doi:10.3969/j.issn.1672-1292.2022.04.003

# 基于高斯噪声的孪生近端最小二乘支持向量回归模型研究及应用

袁秋云[1],张仕光[2],刘士琴[3],郭双乐[4]

(1.商丘工学院信息与电子工程学院,河南 商丘 476000)
(2.山东管理学院信息工程学院,山东 济南 250357)
(3.衡水学院数学与计算机学院,河北 衡水 053000)
(4.滨州学院 信息工程学院,山东 滨州 256600)

[摘要] 孪生近端最小二乘支持向量回归机(twin proximal least squares support vector regression,TPLSSVR)是在PLSSVR 模型的理论基础上结合 TSVR 模型的双超平面理念而设计的一种新的回归模型. 本文利用 TPLSSVR 模型框架构建了基于高斯噪声的孪生近端最小二乘支持向量回归模型. 该模型利用最小二乘方法,分别加入正则化项 $b_1^2$、$b_2^2$,将一个不等式约束问题转化为两个更简单的等式约束问题,提高了模型的泛化能力,有效提升了预测精度. 为解决模型的参数选择问题,选用收敛速度快、鲁棒性好的粒子群优化算法对模型参数进行优化选择. 将新构建的模型应用于人工数据集和风速数据集,实验结果显示该模型有较好的预测效果.

[关键词] 孪生近端最小二乘支持向量回归机,高斯噪声,风速预测,等式约束
[中图分类号]TP301　[文献标志码]A　[文章编号]1672-1292(2022)04-0019-10

# Research and Application of Twin Proximal Least Squares Support Vector Regression Model Based on Gaussian Noise

Yuan Qiuyun[1],Zhang Shiguang[2],Liu Shiqin[3],Guo Shuangle[4]

(1.School of Information and Electronic Engineering,Shangqiu Institute of Technology,Shangqiu 476000,China)
(2.School of Information Engineering,Shandong Management University,Jinan 250357,China)
(3.College of Mathematics and Computer science,Hengshui University,Hengshui 053000,China)
(4.School of Information Engineering,Binzhou University,Binzhou 256600,China)

**Abstract**:Twin proximal least squares support vector regression(TPLSSVR) is a new regression model designed on the basis of PLSSVR model and TSVR model's double hyperplane concept. In this paper,we use the above model framework to build the twin proximal least squares support vector regression model based on Gaussian noise. The least square method is introduced and the regularization terms $b_1^2$ and $b_2^2$ are added respectively. It transforms an inequality constraint problem into two simpler equality constraint problems, which not only improves the generalization ability, but also effectively improves the prediction accuracy. In order to solve the parameter selection problem of the model,the particle swarm optimization algorithm with fast convergence speed and good robustness is selected to optimize its parameters. The new model is applied to artificial data set and wind speed data set,the experimental results show that the model has better prediction effect.

**Key words**:twin proximal least squares support vector regression,Gaussian noise,wind speed prediction,equality constraint

近几十年,由于煤炭等不可再生化石燃料快速消耗,人类急需开发新的能源. 风能属于可再生的清洁能源. 我国风能开发利用较晚,但发展十分迅速,目前我国风能年发电量高达四千多亿 kW·h. 然而,由于风能的间歇性和不稳定性,风机并网后传输的电能很不稳定,给电力系统带来了很大影响. 如何有效预测

**收稿日期**:2022-08-08.
**基金项目**:山东省自然科学基金面上项目(ZR2022MF242)、河南省高等学校重点科研项目(21A520020).
**通讯作者**:郭双乐,博士,副教授,研究方向:模式识别. E-mail:guoshuangle@ aliyun.com

风速已成为当前一大难题.

传统的支持向量机(support vector machine,SVM)[1-2]是一个二元分类算法,经扩展可应用于回归问题. 但原始 SVM 需要解决大规模二次规划问题(QPP),导致计算复杂度较高,支持向量机的改进模型因此陆续出现.

最小二乘支持向量回归(least squares support vector regression,LSSVR)模型用等式约束代替不等式约束,求解线性方程组,而不是经典的二次规划问题,从而降低了计算复杂度[3]. Fung 等[4]提出了一种新的近端支持向量机(proximal support vector machine,PSVM)学习方法,对 SVM 做出改进,将两个平行超平面分别外移一段距离,使样本点集中在两个平行超平面附近,将两个平行超平面尽可能分开,最大化平面间隔. 该方法同样将不等式约束变为等式约束,相对 LSSVM,PSVM 增加了偏置项,导致目标函数具有强凸性,不仅可降低计算复杂度,训练精度也高于或等于传统的支持向量机.

为了提高 SVM 的训练速度,Jayadeva 等[5]在 SVM 的基础上提出一种新的孪生支持向量机(twin support vector machine,TSVM)机器学习方法. 与 SVM 不同,TSVM 需寻找两个非平行的分类超平面,要求其中一个超平面离一类样本尽可能地近,离另一类样本尽可能地远. Peng 等将 TSVM 推广到回归问题领域,提出了孪生支持向量回归机(twin support vector regression,TSVR)[6-7]. 很多研究不仅对目标函数提出改进,还结合不同的优化算法进行参数调整,以寻求最优解[8-15].

近端支持向量回归(proximal support vector regression,PSVR)模型训练效果高效,但泛化性能和预测精度不高. 本文在 PSVR 基础上,利用 TSVR 双超平面原理,提出一种孪生近端最小二乘支持向量回归(TPLSSVR)模型,将一个不等式约束问题变成两个更简单的等式约束问题,以降低计算复杂度,有效增强训练精度.

由于 PSVM 对噪声和异常值敏感,导致泛化性能和鲁棒性较差,很多研究者在提高鲁棒性方面做出了改进. Wang 等[16]提出一种基于最大熵原则的近端支持向量回归模型(robust proximal support vector regression based on maximum correntropy criterion,RPSVR-MCC),试图抑制离群值的负面影响,增强 PSVR 的鲁棒性.

在实际应用中,首先需确定噪声类型,根据贝叶斯定理及极大似然估计法,推导出损失函数. 张仕光等[17]提出一种基于不等式约束的高斯噪声特性区间 $v$-支持向量机,可有效预报某个目标值的区间值. 本文将该模型应用到风速预测方面,提出了一种基于高斯噪声的孪生近端最小二乘支持向量模型.

# 1　相关工作

给定一组训练数据集 $T=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}$,$(i=1,\cdots,n)$. 其中,$\boldsymbol{x}_i\in\mathbf{R}^n$ 为输入的特征向量;$y_i$ 为对应的目标值;$n$ 为数据集的大小.

## 1.1　孪生支持向量回归机 TSVR

与 SVR 不同,TSVR 训练数据点两侧产生一对不平行的函数,一个是 $\varepsilon$ 不敏感上界函数,另一个是 $\varepsilon$ 不敏感下界函数. TSVR 需求解两个凸二次规划问题,每个凸规划问题只有一个约束条件,而 SVR 有两个约束条件,使得 TSVR 比 SVR 运算速度更快. TSVR 的原问题为:

$$\min\left\{P_{\mathrm{TSVR}}=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{e}\varepsilon_1-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_1+\boldsymbol{e}b_1))^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{e}\varepsilon_1-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_1+\boldsymbol{e}b_1))+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}\right\},$$
$$\text{s.t.}\quad \boldsymbol{y}-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_1+\boldsymbol{e}b_1)\geqslant\boldsymbol{e}\varepsilon_1-\boldsymbol{\xi},\boldsymbol{\xi}\geqslant 0; \tag{1}$$

$$\min\{P_{\mathrm{TSVR}}=\frac{1}{2}(\boldsymbol{y}+\boldsymbol{e}\varepsilon_2-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_2+\boldsymbol{e}b_2))^{\mathrm{T}}(\boldsymbol{y}+\boldsymbol{e}\varepsilon_2-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_2+\boldsymbol{e}b_2))+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}\},$$
$$\text{s.t.}\quad (\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_2+\boldsymbol{e}b_2)-\boldsymbol{y}\geqslant\boldsymbol{e}\varepsilon_2-\boldsymbol{\eta},\boldsymbol{\eta}\geqslant 0; \tag{2}$$

式中,$\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})$元素是 $\boldsymbol{K}_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_i^{\mathrm{T}})$ 的核矩阵;$\boldsymbol{e}$ 为适量大小的单位矩阵;$\boldsymbol{\xi}$ 和 $\boldsymbol{\eta}$ 为松弛变量;$\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})=\varphi(\boldsymbol{x}_i)^{\mathrm{T}}\varphi(\boldsymbol{x}_j)$是 $\varphi(\boldsymbol{x}_i)$和 $\varphi(\boldsymbol{x}_j)$在高维向量空间的内积. $\varphi(\boldsymbol{x}_i)$是非线性映射,从低维空间映射到高维空间.

利用拉格朗日乘子法,可将 TSVR 的对偶问题构造如下:

$$\min\{D_{\mathrm{TSVR}}=\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{g}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\alpha}\},$$
$$\text{s.t.}\quad 0\leqslant\boldsymbol{\alpha}\leqslant C_1\boldsymbol{e}; \tag{3}$$

$$\min\{D_{\mathrm{TSVR}}=\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\beta}-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\beta}+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\beta}\},$$
$$\text{s.t.}\quad 0\leqslant\boldsymbol{\beta}\leqslant C_2\boldsymbol{e}; \tag{4}$$

式中,$\boldsymbol{H}=[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}}),\boldsymbol{e}]$,$\boldsymbol{g}=\boldsymbol{y}-\boldsymbol{e}\varepsilon_1$,$\boldsymbol{h}=\boldsymbol{y}+\boldsymbol{e}\varepsilon_2$.

当得到 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 的最优解时,最优权重 $\boldsymbol{\omega}_1$、$\boldsymbol{\omega}_2$ 和偏置项 $\boldsymbol{b}_1$、$\boldsymbol{b}_2$ 如下:

$$(\boldsymbol{\omega}_1,\boldsymbol{b}_1)^{\mathrm{T}}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{\alpha}),$$
$$(\boldsymbol{\omega}_2,\boldsymbol{b}_2)^{\mathrm{T}}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{h}-\boldsymbol{\beta});$$

可得下界函数 $f_1(\boldsymbol{x})$ 和上界函数 $f_2(\boldsymbol{x})$:

$$f_1(\boldsymbol{x})=\boldsymbol{\omega}_1^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_1,$$
$$f_2(\boldsymbol{x})=\boldsymbol{\omega}_2^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_2;$$

及预测函数:

$$f_{\mathrm{TSVR}}(\boldsymbol{x})=\frac{f_1(\boldsymbol{x})+f_2(\boldsymbol{x})}{2}=\frac{1}{2}(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\frac{1}{2}(\boldsymbol{b}_1+\boldsymbol{b}_2).$$

式(1)的第一项是全部训练样本到 $f_1(\boldsymbol{x})+\boldsymbol{\varepsilon}_1$ 的训练误差的二范式,第二项为下界函数下面的训练样本到 $f_1(\boldsymbol{x})$ 的训练误差,说明下界函数下方的训练样本对模型的影响较大,所以被训练了两次,而下界函数上方的训练样本只训练一次. 同理可描述上界函数上方的训练样本对模型的影响.

将 TSVR 的不等式约束替换为等式约束,直接在原空间对带有等式约束的二次规划问题进行求解,而不是在对偶空间求解,这一方法被称为孪生最小二乘支持向量回归机(twin least squares support vector regression,TLSSVR). 模型 TLSSVR 的原问题为:

$$\min\left\{P_{\mathrm{TLSSVR}}=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{e}\varepsilon_1-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_1+\boldsymbol{e}\boldsymbol{b}_1))^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{e}\varepsilon_1-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_1+\boldsymbol{e}\boldsymbol{b}_1))+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}\right\},$$
$$\text{s.t.}\quad \boldsymbol{y}-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_1+\boldsymbol{e}\boldsymbol{b}_1)=\boldsymbol{e}\varepsilon_1-\boldsymbol{\xi},\boldsymbol{\xi}\geqslant 0;$$
$$\min\{P_{\mathrm{TLSSVR}}=\frac{1}{2}(\boldsymbol{y}+\boldsymbol{e}\varepsilon_2-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_2+\boldsymbol{e}\boldsymbol{b}_2))^{\mathrm{T}}(\boldsymbol{y}+\boldsymbol{e}\varepsilon_2-(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_2+\boldsymbol{e}\boldsymbol{b}_2))+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}\},$$
$$\text{s.t.}\quad (\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{\omega}_2+\boldsymbol{e}\boldsymbol{b}_2)-\boldsymbol{y}=\boldsymbol{e}\varepsilon_2-\boldsymbol{\eta},\boldsymbol{\eta}\geqslant 0.$$

TLSSVR 只需求解两个较小规模的线性方程组就能得到最终的回归函数,大大降低了计算复杂度.

### 1.2 近端最小二乘支持向量回归机

支持向量分类模型建立在两类样本之间,使用带有最大间隔的两个平行超平面来确定最优解. 对于比较复杂的问题,两个平行超平面的间隔太小导致难以获取最优解. PSVM 让两个平行超平面分别外移一段距离[4],使样本点集中在两个平行超平面附近,再将两个平行超平面尽可能分开,$(\boldsymbol{\omega}\cdot\boldsymbol{x})+\boldsymbol{b}=\pm 1$ 不再是边界平面,而变成了“近端”平面,对于方向 $\boldsymbol{\omega}$ 和相对位置 $\boldsymbol{b}$,边界平面之间的间距最大化,两个超平面间隔改为 $\dfrac{2}{\left\|\begin{bmatrix}\boldsymbol{\omega}\\ \boldsymbol{b}\end{bmatrix}\right\|}$.

文献[18]将近端支持向量分类机拓展为近端最小二乘支持向量回归(proximal least squares support vector regression,PLSSVR)模型,在目标函数中添加了偏置项 $\boldsymbol{b}^2$,使对应的优化问题转化为严格的凸二次规划,且能够求出解析解,大大提高了训练速度. 模型 PLSSVR 的原问题为:

$$\min_{\omega,b}\left\{P_{\mathrm{PLSSVR}}=\frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{1}{2}\boldsymbol{b}^2+\frac{C}{2}\sum_{i=1}^{n}\boldsymbol{\xi}_i^2\right\},$$
$$\text{s.t.}\quad \boldsymbol{y}_i-\boldsymbol{\omega}^{\mathrm{T}}\varphi(\boldsymbol{x}_i)-\boldsymbol{b}=\boldsymbol{\xi}_i,(i=1,\cdots,n); \tag{5}$$

式中,$\boldsymbol{\xi}_i$ 为松弛变量;$C$ 为惩罚参数;$\varphi(\boldsymbol{x}_i)$ 为欧式空间 $\mathbf{R}^n$ 到内积空间 $\boldsymbol{H}$ 的映射. 构造拉格朗日泛函 $L(\boldsymbol{\omega},\boldsymbol{b},\boldsymbol{\xi}_i,\boldsymbol{\alpha}_i)$ 如下:

$$L(\boldsymbol{\omega},\boldsymbol{b},\boldsymbol{\xi}_i,\boldsymbol{\alpha}_i)=\frac{1}{2}\|\boldsymbol{\omega}\|^2+\frac{1}{2}\boldsymbol{b}^2+\frac{C}{2}\sum_{i=1}^{n}\boldsymbol{\xi}_i^2+\sum_{i=1}^{n}\boldsymbol{\alpha}_i(\boldsymbol{y}_i-\boldsymbol{\omega}^{\mathrm{T}}\varphi(\boldsymbol{x}_i)-\boldsymbol{b}-\boldsymbol{\xi}_i), \tag{6}$$

式中,$\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\cdots,\alpha_n]^{\mathrm{T}}$ 是拉格朗日乘子向量.

利用最优化理论可求得模型 PLSSVR 的对偶问题为:

$$\max_{\alpha}\left\{D_{\mathrm{PLSSVR}}=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j-\frac{1}{2C}\sum_{i=1}^{n}\boldsymbol{\alpha}_i^2+\sum_{i=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{y}_i\right\},\tag{7}$$

为简洁起见,消去向量 $\boldsymbol{\omega}$ 和 $\boldsymbol{\xi}_i$,得到以下线性方程组:

$$\begin{aligned}&\left(\boldsymbol{K}+\frac{1}{C}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}=\boldsymbol{y},\\&\boldsymbol{b}=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}.\end{aligned}\tag{8}$$

式中,$\boldsymbol{I}_n$ 表示 $n\times n$ 的单位矩阵;$\boldsymbol{e}=(1,1,\cdots,1)^{\mathrm{T}}$;$\boldsymbol{K}=(\boldsymbol{K}_{ij})_{n\times n}$ 表示核函数. 求解出 $\boldsymbol{\alpha}$ 和 $\boldsymbol{b}$,PLSSVR 的决策函数即可表示为:

$$f(\boldsymbol{x})=\boldsymbol{\omega}^{\mathrm{T}}\cdot\varphi(\boldsymbol{x}_i)+\boldsymbol{b}=\sum_{i=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})+\boldsymbol{b}.$$

PSVM 通过使间隔实现最大化来得到划分平面,平面 $\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{b}=\pm1$ 不再是两类点分布的边界,而变成了两类点各自的聚类中心. 两个平面由于最优化问题中的$(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}+\boldsymbol{b}^2)$而尽量拉开距离,使得训练集能够被更好地分开.

## 2　高斯噪声特性孪生近端最小二乘支持向量回归模型

文献[19]利用黑龙江发电厂的风速数据对风速分布进行统计分析,可知风速预测噪声服从高斯分布. 高斯分布的概率密度函数为 $P(\boldsymbol{\xi}_i)=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{\xi_i^2}{2}}$. 一般地,利用最大似然估计方法可得噪声分布的最优损失函数为 $C(\boldsymbol{\xi}_i)=-\log P(\boldsymbol{\xi}_i)$,故高斯噪声的损失函数为 $C(\boldsymbol{\xi}_i)=\frac{1}{2}\boldsymbol{\xi}_i^2$.

TSVR 模型是将训练数据集分成两部分,分别选择合适的损失函数,构建出两个不平行的决策函数,模型复杂度低、泛化性能好. 模型 PLSSVR 只需求解两组线性等式即可得到结果,而不需要求解两个带约束的二次规划问题,大大降低了计算消耗,提高了模型求解速度. 本文结合 TSVR 和 PLSSVR 的优势,将其模型结构引入到孪生近端最小二乘支持向量回归模型中,研究基于高斯噪声的孪生近端最小二乘支持向量回归模型(twin proximal least squares support vector regression model based on gaussian noise,TPLSSVR).

### 2.1　基于高斯噪声的孪生近端最小二乘支持向量回归机

应用统计学习理论,构造 TPLSSVR 的下界非线性回归函数 $f_1(\boldsymbol{x})=\boldsymbol{\omega}_1^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_1$ 和上界非线性回归函数 $f_2(\boldsymbol{x})=\boldsymbol{\omega}_2^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_2$. 一般地,数据集中是带有噪声的,回归函数 $f(\boldsymbol{x})$ 是未知的,一般方法是最小化目标函数:$H[f]=\lambda\|f\|_{\boldsymbol{K}}^2+\sum_{i=1}^{n}C(\boldsymbol{\xi}_i)$,$C(\boldsymbol{\xi}_i)$为损失函数.

TPLSSVR 的原问题可形式化为:

$$\min_{\boldsymbol{\omega},\boldsymbol{b}}\left\{P_{\mathrm{TPLSSVR}}=\frac{1}{2}\|\boldsymbol{\omega}_1\|^2+\frac{1}{2}\boldsymbol{b}_1^2+\frac{C_1}{2}\sum_{i=1}^{n}\boldsymbol{\xi}_i^2\right\},\tag{9}$$

$$\text{s.t. }\boldsymbol{y}_i=\boldsymbol{\omega}_1^{\mathrm{T}}\varphi(\boldsymbol{x}_i)+\boldsymbol{b}_1-\boldsymbol{\xi}_i-\boldsymbol{\varepsilon}_1,(i=1,\cdots,n);$$

$$\min_{\boldsymbol{\omega},\boldsymbol{b}}\left\{P_{\mathrm{TPLSSVR}}=\frac{1}{2}\|\boldsymbol{\omega}_2\|^2+\frac{1}{2}\boldsymbol{b}_2^2+\frac{C_2}{2}\sum_{i=1}^{n}\boldsymbol{\xi}_i^{*2}\right\},\tag{10}$$

$$\text{s.t. }\boldsymbol{y}_i=\boldsymbol{\omega}_2^{\mathrm{T}}\varphi(\boldsymbol{x}_i)+\boldsymbol{b}_2+\boldsymbol{\xi}_i^{*}+\boldsymbol{\varepsilon}_2,(i=1,\cdots,n);$$

式中,$\boldsymbol{\varepsilon}_1$、$\boldsymbol{\varepsilon}_2\geqslant0$,为临界参数;$C_1$、$C_2>0$,为惩罚参数;$\boldsymbol{\xi}_i$、$\boldsymbol{\xi}_i^{*}$ 为松弛变量.

在式(9)中引入拉格朗日函数 $L(\boldsymbol{\omega}_1,\boldsymbol{b}_1,\boldsymbol{\xi}_i,\boldsymbol{\alpha}_i)$得:

$$L(\boldsymbol{\omega}_1,\boldsymbol{b}_1,\boldsymbol{\xi}_i,\boldsymbol{\alpha}_i)=\frac{1}{2}\|\boldsymbol{\omega}_1\|^2+\frac{1}{2}\boldsymbol{b}_1^2+\frac{C_1}{2}\sum_{i=1}^{n}\boldsymbol{\xi}_i^2+\sum_{i=1}^{n}\boldsymbol{\alpha}_i(\boldsymbol{y}_i-\boldsymbol{\omega}_1^{\mathrm{T}}\varphi(\boldsymbol{x}_i)-\boldsymbol{b}_1+\boldsymbol{\xi}_i+\boldsymbol{\varepsilon}_1).\tag{11}$$

为使 $L(\boldsymbol{\omega}_1,\boldsymbol{b}_1,\boldsymbol{\xi}_i,\boldsymbol{\alpha}_i)$ 最小,利用凸优化理论分别对 $\boldsymbol{\omega}_1,\boldsymbol{b}_1,\boldsymbol{\xi}_i,\boldsymbol{\alpha}_i$ 求偏导. 根据 KKT(Karush-Kuhn-Tucker)条件:

$$\nabla_{\omega_1}(L)=0,\ \nabla_{b_1}(L)=0,\ \nabla_{\xi_i}(L)=0,\ \nabla_{\alpha_i}(L)=0, \tag{12}$$

得:

$$\begin{cases}\dfrac{\partial L}{\partial \boldsymbol{\omega}_1}=0\rightarrow \boldsymbol{\omega}_1=\sum\limits_{i=1}^{n}\boldsymbol{\alpha}_i\varphi(\boldsymbol{x}_i),\\ \dfrac{\partial L}{\partial \boldsymbol{b}_1}=0\rightarrow \boldsymbol{b}_1=\sum\limits_{i=1}^{n}\boldsymbol{\alpha}_i,\\ \dfrac{\partial L}{\partial \boldsymbol{\xi}_i}=0\rightarrow \boldsymbol{\xi}_i=-\dfrac{1}{C_1}\boldsymbol{\alpha}_i,\\ \dfrac{\partial L}{\partial \boldsymbol{\alpha}_i}=0\rightarrow \boldsymbol{y}_i-\boldsymbol{\omega}_1^{\mathrm{T}}\varphi(\boldsymbol{x}_i)-\boldsymbol{b}_1+\boldsymbol{\xi}_i+\boldsymbol{\varepsilon}_1=0.\end{cases} \tag{13}$$

将式(13)的前3个等式代入第四个等式,可得到下界回归方程组为:

$$\begin{aligned}&\left(\boldsymbol{K}+\frac{1}{C_1}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}=\boldsymbol{y}+\boldsymbol{\varepsilon}_1\boldsymbol{e},\\ &\boldsymbol{b}_1=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}.\end{aligned} \tag{14}$$

式中,$\boldsymbol{I}_n$ 表示 $n\times n$ 的单位矩阵;$\boldsymbol{e}=(1,1,\cdots,1)^{\mathrm{T}}$;$\boldsymbol{K}=(\boldsymbol{K}_{ij})_{n\times n}$表示核函数. 求解方程组(14)可得 $\boldsymbol{\alpha}$ 和 $\boldsymbol{b}_1$,进而可求出下界回归函数:

$$f_1(\boldsymbol{x})=\boldsymbol{\omega}_1^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_1=\sum_{i=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})+\boldsymbol{b}_1.$$

同理可求出上界非线性回归函数:

$$f_2(\boldsymbol{x})=\boldsymbol{\omega}_2^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_2=\sum_{i=1}^{n}\boldsymbol{\alpha}_2^{*}\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})+\boldsymbol{b}_2.$$

即可得:

$$f_{\mathrm{TPLSSVR}}(\boldsymbol{x})=\frac{f_1(\boldsymbol{x})+f_2(\boldsymbol{x})}{2}.$$

模型 TPLSSVR 是将问题简化为求解一个对称正定矩阵的逆,这大大降低了计算复杂度,且其核函数不需要满足 Mercer 正定条件. 因此该模型的学习速度更快,泛化能力较强.

### 2.2 算法设计

模型的性能主要依赖于选择合适的核函数和惩罚参数,若模型参数选取不当,将极大影响模型预测效果. 粒子群算法(PSO)是根据鸟类觅食活动模拟出的一种随机搜索算法,在动态目标寻优和多维空间函数寻优等方面有着鲁棒性好、训练速度快、寻优效果好等优点. 本文采用粒子群算法对模型 TPLSSVR 的参数进行寻优. PSO 初始化为一群随机粒子,通过迭代找到最优解. 在每一次的迭代中,粒子通过跟踪两个“极值”(*pbest*,*gbest*)来更新自己. 找到这两个最优值后,粒子按下式来更新自己的速度和位置:

$$\begin{aligned}&v_i=v_i+c_1\times\mathrm{rand}(\ )\times(pbest_i-x_i)+c_2\times\mathrm{rand}(\ )\times(gbest_i-x_i),\\ &x_i=x_i+v_i.\end{aligned} \tag{15}$$

式中,$i=1,2,3,\cdots,L$,$L$ 是此群中粒子的总数;$v_i$ 是粒子的速度;rand( ) 表示介于(0,1)之间的随机数;$x_i$ 是粒子的当前位置;$c_1$ 和 $c_2$ 是学习因子.

为了提高 PSD 算法的收敛性能和避免算法陷入局部最优,在式(15)的基础上增加了惯性权重,用以描述粒子上一代速度对当前速度的影响:

$$v_i=w\cdot v_i+c_1\times\mathrm{rand}(\ )\times(pbest_i-x_i)+c_2\times\mathrm{rand}(\ )\times(gbest_i-x_i), \tag{16}$$

式中,$w\geqslant 0$ 是惯性因子,当其值较大时,全局寻优能力强,局部寻优能力弱;当其值较小时,全局寻优能力弱,局部寻优能力强. 动态 $w$ 比静态寻优效果好.

$$w^{(t)}=\frac{(w_{\mathrm{st}}-w_{\mathrm{end}})(k-g)}{k}+w_{\mathrm{end}},$$

式中,$k$ 为最大迭代次数;$w_{\mathrm{st}}$为初始惯性权值;$w_{\mathrm{end}}$为迭代至最大次数时的惯性权值.

利用粒子群算法优化 TPLSSVR 模型的参数步骤如下:

**步骤 1**　为了降低参数选择的复杂度,令 $C_1=C_2$、$r_1=r_2$,将每一组模型参数($C$、$r$、$\varepsilon_1$、$\varepsilon_2$)的组合当成一个粒子,其中第 $i$ 个粒子的位置表示为 $X_i=(x_{i1},x_{i2},x_{i3},x_{i4})$,$x_{i1}$、$x_{i2}$、$x_{i3}$、$x_{i4}$分别代表 TPLSSVR 模型的惩罚参数 $C$、$r$、$\varepsilon_1$、$\varepsilon_2$. 初始化粒子群的种群规模、惯性因子 $w$ 的变化范围、学习率和最大迭代次数.

**步骤 2**　将初始化得到的粒子代入到 TPLSSVR 模型中,以平均绝对误差(MAE)作为粒子群算法的适应值函数:$\mathrm{MAE}=\dfrac{1}{l}\sum\limits_{i=1}^{l}(|y_i-y_i^*|)$,其中,$y_i$ 是真实数据,$y_i^*$ 是预测结果,初始适应度值 $\mathrm{fit}(0)=1\ 000$.

**步骤 3**　根据式(16)更新粒子的速度与位置,计算当前粒子的适应度值. 若小于最好位置的适应度值,则更新个体极值;若所有粒子最好位置的适应度值低于全局最好位置的适应度值,则更新全局极值,并记录获得全局最优值时粒子的位置及参数组合中每个参数的取值.

**步骤 4**　判断迭代次数是否达到最大迭代次数,若迭代完成,即可得到最优参数组合;若迭代未完成,则返回步骤 3.

**步骤 5**　迭代结束,得到最优参数组合,并将参数代入到 TPLSSVR 模型中.

算法 TPLSSVR-PSO 如下所示:

**算法 TPLSSVR-PSO**

1. 输入数据集,分配训练数据集和测试数据集.
2. 初始化:令 $t=1$,初始化 PSO 参数(粒子数目、寻优参数个数、最大迭代次数、学习因子 $C_1$、$C_2$)和适应度值.
3. 根据式(9)、(10),求解$\left(\boldsymbol{K}+\dfrac{1}{C_1}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}=y-\boldsymbol{\varepsilon}_1\boldsymbol{e}$、$\left(\boldsymbol{K}+\dfrac{1}{C_2}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}^*=\boldsymbol{y}+\boldsymbol{\varepsilon}_2\boldsymbol{e}$,得到拉格朗日乘子序列 $\boldsymbol{\alpha}^t$、$\boldsymbol{\alpha}^{*t}$,并获得索引集 $S_1$ 和 $S_2$,其元素分别对应于 $\boldsymbol{\alpha}_i^t<0$、$\boldsymbol{\alpha}_i^{*t}<0$.
4. 令 $V_i=\begin{cases}\dfrac{1}{2\sigma^2}, & \boldsymbol{\alpha}_i^t<0\\ 1, & \boldsymbol{\alpha}_i^t\geqslant 0\end{cases}$、$V_i^*=\begin{cases}\dfrac{1}{2\sigma^{*2}}, & \boldsymbol{\alpha}_i^{*t}\geqslant 0;\\ 1, & \boldsymbol{\alpha}_i^{*t}<0.\end{cases}$
5. 将 $V_i$ 和 $V_i^*$ 分别代入到$\left(\boldsymbol{K}+\dfrac{1}{C_1*V_i}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}^{t+1}=\boldsymbol{y}-\boldsymbol{\varepsilon}_1\boldsymbol{e}$ 和$\left(\boldsymbol{K}+\dfrac{1}{C_2*V_i^*}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}^{*t+1}=\boldsymbol{y}+\boldsymbol{\varepsilon}_2\boldsymbol{e}$ 中,得到 $\boldsymbol{\alpha}^{t+1}$ 和 $\boldsymbol{\alpha}^{*t+1}$,利用 $\boldsymbol{b}_1=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}^{t+1}$ 和 $b_2=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}^{*t+1}$,求得 $\boldsymbol{b}_1^{t+1}$ 和 $\boldsymbol{b}_2^{t+1}$.
6. 得到下界函数和上界函数:

$$f_1(\boldsymbol{x})=\boldsymbol{\omega}_1^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_1=\sum_{i=1}^{n}\boldsymbol{\alpha}_i\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})+\boldsymbol{b}_1,$$

$$f_2(\boldsymbol{x})=\boldsymbol{\omega}_2^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{X},\boldsymbol{x})+\boldsymbol{b}_2=\sum_{i=1}^{n}\boldsymbol{\alpha}_i^*\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})+\boldsymbol{b}_2.$$

7. 得到 $f_{\text{预}}(\boldsymbol{x})=\dfrac{f_1(\boldsymbol{x})+f_2(\boldsymbol{x})}{2}$,计算适应度值,利用粒子群算法更新粒子的速度和位置.
8. 判断迭代次数是否达到最大迭代次数,若迭代完成,即可得到最优参数组合;若迭代未完成,则返回步骤 3.
9. 将最优参数代入到 TPLSSVR 模型中,输出预测值.

初始化参数 $V_i=\begin{cases}\dfrac{1}{2\sigma^2}, & \boldsymbol{\alpha}_i^t<0\\ 1, & \boldsymbol{\alpha}_i^t\geqslant 0\end{cases}$,$\sigma$ 表示方差. $V_i$ 可视为权重,用于加权松弛误差 $\boldsymbol{\xi}_i$,当权重因子 $V_i$ 越大时,松弛误差 $\boldsymbol{\xi}_i$ 越小. 首先需利用 PLSSVR 模型估计样本的大概位置,了解哪些样本位于下界函数的下面,哪些位于上界函数的上面,然后将 $V_i$ 插入到等式$\left(\boldsymbol{K}+\dfrac{1}{C_1*V_i}\boldsymbol{I}_n+\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\boldsymbol{\alpha}^{t+1}=\boldsymbol{y}-\boldsymbol{\varepsilon}_1\boldsymbol{e}$ 中,从而获得一个新的 $\boldsymbol{\alpha}^{t+1}$,更新 $\boldsymbol{b}_1^{t+1}$ 的同时得到上下限函数.

## 3　实验分析

本文将提出的模型应用于人工数据集和风速数据集中,以验证 TPLSSVR 的预测效果,同时将 TPLSSVR 与 PLSSVR、LSSVR 及 TLSSVR 进行仿真实验对比. 以上模型实验均在 Windows10(内存 8G)和 python3.8 的环境中进行,所有算法参数的选取均从人工数据集、风速数据集中随机抽取 50%作为训练集,利用粒子群算法搜索模型的最优参数. 本文模型中的核函数采用高斯核函数:$\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\mathrm{e}^{-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2/2r^2}$.

本实验通过平均绝对误差 MAE、均方根误差 RMSE、误差平方和 SSE 与总平方和 SST 之间的比率(SSE/SST)、偏差平方和 SSR 与总平方和 SST 之间的比率(SSR/SST)来评判算法的好坏,计算公式如下:

$$\mathrm{MAE}=\frac{1}{l}\sum_{i=1}^{l}(|y_i-y_i^*|),$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{l}\sum_{i=1}^{l}(y_i-y_i^*)^2},$$

$$\mathrm{SSE}=\sum_{i=1}^{l}(y_i-y_i^*)^2,$$

$$\mathrm{SST}=\sum_{i=1}^{l}(y_i-\bar{y})^2,$$

$$\mathrm{SSR}=\sum_{i=1}^{l}(y_i^*-\bar{y})^2,$$

$$\frac{\mathrm{SSE}}{\mathrm{SST}}=\frac{\sum_{i=1}^{l}(y_i-y_i^*)^2}{\sum_{i=1}^{l}(y_i-\bar{y})^2},$$

$$\frac{\mathrm{SSR}}{\mathrm{SST}}=\frac{\sum_{i=1}^{l}(y_i^*-\bar{y})^2}{\sum_{i=1}^{l}(y_i-\bar{y})^2}.$$

式中,$l$ 为样本的大小;$y_i$ 为真实数据;$y_i^*$ 为预测结果;$\bar{y}=\dfrac{\sum_{i=1}^{l}y_i}{l}$ 是 $y_1,y_2,y_3,\cdots,y_l$ 的平均值.

RMSE 和 MAE 越小,表明拟合性能越好,两者都是关于测量实际值和预测值之间的偏差. 在大多数情况下,小的 SSE/SST 意味着估计和实际值之间的良好一致性,而获得较小的 SSE/SST 通常伴随着 SSR/SST 的增加. 然而,SSE/SST 的极小值实际上并不好,因为其可能意味着回归器的过度拟合. 因此,一个好的估计器应该在 SSE/SST 和 SSR/SST 之间取得平衡.

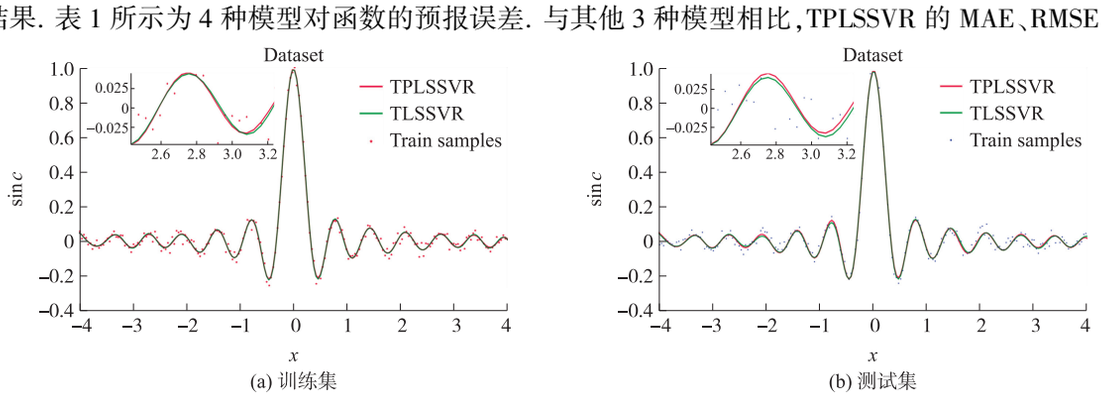
### 3.1 人工数据集

本文在函数 sinc 上测试了所提出算法的性能. 函数 sinc 的表达式为:

$$y_i=\mathrm{sinc}(x_i)=\frac{\sin(\pi x_i)}{\pi x_i}+\xi_i,x_i\in[-4,4],\xi_i\in N(0,0.15^2),\tag{17}$$

式中,$\xi_i$ 表示受污染数据集的噪声;$N(0,0.15^2)$ 表示噪声 $\xi_i$ 遵循均值为 0,方差为 $0.15^2$ 的高斯随机分布. 根据函数 sinc 随机生成 200 个测试样本和 200 个训练样本,分别采用 4 种模型对数据集进行预测.

图 1 所示为 TPLSSVR 和 TLSSVR 对函数的预测效果,其中(a)为训练集的预测结果,(b)为测试集的预测结果. 表 1 所示为 4 种模型对函数的预报误差. 与其他 3 种模型相比,TPLSSVR 的 MAE、RMSE、SSE、




(a) 训练集　　(b) 测试集

**图 1　TPLSSVR、TLSSVR 对函数的预测**

**Fig. 1　Prediction of function by TPLSSVR and TLSSVR**

SSE/SST 都相对较小,SSR/SST 相对较大,说明 TPLSSVR 的预测效果比其他 3 种模型的预测效果更好.

**表 1　4 种模型对函数 sinc 的预报误差**

**Table 1　Prediction of sinc function by four models**

| 模型 | MAE | RMSE | SSE | SSE/SST | SSR/SST | 时间/s |
|---|---|---|---|---|---|---|
| TPLSSVR | 0.019 478 | 0.024 816 | 0.123 164 | 0.015 974 | 0.958 558 | 2.75 |
| TLSSVR | 0.020 534 | 0.026 310 | 0.138 446 | 0.017 956 | 0.928 417 | 2.69 |
| PLSSVR | 0.020 204 | 0.025 715 | 0.132 254 | 0.017 153 | 0.937 246 | 1.12 |
| LSSVR | 0.021 812 | 0.027 779 | 0.154 332 | 0.020 016 | 0.917 318 | 1.10 |

### 3.2　短期风速预测

本文利用黑龙江省某一风电场的风速采样数据集 $T$ 来验证模型的有效性. 该风电场的实时采样间隔为 5 s,利用统计学习方法得到每 10 min 的风速属性值,每天记录 144 次,数据集共有 62 466 次记录. 数据集各列包括平均风速、方差、最小值、最大值等多个属性因子. 任选某一时间段内 6 天数据,其中前 3 天的 432 个样本作为训练集,后 3 天的 432 个样本作为测试集. 本文只采用第一列平均风速,利用历史平均风速数据与未来风速数据的依赖关系来预测未来风速大小,将风速序列转化为多变量任务. 其预报模式为:利用向量 $\boldsymbol{x}_i=(x_{i-p},x_{i-p-1},x_{i-p-2},\cdots,x_{i-1},x_i)$ 预测 $\boldsymbol{x}_{i+\text{step}},i=1,2,3,\cdots,864$,$p$ 表示输入的维数,step 表示提前预测的步数,当 step=1、3 时,分别表示利用前 $p$ 次的风速预测后 10 min、30 min 的风速. 在实验中,并非输入维数越大,预测效果越好,输入的数据中总会有有效和无效的数据,当有效数据所占比例越高时,该模型预报的效果越好;当有效数据所占比例越低时,该模型预报的效果越差;因而需要寻找预报效果最佳的组合. 为验证这一猜想,本文将提前步数 step 设置为 1、3,将维度 $p$ 范围设置为(2,15),将平均绝对误差 MAE 作为误差评判标准,观察其误差变化趋势,并寻找其最小误差.

从图 2、图 3 可以看出,误差先下降,当到达拐点时,再缓缓上升,存在一个最佳输入维数. 当选择 MAE 最小时,对应的输入维数即为最佳输入维数. 实验发现,在对风速预报提前 10 min、30 min,输入维数分别为 5、4 时,误差最小.

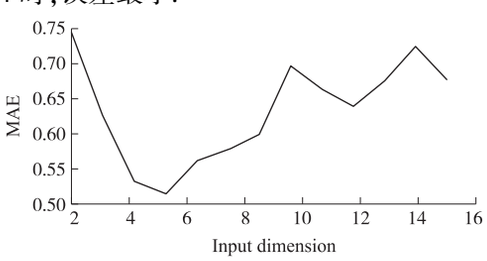



**图 2　提前 10 min 误差随输入维数的变化趋势**

**Fig. 2　The variation trend of error with input dimension in advance of 10 minutes**

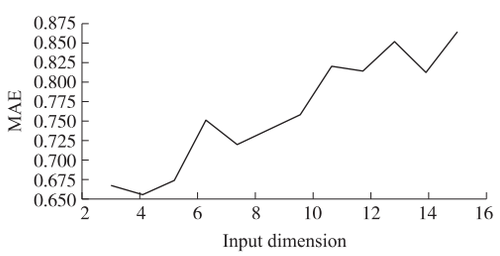



**图 3　提前 30 min 误差随输入维数的变化趋势**

**Fig. 3　The variation trend of error with input dimension in advance of 30 minutes**

在风速预测方面,本文用 432 个样本做测试集,432 个样本做训练集,为获取参数,使用粒子群算法搜索最优参数. 本文分别用 PLSSVR、LSSVR、TPLSSVR、TLSSVR 4 种模型对风速数据集进行 10 min、30 min 预测. 若对风速提前 10 min 预报,输入维数为最优输入维数 5;若对风速提前 30 min 预测,输入维数为最优输入维数 4. 对应的预测效果如图 4 和图 5 所示.

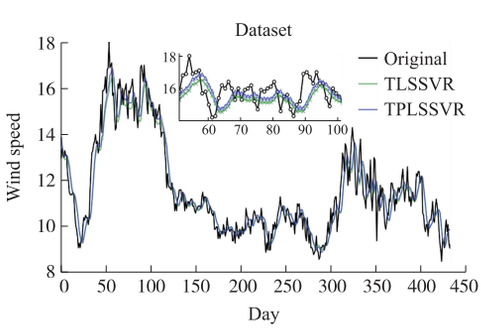



**图 4　TPLSSVR 和 TLSSVR 预测 10 min 后的风速预报**

**Fig. 4　Wind speed forecast after 10 minutes forecast by TPLSSVR and TLSSVR**

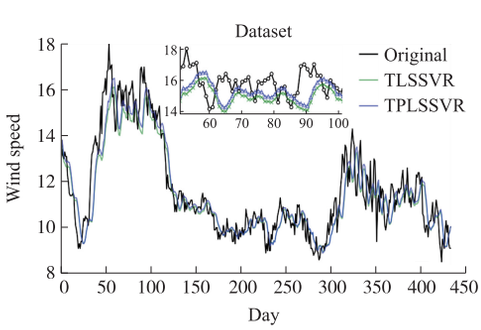



**图 5　TPLSSVR 和 TLSSVR 预测 30 min 后的风速预报**

**Fig. 5　Wind speed forecast after 30 minutes forecast by TPLSSVR and TLSSVR**

图 4 显示了 TPLSSVR 和 TLSSVR 预测 10 min 后的风速预报效果,表 2 为 4 种模型预测 10 min 后的风速预报误差分析;图 5 显示了 TPLSSVR 和 TLSSVR 预测 30 min 后的风速预报效果,表 3 为 4 种模型预测 30 min 后的风速预报误差分析. 由表 2、表 3 可以看出,与其他 3 种模型相比,TPLSSVR 的 MAE、RMSE、SSE、SSE/SST 都相对较小,预测结果更接近原始数据,4 种模型提前 10 min 的预测比提前 30 min 预测效果都要更好. 此外,TPLSSVR 与 TLSSVR 运行时间接近一致.

**表 2　4 种模型对 10 min 后的风速预报误差**

**Table 2　Error of wind speed forecast after 10 minutes by four models**

| 模型 | MAE | RMSE | SSE | SSE/SST | SSR/SST | 时间/s |
|---|---|---|---|---|---|---|
| TLSSVR | 0.523 681 | 0.700 448 | 211.950 912 | 0.110 751 | 0.802 504 | 11.64 |
| PLSSVR | 0.516 503 | 0.690 090 | 205.728 874 | 0.107 500 | 0.894 592 | 4.50 |
| LSSVR | 0.539 344 | 0.718 678 | 223.127 020 | 0.116 591 | 0.811 848 | 4.45 |
| TPLSSVR | 0.515 387 | 0.688 851 | 204.991 033 | 0.107 115 | 0.885 159 | 11.52 |

**表 3　4 种模型对 30 min 后的风速预报误差**

**Table 3　Error of wind speed forecast after 30 minutes by four models**

| 模型 | MAE | RMSE | SSE | SSE/SST | SSR/SST | 时间/s |
|---|---|---|---|---|---|---|
| TLSSVR | 0.682 782 | 0.917 843 | 363.932 323 | 0.189 724 | 0.711 200 | 11.90 |
| PLSSVR | 0.691 473 | 0.934 661 | 377.391 160 | 0.196 740 | 0.661 082 | 10.39 |
| LSSVR | 0.692 164 | 0.935 258 | 377.873 824 | 0.196 992 | 0.673 644 | 10.61 |
| TPLSSVR | 0.662 730 | 0.884 599 | 338.046 398 | 0.176 229 | 0.791 378 | 11.78 |

# 4　结论

为了解决风速的不稳定性给电力系统预测带来的麻烦,在分析风速数据集噪声满足高斯分布的基础上,本文提出了一种基于高斯噪声孪生近端最小二乘支持向量机模型 TPLSSVR. 在对风速进行预报时,通过寻找提前 10 min 和提前 30 min 时的最佳输入维数,分别提前 10 min 和 30 min 对风速进行预报,得到了最佳预报效果. 经过对人工数据集和风速数据集的实验,显示模型 TPLSSVR 较 PLSSVR、LSSVR、TLSSVR 有更好的预报效果.

## [参考文献](References)


[1] VAPNIK V N. The Nature of Statistical Learning Theory[M]. New York,USA:Springer NY,1995.

[2] CORTES C,VAPNIK V. Support-vector networks[J]. Machine Learning,1995,20(3):273-297.

[3] SUYKENS J A K,VANDEWALLE J. Least squares support vector machine classifiers[J]. Neural Processing Letters,1999,9(3):293-300.

[4] FUNG G,MANGASARIAN O L. Proximal support vector machine classifiers[C]//Proceedings of the 7th ACM SIGKDD International Conference on Knowlege Discovery and Data Mining. San Francisco,USA:ACM,2001.

[5] JAYADEVA, KHEMCHANDANI R, CHANDRA S. Twin support vector machines for pattern classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2007,29(5):905-910.

[6] PENG X J. TSVR:an efficient twin support vector machine for regression[J]. Neural Networks,2010,23(3):365-372.

[7] PENG X J,CHEN D. PTSVRs:regression models via projection twin support vector machine[J]. Information Sciences,2018,435:1-14.

[8] 杨晓敏. 改进灰狼算法优化支持向量机的网络流量预测[J]. 电子测量与仪器学报,2021,35(3):211-217.

[9] 叶黎明,陈素根. 基于粒子群算法的投影孪生支持向量机[J]. 淮北师范大学学报(自然科学版),2021,42(1):29-35.

[10] 顾吉峰,王蓓. 基于改进粒子群算法的孪生支持向量机[J]. 计算机工程与设计,2020,41(11):3078-3082.

[11] DING S F,ZHANG X K,YU J Z. Twin support vector machines based on fruit fly optimization algorithm[J]. International Journal of Machine Learning and Cybernetics,2016,7(2):193-203.

[12] DING S F,AN Y X,ZHANG X K,et al. Wavelet twin support vector machines based on glowworm swarm optimization[J]. Neurocomputing,2017,225:157-163.

[13] 张谢锴,丁世飞. 基于马氏距离的孪生多分类支持向量机[J]. 计算机科学,2016,43(3):49-53.

[14] SARTAKHTI J S,AFRABANDPEY H,SARAEE M. Simulated annealing least squares twin support vector machine(SA-LSTSVM) for pattern classification[J]. Soft Computing,2017,21(15):4361-4373.

[15] 黄宏运,吴礼斌,李诗争. GA 优化的 SVM 在量化择时中的应用[J]. 南京师范大学学报(工程技术版),2017,17(1):72-79.

[16] WANG K N,PEI H M,DING X S,et al. Robust proximal support vector regression based on maximum correntropy criterion[J]. Scientific Programming,2019(3):7102946.

[17] 张仕光,周婷,刘超,等. 高斯噪声特性区间 $\nu$-支持向量回归机[J]. 山西大学学报(自然科学版),2020,43(4):880-884.

[18] 余乐安. 基于最小二乘近似支持向量回归模型的电子商务信用风险预警[J]. 系统工程理论与实践,2012,32(3):508-514.

[19] HU Q H,ZHANG S G,YU M,et al. Short-term wind speed or power forecasting with heteroscedastic support vector regression[J]. IEEE Transactions on Sustainable Energy,2016,7(1):241-249.

[责任编辑:严海琳]