

# 海量数据中的分布式支持向量回归

梁姝娜, 张 齐\*

青岛大学, 山东 青岛

收稿日期: 2022年3月20日; 录用日期: 2022年4月14日; 发布日期: 2022年4月22日

## 摘 要

大规模的数据给传统的统计推断方法带来了新的挑战, 比如在分析超过一台计算机容量的海量数据集时, 由于数据太大, 无法保存在计算机内存中, 计算任务可能需要花费很长时间才能获得结果。为了有效地解决海量数据情形下的诸多问题, 本文研究了支持向量回归的分布式估计。首先采用平滑技术发展了一种平滑支持向量回归(S-SVR)估计方法。然后基于分而治之的思想, 针对海量数据集对S-SVR估计方法提出了分而治之的支持向量回归估计算法(DC-SVR), 该方法解决了内存限制和计算时间的问题。此外, 本文中提出的DC-SVR方法中的参数可通过网格搜索和交叉验证相结合的方法获得, 具有自适应性, 其中最优的参数是由每次数据自动选择的。在模拟研究中, 通过不同情形的实验表明了文章所提估计量的优越性, 模拟结果显示通过DC-SVR所得的估计量在平均绝对偏差和均方误差评价准则下的差异更小。

## 关键词

支持向量回归, 分治法, 平滑化, 海量数据

# Distributed Support Vector Regression of Massive Datasets

Shu'na Liang, Qi Zhang\*

Qingdao University, Qingdao Shandong

Received: Mar. 20<sup>th</sup>, 2022; accepted: Apr. 14<sup>th</sup>, 2022; published: Apr. 22<sup>nd</sup>, 2022

## Abstract

Large scale data brings new challenges to the traditional statistical inference methods. For example, when analyzing massive datasets whose sizes usually exceed the capacity of a single computer, the data is too large to be saved in computer memory and computing task may take too long to get the results. In order to effectively solve many problems in the case of massive data, this paper stu-

\*通讯作者。

dies the distributed estimation of support vector regression. Firstly, we develop smoothed support vector regression (S-SVR) estimation method. Then based on the idea of divide-and-conquer, we propose divide-and-conquer support vector regression estimation algorithm (DC-SVR) for the S-SVR estimation method of massive datasets. This method solves the problems of memory limitation and computing time. In addition, the parameters in the DC-SVR method can be obtained by the combination of grid search and cross validation, which is adaptive. The optimal parameters are automatically selected by each data. In the simulation study, a lot of numerical simulation studies are carried out to verify the superiority of our proposed distributed estimator. The simulation results show that the DC-SVR estimator has less difference under the evaluation criteria of mean absolute deviation and mean square error.

## Keywords

Support Vector Regression, Divide-and-Conquer Method, Smoothing, Massive Datasets

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

近年来,对海量数据集的统计分析已成为人们日益关注的课题。数据集的规模不断扩大,部分原因是它们越来越多地被无处不在的信息传感移动设备、遥感技术和无线传感器网络等收集。这些数据的规模给传统的统计估计方法带来了新的挑战。数据库有数百个字段、数十亿条记录和数兆字节的信息并不少见。例如,大型文本语料库很容易超过内存限制,因此不能一次全部加载到内存中。例如,在给定的样本的情况下,经典的估计方法通常会建立一个极大似然估计(MLE)问题,然后用确定性优化方法(如梯度下降法或牛顿法)求解极大似然估计。然而,当样本量过大时,采用这种方法有两个主要障碍。首先,在分析大小通常超过一台计算机容量的海量数据集时,机器没有足够的内存来一次加载整个数据集。其次,确定性优化方法计算量大,计算任务可能需要太长时间才能获得结果。为了解决存储限制和计算问题,最近在统计学中引入了起源于计算机科学文献的分布式计算方法。一种通用的分布式计算方案是将整个数据集划分为多个部分,然后对每个部分独立执行统计分析,即每台机器用存储在本地的数据形成一个局部估计量。最终的估计量将通过局部估计量之间的一些聚合来获得。

在分布式统计学习中,最简单和最流行的方法是平均法。平均最先是 by McDonald *et al.* (2009) [1] 提出的,用于多项式回归,他们给出了估计误差的非渐近误差界,表明平均化降低了局部估计量的方差,但对偏差没有影响。在后续工作中,Zinkevich *et al.* (2010) [2] 研究了平均的一种变形,其中每台机器都在数据集的随机子集上计算具有随机梯度下降(SGD)的局部估计量,证明了他们的估计器收敛于集中估计器。Zhang *et al.* (2013) [3] 研究了两种基于大规模数据的分布式统计优化算法。第一个算法是平均法,第二种算法是一种基于适当形式的引导的新方法,只需要进行一轮通信。Cheyer *et al.* (2014) [4] 提出了一种分治方法,并使用几种计算密集型惩罚回归方法对其进行了分析。这种分治方法可以大大减少计算时间和计算机内存需求。2016年 Jordan *et al.* [5] 提出了一种解决分布式统计学习问题的 Communication-efficient Surrogate Likelihood (CSL) 框架。2017年 Lee *et al.* [6] 设计了一种在高维环境下实现分布式稀疏回归的高效通信方法。在2020年,Chen *et al.* [7] 研究并提出了主成分分析(PCA)的分布估计问题。

支持向量机(Support Vector Machine)是由 Cortes 和 Vapnik [8] 于 1995 年最先提出的,是统计机器学习

习中最流行的方法之一, 具有相对优良的性能指标, 在图像分析、医学、神经网络等领域有着广泛的应用。SVM 最初是针对经典的二分类问题提出的(Liu *et al.* 2007 [9]; Hsieh *et al.* 2013 [10]; Lian *et al.* 2018 [11]), 支持向量分类(Support Vector Classification)。然而, 随着支持向量机方法理论的发展, 支持向量机应用的领域越来越广泛, 其中一个重要的应用是解决回归问题, 即支持向量回归(Support Vector Regression)。支持向量回归与传统机器学习方法相比有较好的学习性能, 是一种稳健而有效的估计方法, 成为统计机器学习理论的研究热点。2004 年 Smola *et al.* [12]详细概述了支持向量机用于回归和函数估计的基本思想。2009 年 ReShma *et al.* [13]通过用正则化最小二乘的模糊支持向量回归, 以此来处理相应的金融时间序列预测。2013 年 Rivas *et al.* [14]提出了一种训练大规模 SVR 模型的求解方法, 该方法将 SVR 问题转化为一个线性规划问题。2017 年 Anyu Cheng *et al.* [15]通过基于混沌理论、支持向量回归的多源多尺度的交通流预测算法, 并由支持向量回归模型来预测某市区的实际交通流量的变化。2019 年 S. Maldonado *et al.* [16]提出了一套基于  $\varepsilon$ -SVR 的时间序列预测问题滞后期自动选择的算法并且验证了该算法的优越性能。

SVR 需要解决一个非光滑优化问题, 这对于海量数据集来说计算是非常密集的。因此, 在第二章中本文采用平滑技术(Horowitz 1998 [17]; Pang *et al.* 2012 [18]; Chen *et al.* 2018 [19]和 Wang *et al.* 2019 [20]), 提出了平滑支持向量回归(S-SVR)估计方法。然后, 在有限的内存约束下, 在第三章中提出了海量数据集的 DC-SVR 方法, 该方法解决了内存限制和计算时间问题。在实践中, 具有固定(非自适应)参数的学习过程只有在某些情况下才有其优势, 因为不同的参数可能适合不同的数据, 在以往的支持向量回归中, 参数一般是提前指定的, 它是非自适应性的, 所以在第四章中我们考虑了 SVR 的自适应参数, 并且给出了本文的算法。第五章中, 本文通过数值模拟验证了该方法的有效性。

## 2. 支持向量回归

### 2.1. 经典支持向量回归

本文假设有  $n$  个样本  $\{(X_i, y_i)\}$  来自以下模型:

$$y_i = X_i^T \beta_- + e_i, i=1, 2, \dots, n, \quad (1)$$

考虑随机变量  $(X_-, Y)$ , 其中  $X_- = (X_1, \dots, X_p)^T \in \mathbb{R}^p$ 。定义  $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T = (\beta_0, \beta_-)^T \in \mathbb{R}^{(p+1)}$ ,  $X = (1, X_1, \dots, X_p)^T$ , 超平面  $\beta_0 + X_-^T \beta_- = 0$ 。SVR 的目标是估计由  $\beta_0 + X_-^T \beta_- = X^T \beta$  定义的超平面通过求解优化问题:

$$\begin{aligned} \min_{\beta_-, \beta_0} & \frac{1}{2} \|\beta_-\|^2 \\ \text{s.t.} & |y_i - X_i^T \beta| \leq \varepsilon, i=1, 2, \dots, n \end{aligned} \quad (2)$$

通过引入两个松弛变量  $(\xi_i, \xi'_i)$ , 对 SVR 问题进行优化, 对于任何样本  $(X_i, y_i)$ , 当划分有误差时, 松弛变量都大于 0, 误差不存在时松弛变量为 0。其中观测值  $y_i \geq X_i^T \beta + \varepsilon$  出现误差时对应于松弛变量  $\xi_i > 0$ , 观测值  $y_i \leq X_i^T \beta - \varepsilon$  出现误差时对应的松弛变量是  $\xi'_i > 0$ 。所以引入松弛变量后, 可以将目标问题等式(2)转化为如下式(3)求解问题:

$$\begin{aligned} \min_{\beta_-, \beta_0, \xi_i, \xi'_i} & \frac{1}{2} \|\beta_-\|^2 + \lambda' \sum_{i=1}^n (\xi_i + \xi'_i) \\ \text{s.t.} & X_i^T \beta - y_i \leq \varepsilon + \xi_i \\ & y_i - X_i^T \beta \leq \varepsilon + \xi'_i \\ & \xi_i \geq 0, \xi'_i \geq 0, i=1, 2, \dots, n \end{aligned} \quad (3)$$

上述是一个凸二次规划问题, 可以把式(3)中引入拉格朗日乘子, 然后转化为求解它的对偶问题。对式(3)的每个约束条件添加拉格朗日乘子  $\mu_i \geq 0$ ,  $\mu'_i \geq 0$ ,  $\alpha_i \geq 0$ ,  $\alpha'_i \geq 0$ , 从而得到拉格朗日函数式(4):

$$\begin{aligned} L(\beta_-, \beta_0, \alpha, \alpha', \xi, \xi', \mu, \mu') &= \frac{1}{2} \|\beta_-\|^2 + \lambda' \sum_{i=1}^n (\xi_i + \xi'_i) - \sum_{i=1}^n \mu_i \xi_i - \sum_{i=1}^n \mu'_i \xi'_i \\ &+ \sum_{i=1}^n \alpha_i (X_i^T \beta - y_i - \varepsilon - \xi_i) + \sum_{i=1}^n \alpha'_i (y_i - X_i^T \beta - \varepsilon - \xi'_i) \end{aligned} \quad (4)$$

把决策函数  $\beta_0 + X^T \beta_- = X^T \beta$  代入上式, 再令  $L(\beta_-, \beta_0, \alpha, \alpha', \xi, \xi', \mu, \mu')$  对  $\beta_-, \beta_0, \xi$  和  $\xi'$  的偏导为零可得

$$\begin{aligned} \beta_- &= \sum_{i=1}^n (\alpha'_i - \alpha_i) X_i, \\ 0 &= \sum_{i=1}^n (\alpha'_i - \alpha_i), \\ \lambda' &= \alpha_i + \mu_i, \\ \lambda' &= \alpha'_i + \mu'_i. \end{aligned} \quad (5)$$

接着把上式代入拉格朗日函数  $L(\beta_-, \beta_0, \alpha, \alpha', \xi, \xi', \mu, \mu')$ , 最后得到 SVR 的对偶问题

$$\begin{aligned} \max_{\alpha, \alpha'} \quad & \sum_{i=1}^n y_i (\alpha'_i - \alpha_i) - \varepsilon (\alpha'_i - \alpha_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha'_i - \alpha_i) (\alpha'_j - \alpha_j) X_i^T X_j \\ \text{s.t.} \quad & \sum_{j=1}^n (\alpha'_j - \alpha_j) = 0, \\ & 0 \leq \alpha_i, \alpha'_i \leq \lambda'. \end{aligned} \quad (6)$$

上述过程的求解需要满足 KKT 条件:

$$\begin{cases} \alpha_i (X_i^T \beta - y_i - \varepsilon - \xi_i) = 0, \\ \alpha'_i (y_i - X_i^T \beta - \varepsilon - \xi'_i) = 0, \\ \alpha_i \alpha'_i = 0, \quad \xi_i \xi'_i = 0, \\ (\lambda' - \alpha_i) \xi_i = 0, \quad (\lambda' - \alpha'_i) \xi'_i = 0. \end{cases} \quad (7)$$

由 KKT 条件, 当且仅当  $X_i^T \beta - y_i - \varepsilon - \xi_i = 0$  时  $\alpha_i$  可以取非零值, 当且仅当  $y_i - X_i^T \beta - \varepsilon - \xi'_i = 0$  时,  $\alpha'_i$  可以取非零值。换言之, 仅当样本  $(X_i, y_i)$  不落入  $\varepsilon$  间隔带中时, 对应的乘子  $\alpha_i$  和  $\alpha'_i$  才取非零值。除此之外, 约束条件  $X_i^T \beta - y_i - \varepsilon - \xi_i = 0$ 、 $y_i - X_i^T \beta - \varepsilon - \xi'_i = 0$  不能同时成立, 所以  $\alpha_i$  和  $\alpha'_i$  中至少有一个为 0。

把系数  $\beta_- = \sum_{i=1}^n (\alpha'_i - \alpha_i) X_i$  代入决策函数  $\beta_0 + X^T \beta_- = X^T \beta$ , 可得到 SVR 解的形式:

$$f(X) = \sum_{i=1}^n (\alpha'_i - \alpha_i) X_i^T X + \beta_0 \quad (8)$$

对于等式(8), 满足  $\alpha'_i - \alpha_i \neq 0$  的样本就是支持向量, 这些样本点必然会落在  $\varepsilon$ -间隔带之外, 而在间隔带之间的样本点都满足  $\alpha'_i = \alpha_i = 0$ 。所以 SVR 的支持向量仅仅是训练样本的一部分, 支持向量的数量是一个与误差  $\varepsilon$  有关的函数, 误差  $\varepsilon$  越大, 支持向量的个数越少。

由于  $\alpha'_i$  和  $\alpha_i$  不能全是零, 根据 KKT 条件可知, 对每个样本  $(X_i, y_i)$  有  $(\lambda' - \alpha_i) \xi_i = 0$  或者  $(\lambda' - \alpha'_i) \xi'_i = 0$ , 进而可以得到  $\alpha_i (X_i^T \beta - y_i - \varepsilon - \xi_i) = 0$  或者  $\alpha'_i (y_i - X_i^T \beta - \varepsilon - \xi'_i) = 0$  成立, 于是在求出  $\alpha'_i$  或  $\alpha_i$  后, 可以得到参数  $\beta_0$ 。比如说, 若  $0 < \alpha_i < \lambda'$ , 则必有  $\xi_i = 0$ , 所以最后可以求出  $\beta_0$  的等式:

$$\beta_0 = y_i + \varepsilon - \sum_{i=1}^n (\alpha'_i - \alpha_i) X_i^T X_i \quad (9)$$

所以在对(6)的求解过程中得到  $\alpha_i$  后, 从理论上说, 可任意选取满足  $0 < \alpha_i < \lambda'$  的样本通过等式(9)求得  $\beta_0$ 。

## 2.2. 平滑支持向量回归估计

在这一部分中, 我们提出了一种支持向量回归的线性型估计, 称为平滑支持向量回归(Smooth Support Vector Regression, S-SVR)。带松弛变量的支持向量回归还有另一种解释,  $\epsilon$ -不敏感损失函数 + L2 正则, 就是最小化以下目标函数:

$$L(\beta) = \frac{1}{n} \sum_{i=1}^n |y_i - X_i^T \beta|_{\epsilon} + \frac{\lambda}{2} \|\beta\|_2^2, \quad (10)$$

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{p+1}} L(\beta)$$

其中  $|v|_{\epsilon} = \begin{cases} |v| - \epsilon, & |v| \geq \epsilon \\ 0, & \text{其它} \end{cases}$  是  $\epsilon$ -不敏感损失函数,  $\lambda > 0$  是正则化参数,  $\|\cdot\|_2$  表示向量的欧几里德范数。

对于(10), 由于绝对值和不敏感损失函数的存在, 目标函数是不可微的, 所以接下来要解决这个问题。首先, 引入量  $Z$  来解决绝对值问题, 使得  $z_i = \begin{cases} -1, & y_i - X_i^T \beta \leq 0 \\ 1, & y_i - X_i^T \beta > 0 \end{cases}$ , 所以  $z_i (y_i - X_i^T \beta) \geq 0$ 。即可以用  $z_i (y_i - X_i^T \beta)$  来取代绝对值  $|y_i - X_i^T \beta|$ , 因此  $\epsilon$ -不敏感损失函数可以写成  $|y_i - X_i^T \beta|_{\epsilon} = \max(|y_i - X_i^T \beta| - \epsilon, 0) = \max(z_i (y_i - X_i^T \beta) - \epsilon, 0)$ 。所以目标函数变为:

$$L(\beta) = \frac{1}{n} \sum_{i=1}^n [z_i (y_i - X_i^T \beta)]_{\epsilon} + \frac{\lambda}{2} \|\beta\|_2^2,$$

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{p+1}} L(\beta)$$

其中  $[v]_{\epsilon} = \max(v - \epsilon, 0)$  是  $\epsilon$ -不敏感损失函数。

然后基于平滑技术(Horowitz 1998 [17]; Pang et al. 2012 [18]; Chen et al. 2018 [19]和 Wang et al. 2019 [20]), 考虑了一个平滑函数  $S(\cdot)$ , 即  $S(t) = \begin{cases} 1, & t \geq 1 \\ 0, & t \leq -1 \end{cases}$ 。我们用平滑近似  $D_h(t) = tS\left(\frac{t}{h}\right)$  代替  $\epsilon$ -不敏感损失函数, 其中  $h \rightarrow 0$  是带宽,  $S\left(\frac{t}{h}\right)$  近似示性函数  $I\{t \geq 0\}$ ,  $D_h(t)$  近似于  $\max(t, 0)$ 。所以  $\epsilon$ -不敏感损失函数  $[z_i (y_i - X_i^T \beta)]_{\epsilon} = [z_i (y_i - X_i^T \beta) - \epsilon] S\left(\frac{z_i (y_i - X_i^T \beta) - \epsilon}{h}\right)$ , 并定义以下目标函数和估计量:

$$L(\beta) = \frac{1}{n} \sum_{i=1}^n [z_i (y_i - X_i^T \beta) - \epsilon] S\left(\frac{z_i (y_i - X_i^T \beta) - \epsilon}{h}\right) + \frac{\lambda}{2} \|\beta\|_2^2 \quad (11)$$

$$\hat{\beta}_h = \arg \min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^n [z_i (y_i - X_i^T \beta) - \epsilon] S\left(\frac{z_i (y_i - X_i^T \beta) - \epsilon}{h}\right) + \frac{\lambda}{2} \|\beta\|_2^2 \quad (12)$$

$$= \arg \min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^n D_h[z_i (y_i - X_i^T \beta) - \epsilon] + \frac{\lambda}{2} \|\beta\|_2^2$$

目标函数(12)是可微的, 通过一阶最优性条件, 解  $\widehat{\beta}_h$  满足:

$$\frac{1}{n} \sum_{i=1}^n D'_h \left[ z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon \right] + \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{h-} \end{pmatrix} = 0 \quad (13)$$

其中

$$D'_h \left[ z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon \right] = (-z_i X_i) \left[ S \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon}{h} \right) + \frac{z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon}{h} \right) \right]$$

根据(13), 可以通过以下公式表示  $\widehat{\beta}_h$  :

$$\begin{aligned} \widehat{\beta}_h &= \left[ \frac{1}{n} \sum_{i=1}^n \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon}{h} \right) \right]^{-1} \\ &\times \left\{ \frac{1}{n} \sum_{i=1}^n z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon}{h} \right) + \frac{z_i y_i - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_h) - \epsilon}{h} \right) \right] - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{h-} \end{pmatrix} \right\} \end{aligned} \quad (14)$$

然而, 这个不动点方程中没有  $\widehat{\beta}_h$  的封闭解, 所以将(14)右边的  $\widehat{\beta}_h$  替换为一致的初始估计  $\widehat{\beta}_0$ 。得到下面的 S-SVR 估计量:

$$\begin{aligned} \widehat{\beta} &= \left[ \frac{1}{n} \sum_{i=1}^n \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \right]^{-1} \\ &\times \left\{ \frac{1}{n} \sum_{i=1}^n z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) + \frac{z_i y_i - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \right] - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{0-} \end{pmatrix} \right\} \end{aligned} \quad (15)$$

初始估计量可以通过求解小样本的线性支持向量回归来构造, 所以我们对小批量样本(如第一台机器上的样本, 见下一节)的原始支持向量回归(10)进行了优化得到初始估计量。然后给定初始估计量, 可以通过分布式方案解决(15)的 S-SVR。

### 3. 分布式支持向量回归

当无法将所有数据加载到内存中时, 很难求解(15)。但我们可以通过分布式方案来实现。具体来说, 基于 S-SVR, 引入分治法提出分治支持向量回归估计(Divide and Conquer Support Vector Regression, DC-SVR)来计算  $\widehat{\beta}$ 。

首先, 将总数据集  $\{1, \dots, n\}$  分为  $L$  个子集  $\{\mathcal{N}_1, \dots, \mathcal{N}_L\}$ , 每个子集的大小都相等  $m = n/L$ 。用  $\mathcal{D}_l = \{(X_i, y_i, z_i) : i \in \mathcal{N}_l\}$  表示第  $l$  台本地机器中的数据, 第  $l$  台机器包含  $m$  数据。然后基于  $\mathcal{D}_l$  获得初始估计值  $\widehat{\beta}_0$  :

$$\widehat{\beta}_0 = \arg \min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{m} \sum_{i \in \mathcal{N}_1} |y_i - X_i^T \beta|_{\epsilon} + \frac{\lambda}{2} \|\beta_{-}\|_2^2 \quad (16)$$

为了计算  $\widehat{\beta}$ , 对于每一批数据  $\mathcal{D}_l$ ,  $l = 1, \dots, L$ , 计算以下量:

$$\begin{aligned}
 A_l &= \frac{1}{m} \sum_{i \in \mathcal{N}_l} \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \\
 B_l &= \frac{1}{m} \sum_{i \in \mathcal{N}_l} z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) + \frac{z_i y_i - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \right]
 \end{aligned} \tag{17}$$

在每台机器中计算  $A_l, B_l$ , 然后计算出每一台机器中存储的数据所对应的局部估计量  $\widehat{\beta}_{(l)}$ :

$$\begin{cases}
 \widehat{\beta}_{(1)} = A_1^{-1} \left( B_1 - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{0-} \end{pmatrix} \right) \\
 \widehat{\beta}_{(2)} = A_2^{-1} \left( B_2 - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{0-} \end{pmatrix} \right) \\
 \vdots \\
 \widehat{\beta}_{(L)} = A_L^{-1} \left( B_L - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{0-} \end{pmatrix} \right)
 \end{cases} \tag{18}$$

所有机器接收到  $A_l, B_l$  计算出的局部估计量后, 汇总数据将各个机器的局部估计量取平均得到最终估计量:

$$\widehat{\beta} = \frac{1}{L} \sum_{l=1}^L \widehat{\beta}_{(l)} \tag{19}$$

下面分析一下估计的渐近性质, 根据(15),  $\widehat{\beta}$  和真实系数  $\beta^*$  之间的差为:

$$\begin{aligned}
 \widehat{\beta} - \beta^* &= \left[ \frac{1}{n} \sum_{i=1}^n \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \right]^{-1} \\
 &\times \left\{ \frac{1}{n} \sum_{i=1}^n z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) + \frac{z_i (y_i - X_i^T \beta^*) - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \right] - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{0-} \end{pmatrix} \right\}
 \end{aligned}$$

因此  $\widehat{\beta} - \beta^*$  可以被表示为

$$\widehat{\beta} - \beta^* = M^{-1} \left( Q - \lambda \begin{pmatrix} 0 \\ \widehat{\beta}_{0-} \end{pmatrix} \right) \tag{20}$$

其中

$$\begin{aligned}
 M &= \frac{1}{n} \sum_{i=1}^n \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \\
 Q &= \frac{1}{n} \sum_{i=1}^n z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) + \frac{z_i (y_i - X_i^T \beta^*) - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \widehat{\beta}_0) - \epsilon}{h} \right) \right]
 \end{aligned}$$

为了证明所提出的估计量的渐近性质, 首先引入一些正则性条件和假设。



假设 1:  $e$  的密度函数  $f(\cdot)$  是 Lipschitz 连续的。另外, 对于某些常量  $c_1, c_2$ , 假设  $0 < c_1 \leq \lambda_{\min}(M') \leq \lambda_{\max}(M') \leq c_2$ 。

假设 2: 平滑函数  $S(\cdot)$  满足  $S(u) = \begin{cases} 1, & u \geq 1 \\ 0, & u \leq -1 \end{cases}$ 。进一步, 假设  $S(\cdot)$  是两次可微的, 并且它的二阶导数  $S^{(2)}(\cdot)$  是有界的。此外, 我们假设带宽  $h = o(1)$ 。

假设 3: 对于某些  $t > 0$  和  $C > 0$ , 假设  $p = o(nh/(\log n))$  和  $\sup_{\|v\|_2 \leq 1} Ee^{(v^T X)^t} \leq C$ 。

假设 4: 独立随机向量  $\{X_i, 1 \leq i \leq n\}$  满足  $\sup_{i,j} E|X_{ij}|^3 = O(1)$ 。

定理 1. 在满足假设条件 1-4 的情况下, 并且假设初始估计  $\hat{\beta}_0$  满足  $\|\hat{\beta}_0 - \beta^*\|_2 = O_p\left(\sqrt{\frac{p}{m}}\right)$ 。然后我们有,

$$\hat{\beta} - \beta^* = \left(\frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T\right)^{-1} \left(\frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} - \lambda \begin{pmatrix} 0 \\ \beta_-^* \end{pmatrix}\right) + r_n \quad (21)$$

其中

$$\|r_n\|_2 = O_p\left(\sqrt{\frac{p^2 \log n}{n^2 h}} + \sqrt{\frac{ph \log n}{n}} + \frac{p}{m} + h^2\right) \quad (22)$$

证明: 如果初始估计值  $\hat{\beta}_0$  接近真实系数  $\beta^*$ , 则  $M$  和  $Q$  分别接近  $M(\beta^*)$  和  $Q(\beta^*)$ 。

$$M(\beta^*) = \frac{1}{n} \sum_{i=1}^n \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \beta^*) - \epsilon}{h} \right)$$

$$Q(\beta^*) = \frac{1}{n} \sum_{i=1}^n z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \beta^*) - \epsilon}{h} \right) + \frac{z_i (y_i - X_i^T \beta^*) - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \beta^*) - \epsilon}{h} \right) \right]$$

假设  $e_i = z_i (y_i - X_i^T \beta) - \epsilon$ , ( $e_i = |y_i - X_i^T \beta| - \epsilon$ ), 则有:

$$M(\beta^*) = \frac{1}{n} \sum_{i=1}^n \frac{X_i X_i^T}{h} S' \left( \frac{e_i}{h} \right)$$

$$Q(\beta^*) = \frac{1}{n} \sum_{i=1}^n z_i X_i \left[ S \left( \frac{e_i}{h} \right) + \frac{e_i}{h} S' \left( \frac{e_i}{h} \right) \right]$$

$\frac{1}{h} S' \left( \frac{e_i}{h} \right)$  和  $S \left( \frac{e_i}{h} \right) + \frac{e_i}{h} S' \left( \frac{e_i}{h} \right)$  分别近似于  $\delta(e_i)$  和  $I\{e_i \geq 0\}$ , 所以  $M(\beta^*)$  和  $Q(\beta^*)$  近似于  $\frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T$  和  $\frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\}$ 。

根据上述分析, 当  $\hat{\beta}_0$  近似  $\beta^*$  时,  $M$  和  $Q$  分别近似  $\frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T$  和  $\frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\}$ 。因此,  $\hat{\beta} - \beta^*$  近似于:

$$\left(\frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T\right)^{-1} \left(\frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} - \lambda \begin{pmatrix} 0 \\ \beta_-^* \end{pmatrix}\right)$$



假设有一个初始估计  $\widehat{\beta}_0$  且  $\|\widehat{\beta}_0 - \beta^*\|_2 = O_p(a_n)$ ,  $a_n = O(h)$ 。则有,

$$\left\| Q - \frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} \right\|_2 = O_p \left( \sqrt{\frac{ph \log n}{n}} + a_n^2 + h^2 \right)$$

$$\left\| M - \frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T \right\|_2 = O_p \left( \sqrt{\frac{p \log n}{nh}} + a_n + h \right)$$

对于独立随机向量  $\{X_i, 1 \leq i \leq n\}$  且  $\sup_j E|X_j|^3 = O(1)$ , 有  $E \left\| \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} \right\|_2^2 = O(np)$ 。所以能够得到:

$$\left\| \frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} \right\|_2 = O_p \left( \sqrt{\frac{p}{n}} \right)$$

根据上述结果, 可以得出:

$$\widehat{\beta} - \beta^* = \left( \frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} - \lambda \begin{pmatrix} 0 \\ \beta_-^* \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ \beta_-^* - \widehat{\beta}_0 \end{pmatrix} \right) + r_n$$

其中

$$\|r_n\|_2 = O_p \left( \sqrt{\frac{p^2 \log n}{n^2 h}} + \sqrt{\frac{ph \log n}{n}} + \sqrt{\frac{p \log n}{nh}} \lambda + \lambda h + h^2 \right)$$

设  $\lambda \leq h$  和  $\|\widehat{\beta}_0 - \beta^*\|_2 = O_p \left( \sqrt{\frac{p}{m}} \right)$ , 则有

$$\widehat{\beta} - \beta^* = \left( \frac{1}{n} \sum_{i=1}^n \delta(e_i) X_i X_i^T \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n z_i X_i I\{e_i \geq 0\} - \lambda \begin{pmatrix} 0 \\ \beta_-^* \end{pmatrix} \right) + r_n$$

其中

$$\|r_n\|_2 = O_p \left( \sqrt{\frac{p^2 \log n}{n^2 h}} + \sqrt{\frac{ph \log n}{n}} + \frac{p}{m} + h^2 \right)$$

#### 4. 参数优化

对于支持向量回归, 模型参数的选择对于预测结果有着重要影响。在建立 SVR 模型后, 需要对模型参数进行标定, 使得最终模型具有较好的预测效果以及泛化能力, 需要标定的模型参数及其具体含义如下:

##### 1) 不敏感参数 $\epsilon$

该参数控制不敏感带宽度的大小, 影响着支持向量的数目。当值较小时, 模型回归精度较高, 支持向量的数量增多, 易产生过拟合。反之, 模型回归精度较低, 支持向量的数量减少, 易导致模型欠拟合。

##### 2) 正则化参数 $\lambda$

SVR 模型的惩罚因子, 该参数控制模型的复杂度与模型训练误差项的比重, 反映了对超出不敏感带宽度数据的惩罚程度。正则化参数  $\lambda$  取值过大, 对超出不敏感带宽度数据的惩罚小, 会使得模型训练误差增大, 使得模型处于欠学习状态, 而  $\lambda$  取值过小, 能够提升模型回归精度, 但模型泛化能力减弱, 对

于新数据的测试误差将变大, 使得模型处于过学习状态。

许多研究人员使用交叉验证来选择参数(Chen *et al.* (2007) [21]; Anass *et al.* (2018) [22])。在实践中, 具有固定(非自适应)参数的学习过程只有在某些情况下才有其优势, 因为不同的参数可能最适合不同的数据, 这促使我们考虑对 SVR 中参数进行自适应。因此, 基于上述分析, 为了使 SVR 的预测效果达到最优, 在本文中, 我们提出一种自适应参数学习过程, 根据不同的数据本文使用网格搜索和  $k$ -折交叉验证相结合的方法自适应的选择 SVR 模型最优参数组合  $(\lambda, \epsilon)$ 。

利用网格搜索和  $k$ -折交叉验证相结合的方法优化 SVR 中的参数  $(\lambda, \epsilon)$  的具体步骤如下:

1) 建立网格坐标。设定参数  $\lambda$  和  $\epsilon$  的范围, 设定搜索步长, 在  $\lambda$ 、 $\epsilon$  坐标系上建立二维网格, 网格的节点就是  $\lambda$ 、 $\epsilon$  组成的参数对;

2) 划分样本。利用  $k$ -折交叉验证法时, 将原始样本均分成  $k$  组, 每组轮流作为验证集, 测试其他  $k-1$  组训练得到的模型;

3) 确定预测误差。取上述  $k$  次测试结果的均方误差的平均值作为本次模型的性能指标。对每一组  $\lambda$ 、 $\epsilon$  的值, 计算支持向量回归  $k$  次交叉验证的均方误差(MSE);

4) 确定最优参数组合。网格上所有交叉点处的参数组合经过  $k$ -折交叉验证后, MSE 最小值所对应的参数组合就是模型的最优参数。

本文算法:

**Step 1.** 输入数据集  $\{\mathcal{N}_1, \dots, \mathcal{N}_L\}$  在每台机器上, 平滑函数  $S(\cdot)$ , 带宽  $h$ , 不敏感参数  $\epsilon$  和正则化参数  $\lambda$ ;

**Step 2.** 基于第一台机器  $\mathcal{D}_1$  的数据计算初始估计  $\hat{\beta}_0$  :

$$\hat{\beta}_0 = \arg \min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{m} \sum_{i \in \mathcal{N}_1} |y_i - X_i^T \beta|_c + \frac{\lambda}{2} \|\beta\|_2^2$$

**Step 3.** 对于每一批数据  $\mathcal{D}_l$ ,  $l=1, \dots, L$ , 计算  $(A_l, B_l)$  :

$$A_l = \frac{1}{m} \sum_{i \in \mathcal{N}_l} \frac{X_i X_i^T}{h} S' \left( \frac{z_i (y_i - X_i^T \hat{\beta}_0) - \epsilon}{h} \right)$$

$$B_l = \frac{1}{m} \sum_{i \in \mathcal{N}_l} z_i X_i \left[ S \left( \frac{z_i (y_i - X_i^T \hat{\beta}_0) - \epsilon}{h} \right) + \frac{z_i y_i - \epsilon}{h} S' \left( \frac{z_i (y_i - X_i^T \hat{\beta}_0) - \epsilon}{h} \right) \right]$$

**Step 4.** 计算局部估计量  $\hat{\beta}_{(1)}, \hat{\beta}_{(2)}, \dots, \hat{\beta}_{(L)}$ ;

**Step 5.** 汇总数据将各个机器的局部估计量取平均得到最终估计量:

$$\hat{\beta} = \frac{1}{L} \sum_{l=1}^L \hat{\beta}_{(l)}$$

## 5. 数值模拟

在本节中, 我们通过仿真来说明 DC-SVR 的性能。数据由以下模型生成:

$$Y_i = X_i^T \beta + e_i, i=1, 2, \dots, n$$

其中  $\beta = \mathbf{1}_p$  和  $X$  服从多元正态分布  $N(0, \Sigma)$ ,  $\Sigma_{ij} = 0.5^{|i-j|}$ ,  $1 \leq i, j \leq p$ 。在我们的模拟中, 误差  $e_i$  相互独立且产生于以下两种分布:

- 1) 标准正态分布,  $e_i \sim N(0,1)$ ;
- 2) 自由度为 3 的  $t$  分布,  $e_i \sim t(3)$ 。

最优参数组合  $(\lambda, \epsilon)$  是由每组数据自动选择的。在模拟中, 取  $k$  为 5, 即 5-折交叉验证。根据经验, 将参数  $\epsilon$  的变化范围设置为  $[2^{-6}, 2^{-5}, \dots, 2^1]$ , 并将参数  $\lambda$  的变化范围设置为  $[2^{-8}, 2^{-7}, \dots, 2^{-1}]$ , 真参数设置为  $\beta^* = (0, 1, \dots, 1)^T \in \mathbb{R}^{(p+1)}$ 。使用核函数的积分作为平滑函数:

$$S(t) = \begin{cases} 0, & \text{if } t \leq -1 \\ \frac{1}{2} + \frac{15}{16} \left( t - \frac{2}{3}t^3 + \frac{1}{5}t^5 \right), & \text{if } |t| < 1 \\ 1, & \text{if } t \geq 1 \end{cases}$$

在本文中, 所有模拟结果均为 500 次独立运行的平均值。我们将分而治之的方法应用于经典支持向量回归中, 并与本文的方法进行比较。在这里, 我们还用完整数据集(非分布式方法)估计参数, 观察 DC-SVR 方法是否接近它。总结一下, 我们将报告三种比较方法:

- 1) 用完整数据集估计 SVR 的非分布式方法(SVR-ALL);
- 2) naive 分布式经典支持向量回归方法(ND-SVR);
- 3) 分治平滑支持向量回归方法(DC-SVR)。

为了评估不同方法的表现, 本文采用平均绝对偏差(MAD)和均方误差(MSE)作为评价指标。

$$\text{MAD} = \frac{1}{p} \sum_{k=1}^p |\hat{\beta}_k - \beta_k^*|$$

$$\text{MSE} = \frac{1}{p} \sum_{k=1}^p (\hat{\beta}_k - \beta_k^*)^2$$

首先, 在表 1 和表 2 中, 固定了样本容量  $n$  和维数  $p$  的情况下, 研究了机器容量  $m$  不同时  $\hat{\beta}$  的 MAD 和 MSE 的情况。考虑了两种设置:  $p=3, n=10^4$  和  $p=15, n=10^5$ 。

**Table 1.** MAD( $\times 10^{-2}$ ), MSE( $\times 10^{-4}$ ) of SVR-ALL, ND-SVR and DC-SVR with  $p=3, n=10^4$

**表 1.**  $p=3, n=10^4$ , SVR-ALL、ND-SVR 和 DC-SVR 的 MAD( $\times 10^{-2}$ ), MSE( $\times 10^{-4}$ )

		$m=50$		$m=100$		$m=200$	
		MAD	MSE	MAD	MSE	MAD	MSE
$N(0,1)$	SVR-ALL	1.041	2.278	1.041	2.278	1.041	2.278
	ND-SVR	1.504	4.764	1.422	3.138	1.334	2.798
	DC-SVR	1.307	2.964	1.275	2.582	1.184	2.402
$t(3)$	SVR-ALL	1.177	2.938	1.177	2.938	1.177	2.938
	ND-SVR	1.799	6.549	1.536	3.781	1.459	3.399
	DC-SVR	1.532	4.033	1.352	3.527	1.273	3.256

**Table 2.**  $MAD(\times 10^{-2})$ ,  $MSE(\times 10^{-4})$  of SVR-ALL, ND-SVR and DC-SVR with  $p=15$ ,  $n=10^5$ **表 2.**  $p=15$ ,  $n=10^5$ , SVR-ALL、ND-SVR 和 DC-SVR 的  $MAD(\times 10^{-2})$ ,  $MSE(\times 10^{-4})$ 

		$m=200$		$m=500$		$m=1000$	
		MAD	MSE	MAD	MSE	MAD	MSE
$N(0,1)$	SVR-ALL	0.331	0.189	0.331	0.189	0.331	0.189
	ND-SVR	0.450	0.333	0.388	0.251	0.373	0.241
	DC-SVR	0.406	0.257	0.374	0.232	0.352	0.192
$t(3)$	SVR-ALL	0.399	0.257	0.399	0.257	0.399	0.257
	ND-SVR	0.505	0.423	0.463	0.354	0.442	0.330
	DC-SVR	0.453	0.343	0.437	0.315	0.421	0.285

从表 1 和表 2 可以看出, 在标准正态误差分布和  $t$  误差分布下, DC-SVR 始终优于 ND-SVR 方法。当  $m$  变大时, DC-SVR 和 ND-SVR 的 MAD 和 MSE 变小。从上述表还可以看出, 与在整个数据集上计算的 SVR-ALL 相比, DC-SVR 比 ND-SVR 方法更加接近 SVR-ALL。

类似地, 在表 3 和表 4 中, 我们固定了  $p$  和  $m$ , 改变样本大小  $n$  来观察估计量  $\hat{\beta}$  的 MAD 和 MSE。同样考虑了两种设置:  $p=3$ ,  $m=100$  和  $p=15$ ,  $m=500$ 。

从表 3 和表 4 中, 同样可以看出在标准正态误差分布和  $t$  误差分布下, DC-SVR 的性能优于 ND-SVR 方法。我们还观察到, MAD 和 MSE 随着样本  $n$  的增加而减少。上述表中结果也表明 DC-SVR 与在整个数据集上计算的支持向量回归估计量 SVR-ALL 进行比较时表现良好。

**Table 3.**  $MAD(\times 10^{-2})$ ,  $MSE(\times 10^{-4})$  of SVR-ALL, ND-SVR and DC-SVR with  $p=3$ ,  $m=100$ **表 3.**  $p=3$ ,  $m=100$ , SVR-ALL、ND-SVR 和 DC-SVR 的  $MAD(\times 10^{-2})$ ,  $MSE(\times 10^{-4})$ 

		$n=10000$		$n=30000$		$n=50000$	
		MAD	MSE	MAD	MSE	MAD	MSE
$N(0,1)$	SVR-ALL	1.041	2.278	0.656	0.676	0.498	0.422
	ND-SVR	1.422	3.138	0.835	1.444	0.725	1.066
	DC-SVR	1.275	2.582	0.764	1.021	0.631	0.853
$t(3)$	SVR-ALL	1.177	2.938	0.725	1.093	0.536	0.598
	ND-SVR	1.536	3.781	0.935	1.832	0.820	1.347
	DC-SVR	1.352	3.527	0.882	1.387	0.641	0.931

**Table 4.**  $MAD(\times 10^{-2})$ ,  $MSE(\times 10^{-4})$  of SVR-ALL, ND-SVR and DC-SVR with  $p=15$ ,  $m=500$ **表 4.**  $p=15$ ,  $m=500$ , SVR-ALL、ND-SVR 和 DC-SVR 的  $MAD(\times 10^{-2})$ ,  $MSE(\times 10^{-4})$ 

		$n=20000$		$n=50000$		$n=100000$	
		MAD	MSE	MAD	MSE	MAD	MSE
$N(0,1)$	SVR-ALL	0.736	0.921	0.474	0.395	0.331	0.189
	ND-SVR	0.869	1.264	0.570	0.539	0.388	0.251
	DC-SVR	0.796	1.021	0.504	0.485	0.374	0.232
$t(3)$	SVR-ALL	0.935	1.476	0.600	0.618	0.399	0.257
	ND-SVR	0.963	1.544	0.608	0.628	0.463	0.354
	DC-SVR	0.952	1.501	0.603	0.624	0.437	0.315

## 6. 结论

本文针对海量数据集提出了一种基于分而治之的平滑支持向量回归(DC-SVR)估计方法。我们的方法解决了内存限制和计算时间的问题,其模拟研究也表明了该方法的优越性。本文我们还提出了 SVR 参数的自适应学习过程,其中最优的参数  $(\lambda, \varepsilon)$  是由每次数据自动选择的。选择  $(\lambda, \varepsilon)$  的过程是实现 DC-SVR 的一个重要步骤,目前本文采用网格搜索和  $k$ -折交叉验证相结合的方法来优化参数。然而,或许可以设计一种更有效的方法,相关的研究需要进一步的工作。

## 参考文献

- [1] Mcdonald, R., Mohri, M., Silberman, N., Walker, D. and Mann, G.S. (2009) Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models. In: *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, 1231-1239.
- [2] Zinkevich, M., Weimer, M., Li, L.H. and Smola, A.J. (2010) Parallelized Stochastic Gradient Descent. In: *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, 2595-2603.
- [3] Zhang, Y., Duchi, J.C. and Wainwright, M.J. (2013) Communication-Efficient Algorithms for Statistical Optimization. *The Journal of Machine Learning Research*, **14**, 3321-3363.
- [4] Cheyer, A.J., Guzzoni, D.R., Gruber, T.R. and Brigham, C.D. (2014) Service Orchestration for Intelligent Automated Assistant. US20130111487A1.
- [5] Jordan, M.I., Lee, J.D. and Yang, Y. (2016) Communication-Efficient Distributed Statistical Learning. arXiv:1605.07689.
- [6] Lee, J.D., Liu, Q., Sun, Y. and Taylor, J.E. (2017) Communication-Efficient Sparse Regression. *Journal of Machine Learning Research*, **18**, 1-30.
- [7] Chen, X., Lee, J.D., Li, H. and Yang, Y. (2020) Distributed Estimation for Principal Component Analysis: A Gap-Free Approach. arXiv:2004.02336.
- [8] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. <https://doi.org/10.1007/BF00994018>
- [9] Liu, Y.F., Zhang, H.H., Park, C. and Ahn, J.Y. (2007) Support Vector Machines with Adaptive  $L_q$  Penalty. *Computational Statistics & Data Analysis*, **51**, 6380-6394. <https://doi.org/10.1016/j.csda.2007.02.006>
- [10] Hsieh, C.J., Si, S. and Dhillon, I.S. (2013) A Divide-and-Conquer Solver for Kernel Support Vector Machines. arXiv:1311.0914.
- [11] Lian, H. and Fan, Z.Y. (2018) Divide-and-Conquer for Debaised  $l_1$ -Norm Support Vector Machine in Ultra-High Di-

- mensions. *Journal of Machine Learning Research*, **18**, 1-26.
- [12] Smola, A.J. and Schölkopf, B. (2004) A Tutorial on Support Vector Regression. *Statistics and Computing*, **14**, 199-222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>
- [13] Khemchandani, R., Jayadeva and Chandra, S. (2009) Regularized Least Squares Fuzzy Support Vector Regression for Financial Time Series Forecasting. *Expert System with Applications*, **36**, 132-138. <https://doi.org/10.1016/j.eswa.2007.09.035>
- [14] Rivas-Perea, P. and Cota-Ruiz, J. (2013) An Algorithm for Training a Large Scale Support Vector Machine for Regression Based on Linear Programming and Decomposition Methods. *Pattern Recognition Letters*, **34**, 439-451. <https://doi.org/10.1016/j.patrec.2012.10.026>
- [15] Cheng, A.Y., Jiang, X., Li, Y.F., Chao, Z. and Zhu, H. (2016) Multiple Sources and Multiple Measures Based Traffic Flow Prediction Using the Chaos Theory and Support Vector Regression Method. *Physica A: Statistical Mechanics and Its Applications*, **466**, 422-434. <https://doi.org/10.1016/j.physa.2016.09.041>
- [16] Maldonado, S., Gonzalez, A. and Crone, S. (2019) Automatic Time Series Analysis for Electric Load Forecasting via Support Vector Regression. *Applied Soft Computing Journal*, **83**, Article ID: 105616. <https://doi.org/10.1016/j.asoc.2019.105616>
- [17] Horowitz, J.L. (1998) Bootstrap Methods for Median Regression Models. *Econometrica*, **66**, 1327-1351. <https://doi.org/10.2307/2999619>
- [18] Pang, L., Lu, W.B. and Wang, H.J. (2012) Variance Estimation in Censored Quantile Regression via Induced Smoothing. *Computational Statistics & Data Analysis*, **56**, 785-796. <https://doi.org/10.1016/j.csda.2010.10.018>
- [19] Chen, X., Liu, W.D. and Zhang, Y.C. (2018) Quantile Regression under Memory Constraint. *Annals of Statistics*, **44**, 3244-3273. <https://doi.org/10.1214/18-AOS1777>
- [20] Wang, X., Yang, Z., Chen, X. and Liu, W. (2019) Distributed Inference for Linear Support Vector Machine. *Journal of Machine Learning Research*, **20**, 1-41.
- [21] Chen, K.Y. (2007) Forecasting Systems Reliability Based on Support Vector Regression with Genetic Algorithms. *Reliability Engineering & System Safety*, **92**, 423-432. <https://doi.org/10.1016/j.ress.2005.12.014>
- [22] Nahil, A. and Lyhyaoui, A. (2018) Short-Term Stock Price Forecasting Using Kernel Principal Component Analysis and Support Vector Machines: The Case of Casablanca Stock Exchange. *Procedia Computer Science*, **127**, 161-169. <https://doi.org/10.1016/j.procs.2018.01.111>