Neurocomputing 138 (2014) 131–141

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A twin projection support vector machine for data regression

Xinjun Peng [a,b,*], Dong Xu [a], Jindong Shen [c]

[a] Mathematics Department, Shanghai Normal University, Shanghai 200234, PR China
[b] Scientific Computing Key Laboratory of Shanghai Universities, Shanghai 200234, PR China
[c] Mathematics Department, China Jiliang University, Hangzhou 310018, PR China

ARTICLE INFO

Article history:
Received 23 February 2013
Received in revised form
4 February 2014
Accepted 7 February 2014
Communicated by W.S. Hong
Available online 8 April 2014

Keywords:
Support vector regression
Projection axis
Data structural information
Empirical correlation coefficient
Projection variance

ABSTRACT

In this paper, an efficient twin projection support vector regression (TPSVR) algorithm for data regression is proposed. This TPSVR determines indirectly the regression function through a pair of nonparallel up- and down-bound functions solved by two smaller sized support vector machine (SVM)-type problems. In each optimization problem of TPSVR, it seeks a projection axis such that the variance of the projected points is minimized by introducing a new term, which makes it not only minimize the empirical variance of the projected inputs, but also maximize the empirical correlation coefficient between the up- or down-bound targets and the projected inputs. In terms of generalization performance, the experimental results indicate that TPSVR not only obtains the better and stabler prediction performance than the classical SVR and some other algorithms, but also needs less number of support vectors (SVs) than the classical SVR.

1. Introduction

In the past decade, support vector machines (SVMs) [1,2], including support vector classification (SVC) and support vector regression (SVR), have become useful tools for data classification and regression due to the excellent generalization performance and have been successfully applied to a variety of real-world problems [3]. SVMs are principled and implement structural risk minimization (SRM) that minimizes the upper bound of the generalization error [1,2].

For the classical SVR, it finds a function $f(\boldsymbol{x})$ that has at most ε deviation from the actually obtained targets for all the training data, and at the same time is as flat as possible. In other words, we do not care about errors as long as they are less than ε, but will not accept any deviation larger than this. There exist many algorithms to learn SVR, such as the sequential minimal optimization (SMO) algorithm [4] and smooth SVR [5]. On the other hand, researchers have proposed some new models, such as the least squares SVR (LS-SVR) [6,7], the Huber loss function [1,2], and the parametric insensitive SVR [8]. Some other methods include the normal LS-SVR [9], geometric methods [10,11], etc.

Recently, in the spirit of twin support vector machine (TWSVM) [12] and its extensions [13], we have presented a class of novel SVR algorithms for data regression, including twin SVR (TSVR) [14,15] and twin parametric insensitive SVR (TPISVR) algorithms [16]. These two algorithms determine indirectly the regressor through a pair of nonparallel up- and down-bound functions solved by two smaller sized SVM-type problems, which make them have a faster learning speed than classical SVR. Specifically, the two optimization problems of TSVR determine the ϵ-insensitive down- and up-bound functions, while the two optimization problems of TPISVR determine the parametric insensitive down- and up-bound functions. Experimental results have shown that the two algorithms obtain better generalization performance than the classical SVR, especially when the noise is heteroscedastic, that is, the noise strongly depends on the inputs [14,16]. However, the two algorithms only aim at minimizing the empirical loss, but not embedding any prior structural information of data into the learning process, which leads the down- and up-bound functions to be possibly contaminated by noise points.

In this paper, we present a novel SVR algorithm for data regression, called the twin projection support vector regression (TPSVR). This TPSVR algorithm also finds a pair of nonparallel down- and up-bound functions by two smaller-sized SVM-type optimization problems. More importantly, it introduces a pair of new terms into the optimization problems to find two projection axes for the training points, such that the projected points have as small as possible empirical variance values on the down- and up-bound functions. That is, it embeds the prior structural information of data into the learning process. Compared with the TSVR and TPISVR algorithms, this TPSVR absorbs the merits of TSVR and TPISVR algorithms, i.e., a faster learning speed than the classical SVR. Further, the up- and down-bound functions also reflect the

* Corresponding author at: Mathematics Department, Shanghai Normal University, Shanghai 200234, PR China.
E-mail address: xjpeng@shnu.edu.cn (X. Peng).

http://dx.doi.org/10.1016/j.neucom.2014.02.028
0925-2312/

characteristics of data points. More importantly, the introduced projection axes in TPSVR make the projected points on the normal directions of up- and down-bound functions have small variances. Then, it leads TPSVR to obtain a better fitting on training points since the prior structural information of data is embedded into the learning process. Computational comparisons on some other SVR algorithms in terms of generalization performance have been made on several artificial and benchmark datasets, indicating that the TPSVR not only obtains better generalization performance than the classical SVR, TSVR, and TPISVR, but it also needs less numbers of support vectors (SVs) than the classical SVR. In addition, the results also show that this TPSVR is stabler than TPISVR in terms of penalty factors.

The rest of this paper is organized as follows: Section 2 briefly introduces the classical TSVR [14] and TPISVR [16]. Section 3 presents the proposed projection twin support vector regression (TPSVR) model. Experimental results on some toy and benchmark datasets are given in Section 4. Some conclusions and remarks are drawn in Section 5.

2. Background

In this section, we briefly introduce classical TSVR [14] and TPISVR [16]. Without loss of generality, the training samples are denoted by a set $\mathcal{D}=\{z_i=(\boldsymbol{x}_i;y_i), i=1,\ldots,n\}$, where the inputs $\boldsymbol{x}_i\in\mathcal{X}\subset\mathcal{R}^m$, the targets (or responses) $y_i\in\mathcal{R}$, $i=1,\ldots,n$, and $\mathcal{X}$ denotes the space of the input patterns. Without loss generalization, we use the matrix $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n]\in\mathcal{R}^{m\times n}$ and the vector $\boldsymbol{y}=[y_1,\ldots,y_n]^T\in\mathcal{R}^n$ to denote the inputs and targets.

2.1. Twin support vector regression

TSVR [14] finds a pair of nonparallel functions around the data points. In general, it considers the following pair of functions for the linear case:

$$f_1(\boldsymbol{x})=\boldsymbol{w}_1^T\boldsymbol{x}+b_1 \quad \text{and} \quad f_2(\boldsymbol{x})=\boldsymbol{w}_2^T\boldsymbol{x}+b_2, \tag{1}$$

each one determines the ϵ-insensitive down- or up-bound function, respectively, while the end regressor is defined as $f(\boldsymbol{x})=\frac{1}{2}(f_1(\boldsymbol{x})+f_2(\boldsymbol{x}))=\frac{1}{2}(\boldsymbol{w}_1+\boldsymbol{w}_2)^T\boldsymbol{x}+\frac{1}{2}(b_1+b_2)$. The functions $f_k(\boldsymbol{x})$, $k=1,2$ are obtained by solving the following pair of QPPs:

$$\begin{aligned}\min\quad & \frac{1}{2}\sum_{i=1}^n(y_i-\epsilon_1-(\boldsymbol{w}_1^T\boldsymbol{x}_i+b_1))^2+\frac{c_1}{n}\sum_{i=1}^n\xi_i\\ \text{s.t.}\quad & y_i-(\boldsymbol{w}_1^T\boldsymbol{x}_i+b_1)\ge\epsilon_1-\xi_i,\ \xi_i\ge0,\ \forall i,\end{aligned} \tag{2}$$

and

$$\begin{aligned}\min\quad & \frac{1}{2}\sum_{i=1}^n(y_i+\epsilon_2-(\boldsymbol{w}_2^T\boldsymbol{x}_i+b_2))^2+\frac{c_2}{n}\sum_{i=1}^n\eta_i\\ \text{s.t.}\quad & (\boldsymbol{w}_2^T\boldsymbol{x}_i+b_2)-y_i\ge\epsilon_2-\eta_i,\ \eta_i\ge0,\ \forall i,\end{aligned} \tag{3}$$

where $\epsilon_k\ge0$, $k=1,2$, are insensitive parameters and $c_k>0$, $k=1,2$, are penalty factors given by users. By introducing the Lagrangian functions of (2) and (3) and the Lagrangian vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we obtain the dual QPPs, which are

$$\begin{aligned}\max\quad & -\tfrac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\alpha}+\boldsymbol{f}^T\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\alpha}-\boldsymbol{f}^T\boldsymbol{\alpha}\\ \text{s.t.}\quad & \boldsymbol{0}\le\boldsymbol{\alpha}\le\frac{c_1}{n}\boldsymbol{e},\end{aligned} \tag{4}$$

and

$$\begin{aligned}\max\quad & -\tfrac{1}{2}\boldsymbol{\beta}^T\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\beta}-\boldsymbol{h}^T\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\beta}+\boldsymbol{h}^T\boldsymbol{\beta}\\ \text{s.t.}\quad & \boldsymbol{0}\le\boldsymbol{\beta}\le\frac{c_2}{n}\boldsymbol{e},\end{aligned} \tag{5}$$

where the vectors $\boldsymbol{f}=\boldsymbol{y}-\boldsymbol{e}\epsilon_1$, $\boldsymbol{h}=\boldsymbol{y}+\boldsymbol{e}\epsilon_2$, and $\boldsymbol{H}=[\boldsymbol{X}^T\ \boldsymbol{e}]$.

After optimizing (4) and (5), we obtain the augmented vectors

$$\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}=(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T(\boldsymbol{f}-\boldsymbol{\alpha}),\qquad \begin{bmatrix}\boldsymbol{w}_2\\ b_2\end{bmatrix}=(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T(\boldsymbol{h}+\boldsymbol{\beta}). \tag{6}$$

Then we obtain the estimated regressor.

For the nonlinear case, we will obtain similar formulations if we map $\boldsymbol{x}$ to $[k(\boldsymbol{x}_1,\boldsymbol{x}),\ldots,k(\boldsymbol{x}_n,\boldsymbol{x})]^T$, where $k(\cdot,\cdot)$ is the inner product in the feature space $\mathcal{H}$, such as the Gaussian kernel. Here we omit the description because of the space limitation. The readers can refer to [14,15].

2.2. Twin parametric insensitive support vector regression

TPISVR [16] also derives a pair of nonparallel functions around the data points through two QPPs. Specifically, it finds two linear functions (1); each one determines the parametric insensitive down- and up-bound regression functions.

The parametric insensitive down- and up-bound regression functions are optimized by solving the following pair of QPPs:

$$\begin{aligned}\min\quad & \frac{1}{2}\boldsymbol{w}_1^T\boldsymbol{w}_1-\frac{\nu_1}{n}\sum_{i=1}^n(\boldsymbol{w}_1^T\boldsymbol{x}_i+b_1)+\frac{c_1}{n}\sum_{i=1}^n\xi_i\\ \text{s.t.}\quad & y_i\ge\boldsymbol{w}_1^T\boldsymbol{x}_i+b_1-\xi_i,\ \xi_i\ge0,\ \forall i,\end{aligned} \tag{7}$$

and

$$\begin{aligned}\min\quad & \frac{1}{2}\boldsymbol{w}_2^T\boldsymbol{w}_2+\frac{\nu_2}{n}\sum_{i=1}^n(\boldsymbol{w}_2^T\boldsymbol{x}_i+b_2)+\frac{c_2}{n}\sum_{i=1}^n\eta_i\\ \text{s.t.}\quad & y_i\le\boldsymbol{w}_2^T\boldsymbol{x}_i+b_2+\eta_i,\ \eta_i\ge0,\ \forall i.\end{aligned} \tag{8}$$

For the optimization problem (7), the second term in the objective function is to optimize the sum of estimation values of training points by $f_1(\boldsymbol{x})$. Specifically, the objective function of (7) maximizes $\sum_{i=1}^n(\boldsymbol{w}_1^T\boldsymbol{x}_i+b_1)$. Therefore, optimizing it leads the function $f_1(\boldsymbol{x})$ to be as large as possible. The constraints require the estimated values of training points obtained by $f_1(\boldsymbol{x})$ to be less than the response values of the training points. That is, the response values of training points should be larger than the estimation values obtained by $f_1(\boldsymbol{x})$. Otherwise, the slack variables $\xi_i\ge0$, $i=1,\ldots,n$, are introduced to measure the errors. The third term of the objective function minimizes the sum of error variables, which attempts to over-fit the training points. For the optimization problem (8), it has similar interpretations.

By introducing the Lagrangian functions and vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for the problems (7) and (8), we obtain the corresponding dual QPPs, which are

$$\begin{aligned}\max\quad & -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}-\boldsymbol{y}^T\boldsymbol{\alpha}+\frac{\nu_1}{n}\boldsymbol{e}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}\\ \text{s.t.}\quad & \boldsymbol{0}\le\boldsymbol{\alpha}\le\frac{c_1}{n}\boldsymbol{e},\quad \boldsymbol{e}^T\boldsymbol{\alpha}=\nu_1,\end{aligned} \tag{9}$$

and

$$\begin{aligned}\max\quad & -\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{y}^T\boldsymbol{\beta}+\frac{\nu_2}{n}\boldsymbol{e}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}\\ \text{s.t.}\quad & \boldsymbol{0}\le\boldsymbol{\beta}\le\frac{c_2}{n}\boldsymbol{e},\quad \boldsymbol{e}^T\boldsymbol{\beta}=\nu_2.\end{aligned} \tag{10}$$

After optimizing (4) and (5), we obtain the weight vectors $\boldsymbol{w}_k$, $k=1,2$

$$\boldsymbol{w}_1=\sum_{i=1}^n\left(\frac{\nu_1}{n}-\alpha_i\right)\boldsymbol{x}_i,\quad \boldsymbol{w}_2=\sum_{i=1}^n\left(\beta_i-\frac{\nu_2}{n}\right)\boldsymbol{x}_i. \tag{11}$$

Then, we can predict the target value for an unknown input $\boldsymbol{x}$.

Note that we can use the same method as the classical SVR to deal with the nonlinear TPISVR. Here we also omit the details because of the space limitation. The interested readers can refer to [16]. In general, this TPISVR model is suitable for many real-world problems, especially when the noise is heteroscedastic. However, it often leads to over-fitting results for learning the parametric