文章编号: 1001-0920(2006)07-0837-04

支持向量回归的一种网络优化策略

王 玲, 穆志纯, 郭 辉
(北京科技大学 信息工程学院, 北京 100083)

摘 要: 提出一种用于支持向量回归的网络优化策略. 学习策略分为两个阶段: 首先训练支持向量机, 得到支持向量回归的初始结构和参数, 构造一个无阈值的支持向量回归网络; 然后通过带有遗忘因子的递归最小二乘算法, 优化计算支持向量回归网络的权值, 以达到更好的函数拟合精度. 与支持向量回归相比, 这种策略可以得到最优的权值和阈值. 仿真结果表明, 该网络性能优良, 具有在线应用的潜力.

关键词: 支持向量回归; 递归最小二乘算法; 网络优化
中图分类号: TP393 **文献标识码**: A

Network Optimization Strategy of Support Vector Regression

WANG Ling, MU Zhi-chun, GUO Hui
(School of Information Engineering, Beijing University of Science and Technology, Beijing 100083, China. Correspondent: WANG Ling, E-mail: linda_gh@sina.com)

Abstract: A new network optimization strategy of support vector regression (SVR) is presented. The proposed learning method includes two steps: First, the initial structure and parameters of SVR are obtained by learning, which is applied to construct support vector regression network without bias. Then the weights of SVRN are updated by the recursive least square method with forgetting factor for the better function approximation accuracy. Compared with the SVR, it can find the optimal weights and bias. The simulation result shows the effectiveness of the optimization method and can be applied to model online.

Key words: Support vector regression; Recursive least squares; network optimization

1 引 言

支持向量机(SVM)是 20 世纪 90 年代提出的一种新的学习机[1,2]. 与神经网络等现有的学习机相比, SVM 是基于统计学习理论中结构风险最小化准则的具体实现, 明显优于传统的基于经验风险最小化的神经网络. 支持向量机在解决小样本和非线性问题方面表现出良好的泛化能力, 已在分类、时间预测、函数估计等领域得到了广泛应用. 由于核函数利用隐式非线性变换, 巧妙地解决了维数灾难问题, 可以有效地解决高维函数估计问题, 且不存在局部最优问题, 近几年得到了广泛的研究.

SVM 用于非线性系统的回归函数估计, 通常称为 SVR[3~8]. 如果 SVR 的核函数选择径向基函数, 则其结构类似于径向基神经网络. 径向基神经网络的基函数中心点难以确定, 而在标准的支持向量回归中, 每个基函数中心对应一个支持向量, 它们及输出权值都是由算法自动确定的. 但阈值的求解通常根据最优化理论取边界上的一点或其平均值, 这样获得的阈值并不是最优解.

本文提出一种新的适用性更广的支持向量回归的网络优化策略. 与 SVR 相比, 它的输出是无阈值的, 其输出权值可通过带有遗忘因子的递归最小二乘算法进行在线优化. 该算法可提高模型的精度, 具有良好的在线应用潜力.

收稿日期: 2005-05-12; **修回日期**: 2005-08-31.
基金项目: 国家 863 计划项目(2002AA412010); 科技部科技攻关项目(2003EG113016); 北京市教委重点学科共建项目.
作者简介: 王玲(1974—), 女, 北京人, 博士, 从事人工智能、机器学习等研究; 穆志纯(1952—), 男, 天津人, 教授, 博士生导师, 从事人工智能、模式识别等研究.

2 支持向量回归的网络优化策略

2.1 支持向量回归

支持向量机函数回归[1] 问题可以表述为: 给定训练样本集$\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \cdots, (\vec{x}_N, y_N)\}$, 其中 $\vec{x}_i \in R^m, y_i \in R, N$ 为样本数. 要求拟合的函数为

$$f(\vec{x}_k) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(\vec{x}_k, \vec{x}_i) + b. \quad (1)$$

其中: 核函数 $K(\vec{x}_k, \vec{x}_i)$ 把输入向量映射到高维特征空间, α_i^* 和 α_i 为拉格朗日乘子, b 为阈值. 核函数采用径向基函数, 即

$$K(\vec{x}_k, \vec{x}_i) = \exp(- \| \vec{x}_k - \vec{x}_i \|^2 / 2\gamma^2).$$

通过引入两个松弛变量, 即不敏感损失函数 ϵ 和惩罚因子 C, 再根据拉格朗日函数的极值满足条件, 这样的回归问题可转化成下列二次优化问题:

$$\min_{\alpha, \alpha^*} w(\alpha, \alpha^*) = \min_{\alpha, \alpha^*} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\vec{x}_i, \vec{x}_j) - \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) y_i + \epsilon \sum_{i=1}^{N} (\alpha_i^* - \alpha_i). \quad (2)$$

约束条件为

$$\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0, \alpha_i, \alpha_i^* \in [0, C].$$

解上述优化问题可求得拉格朗日乘子, 拉格朗日乘子非零的训练样本点称为支持向量.

根据 Karush-Kuhn-Tucker (KKT) 条件[2], 可计算阈值

$$b = \frac{1}{2}\Big[y_k + y_j - \Big(\sum_{j=1}^{N} (\alpha_j - \alpha_j^*) k(\vec{x}_k, \vec{x}_j) - \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) k(\vec{x}_k, \vec{x}_j)\Big)\Big]. \quad (3)$$

式(3) 给出一种取平均阈值的方法. 也可从所有阈值中任取一个, 但这并不是最优解. 实际上, 大多数不为零的 Lagrange 乘子也并非最优解.

考虑到 RBF 神经网络对于输出层的权值通常采用最小二乘或梯度法进行优化调整, 如何优化更新支持向量机的权值和阈值, 使得函数值与实际值拟合得更好, 是一个值得探讨的问题. 为此, 本文针对多输入单输出的样本集, 提出一种两阶段的网络优化学习策略.

2.2 SVR 网络初始化阶段

在支持向量回归理论的指导下, 假设利用序贯最小优化算法[9] 找出不为零的 Lagrange 因子及其对应的 n 个支持向量, 令权值 $\beta_j = \alpha_i^* - \alpha_i, j = 1, 2, \cdots, n$. 为方便起见, 权值和阈值表示为 $[\beta_1, \cdots, \beta_n, \beta_{n+1}] = [\beta_1, \cdots, \beta_n, b]$, 核函数表示为 $K_j(\vec{x}_i) = K(\vec{x}_i, \vec{x}_k)$, 其中 $\vec{x}_k$ 是第 j 个支持向量. 重写式(1) 为

$$y_i = \sum_{j=1}^{n} \beta_j K_j(\vec{x}_i) + \beta_{n+1}. \quad (4)$$

令

$$\lambda(\vec{x}_i) = \sum_{j=1}^{n} K_j(\vec{x}_i), \quad (5)$$

$$G = \begin{bmatrix} K_1(\vec{x}_1) & \cdots & K_n(\vec{x}_n) \\ \vdots & & \vdots \\ K_1(\vec{x}_1) & \cdots & K_n(\vec{x}_n) \end{bmatrix}, \quad (6)$$

$$L = \mathrm{diag}[\lambda(\vec{x}_i)]. \quad (7)$$

则可变换成如下矩阵形式:

$$Y = G \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \beta_{n+1} \\ \vdots \\ \beta_{n+1} \end{bmatrix}, \quad (8)$$

其中 $Y = [y_1, y_2, \cdots, y_n]^{\mathrm{T}}$. 因为 $\sum_{j=1}^{n} K_j(\vec{x}_i) / \lambda(\vec{x}_i) = 1$, 所以

$$L^{-1} G \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (9)$$

则式(7) 可重写为

$$Y = G \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + L^{-1} G \begin{bmatrix} \beta_{n+1} \\ \vdots \\ \beta_{n+1} \end{bmatrix} = L^{-1} G \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix}, \quad (10)$$

$$G \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} = L G \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + G \begin{bmatrix} \beta_{n+1} \\ \vdots \\ \beta_{n+1} \end{bmatrix}. \quad (11)$$

重新构造式(10), 则实际的权值 $B_1, B_2, \cdots, B_n$ 为 $\beta_1, \beta_2, \cdots, \beta_{n+1}$ 的组合表达式, 即

$$\begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} = (G^{\mathrm{T}} G)^{-1} G^{\mathrm{T}} L G \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \beta_{n+1} \\ \vdots \\ \beta_{n+1} \end{bmatrix}. \quad (12)$$

由式(11) 可表示支持向量回归网络

$$y_i = \frac{\sum_{j=1}^{n} B_j K_i(\vec{x}_i)}{\sum_{j=1}^{n} K_i(\vec{x}_i)} = \sum_{j=1}^{n} B_j N_j(\vec{x}_i), \quad (13)$$

其中 $N_j(\vec{x}_i) = K_j(\vec{x}_i) / \lambda(\vec{x}_i)$ 为归一化基函数. 当核函数采用径向基核时, 式(12) 类似于加权平均的单输出 RBF 网络, 其核函数相当于径向基网络的隐层单元基函数, 而支持向量相当于隐层基函数的中心向量. 本文将式(12) 的描述形式称为支持向量回归

网络(SVRN), 其结构如图 1 所示

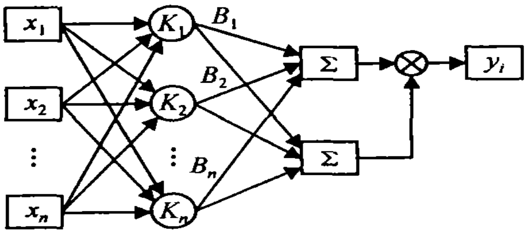

图 1　支持向量回归网络结构

2.3　SVR 网络学习阶段

这一阶段主要是寻找最优的权值$\{B_1, B_2, \cdots, B_n\}$. 将式(12) 改写成矩阵形式, 有

$$y_j = A^{\mathrm{T}}(\vec{x}_j)B. \tag{14}$$

其中

$$A(\vec{x}_j) = [N_1(\vec{x}_j), \cdots, N_n(\vec{x}_j)]^{\mathrm{T}},$$

$$B = [B_1, B_2, \cdots, B_n]^{\mathrm{T}}.$$

这里采用带有遗忘因子的递归最小二乘算法[10] 调整权值. 为使函数值接近于实际值, 评价函数定义为

$$\min E(B) = \sum_{j=1}^{n} \rho^{n-1}(y_j - f(\vec{x}_j)^2). \tag{15}$$

其中: 遗忘因子 $\rho \in (0, 1)$, 假定 j 为数据采样时刻, 则整个递推过程为

$$B_{j+1} = B_j + \eta_{j+1} e_{j+1}^{\mathrm{T}}. \tag{16}$$

其中

$$\eta_{j+1} = \frac{\rho^{-1} P_j A(\vec{x}_{j+1})}{1 + \rho^{-1} A^{\mathrm{T}}(\vec{x}_{j+1}) P_j A(\vec{x}_{j+1})},$$

$$e_{j+1} = y_{j+1} - f(\vec{x}_{j+1}),$$

$$P_{j+1} = \rho^{-1} P_j - \rho^{-1} \eta_{j+1} A(\vec{x}_{j+1}) P_j,$$

$$P_0 = I.$$

整个支持向量回归网络优化的学习策略的具体步骤如下:

1) 给定训练集$\{(\vec{x}_i, y_i), i = 1, 2, \cdots, N\}$, $\vec{x}_i \in R^m$, $y_i \in R$; 测试集$\{(\vec{x}_k, y_k), i = 1, 2, \cdots, N\}$, $\vec{x}_k \in R^m$, $y_k \in R$.

2) 选择适当的正数 ϵ和 C, 选择适当的核函数; 根据文献[9] 找出不为零的 Lagrange 因子及其对应的支持向量, 按式(13) 重新构造支持向量回归网络.

3) 利用带有遗忘因子的递归最小二乘算法, 根据式(16) 计算权值和阈值, 判断期望输出与实际输出的误差, 直到满足条件为止.

3　实例仿真实验

本文考虑由如下差分方程给出的非线性系统:

$$y_k = 0.5^2 y_{k-1} - 0.7 y_{k-2} + 1.2 u_{k-1} + 0.5 u_{k-2} + \eta_k.$$

其中: y_k 表示 k 时刻的输出, η_k 表示 k 时刻的加性噪声. 定义输入向量

$$\vec{x}_k = (y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}),$$

理想输出为标量y_k. 对系统加入伪随机信号, 得到一组 200 个输入输出的训练数据, 这组数据表示了离散时间序列的动态行为. 然后把样本集分成两部分: 60 个样本用来训练支持向量回归网络; 140 个样本用来测试训练过的支持向量回归网络的性能. 取容量控制因子 $C = 100$, 不敏感损失函数 $\epsilon = 0.5$, 核函数参数 $\gamma = 10$. 通过求解二次规划最优问题, 利用 60 个样本训练得到了 10 个支持向量. 于是支持向量回归函数为

$$y_k = \sum_{i=1}^{10} \beta_i \exp\left(-\frac{\|\vec{x}_k - \vec{x}_i\|^2}{2\gamma^2}\right) + b,$$

其中 $\vec{x}_i$ 是第 i 个支持向量. 则支持向量回归网络可表示为

$$y_k = \frac{\sum_{i=1}^{10} B_i \exp(-\|\vec{x}_k - \vec{x}_i\|^2)}{\sum_{i=1}^{10} \exp(-\|\vec{x}_k - \vec{x}_i\|^2)}.$$

权值B_i 可通过式(16) 计算得到.

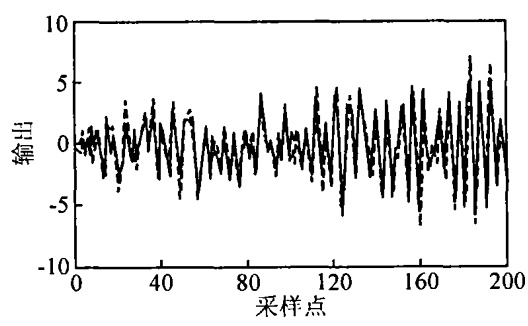

图 2　SVRN 输出与 SVR 输出的比较

描述训练集和测试集样本的拟合情况如图 2 所示. 图中点划线为 SVR 预测值, 直线为 SVRN 预测值. 用标准均方误差(NMSE) 作为评价准则, 定义如下:

$$\mathrm{NMSE} = \frac{1}{\delta^2 n} \sum_{i=1}^{n} (y_i - \hat{y})^2,$$

$$\delta^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2.$$

其中: $\hat{y}$ 表示预测值, $\bar{y}$ 表示实际值的均值.

SVR 输出和 SVRNN 输出及与实际值的平均 NMSE 如表 1 所示. 由表可见, 训练过的支持向量回归网络能够精确地描述系统的动态行为, 取得比 SVR 更好的预测结果.

表 1 两种方法的标准均方误差

方 法	NMSE
SVR	0.116
SVRN	0.086

4 结 论

本文在支持向量回归的基础上提出一种网络优化策略, 并通过此方法对非线性系统的建模问题进行研究. 该算法能够克服 SVR 学习过程中权值和阈值难以在线寻优的问题, 得出最优的权值和阈值, 提高 SVR 的精度. 仿真实验表明, 这种网络优化策略具有较高的预测精度, 可以推广用于非线性系统的控制等过程. 进一步的工作是将该方法推广到可以优化支持向量机的核参数.

参考文献(References)

[1] Vapnik V N. *统计学习理论的本质*[M]. 张学工译. 北京: 清华大学出版社, 2000: 96-107.
(Vapnik V N. *The Nature of Statistical Learning Theory*[M]. Berlin: Springer, 1995.)

[2] 邓乃扬, 田英杰. *数据挖掘中的新方法: 支持向量机*[M]. 北京: 科学出版社, 2004: 49-95.
(Deng N Y, Tian Y J. *The New Method of Data Mining: Support Vector Machines*[M]. Beijing: Science Press, 2004: 49-95.)

[3] Trafalis T B, Inco H. Support Vector Machine for Regression and Applications to Financial Forecasting[A]. *Proc of the IEEE INNS-ENNS Int Joint Conf on Neural Networks*[C]. Como, 2000, 6: 348-353.

[4] Tay F E H, Cao L. Application of Support Vector Machines in Financial Time Series Forecasting[J]. *Omega*, 2001, 29(4): 309-317.

[5] Gestel T V, Suykens J A, Baestaens D E, 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.

[6] Drezet P M L, Harrison R F. Support Vector Machines for System Identification[A]. *UKACC Int Conf on Control*[C]. Swansea, 1998: 688-692.

[7] Gretton A, Doucet A, Herbrich R, et al. Support Vector Regression for Black-box System Identification[A]. *Proc of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*[C]. Singapore, 2001: 341-344.

[8] Suykens J. Nonlinear Modeling and Support Vector Machines[A]. *IEEE Instrumentation and Measurement Technology Conf*[C]. Budapest, 2001: 287-294.

[9] Flake G W, Lawrence S. Efficient SVM Regression Training with SMO[J]. *Machine Learning*, 2002, 41(1): 271-290.

[10] 刘宏才. *系统辨识与参数估计*[M]. 北京: 冶金工业出版社, 1999: 74-88.
(Liu H C. *System Identification and Parameter Estimate*[M]. Beijing: Metallurgy Industry Press, 1999: 74-88.)

下 期 要 目