Information Sciences 201 (2012) 37–52

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Online independent reduced least squares support vector regression

Yong-Ping Zhao [a,*], Jian-Guo Sun [b], Zhong-Hua Du [a], Zhi-An Zhang [a], Ye-Bo Li [b]

[a] ZNDY of Ministerial Key Laboratory, Nanjing University of Science and Technology, Nangjing 210094, PR China
[b] College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, PR China

ARTICLE INFO

Article history:
Received 13 August 2010
Received in revised form 4 December 2011
Accepted 26 February 2012
Available online 5 March 2012

Keywords:
Machine learning
Support vector regression
Online algorithm
Offline algorithm
Linear independence

ABSTRACT

In this paper, an online algorithm, viz. online independent reduced least squares support vector regression (OIRLSSVR), is proposed based on the linear independence and the reduced technique. As opposed to some offline algorithms, OIRLSSVR takes the realtime advantage, which is confirmed using benchmark data sets. In comparison with online algorithm, the realtime of OIRLSSVR is also favorable. As for this point, it is tested with experiments on the benchmark data sets and a more realistic scenario namely a diesel engine example. All in all, OIRLSSVR can enhance the modeling realtime, especially for the case where the samples enter in a flow mode.

1. Motivation

Rooted in the statistical learning theory, support vector machine (SVM) [5,8,30,37], as an excellent and powerful tool for classification and regression in machine learning community, has obtained a lot of successful applications in many research fields [11,25,29,38,43] like feature selection [22,23,36], text categorization [3], and time series prediction [15] due to its outperforming generalization performance. The least squares version of SVM, viz least squares support vector machines (LSSVM) [33,34], also has gained a lot of popularity in plenty of fields of science and engineering [20,21,27,35,39,41,49]. As opposed to traditional SVM, LSSVM cuts down the training complexity obviously, that is to say, LSSVM only needs to solve a linear equation as a surrogate of quadratic programming. Still, its solution suffers from lack of parsimoniousness [31], because the equality constraints are replaced with inequality ones. To this end, Suykens et al. [32] firstly proposed a pruning mechanism of omitting the training observations owning the small training errors. Subsequently, De Kruif and De Vries [9] presented a more sophisticated pruning approach, i.e., deleting the training observations holding the better prediction accuracy while excluding them to build LSSVM. Kuh and De Wilde [14] accelerated De Kruif et al.'s pruning method [9]. Based on sequential minimization optimization [26,13], Zeng and Chen [42] gave another pruning strategy to realize the parsimonious LSSVM. Jiao et al. [12] presented a fast sparse approximation scheme, iteratively building the decision function by adding one basis function which makes most contribution to the objective function from a kernel-based dictionary, for LSSVM to overcome its limitation of lack of parsimoniousness. However, the above-mentioned algorithms are offline. That is to say, when the training observations arrive sequentially, they must be retrained from scratch. A system involving online learning must face a potential flow of training observations, updating its knowledge after each new observation. In this situation, these offline algorithms may not provide the satisfying realtime. Hence, it is necessary to develop online learning ones. In general, an excellent online algorithm includes two characters at least. One is to update itself incrementally after obtaining a new observation instead of to retrain from scratch, thus reducing the training time. The other is to keep a parsimonious solution in order to shorten the testing time. The smaller the sum of the training and testing time is, the better the reatime is.

* Corresponding author.
E-mail address: y.p.zhao@163.com (Y.-P. Zhao).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2012.02.052

Paralleled with offline algorithms realizing LSSVM's parsimoniousness, online learning ones [40,45,46,48] have not yet been fully developed. The main reason is that it is not easy to develop a better online algorithm which needs smaller incrementally computational complexity and meantime more parsimonious solution. The smaller incrementally computational complexity insinuates the effectively updating strategy, while the more parsimonious solution needs fewer support vectors. Although the solution of SVM is parsimonious, it owns currently considerably slower testing speed caused by the number of support vectors, which has been a serious limitation to some applications. Hence, Li et al. [17] proposed an adaptive scheme for simplifying solution of SVM based on the linear independence in the feature space. According to recent research [19], we know that there are some dispensable support vectors in the solution of SVM, about 1–9% (in some scenarios, more than 50%). Therefore, some strategies [18,19] were proposed to prune these dispensable support vectors with a negligible reduction in classification accuracy by virtue of the linear independence in the defined kernel row space. To be more important, this kernel row space is equivalent to the feature space defined by the feature mapping induced by kernel function [19]. Unfortunately, these pruning algorithms are offline. More recently, Orabona et al. [24] proposed an online algorithm, called online independent support vector machine (OISVM), which approximately converges at traditional SVM solution each time new observations are added. OISVM selects a set of linearly independent observations and tries to project every new observation onto the set obtained so far, dramatically reducing time and space requirement at the price of a negligible loss in accuracy. In addition, the reduced technique [16] is another excellent strategy to imperatively restrict the number of support vectors (#SV). From the reduced technique, some offline algorithms owning better parsimoniousness were proposed, such as, reduced least squares support vector regression (RLSSVR) and recursive reduced least squares support vector regression (RR-LSSVR) [47]. Enlightened by these ideas and combining them, an online learning algorithm, i.e., online independent reduced least squares support vector regression (OIRLSSVR), is proposed. In the case of OIRLSSVR, the linearly independent training samples are selected to construct the set of support vectors, and then using this selected set builds RLSSVR, thus cutting down #SV and the testing time. Furthermore, after new observation we can update RLSSVR from the already-built RLSSVR rather than rebuild it, thus lessening the retraining time. As a result, the sum of the training and testing time is reduced. Naturally, the realtime of our proposed algorithm is enhanced. Finally, to confirm the feasibility and effectiveness of OIRLSSVR, extensive evaluations on benchmark data sets are carried out. The experimental results show that, as opposed to RR-LSSVR and RLSSVR, OIRLSSVR owns the realtime advantage. Meantime, compared with other online LSSVR algorithm, the realtime of OIRLSSVR is also favorable, which is reported by the experiments on the benchmarks and a more realistic example.

This paper is organized as follows. In Section 2, reduced least squares support vector regression is introduced briefly. In the following, we discuss how to select the basis vectors of feature space online as support vectors. Based on these chosen support vectors, an online algorithm, viz OIRLSSVR, is proposed in Section 4. To show the efficacy and feasibility of the presented OIRLSSVR, as opposed to the offline algorithms, experiments on the benchmarks are carried out. Further more, we compare OIRLSSVR with the online algorithm using experiments on the benchmarks and a more realistic scenario namely a diesel engine example in Section 5. Finally, conclusions follow.

2. Reduced least squares support vector regression

Before introducing RLSSVR, we briefly depict least squares support vector regression, and then RLSSVR is brought out. Given a training set $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in \mathbb{R}^{\boldsymbol{m}}$ is the input observation, $d_i \in \mathbb{R}$ is the corresponding output observation, after the equality constraints instead of the inequality ones, Suykens and Vandewalle [34] gave the mathematical model of least squares support vector regression (LSSVR) as:

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

where $\boldsymbol{w}$ can control the model complexity, b is the bias, $\boldsymbol{e} = [e_1, \ldots, e_N]^T$ represents the training errors, C is the regularization parameter adjusting the tradeoff between the model flatness $\boldsymbol{w}^T\boldsymbol{w}$ and the fitness $\sum_{i=1}^{N} e_i^2$, $\varphi(\cdot)$ is called the feature map realizing the transformation from the finite-dimensional input space to the high-dimensional feature space. To solve (1), the Lagrangian function is constructed

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

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$ is the Lagrange multiplier vector. According to the dual theorem, the optimality conditions are:

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{w}} = 0 \to \boldsymbol{w} = \sum\limits_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial L}{\partial b} = 0 \to \sum\limits_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \to \alpha_i = Ce_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \to \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b + e_i - d_i = 0 \end{cases} \tag{3}$$