

基于空间投影和聚类划分的SVR加速算法

王梅^{1,2}, 张天时¹, 王志宝¹, 任怡果¹

(1. 东北石油大学 计算机科学与技术学院, 黑龙江 大庆 163318;
2. 黑龙江省石油大数据与智能分析重点实验室(东北石油大学), 黑龙江 大庆 163318)

摘要:数据不仅能产生价值,还对统计学的科学发展提供了动力。随着科技的飞速发展,海量数据得以涌现,但大规模的数据会导致很多传统处理方法很难满足各领域对数据分析的需求。面对海量数据时代学习算法的低效性,分治法通常被认为是解决这一问题最直接、最广泛使用的策略。SVR是一种强大的回归算法,在模式识别和数据挖掘等领域有广泛应用。然而在处理大规模数据时,SVR训练效率低。为此,该文利用分治思想提出一种基于空间投影和聚类划分的SVR加速算法(PKM-SVR)。利用投影向量将数据投影到二维空间;利用聚类方法将数据空间划分为 k 个互不相交的区域;在每个区域上训练SVR模型;利用每个区域的SVR模型预测落入同一区域的待识别样本。在标准数据集上与传统的划分方法进行对比实验,实验结果表明该算法训练速度较快,并表现出更好的预测性能。

关键词:大规模数据;分治法;支持向量回归;主成分分析;聚类

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2024)04-0024-06

doi:10.20165/j.cnki.ISSN1673-629X.2024.0004

An Accelerator for SVR Algorithms Based on Spatial Projection and Clustering Partitioning

WANG Mei^{1,2}, ZHANG Tian-shi¹, WANG Zhi-bao¹, REN Yi-guo¹

(1. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China;
2. Heilongjiang Key Laboratory of Petroleum Big Data and Intelligent Analysis (Northeast Petroleum University),
Daqing 163318, China)

Abstract: Data not only generates value, but also provides the impetus for the scientific development of statistics. With the rapid development of science and technology, massive data has emerged, but the large-scale data makes it difficult for many traditional processing methods to meet the needs of data analysis in various fields. Facing the inefficiency of learning algorithms in the era of massive data, partitioning is usually considered as the most direct and widely used strategy to solve this problem. SVR is a powerful regression algorithm with wide applications in the fields of pattern recognition and data mining. However, SVR is inefficient in training when dealing with large-scale data. For this reason, we propose a SVR acceleration algorithm based on spatial projection and clustering division (PKM-SVR) by utilizing the idea of partitioning. The projection vector is used to project the data into a two-dimensional space; the clustering method is used to divide the data space into k disjoint regions; the SVR model is trained on each region; and the SVR model in each region is used to predict the to-be-recognized samples that fall into the same region. Comparison experiments are conducted with the traditional data partitioning method on standard datasets, and the experimental results show that the proposed algorithm is faster to train and exhibits better prediction performance.

Key words: large-scale data; divide and rule method; support vector regression; principal components analysis; clustering

0 引言

回归分析是一项预测性建模技术,其重点在于研究自变量和因变量之间的关系,是常用于数据拟合和预测的数学模型^[1-4]。支持向量回归(Support Vector

Regression, SVR)使用核函数构建非线性回归模型,增强了学习器对非线性问题的处理能力^[5-7]。作为基于核方法的回归学习方法,SVR在处理大规模数据时存在消耗内存过大和训练速度过慢的问题。

收稿日期:2023-06-09

修回日期:2023-10-12

基金项目:国家自然科学基金项目(51774090);黑龙江省博士后科研启动资金资助项目(LBH-Q20080)

作者简介:王梅(1976-),女,博士,教授,CCF会员(33191M),研究方向为机器学习、核方法、模型选择;张天时(1998-),女,硕士研究生,研究方向为机器学习。

如何提高效率,使SVR能适用于大规模数据一直是研究重点。Virupaksha等人^[8]提出一种结合了均值漂移聚类(Mean Shift Clustering, MSC)和SVR的超声探伤预测方法。该方法利用MSC对大数据样本进行聚类,生成聚类中心,基于此进行SVR预测。相比原始的大数据集,聚类中心具有更小的容量,可显著减少执行时间。但该方法缺乏SVR参数调整,从而导致预测精度不够理想。曹卫东等人^[9]在此基础上提出一种基于MSC与海洋捕食者算法(Marine Predators Algorithm, MPA)的参数自适应SVR方法,利用MPA获得最优SVR参数组,获得较好的预测精度。Peng等人^[10]提出了孪生支持向量回归(Twin Support Vector Regression, TSVR)算法,相较于传统的SVR方法,TSVR通过解决两个较小规模的二次规划问题来寻找回归函数,从而在泛化性能和训练速度方面表现更优。在此基础上,Peng等人^[11]又提出双边移位投影孪生支持向量回归(Pair-shifted PTSVR, PPTSVR)算法和单边移位投影孪生支持向量回归(Single-shifted PTSVR, SPTSVR)算法。但现有PPTSVR算法在训练阶段未考虑不同位置样本对超平面构造的影响,以及异常点存在时降低算法拟合性能,针对该问题,徐奔业等人^[12]提出了一种加权光滑投影孪生支持向量回归算法。梁姝娜等人^[13]在已提出的平滑支持向量回归(Smooth Support Vector Regression, S-SVR)估计方法的基础上,提出了两种大规模数据集的分布式方法,分别为naive分布式平滑支持向量回归估计方法(Naive Distributed Smooth Support Vector Regression, NDS-SVR)和改进后的分布式平滑支持向量回归估计方法(Advanced Distributed Smooth Support Vector Regression, ADS-SVR)。

基于数据划分的方法被普遍认为是解决大规模数据问题的必要途径^[14-17]。近年来,研究人员不断探索分治策略,使得分治方法在机器学习领域应用十分广泛。常用的数据集划分方法主要包括随机划分和聚类划分。随机划分是指从原始数据集中无放回随机抽样,将其元素分配到不同的子集中。Zhang等人^[18]采用随机抽样的方法,将训练集分割为大小相等,互不相交的若干子集,在每个子集中分别训练模型,对于待识别的样本通过对所有模型的预测结果进行求和平均得出最终预测结果。Chang等人^[19]提出了一种分治局部平均回归的技术,其基本思想与随机划分方式类似。虽然随机划分方法操作简单快捷,但由于其具有随机性,会导致输入空间存在相交的可能性,这会影响模型的准确性和稳定性,从而限制了该方法的应用范围。此外,该方法还未考虑数据之间的相关性。因此,研究人员开始用聚类方法进行数据划分,它将原样本

集分成多个由相似样本组成的类。Tandon等人^[20]研究了一种基于核K-means的学习方法,该方法通过将输入空间进行区域化,将每个样本限制在其所属的区域中进行预测学习。该方法可以提高模型的泛化能力、减少约束条件,并且有理论支撑。Hsieh等人^[21]采用分治思想利用K-means聚类方法划分数据集来求解核支持向量机的子模型。相比于随机划分,聚类分析在考虑数据之间的近邻性和相似性方面更为准确,并且能够保持数据的几何特征。

除了这两种常见的数据划分方法,Asimov等人^[22]提出了基于投影的数据划分算法,利用Grand Tour算法将实例投影到一个向量上,然后按照投影值排序并逐段截取来分解实例集合为若干个容量近似相等且互不相交的子集。宋云胜等人^[23]结合SVM分类特点,采用核Fisher线性判别分析方法获得最优投影向量,并在每个区域上分别训练模型,对于待识别实例判断其落入区域,然后利用该区域对应的模型进行预测。刘恩江等人^[24]在文献中采用主成分分析(Principal Component Analysis, PCA)方法获得第一主成分向量作为投影向量后进行数据划分,并在每个区域上分别训练模型。这种划分方法在速度上可以近似于随机划分,大大提高了算法效率,但只是进行单一主方向的选取,会使得划分后的数据丢失部分信息。

SVR算法性能与数据的分布密切相关,利用数据划分对其加速时应促使数据分布的一致性,即数据在划分后的子集内的分布应尽量与原始数据分布保持一致。综上所述,该文提出了一种基于空间投影和聚类划分的SVR加速算法。该算法首先运用PCA获得第一、二主成分并作为投影向量向二维空间投影,接着采用K-means聚类算法将数据划分成互不相交的若干子集,在每个子集上分别训练SVR预测模型。对于待识别样本数据,首先判断其落入的子集,然后利用该子集对应的模型进行预测。该方法能最大程度上保持了数据的局部信息,并且能够提升大数据下的学习算法的执行效率。

1 预备知识

1.1 SVR 算法

令 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ 是容量为 n 的数据集,其中 $\mathbf{x}_i \in R^m$ 为第 i 个输入数据, $y_i \in R$ 表示响应变量。SVR模型的优化问题为:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \begin{cases} y_i - \mathbf{w}^T \cdot \boldsymbol{\varphi}(\mathbf{x}_i) - b \leq \varepsilon + \xi_i^* \\ \mathbf{w}^T \cdot \boldsymbol{\varphi}(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \quad (i = 1, 2, \dots, n) \end{cases} \end{aligned} \quad (1)$$

式中, \mathbf{w} 为权向量; 参数 $C > 0$ 为惩罚因子; $\varphi(x)$ 为由输入空间 x 到高维特征空间的非线性映射; b 为偏置; ε 为误差上限; ξ_i, ξ_i^* 是考虑到超出 ε 范围的数据时引入的松弛因子。引入拉格朗日乘子 a_i, a_i^* 和核函数 $k(\mathbf{x}_i, \mathbf{x}_j)$ 将上式转化成如下对偶问题进行求解, 即:

$$\max_{a_i, a_i^*} - \frac{1}{2} \sum_i \sum_{j=1}^n (a_i - a_i^*) (a_j - a_j^*) k(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon \sum_{i=1}^n y_i (a_i - a_i^*) \quad (2)$$

$$\text{s. t. } \begin{cases} \sum_{i=1}^n (a_i - a_i^*) = 0 \\ 0 \leq a_i, a_i^* \leq C \quad (i = 1, 2, \dots, n) \end{cases}$$

求解上式得到 SVR 最终目标决策函数为:

$$f(x) = \sum_{i=1}^n (a_i - a_i^*) k(\mathbf{x}_i, \mathbf{x}) + b \quad (3)$$

其中, 核函数 $k(\mathbf{x}_i, \mathbf{x}_j)$ 有多种形式, 该文使用高斯函数作为核函数, 因此待识别样本 x 预测值 y_{pre} 为:

$$y_{\text{pre}} = \sum_{i=1}^n (a_i - a_i^*) \exp \left\{ -\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2} \right\} + b \quad (4)$$

其中, σ 是高斯核的参数。

1.2 PCA 算法

主成分分析 (PCA) 是一种基于线性映射的特征提取方法。主要思想是利用一种数学变换将 n 维特征空间中的数据点映射到一个新的 m 维空间中, 其中 m 维是全新的正交特征, 也称为主成分, 由原始维特征重构而成^[25]。PCA 通过某些变换将高维数据映射到新的低维空间中, 将数据的最大方差投影在第一主成分上, 第二大方差被投影在第二主成分上, 依此类推。PCA 使用少数几个主成分就能最大限度地保留原始高维数据的信息^[26], 协方差矩阵的特征向量对应着数据的主要方向, 特征值则表示数据在该方向上的离散程度。通过选择具有最大特征值的特征向量, 就能够找到数据中方差最大的方向, 从而实现降维处理。

根据最大方差理论, 少数主成分就包含了数据 85% 以上的信息。第一、二主成分是数据特征的线性组合, 代表了原始数据的绝大部分信息。按照最大可分性, 根据这两个主成分进行划分可以使得划分后的区域差异性达到最大, 这样使得每个数据都能划分到适合的区域。为此, 该文选用前两个主成分所在的方向作为投影向量, 将数据投影到二维主成分空间中, 以最小化信息损失的同时最大化区域之间的差异性。

1.3 K-means 算法

聚类算法是一种无监督学习算法, 目的是通过比较样本数据的相似性将相似的样本自动归类到一个簇或类别中。聚类后, 同一类别的数据应尽可能划分到同一个簇或类中, 而不同类别的数据则应尽可能分离,

以使得同一类中的数据相似性最大, 不同类之间的数据差异性也最大化^[27-29]。K-means 是一种经典的聚类算法, 大体思想^[30]是将 n 个数据对象划分成 k 个簇。具体步骤如下: 随机选择 k 个数据对象作为初始质心。对于剩下的每个数据对象, 根据其各个质心的距离, 将其归入最近的簇中, 并与同簇的数据对象形成一个新的簇。重新计算每个簇的质心。迭代执行以上步骤 r 次, 直到质心位置不再移动或准则函数收敛, 即完成聚类。

2 PKM-SVR 算法

给定数据 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, 将特征空间 δ 划分为 $\{P_1, P_2, \dots, P_k\}$, $\delta = \bigcup_{i=1}^k P_i$, $P_i \cap P_j = \emptyset$, $i \neq j$, 则每个样本将属于唯一的一个划分 P_i , 每个 P_i 中的全部样本被视作一个子样本集, 由此便将原样本集划分成 k 互不相交的子集 $\{D_1, D_2, \dots, D_k\}$ 。

2.1 PKM 算法

该文采用主成分分析方法获得投影向量 $\mathbf{v}_1, \mathbf{v}_2$, 将数据空间映射至二维空间后, 运用 K-means 聚类算法把原始训练数据分割成若干训练子集, 从而使得原样本空间中邻近的数据在低维空间中依旧保持近邻关系, 并在划分后处于同一区域。PKM 算法主要分为两个步骤: 投影向量的获取和划分数据集。

2.1.1 投影向量的获取

给定 n 个 m 维数据 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, 进行数据标准化:

$$\mathbf{x}_{ij}' = \frac{\mathbf{x}_{ij} - \bar{\mathbf{x}}_j}{S_j} \quad (5)$$

式中, S_j 为标准差。

$$S_j = \sqrt{\frac{\sum_{i=1}^n (\mathbf{x}_{ij} - \bar{\mathbf{x}}_j)^2}{n-1}} \quad (6)$$

式中, $\bar{\mathbf{x}}_j = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{ij}$, $1 \leq j \leq m$ 。样本矩阵经过标准化变为 $\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_n$ 。

$$\mathbf{R}_{pq} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}'_{ip} - \bar{\mathbf{x}}_p)(\mathbf{x}'_{iq} - \bar{\mathbf{x}}_q) \quad (7)$$

式中, $\bar{\mathbf{x}}_p = \frac{1}{n} \sum_{i=1}^n \mathbf{x}'_{ip}$, $1 \leq p \leq m$; \mathbf{R}_{pq} 是第 p 个指标和第 q 个指标的相关系数, 由此得到标准化样本的协方差矩阵 Σ 为:

$$\Sigma = (\mathbf{R}_{pq})_{m \times m} \quad (8)$$

根据协方差矩阵 Σ 计算其特征值与特征向量。

$$\Sigma \mathbf{v} = \lambda \mathbf{v} \quad (9)$$

式中, λ 为特征值, \mathbf{v} 为特征向量。

对特征值从大到小排列 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$, 得到对应的特征向量 v_1, v_2, \dots, v_m , v_1, v_2 为第一、二主成分, 是下文所求的划分向量。

2.1.2 划分数据集

x_1, x_2, \dots, x_n 在划分向量 v_1, v_2 构成的投影平面 v' 投影后, 得到 z_1, z_2, \dots, z_n , 其中 $z_i = x_i \cdot v'$ 。首先要对选取的 k 个聚类中心进行初始化, 再计算每个数据点到 k 个聚类中心的距离。

$$\text{dis}(z_i, C_t) = \sqrt{\sum_{j=1}^m (z_{ij} - C_{tj})^2} \quad (10)$$

其中, C_t 表示第 t 个聚类中心, $1 \leq t \leq k$, z_{ij} 表示第 i 个数据的第 j 个属性, C_{tj} 表示第 t 个聚类中心的第 j 个属性。然后依次比较每个数据到 k 个聚类中心的距离, 将数据点划分到距离最小的聚类中心所在的簇中, 再计算每个簇中所有数据的均值并作为新的聚类的中心。

$$C_t = \frac{\sum_{z_i \in G_t} z_i}{|G_t|} \quad (11)$$

其中, G_t 表示第 t 个类簇, $|G_t|$ 表示第 t 个类簇中数据的个数。迭代停止条件是聚类中心不再变化或者达到最大的迭代次数, 最终得到 k 个类 D_1, D_2, \dots, D_k 。由此利用 K-means 划分后得到了 k 个子集, 每个子集所在区域是 D_1, D_2, \dots, D_k 。

2.2 PKM-SVR 算法

PKM-SVR 算法是运用分治思想将大规模问题转变为若干较小子问题的算法。首先将原始样本空间划分为若干子集, 在每个子集上建立相应的 SVR 预测模型, 然后识别待测样本所属的子集区域, 最后利用该区域上的 SVR 模型进行预测。算法伪代码如下:

算法 1: PKM-SVR 算法。

输入: 数据集 $T = \{(x_i, y_i)\}$, $i = 1, 2, \dots, n$, 子集个数 k , 待识别样本;

输出: 待识别样本预测值。

步骤 1: 利用 PCA 获得投影向量 v_1, v_2 构造投影平面 v' , 将数据集向二维平面投影;

步骤 2: 利用 K-means 聚类将投影后的数据划分为 k 个不相交的子集 D_1, D_2, \dots, D_k ;

步骤 3: 对每个子集分别建立 SVR 模型;

步骤 4: 判断未标记样本所属的子集, 并用该子集上的 SVR 模型进行预测。

2.3 算法时间复杂度分析

提出的 PKM-SVR 算法由数据划分和在每个子集上训练模型两部分构成。假设数据集的规模为 n , 数据的维度为 m , 划分后子集个数为 k 。数据划分算法主要分为两个阶段: 第一阶段是投影向量 v_1, v_2 的获取, 因为投影向量通过 PCA 算法得到, 故时间复杂度

是 $O(nm^2)$, 第二阶段数据集划分的时间复杂度是 $O(2kn)$ 。SVR 算法的时间复杂度是 $O(n^3)$, 在每个子集上训练 SVR 模型的时间复杂度是 $O(\sum_{i=1}^k n_i^3)$ 。该文提出的基于空间投影和聚类划分的 SVR 加速算法的时间复杂度为 $O(nm^2) + O(2kn) + O(\sum_{i=1}^k n_i^3)$ 。

3 实验分析

3.1 数据集

为验证文中算法的性能, 选取 UCI 数据库中规模较大的 4 组数据集进行实验, 其中包括蛋白质三级结构理化性质数据集 (CASP)、联合循环发电厂数据集 (CCPP)、钢铁行业能耗数据集 (Steel Industry)、超导数据集 (Superconduct), 详细信息如表 1 所示。对 4 组数据集进行归一化处理以确保结论的普适性。

表 1 实验数据集

数据集	样本数量	特征数量
CASP	45 730	9
CCPP	9 568	4
Steel Industry	35 040	11
Superconduct	21 263	81

3.2 实验方法和结果分析

本节将通过实验来验证该文提出的数据划分方法的性能。将提出的算法 PKM-SVR 与 PHP 算法和 DC 算法在 SVR 上进行对比实验。实验采用均方根误差 (RMSE) 作为评价指标, 同时比较全部方法的训练时间。实验中使用高斯径向基作为核函数, 其中 gamma 参数值和 SVR 求解中涉及参数 C 和 ε 的选取, 该文用由五折交叉验证方法找出最优解。讨论子集的个数 k 的设置, 对于三种算法, 均设置 $k = 2r$, $r = 2, 3, 4$, 便于划分数据集。为了研究数据集大小和划分子集个数对测试误差的影响, 在训练过程中改变训练集数据个数。

实验结果如表 2、表 3 和图 1 所示。

根据表 2 和图 1 的结果可以得出, 文中算法在整个数据集上的误差与 Whole-SVR 算法的相当接近, 测试精度上优于 PHP-SVR 算法和 DC-SVR 算法。文中算法在 CCPP 数据集上和其他算法无太大差异。在 CASP, Steel Industry, Superconduct 上, 文中算法均达到了 Whole-SVR 算法的精度, 而 PHP-SVR 算法和 DC 算法明显误差增大。

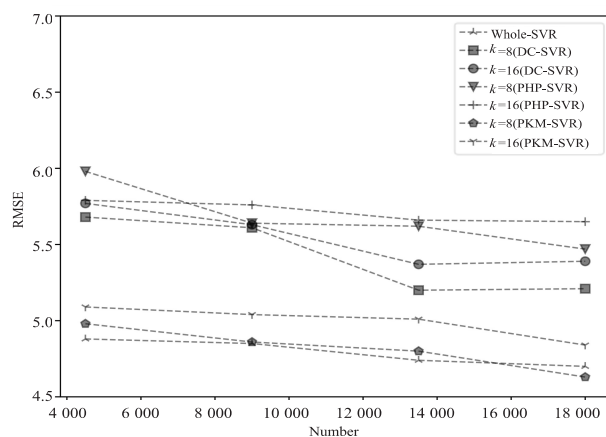
根据图 1 的结果可以得出结论: PKM-SVR 算法在前后相邻划分时表现稳定, 波动性较小。这表明该文选择的两个投影方向避免了单一方向选取所导致的数据信息丢失。

表 2 真实数据集上测试误差

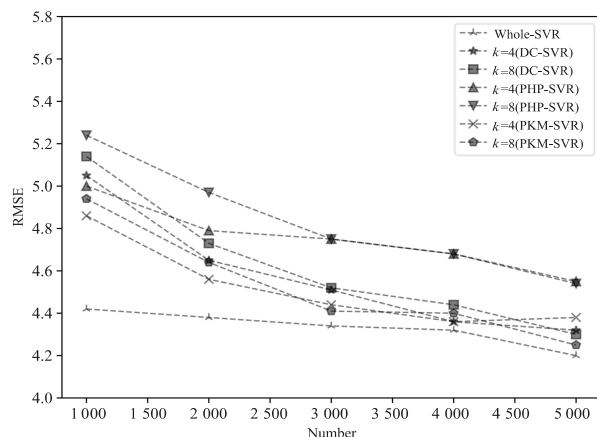
数据集	#k	Whole-SVR	PKM-SVR	PHP-SVR	DC-SVR
CASP	4		4.71	5.72	5.23
	8	4.69	4.77	5.72	5.22
	16		4.08	6.01	5.18
CCPP	4		4.34	4.99	4.40
	8	3.98	4.29	4.31	4.34
	16		4.44	4.78	4.48
Steel Industry	4		4.03	6.81	5.53
	8	4.03	4.57	8.24	5.10
	16		4.52	10.05	4.97
Superconduct	4		10.32	13.24	11.46
	8	10.07	10.08	15.38	11.16
	16		10.82	16.49	11.85

表 3 真实数据集上的训练时间 s

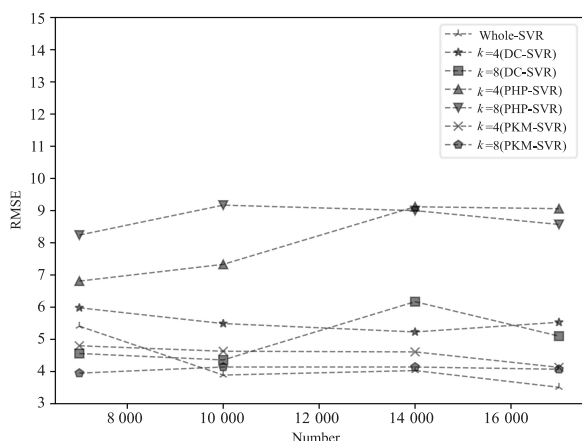
数据集	#k	Whole-SVR	PKM-SVR	PHP-SVR	DC-SVR
CASP	4		6.68	4.83	7.10
	8	51.75	3.56	2.04	3.97
	16		2.51	0.97	2.67
CCPP	4		1.16	0.89	1.23
	8	3.96	1.01	0.72	1.12
	16		1.11	0.41	1.10
Steel Industry	4		5.47	4.05	4.49
	8	39.69	2.59	1.10	2.41
	16		2.52	0.48	2.23
Superconduct	4		3.07	2.42	5.29
	8	34.13	2.85	0.86	3.11
	16		1.90	0.36	2.79



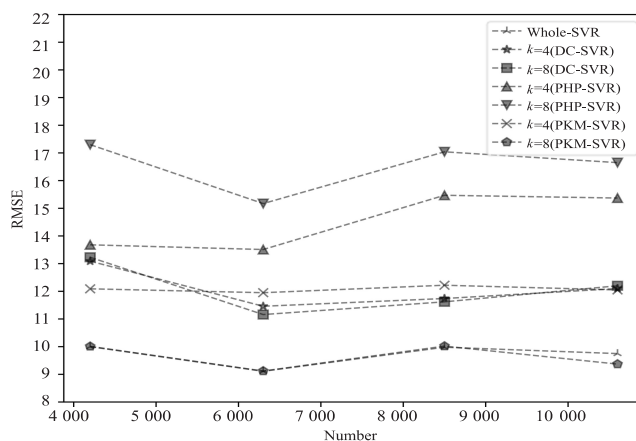
(a) CASP



(b) CCPP



(c) Steel Industry



(d) Superconduct

图 1 在 4 个数据集上的测试误差曲线

根据表 3 可知,文中算法在全部数据集上的训练时间都明显优于 Whole-SVR 算法。在全部数据集上,文中算法的训练时间都大于 PHP-SVR 算法的训练时间,分析原因可能是 PHP 算法进行数据划分时子集划分均匀且时间复杂度低。文中算法在 CASP, CCPP, Superconduct 上的训练时间优于 DC-SVR 算法,在 Steel Industry 上的训练时间略逊于 DC-SVR 算法。

4 结束语

针对 SVR 算法面对大规模数据时存在的训练效率低的问题,采用分治法思想,利用平面投影和聚类方法,将大规模数据集划分成若干个互不相交的子集,并在每个子集上训练 SVR 预测模型。该数据划分方法能最大限度地保留数据的局部性质,并通过实验证明,该算法使得 SVR 模型在真实数据上获得了很好的预

测精度,并有效缩短了训练时间。

参考文献:

- [1] DURRLEMAN S, SIMON R. Flexible regression models with cubic splines[J]. *Statistics in Medicine*, 1989, 8(5): 551–561.
- [2] PHYO P P, JEENANUNTA C. Daily load forecasting based on a combination of classification and regression tree and deep belief network[J]. *IEEE Access*, 2021, 9: 152226–152242.
- [3] WU Q, LI H, MENG F, et al. Generic proposal evaluator: a lazy learning strategy toward blind proposal quality assessment[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2017, 19(1): 306–319.
- [4] TAYLOR J W. A quantile regression neural network approach to estimating the conditional density of multiperiod returns[J]. *Journal of Forecasting*, 2000, 19(4): 299–311.
- [5] HUANG Z T, WU L L, LIU Z M. Toward wide-frequency-range direction finding with support vector regression[J]. *IEEE Communications Letters*, 2019, 23(6): 1029–1032.
- [6] SHEVADE S K, KEERTHI S S, BHATTACHARYYA C, et al. Improvements to the SMO algorithm for SVM regression[J]. *IEEE Transactions on Neural Networks*, 2000, 11(5): 1188–1193.
- [7] SCHOLKOPF B, MIKA S, BURGESS C J C, et al. Input space versus feature space in kernel-based methods[J]. *IEEE Transactions on Neural Networks*, 1999, 10(5): 1000–1017.
- [8] VIRUPAKSHAPPA K, ORUKLU E. Unsupervised machine learning for ultrasonic flaw detection using Gaussian mixture modeling, k-means clustering and mean shift clustering[C]//2019 IEEE international ultrasonics symposium (IUS). Graz: IEEE, 2019: 647–649.
- [9] 曹卫东, 倪建军, 姜博严. 支持大数据的参数自适应支持向量回归方法[J]. *计算机集成制造系统*, 2023, 29(2): 511–521.
- [10] PENG X. TSVR: an efficient twin support vector machine for regression[J]. *Neural Networks*, 2010, 23(3): 365–372.
- [11] PENG X, CHEN D. PTSVRs: Regression models via projection twin support vector machine[J]. *Information Sciences*, 2018, 435: 1–14.
- [12] 徐奔业, 顾斌杰, 潘 丰, 等. 加权光滑投影孪生支持向量回归算法[J]. *计算机工程*, 2022, 48(12): 104–111.
- [13] 梁姝娜, 张 齐. 海量数据中的分布式支持向量回归[J]. *应用数学进展*, 2022, 11(4): 1876–1889.
- [14] BOTTOU L, VAPNIK V. Local learning algorithms[J]. *Neural Computation*, 1992, 4(6): 888–900.
- [15] ZHANG Y, WAINWRIGHT M J, DUCHI J C. Communication-efficient algorithms for statistical optimization[J]. *Advances in Neural Information Processing Systems*, 2012, 25: 6792.
- [16] KRESSNER D, MASSEI S, ROBOL L. Low-rank updates and a divide-and-conquer method for linear matrix equations[J]. *SIAM Journal on Scientific Computing*, 2019, 41(2): A848–A876.
- [17] PAN Y, XIA R, YIN J, et al. A divide-and-conquer method for scalable robust multitask learning[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2015, 26(12): 3163–3175.
- [18] ZHANG Y, DUCHI J, WAINWRIGHT M. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates[J]. *The Journal of Machine Learning Research*, 2015, 16(1): 3299–3340.
- [19] CHANG X, LIN S, WANG Y. Divide and conquer local average regression[J]. *Electronic Journal of Statistics*, 2017, 11: 1326–1350.
- [20] TANDON R, SI S, RAVIKUMAR P, et al. Kernel ridge regression via partitioning[J]. *arXiv*: 1608.01976, 2016.
- [21] HSIEH C J, SI S, DHILLON I. A divide-and-conquer solver for kernel support vector machines[C]//International conference on machine learning. Beijing: IMLS, 2014: 566–574.
- [22] ASIMOV D. The grand tour: a tool for viewing multidimensional data[J]. *SIAM Journal on Scientific and Statistical Computing*, 1985, 6(1): 128–143.
- [23] SONG Y, LIANG J, WANG F. An accelerator for support vector machines based on the local geometrical information and data partition[J]. *International Journal of Machine Learning and Cybernetics*, 2019, 10: 2389–2400.
- [24] 刘恩江, 宋云胜, 梁吉业. 基于数据划分的核岭回归加速算法[J]. *中国科学技术大学学报*, 2018, 48(4): 284–289.
- [25] GOOD R P, KOST D, CHERRY G A. Introducing a unified PCA algorithm for model size reduction[J]. *IEEE Transactions on Semiconductor Manufacturing*, 2010, 23(2): 201–209.
- [26] DRAY S. On the number of principal components: a test of dimensionality based on measurements of similarity between matrices[J]. *Computational Statistics & Data Analysis*, 2008, 52(4): 2228–2237.
- [27] FAHIM A M, SALEM A M, TORKEY F A, et al. An efficient enhanced k-means clustering algorithm[J]. *Journal of Zhejiang University-Science A*, 2006, 7: 1626–1633.
- [28] 汪 敏, 武禹伯, 闵 帆. 基于多种聚类算法和多元线性回归的多分类主动学习算法[J]. *计算机应用*, 2020, 40(12): 3437–3444.
- [29] XU R, WUNSCH D. Survey of clustering algorithms[J]. *IEEE Transactions on Neural Networks*, 2005, 16(3): 645–678.
- [30] SINAGA K P, YANG M S. Unsupervised K-means clustering algorithm[J]. *IEEE Access*, 2020, 8: 80716–80727.