

Keerthi 的 SMO 算法的偏置计算改进

陈凯亚, 王敏锡

(西南交通大学电磁场与微波技术研究所, 成都 610031)

摘要: 指出 Keerthi 的 SMO 算法存在的问题。该算法由于采用“取中法”求偏置, 在优化条件不满足的情况下, 偏置值有可能出现偏差, 从而劣化 SVM 的建模性能。该文从 SVM 回归的原问题出发, 导出求偏置的新方法并将其归结为一维凸函数最优化问题, 将新算法应用于高斯函数的回归和记忆非线性功率放大器的预失真器的建模中, 结果显示了新算法的正确性和有效性, 建模精度提高 10% 左右。

关键词: 支持向量机; SMO 算法; 回归

Improvement of Bias Computing on Keerthi's SMO Algorithm

CHEN Kai-ya, WANG Min-xi

(Institute of Electromagnetic Fields and Microwave Technology, Southwest Jiaotong University, Chendu 610031)

【Abstract】 A source of error in Keerthi's Sequential Minimal Optimization(SMO) regression algorithm is pointed out. The bias value is acquired by 'median-finding' technique in this algorithm, when the optimality condition does not hold, the bias value may be inaccurate, and make the Support Vector Machine(SVM) model performance worse. By analyzing the primal problem of SVM regression, a new kind of computing method is derived to get bias, and proved to be an optimization problem of one dimension convex function. The modified algorithm is confirmed to be correct and effective when it is used to estimate Gaussian function and model predistorter of nonlinear memory power amplifier, the SVM model trained by the new algorithm performs more accurately by 10% than Keerthi's one does.

【Key words】 Support Vector Machine(SVM); SMO algorithm; regression

1 概述

SVM^[1]采用结构风险最小化原则, 避免了神经网络单纯追求最小化经验风险而导致的过拟合问题, 具有很强的推广能力。其目标函数可转化为一个标准的二次规划(QP)问题, 避免了局部极值。其结构复杂度通过求解该QP问题得到的支持向量个数自动确定, 不需先验确定结构。因此, SVM广泛地应用于模式识别和函数回归领域。采用传统方法求解SVM的QP问题需要数据以核矩阵的形式存储在内存中, 矩阵规模上升需占用大量的计算机内存。分解算法^[2]则将问题分解为一系列规模较小的QP问题, 在每次迭代中只更新a的一个分量子集。SMO算法^[3]将分解算法的思想推向极致, 每次迭代仅优化2个点的最小子集, 由于2个数据点的优化问题可以获得解析解, 因此避免将QP问题作为算法的一部分。Smola^[4]用类比的方法将SMO算法推广到回归问题。

文献[5]指出单阈值对算法速度的不利影响, 提出基于上下阈值概念的改进SMO回归算法。该算法的偏置b是将上下阈值(b_{up} 和 b_{lo})取中得到的, 当优化条件满足时, 由于 $b_{lo} - b_{up} > 2r$, 因此通过取中得到的b值较为精确。但若优化条件不满足, 且算法由于C值的限制而无法进一步优化时, “取中法”得到的b值难以保证获得最优值, 从而直接恶化了SVM模型的性能。

本文从支持向量机回归的原问题出发, 将最优b值的求解问题转化为一维凸函数的最优化问题, 通过一维搜索算法即可求解。

2 支持向量机回归问题概述

对于非线性回归问题, SVM首先通过一个非线性映射 $\varphi(\cdot)$ 把训练数据映射到一个高维特征空间, 然后在这个空间中建

立一个线性回归函数, 即

$$f(x) = w^T \varphi(x) + b \quad (1)$$

引进下述结构风险函数:

$$R = \frac{1}{2} \|w(a, a^*)\|^2 + CR_{emp}(f) \quad (2)$$

其中, C为常数, 其作用是在经验风险(R_{emp})和模型复杂度($\|w\|^2$)之间取一折中, 由于a和 a^* 在 $[0, C]$ 区间内取值, 因此不同的C值确定了不同的 $\|w\|^2$ 。 R_{emp} 又称为损失函数, 可以有多种选择。其中, ε -不敏感损失函数最为常用, 它不惩罚偏差小于 ε 的误差项。则式(2)在 ε -不敏感损失函数下为

$$R = \frac{1}{2} \|w(a, a^*)\|^2 + C \sum_i (\xi_i + \xi_i^*) \quad (3)$$

式(3)被称为原问题, 通过对式(3)求最小, 可得到拉格朗日乘子a, a^* 和偏置b。从而得到回归函数:

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

其中, $K(\cdot)$ 为核函数, 反映了向量在特征空间的内积。

3 Keerthi 的 SMO 算法偏置计算中存在的问题

该算法在进行优化的过程中, 将数据点归结为5类: $I_{0a} = \{i: 0 < a_i < C\}$; $I_{0b} = \{i: 0 < a_i^* < C\}$; $I_1 = \{i: a_i = 0, a_i^* = 0\}$; $I_2 = \{i: a_i = 0, a_i^* = C\}$; $I_3 = \{i: a_i = C, a_i^* = 0\}$ 。并定义了如下2个变量:

$$\tilde{F}_i = \begin{cases} F_i + \varepsilon & \text{if } i \in I_{0b} \cup I_2 \\ F_i - \varepsilon & \text{if } i \in I_{0a} \cup I_1 \end{cases} \quad (5)$$

基金项目: 国家自然科学基金资助项目(60272003)

作者简介: 陈凯亚(1971-), 男, 博士研究生, 主研方向: 微波电路, 信号处理, 机器学习; 王敏锡, 教授

收稿日期: 2007-06-21 **E-mail:** cky_cy@sohu.com

$$\bar{F}_i = \begin{cases} F_i - \varepsilon & \text{if } i \in I_{0a} \cup I_3 \\ F_i + \varepsilon & \text{if } i \in I_{0b} \cup I_1 \end{cases} \quad (6)$$

其中, $F_i = d_i - w(a, a^*) \cdot \phi(x_i)$ 。从式(5)和式(6) 2个变量可得到上下阈值:

$$b_{up} = \min \{ \bar{F}_i : i \in I_0 \cup I_1 \cup I_3 \} \quad (7)$$

$$b_{lo} = \max \{ \bar{F}_i : i \in I_0 \cup I_1 \cup I_2 \} \quad (8)$$

其中, $I_0 = I_{0a} \cup I_{0b}$ 。当 $b_{lo} - b_{up} \geq 2r$ 成立时, 则优化条件满足。

该算法的优化过程可以分为2步:

步骤1 遍历属于 $I(I=I_0 \cup I_1 \cup I_2 \cup I_3)$ 的数据集, 进行优化, 若 $numberchanged=0$ (表示数据集中没有数据优化成功), 则结束优化; 若 $numberchanged > 0$ (表示数据集中有数据优化成功), 则转到步骤2。

步骤2 反复遍历属于 I_0 的数据集, 进行优化, 直到 $numberchanged=0$ 为止, 并转到步骤1。

分析以上步骤可知, 只要在步骤1中 $numberchanged=0$, 算法结束。而此时可能有2种结果, 对于数据集 I , 优化条件成立 (即 $b_{lo} - b_{up} \geq 2r$) 或不成立 (即 $b_{lo} - b_{up} < 2r$)。前者可以通过原文中的“取中法”求得偏置 b 。后者则应采用新的方法, 否则得到的偏置并不能保证最优。

造成算法结束而优化条件不成立的原因在于, 决定回归函数结构的拉格朗日乘子 a 和 a^* 受限于 C 构成的盒约束, C 值越大, a 和 a^* 的可行解集合越大, 优化算法所付出的时间代价则越大, 而模型的性能改善可能并不显著。因此针对实际的回归问题, 往往选择能满足性能要求的 C 值, 从而导致算法结束而优化条件可能并不满足的情况。针对这种情况, 有必要找到一种新的求解偏置的方法, 弱化优化条件需成立的要求。使得优化条件仅仅用于优化过程, 而不用用于指导求解偏置。

4 偏置计算的改进

从原问题出发, 对于特定的 C 值, 当 SMO 算法结束时, 拉格朗日乘子 a 和 a^* 已确定, 式(3)中的 w^2 也随之确定, 为使式(3)最小, 应最小化下式:

$$R = \sum_i (\xi_i + \xi'_i) \quad (9)$$

其中, ξ_i 和 ξ'_i 为预测值同期望值的绝对偏差大于 ε 的误差值。其意义为

$$\xi_i = \begin{cases} (\omega \cdot \phi(x_i) + b) - d_i - \varepsilon & \text{if } (\omega \cdot \phi(x_i) + b) - d_i > \varepsilon \\ 0 & \text{if } (\omega \cdot \phi(x_i) + b) - d_i \leq \varepsilon \end{cases} \quad (10)$$

$$\xi'_i = \begin{cases} d_i - (\omega \cdot \phi(x_i) + b) - \varepsilon & \text{if } (\omega \cdot \phi(x_i) + b) - d_i < -\varepsilon \\ 0 & \text{if } (\omega \cdot \phi(x_i) + b) - d_i \geq -\varepsilon \end{cases} \quad (11)$$

为清楚地了解式(9)的特点, 可将式(9)分解为

$$R = R_1 + R_2 = \sum_i \xi_i + \sum_i \xi'_i \quad (12)$$

其中, R_1 和 R_2 分别对应于式(12)中右式的第1、第2项, R_1 和 R_2 均为偏置 b 的函数。随着 b 的取值由小到大变化, 由式(10)和式(11)可知, R_1 和 R_2 的变化可分为3个阶段。第1个阶段: R_2 递减, R_1 恒为零; 第2个阶段: R_2 递减, R_1 递增; 第3个阶段: R_2 恒为零, R_1 递增。显然, 在第1个阶段, 当 b 取值最大 (假定为 m) 时, R 取得最小值。在第3个阶段, 当 b 取值最小 (假定为 n) 时, R 取得最小值, 因此应在第2个阶段内以及第1、第3阶段的 m 和 n 点上讨论该优化问题, 即 $b \in [m, n]$ 。当 b 在区间 $[m, n]$ 内取值时, R_2 为递减的凸函数, 这是因为当 b 以 Δb 等间隔递增时, 不仅 R_2 递减而且 R_2 的减少量也递减, 当 Δb 趋于零时, R_2 的一阶和二阶导数均为负。同理可知, R_1 为递增的

凸函数。由于2个凸函数的和仍为凸函数, 因此 R 存在全局最小点。采用一维搜索算法即可求得最优的偏置 b_{opt} 。

5 仿真实例

本节采用2个仿真算例来验证提出的偏置计算方法的正确性和有效性。求解 b_{opt} 的一维搜索算法采用黄金分割算法。第1个算例为高斯函数的回归问题, 取 $C=1$ 时, Keerthi 的 SMO 算法能得到满足优化条件的阈值 ($b_{lo}=0.0298, b_{up}=0.0296, b_{lo}-b_{up} \geq 2r, r=1e-4$), 从而得到偏置 $b=0.0297$, 而采用新算法搜索到的偏置也为 0.0297 。可见, 新算法在这种情况下同 Keerthi 的 SMO 算法是一致的。第2个算例为记忆非线性功率放大器的预失真器建模。

图1中预失真器构造的特性同放大器相反, 以补偿放大器的记忆非线性, 使得输出 $y(n)$ 为输入 $x(n)$ 的线性放大。

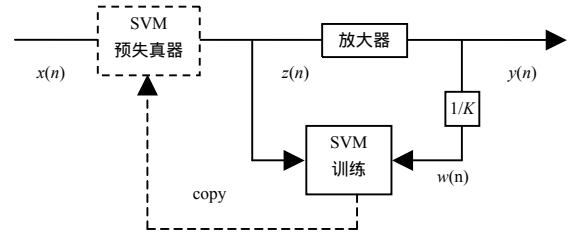


图1 预失真器的非直接学习结构

利用 SVM 对预失真器进行建模。令 $x(n) = I_{in}(n) + jQ_{in}(n)$, $z(n) = I_{out}(n) + jQ_{out}(n)$, 则预失真器的函数关系为

$$I_{out}(n) = f_I[I_{in}(n), I_{in}(n-1), \dots, I_{in}(n-M), Q_{in}(n), Q_{in}(n-1), \dots, Q_{in}(n-M)] \quad (13)$$

$$Q_{out}(n) = f_Q[I_{in}(n), I_{in}(n-1), \dots, I_{in}(n-M), Q_{in}(n), Q_{in}(n-1), \dots, Q_{in}(n-M)] \quad (14)$$

其中, M 为记忆长度。在训练过程中, 预失真器禁用, 此时 $z(n) = x(n)$, 将 $[w(n), z(n)]$ 作为输入输出训练对, 通过式(15)和式(16)训练 SVM, 以实现 f_I 和 f_Q 进行建模。令 $w(n) = I(n) + jQ(n)$, $z(n) = I_{desire}(n) + jQ_{desire}(n)$, 则有

$$I_{desire}(n) = f_I[I(n), I(n-1), \dots, I(n-M), Q(n), Q(n-1), \dots, Q(n-M)] \quad (15)$$

$$Q_{desire}(n) = f_Q[I(n), I(n-1), \dots, I(n-M), Q(n), Q(n-1), \dots, Q(n-M)] \quad (16)$$

本文仿真平台为 Matlab, 采用 16-QAM 调制信号, 并经升余弦滤波器进行整形。对整形后的信号进行采样得到 3500 个样本, 前 500 个样本用于训练, 后 3000 个样本用于测试, 仿真中随机产生了 10 组样本。通过仿真比较了 Keerthi 的 SMO 算法同新算法的建模性能。该性能由 EVM 值来反映, 值越小, 则性能越好。

$$EVM = \sqrt{\frac{\sum [(I_d - I)^2 + (Q_d - Q)^2]}{\sum (I_d^2 + Q_d^2)}} \quad (17)$$

其中, I_d 和 Q_d 为串联了 SVM 预失真器的放大器的期望输出值; I 和 Q 为实际输出值。

当 $C=1$ 时, 由于 Keerthi 的 SMO 算法得到的上下阈值难以满足优化条件, 由表 1 中可见, I 信道和 Q 信道的 Δb (即 $b_{lo} - b_{up}$) 均大于 $2r$, 故直接通过“取中法”得到的偏置常常偏离了最优值, 使得 SVM 建模的性能变差, 在本例中表现为 EVM 性能指标偏大。在样本 1、样本 6、样本 7 和样本 10 的

(下转第 12 页)