预测、期货预测等)<sup>[6-7, 18-20]</sup>和非线性系统参数辨识<sup>[21-22]</sup>、建模<sup>[23-24]</sup>与控制<sup>[25-26]</sup>等一些方面。

本文针对回归估计问题, 描述了支持向量机的基本方法及当前的主要研究内容。作为新的非线性函数逼近方法, 支持向量机仍有许多尚未解决或尚未充分解决的问题, 尤其在应用方面的研究仍相当缺乏。以系统建模与控制为例, 正如神经网络方法在这方面的作为, 支持向量机方法也可大有作为, 但需要解决的关键是在线训练算法的研究。在采用神经网络进行非线性建模时, 预先规定了网络结构及节点数, 这样在后续的学习过程中, 只需对固定节点数的权重进行学习修正, 其修正算法可以是一个迭代过程。而支持向量机不同, 随着新学习样本的加入, 不仅支持向量对应的支持值发生变化, 支持向量的数量也可能会增加, 如何采用迭代方式在线实时修正支持向量及其支持值是将 SVM 应用于建模与控制的关键。增量学习方法根据前一次学习结果及当前的学习样本数据得到新的支持向量及支持值, 但并不是以迭代方式实现的, 而是一次性计算的, 因此在支持向量数量较多时, 其求解过程非常缓慢, 不适宜在线调整。

## 参考文献:

- [1] Vapnik V N. 统计学习理论的本质[M]. 张学工. 北京: 清华大学出版社, 2000.
- [2] Vapnik V N. Statistical learning theory[M]. New York, 1998.
- [3] Schölkopf B, Smola A J, Bartlett P L. New support vector algorithms[J]. Neural Computation, 2000, 12: 1207-1245.
- [4] Suykens J A K, Branbanter J K, Lukas L, et al. Weighted least squares support vector machines: robustness and sparse approximation [J]. Neurocomputing, 2002, 48(1): 85-105.
- [5] Lin C-F, Wang S-D. Fuzzy support vector machines[J]. IEEE Trans on Neural Networks, 2002, 13(2): 464-471.
- [6] Tay F E H, Cao L J. Modified support vector machines in financial time series forecasting[J]. Neurocomputing, 2002, 48: 847-861.
- [7] Tay F E H, Cao L J.  $\mathbf{e}$ -Descending support vector machines for financial time series forecasting[J]. Neural Processing Letters, 2002, 15(2): 179-195.
- [8] Keoman V, Hadzic I. Support vectors selection by linear programming[A]. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks[J]. Como, Italy, 2000, 5: 193-198.
- [9] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machine[A]. Proc. the 1997 IEEE workshop on neural networks for signal processing[C]. Amelie Island, FL, 1997, 276-285.
- [10] Laskov P. Feasible direction decomposition algorithms for training support vector machines[J]. Machine Learning, 2002, 46(1): 315-349.
- [11] Collobert R. SVM-Torch: support vector machines for large-scale regression problems[J]. Journal of Machine Learning Research, 2001, 1: 143-160.

(下转第 1633 页)

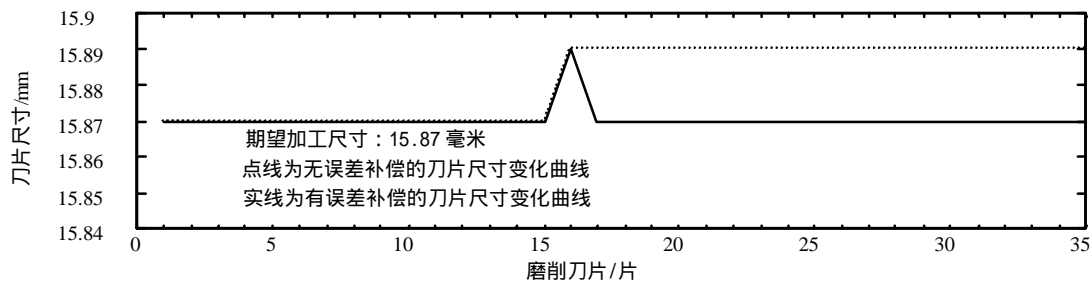


图2 智能误差补偿控制系统的阶跃响应

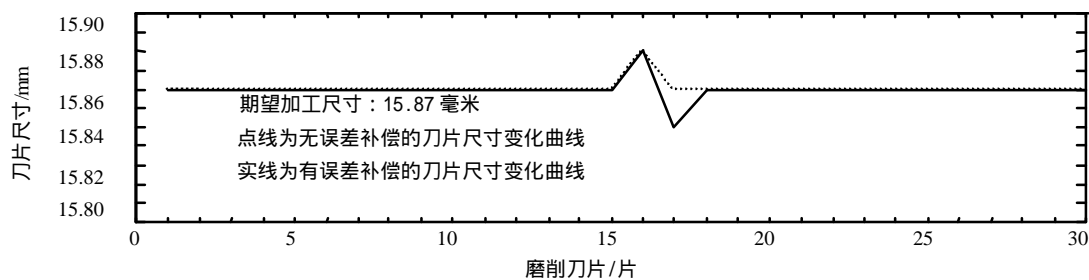


图3 智能误差补偿控制系统的脉冲响应

### 3 系统输入输出响应

通过对上述算法进行仿真,得到如图2和图3所示的智能误差补偿系统输入输出响应。

### 4 结论

由系统的输入输出响应可以看出:系统对应阶跃输入的输出经过两个学习周期即可达到稳定的期望输出;对应脉冲输入的系统输出经过三个学习周期即可达到稳定的期望输出;系统的输出始终以期望输出为稳定值,系统具有良好的稳定性。

### 参考文献:

(上接第1585页)

- [12] Laskov P. An improved decomposition algorithm for regression support vector machines[A]. In: Advances in Neural Information Processing System[C] 12, Solla S A, Leen T K, Müller K-R. MIT Press, 2000, 484-490.
- [13] Platt J C. Fast training of support vector machines using sequential minimal optimization[A]. In: Schölkopf B, Burges C J C, Smola A J (Eds). Advances in kernel method-support vector learning[C]. Cambridge, MA:MIT Press, 1999, 185-208.
- [14] Smola A J, Schölkopf B. A tutorial on support vector regression[R]. Royal Holloway College, NeuroCOLT Technical Report TR-1998-030, 1998.
- [15] Shevade S K, et al. Improvements to the SMO algorithms for SVM regression[J]. IEEE Trans on Neural Networks, 2000, 11(5): 1188-1193.
- [16] Flake G W, Lawrence S. Efficient SVM regression training with SMO[J]. Machine Learning, 2002, 41(1): 271-290.
- [17] Carozza M, Rampone S. Towards an incremental SVM for regression[A]. Proc. of the IEEE-ENNS Int Joint Conf on Neural Networks[C]. 2000, 6: 405-410.
- [18] Trafalis T B, Inco H. Support vector machine for regression and applications to financial forecasting[A]. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks[C]. Como, Italy, 2000, 6: 348-353.
- [19] Tay F E H, Cao L. Application of support vector machines in financial time series forecasting[J]. Omega, 2001, 29(4): 309-317.
- [20] Gestel T V, et al. Financial time series prediction using least squares support vector machines within the evidence framework[J]. IEEE Trans on Neural Networks, 2001, 12(4): 809-821.
- [21] Drezet P M L, Harrison R F. Support vector machines for system identification[A]. UKACC Int Conf on Control[C], 1998, 688-692.
- [22] Gretton A, et al. Support vector regression for black-box system identification[A]. Proceedings of the 11<sup>th</sup> IEEE Signal Processing Workshop on Statistical Signal Processing[C]. Singapore, 2001, 341-344.
- [23] Suykens J. Nonlinear modeling and support vector machines[A]. IEEE Instrumentation and Measurement Technology Conference[C]. Budapest, Hungary, 2001, 287-294.
- [24] Chan W C, Chan C W, Cheung K C, et al. On the modelling of nonlinear dynamic systems using support vector neural networks[J]. Engineering Application of Artificial Intelligence, 2001, 14(2): 105-113.
- [25] Kruif B J, Vries T J A. On using a support vector machine in learning feed-forward control[A]. Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics[C]. Como, Italy, 2001, 1: 272-277.