

文章编号:1001-9081(2010)03-0776-03

基于 AdaBoost 算法的加权二乘向量回归机

彭代强, 林幼权

(南京电子技术研究所, 南京 210039)

(pengdaiq@gmail.com; peng.d.q@126.com)

摘要:针对二乘向量机(LS-SVM)对所有样本误差惩罚相同、预测精度不高的问题,提出了一种基于 AdaBoost 模型的二乘向量回归机。该算法使用多个二乘向量机按照某种学习规则协调各二乘向量机的输出,同时根据回归精度,建立各二乘向量机中每一个样本的误差惩罚权重,以突出样本的惩罚差异性,提高算法的泛化性能。实验结果表明,提出的算法提高了二乘向量回归机的预测精度,优化了学习机的性能。

关键词:AdaBoost 算法;二乘向量机;回归

中图分类号:TP391 **文献标志码:**A

Weighted least squares support vector regression based on AdaBoost algorithm

PENG Dai-qiang, LIN You-quan

(Nanjing Institute of Electronic Technology, Nanjing Jiangsu 210039, China)

Abstract: In the standard Least Squares Support Vector Machine (LS-SVM) for regression, every training sample is equally considered, which is unsuitable when there exists significant difference among the training samples. The weighted least squares support vector regression based on AdaBoost algorithm was proposed. Learning by a series of support vector regressions, the proposed approach combined all the results in accordance with some rule. At the same time, adaptive weighted factors in LS-SVM were constructed to control the error function according to the regression error. It emphasized the significant difference among the training samples by adaptive weighted factors and improved the performance of generalization error. The experimental results demonstrate that the proposed approach has a competitive learning ability and acquires better accuracy than LS-SVM.

Key words: AdaBoost algorithm; Least Squares Support Vector Machine (LS-SVM); regression

0 引言

支持向量机(Support Vector Machine, SVM)是 Vapnik 等人^[1]根据统计学习理论提出的一种学习方法,近年来在模式识别、回归分析和特征提取等方面得到了很多应用。该方法根据结构风险最小化原则,尽量提高学习机的泛化能力,即由有限训练样本得到的决策规则对独立的测试集仍能够得到小的误差,是兼顾到经验风险和置信范围的一种折中的思想,为解决不等式约束的二次优化问题, Suykenst 等人^[2]在 SVM 的优化函数中引入误差二次项,并将 SVM 中的不等式约束条件改为等式约束,提出了一种以二次等式约束条件为基础的最小二乘向量机(Least Squares Support Vector Machine, LS-SVM),LS-SVM 将 QP 问题有效地转化为一个线性 KKT (Karush-Kuhn-Tucker)系统问题,使得 LS-SVM 的计算复杂度和内存需求比标准 SVM 低得多。

然而在 LS-SVM 模型中,当对所有样本误差惩罚相同时,噪声对支持向量的估计影响较大,数据的稀疏特性会丢失^[3]。为了进一步提高二乘向量机的泛化精度,本文将 AdaBoost 模型框架引入到 LS-SVM 算法中,依次训练一组二乘向量机,在循环迭代中,根据弱学习机的反馈,自适应调整各二乘向量机中每个样本的误差惩罚权重,使得本次训练误

差较大的样本在下次训练中得到更多关注,而不是对训练数据中的所有样本惩罚程度一致,以突出样本的惩罚差异性,提高算法的鲁棒性,最终的回归结果则是根据这些二乘向量机的结果共同决定,这将有效地增强各二乘向量机的互补性,将“弱”二乘向量机经过算法提升后得到“强”二乘向量机,提高算法的泛化性能。

1 二乘向量回归机及 AdaBoost 算法

1.1 二乘向量回归机

对于一组给定的训练样本 $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ 和函数 $f: \mathbf{X} \rightarrow \mathbf{Y}$,其中 N 为样本数, $\mathbf{x}_i \in \mathbf{R}^d$ 为输入数据且 $\mathbf{x}_i \in \mathbf{X}$, $y_i \in \mathbf{R}$ 为输出数据且 $y_i \in \mathbf{Y}$,LS-SVM 目标函数为^[3]:

$$\min J = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\gamma}{2} \sum_{i=1}^N e_i^2$$

$$\text{s. t. } y_i = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b + e_i \quad (1)$$

其中: $\boldsymbol{\varphi}(\cdot)$ 为非线性变换函数,它将输入空间非线性变换到高维空间; e_i 为松弛项; γ 为错误惩罚因子,它是常数。相应的拉格朗日函数为:

$$L = J - \sum_{i=1}^N \alpha_i \{ \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b + e_i - y_i \} \quad (2)$$

其中 α_i 为拉格朗日乘子。对于式(2),根据 KKT 条件消去 \mathbf{w} 和

收稿日期:2009-09-09;修回日期:2009-10-27。

作者简介:彭代强(1975-),男,四川苍溪人,博士研究生,主要研究方向:模式识别、图像处理;林幼权(1966-),男,浙江余姚人,研究员,博士研究生,主要研究方向:信号处理、雷达系统设计。

e 得到如下线性方程:

$$\begin{bmatrix} 0 & 1^T \\ 1 & K + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \quad (3)$$

其中: $Y = (y_1; \dots; y_N)$, $1^T = (1, \dots, 1)$, $\alpha = (\alpha_1; \dots; \alpha_N)$, K 为核函数 $K(x, x_i) = \varphi(x)\varphi(x_i)$, $i = 1, 2, \dots, N$, 这样基于支持向量的函数估计形式为:

$$f(x) = \sum_{i=1}^N \alpha_i K(x, x_i) + b \quad (4)$$

α 和 b 是式(3)的解, 具体形式取决于错误惩罚因子 γ 及核函数的类型:

$$\begin{cases} \alpha = \Omega^{-1}(Y - 1b) \\ b = \frac{1^T \Omega^{-1} Y}{1^T \Omega^{-1} 1} \end{cases} \quad (5)$$

其中 $\Omega = K(x_i, x_j) + \gamma^{-1}I$.

1.2 AdaBoost 算法

Boosting 算法由 Schapire 提出^[4], 其原理是在一个线性决策规则中组合几个弱学习算法, 这样组合的线性决策规则比任何一个弱学习算法的性能都好很多. AdaBoost 算法是一种自适应 Boosting 算法, 它是 Boosting 家族中最具代表性的算法, 其算法如下.

给定 N 个训练样本: $(x_1, y_1), \dots, (x_N, y_N)$, 其中: $x_i \in X$, $y_i \in Y = \{-1, +1\}$.

1) 对于 $t = 1, \dots, T$:

- ① 由已知样本训练弱分类器, 使得 $f_t: X \rightarrow \{-1, +1\}$;
- ② 根据一定规则调整训练序列.

2) 最终分类结果 $H(x)$ 由所有弱分类器的组合函数 h 求取 $H(x) = h(f(x))$.

可以看出, 算法的关键是如何调整训练序列, 并组合各弱分类器的输出. Freund 等人^[5]有效解决了实际应用中的这些问题, 提出了 AdaBoost.M1 算法. 在 AdaBoost.M1 算法中, 每个训练样本都被赋予一个权重, 表明它被某个分量分类器选入训练集的概率. 如果某个样本点已经被准确地分类, 那么在构造下一个训练集中, 它被选中的概率就被降低; 相反, 如果某个样本点没有被正确分类, 那么它的权重就得到提高. 单一分类器可能会出现某些疑难的样本分类效果不好, 通过这样的方式, AdaBoost 方法能够聚焦于那些较困难(更富信息)的样本上, 并通过一个加权的多数表决来合并全部弱分类器, 以产生较强的分类机器系统^[6]. 最初 AdaBoost 算法只能解决两类问题, 后来 AdaBoost 算法被扩展为多类问题算法, 同时还被广泛用于回归估计中^[6], 该算法不需要任何参数设置, 拥有较好的性能, 因而被使用广泛.

2 基于 AdaBoost 的加权二乘向量回归机

由式(1)可以看出, 标准 LS-SVM 回归机对训练数据中所有样本的松弛项 e_i 惩罚程度一致, 为突出样本间的差异性, 对每个样本定义加权惩罚因子 v_i , 于是, 修改后的 LS-SVM 目标函数为:

$$\min J^* = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^N v_i e_i^2 \quad (6)$$

s. t. $y_i = w^T \varphi(x_i) + b + e_i$

相应的拉格朗日函数为:

$$L = J^* - \sum_{i=1}^N \alpha_i \{w^T \varphi(x_i) + b + e_i - y_i\} \quad (7)$$

式(7)消去 w 和 e 得到如下线性方程:

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & K + V_\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \quad (8)$$

其中 V_γ 为对角矩阵, 有 $V_\gamma = \text{diag}\{\frac{1}{\gamma v_1}, \dots, \frac{1}{\gamma v_N}\}$, v_i 为第 i 个样本所对应的权值, 控制着松弛项 e_i 的惩罚力度. 如何自适应地调整参数 v_i 是一个需要解决的问题, 该值应与松弛项 e_i 有关, e_i 较大的样本对应的权重 v_i 应较大, e_i 较小的样本对应的权重 v_i 应较小, 本文将 AdaBoost 算法的参数调整策略引入到 LS-SVM 中, 设置 v_i 的初始值为 $1/N$, 该值的更新可与前一次的训练松弛项 e_i 有关, 由拉格朗日函数式(7)可得 $e_i(i)$:

$$e_i(i) = \frac{\alpha_i(i)}{\gamma v_i(i)}; \quad t = 1, \dots, T \quad (9)$$

定义训练样本 x_i 对应的损失函数 $L_i(i)$ 为:

$$L_i(i) = \frac{\text{abs}(e_i(i))}{\text{enom}_i}; \quad \text{enom}_i = \max_{i=1,2,\dots,N} (\text{abs}(e_i(i))), \quad t = 1, \dots, T \quad (10)$$

于是回归机第 t 次的平均损失函数 \bar{L}_t 为:

$$\bar{L}_t = \sum_{i=1}^N L_i(i) v_i(i) \quad (11)$$

将回归机的权重系数 β_t 取为平均损失函数的函数:

$$\beta_t = \frac{\bar{L}_t}{(1 - \bar{L}_t)} \quad (12)$$

当前惩罚权重 $\gamma_{t+1}(i)$ 更新为:

$$\gamma_{t+1}(i) = \frac{\gamma_t(i) \beta_t^{(1-L_i^*(i))}}{Z_t} \quad (13)$$

其中: $L_i^*(i) = \frac{\text{abs}(e_{\text{mean}}(i))}{\text{enom}_{\text{mean}}}$, $\text{enom}_{\text{mean}} = \max_{i=1,2,\dots,N} (\text{abs}(e_{\text{mean}}(i)))$, $t = 1, \dots, T$, $e_{\text{mean}}(i)$ 为第 t 次训练松弛项 $e_i(i)$ 在窗口内的局部均值, Z_t 为归一化因子, $Z_t = \sum_{i=1}^N \gamma_{t+1}(i)$, 最终总体回归输出可以使用各个分量回归机加权平均来得到:

$$f(i) = \frac{\sum_t \log\left(\frac{1}{\beta_t}\right) f_t(i)}{\sum_t \log\left(\frac{1}{\beta_t}\right)} \quad (14)$$

值得注意的是惩罚权重 $\gamma_t(i)$ 在第 t 循环中的相对大小主要与 $L_i^*(i)$ 有关, 它是损失函数 $L_i(i)$ 的一种局部平均, 这样可以有效地避免噪声对惩罚权重的影响; 同时权重系数 β_t 是对当前回归机 $f_t(i)$ 可信度的一种度量, 它与平均损失函数 \bar{L}_t 有关, 当前回归误差较大时, $f_t(i)$ 在 $f(i)$ 中所占比例将相对较小, 可有效地增强回归机的推广能力.

3 实验与结果分析

3.1 一维函数仿真

为验证算法的有效性, 采用函数 $y_i = g_i + n_i$, 其中 g_i 为未被噪声污染的真实信号, $g_i = \frac{\sin(\pi t)}{\pi t}$, $t \in [-3, 3]$, t 在输入区间的步长为 0.005, $n \sim N(0, \sigma^2)$ 为噪声, y_i 为观察信号, 估计信号为 y_i^e , 定义预测误差 (Prediction Error, PE) 和模型误

差 (Model Error, ME):

$$PE = \frac{1}{N} \sum_{i=1}^N (y_i^e - y_i)^2$$

(15)

$$ME = \frac{1}{N} \sum_{i=1}^N (y_i^e - g_i)^2$$

(16)

在实验中,选用径向基函数(Radial Basis Function, RBF)作为核函数,图 1 为标准 LS-SVM 与本文算法的仿真结果对比图,图 2 为估计信号与观察信号的差值比较图,表 1 为预测误差和模型误差计算结果。从以上数据可以看出,本文得出的误差分布更趋近于高斯噪声,能较好地描述信号模型误差小,具有好的泛化和推广性能。

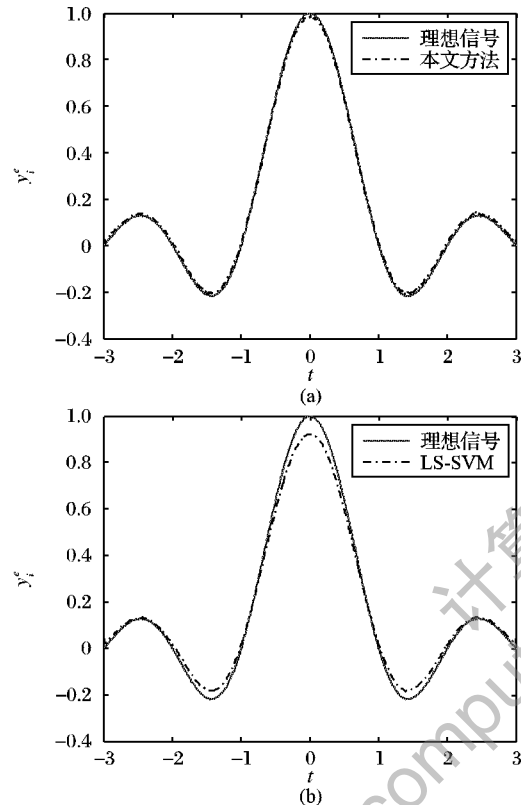


图 1 sinc 函数回归结果

3.2 Boston Housing 数据预测

Boston Housing 数据由 Boston 区域中线房价构成,可从 UCI 数据库中获取,该数据有 506 个样本,每个样本的特征变量维数为 14,其中 13 个特征量为输入变量,这些特征是影响房价的重要因素,主要包括该区域的犯罪率、零售业情况、与五大工作区的加权距离、高速路交通易达状况等,最后一个特征量为输出变量即 Boston 区域房价。由于该数据的理想值无法获取,只能计算预测误差,为消除不同特征量取值范围不同对房价估计的影响,首先对输入特征变量进行归一化处理,同时随机选取其中的 406 个数据进行训练,用剩余的 100 点数据进行预测,以检查模型的预测误差。

表 1 预测误差和模型误差的计算结果

实验方法	误差模型	
	PE	ME
本文方法	0.048	0.012
LS-SVM	0.136	0.065

表 2 为本文方法及标准二乘向量机的回归结果,经过 25 次循环,本文方法在训练样本上的预测误差为 0.005 2,标准

二乘向量机的预测误差为 0.005 0,两者有相似的回归精度;将训练后的模型用于测试样本,本文方法的预测误差为 0.023 4,标准二乘向量机的预测误差为 0.045 6。可以看出,在测试样本上本文方法比传统的 LS-SVM 方法提高了 0.022 2,这说明本文方法对于二乘向量机的建模与预测更加有效。

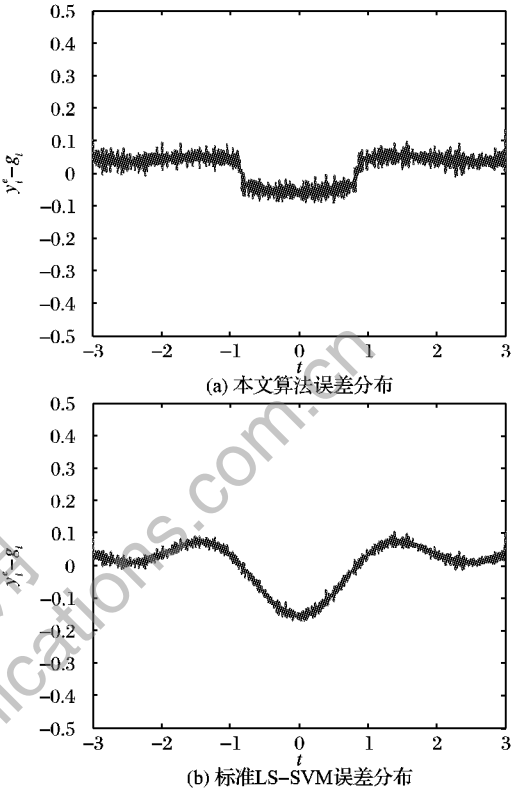


图 2 回归误差分布

表 2 Boston Housing 数据预测误差结果

实验方法	预测误差	
	训练样本	测试样本
本文方法	0.005 2	0.023 4
LS-SVM	0.005 0	0.045 6

4 结语

本文提出了基于 AdaBoost 的二乘向量机集成算法,该算法能有效地利用 AdaBoost 决策规则决定二乘向量机误差惩罚因子,将多个弱回归机组合成一个强回归机。实验结果表明该方法具有更好的泛化和推广性能。

参考文献:

[1] VAPNIK V. An overview of statistical learning theory [J]. IEEE Transactions on Neural Networks, 1999, 10(5): 988-999.

[2] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999, 9(3): 293-300.

[3] SUYKENS J A K, DEBRABANTER J, LUKAS L, et al. Weighted least squares support vector machines: Robustness and sparse approximation [J]. Neurocomputing, 2002, 48(1/4): 85-105.

[4] SCHAPIRE R. The strength of weak learnability [J]. Machine Learning, 1990, 5(2): 197-227.

[5] FREUND Y, SCHAPIRE R E. A decision theoretic generalization of on-line learning and an application to boosting [J]. Journal of Computer and System Science, 1997, 55(1): 119-139.

[6] 解洪胜,张虹.基于内容的图像检索中 SVM 和 Boosting 方法集成应用[J].计算机应用,2009,29(4): 979-981.