Online Support Vector Regression for System Identification

Zhenhua Yu, Xiao Fu, and Yinglu Li

School of Telecommunication Engineering,
Air Force Engineering University, Xi'an 710077, China
zhenhua_yu@163.com

Abstract. Conventional Support Vector Regression (SVR) is not capable of online setting and its training algorithm is inefficient in real-time applications. Through analyzing the possible variation of support vector sets after new samples are added to the training set, and extending the incremental support vector machine for classification, an online learning algorithm for SVR is proposed. To illustrate the favorable performance of the online learning algorithm, a nonlinear system identification experiment is considered. The simulation results indicate that the learning efficiency and prediction accuracy of the online learning algorithm are higher than that of the existing algorithms, and it is more suitable for system identification.

1 Introduction

Support vector machine (SVM) is a new universal learning machine in the framework of Structural Risk Minimization (SRM) [1], which is based on statistical learning theory. SRM has greater generalization ability and is superior to the traditional Empirical Risk Minimization (ERM) principle adopted in many conventional neural networks. In SVM, the results guarantee global minima whereas ERM can only locate local minima. Initially, SVM is designed to solve pattern recognition problems. Recently, SVM has also been successfully applied to regression estimation [2], [8], and the approach is often referred to as the support vector regression (SVR). Conventionally SVR is used for regression estimation of input data that are supplied in batch. In many application problems, such as system identification, time series prediction and signal processing, data are obtained in a sequence and learning has to be done from scratch. Therefore, it is time consuming to achieve the regression using the conventional SVR and it is not possible to apply the SVR for real-time regression problems. Recently several online learning algorithms have been proposed [4], [5], [6]. However, most of these algorithms are only described for classification. In existing online learning algorithms, some algorithms are not suitable for online adjusting due to low efficiency when coping with a large number of support vectors, and others only return approximate solutions for remaining support vectors. In this paper, a novel online learning algorithm for SVR (online SVR) is proposed, which is an extension of the work presented in [3], [7]. Finally, the online SVR is applied to

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 627–630, 2005.

the nonlinear system identification to test the efficiency. Simulation shows the online SVR is adaptive to system identification.

2 Online Learning Algorithm for Support Vector Regression

A more detailed description of SVR can be found in [2]. In SVR, according to dual theory and Lagrangian function, we get the following formulation:

$$\begin{aligned} L_D = \frac{1}{2}\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)Q_{ij}(\alpha_j - \alpha_j^*) + \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \\ - \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) + \delta\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) \end{aligned} \tag{1}$$

with the first order conditions for L_D:

$$g_i = \frac{\partial L_D}{\partial \alpha_i} = \sum_{j=1}^{l} Q_{ij}(\alpha_j - \alpha_j^*) + \varepsilon - y_i + \delta = 0 \tag{2}$$

$$g_i^* = \frac{\partial L_D}{\partial \alpha_i^*} = -\sum_{j=1}^{l} Q_{ij}(\alpha_j - \alpha_j^*) + \varepsilon + y_i - \delta = 0 \tag{3}$$

$$\frac{\partial L_D}{\partial \delta} = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{4}$$

A coefficient θ_i is defined as $\theta_i = \alpha_i - \alpha_i^*$, and θ_i is determined by both α_i and α_i^*. The first order conditions for L_D lead to KKT conditions, which can divide the whole training samples into the following sets: margin support vectors S, error support vectors E, and remaining vectors R. Specifically, centering on g_i, KKT conditions are:

$$\begin{aligned} 2\varepsilon < g_i \to g_i^* < 0 \quad \theta_i = -C \quad i \in E \\ g_i = 2\varepsilon \to g_i^* = 0 \quad -C < \theta_i < 0 \quad i \in S \\ 0 < g_i < 2\varepsilon \to 0 < g_i^* < 2\varepsilon \quad \theta_i = 0 \quad i \in R \\ g_i = 0 \to g_i^* = 2\varepsilon \quad 0 < \theta_i < C \quad i \in S \\ g_i < 0 \to g_i^* > 2\varepsilon \quad \theta_i = C \quad i \in E \end{aligned} \tag{5}$$

Variations in θ_c of the new sample x_c influence g_i, g_i^*, and θ_i of the other samples in training samples, so the transfer of some vectors from on set S, R, E to another set may be forced. From (5), if one sample remains in S, its g_i does not change. While one sample remains in E and R, its θ_i does not change. The variation in g_i, g_i^* and θ_i are calculated as follows when a new sample with influence θ_c is added.

$$\triangle g_i = Q_{ic}\triangle\theta_c + \sum_{j\in S} Q_{ij}\triangle\theta_j + \triangle b \tag{6}$$

$$\triangle g_i^* = - \triangle g_i \tag{7}$$

$$\triangle\theta_c + \sum_{j \in S} \triangle\theta_j = 0 \tag{8}$$

Due to limited space, how the variation in the θ_c of a new sample x_c influences θ_j of samples $j \in S$ or $j \in E \bigcup R$ and the computation of $\Delta\theta_c$ during the migration process can be found in [3]. The online learning algorithm is obtained as follows [7]:

1. Set the coefficient $\theta_c = 0$ of the new sample x_c;
2. if $g_c > 0$ and $g_c^* > 0$, then x_c is added to R and exit;
3. If $g_c < 0$ then increment θ_c, updating θ_i in S and g_i, g_i^*in E, R, until one of the following conditions holds:
   - $g_c = 0$: add x_c to S, update the matrix Q^{-1} and exit;
   - $\theta_c = C$: add x_c to S and exit;
   - samples in S, E or R may migrate and update the matrix Q^{-1};

   Else $g_c^* < 0$ then decrement θ_c, updating θ_i in S and g_i, g_i^* in E, R, until one of the following conditions holds:
   - $g_c^* = 0$: add x_c to S, update the matrix Q^{-1} and exit;
   - $\theta_c = -C$: add x_c to E and exit;
   - samples in S, E or R may migrate and update the matrix Q^{-1};
4. Return to 1.

3 Application of Online SVR to System Identification

Nonlinear system identification is a crucial but complex problem, where data are obtained in a sequence. Therefore, the online learning algorithm for SVR is particularly well suited. Now, we validate the performance of online SVR by simulation experiments and compare its performance to existing SVR algorithms.

Online SVR is applied to nonlinear system identification and compared with the existing incremental algorithm [6] and batch SVR algorithm (LibSVM). In these experiments, the kernel function is Gaussian function $K(x_i, x_j) = exp(-\lambda\|x_i - x_j\|^2), \lambda = 1$, and MSE is used to quantify the performance.

A nonlinear system [9] to be identified is governed by the difference equation

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k)) \tag{9}$$

where $f(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u)$, and the input $u(k) = \sin(2\pi k/250)$. One takes 1500 points as the training samples, and these samples are sequentially obtained. Table 1. lists the approximation errors and speed using the three algorithms.

From the simulation experiment, one can find the proposed online learning algorithm outperforms the incremental algorithm and batch algorithm. Online algorithm and batch algorithm produce almost the same error, while the accuracy of these algorithms is higher than the incremental algorithm for only providing an approximation solution. As the procedure of online algorithm is iterative, its learning efficiency is higher than the other algorithms; the batch learning algorithm has to be done from scratch when data are obtained in a sequence, so its speed is rather slow.

Table 1. Performance comparison for three algorithms

Learning algorithm	Learning speed(s)	MSE
Online learning	8.46	0.0105
Incremental learning	12.15	0.0332
LibSVM	23.37	0.0109

4 Conclusions

As conventional SVR suffer from the problem of large memory requirement and CPU time when trained in batch mode on large-scale sample sets, this paper presents an online learning algorithm for SVR that have input data supplied in sequence rather than in batch. Online learning algorithm analyzes the possible change of support vectors after new samples are added to training set. Online learning algorithm for SVR is applied to nonlinear system identification. Simulation shows that the online learning algorithm has a much faster convergence and a better generalization performance in comparison with the existing algorithms.

References

1. Wang, L.P.: Support Vector Machines: Theory and Application. Springer-Verlag, Berlin Heidelberg New York (2005)
2. Smola, A.J., Schökopf, B.: A tutorial on support vector regression. NeuralCOLT2 Technical Report Series, No. NC2-TR-98-030. London, Royal Holloway Colledge, University of London (1998)
3. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning. In: Dietterich, T.G., Leen, T.K., and Tresp, V. (eds.): Advances in Neural Information Processing Sytems. MIT Press (2001) 409-415
4. Gentile, C.: A new approximate maximal margin classification algorithm. Journal of Machine Learning Research 2 (2001) 213-242
5. Ralaivola, L., d'Alche-Buc, F.: Incremental support vector machine learning: A local approach. Lecture Notes in Computer Science, Vol. 2130. Springer-Verlag, Berlin Heidelberg New York (2001) 322-330
6. Syed, N., Liu, H., Sung, K.K.: Incremental Learning with Support Vector Machines. In Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence. Stockholm, Sweden. (1999)
7. Martin, M.: On-line support vector machine regression. Lecture Notes in Computer Science, Vol. 2430. Springer-Verlag, Berlin Heidelberg New York (2002) 282-294
8. Vapnik, V., Golowich, S., and Smola, A.: Support vector method for function approximation, regression estimation, and signal processing. In: Mozer, M.C., Jordan, M.I., and Petsche, T.: Advances in Neural Information Processing Systems. MIT Press (1997) 281-287
9. Narendra, K., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. on Neural Networks 1 (1990) 4-27