Information Sciences 296 (2015) 160–174

Contents lists available at ScienceDirect

Information Sciences

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

A pruning method of refining recursive reduced least squares support vector regression

Yong-Ping Zhao [a,*], Kang-Kang Wang [a], Fu Li [b]

[a] School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, PR China
[b] ZTE Corporation, Nanjing 210012, PR China

ARTICLE INFO

Article history:
Received 23 February 2014
Received in revised form 20 October 2014
Accepted 27 October 2014
Available online 7 November 2014

Keywords:
Support vector machine
Kernel method
Least squares support vector regression
Pruning method

ABSTRACT

In this paper, a pruning method is proposed to refine the recursive reduced least squares support vector regression (RRLSSVR) and its improved version (IRRLSSVR), and thus two novel algorithms PruRRLSSVR and PruIRRLSSVR are yielded. This pruning method ranks support vectors by defining a contribution function to the objective function, and then the support vector with the least contribution is pruned unless it is the most recently selected support vector. Consequently, PruRRLSSVR and PruIRRLSSVR outperform RRLSSVR and IRRLSSVR respectively in terms of the number of support vectors while not impairing the generalization performance. In addition, a speedup scheme is presented that reduces the computational burden of computing the contribution function. To show the effectiveness and feasibility of the proposed PruRRLSSVR and PruIRRLSSVR, experiments are performed on ten benchmark data sets and a gas furnace instance.

1. Motivation

Support vector machine (SVM) [4,25,26] is widely used in a variety of fields, ranging from pattern classifications [2] to function approximations [19,24], owing to its well-known ability to perform at a superior level. As a member of the SVM family, least squares support vector machine (LSSVM) [22,23] also has drawn much attention from researchers and engineers in the past few years. In comparison with SVM, LSSVM has a faster training speed. This comes from the fact that the equality constraints are used as a surrogate of the inequality ones, hence solving the linear system rather than the quadratic programming problem. However, there are some shortcomings in LSSVM, one of which is that its solution is not sparse [20], i.e., every training example is a support vector. In contrast, SVM obtains a sparser solution. Since sparseness is significant for reducing the time required in the prediction phase, LSSVR is inferior to SVM in terms of prediction time.

Several studies have attempted to mitigate this particular shortcoming. The first of such attempts was in [21], where the sparseness was imposed by pruning support values from the sorted support value spectrum which results from the solution to the linear system. A sparse weighted LSSVM (SWLSSVM) was proposed [20], which is able to do pruning on the basis of the sorted support values. Different from [20,21] where the sparseness was imposed by recursively solving the approximation problem and subsequently omitting sample that has a small error in the previous pass, a more sophisticated pruning mechanism was proposed in [7], where the training samples that introduce the smallest approximation errors are omitted. Through adding regularization, Kuh and De Wilde [13] expedited De Kruif and De Vries's pruning scheme [7]. Hoegaerts

* Corresponding author.
E-mail addresses: y.p.zhao@163.com, y.p.zhao@mail.njust.edu (Y.-P. Zhao).

http://dx.doi.org/10.1016/j.ins.2014.10.058
0020-0255/

et al. [10] concluded that omitting a sample based upon its weighed support value rarely yields satisfactory pruning results, so they suggested to use instead as criterion the sum of these values at all training samples, yielding significant improvements. Based on sequential minimal optimization method [18], Zeng and Chen [28] presented a new criterion, which deletes the training samples that introduce minimum changes to the dual objective function instead of determining the pruning samples by errors.

In general, the aforementioned pruning algorithms employ backward greedy learning. Correspondingly, there are forward greedy algorithms to achieve the sparseness. After the addition of a bias term to the objective function, LSSVM is solved using the forward least squares approximation, producing a sparse solution in the least square sense [27]. Based on the Nyström approximation and quadratic Rényi entropy, a fixed-size LSSVM (FSLSSVM) was proposed to realize the sparse representation in primal weight space and applied successfully to electric load forecasting [6,9,22]. In [12], a fast sparse approximation for LSSVM (FSALSSVM) was presented. FSALSSVM sequentially picks up support vectors, which make most contributions to the objective function, from all the training samples in a greedy manner. Subsequently, an improved version of FSALSSVM (IFSALSSVM) was proposed using the partial reduction strategy [30]. Compared with FSALSSVM which adopts the complete reduction strategy, IFSALSSVM obtains more parsimonious solution. Zhao and Sun [35] combined the idea behind FSALSSVM with the reduced technique [14,15] and proposed recursive reduced least squares regression (RRLSSVR), which has already been applied to aero-engine adaptive models [11]. Additionally the online version of RRLSSVR was also developed [32]. Recently, Zhao et al. presented an improved scheme for RRLSSVR called IRRLSSVR [33]. Similar to RRLSSVR, IRRLSSVR selects the training samples leading to the largest reductions on the objective function. However, contrary to RRLSSVR, during the selection process it considers the effects of the previously selected samples. RRLSSVR freezes the support weights of the support vectors that have already been selected, while IRRLSSVR does away with this restriction. In fact, the candidate support vectors and the previous selected support vectors are not independent, so IRRLSVR gets much sparser solution than RRLSSVR.

Generally speaking, forward greedy algorithms are computationally cheaper and tend to have modest memory requirements. In contrast, backward greedy algorithms need higher computational cost since the full-order matrix is factorized prior to iteratively annihilating columns which lead to minimal increment in the residual error [5,17]. From a theoretical point of view, backward greedy algorithms have provable convergence properties. By contrast, there are no such known guarantees for forward greedy algorithms. Therefore it is fair to say that both backward and forward greedy algorithms have their fair share of advantages and disadvantages. So it may be a good idea to integrate them so that we can get the best of both worlds. The pruning method is a commonly-used strategy of realizing backward greedy algorithms. Hence, in this paper we attempt to integrate the pruning idea into the forward greedy algorithms namely RRLSSVR and IRRLSSVR to refine their sparseness performance, and thus PruRRLSSVR and PruIRRLSSV are obtained, respectively. PruRRLSSVR and PruIRRLSSVR gain better sparseness than their corresponding unpruned versions. That is, much sparser solutions are gained without worsening their generalization performance. To confirm the effectiveness and feasibility of the proposed PruRRLSSVR and PruIRRLSSVR algorithms, experimental results on ten benchmark data sets and a gas furnace instance are conducted, whose results are demonstrated in this paper.

The remainder of this paper is organized as follows: in Section 2 reduced least squares support vector regression (RLSSVR) is introduced. Section 3 discusses the procedures of both RRLSSVR and IRRLSSVR. A pruning method is given in Section 4 and PruRRLSSVR and PruIRRLSSVR are obtained after combining the pruning method with RRLSSVR and IRRLSSVR, respectively. To verify these proposed methods in this paper, experiments on ten benchmark data sets and a gas furnace instance are reported in Section 5. Conclusions follow in the final section.

2. Reduced least squares support vector regression

Considering a sample set $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^N$ where $\boldsymbol{x}_i \in \Re^l$ is the l-dimensional input and $d_i \in \Re$ is its corresponding output value, least squares support vector regression (LSSVR) was defined by Suykens et al. [22,23] as

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{e},b}\left\{\frac{1}{2}\|\boldsymbol{w}\|_2^2+\frac{C}{2}\sum_{i=1}^{N}e_i^2\right\} \\ &s.t.\ d_i=\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)+b+e_i,\ \ i=1,\ldots,N \end{aligned} \tag{1}$$

where $\boldsymbol{w}$ is the normal vector of the hyperplane, b is the bias, $\boldsymbol{e} = [e_1, \ldots, e_N]^T$ is the learning residual vector, $C \in \Re^+$ is the regularization parameter controlling the model complexity and the overfitting phenomenon, $\varphi(\cdot)$ is usually a nonlinear mapping from the input space into the so-called feature space. Eq. (1) is solved by constructing the Lagrangian

$$L(\boldsymbol{w},b,\boldsymbol{e};\boldsymbol{\alpha})=\frac{1}{2}\|\boldsymbol{w}\|_2^2+\frac{C}{2}\sum_{i=1}^{N}e_i^2+\sum_{i=1}^{N}\alpha_i\left(d_i-\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)-b-e_i\right) \tag{2}$$

where α is the Lagrangian multiplier vector. This optimization problem comes down to solving the following linear system

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \overline{K} \end{bmatrix}\begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix}=\begin{bmatrix} 0 \\ \boldsymbol{d} \end{bmatrix} \tag{3}$$