Neurocomputing 130 (2014) 10–19

Contents lists available at ScienceDirect

Neurocomputing

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

A fast and robust model selection algorithm for multi-input multi-output support vector machine

Wentao Mao [a,*], Jiucheng Xu [a], Chuan Wang [a], Longlei Dong [b]

[a] College of Computer and Information Technology, Henan Normal University, Xinxiang 453007, PR China
[b] State Key Laboratory for Strength and Vibration, Xi'an Jiaotong University, Xi'an 710049, PR China

ARTICLE INFO

Article history:
Received 19 June 2012
Received in revised form
14 November 2012
Accepted 5 January 2013
Communicated by T. Heskes
Available online 29 August 2013

Keywords:
Support vector machine
Leave-one-out error
Model selection
MIMO
Gradient decent

ABSTRACT

Multi-Input Multi-Output (MIMO) regression estimation problems widely exist in engineering fields. As an efficient approach for MIMO modeling, multi-dimensional support vector regression, named M-SVR, is generally capable of obtaining better predictions than many traditional methods. However, M-SVR is sensitive to the perturbation of hyper-parameters when facing small-scale sample problems, and most of currently used model selection methods for conventional SVR cannot be applied to M-SVR directly due to its special structure. In this paper, a fast and robust model selection algorithm for M-SVR is proposed. Firstly, a new training algorithm for M-SVR is proposed to reduce efficiently the numerical errors in training procedure. Based on this algorithm, a new leave-one-out (LOO) error estimate for M-SVR is derived through a virtual LOO cross-validation procedure. This LOO error estimate can be straightway calculated once a training process ended with less computational complexity than traditional LOO method. Furthermore, a robust implementation of this LOO estimate via Cholesky factorization is also proposed. Finally, the gradients of the LOO estimate are calculated, and the hyper-parameters with lowest LOO error can be found by means of gradient decent method. Experiments on toy data and real-life dynamical load identification problems are both conducted, demonstrating comparable results of the proposed algorithm in terms of generalization performance, numerical stability and computational cost.

1. Introduction

In many real-life engineering applications, nonlinear black-box modeling based on machine learning is widely used as an effective soft-sensing technique. Especially, it generally needs to estimate and predict several variables or targets in the fields of system identification and state estimation [1,2], etc. In this scenario, system's output is a vector $y \in \mathbb{R}^k (k > 1)$, which is called Multi-Input Multi-Output (MIMO) regression problem [3]. It is of very important significance to improve the precision and speed of MIMO modeling.

The traditional solution of MIMO problem is splitting multi-dimensional output into multiple single-dimensional outputs, which means constructing an independent regression model for each output dimension [4]. Although this kind of method has simple implementation, it is computationally expensive and incapable of containing useful information among outputs. Another solution is multivariate statistical regression [5]. However, this kind of method is sensitive to the changes of data so that it cannot be applied broadly. In present machine learning techniques, artificial neural network is the most common method to establish MIMO model [6]. However, when facing small-scale sample problem, this method easily falls into local minimum and leads to over-fitting.

Different with neural network, support vector machines (SVMs), introduced by Vapnik [7], root from statistical learning theory, and seek best generalization performance of decision model while minimizing the training error especially for small-sample learning problems. However, the conventional SVM regressions (SVRs) only have one-dimensional output [8]. As a pioneer research, Pérez-Cruz [9] developed an efficient multi-dimensional regression tool which has its roots in SVM. This MIMO SVM approach, named M-SVR, has become a promising tool for solving the problems of nonlinear channel estimation [10] and biophysical parameter evaluation [11].

However, according to our experimental results, M-SVR is somewhat sensitive to the perturbation of hyper-parameters in some small-scale sample problems. A little deviation of hyper-parameters will result in large bias of prediction performance. How to choose optimal hyper-parameters, called also *model selection* [12,13], is a key problem in M-SVR applications. Generally speaking, model selection mainly contains two key issues: generalization performance evaluation and selection strategy. In practice, cross-validation (CV) and leave-one-out (LOO) errors are widely used to obtain a reliable estimate of generalization error [14]. These methods are known to be simple and efficient but computationally

* Corresponding author. Tel.: +86 186 3737 9430.
E-mail address: maowt.mail@gmail.com (W. Mao).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.01.058

expensive. To overcome this problem, various theoretical generalization error bounds for SVM were analyzed by many previous researches [15–18]. Especially aiming at regression problems, Chang [19] derived various LOO bounds, e.g., Radius-Margin Bound and Span Bound for conventional SVM regression (SVR), and Cawley [20] proposed another simple LOO estimate for least squares SVM (LS-SVM). But most of these LOO bounds for conventional SVR are hardly applicable to M-SVR due to its special form of primal optimization. The present model selection methods for M-SVR mainly use 8-fold cross-validation [9] or LOO error [21] to estimate generalization performance. Obviously, M-SVR lacks a simple but efficient generalization estimator. Moreover, because the model selection of M-SVR needs to traverse the whole parameter space under a certain optimization strategy, the training procedure of M-SVR may need to compute the inverse of approximate singular matrix when using some "seemingly bad" hyper-parameters. Perhaps these parameters are exactly best ones in fact. Unfortunately, they will be left out due to the unnecessary numerical errors.

Based on the above analysis, the model selection of M-SVR needs to solve two problems: (1) how to evaluate the generalization ability of M-SVR model in a fast and accurate manner, and (2) how to improve the numerical stability of M-SVR training procedure. Inspired by [20], a fast and robust model selection algorithm for M-SVR specific to RBF kernel is proposed in this paper. This algorithm firstly presents a new and robust solution for M-SVR which avoids computing the inverse of approximate singular matrix. Moreover, a simple LOO error estimate for M-SVR is proposed in this paper to solve this problem through a virtual LOO cross-validation procedure. This LOO estimate can remarkably reduce computational complexity than traditional LOO method. And an efficient implementation of this LOO estimate is also proposed. Finally, the parameters with lowest LOO error are found using gradient descend optimization. Experimental results on toy and engineering data sets both show the benefit of the proposed model selection algorithm. The rest of this paper is organized as follows. In Section 2, a brief review to M-SVR is given. In Section 3, a new training algorithm of M-SVR is provided. In Section 4, a theoretical derivation about LOO error estimate of M-SVR and its efficient implementation are both presented. Section 5 further proposes a model selection algorithm of M-SVR based on gradient descend optimization. Section 6 is devoted to computer experiments, followed by a conclusion of the paper in the last section.

2. Brief introduction of M-SVR

The key idea of M-SVR is extending Vapnik $\varepsilon-$insensitive loss function to multi-dimensional output case, i.e., a hyper-spherical insensitive zone, which handles all the outputs together. Therefore, M-SVR can improve generalization performance of decision model especially when only scarce samples are available. Here a brief introduction of M-SVR will be provided as follows.

Given a set of *i.i.d.* training samples $\{(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_l,\mathbf{y}_l)\}\subset \mathbb{R}^d\times\mathbb{R}^Q$, M-SVR is formulated as minimization of the following functional [10]:

$$\min_{\mathbf{W},\mathbf{b}} Lp=\frac{1}{2}\sum_{j=1}^{Q}\|w^j\|^2+\gamma\sum_{i=1}^{l}L(u_i) \tag{1}$$

where $\mathbf{W}=[\mathbf{w}^1,\ldots,\mathbf{w}^Q]$, $\mathbf{b}=[b^1,\ldots,b^Q]^T$, $\mathbf{e}_i^T=\mathbf{t}_i^T-\phi^T(\mathbf{x}_i)\mathbf{W}-\mathbf{b}^T$, $\phi(\cdot)$ is mapping function from primal space to feature space, and

$$L(u)=\begin{cases}0, & u<\varepsilon\\ u^2-2u\varepsilon+\varepsilon^2, & u\geq\varepsilon\end{cases} \tag{2}$$

$$u_i=\|\mathbf{e}_i\|=\sqrt{\mathbf{e}_i^T\mathbf{e}_i} \tag{3}$$

By adopting the cost function $L(u)$ described in Eqs. (2) and (3), M-SVR is capable of finding the dependencies between outputs, and can take advantage of the information of all outputs to get a robust solution. As Eq. (1) cannot be solved straightforwardly, [10] utilized an iterative method, named IRWLS, to obtain a desired solution. By introducing a first-order Taylor expansion of cost function $L(u)$, the objective of Eq. (1) will be approximated by the following equation [10]:

$$Lp'(\mathbf{W},\mathbf{b})=\frac{1}{2}\sum_{j=1}^{Q}\|w^j\|^2+\frac{1}{2}\sum_{i=1}^{l}a_iu_i^2+CT \tag{4}$$

where

$$a_i=\begin{cases}0, & u_i^k<\varepsilon\\ \dfrac{2\gamma(u_i^k-\varepsilon)}{u_i^k}, & u_i^k\geq\varepsilon\end{cases} \tag{5}$$

and CT is constant term which does not depend on $\mathbf{W}$ and $\mathbf{b}$, the superscript k denotes kth iteration.

To optimize Eq. (4), an IRWLS procedure is constructed which linearly searched the next step solution along the descending direction based on the previous solution. According to the Representer Theorem [22], the best solution of minimization of Eq. (4) in feature space can be expressed as $\mathbf{w}^j=\sum_i\phi(\mathbf{x}_i)\beta^j=\mathbf{\Phi}^T\boldsymbol{\beta}^j$, so the target of M-SVR is transformed into finding best $\boldsymbol{\beta}$ and $\mathbf{b}$. The IRWLS of M-SVR can be summarized in the following steps [10]:

*Step*1: Set $k=0$, $\boldsymbol{\beta}^k=\mathbf{0}$, $\mathbf{b}^k=\mathbf{0}$. Calculate $u_i{}^k$ and a_i.

*Step*2: Compute the solution $\boldsymbol{\beta}^s$ and $\mathbf{b}^s$ according to the next equation:

$$\begin{bmatrix}\mathbf{K}+\mathbf{D}_a^{-1} & \mathbf{1}\\ \mathbf{a}^T\mathbf{K} & \mathbf{1}^T\mathbf{a}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}^j\\ b^j\end{bmatrix}=\begin{bmatrix}\mathbf{y}^j\\ \mathbf{a}^T\mathbf{y}^j\end{bmatrix},\quad j=1,\ldots,Q \tag{6}$$

where $\mathbf{a}=[a_1,\ldots,a_l]^T$, $(\mathbf{D}_a)_{ij}=a_i\delta(i-j)$, $\mathbf{1}$ is a column vector of l ones, and $\mathbf{K}$ is kernel matrix.

*Step*3: Define the corresponding descending direction $\mathbf{P}^k=[{\mathbf{W}^s-\mathbf{W}^k}\atop{(\mathbf{b}^s-\mathbf{b}^k)^T}]$. Use a backtracking algorithm to compute $\boldsymbol{\beta}^{k+1}$ and $\mathbf{b}^{k+1}$, and further obtain u_i^{k+1} and a_i. Go back to Step 2 until convergence.

The proof of convergence of the above algorithm is given in [10]. Once convergence is reached, $\boldsymbol{\beta}^{k+1}$ and b^j are model parameters of jth output regressor. Because $u_i{}^k$ and a_i are calculated using every dimension of $\mathbf{y}$, each individual regressor contains the information of all outputs, which improves the prediction performance of M-SVR.

3. Robust implementation of M-SVR

Note that the matrix on the left-hand side in Eq. (6) is approximate singular or not positive definite for some hyper-parameters, which will cause large deviation when computing model parameters $\boldsymbol{\beta}$ and b. In order to conduct effective model selection, this paper will firstly improve the training procedure of M-SVR in terms of numerical stability.

Without loss of generality, consider the jth dimensional output independently. Denoting $\mathbf{M}=\mathbf{K}+\mathbf{D}_a^{-1}$, $\boldsymbol{\beta}=\boldsymbol{\beta}^j$, $\mathbf{y}=\mathbf{y}^j$, Eq. (6) turns into the following form:

$$\begin{bmatrix}\mathbf{M} & \mathbf{1}\\ \mathbf{a}^T\mathbf{K} & \mathbf{1}^T\mathbf{a}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}\\ b\end{bmatrix}=\begin{bmatrix}\mathbf{y}\\ \mathbf{a}^T\mathbf{y}\end{bmatrix} \tag{7}$$

where the first row of equation can be re-written as

$$\mathbf{M}[\boldsymbol{\beta}+\mathbf{M}^{-1}\cdot\mathbf{1}\cdot b]=\mathbf{y} \tag{8}$$