

精确增量式 ε 型孪生支持向量回归机

曹 杰, 顾斌杰[†], 潘 丰, 熊伟丽

(江南大学 轻工过程先进控制教育部重点实验室, 江苏 无锡 214122)

摘要: 为了解决现有 ε 型孪生支持向量回归机的训练算法无法高效处理线性回归的增量学习问题, 提出了一种精确增量式 ε 型孪生支持向量回归机(AIETSVR). 首先通过计算新增样本的拉格朗日乘子以及调整边界样本的拉格朗日乘子, 尽可能减少新增样本的二次损失对原有样本的影响, 使得大部分原有样本依然满足Karush–Kuhn–Tucker (KKT)条件, 从而获得一个有效的初始状态; 其次对异常拉格朗日乘子逐步调整至满足KKT条件; 然后从理论上分析了AIETSVR的可行性和有限收敛性; 最后在基准测试数据集上进行仿真. 结果表明, 与现有的代表性算法相比, AIETSVR能够获得精确解, 在缩短大规模数据集的训练时间上优势显著.

关键词: 机器学习; 增量学习; 在线学习; 孪生支持向量回归机; 学习算法; 可行性分析; 有限收敛性分析

引用格式: 曹杰, 顾斌杰, 潘丰, 等. 精确增量式 ε 型孪生支持向量回归机. 控制理论与应用, 2022, 39(6): 1020 – 1032

DOI: 10.7641/CTA.2021.00517

Accurate incremental ε -twin support vector regression

CAO Jie, GU Bin-jie[†], PAN Feng, XIONG Wei-li

(Jiangnan University, Key Laboratory of Advanced Process Control for Light Industry,
Ministry of Education, Wuxi Jiangsu 214122, China)

Abstract: In ε -twin support vector regression, to solve the problem that the existing algorithms can not efficiently deal with the incremental learning for linear regression, an accurate incremental ε -twin support vector regression (AIETSVR) is proposed. First, by calculating the Lagrangian multiplier of the new sample and adjusting the Lagrangian multipliers of the boundary samples, the influence generated by the quadratic loss of the new sample on the existing samples is minimized. Therefore, most of the existing samples still meet the Karush–Kuhn–Tucker (KKT) conditions, and a valid initial state is obtained. Then, the exceptional Lagrangian multipliers are gradually adjusted to conform to the KKT conditions. Next, the feasibility and finite convergence of AIETSVR are theoretically analyzed. Finally, the simulation is conducted on benchmark datasets. Compared with the existing representative algorithms, the results show that AIETSVR can obtain accurate solutions and has a great advantage in shortening training time for large-scale dataset.

Key words: machine learning; incremental learning; online learning; twin support vector regression; learning algorithm-s; feasibility analysis; finite convergence analysis

Citation: CAO Jie, GU Binjie, PAN Feng, et al. Accurate incremental ε -twin support vector regression. *Control Theory & Applications*, 2022, 39(6): 1020 – 1032

1 引言

支持向量机(support vector machine, SVM)^[1]是一种经典的分类算法, 被广泛运用于文本分类^[2]、图像分类^[3]、自动程序验证^[4]等领域. 之后, SVM的思想被用于解决回归问题, 即支持向量回归机(support vector regression, SVR)^[5]. SVR通过将样本的输出值向上和向下平移并作为特征值, 再分别设置正负标签, 形成两类样本, 然后构造将两类样本分开的超平面, 获得原始样本的回归平面, 即将回归问题转化为分类问题

来解决.

近年来, 由SVM发展而来的一种分类算法, 孪生支持向量机(twin support vector machine, TSVM)^[6]兴起. 与SVM不同, TSVM通过求解两个小规模二次规划问题, 构造两个非平行的超平面, 把训练时间减小到SVM的四分之一, 并且具有更好的泛化性能. 之后, Peng基于TSVM的思想提出了孪生支持向量回归机(twin support vector regression, TSVR)^[7]. 实验结果表明, TSVR在训练速度和泛化性能上均优于传统

收稿日期: 2020–08–08; 录用日期: 2021–08–24.

[†]通信作者. E-mail: gubinjie1980@126.com. Tel.: +86 13812203908

本文责任编辑: 陈杰.

国家自然科学基金项目(61773182)资助.

Supported by the National Natural Science Foundation of China (61773182).

SVR. 因此, TSVR迅速成为支持向量机领域的研究热点. 例如, 约简TSVR^[8]通过减小核矩阵的规模, 加快了训练速度; ε 型TSVR (ε -twin support vector regression, ETSVR)^[9]改进了目标函数并运用逐次超松弛技术加快了训练速度; ν 型TSVR^[10]通过引入 $\nu\varepsilon$ 项, 优化 ε 的值, 提高了模型的泛化性能; 基于快速聚类的加权TSVR^[11]通过引入协方差矩阵和加权对角矩阵, 提高了算法的抗干扰能力.

以上研究致力于目标函数或者相关离线训练算法的研究. 在实际应用场景中, 样本一般是在线增量提供的, 如果使用离线算法训练模型, 那么每次更新都需要重新训练, 无法满足系统实时性的要求. 而增量学习算法利用已有的训练结果, 快速更新模型, 能够提高系统的实时性. 目前为止, 在增量式SVM与SVR的研究上, 学者们取得了很多研究成果. 2000年, Cauwenberghs等人创新性地提出了增量式与减量式SVM^[12], 基于已有模型通过有限次数迭代使得新增样本满足Karush-Kuhn-Tucker (KKT)条件. 之后, 该方法又被扩展到解决精确增量式SVR (accurate online support vector regression, AOSVR)^[13]、多增量式与减量式SVM^[14]、精确增量式 ν 型SVR^[15]等问题. 并且该类算法的可行性和有限收敛性在理论上也得到了严格的证明^[16].

然而, 对于增量式TSVM和增量式TSVR, 目前的研究还较少, 主要是针对最小二乘型问题或者是使用梯度下降法进行优化. 例如, 改进增量式最小二乘TSVM^[17]改善了低维数据情况下模型的训练速度; 增量式最小二乘TSVR^[18]简化了矩阵求逆运算的复杂度, 提高了算法运行的效率; 增量式和减量式模糊有界TSVM^[19]利用具有收缩策略的对偶坐标下降法求解对偶问题, 提高了模型更新的速度.

对于ETSVR模型, 目前还没相应的增量学习算法, 本文尝试把AOSVR的思想用于解决ETSVR的增量学习问题, 提出一种精确增量式 ε 型孪生支持向量回归机 (accurate incremental ε -twin support vector regression, AIETSVR). 首先, 通过提前调整拉格朗日乘子, 判断可行域内异常拉格朗日乘子的目标值并在计算调整步长时进行修正, 解决如下两个问题: 1) 目标函数中的二次损失项使得新增样本后模型的对偶问题产生较大的变化, 可能出现大量原有样本的拉格朗日乘子不符合KKT条件的情况, 无法获得有效的初始状态; 2) 可能出现异常样本的拉格朗日乘子在边界内或者在边界外的情况, 当在边界内时, 往两边调整都有可能无法确定方向, 当在边界外时, 需要判断移入各个集合步长的有效性. 接着, 从理论上分析了AIETSVR的可行性和有限收敛性. 最后, 为了验证AIETSVR算法的可行性和有限收敛性以及解的精确性和在缩短训练时间上的优势, 在基准测试数据集上进行实

验.

为了便于理解, 下面列出本文算法描述中用到的符号表示的含义: $\varnothing$ 表示空集; Δ 表示在调整过程中的变化量; $\det(\cdot)$ 表示对应方阵的行列式; $\mathbf{0}$ 和 $\mathbf{e}$ 为相应维数元素全为0和1的列向量; $\mathbf{I}$ 为相应维数的单位矩阵. G_{ij} 表示矩阵 $\mathbf{G}$ 的第 i 行第 j 列的元素; $\mathbf{G}_{\setminus ij}$ 表示删除了矩阵 $\mathbf{G}$ 第 i 行元素和第 j 列元素后的矩阵; $\mathbf{G}_{*j}$ 表示矩阵 $\mathbf{G}$ 的第 j 列向量; $\mathbf{G}_{i*}$ 表示矩阵 $\mathbf{G}$ 的第 i 行向量; $\mathbf{G}_{S_1*}$ 表示由向量 $\mathbf{G}_{i*}$, $i \in S_1$ 组成的矩阵; $\mathbf{G}_{S_1 S_1}$ 表示由 $\mathbf{G}_{ij}$, $i, j \in S_1$ 组成的矩阵. α_i 表示向量 α 的第 i 个元素; α_{S_1} 表示由 α_i , $i \in S_1$ 组成的向量; $\alpha_{\setminus i}$ 表示删除向量 α 的第 i 个元素后的向量. $S_{1(i)}$ 表示 S_1 集合中第 i 个样本在训练集中的索引, $\tilde{\mathbf{M}}$ 表示更新后的 $\mathbf{M}$. 以此类推.

2 ε 型孪生支持向量回归机

对于一个给定的训练集数据 $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, 其中: $\mathbf{x}_i \in \mathbb{R}^m$ 为输入, $y_i \in \mathbb{R}$ 为输出, n 为训练集样本的个数, m 为每个样本的特征数, 则样本矩阵和输出矩阵可以分别表示为 $\mathbf{A} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n]^T \in \mathbb{R}^{n \times m}$ 和 $\mathbf{Y} = [y_1 \ y_2 \ \cdots \ y_n]^T \in \mathbb{R}^n$.

不同于传统支持向量回归机, ε 型孪生支持向量回归机减小了二次规划问题的规模, 通过寻找下界回归函数 $f_1(\mathbf{x}) = \mathbf{w}_1^T \mathbf{x} + b_1$ 和上界回归函数 $f_2(\mathbf{x}) = \mathbf{w}_2^T \mathbf{x} + b_2$, 其中 $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^m$ 为权重向量, $b_1, b_2 \in \mathbb{R}$ 为偏置, 构造两个非平行的超平面. 其原始问题可以表示为如下约束最优化问题^[9]:

$$\begin{cases} \min_{\mathbf{w}_1, b_1, \xi, \xi^*} \frac{1}{2} c_3 (\|\mathbf{w}_1\|^2 + b_1^2) + \frac{1}{2} (\xi^*)^T \xi^* + c_1 \mathbf{e}^T \xi, \\ \text{s.t. } \mathbf{Y} - (\mathbf{A}\mathbf{w}_1 + \mathbf{e}b_1) = \xi^*, \\ \mathbf{Y} - (\mathbf{A}\mathbf{w}_1 + \mathbf{e}b_1) \geq -\varepsilon_1 \mathbf{e} - \xi, \xi \geq \mathbf{0}, \end{cases} \quad (1)$$

$$\begin{cases} \min_{\mathbf{w}_2, b_2, \nu, \nu^*} \frac{1}{2} c_4 (\|\mathbf{w}_2\|^2 + b_2^2) + \frac{1}{2} (\nu^*)^T \nu^* + x c_2 \mathbf{e}^T \nu, \\ \text{s.t. } (\mathbf{A}\mathbf{w}_2 + \mathbf{e}b_2) - \mathbf{Y} = \nu^*, \\ (\mathbf{A}\mathbf{w}_2 + \mathbf{e}b_2) - \mathbf{Y} \geq -\varepsilon_2 \mathbf{e} - \nu, \nu \geq \mathbf{0}, \end{cases} \quad (2)$$

其中: $c_1, c_2 > 0$ 为惩罚参数, $c_3, c_4 > 0$ 为正则化参数, $\varepsilon_1, \varepsilon_2 > 0$ 为不敏感因子, $\xi, \xi^*, \nu, \nu^* \in \mathbb{R}^n$ 为松弛向量.

令 $\mathbf{G} = [\mathbf{e} \ \mathbf{A}]$, 则可得式(1)和式(2)的对偶问题如下:

$$\begin{cases} \min_{\alpha} \frac{1}{2} \alpha^T \mathbf{G} (\mathbf{G}^T \mathbf{G} + c_3 \mathbf{I})^{-1} \mathbf{G}^T \alpha - \\ \mathbf{Y}^T \mathbf{G} (\mathbf{G}^T \mathbf{G} + c_3 \mathbf{I})^{-1} \mathbf{G}^T \alpha + (\mathbf{e}^T \varepsilon_1 - \mathbf{Y}^T) \alpha, \\ \text{s.t. } \mathbf{0} \leq \alpha \leq c_1 \mathbf{e}, \end{cases} \quad (3)$$

$$\begin{cases} \min_{\gamma} \frac{1}{2} \gamma^T G(G^T G + c_4 I)^{-1} G^T \gamma + \\ Y^T G(G^T G + c_4 I)^{-1} G^T \gamma - (Y^T - e^T \varepsilon_2) \gamma, \\ \text{s.t. } 0 \leq \gamma \leq c_2 e. \end{cases} \quad (4)$$

令增广向量 $u_1 = [b_1 \ w_1^T]^T$, $u_2 = [b_2 \ w_2^T]^T$, 可由式(5)和式(6)求得模型的解

$$u_1 = (G^T G + c_3 I)^{-1} G^T (Y - \alpha), \quad (5)$$

$$u_2 = (G^T G + c_4 I)^{-1} G^T (Y + \gamma), \quad (6)$$

其中拉格朗日乘子 α 和 γ 通过求解式(3)和式(4)可得.

对于某一测试输入 x , 其对应输出为 $f(x) = (f_1(x) + f_2(x))/2$.

令 $M_1 = G^T G + c_3 I$, $Q_1 = G M_1^{-1} G^T$, 式(3)中的目标函数等价于

$$D_1 = \frac{1}{2} \alpha^T Q_1 \alpha - Y^T Q_1 \alpha + (e^T \varepsilon_1 + Y^T) \alpha. \quad (7)$$

基于KKT定理, 由 D_1 的一阶导数可得式(3)的KKT条件如下:

$$h_1(i) = \frac{\partial D_1}{\partial \alpha_i} = \begin{cases} \geq 0, & \alpha_i = 0, \\ = 0, & 0 < \alpha_i < c_1, \\ \leq 0, & \alpha_i = c_1, \end{cases} \quad (8)$$

其中 $h_1(i)$ 为边界距离函数.

由式(8)可知, 训练样本可以分为3个独立的子集, 由上到下分别为保留支持向量集 R_1 , 边界支持向量集 S_1 和错误支持向量集 E_1 . 同理可得, 式(4)的KKT条件.

3 增量算法

由于ETSVR中含有 $(\xi^*)^T \xi^*$ 和 $(\nu^*)^T \nu^*$ 项, 导致即使新增样本的拉格朗日乘子为0, 也会使 $f_1(x)$ 和 $f_2(x)$ 产生很大变化, 可能出现大量原有样本的拉格朗日乘子不满足KKT条件的情况, 使得逐步迭代调整的效率不高, 这也是AOSVR的方法不能直接用于解决ETSVR的增量学习问题的重要原因.

为此本文提出了一种精确增量式 ε 型孪生支持向量回归机. 该算法包含3个部分: 1) 高效更新逆矩阵 M_1^{-1} 和 $Q_1^{-1}(S_1, S_1)$, 这是由ETSVR模型包含一个二次损失的特点决定的, 每新增一个样本对偶问题就会发生较大改变; 2) 提前调整, 主要用于减少 M_1^{-1} 更新后不满足KKT条件的原有样本的个数; 3) 检查每个样本的KKT条件, 通过增量关系式, 对异常样本的拉格朗日乘子进行逐步调整.

当新增样本 (x_{n+1}, y_{n+1}) 时, 本文算法基于已有训练信息更新 $f_1(x)$ 和 $f_2(x)$ 的增广向量 u_1 和 u_2 . 由于 u_2 与 u_1 的更新相似, 接下来以更新 u_1 为例详细阐述

本文的增量学习算法. 令

$$R = Q_{1(S_1, S_1)}^{-1} = (G_{S_1*} M_1^{-1} G_{S_1*}^T)^{-1},$$

$$g_{n+1} = [1, x_{n+1}^T],$$

α_{n+1} 为新增样本的拉格朗日乘子.

3.1 高效更新逆矩阵 M_1^{-1} 和 R

当新增样本 (x_{n+1}, y_{n+1}) 加入训练集后, 首先需要更新逆矩阵 M_1^{-1} 和 R 进行更新.

引理1 设 $A \in \mathbb{R}^{n \times n}$ 正定, $u, v \in \mathbb{R}^n$ 是任意向量, 若 $1 + v^T A^{-1} u \neq 0$, 则有^[20]

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1} uv^T A^{-1}}{1 + v^T A^{-1} u}.$$

由引理1可知, M_1^{-1} 和 R 更新公式如下:

$$\begin{aligned} \tilde{M}_1^{-1} &= (G^T G + g_{n+1} g_{n+1}^T + c_3 I)^{-1} = \\ &= (M_1 + g_{n+1} g_{n+1}^T)^{-1} = \\ &= M_1^{-1} - Z Z^T / J, \end{aligned} \quad (9)$$

其中: $Z = M_1^{-1} g_{n+1}$, $J = 1 + g_{n+1}^T Z$.

$$\begin{aligned} \tilde{R} &= (G_{S_1*} \tilde{M}_1^{-1} G_{S_1*}^T)^{-1} = \\ &= (G_{S_1*} (M_1^{-1} - Z Z^T / J) G_{S_1*}^T)^{-1} = \\ &= (Q_{1(S_1, S_1)} - Z_1 Z_1^T / J)^{-1} = \\ &= R + Z_2 Z_2^T / J_1, \end{aligned} \quad (10)$$

其中: $Z_1 = G_{S_1*} Z$, $Z_2 = R Z_1$, $J_1 = J - Z_1^T Z_2$.

3.2 提前调整

由于式(1)的目标函数中二次损失项 $(\xi^*)^T \xi^*$ 的存在, 由式(9)更新 M_1^{-1} 将导致 M_1^{-1} 和 Q_1 发生较大的改变, 影响原有样本的边界距离. 为了尽可能减少影响保证原有样本边界距离不变, 使得大部分原有样本的拉格朗日乘子符合KKT条件, 获得一个有效的初始状态, 通过以下两个步骤提前计算 α_{n+1} 并调整 α_{S_1} .

步骤1 求 α_{n+1} 使逆矩阵 M_1^{-1} 更新后原有样本边界距离 $\tilde{h}_1(i)$, $i \leq n$ 尽可能保持不变, 推导如下:

$$\begin{aligned} \tilde{h}_1(i) &= y_i - (\tilde{Q}_{1(i*)}) \begin{bmatrix} Y - \alpha \\ y_{n+1} - \alpha_{n+1} \end{bmatrix} - \varepsilon_1 = \\ &= y_i - G_{i*} \tilde{M}_1^{-1} G^T (Y - \alpha) - \\ &= G_{i*} \tilde{M}_1^{-1} g_{n+1}^T (y_{n+1} - \alpha_{n+1}) + \varepsilon_1, \end{aligned} \quad (11)$$

其中 $\tilde{Q}_{1(i*)} = G_{i*} \tilde{M}_1^{-1} [G^T \ g_{n+1}^T]$.

将式(9)代入式(11)可得

$$\begin{aligned} \tilde{h}_1(i) &= y_i - G_{i*} (M_1^{-1} - Z Z^T / J) G^T (Y - \alpha) - \\ &= G_{i*} \tilde{M}_1^{-1} g_{n+1}^T (y_{n+1} - \alpha_{n+1}) + \varepsilon_1 = \\ &= y_i - G_{i*} M_1^{-1} G^T (Y - \alpha) + \varepsilon_1 + \\ &= G_{i*} (Z Z^T / J) G^T (Y - \alpha) - \\ &= G_{i*} \tilde{M}_1^{-1} g_{n+1}^T (y_{n+1} - \alpha_{n+1}) = \end{aligned}$$

$$h_1(i) + l_i - L_i(y_{n+1} - \alpha_{n+1}), \quad (12)$$

其中: $h_1(i) = y_i - \mathbf{G}_{i*} \mathbf{M}_1^{-1} \mathbf{G}^T (\mathbf{Y} - \boldsymbol{\alpha}) + \varepsilon_1$, $l_i = \mathbf{G}_{i*} (\mathbf{Z} \mathbf{Z}^T / J) \mathbf{G}^T (\mathbf{Y} - \boldsymbol{\alpha})$, $L_i = \mathbf{G}_{i*} \tilde{\mathbf{M}}_1^{-1} \mathbf{g}_{n+1}^T$.

令 $\tilde{\mathbf{H}}_1$ 和 $\mathbf{H}_1$ 分别为 $\tilde{h}_1(i)$ 和 $h_1(i)$ 组成的列向量, 则 $\tilde{\mathbf{H}}_1 - \mathbf{H}_1 = \mathbf{l} - \mathbf{L}(y_{n+1} - \alpha_{n+1})$, 通过求解如下无约束最小化问题, 可获得使得边界函数更新后原有样本边界距离变化尽可能小的 α_{n+1}

$$\min_{\alpha_{n+1}} (\mathbf{L}(y_{n+1} - \alpha_{n+1}) - \mathbf{l})^2, \quad (13)$$

上式的最优解为 $\alpha_{n+1} = y_{n+1} - (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T \mathbf{l}$.

步骤2 尽管原有样本的边界距离变化较小, 但是依然会导致 S_1 集中的样本偏离边界. 为了消除这种影响, 利用 $0 < \alpha_{S_1} < c_1$ 的可调整性, 使得 $\tilde{\mathbf{H}}_{1(S_1)}$ 为0, 即 S_1 集中的样本完全在边界上. 计算 $\Delta \alpha_{S_1}$ 以及更新 α_{S_1} 的公式如下:

$$\begin{cases} \Delta \alpha_{S_1} = -\tilde{\mathbf{R}} \tilde{\mathbf{H}}_{1(S_1)}, \\ \alpha_{S_1} \leftarrow \alpha_{S_1} + \Delta \alpha_{S_1}. \end{cases} \quad (14)$$

3.3 筛选异常样本

令 $\boldsymbol{\alpha} \leftarrow [\boldsymbol{\alpha}^T \alpha_{n+1}]^T$, $\mathbf{G} \leftarrow [\mathbf{G}^T \mathbf{g}_{n+1}^T]^T$, $\mathbf{Y} \leftarrow [\mathbf{Y}^T y_{n+1}]^T$, 重新计算 $\tilde{\mathbf{H}}_1$ 和 $\tilde{h}_1(n+1)$, 并令 $\tilde{\mathbf{H}}_1 \leftarrow [\tilde{\mathbf{H}}_1^T \tilde{h}_1(n+1)]^T$. 然后根据式(8), 筛选出拉格朗日乘子异常的样本组成集合 F_1 , 分为两种情况: 当 α_i 往 c_1 的方向调整, 样本 i 移入 I_1 集合; 当 α_i 往0的方向调整, 样本 i 移入 D_1 集合.

对于 S_1 集中的样本, 由于第3.2节步骤2中对 α_{S_1} 的调整, α_i 可能有两种不符合式(8)的情况: 当 $\alpha_i \leq 0$, 样本 i 移入 I_1 集合; 当 $\alpha_i \geq c_1$, 样本 i 移入 D_1 集合. 并且都需要更新逆矩阵 $\tilde{\mathbf{R}}$.

对于 R_1 集中的样本, 当 $\tilde{h}_1(i) \leq 0$, 样本移入 I_1 集合.

对于 E_1 集中的样本, 当 $\tilde{h}_1(i) \geq 0$, 样本移入 D_1 集合.

对于 $(\mathbf{x}_{n+1}, y_{n+1})$, 如果满足式(8), 移入相应集合. 如果不满足: 当 $\alpha_{n+1} \leq 0$, 移入 I_1 集合; 当 $\alpha_{n+1} \geq c_1$, 移入 D_1 集合; 否则, 由第5.2节定理5可知, 如果 $\tilde{h}_1(n+1) > 0$, 移入 D_1 集合, 如果 $\tilde{h}_1(n+1) < 0$, 移入 I_1 集合.

综上可得, 异常样本集合 $F_1 = I_1 \cup D_1$.

3.4 增量关系推导

在增量调整过程中, α_{R_1} 和 α_{E_1} 是不变的分别为0和 c_1 , 而 α_{S_1} 在0到 c_1 范围内可调, α_{F_1} 是需要调整的. 因此可以通过逐步调整 α_{S_1} 和 α_{F_1} , 在保证当前 S_1 , R_1 , E_1 集中样本始终满足KKT条件的前提下, 使 F_1 集合样本满足KKT条件. 由式(8)可求得 $\Delta \tilde{h}_1(i)$,

$\Delta \alpha_{S_1}$ 和 $\Delta \alpha_{F_1}$ 的关系式如下:

$$\Delta \tilde{h}_1(i) = \sum_{j \in S_1} \tilde{\mathbf{Q}}_{1(ij)} \Delta \alpha_j + \sum_{j \in F_1} \tilde{\mathbf{Q}}_{1(ij)} \Delta \alpha_j. \quad (15)$$

对于 S_1 集中的样本, 在增量调整过程中, $\tilde{h}_1(i)$ 的值恒为0. 因此, 可得 $\Delta \alpha_{S_1}$ 和 $\Delta \alpha_{F_1}$ 之间的关系式如下:

$$\Delta \alpha_{S_1} = -\tilde{\mathbf{R}} \tilde{\mathbf{Q}}_{1(S_1 F_1)} \Delta \alpha_{F_1} = \beta \Delta \alpha_{F_1}, \quad (16)$$

其中: $\beta = -\tilde{\mathbf{R}} \tilde{\mathbf{Q}}_{1(S_1 F_1)}$, $\beta_{ij} = -\tilde{\mathbf{R}}_{i*} \tilde{\mathbf{Q}}_{1(S_1 j)}$.

由式(15)和式(16)可得 $\Delta \tilde{\mathbf{H}}_1$ 和 $\Delta \alpha_{F_1}$ 之间的关系如下:

$$\begin{aligned} \Delta \tilde{\mathbf{H}}_1 &= \tilde{\mathbf{Q}}_{1(*S_1)} \Delta \alpha_{S_1} + \tilde{\mathbf{Q}}_{1(*F_1)} \Delta \alpha_{F_1} = \\ &(\tilde{\mathbf{Q}}_{1(*S_1)} \beta + \tilde{\mathbf{Q}}_{1(*F_1)}) \Delta \alpha_{F_1} = \rho \Delta \alpha_{F_1}, \end{aligned} \quad (17)$$

其中: $\rho = \tilde{\mathbf{Q}}_{1(*S_1)} \beta + \tilde{\mathbf{Q}}_{1(*F_1)}$, $\rho_{ij} = \tilde{\mathbf{Q}}_{1(iS_1)} \beta_{*j} + \tilde{\mathbf{Q}}_{1(ij)}$.

由于 F_1 集中的样本个数可能大于1, 因此借鉴文献[14], 寻求 α_{F_1} 向目标值调整的最短路径, 其中 I_1 , D_1 集合样本的目标值分别为 c_1 和0.

令 $\mathbf{d}$ 为 α_{F_1} 调整的目标值组成的向量, $\Delta \alpha_{F_1} = (\mathbf{d} - \alpha_{F_1}) \eta$, 其中 $\eta \geq 0$ 是步长, 代入式(16)和式(17)可得, $\Delta \alpha_{S_1}$ 和 $\Delta \tilde{\mathbf{H}}_1$ 与 η 之间的关系式分别如下:

$$\begin{cases} \Delta \alpha_{S_1} = \beta \Delta \alpha_{F_1} = \beta (\mathbf{d} - \alpha_{F_1}) \eta = \hat{\beta} \eta, \\ \Delta \tilde{\mathbf{H}}_1 = \rho \Delta \alpha_{F_1} = \rho (\mathbf{d} - \alpha_{F_1}) \eta = \hat{\rho} \eta, \end{cases} \quad (18)$$

其中: $\hat{\beta} = \beta (\mathbf{d} - \alpha_{F_1})$, $\hat{\rho} = \rho (\mathbf{d} - \alpha_{F_1})$.

3.5 计算最大步长 $\eta^{\max}$

每迭代过程中, 需要在以下4种情况中搜索最大步长 $\eta^{\max}$:

情况1 在 F_1 集合中, 样本 $(\mathbf{x}_i, y_i)$ 移入 S_1 , R_1 和 E_1 集合的步长对应如下:

$$\eta_i^{C_1} = \begin{cases} \frac{-\tilde{h}_1(i)}{\hat{\rho}_i}, & \tilde{h}_1(i) \hat{\rho}_i < 0, \\ \infty, & \tilde{h}_1(i) \hat{\rho}_i \geq 0, \end{cases} \quad (19)$$

$$\eta_i^{C_2} = \begin{cases} \frac{0 - \alpha_i}{d_i - \alpha_i}, & (0 - \alpha_i)(d_i - \alpha_i) > 0, \\ \infty, & (0 - \alpha_i)(d_i - \alpha_i) \leq 0, \end{cases} \quad (20)$$

$$\eta_i^{C_3} = \begin{cases} \frac{c_1 - \alpha_i}{d_i - \alpha_i}, & (c_1 - \alpha_i)(d_i - \alpha_i) > 0, \\ \infty, & (c_1 - \alpha_i)(d_i - \alpha_i) \leq 0. \end{cases} \quad (21)$$

由第3.3节可知, 可能出现 α_i 在可行域外的情况, 此时需要对 $\eta_i^{C_1}$, $\eta_i^{C_2}$ 和 $\eta_i^{C_3}$ 的有效性进行判断, 并对无

效的步长进行修正,共有如下6种情况:

$$\begin{cases} \eta_i^{C_2} = \eta_i^{C_1} = \infty, & \alpha_i < 0, \tilde{h}_1(i) > 0, \eta_i^{C_2} > \eta_i^{C_1}, \\ \eta_i^{C_1} = \infty, & \alpha_i < 0, \tilde{h}_1(i) < 0, \eta_i^{C_2} > \eta_i^{C_1}, \\ \eta_i^{C_2} = \infty, & \alpha_i < 0, \tilde{h}_1(i) < 0, \eta_i^{C_2} < \eta_i^{C_1}, \\ \eta_i^{C_3} = \eta_i^{C_1} = \infty, & \alpha_i > c_1, \tilde{h}_1(i) < 0, \eta_i^{C_3} > \eta_i^{C_1}, \\ \eta_i^{C_1} = \infty, & \alpha_i > c_1, \tilde{h}_1(i) > 0, \eta_i^{C_3} > \eta_i^{C_1}, \\ \eta_i^{C_3} = \infty, & \alpha_i > c_1, \tilde{h}_1(i) > 0, \eta_i^{C_3} < \eta_i^{C_1}. \end{cases} \quad (22)$$

情况2 在 S_1 集合中,样本 (x_i, y_i) 移入 R_1 或 E_1 集合的步长如下:

$$\eta_i^{S_1} = \begin{cases} \frac{0 - \alpha_i}{\hat{\beta}_i}, & \hat{\beta}_i < 0, \\ \frac{c_1 - \alpha_i}{\hat{\beta}_i}, & \hat{\beta}_i > 0, \\ \infty, & \hat{\beta}_i = 0. \end{cases} \quad (23)$$

情况3 在 R_1 集合中,样本 (x_i, y_i) 移入 S_1 集合的步长如下:

$$\eta_i^{R_1} = \begin{cases} \frac{-\tilde{h}_1(i)}{\hat{\rho}_i}, & \hat{\rho}_i \tilde{h}_1(i) < 0, \\ \infty, & \hat{\rho}_i \tilde{h}_1(i) \geq 0. \end{cases} \quad (24)$$

情况4 在 E_1 集合中,样本 (x_i, y_i) 移到 S_1 集合的步长如下:

$$\eta_i^{E_1} = \begin{cases} \frac{-\tilde{h}_1(i)}{\hat{\rho}_i}, & \hat{\rho}_i \tilde{h}_1(i) < 0, \\ \infty, & \hat{\rho}_i \tilde{h}_1(i) \geq 0. \end{cases} \quad (25)$$

最终, $\eta^{\max}$ 由式(26)确定

$$\eta^{\max} = \min(\eta^{C_1}, \eta^{C_2}, \eta^{C_3}, \eta^{S_1}, \eta^{R_1}, \eta^{E_1}). \quad (26)$$

接下来,由式(18)更新 $\Delta\alpha_{S_1}$, $\Delta\tilde{H}_1$, α_{F_1} , α_{S_1} 以及 $\tilde{H}_1$,并更新相应的集合,当 S_1 集合发生变动时,需要更新 $\tilde{R}$,直到 $F_1 = \emptyset$ 时停止迭代.

3.6 高效更新逆矩阵 $\tilde{R}$

在筛选异常样本以及迭代调整的过程中,当 S_1 集合发生改变时,需要对 $\tilde{R}$ 进行更新.

引理2 设 $A \in \mathbb{R}^{n \times n}$ 正定, $b \in \mathbb{R}^n$, $d \in \mathbb{R}$, 且 $d - b^T A^{-1} b \neq 0$, 令 $Z = -A^{-1}b$, $J = d + b^T Z$, 则有^[21]

$$\begin{bmatrix} A & b \\ b^T & d \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0^T & 0 \end{bmatrix} + \frac{1}{J} \begin{bmatrix} Z \\ 1 \end{bmatrix} \begin{bmatrix} Z^T \\ 1 \end{bmatrix}.$$

当样本 t 移入 S_1 集合时,由引理2可知, $\tilde{R}$ 的扩展公式如下:

$$\tilde{R} \leftarrow \begin{bmatrix} \tilde{R} & 0 \\ 0^T & 0 \end{bmatrix} + \frac{1}{\rho_{tt}} \begin{bmatrix} \beta_{*t} \\ 1 \end{bmatrix} \begin{bmatrix} \beta_{*t}^T \\ 1 \end{bmatrix}. \quad (27)$$

当样本 t 移出 S_1 集合时, $\tilde{R}$ 的收缩公式如下^[15]:

$$\tilde{R} \leftarrow \tilde{R}_{\setminus tt} - (\tilde{R}_{*t} \tilde{R}_{t*})_{\setminus tt} / \tilde{R}_{tt}. \quad (28)$$

3.7 算法步骤

AIETSVR算法的整体步骤如下:

步骤1 读取一个新增样本 (x_{n+1}, y_{n+1}) , 由式(9)和式(10)求得 $\tilde{M}_1^{-1}$ 和 $\tilde{R}$;

步骤2 计算 α_{n+1} 并由式(14)调整 α_{S_1} (详见第3.2节);

步骤3 接下来, 令 $\alpha \leftarrow [\alpha^T \ \alpha_{n+1}]^T$, $G \leftarrow [G^T \ g_{n+1}^T]^T$, $Y \leftarrow [Y^T \ y_{n+1}]^T$, 重新计算 $\tilde{H}_1$ 和 $\tilde{h}_1(n+1)$, 并令 $\tilde{H}_1 \leftarrow [\tilde{H}_1^T \ \tilde{h}_1(n+1)]^T$, 然后检查所有样本的KKT条件, 筛选出异常样本集合 F_1 (详见第3.3节);

步骤4 由式(18)分别求得 $\hat{\beta}$ 和 $\hat{\rho}$, 然后由式(19)~(26)计算最大步长 $\eta^{\max}$, 并更新 α_{F_1} , α_{S_1} , $\tilde{H}_1$, R_1 集合, S_1 集合, E_1 集合, F_1 集合和 $\tilde{R}$, 直至 $F_1 = \emptyset$ 时停止迭代(详见第3.4节和第3.5节);

步骤5 令 $M_1^{-1} = \tilde{M}_1^{-1}$, $R = \tilde{R}$, 由式(5)计算更新后模型的解 u_1 , 并保存 u_1 , α , M_1^{-1} , R , R_1 集合, S_1 集合和 E_1 集合.

同理, 可得到新增样本后上界回归函数的增广向量 u_2 .

4 算法复杂度分析与对比

4.1 AIETSVR算法复杂度分析

为了说明本文算法在处理增量问题时, 训练时间上的优势, 以更新增广向量 u_1 为例, 对第3.7节中步骤1到步骤5分别分析其时间复杂度, 由于乘法的复杂度远高于加法, 因此以下分析中只考虑乘法运算的次数. 假设当前已更新完 n 个 m 维样本, S_1 和 F_1 集合中的样本个数分别为 l_{S_1} 和 l_{F_1} .

考虑到第3.2节中的步骤1, 使得新增样本在二次项上产生的损失对原有样本的影响变小, 因此可以认为 $l_{F_1} \approx 1$. 并且在增量训练时, 数据特征维度 m 不会改变, 一般情况下 $m \ll n$, 数据量 n 和支持向量数 l_{S_1} 会有增长, 所以第3.7节中步骤1到步骤5的时间复杂度可以分别简化为: 步骤1: $O(l_{S_1}^2) + O(l_{S_1})$; 步骤2: $O(n) + O(l_{S_1}^2)$; 步骤3: $O(n)$; 步骤4: 计算 $\hat{\beta}$ 和 $\hat{\rho}$ 的时间复杂度为 $O(l_{S_1}^2) + O(l_{S_1}) + O(n)$, $\tilde{R}$ 扩展的时间复杂度为 $O(l_{S_1}^2) + O(l_{S_1})$, $\tilde{R}$ 收缩的时间复杂度为 $O(l_{S_1}^2)$; 步骤5: $O(l_{S_1}^2) + O(n)$. 由于 S_1 集合中的样本为边界上的样本, 即支持向量, 具有稀疏性且增长缓慢, 因此, 步骤4中每次迭代的时间复杂度较小, 且随着样本的增加, 增长缓慢. 综上所述, AIETSVR的时间复杂度为: 迭代前 $O(l_{S_1}^2) + O(l_{S_1}) + O(n)$, 每次迭代 $O(l_{S_1}^2) + O(l_{S_1}) + O(n)$.

但是, 本文算法需要在计算过程中保存逆矩阵 M_1^{-1} 和 R , 大小分别为 m^2 和 $l_{S_1}^2$. 由于 m 不会改变且一般情况下 $m \ll n$, 支持向量具有稀疏性, l_{S_1} 不会很大, 因此这两个逆矩阵不会占用很大的内存空间. 除此以外, 训练样本矩阵和3个集合的划分情况(以向量形式)需要保存, 大小分别为 $n \times m$ 和 $n \times 1$.

4.2 不同算法的时间复杂度对比

本节选取算法思想相近的AOSVR, 以及ETSVR的两种常用求解算法: 逐次超松弛法(ETSVRSOR)和内点法(ETSVRQP), 分析时间复杂度, 体现AIETSVR在训练时间上的优势. 令AOSVR算法的支持向量个数为 l_S .

对于小规模数据集, $m \approx l_S \approx l_{S_1} \approx n$, 各个算法的时间复杂度如下: AOSVR迭代前 $O(n^2)$, 每次迭代 $O(n^2) + O(n)$; ETSVRSOR迭代前 $O(n^3)$, 每次迭代 $O(n^2) + O(n)$; ETSVRQP 迭代前 $O(n^3)$, 每次迭代 $O(n^3)$; AIETSVR 迭代前 $O(n^2) + O(n)$, 每次迭代 $O(n^2) + O(n)$. 综上所述, 在小规模数据集上, ETSVRQP的时间复杂度较高, 而AOSVR, ETSVRSOR和AIETSVR时间复杂度相当.

对于大规模数据集, $m \ll n$, 各个算法的时间复杂度如下: AOSVR迭代前 $O(n)$, 每次迭代 $O(l_S) + O(l_S^2) + O(n)$; ETSVRSOR迭代前 $O(n) + O(n^2)$, 每次迭代 $O(n^2) + O(n)$; ETSVRQP 迭代前 $O(n) + O(n^2)$, 每次迭代 $O(n^3)$.

综上所述, ETSVRSOR时间复杂度的数量级为 $O(n^2)$ 要小于ETSVRQP的数量级 $O(n^3)$. AOSVR时间复杂度的数量级为 $O(n) + O(l_S^2)$, 由于在大规模数据集上支持向量个数 l_S 远小于样本数 n , 因此时间复杂度小于ETSVRQP和ETSVRSOR.

而AIETSVR的数量级为 $O(l_{S_1}^2) + O(n)$, 由于ETSVR单个目标函数仅有一个不等约束是SVR的一半, 即 $l_{S_1} \approx l_S/2$. 因此AIETSVR的时间复杂度小于AOSVR, 延续了ETSVR相对于SVR在训练时间上的优势, 并且时间复杂度远小于ETSVRSOR和ETSVRQP这两种离线算法, 适合于大规模数据集的在线增量训练.

5 可行性和有限收敛性分析

通过可行性分析, 证明AIETSVR算法中逆矩阵更新的可行性以及拉格朗日乘子调整的可靠性. 其中, 定理1是对文献[16]中定理2的进一步延伸, 定理2和定理3由文献[16]中的证明推广而来. 通过有限收敛性分析, 证明AIETSVR算法经过有限次数调整收敛到最优解. 其中, 定理4和定理5的证明分别参照文献[16]和文献[15]. 令 (x_c, y_c) 为 F_1 集合中的一个候选样本.

5.1 可行性分析

本文把算法的可行性证明, 分为以下2个问题:

1) 当 $S_1 \neq \emptyset$ 时, $Q_{1(S_1 S_1)}$ 是正定矩阵即 $\tilde{R} =$

$Q_{1(S_1 S_1)}^{-1}$ 始终存在(定理1);

2) 当样本 t 移入(移出) S_1 集合后, 在下一步调整中, 样本 t 不会反向移出(移入) S_1 集合(定理2和定理3).

引理3 若 $A \in \mathbb{R}^{m \times n}$ 行满秩, $B \in \mathbb{R}^{n \times n}$ 是正定阵, 则 ABA^T 是正定阵. 证明详见文献[21].

定理1 当 $S_1 \neq \emptyset$ 时, $Q_{1(S_1 S_1)}$ 是正定矩阵即 $\tilde{R} = Q_{1(S_1 S_1)}^{-1}$ 始终存在.

证 已知 $\tilde{Q}_{1(S_1 S_1)} = G_{S_1*} \tilde{M}_1^{-1} G_{S_1*}^T$. 由于 $\tilde{M}_1^{-1}$ 中存在正则项, 易知 $\tilde{M}_1^{-1}$ 是正定矩阵. 结合引理3可知, 当矩阵 G_{S_1*} 行满秩时, 矩阵 $\tilde{Q}_{1(S_1 S_1)}$ 是正定的. 接下来, 需要证明 S_1 集合不会移入与 S_1 集合中已有样本线性相关的样本.

假设存在样本 t , 使得 $G_{t*} = b^T G_{S_1*}$, 其中 b 为相应维度元素不全为0的列向量, 即 G_{t*} 与 G_{S_1*} 中行向量线性相关, 那么有

$$\begin{aligned} \rho_{tc} &= \tilde{Q}_{1(t S_1)} \beta_{*c} + \tilde{Q}_{1(tc)} = \\ &G_{t*} \tilde{M}_1^{-1} G_{S_1*}^T (-\tilde{R} \tilde{Q}_{1(S_1 c)}) + G_{t*} \tilde{M}_1^{-1} G_{c*}^T = \\ &b^T G_{S_1*} \tilde{M}_1^{-1} G_{S_1*}^T (-\tilde{R} \tilde{Q}_{1(S_1 c)}) + \\ &b^T G_{S_1*} \tilde{M}_1^{-1} G_{c*}^T = \\ &b^T \tilde{Q}_{1(S_1 S_1)} (-\tilde{R} \tilde{Q}_{1(S_1 c)}) + b^T \tilde{Q}_{1(S_1 c)} = 0, \end{aligned}$$

即 $\rho_{t*} = 0^T$, 由式(19)可知 $\Delta \tilde{h}_1(t) = 0$, 即样本 t 不会移入 S_1 集合.

综上得证定理1. 证毕.

引理4 当向 S_1 集合增加一个样本 t 时, 相应地 $\rho_{tt} > 0$.

证 由于 $\tilde{M}_1^{-1}$ 是正定矩阵, 存在唯一cholseky分解 $\tilde{M}_1^{-1} = L' L'^T$. 令 $L'' = GL'$, 那么 $\tilde{Q}_1 = L'' L''^T$. 基于此, 进一步有

$$\begin{aligned} \rho_{tt} &= \tilde{Q}_{1(tt)} + \tilde{Q}_{1(t S_1)} \beta_{*t} + \beta_{*t}^T \rho_{S_1 t} = \\ &\tilde{Q}_{1(tt)} + \tilde{Q}_{1(t S_1)} \beta_{*t} + \\ &\beta_{*t}^T (\tilde{Q}_{1(S_1 S_1)} \beta_{*t} + \tilde{Q}_{1(S_1 t)}) = \\ &L''_{t*} L''_{t*}^T + L''_{t*} L''_{S_1*}^T \beta_{*t} + \\ &\beta_{*t}^T (L''_{S_1*} L''_{S_1*}^T \beta_{*t} + L''_{S_1*} L''_{t*}^T) = \\ &L''_{t*} (L''_{t*}^T + L''_{S_1*}^T \beta_{*t}) + \\ &\beta_{*t}^T L''_{S_1*} (L''_{S_1*}^T \beta_{*t} + L''_{t*}^T) = \\ &(L''_{t*} + \beta_{*t}^T L''_{S_1*}) (L''_{t*}^T + L''_{S_1*}^T \beta_{*t}) = \\ &\|L''_{t*} + \beta_{*t}^T L''_{S_1*}\|^2 \geq 0. \end{aligned}$$

如果 $L''_{t*} + \beta_{*t}^T L''_{S_1*} = 0$, 进一步可得 $\rho_{tt} = 0$, 这与前提条件向 S_1 集合增加一个样本 t 相矛盾, 因此 $L''_{t*} + \beta_{*t}^T L''_{S_1*} \neq 0$, 所以得证 $\rho_{tt} > 0$.

引理5 当 S_1 集合移出样本 t 时, $\tilde{R}_{tt} > 0$.

证 由逆矩阵的定义及定理1, 易得出该结论, 此

处省略. 证毕.

定理2 当 S_1 集合移入样本 t 后, 下一步调整中, 样本 t 不会反向移出.

证 假设前一次调整记为 k 且 $S_1^{[k]}$ 中有 n 个样本, 则下次调整记为 $k+1$, $S_1^{[k+1]}$ 中有 $n+1$ 个样本, 那么有

$$\begin{aligned} \beta_{*c}^{[k+1]} &= \begin{bmatrix} \beta_{(S_1^{[k+1]c})}^{[k+1]} \\ \beta_{(tc)}^{[k+1]} \end{bmatrix} = -\tilde{R}^{[k+1]} \cdot \begin{bmatrix} \tilde{Q}_{1(S_1^{[k+1]c})} \\ Q_{1(tc)} \end{bmatrix} = \\ &= -\left(\begin{bmatrix} \tilde{R}^{[k]} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{\rho_{tt}^{[k]}} \begin{bmatrix} \beta_{*t}^{[k]} \\ 1 \end{bmatrix} \begin{bmatrix} \beta_{*t}^{[k]} \\ 1 \end{bmatrix}^T \right) \cdot \begin{bmatrix} \tilde{Q}_{1(S_1^{[k]c})} \\ \tilde{Q}_{1(tc)} \end{bmatrix} = \\ &= \begin{bmatrix} \beta_{*c}^{[k]} \\ 0 \end{bmatrix} - \frac{1}{\rho_{tt}^{[k]}} \begin{bmatrix} \beta_{*t}^{[k]} \\ 1 \end{bmatrix} \begin{bmatrix} \beta_{*t}^{[k]} \\ 1 \end{bmatrix}^T \begin{bmatrix} \tilde{Q}_{1(S_1^{[k]c})} \\ \tilde{Q}_{1(tc)} \end{bmatrix} = \\ &= \begin{bmatrix} \beta_{*c}^{[k]} \\ 0 \end{bmatrix} - \frac{\rho_{ct}^{[k]}}{\rho_{tt}^{[k]}} \begin{bmatrix} \beta_{*t}^{[k]} \\ 1 \end{bmatrix}. \end{aligned}$$

于是有 $\beta_{tc}^{[k+1]} = -\rho_{ct}^{[k]}/\rho_{tt}^{[k]}$, 又因为

$$\begin{aligned} \rho_{tc}^{[k+1]} &= \tilde{Q}_{1(tc)} + \tilde{Q}_{1(tS_1^{[k+1]})} \beta_{*c}^{[k+1]} = \\ &= \tilde{Q}_{1(tc)} + [\tilde{Q}_{1(tS_1^{[k]})} Q_{1(tt)}] \cdot \left(\begin{bmatrix} \beta_{*c}^{[k]} \\ 0 \end{bmatrix} - \frac{\rho_{ct}^{[k]}}{\rho_{tt}^{[k]}} \begin{bmatrix} \beta_{*t}^{[k]} \\ 1 \end{bmatrix} \right) = \\ &= \tilde{Q}_{1(tc)} + \tilde{Q}_{1(tS_1^{[k]})} \cdot (\beta_{*c}^{[k]} - (\rho_{ct}^{[k]}/\rho_{tt}^{[k]})\beta_{*t}^{[k]}) - \\ &= Q_{1(tt)} \cdot (\rho_{ct}^{[k]}/\rho_{tt}^{[k]}) = \\ &= \rho_{tc}^{[k]} - (\rho_{ct}^{[k]}/\rho_{tt}^{[k]}) \cdot (Q_{1(tt)} + \tilde{Q}_{1(tS_1^{[k]})} \beta_{*t}^{[k]}) = \\ &= \rho_{tc}^{[k]} - \rho_{ct}^{[k]} = 0, \end{aligned}$$

所以 $\beta_{tc}^{[k+1]} = -\rho_{ct}^{[k]}/\rho_{tt}^{[k]}$.

这表明向 S_1 集合移入样本 t 后, 下一步调整中, 样本 t 不会被反向移出. 因为样本 t 移入 S_1 集合只有两种方式: 从 R_1 集合移入 S_1 集合或从 E_1 集合移入 S_1 集合. 对于方式一, $\alpha_t^{[k]} = 0$, $\tilde{h}_1^{[k]}(t) > 0$ 且 $\rho_{tc}^{[k]} < 0$, 由引理4可得 $\beta_{tc}^{[k+1]} > 0$; 对于方式二, $\alpha_t^{[k]} = c_1$, $\tilde{h}_1^{[k]}(t) < 0$ 且 $\rho_{tc}^{[k]} > 0$, 由引理4可得 $\beta_{tc}^{[k+1]} < 0$, 所以下一步调整中, 样本 t 不会被反向移出. 证毕.

定理3 当 S_1 集合移出样本 t 后, 下一步调整中, 样本 t 不会被反向移入.

证 假设前面一次调整记为 k 且 $S_1^{[k]}$ 中有 n 个样本, 则下一次调整记为 $k+1$, $S_1^{[k+1]}$ 中有 $n-1$ 个样本, 那么有

$$\begin{aligned} \beta_{*c}^{[k+1]} &= -\tilde{R}^{[k+1]} \cdot \tilde{Q}_{1(S_1^{[k+1]c})} = \\ &= -(\tilde{R}_{\setminus tt}^{[k]} - (\tilde{R}_{(*)}^{[k]} \tilde{R}_{(t*)}^{[k]})_{\setminus tt} / \tilde{R}_{tt}^{[k]}) \cdot \tilde{Q}_{1(S_1^{[k+1]c})} = \\ &= (\beta_{*c}^{[k]})_{\setminus t} - (\tilde{R}_{*t}^{[k]})_{\setminus t} \cdot \tilde{Q}_{1(tc)} + \\ &= (-\tilde{R}_{*t}^{[k]})_{\setminus t} \cdot \beta_{tc}^{[k]} + (\tilde{R}_{*t}^{[k]})_{\setminus t} \cdot \tilde{R}_{tt}^{[k]} \tilde{Q}_{1(tc)} / \tilde{R}_{tt}^{[k]} = \\ &= (\beta_{*c}^{[k]})_{\setminus t} - (\tilde{R}_{*t}^{[k]})_{\setminus t} \cdot \beta_{tc}^{[k]} / \tilde{R}_{tt}^{[k]}. \end{aligned}$$

于是有

$$\begin{aligned} \rho_{tc}^{[k+1]} &= \tilde{Q}_{1(tc)} + \tilde{Q}_{1(tS_1^{[k+1]})} \beta_{*c}^{[k+1]} = \\ &= \tilde{Q}_{1(tc)} + \tilde{Q}_{1(tS_1^{[k+1]})} \cdot ((\beta_{*c}^{[k]})_{\setminus t} - (\tilde{R}_{*t}^{[k]})_{\setminus t} \cdot \beta_{tc}^{[k]} / \tilde{R}_{tt}^{[k]}) = \\ &= \rho_{tc}^{[k]} - (\tilde{Q}_{1(tt)} \tilde{R}_{tt}^{[k]} + \beta_{tc}^{[k]} \tilde{Q}_{1(tS_1^{[k+1]})} (\tilde{R}_{*t}^{[k]})_{\setminus t}) / \tilde{R}_{tt}^{[k]} = \\ &= -\beta_{tc}^{[k]} / \tilde{R}_{tt}^{[k]}, \end{aligned}$$

其中 $\rho_{tc}^{[k]} = 0$.

由式(18)可知 $\tilde{Q}_{1(tt)} \tilde{R}_{tt}^{[k]} + \tilde{Q}_{1(tS_1^{[k+1]})} (\tilde{R}_{*t}^{[k]})_{\setminus t} = 1$. 同定理2的分析, 可知: 当 S_1 集合移出样本 t 后, 下一步调整中, 样本 t 不会被反向移入. 证毕.

5.2 有限收敛性分析

有限收敛性分析将确保本文算法在有限次数调整后收敛到问题的最优解. 首先, 证明在迭代过程中, 当 α_{F_1} 在 $[0, c_1]$ 区间内调整时, 目标函数 D_1 单调递减(定理4). 然后, 以此为基础, 证明经过有限步迭代后, AIETSVR算法将收敛到最优解(定理5).

引理6 在迭代时, $\forall i \in S_1, \beta_{ic} \neq \pm\infty$, 并且 $\forall i \in R_1 \cup E_1, \rho_{ic} \neq \pm\infty$.

证

$$\beta_{ic} = -\frac{(\sum_{j \in S_1} (-1)^{i+j} \det(\tilde{Q}_{1(S_1 S_1) \setminus ji}) \tilde{Q}_{1(jc)})}{\det(\tilde{Q}_{1(S_1 S_1)})}.$$

假设存在一个样本 $i \in S_1$, 使得 $\beta_{ic} \neq \pm\infty$, 那么必然有 $\det(\tilde{Q}_{1(S_1 S_1)}) = 0$, 这与定理1相矛盾. 所以 $\forall i \in S_1, \beta_{ic} \neq \pm\infty$; 并根据式(17), 也有 $\forall i \in R_1 \cup E_1, \rho_{ic} \neq \pm\infty$. 证毕.

定理4 当 α_{F_1} 在 $[0, c_1]$ 区间内调整时, 目标函数 D_1 单调递减.

证 假设前一次调整记为 k , 则下一次调整记为 $k+1$, 并令 $\hat{\beta}_{R_1}^{[k]} = 0, \hat{\beta}_{E_1}^{[k]} = 0, \hat{\beta}_{F_1}^{[k]} = d - \alpha_{F_1}, \hat{\rho}_{S_1}^{[k]} = 0$, 那么有

$$\begin{aligned} D_1^{[k+1]} &= \\ &= \frac{1}{2} (\alpha^{[k]} + \hat{\beta}^{[k]} \eta)^T \tilde{Q}_1 (\alpha^{[k]} + \hat{\beta}^{[k]} \eta) + \\ &= (e^T \varepsilon_1 + Y^T - Y^T \tilde{Q}_1) (\alpha^{[k]} + \hat{\beta}^{[k]} \eta) = \\ &= D_1^{[k]} + \frac{1}{2} (\hat{\beta}^{[k]} \eta)^T \tilde{Q}_1 \hat{\beta}^{[k]} \eta + (\hat{\beta}^{[k]} \eta)^T \tilde{H}_1^{[k]} = \\ &= D_1^{[k]} + \frac{1}{2} [\hat{\rho}_{S_1}^{[k]} \hat{\rho}_{F_1}^{[k]}]^T \begin{bmatrix} \hat{\beta}_{S_1}^{[k]} \\ \hat{\beta}_{F_1}^{[k]} \end{bmatrix} \eta^2 + \\ &= (\hat{\beta}_{F_1}^{[k]})^T \tilde{H}_{1(F_1)}^{[k]} \eta = \\ &= D_1^{[k]} + \frac{1}{2} (\hat{\rho}_{F_1}^{[k]})^T \hat{\beta}_{F_1}^{[k]} \eta^2 + (\hat{\beta}_{F_1}^{[k]})^T \tilde{H}_{1(F_1)}^{[k]} \eta = \\ &= D_1^{[k]} + (\frac{1}{2} \hat{\rho}_{F_1}^{[k]} \eta + \tilde{H}_{1(F_1)}^{[k]})^T \hat{\beta}_{F_1}^{[k]} \eta = \\ &= D_1^{[k]} + (\frac{1}{2} \rho_{F_1*}^{[k]} \Delta \alpha_{F_1} + \tilde{H}_{1(F_1)}^{[k]})^T \Delta \alpha_{F_1}. \end{aligned}$$

即只需要证明: $\forall(\mathbf{x}_c, y_c) \in F_1$,

当 $\Delta\alpha_c > 0$, $\Delta\alpha_c \in (0, -2\tilde{h}_1^{[k]}(c)/\rho_{cc}^{[k]})$;

当 $\Delta\alpha_c < 0$, $\Delta\alpha_c \in (-2\tilde{h}_1^{[k]}(c)/\rho_{cc}^{[k]}, 0)$;

当 $\alpha_{F_1} \in [0, c_1]$, 如果 $\tilde{h}_1^{[k]}(c) > 0$, $\Delta\alpha_c \in [-\tilde{h}_1^{[k]}(c)/\rho_{cc}^{[k]}, 0)$; 如果 $\tilde{h}_1^{[k]}(c) < 0$, $\Delta\alpha_c \in (0, -\tilde{h}_1^{[k]}(c)/\rho_{cc}^{[k]})$.

而选取的步长为搜索空间中的最小步长, 因此 $\Delta\alpha_{F_1}$ 满足上述要求, 即 $D_1^{[k+1]} - D_1^{[k]} < 0$. 证毕.

定理 5 经过有限步迭代后, AIETSVR 算法将收敛于最优解.

证 令 $(\alpha_c^{[k]})$ 和 $(D_1^{[k]})$ 为迭代中生成的 2 个序列. 首先, 需要证明 $(D_1^{[k]})$ 为有限序列. 当 α_c 在可行域内调整即 $\alpha_c \in [0, c_1]$ 时, 由定理 4 可知, $(D_1^{[k]})$ 必定是无限的单调递减序列. 设 D_1^{opt} 是式 (3) 的最优解, 容易验证当 $k \rightarrow \infty$ 时, $D_1^{[k]} \geq D_1^{\text{opt}}$, 即 $(D_1^{[k]})$ 是收敛的, 显然这与 $D_1^{[k+1]} < D_1^{[k]}$ 相互矛盾, 故 $(D_1^{[k]})$ 是有限长序列; 当 α_c 在可行域外调整即 $\alpha_c^{[k]} \in (-\infty, 0)$ 或 $\alpha_c^{[k]} \in (c_1, +\infty)$ 时, 如果 $\alpha_c^{[k]} \in (-\infty, 0)$, 必有 $\alpha_c^{[k]} < \alpha_c^{[k+1]} < 0$, 因此 $(\alpha_c^{[k]})$ 为有限单调递增序列, 如果 $\alpha_c^{[k]} \in (c_1, +\infty)$, 同理可得 $(\alpha_c^{[k]})$ 为有限单调递减序列, 故 $(D_1^{[k]})$ 是有限长序列. 综上可得, 在任意情况下, $(D_1^{[k]})$ 都是有限长序列, 表示为 $(D_1^{[1]}, D_1^{[2]}, \dots, D_1^{[m]})$.

假设 $(D_1^{[k]})$ 不能收敛到最优解, 那么由强对偶理论可得, 第 m 次迭代将不满足其 KKT 条件, 显然这与增量调整终止在第 m 次相互矛盾. 因此, 经过有限次数调整, $(D_1^{[1]}, D_1^{[2]}, \dots, D_1^{[m]})$ 必将收敛到最优解.

证毕.

6 数值实验与分析

6.1 实验设计和参数设置

为了验证本文算法的可行性和有效性, 从以下两个角度出发设计实验: 1) 将目前常用的 ETSVR 的求解算法与 AIETSVR 在训练速度和精度上进行对比; 2) AIETSVR 是否能够延续 ETSVR 相对于 SVR 的优势, 比 AOSVR 的训练速度更快.

选取 AOSVR^[13], ETSVRQP^[9], ETSVRSOR^[9] 和本文算法 AIETSVR, 在基准测试数据集上进行仿真实验. 其中, ETSVRQP 使用 MATLAB 二次规划工具箱中的内点法求解 ETSVR 的对偶问题, 能够获得精确解, 且不需要前一时刻的模型信息, 属于离线批处理训练算法, 但是时间复杂度较高; ETSVRSOR 使用逐次超松弛法求解 ETSVR 的对偶问题, 时间复杂度较低, 且不需要前一时刻的模型信息, 也属于离线批处理训练算法, 但是只能求得近似解, 代码可以从 <http://www.optimal-group.org/Resources/Code/ETSVR.html> 下载; AOSVR 使用文献 [13] 中提出的在线增量算法求得精确解, 时间复杂度低, 需要保存前一时刻的模型信息, 代码可以从 <https://github.com/fp2556/online-svr> 下载. 所有实验在 Intel i5-8300H (@2.3 GHz) 处理器,

8 G 内存的计算机, MATLAB 2018a 软件平台上完成.

实验中使用的 UCI 基准测试数据集如表 1 所示.

表 1 实验中使用的基准测试数据集

Table 1 The benchmark datasets used in our experiments

数据集	特征数	训练集 样本数	测试集 样本数
Triazines	60	150	36
Residential Building	105	300	72
Concrete CS	8	800	230
Cpusmall	12	5000	3192
UJIndoorLoc	520	8000	13048
RLCTSA	385	10000	42500

首先把每组数据集的特征归一化为 $[0, 1]$, 然后划分训练集和测试集, 由于训练集中含有重复样本会导致多个样本同时到达边界, 不符合每次仅有一个样本到达或离开边界的假设^[16], 因此重复样本不选入训练集中. 在训练集上, 采用 5 次五折交叉验证的方式, 共 25 次实验的平均值进行网格参数寻优. 在参数寻优时, SVR 使用 LibSVM 提供的离线算法进行训练, ETSVR 使用 ETSVRQP 进行离线训练. 最终以训练集上的最优模型在测试集上的表现来评估该模型的好坏. 采用如下性能评价指标:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}, \quad (29)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \quad (30)$$

其中: RMSE 为均方根误差, MAE 为平均绝对误差, $\hat{y}_i$ 为第 i 个样本的预测值, y_i 为第 i 个样本的实际输出值, N 为训练样本的个数.

此外, 平均新增一个样本所需的训练时间也作为性能评价指标. 并且列出增量算法需要调整的拉格朗日乘子的平均数量 τ , 表明本文算法通过第 3.2 节提前调整后获得了有效的初始状态. 对于 AOSVR, 由文献 [13] 可知 τ 始终为 1; 对于本文算法, $\tau = l_{F_1} + l_{F_2}$; 由于离线算法 ETSVRSOR 和 ETSVRQP 每一次迭代都是对拉格朗日乘子的整体调整, 并不区分拉格朗日乘子是否异常, 因此 τ 无意义.

AOSVR 的参数设置为 $\epsilon = 0.01$, $C = 2^i$, 在 $i \in [-8, 8]$ 的范围内寻找最优值, ETSVRQP, ETSVRSOR 和 AIETSVR 共有的参数 $c_1 = c_2 = 2^i$, $c_3 = c_4 = 2^i$, 在 $i \in [-8, 8]$ 的范围内寻找最优值, 考虑到不敏感因子 ϵ_1 , ϵ_2 对算法的影响较小^[23-25], 并为了对比公平起见, 取 $\epsilon_1 = \epsilon_2 = 0.01$. 由文献 [9] 可知, ETSVRSOR 需要设置参数 $t = 0.9$ 和停止迭代的阈值为 0.001.

6.2 实验结果与分析

表2列出了4种不同的回归算法在基准数据集上的实验结果,“-”表示该处指标无意义,粗体表示最优指标.从表2可以看出,与AOSVR相比,其他3种算法的泛化性能更好,即ETSVR模型在泛化性能上要优于SVR模型,这与文献[9]中的结论相同.由于ETSVR-QP使用MATLAB的二次规划工具箱进行训练,因此

可以把ETSVRQP的解作为ETSVR模型的精确解.对于ETSVR的3种训练算法ETSVRQP,ETSVRSOR和AIETSVR,AIETSVR与ETSVRQP在RMSE和MAE指标上的结果完全相同,但是ETSVRSOR却有差异.因此本文算法能够获得ETSVR模型的精确解,而ETSVRSOR由于收敛性差且梯度信息不准确最终得到的只是近似解.

表2 4种不同回归算法在基准数据集上的实验结果

Table 2 The experimental results of four different algorithms on benchmark datasets

数据集	算法	最优参数(c_1, c_3, C)	RMSE	MAE	CPU time/s	τ
Triazines	AOSVR	$-, -, 2^{-6}$	0.1845	0.1144	5.7901×10^{-3}	1
	ETSVRQP	$2^{-4}, 2^1, -$	0.1654	0.1167	7.8283×10^{-3}	—
	ETSVRSOR	$2^{-4}, 2^1, -$	0.1659	0.1180	1.3012×10^{-3}	—
	AIETSVR	$2^{-4}, 2^1, -$	0.1654	0.1167	7.0055×10^{-3}	2
Residential Building	AOSVR	$-, -, 2^8$	26.9813	16.7909	1.9211×10^{-2}	1
	ETSVRQP	$2^7, 2^{-5}, -$	24.6260	15.9980	2.0234×10^{-2}	—
	ETSVRSOR	$2^7, 2^{-5}, -$	26.5846	17.0691	1.3764×10^{-2}	—
	AIETSVR	$2^7, 2^{-5}, -$	24.6260	15.9980	7.6123×10^{-3}	2
Concrete CS	AOSVR	$-, -, 2^3$	10.7604	8.6646	2.0311×10^{-2}	1
	ETSVRQP	$2^4, 2^0, -$	10.6349	8.6963	6.9912×10^{-2}	—
	ETSVRSOR	$2^4, 2^0, -$	10.6699	8.7338	2.9356×10^{-2}	—
	AIETSVR	$2^4, 2^0, -$	10.6349	8.6963	7.2862×10^{-3}	2
Cpusmall	AOSVR	$-, -, 2^7$	13.7154	4.3178	1.7357×10^{-1}	1
	ETSVRQP	$2^1, 2^{-2}, -$	10.2393	5.9110	5.0735×10^0	—
	ETSVRSOR	$2^1, 2^{-2}, -$	10.1864	6.1493	1.2187×10^0	—
	AIETSVR	$2^1, 2^{-2}, -$	10.2393	5.9110	1.0712×10^{-2}	2
UJIndoor Loc	AOSVR	$-, -, 2^{-2}$	34.5985	25.4870	2.9227×10^{-1}	1
	ETSVRQP	$2^2, 2^1, -$	33.3310	24.6792	2.0884×10^1	—
	ETSVRSOR	$2^2, 2^1, -$	33.3569	24.7539	3.2219×10^0	—
	AIETSVR	$2^2, 2^1, -$	33.3310	24.6792	1.0272×10^{-1}	2
RLCTSAA	AOSVR	$-, -, 2^2$	9.0352	5.8852	2.6558×10^0	1
	ETSVRQP	$2^3, 2^1, -$	8.3974	6.1113	4.4695×10^1	—
	ETSVRSOR	$2^3, 2^1, -$	8.3906	6.2048	8.7383×10^0	—
	AIETSVR	$2^3, 2^1, -$	8.3974	6.1113	1.7764×10^{-1}	2

对于平均单个样本的增量时间,AIETSVR除了在第1个规模最小的数据集上不是最优外,在其余数据集上均表现出很大的优势.结合第4.2节中的分析可知:在小规模数据集上,ETSVRQP的时间复杂度较高,其他3种算法的时间复杂度相当;在大规模数据集上,ETSVRQP的时间复杂度最高,ETSVRSOR次之,这两种算法是离线训练算法,并不适合处理大规模数据集的在线增量学习问题,而AOSVR和AIETSVR这两种在线增量算法的时间复杂度远低于离线算法,且AIETSVR比AOSVR训练效率更高,适合于大规模数据集的在线增量训练.主要原因是:与AOSVR相比,ETSVR模型中的每个目标函数都只有一个不等约束,

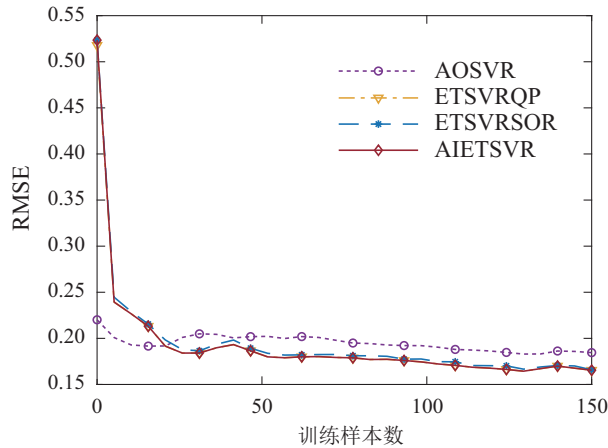
相当于把SVR模型的 S 集合分为了 S_1 集合与 S_2 集合,因此AIETSVR算法迭代时需要更新的2个逆矩阵的大小为AOSVR算法对应逆矩阵的1/4左右,使得AIETSVR训练时间远小于AOSVR.与离线训练算法ETSVRQP和ETSVRSOR相比,AIETSVR基于已有模型信息进行增量,训练时间大为减少,并且在每个数据集上都能得到与ETSVRQP相同的RMSE和MAE指标,即该算法能够获得精确解.

从 τ 指标上可以看出,第3.2节的提前调整有效地减少了异常样本并获得有效的初始状态.由第3节可知,本文算法时间复杂度随着 τ 的增加,成倍增长,而通过提前调整,避免了新增样本后大量异常拉格朗日

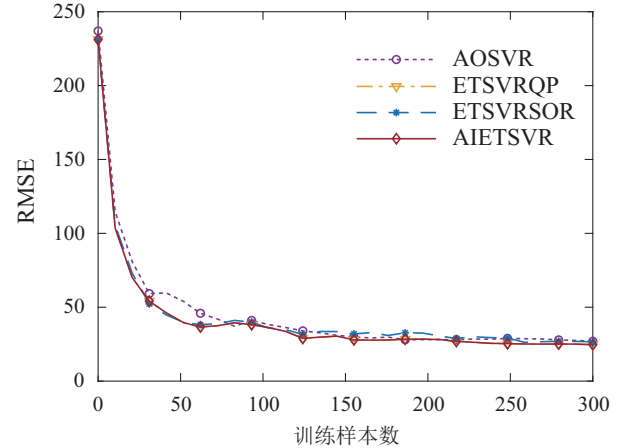
乘子的产生, 使得算法的时间复杂度不会急剧恶化.

图1为随着训练样本个数增加, 4种算法的RMSE性能指标的变化过程. 从图1中可以看出, AIETSVR和ETSVRQP的RMSE变化曲线完全相同, 而ETSVRSOR和ETSVRQP的RMSE变化曲线只是趋势相同, 结果相近. 这就意味着AIETSVR通过增量关系, 在逐步调整过程中, 保证非异常样本的拉格朗日乘子始终

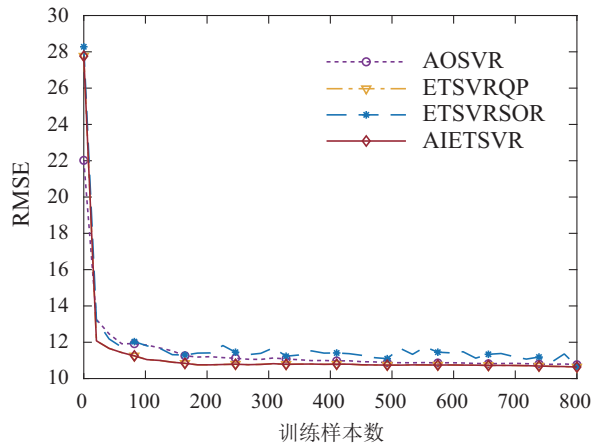
满足KKT条件, 并获得一个十分精确的解. 而ETSVRSOR由于收敛性差且梯度信息不准确, 只能获得一个近似解. 与SVR相比, ETSVR的RMSE在6个数据集上均取得了更好的结果, 说明ETSVR在RMSE上具有更好的泛化性能, 原因是ETSVR目标函数中包含了二次损失项, 相当于对RMSE指标进行了优化.



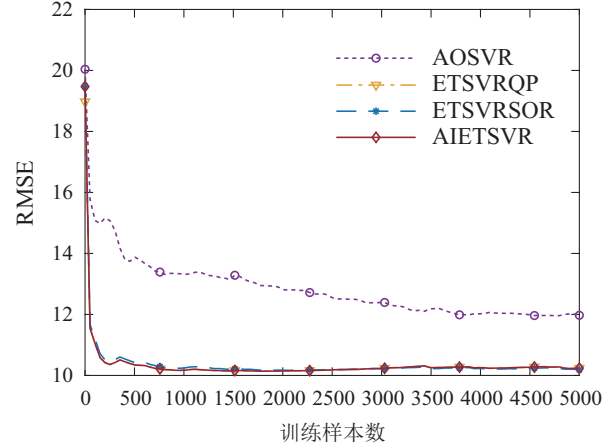
(a) Triazines



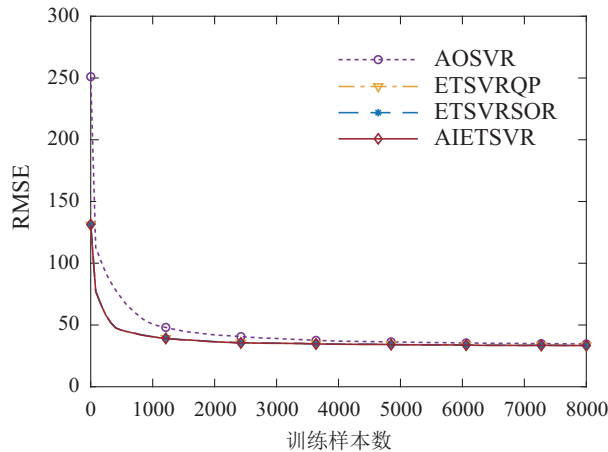
(b) Residential Building



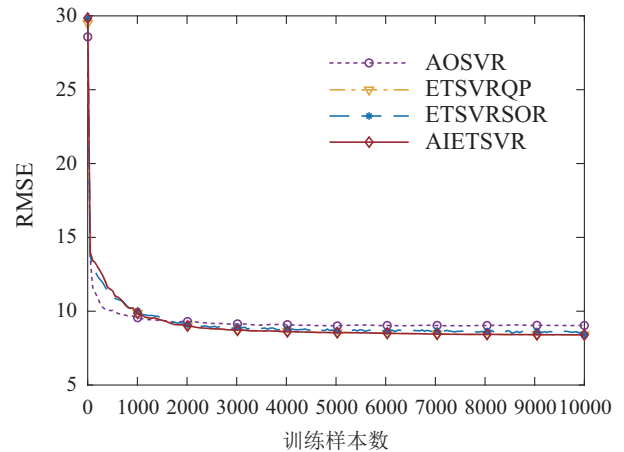
(c) Concrete CS



(d) Cpusmall



(e) UJIndoor Loc



(f) RLCTSA

图1 RMSE随训练样本增加的变化过程

Fig. 1 The changing process of RMSE versus the number of training samples

图2为随着训练样本个数增加,4种算法的MAE性能指标的变化过程.从图2中可以看出,AIETSVR和ETSVRQP的MAE变化曲线同样完全相同,而ETSVRSOR和ETSVRQP的MAE变化曲线只是趋势相同,结果相近.这进一步证实了本文算法能够获得精确解.但是,与SVR相比,ETSVR在MAE上略差.原因是SVR中的 ε 不敏感损失与MAE类似,相当于对MAE的

直接优化,而ETSVR中包含的为单边 ε 不敏感损失,相比而言约束较弱.

图3为随着训练样本个数增加,每种算法新增一个样本所需训练时间的变化过程.从图3中可以看出,除第一个规模最小的数据集外,与其余3种算法相比,AIETSVR在训练时间上具有一定的优势,且随着样本量增加,新增一个样本模型所需的训练时间增长缓慢.

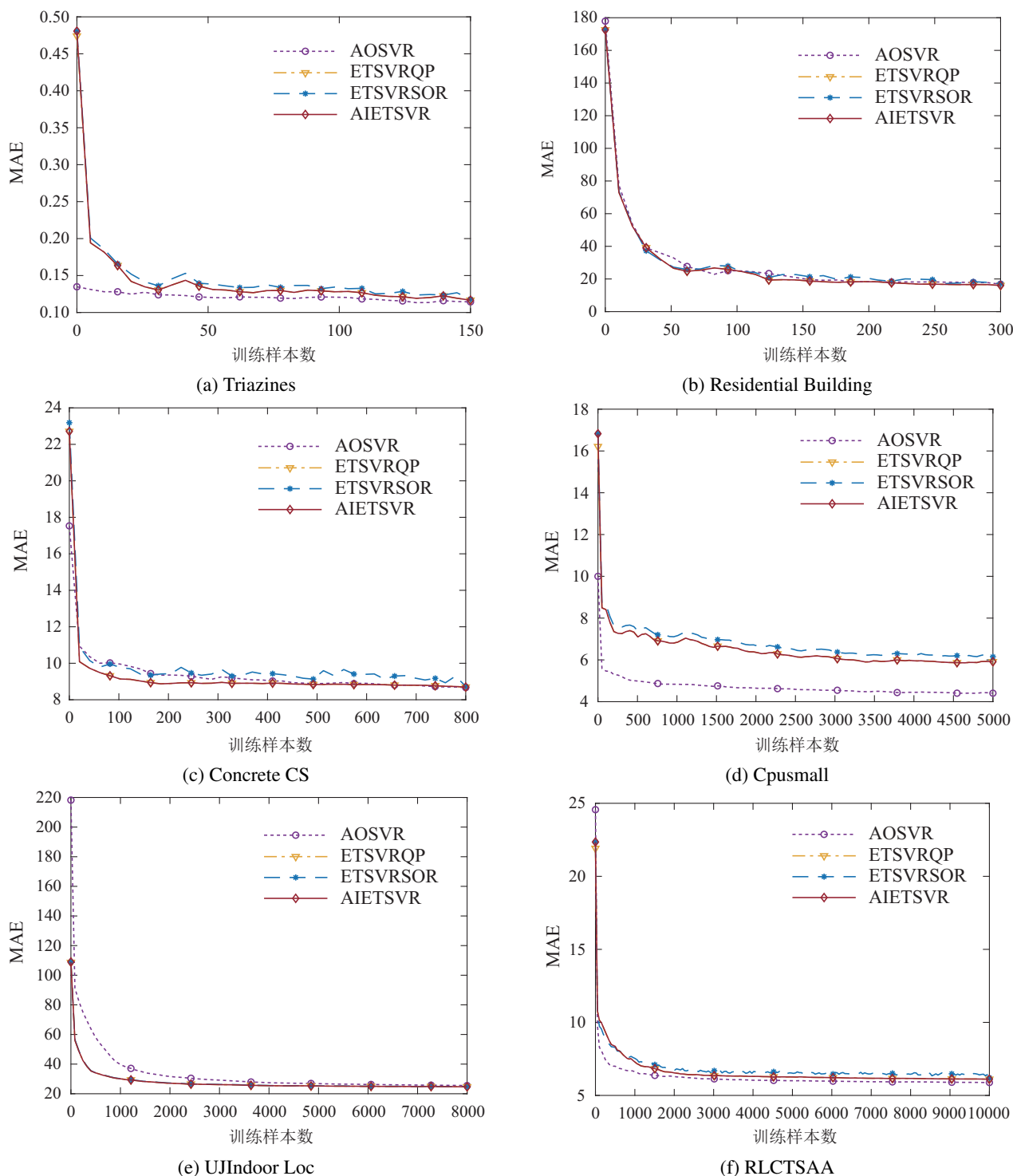


图2 MAE随训练样本增加的变化过程

Fig. 2 The changing process of MAE versus the number of training samples

由第4.2节可知, 在小规模数据集上, ETSVRQP的算法复杂度较高, 其他3种算法复杂度相当, 但是由于实际的影响因素较多, 因此出现了图4(a)和(b)上训练时间增长很不稳定的现象。

结合第4.1节和第4.2节可知, 在大规模数据集上, $m \ll n$, 除 n 外, AIETSVR算法的时间复杂度主要取决于 l_{F_1} , l_{F_2} , l_{S_1} , l_{S_2} . 本文通过提前增量调整使得 $\tau = l_{F_1} + l_{F_2} = 2$, 避免了由于 τ 增加使得时间复杂度急剧

恶化. 又由于ETSVR模型本身的优势, 每个目标函数都只有单个边界, 支持向量都有很好的稀疏性, 即 l_{S_1} 和 l_{S_2} 较小, 迭代过程中需要更新的逆矩阵 R 规模较小, 故每次迭代消耗的时间较少. 而对于小规模数据集, 由于其余算法所求逆矩阵规模均很小, 体现不出本文算法在迭代过程中所需更新的逆矩阵 R 规模小的优势. 故AIETSVR更加适合于处理大规模数据集上增量学习问题。

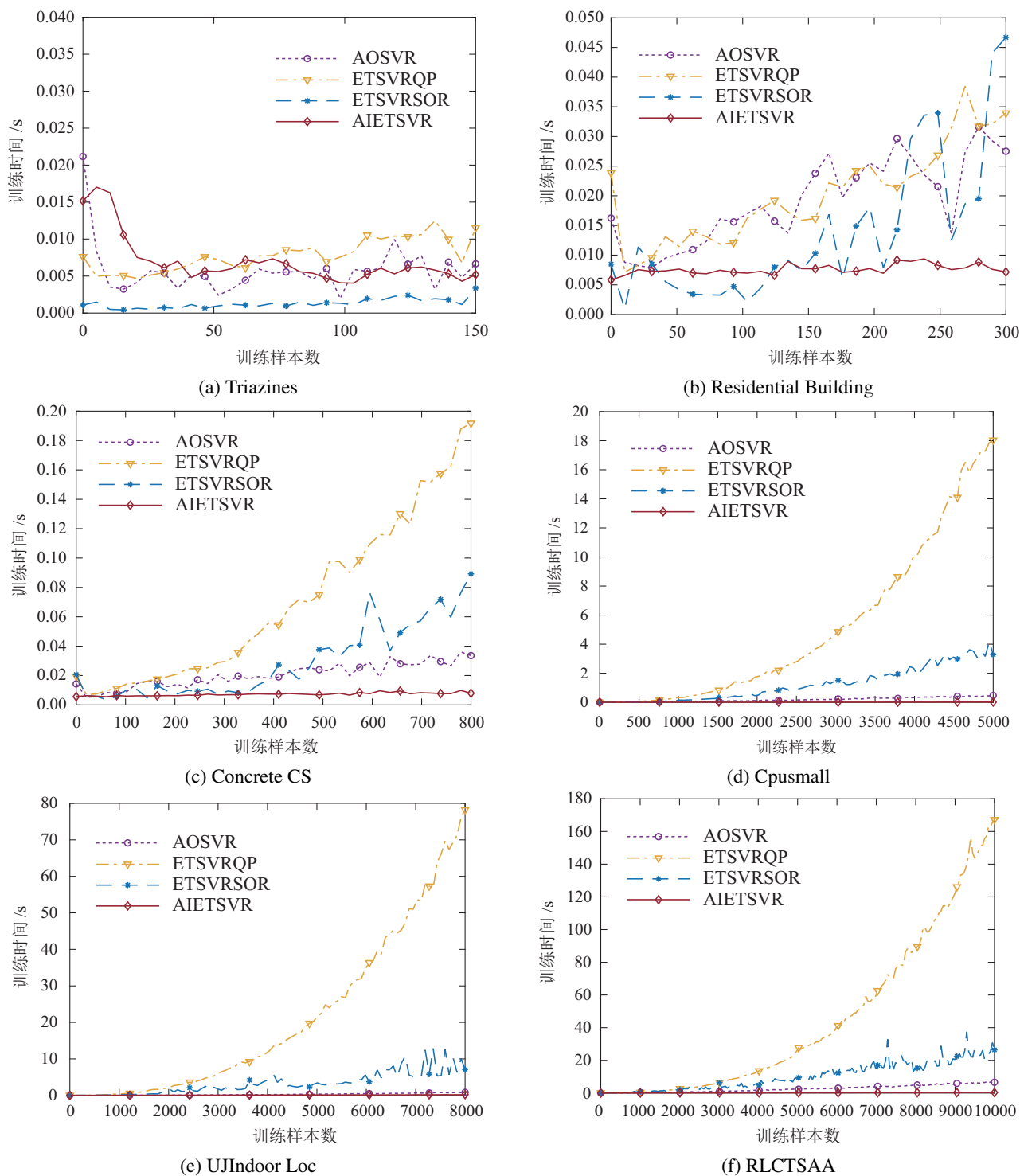


图3 新增一个样本所需训练时间的变化过程

Fig. 3 The changing process of the training time elapsed when adding a new sample

7 总结

本文设计了一种精确增量式 ε 型孪生支持向量回归机算法,解决了增量环境下无法高效更新线性 ε 型孪生支持向量回归机模型的问题.该算法通过计算新增样本的拉格朗日乘子以及调整支持向量集的拉格朗日乘子获得迭代调整前的有效初始状态,然后基于AOSVR的方法,调整异常拉格朗日乘子,并且从理论上分析了算法的可行性和有限收敛性.实验结果表明,在有限步迭代过程中,本文算法仅需更新两个较小规模的逆矩阵,极大地减少了大规模数据集下模型更新的时间,而且能获得精确解.

参考文献:

- [1] VAPNIK V N. *The Nature of Statistical Learning Theory*. New York: Springer Verlag Press, 1999: 3 – 30.
- [2] LILLEBERG J, YUN Z, ZHANG Y. Support vector machines and word2vec for text classification with semantic features. *IEEE International Conference on Cognitive Informatics & Cognitive Computing*. New York: IEEE, 2015: 136 – 140.
- [3] ZHENG B J, MYINT S W, THENKABAIL P S, et al. A support vector machine to identify irrigated crop types using time-series Landsat NDVI data. *International Journal of Applied Earth Observations & Geoinformation*, 2015, 34(1): 103 – 112.
- [4] LI Yi, CAI Tianxun, FAN Jianfeng, et al. SVM-based method for detecting ranking functions in polynomial loop programs. *Journal of Software*, 2019, 30(7): 1903 – 1915.
(李轶, 蔡天训, 樊建峰, 等. 基于SVM的多项式循环程序秩函数生成. 软件学报, 2019, 30(7): 1903 – 1915.)
- [5] COLLOBERT R, BENGIO S. SVM-Torch: support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 2001, 1(2): 143 – 160.
- [6] JAYADEVA, KHEMCHANDANI R, CHANDRA S. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, 29(5): 905 – 910.
- [7] PENG X J. TSVR: An efficient twin support vector machine for regression. *Neural Networks*, 2010, 23(3): 365 – 372.
- [8] SINGH M, CHADHA J, AHUJA P, et al. Reduced twin support vector regression. *Neurocomputing*, 2011, 74(9): 1474 – 1477.
- [9] SHAO Y H, ZHANG C H, YANG Z M, et al. An ε -twin support vector machine for regression. *Neural Computing and Applications*, 2013, 23(1): 175 – 185.
- [10] RASTOGI R, ANAND P, CHANDRA S. A ν -twin support vector machine based regression with automatic accuracy control. *Applied Intelligence*, 2017, 46(3): 1 – 14.
- [11] GU B J, FANG J W, PAN F, et al. Fast clustering-based weighted twin support vector regression. *Soft Computing*, 2020, 24(8): 6101 – 6117.
- [12] CAUWENBERGHS G, POGGIO T. Incremental and decremental support vector machine learning. *International Conference on Neural Information Processing Systems*, Denver. MA: MIT Press, 2000: 388 – 394.
- [13] MA J S, THEILER J, PERKINS S. Accurate online support vector regression. *Neural Computation*, 2003, 15(11): 2683 – 2703.
- [14] KARASUYAMA M, TAKEUCHI I. Multiple incremental decremental learning of support vector machines. *IEEE Transactions on Neural Networks*, 2010, 21(7): 40 – 54.
- [15] GU Binjie, PAN Feng. Accurate incremental online ν -support vector regression learning algorithm. *Control Theory & Applications*, 2016, 33(4): 466 – 478.
(顾斌杰, 潘丰. 精确增量式在线 ν 型支持向量回归机学习算法. 控制理论与应用, 2016, 33(4): 466 – 478.)
- [16] GU Bin, ZHENG Guansheng, WANG Jiandong. Analysis for incremental and decremental standard support vector machine. *Journal of Software*, 2013, 24(7): 1601 – 1613.
(顾彬, 郑关胜, 王建东. 增量和减量式标准支持向量机的分析. 软件学报, 2013, 24(7): 1601 – 1613.)
- [17] YANG L, LIU K, LIANG X D, et al. An incremental learning algorithm for improved least squares twin support vector machine. *IEEE 5th International Conference on Advanced Computational Intelligence (ICACI)*. New York: IEEE, 2012: 463 – 467.
- [18] HAO Yunhe, ZHANG Haofeng. Incremental learning algorithm based on twin support vector regression. *Computer Science*, 2016, 43(2): 230 – 234, 249.
(郝运河, 张浩峰. 基于双支持向量回归机的增量学习算法. 计算机科学, 2016, 43(2): 230 – 234, 249.)
- [19] MELLO A R, STEMMER M R, KOERICH A L. Incremental and decremental fuzzy bounded twin support vector machine. *Information Sciences*, 2020, 526: 20 – 38.
- [20] TYLAVSKY D J, SOHIE G R L. Generalization of the matrix inversion lemma. *Proceedings of the IEEE*, 1986, 74(7): 1050 – 1052.
- [21] GOLUB G H, VAN LOAN C F. Matrix computations. *Mathematical Gazette*, 1996, 47: 392 – 396.
- [22] GERTZ E M, GRIFFIN J D. Using an iterative linear solver in an interior-point method for generating support vector machines. *Computational Optimization and Applications*, 2010, 47: 431 – 453.
- [23] BALASUNDARAM S, TANVEER M. On Lagrangian twin support vector regression. *Neural Computing and Applications*, 2013, 22: 257 – 267.
- [24] TANVEER M, SHUBHAM K, ALDHAIFALLAH M, et al. An efficient regularized k-nearest neighbor-based weighted twin support vector regression. *Knowledge-Based Systems*, 2016, 94: 70 – 87.
- [25] TANVEER M, SHUBHAM K. A regularization on Lagrangian twin support vector regression. *International Journal of Machine Learning and Cybernetics*, 2017, 8(3): 807 – 821.

作者简介:

曹杰 硕士研究生, 目前研究方向为机器学习、模式识别和算法优化, E-mail: 635069140@qq.com.cn;

顾斌杰 博士, 副教授, 目前研究方向为机器学习、发酵过程建模与优化, E-mail: gubinjie1980@126.com;

潘丰 教授, 博士生导师, 目前研究方向为生化过程智能控制、分布式计算机控制系统, E-mail: pan_feng_63@163.com;

熊伟丽 教授, 博士生导师, 目前研究方向为系统辨识、工业过程软测量及优化, E-mail: greenpre@163.com.