

支持向量回归增量学习

张一凡 冯爱民 张正林

(南京航空航天大学计算机科学与技术学院 南京 210016)

摘要 针对支持向量回归因时空复杂度较高而无法处理大规模数据的问题,提出了一个新颖的增量学习模型——L增量 ν 支持向量回归(L Incremental ν Support Vector Regression, LISVR)。该模型针对支持向量丢失所产生的不利影响,通过不断对支持向量样本加权并及时淘汰非支持向量,降低了时空复杂度。从理论上证明了算法可收敛到全局最优解。结合人工数据集、UCI数据集和机场噪声的实际问题对该算法做了相应测试,结果验证了算法的有效性。

关键词 支持向量回归,支持向量,增量学习,机场噪声

中图分类号 TP391.4 **文献标识码** A

Incremental Learning with Support Vector Regression

ZHANG Yi-fan FENG Ai-min ZHANG Zheng-lin

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract In view of costly time-space complexity for support vector regression in dealing with large scale data, this paper presented a novel algorithm named the L Incremental ν Support Vector Regression (LISVR) by means of incremental learning. LISVR eliminates non-support vectors each iteration and then takes the support vectors as the training samples with the weight factor. It reduces time-space complexity and enhances the regression results simultaneously. Theoretically, this paper proved the convergence of the global optimal solution. The experiments on the artificial data sets, UCI data set and airport noises show the effectiveness of the LISVR.

Keywords Support vector regression, Support vectors, Incremental learning, Airport noise

1 引言

由 Vapnik 等人提出的支持向量机(Support Vector Machine)是一个面向有限样本的机器学习方法。该方法很好地解决了模式识别、回归估计和密度估计问题^[9,13],相应的解具有全局最优性、稀疏性,且能利用核函数解决非线性问题,因此具有较好的泛化能力。支持向量机在分类领域的应用,实际上是对分类函数进行估计,若推广到估计实值函数,便成为回归问题。

将支持向量回归应用于大数据集,由于需要解决一个二次规划(Quadratic Programming)问题,相应的时间复杂度和空间复杂度均为 $O(N^3)$, N 为样本个数,因此需要消耗较大的存储空间与运算时间。进一步通过分解技术,如 SVM^{light} 、SMO^[3]、libsvm 等,可将大规模的QP问题分解为较小的子问题从而有效提高训练速度,但由于工作集的选择不得不重新考虑所有样本多次寻找才能最终得到全局最优解,因此并不能本质解决大数据学习问题。

增量学习作为解决大数据学习的重要方法,由于在训练过程中并不一次性需要整个数据集,可降低时空消耗,因此其主要应用于随着时间不断改变的流数据及大数据,如股票交

易数据和 Web 日志记录^[13]等。

这里,根据 SVR 增量学习算法是否保留全部样本,主要分为如下两类:

其一是基于不舍弃样本的算法。J Ma 针对 SVR 提出了精确在线支持向量回归^[8]。该方法是 Cauwenberghs 等提出的精确增量/减量 SVM 算法^[4]的回归版本。通过把样本集分为支持向量集、误差集和剩余集 3 类,可在学习过程中每次增加一个样本,利用 KKT 条件对拉格朗日系数进行矩阵中的迭代计算,最终使新增样本进入 3 类集合之一并且保证达到 KKT 条件,从而得到精确解。P Laskov 的增量学习算法^[23]则根据新增样本初始位置的不同分 4 种情况寻找最佳新样本的拉格朗日乘子变化量,从而使矩阵迭代过程更精确化,并将该算法推广到单类的情况。C P Diehl 的增量 SVM^[24]在学习过程中设置了一个参数 $p \in [0, 1]$,用来表示每个样本的学习程度, $p=0$ 为初始化状态, $p=1$ 为学习完成时的状态,然后使用该参数修改迭代矩阵中的参数。WENJIAN WANG 等则提出一种基于结构风险最小化(structural risk minimization, SRM)构造的增量 SVM^[10],该算法也需要进行复杂的迭代过程。另有一种基于高斯核局部特性的方法^[15],该算法只更新对学习机输出影响最大的拉格朗日系数

到稿日期:2013-07-23 返修日期:2013-12-30 本文受国家自然科学基金(61139002)资助。

张一凡(1988-),男,硕士生,主要研究方向为模式识别,E-mail: zhangyifanf@163.com;冯爱民 女,副教授,硕士生导师,主要研究方向为模式识别、机器学习、系统结构等;张正林 男,硕士生,主要研究方向为模式识别、系统结构。

以减少计算复杂度。上述算法虽大多可得到精确解,但迭代过程均较为复杂,且在计算过程中空间复杂度没有降低。再加之增量学习中每次只能增加一个样本,故在实际应用中存在重要缺陷。

其二则是基于舍弃样本的算法。Syed 等提出了一种增量训练方法^[1]:通过每次将支持向量保留并与新增样本一起训练,可彻底舍弃训练结果中的非支持向量,即每次训练的样本为前一次的支持向量与本次新加入的样本。与标准 SVR 所采用的批处理方法相比,由于每步只需存储支持向量,相应的时空复杂度均有所降低。但由于其未充分考虑新增样本的加入可能导致增量训练过程中出现“震荡”现象,因此会影响算法的收敛速度^[17]。针对上述不足,王媛等提出了交叉验证增量学习算法^[13],通过使用特征空间距离度量从上次训练结果中选择样本保留,可避免将最终的支持向量在增量学习训练过程中误作为非支持向量而被遗漏。但上述计算过程较为复杂,增加了时空复杂度。Mitra 等提出的错误驱动方法^[22](Error-driven technique)则通过每次保留错分样本和新增样本合并训练,不断用上次训练的错分样本修正分类面。由于错分样本仅是部分支持向量从而损失较多的支持向量,因此结果误差较大。

基于上述分析且进一步考虑到大规模数据应用的需要,虽然不舍弃样本的方法大多可得到精确解,但由于其对于空间的消耗并无本质改善,因此本文是基于舍弃样本方法来展开的。

在 Syed 提出的 SVM 简单增量学习算法中,由于舍弃一部分之前训练中的样本而大大忽视了之前训练样本的作用,即默认增加了下一次增量学习过程中样本的作用,使得回归曲线会偏向于后来增量样本那一方。在实际应用中这会导致得到的训练结果和真实的结果之间产生误差。为解决上述问题,这里我们对上一次训练所留下的支持向量添加修正因子 L ,而后与增量样本合并训练,称之为 L 增量 ν 支持向量回归 (L Incremental ν Support Vector Regression, LISVR) 算法。

本文第 2 节简介 ν -SVR 算法并讨论基于 ν -SVR 的 L 增量学习算法,然后证明方法的收敛性;第 3 节在人工数据集、真实数据集和机场噪声数据上对所提出的方法的推广性进行了对比试验;最后为总结与展望。

2 基于增量学习的支持向量回归算法

在标准 SVR 中,存在着参数“管道 (tube)^[9]” ϵ 。这个参数的选取需要利用先验知识,使用较为困难,而 ν -SVR^[5] 算法则不存在选择 ϵ 的问题,且在 ν -SVR 中,参数 ν 表示边界支持向量和支持向量数量的上下限的比例,而我们的算法中每次增量训练都需要保存支持向量,因此使用 ν -SVR 来构建增量学习的模型。

由于本文的工作是在 ν 支持向量回归上展开的,我们先介绍 ν 支持向量回归算法。

2.1 ν 支持向量回归

在 2000 年, Scholkoph 和 Smola 提出的 ν -SVR^[5] 方法中,引入反映超出 ϵ 管道之外的样本数据点 (即边界支持向量数

量)和支持向量数的新参数 ν ,该方法简化了 SVR 的参数调节。

在 ν -SVR 中,优化的目标函数为:

$$\min \tau(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C(\nu\epsilon + \frac{1}{L} \sum_{i=1}^L (\xi_i + \xi_i^*)) \quad (1)$$

$$\begin{aligned} ((w, \phi(x_i)) + b) - y_i &\leq \epsilon + \xi_i^* \\ \text{s. t. } y_i - ((w, \phi(x_i)) + b) &\leq \epsilon + \xi_i \\ \xi_i, \xi_i^*, \epsilon &\geq 0 \end{aligned} \quad (2)$$

其相对应的对偶优化问题为:

$$\max W(\alpha, \alpha^*) = \sum_{i=1}^L (\alpha_i - \alpha_i^*) y_i - \frac{1}{2} \sum_{i,j=1}^L (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(x_i, x_j) \quad (3)$$

$$\begin{aligned} \sum_{i=1}^L (\alpha_i - \alpha_i^*) &= 0 \\ \text{s. t. } \sum_{i=1}^L (\alpha_i + \alpha_i^*) &\leq C\nu \\ \alpha_i, \alpha_i^* &\in [0, \frac{C}{L}] \end{aligned} \quad (4)$$

回归估计的形式为:

$$f(x) = \sum_{i=1}^L (\alpha_i - \alpha_i^*) k(x, x_i) + b \quad (5)$$

其中, α_i, α_i^* 为拉格朗日乘子, b 为偏差项。 $k(x_i, x_j)$ 表示核函数,若无特殊说明,本文均使用径向基 (Radial Basis Function, RBF) 核函数^[17]。

支持向量回归中,有非零拉格朗日乘子的样本称为支持向量。对于边界支持向量 (Bound Support Vector, BSV),有 $\alpha_i = \frac{C}{L}, \alpha_i^* = 0$ 或者 $\alpha_i = 0, \alpha_i^* = \frac{C}{L}$ 。由式(4),对 N_{BSV} 个边界支持向量,有 $N_{BSV} \frac{C}{L} \leq \sum_{i=1}^L (\alpha_i + \alpha_i^*) = C\nu$,即 $\nu \geq \frac{N_{BSV}}{L}$ 。同样对于 N_{SV} 个支持向量,最后我们可以得到 $\frac{N_{BSV}}{L} \leq \nu \leq \frac{N_{SV}}{L}$ 。

由以上分析可知, ν 表示边界支持向量数量上限和支持向量数量的下限^[5]。训练一个问题的最优常数 C 和核参数 σ 一般是通过一些模型选择技术来寻找的,例如交叉验证。

2.2 基于增量学习的 ν 支持向量回归算法

支持向量机的一个重要的特性就是:仅仅只用支持向量训练得到的结果和用整个训练集得到的结果是一致的,且 Syed 等人用实验验证了支持向量的丢失会对学习结果造成不良影响^[1]。

但 Syed 等提出的增量方法和真实结果往往偏差较大。这是因为在增量训练过程中,支持向量在样本集中的分布是未知的,即支持向量只是能够充分描述决策边界,而不是那些训练样本^[6]。但在整个训练集中,每个样本都是等价的,不应该更加重视支持向量而忽视非支持向量。

使用 Syed 提出的方法做如下实验。观察图 1, 10 个样本点由 $y=x$ 产生,每个样本点添加了 0 均值、1 方差的正态分布扰动。将样本分为初始集和增量集两部分,每部分各 5 个样本。从图中实验结果可以看出,在增量训练后(右图),回归曲线会偏向于增量样本集。显然,两次训练中初始样本集的支持向量对回归结果的影响不如增量样本集。而进行多次增量学习时,这种差异会越来越明显。图中“SV”表示使用初始样本训练之后得到的支持向量样本;回归曲线 1 表示用初始样本得到的回归曲线;回归曲线 2 表示第一次加入增量样本

之后得到的回归曲线。

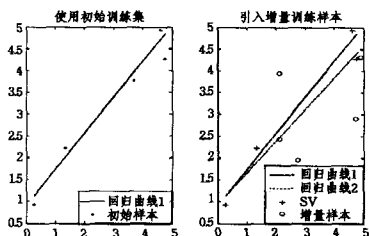


图1 初始训练集合引入增量训练样本对比

以上问题出现的关键在于,在增量学习时,由于会舍弃一部分之前训练中的样本,这就相当于大大忽视了之前训练样本的作用而默认增加了下一次增量学习过程中样本的作用,使得回归曲线会偏向于后来增量样本那一方。在实际应用中会导致得到的训练结果与真实的结果之间产生误差。

由以上分析可以得出,简单 SVM 增量算法遇到了一个问题,即支持向量仅描述了决策函数边界而未描述整个数据集。经过上述分析可知,为了使增量学习能更快收敛,结果更加精确,我们对旧的支持向量给出一个更大的惩罚并因此提出了相应的改进算法——基于 ν -SVR 的 L 增量学习算法(L Incremental ν Support Vector Regression, LISVR),下文简称为 LISVR。

在构建的基于 ν -SVR 增量学习算法中, $(x_i, y_i)_{i \in S}$ 表示旧支持向量, $(x_i, y_i)_{i \in I}$ 表示新样本。所构建的 L 增量 ν -SVR 学习算法表示为:

$$\min \tau(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C(\nu\epsilon + \frac{1}{l} (\sum_{i \in I} (\xi_i + \xi_i^*) + L \sum_{i \in S} (\xi_i + \xi_i^*))) \quad (6)$$

对偶化可得:

$$\max W(\alpha, \alpha^*) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(x_i, x_j) \quad (7)$$

$$\text{s. t. } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0$$

$$\sum_{i=1}^l (\alpha_i + \alpha_i^*) \leq C\nu \quad (8)$$

$$\alpha_{i \in I}, \alpha_{i \in I}^* \in [0, \frac{C}{l}]$$

$$\alpha_{i \in S}, \alpha_{i \in S}^* \in [0, \frac{LC}{l}]$$

这个新的增量学习支持向量回归方法可看作是定义了一个新的损失函数或者是给旧样本一个更高的惩罚。

对于 L, 我们作如下选择: $L = \frac{\# \text{examples}}{\# \text{SVs}}$, 即上一次训练的样本数量和支持向量数量之间的比值。之所以如此的根本原因在于之前训练得到的每一个支持向量都代表着之前样本的一部分, 我们想通过该办法让支持向量能够在一定程度上表示出之前的训练样本, 从而使增量学习结果更快收敛。

2.3 收敛性证明

使用 L 增量 SVR 的主要问题在于, 能否得到一个全局最优解。在实验中, 我们将一个大数据集划分成多个子集, 对子集进行增量式学习。每一次增量学习过程中, 我们获取其中的支持向量, 将支持向量和下一次的样本集合并作为新的训

练集。下面将证明这个方法最终可以在有限步骤内收敛到全局最优解。

设 Ω 为整个样本集, S 为一样本集且 $S \subseteq \Omega$, $W(S)$ 是关于 S 的对偶优化函数式(3)的极大值。对于符号 $S\nu(S) \subseteq S$, $S\nu(S)$ 表示 S 的支持向量, 则有 $W(S) = W(S\nu(S)) \leq W(\Omega)$, $\forall S \subseteq \Omega$ 。

设 F 为 Ω 子集的集合, $S^* \in F$, 则有 $W(F) = \max_{S^* \in F} W(S^*) \leq W(\Omega)$ 。

定理 1^[21] 设 F 和 G 均为 Ω 子集的集合, $T \subseteq G$, $S_F^* \in F$, 并且满足 $S\nu(S_F^*) \subseteq T$, 则有 $W(G) \geq W(F)$ 。

定理 2^[21] 设 F 和 G 均为 Ω 子集的集合, $S_F^* \in F$, $\forall T \subseteq G$, 满足 $S\nu(S_F^*) \subseteq T$ 。若 $W(G) = W(F)$, 那么有 $W(S_F^*) = W(\cup_{T \in G} T)$ 。

定义支持向量回归增量学习是一个 $\{F_t\}$ 序列, 满足:

1) $\forall t > 1, \exists T \in F$, 使得 $S\nu(S_F^*) \subseteq T$;

2) $\exists k > t, \forall T \in F_k$, 可得 $S\nu(S_F^*) \subseteq T$ 和 $\cup(F_k) = \Omega$ 。

定理 3 SVR 增量学习序列 $\{F_t\}$ 在有限步骤内收敛到整个样本集 Ω 的 SVR 解, 即 $\exists t^*, \forall t > t^*, W(F_t) = W(\Omega)$ 。

3 实验及结果分析

为了测试 L 增量算法的性能, 先在 Sin 数据集上做实验, 然后再将该模型在真实数据集中实验对比, 并对参数 L 进行验证分析。本文的评价指标为均方误差 (Mean Square Error, MSE), $MSE = \frac{1}{l} \sum_{i=1}^l (f(x_i) - y_i)^2$ 。将经典的批处理 SVR、ISVR、AOSVR、LISVR 作 MSE 和时间消耗的实验对比。其中, ISVR 是指 Syed 的增量算法, AOSVR 是指 J Ma 的经典精确在线 SVR, 本文推荐的算法称为 LISVR。

3.1 人工数据集

首先使用 Sin 函数人工数据集。给出 l 个样本 (x_i, y_i) , 其中 $x_i \in [-3, 3]$, $y_i = \sin(\pi x_i) / (\pi x_i) + \nu_i$ 。 ν_i 服从高斯分布。本文使用 RBF 核函数 $k(x, x') = \exp(-\frac{(x-x')^2}{2\sigma^2})$ 。

这里随机生成 1000 个 0 均值、1 方差的高斯分布数据, 即 $\frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$, 并作十折交叉验证。通过参数寻优, 此处取 $\nu = 0.2, \sigma = \sqrt{0.5}$, 并取不同的惩罚因子 C 对 MSE 进行对比, 如图 2 所示。

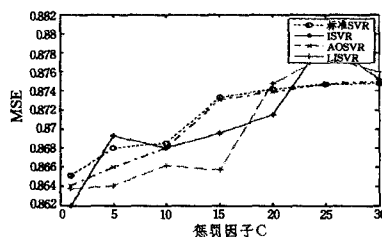


图2 标准 SVR/ISVR/LISVR/AOSVR 关于 MSE 的对比

从图 2 可看出, LISVR 算法在多数时候比增量 SVR 算法的 MSE 小, 但并不显著。产生该结果的原因是支持向量回归属于一个适合于小样本理论的机器学习方法, 当较少的样本已经能够得到一个很好的学习结论时, 学习机很难从之前

的学习过程获取对下一步增量学习有效的训练样本,因而使用 LISVR 算法并没有得到一个理想结果。

现在使用之前 1000 个数据的 Sin 函数样本集。另外添加 200 个高斯噪声。实验结果如图 3 所示。

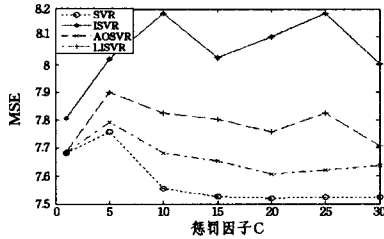


图 3 添加噪声后 SVR/ISVR/AOSVR/LISVR 关于 MSE 的对比

由图 3 可看出,当样本中添加了噪声时,LISVR 所得到的结果比 ISVR 好很多。对该结果的合理解释是:对于某问题,经过一定次数的增量学习后,新增样本含有的有用信息越来越少,而最终达到收敛。由于增量学习过程中我们不断选择样本并对旧的支持向量增加权重,因此在存在噪声的情况下能得出一个较好的解。而 AOSVR 比 LISVR 好,是因为 AOSVR 在训练过程中保留的有效信息较多。增量学习就是不断寻找样本中最富信息(most informative)样本点的过程。

3.2 真实数据集

首先在 UCI 数据集上测试,然后将本文的模型应用到机场噪声中。

3.2.1 UCI 数据集

首先在 UCI 标准数据集 Boston housing(<http://archive.ics.uci.edu/ml/datasets/Housing>)中测试算法性能。该数据集有 13 个属性,总共有 506 个样本。AOSVR 在增量过程中每次只能增加一个数据。实验中,我们使用 481 个数据进行训练,25 个作为测试。

选取惩罚因子 C 为 10, σ 为 4, $\nu = 0.18$ 。可以得到 MSE 的对比结果,如表 1 所列。

表 1 Boston housing 的测试结果

	SVR	ISVR	AOSVR	LISVR
MSE	14.0263	14.3726	13.2033	13.2683

由此可见,AOSVR 结果最好,但是与 LISVR 的差距可以忽略。由于 LISVR 是在 ISVR 上添加了一个系数 L 得到的,在时间方面它们是一致的,因此我们只比较 SVR、AOSVR、LISVR 的时间。如图 4 所示,我们选取不同的样本数来比较训练时间。

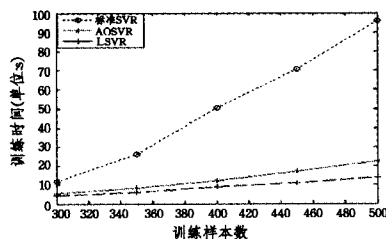


图 4 标准 SVR/AOSVR/LISVR 的时间性能对比

由图 4 可见,LISVR 算法比 SVR 在时间复杂度上有极大提高,而相对于经典的 AOSVR 也有一定程度提高。

3.2.2 机场噪声数据集

本文研究应用的一个重点是机场噪声。我国航空运输业快速繁荣发展,给人们的生活带来便利的同时也带来了一系列问题,其中最严重的是航空噪声扰民问题。我国于 1988 年发布了《机场周围飞机噪声环境标准》(GB9660-88)。现在各民航机场主要使用的噪声预测是美国联邦航空局的噪声预测软件 INM(Integrated Noise Model)。通过噪声软件预测画出噪声等值线。INM 采用了固定的数学模型,因此输入数据对预测结果的影响较大;另外,固定不变的数学公式没有学习能力,不能通过大量的历史数据进行自我学习和更新,因此导致噪声预测软件不能适应现在复杂多变的环境。考虑到机场噪声与影响因子之间复杂的关联关系和输入样本较多的问题,为提高计算模型的适应性,我们将支持向量机增量学习方法引入到机场噪声预测中。本文的算法可以更好地利用大数据以对未来进行更准确的预测。

我们的目的是为国内机场构建一个噪声预测模型,实质就是利用增量 SVR 对机场噪声指标计权等效噪声感觉噪声级 L_{wcpn} 进行预测。本文的原始数据集是以某国际机场的飞行历史记录为基础。该数据集中包含了监测点属性、飞机相关属性、飞过监测点上方航班的相关属性、机场的气象属性等,除了监测点坐标、飞机型号、发动机推力等固定参数,其余参数均由程序模拟生成。由于并没有获取真实的机场噪声值,因此预测目标为基于上述数据通过 Noisemap 软件计算出的 20 个监测点对某一航班测得的一天内的计权等效连续感觉噪声级值。

数据集预处理之后的机场噪声部分样本展示如表 2 所列。

表 2 数据预处理后的机场噪声样本

TRACK TYPE	PLANE ID	POWER	SPEED	EPNL	ALTITUDE	HUMIDITY
1	53	0.23938	43	865	904	52.8551
3	34	0.34861	593	605	2810	27.698
2	85	0.50722	17	906	1958	12.5369
1	53	0.48846	172	865	5270	32.8682

数据集中存在着奇异样本数据,奇异样本数据的存在可能会引起训练过程难以收敛,导致训练时间增加。为了避免这种情况,在训练之前,所有的数据都做了归一化处理。实验结果是通过 10 折交叉运算得出的。增量的过程为,首先处理其中一半的数据,然后增量训练另一半数据,并将这一部分均匀分成 10 个部分,依次加入到训练模型中。机场噪声数据,分别取 3000 和 5000 个数据进行试验。MSE1 表示 3000 个数据的实验结果,MSE2 表示 5000 个数据的实验结果。由于 5000 个数据在进行 SVR 批量训练时内存溢出,使用“-”表示。

使用交叉验证取得参数,其中 C 为 0.6, σ 为 0.9535。结果如表 3 所列。

表 3 3000 样本和 5000 样本机场噪声 MSE 对比

算法	MSE1	MSE2
SVR	42.8028	-
ISVR	39.1446	34.4292
AOSVR	35.2063	-
LISVR	35.1172	31.6300

从表 3 的测试结果可看出, LISVR 优于 SVR 和 ISVR, AOSVR 和 LISVR 的差距可以忽略; 并且由于内存的限制, AOSVR 在数据量较大时和批处理的 SVR 一样, 内存溢出。

表 4 对比 SVR、AOSVR、LISVR 在不同机场噪声样本数时的时间性能, 时间单位为 s。

表 4 SVR/AOSVR/LISVR 的时间性能对比(s)

算法 \ 样本数	SVR	AOSVR	LISVR
200	6.1	2.6	2.8
500	89.7	19.7	11.4
1000	259.8	30.2	25.6
2000	776.5	111.6	70.5
3000	1206.3	205.9	124.1

在时间性能方面, 因为 LISVR 是有条件地选择样本进行训练, 且每次训练时, 对新增样本数量没有限制, 故其在时间性能方面大大优于 AOSVR, 尤其在数据量较多时, 优越性更为明显。

最后, 针对 L 的选择作一个对比。本文选择 L 增量算法中的常数 L, 使用的是启发式选择的 $L = \frac{\# \text{examples}}{\# \text{SVs}}$ 。为了检测 L 对学习结果的影响, 我们做了一些关于 L 选择的试验。其中包含 $0.5 * L$, L , $2 * L$, $5 * L$ 试验, 如表 5 所列, 表明在取值为 $0.5 * L$ 和 $5 * L$ 时, 结果是比较差的。其中 sin 函数表示使用之前的 sin 函数人工数据集。其中 Boston H 是 Boston Housing 数据集。而机场噪声是上文所使用的 5000 样本的数据集。

表 5 不同 L 在不同数据集上的 MSE 对比

数据集	$0.5 * L$	L	$2 * L$	$5 * L$
Sin	0.9589	0.8612	0.8653	0.8924
Boston H	20.7908	13.2683	14.3227	17.2276
机场噪声	36.6719	31.6300	32.8730	33.8912

结束语 本文提出了一个新的增量支持向量回归算法。该方法对一些真实数据集效果较好。通过实验, 我们对比了该算法与已有经典增量算法, 其结果明显优于已有算法, 且时空消耗上远比批量处理算法的小, 也比现有经典在线 SVR 算法的好。

实验过程中, 发现新算法的时间复杂度严重依赖于支持向量数目, 在数量较多时其时间消耗会急剧增加。另外, 本文是对旧样本在训练中增加权重, 进而在实验过程中也可相应减少新增样本的权重, 并能在一定程度上提高算法效率。

参考文献

[1] Syed N, Liu H, Sung K. Incremental learning with support vector machines [C] // Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence (IJCAI-99). Stockholm, Sweden, 1999

[2] Qing L, Ling W, Zheng Z D. Decremental learning based on sample weight Support Vector Regression [M] // Control and Decision. 2010; 1322-1325

[3] Platt J C. Fast Training of Support Vector Machines using Sequential Minimal Optimization [M] // Advances in Kernel Methods. MIT Press Cambridge, MA, USA, 1999; 185-208

[4] Cauwenberghs G, Poggio T. Incremental and decremental sup-

port vector machine learning [M] // Advances in Neural Information Processing Systems. 2001; 409-415

[5] Schölkopf B, Smola A J, Williamson R C, et al. New support vector algorithms [J]. Neural computation, 2000, 12(5): 1207-1245

[6] Ruping S. Incremental learning with support vector machines [C] // Proceedings IEEE International Conference on Data Mining, 2001 (ICDM 2001). IEEE, 2001; 641-642

[7] Li C, Liu K, Wang H. The incremental learning algorithm with support vector machine based on hyperplane-distance [J]. Applied Intelligence, 2011, 34(1): 19-27

[8] Ma J, Theiler J, Perkins S. Accurate on-line support vector regression [J]. Neural Computation, 2003, 15(11): 2683-2703

[9] Smola A J, Schölkopf B. A tutorial on support vector regression [J]. Statistics and computing, 2004, 14(3): 199-222

[10] Wang W. An incremental learning strategy for support vector regression [J]. Neural processing letters, 2005, 21(3): 175-188

[11] Yi Y, Wu J, Xu W. Incremental SVM based on reserved set for network intrusion detection [J]. Expert Systems with Applications, 2011, 38(6): 7698-7707

[12] Hai Y, He W, Fan L. An incremental learning algorithm for SVM based on voting principle [C] // 2010 6th International Conference on Advanced Information Management and Service (IMS). IEEE, 2010; 420-423

[13] 王媛. 支持向量机增量学习算法及其在入侵检测中的应用研究 [D]. 长春: 吉林大学, 2008

[14] Wang J G, Cheng L W, Zhang W X, et al. A modified incremental support vector machine for regression [J]. Applied Mechanics and Materials, 2012, 135: 63-69

[15] Ralaivola L, d'Alché-Buc F. Incremental support vector machine learning: A local approach [M] // Artificial Neural Networks—ICANN 2001. Springer Berlin Heidelberg, 2001; 322-330

[16] Vallada E, Ruiz R. Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem [J]. Omega, 2010, 38(1): 57-67

[17] 萧嵘, 王继成, 孙正兴. 一种 SVM 增量学习算法 α -ISVM [J]. 软件学报, 2001, 12(12): 1818-1824

[18] 宋辰瑶, 徐国华. 前飞状态下直升机旋翼旋转噪声预测 [J]. 声学技术, 2009, 28(2)

[19] 刘占良, 赫旭. 机场航空噪声监测与评价方法 [J]. 中国海洋大学学报, 2006, 36(4): 660-664

[20] 武汉英. 飞机噪声预测模型及其应用 [D]. 天津: 中国民航大学, 2008

[21] Graf H P, Cosatto E, Bottou L, et al. Parallel support vector machines: The cascade svm [J]. Advances in neural information processing systems, 2004, 17: 521-528

[22] Mitra P, Murthy C A, Pal S K. Data condensation in large databases by incremental learning with support vector machines [C] // Proceedings 15th International Conference on Pattern Recognition, 2000. IEEE, 2000, 2: 708-711

[23] Laskov P, Gehl C, Krüger S, et al. Incremental support vector learning: Analysis, implementation and applications [J]. The Journal of Machine Learning Research, 2006, 7: 1909-1936

[24] Diehl C P, Cauwenberghs G. SVM incremental learning, adaptation and optimization [C] // Proceedings of the International Joint Conference on Neural Networks, 2003. IEEE, 2003, 4: 2685-2690