

从近似超平面到 SVR 的算法研究^{*}

曾绍华^{1,2} 魏 延^{1,3} 曹长修¹

(重庆大学自动化学院 重庆 400030)¹ (重庆师范大学管理学院 重庆 400030)²

(重庆师范大学数学与计算机科学学院 重庆 400030)³

摘 要 本文证明了 SVM 存在近似超平面;根据 SV 分布于 SVM 超平面附近,也必然分布于其近似超平面附近的特点,提出了从近似超平面出发,通过向量距近似超平面的距离的大小逐步搜索 SV,建立 SVR 的算法思想;列举了基于该算法思想的一个算法实例——从多元回归平面构建 LS-SVM;分析了其时空复杂度,并与 LS-SVM 的线性方程组解法和直接分解算法进行比较,其结果是该算法能够收敛到 l 个训练样本直接建立的 SVR,并降低了计算时间复杂度和显著降低了计算空间复杂度。

关键词 支持向量回归机,算法,复杂度分析

Research on Algorithms of Constructing SVR Based on its Approximate Hyperplane

ZENG Shao-Hua^{1,2} WEI Yan^{1,3} CAO Chang-Xiu¹

(College of Automation, Chongqing University, Chongqing 400030)¹ (Management of College, Chongqing Normal

University, Chongqing 400030)² (College of Mathematics & Computer Science, Chongqing Normal University, Chongqing 400030)³

Abstract This paper contributes a class of algorithms of constructing SVR based on its different approximate hyperplanes. It proves that SVM has its approximate hyperplanes. According as there are Support Vectors near the SVM and the Support Vectors are consequentially near the approximate hyperplane of the SVM too, to bring forward the idea of constructing SVR—starting from its approximate hyperplane to search the Support Vectors step by step. It represents an algorithm instance—starting from the Multiple Linear Regression Model to constructing LS-SVM, based on the idea, and analyzes its complexity. Comparing between it, the method of solving LS-SVM with system of linear equations and the decomposition algorithm, the result is the algorithm can converge to LS-SVM and decrease the time complexity and reduce evidently the space complexity.

Keywords SVR(Support Vector Regression), Algorithm, Complexity analysis

支持向量机(SVM——Support Vector Machine)是 1992 年 Vapnic 等人发明的^[2],常用来解决分类和回归问题,是解决非线性问题的重要工具。SVM 是凸二次规划问题,大样本条件下其核矩阵非常庞大,解凸二次规划算法的时间复杂度与样本有关^[5],求解困难,因此,快速求解 SVM 成为了广大学者研究的课题。前人在 SVM 上做了大量的研究,对支持向量分类机(SVC——Support Vector Classifier)研究比较成熟,并在分类的基础上提出了用 SVM 解决回归问题^[1,3]。但是,许多能解决分类问题的 SVM 快速算法都不能直接用于解决回归问题,支持向量回归机(SVR——Support Vector Regression)的快速算法还有待进一步研究。

由于 SVM 训练结果只与支持向量(SV——Support Vectors)有关,因此,人们努力去搜寻 SV,以降低解凸二次规划的复杂度。其中一条途径就是分块或分解算法,例如文[1]。C. Bradford 在文[3]中提出了解凸壳的方法,求包含 SV 的凸壳。但是,现有的分块或分解算法求取 SV 的目标性不强,对解凸二次规划的复杂度降低较小;C. Bradford 提出的解凸壳方法运算复杂,不能解决回归问题。希望有一种比文[1]目标性强,比解凸壳简单,适合搜寻回归问题 SV 的简单方法,以降低求解 SVR 的复杂度,建立快速 SVR。

1 算法思想

由于 SVM 训练结果只与支持向量有关,因此求解 SVM

时间复杂度最小的方法就是直接用 SV 去建立 SVM 的超平面。但是,未建立超平面无法预知哪些样本(向量)是 SV,因此,不可能先求得 SV,然后再求取 SVM。

如果有 SVM 的近似超平面,根据支持向量在 SVM 超平面附近的特点,就可以通过该近似超平面搜索 SV。

1.1 SVM 的近似超平面存在定理及证明

定义 1 设 $n+1$ 维向量空间中 Y 与 $X(Y \in R, X \in R^n)$ 之间满足函数关系: $Y = \varphi(X)$, 如果 $f(X)$ 能依概率逼近 $Y = \varphi(X)$, 则称 $f(X)$ 与 $\varphi(X)$ 近似, 记作 $f(X) \sim \varphi(X)$ 。

即: $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} \varphi(X)$, 则称 $f(X) \sim \varphi(X)$ 。 l 为训练样本数。

引理 1 设 $n+1$ 维向量空间中 Y 与 $X(Y \in R, X \in R^n)$ 之间满足函数关系: $Y = \varphi(X)$, 如果 $f(X) \sim \varphi(X)$, 则 $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} \varphi(X)$ 。

证明: 根据定义 1, 显然它是成立的。

引理 2 设 $n+1$ 维向量空间中可用 $f(X)$, $\varphi(X)$ 逼近 Y ($Y \in R, X \in R^n$), 则 $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} \varphi(X)$ 和 $\lim_{l \rightarrow \infty} \varphi(X) \xrightarrow{p} f(X)$ 。

证明: $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} Y$, Y 未知时, 用它的逼近函数 $\varphi(X)$

代替, 所以 $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} \varphi(X)$;

^{*} 基金项目: 重庆市教育委员会科学技术研究基金资助项目(KJ050809, KJ060801)。曾绍华 博士研究生, 主要研究方向: 智能计算与知识发现(KDD); 魏 延 博士研究生, 主要研究方向: 智能计算与知识发现(KDD); 曹长修 教授, 博士生导师, 主要研究方向: 智能计算与知识发现(KDD), 计算机网络及现代通信技术。

同理, $\lim_{l \rightarrow \infty} \varphi(X) \xrightarrow{p} f(X)$ 。

定理 1 设 $n+1$ 维向量空间中存在关于 Y 与 $X(Y \in R, X \in R^n)$ 的 SVM, 有可用来逼近 Y 的函数 $f(X)$, 则 $f(X) \sim \text{SVM}$ 。

证明: $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} Y, Y = \text{SVM}$, 所以 $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} \text{SVM}$, 则 $f(X) \sim \text{SVM}$ 。

定理 2 设 $n+1$ 维向量空间中存在关于 Y 与 $X(Y \in R, X \in R^n)$ 的 SVM, 如果 X 的任一分量在定义域 $v_{i2} < X_i < v_{i1}$ ($i = 1, \dots, n; v_{i1}, v_{i2}$ 为常数) 内, $\exists l_{\min} (l_{\min} \in N)$, 在此定义域内随机抽取大于或等于 $l_{\min}$ 个样本建立的 $Y = f(X)$ ($f(X)$ 可能就是 $\text{SVM}_{l_{\min}}$) 与 SVM 近似, 即 $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} Y, f(X) \sim \text{SVM}$ 。

证明: $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} Y$, 所以 $\lim_{l \rightarrow \infty} |Y - f(X)| \rightarrow 0$;

根据抽样理论, $\exists l_{\min} (l_{\min} \in N), \lim_{l \rightarrow l_{\min}} |Y - f(X)| \xrightarrow{p} \epsilon (\epsilon$

为接近于 0 的随机误差), $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} Y, Y = \text{SVM}$, $\lim_{l \rightarrow \infty} f(X) \xrightarrow{p} \text{SVM}$, 所以, 在定义域内随机抽取 $l (l \geq l_{\min})$ 个样本建立的 $Y = f(X)$ ($f(X)$ 可能就是 $\text{SVM}_{l_{\min}}$) 与 SVM 近似。

即: $f(X) \sim \text{SVM}$

1.2 搜索 SV 的算法设计

根据 1.1 的证明得知, 从样本空间抽样建立的 $Y = f(X)$ 与 SVM 近似。由于 SV 分布于 SVM 超平面附近, 距超平面的相对位置较近^[2], 因而, 它也分布于近似超平面 $f(X)$ 的附近。

1.2.1 搜索策略

近似超平面的一般形式:

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

Step1: 对近似超平面进行向量空间转换;

$$x \mapsto x' = \varphi(x) \quad (2)$$

$$y(x) = w^T x' + b \quad (3)$$

Step2: 计算 x' 到 (3) 式的距离 d ;

$$d = |w^T x' + b - y| / \sqrt{\|w\|^2 + 1} \quad (4)$$

Step3: 按照 d 值的从小到大原则, 选择 m 个对应 d 值较小的 x 与上次搜索中的支持向量一起构成新的训练样本子集 T_s , 训练新的近似超平面, 然后逐步迭代, 构建 SVR。

1.2.2 终止条件

(1) 精确搜索终止条件: 搜索到训练集 T 中的全部支持向量。

(2) 不精确搜索的终止条件: 复相关系数大于 ρ ; 回归剩余离差平方和小于 Q ; 在 T_s 中连续 η 次没有搜索到新的支持向量。上述三项终止条件中的任意一项被满足即可终止。

2 算法实例

根据初始近似超平面和构建的 SVM 类型的不同, 可以设计一类从近似超平面到 SVR 的算法。下面以初始近似超平面为多元回归平面, 构建 LS-SVM 为例列举算法实例。

2.1 LS-SVM 的线性方程组解法^[4]

LS-SVM 的形式:

$$y(x) = \sum_{i=1}^k \alpha_i K(x, x_i) + b \quad (5)$$

其中: $K(x, x_i)$ 为 Mercer 核函数。

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \quad (6)$$

根据 Karush-Kuhn-Tucker (KKT) 条件, 得如下线性方程组:

$$\begin{bmatrix} 0 & 1 \\ 1 & K + \gamma I \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (7)$$

其中: K 为由 (7) 式计算的核矩阵; $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_k)$ 。

解线性方程组, 得 b, a ;

$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & K + \gamma I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (8)$$

$\alpha_i \neq 0$ 的为支持向量。

2.2 数据预处理及多元回归的计算

对数据作归一化处理:

$$x_{ij} = \frac{x_{ij} - x_{j \min}}{x_{j \max} - x_{j \min}} \quad (9)$$

($i = 1, 2, \dots, l; j = 1, 2, \dots, n+1$)。

多元回归模型:

$$y(x) = w^T x + b \quad (10)$$

求解得:

$$[w^T, b] = y \begin{bmatrix} x \\ 1 \end{bmatrix}' \left[\begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}' \right]^{-1} \quad (11)$$

2.3 从多元回归平面构建 LS-SVM 的算法及复杂度分析

2.3.1 从多元回归平面构建 LS-SVM 的算法

Step1: T_s, SV_s 和 SV_r 置空; 在 T 中, 随机抽取 k 个样本写入 T_s , 用公式 (11) 计算 (10) 的参数 W^T 和 b 。

Step2: 用 (4) 式计算 T 中向量 x 距 T_s 建立的回归平面 (10) 的距离, 按照 d 值从小到大, 从 T 中选取对应的 k 个向量更新训练样本子集 T_s , 并在 T 中对这 k 向量作 '*' 标记。

Step3: 用 (6)、(8) 式训练 T_s 的超平面 (5)。

Step4: While ($T - *$ 不为 ϕ)

{

Step4-1: 将 T_s 训练的超平面 (5) 的支持向量追加写入 SV_s ;

Step4-2: 将 T_s 训练的超平面 (5) 转换为 (3);

Step4-3: 将 $T - *$ 中的向量 x 映射到 $x' = \varphi(x)$;

Step4-4: 用 (4) 式计算向量 x' 距 T_s 训练的超平面转换成的 (3) 的距离 d ;

Step4-5: 按照 d 值从小到大和 $x \mapsto x' = \varphi(x)$ 的映射关系, 从 $T - *$ 中选取对应的 m 个向量与 SV_s 中的向量一起更新训练样本子集 T_s ;

Step4-6: 在 T 中对这 m 个样本作 * 标记;

Step4-7: 用 (6)、(8) 式训练 T_s 的超平面 (5)。

}

Step5: If (不满足终止条件)

{

Step5-1: T_s 训练的超平面 (5) 的支持向量置换 SV_r 中向量;

Step5-2: 去除 T 中的所有 * 标记, 然后对 T 中包含在 SV_r 中的向量作 * 标记;

Step5-3: goto Step4.

}

Step6: 结束。

2.3.2 复杂度分析

精确搜索的最坏情况 $m=1$, 循环 l 次 (l 为训练样本数),

(下转第 193 页)

