2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Adaptively Weighted Support Vector Regression for Financial Time Series Prediction

Zhijie Li[1,2], Yuanxiang Li[1], Fei Yu[1], Dahai Ge[1]
[1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China
[2] Computer School, Hunan Institute of Science and Technology, Yueyang 414006, China
lzj0019@163.com, yxli@whu.edu.cn, yufei@whu.edu.cn, 117225075@qq.com

***Abstract*—The financial data are usually volatile and contain outliers. One problem of the standard support vector regression (SVR) for financial time series prediction is that it considers data in a fixed fashion only and lack the robustness to outliers. To tackle this issue, we propose the adaptively weighted support vector regression (AWSVR) model. This novel model is demonstrated to choose the weights adaptively with data. Therefore, the AWSVR can tolerate noise adaptively. The experimental results on three indices: the NASDAQ, the Standard & Poor 500 index (S&P), and the FSTE100 index (FSTE) show its advantages over the standard SVR.**

Keywords—Support vector regression, outliers, data adaptive learning, weighted learning, financial time series prediction.

I. INTRODUCTION

FINANCIAL time series prediction is a challenging task in machine learning [1], [2], [3], [4], [5]. The support vector regression (SVR) has been successfully applied in this problem due to its advantage of generalization with a unique and global optimal solution [6], [7], [8]. The standard SVR adopts the ε-insensitive loss function with a fixed margin to control the sparsity of the solution and reduce the effect of some unimportant data points. This setting is simple and effective in common applications.

Financial data are usually highly volatile and the noise varies over time, and fixing the margin cannot tolerate noise adaptively. In modeling the financial time series, one key problem is outliers detection and reduction [1], [5]. When the data are noise with possible outliers in the data, learning observations without awareness of outliers may lead to fitting unwanted data and interrupt the approximation function, which will result in the lowering of generalization performance. Hence, extending the fixed ε-insensitive loss function to general ε-insensitive with adaptive margin is very important in the prediction of the stock market. Specific techniques, e.g., [9], [10], and [11], have been proposed to gain robustness. In these methods, the corresponding optimization problem either is nonconvex or involve much more computational cost than that of the standard SVR.

In this paper, we proposed an adaptively weighted SVR training algorithm to enhance the robust capability of standard SVR. The weights are chosen adaptively with data, which can reduce the impact of outliers by using smaller weights. As a result, the weighted SVR can deliver better prediction performance than the unweighted ones. Experimental results on real-world financial datasets demonstrate the merits of the proposed weighting and adaptive learning algorithm for financial time series prediction.

II. SUPPORT VECTOR REGRESSION

Given financial time series data set, $\{(x_t, y_t) \mid x_t \in R^d, y_t \in R, t=1,\dots,N\}$, the objective of regression is to find a function which can not only approximate these data well, but also can accurately predict the value of y for future data x.

In standard SVR, the approximating function takes the following linear form:

$$f(x)= \beta^T x + b, \tag{1}$$

where $\beta \in R^d$, $b \in R$. Furthermore, the (1) can be extended into the non-linear model by using Mercer's kernel [12], [13].

Now the task is to determine w and b from the training data by minimizing the regression risk which is defined as follows:

$$R_{reg}(f) = \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{N} l_\varepsilon(y_i - f(x_i)). \tag{2}$$

Where the term $\frac{1}{2}\beta^T\beta$ is a complexity term, C is a regularized constant, and $l_\varepsilon(y - f(x))$ is the ε-insensitive loss function which is defined as

$$l_\varepsilon(x_i, y_i) = [|y_i - f(x_i)| - \varepsilon]_+. \tag{3}$$

Where $[]_+$ mathematically mean the value of what is between brackets if it is non-negative, else the value is zero. In above function, data points in the range of ε-margin are not considered. Therefore, it can reduce their effect to approximation function and controls the sparsity of solution.

The complete optimization of L_2-SVR can be written as follows:

$$\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{4}$$

$$\text{s.t.} \quad y_i - (\beta^T x_i + b) \le \varepsilon + \zeta_i, \tag{5}$$

$$(\beta^T x_i + b) - y_i \le \varepsilon + \zeta_i^*, \tag{6}$$

$$\zeta_i \ge 0,\ \zeta_i^* \ge 0,\ i=1,\cdots,N. \tag{7}$$

Where ζ_i and ζ_i^* are the corresponding positive and negative errors at the i-th point, respectively. The above optimization problem can be solved by the quadratic programming method [8], [14].

III. AWSVR MODEL

A. Weighted SVR

In proposed weighted SVR, we assume that there is a nonnegative weight w_i associated with each pair of observations (x_i, y_i) for $i=1,\cdots,N$. It solves the following optimization problem by the weighted learning:

$$R_{reg}(f)=\frac{1}{2}\beta^T\beta+C\sum_{i=1}^{N}w_i l_\varepsilon(y_i-f(x_i)). \qquad (8)$$

When $w_i=1$, it reduces to the standard SVR. Minimizing the regression risk of (8) by the Lagrange method, the corresponding dual problem becomes

$$\min_{\alpha,\alpha^*}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j)-\sum_{i=1}^{N}y_i\alpha_i+\sum_{i=1}^{N}y_i\alpha_i^* \qquad (9)$$

$$s.t.\ \ \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0,\ \ \alpha_i,\alpha_i^*\in[0,w_iC], i=1,\cdots,N, \qquad (10)$$

where α_i and α_i^* are the corresponding Lagrange multipliers used to push and pull $f(x_i)$ towards the outcome of y_i, respectively. $K(x_i, x_j)$ is the kernel function which satisfies the Mercer's condition.

The above Quadratic Problem(QP) can be easily solved by a commonly used SVR library. Therefore, we obtain the approximation function as $f(x)=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)K(x,x_i)+b$, where the offset b is calculated by exploiting the Karush-Kuhn-Tucher (KKT) conditions in [15].

B. Choice of Weights

AWSVR uses a nonnegative weight w_i associated with each pair of observations, and how to choose weights is a key problem. When there are outliers, it is desirable to give smaller weights to observations that are outliers. One natural approach for the weighted learning is to adopt the results from standard learning to understand relative locations of the training data points to the regression boundary. Therefore, we first train the standard SVR in (2), and get the optimal approximation function and loss function. Denote them as $\hat{f}(\cdot),\hat{l}_\varepsilon(\cdot,\cdot)$, respectively, where $\hat{l}_\varepsilon(x_i,y_i)=[|y_i-\hat{f}(x_i)|-\varepsilon]_+$. Next we train the AWSVR (8) using this weight w_i defined as follows:

$$w_i=1\big/(1+\hat{l}_\varepsilon(x_i,y_i)), \qquad i=1,\cdots,N. \qquad (11)$$

Our motivation of the weight function comes from the form of the loss function $l_\varepsilon=[|y-f(x)|-\varepsilon]_+$. When ε=0, the loss function become $l_\varepsilon(u)=|y-f(x)|=|u|$. To connect the value of target loss $w_i\ l_\varepsilon$ with 0-1, we consider the weight form $w(u)=1/(1+|u|)$, and the target loss function is shown in Figure 1.

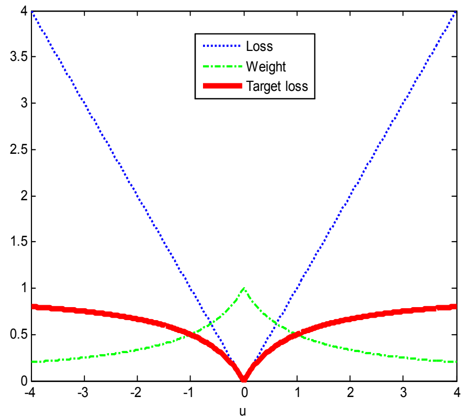

Fig. 1. Plot of the loss function $l_\varepsilon(u)$, the weight function for AWSVR, and the target loss function $w(u)\ l_\varepsilon(u)$ on u

Note that our proposed weighting scheme assigns a smaller weight for any observation (x_i,y_i) with a larger |u|. This matches our goal. In particular, for any observation (x_i,y_i), the larger |u| is, the more likely for (x_i,y_i) to be an outlier, our weight function will assign smaller weights for such observations in the weighted learning. A reason of outlier's weight being not zero is that the outliers still contain some useful information for constructing the approximation function and thus we cannot completely ignore them.

With the loss learning solution $\hat{l}_\varepsilon(x_i,y_i)$ of standard SVR, we use the weight $w_i=1\big/(1+\hat{l}_\varepsilon(x_i,y_i))$ for the *i*-th point. Therefore, the value of target loss function $w_i\hat{l}_\varepsilon$ will be in the range of 0 to 1 for all observations. This motivates us to propose the following "two-phase" procedure:

Phase 1: Train the standard SVR model with fixed ε margin setting, and calculate the weights for all training data points.

Phase 2: Apply the weighted learning to get the optimal approximation function f(x)

As we will show in our numerical results, our proposed choice of weights works very well in financial time series prediction.

IV. EXPERIMENTS

In this section, we implement the above "two-phase" procedure and perform the experiments on three indices: Standard & Poor 500 index (S&P), NASDAQ and the FSTE100 index (FSTE). The data are selected from the daily closing prices of the indices from January 2, 2013 to April 30, 2013. The beginning four-fifth data are used for training and the rest one-fifth data are used in the one-step ahead prediction. All algorithms run on a PC with 2.80GHz CPU and 3.93GB memory.

A. Data Preprocessing and Model Selection

Firstly, the daily closing prices (d_t) of the above three indices are converted to continuously compounded returns as $r_t=\log(d_t+1/d_t)$, and then they are normalized by $y_t=(r_t-\mathrm{Mean}(r_t))/\mathrm{STD}(r_t)$, where the means and STDs are computed for each individual index.

In our experiments, the predicted system is modeled as $\hat{y}_t = f(x_t)$, where x_t is constructed as a four-day's pattern: $x_t=(y_{t-4}, y_{t-3}, y_{t-2}, y_{t-1})$ based on the assumption that (non)linear relationship occurs in sequential five days' prices. The trade-off parameter C and the parameter γ of the RBF kernel $K(\mathrm{x},\mathrm{y}) = \exp\left(-\gamma \|\mathrm{x}-\mathrm{y}\|^2\right)$ are obtained on the following paired points: $[2^{-5}, 2^{-4},\cdots,2^{10}]\times[2^{-5}, 2^{-4},\cdots,2^{10}]$. We just restrict the ε values in the range of 0.0, 0.2, ⋯,1.0 to 2.0 since when $\varepsilon \geq 2.0$, there are not support vectors in the SVR.

In the first phase, we construct the approximation function $f(x_t)$ by performing the SVR (2) on the normalized training data using the above settings. Then we observe that some training data points actually differ largely from the predictive values. Therefore, we update the corresponding w_i according to (11) in the second phase. Hence, we can deflate the influence of those differing points.

B. *Result Analysis*

Tables 1, 2, 3 report the corresponding mean square errors (MSEs) in the NASDAQ, S&P, and FSTE index, respectively. In these tables, we list the results against the different ε's, ranging from 0.0 to 2.0. From these tables, the second phase usually demonstrates better performance than the standard ε-SVR when ε is specified. These observations once again validate that considering data in a proper weight can indeed boost the prediction performance. To make it clear, in Fig.2, we plot the best results given by the first phase and the second phase. Again, the proposed AWSVR demonstrates smaller MSEs in all the three indices. More specifically, the smallest MSEs of the AWSVR are 1.2054 in NASDAQ, 0.9189 in S&P, and 1.5750 in FSTE, while the smallest MSEs of the ε-SVR are 1.2620, 1.0127, and 1.6488, respectively. A paired t-test [16] performed on the best results of two models show that the AWSVR model outperforms the ε-SVR with the α=5% significance level for a one-tailed test.

TABLE I
MSE OF AWSVR AND SVR ON THE NASDAQ INDEX

ε	AWSVR	SVR
0.0	1.3039	1.2620
0.2	1.3024	1.3063
0.4	1.3163	1.3035
0.6	1.2784	1.3175
0.8	1.2253	1.3277
1.0	1.2293	1.3515
2.0	**1.2054**	1.2694

TABLE II
MSE OF AWSVR AND SVR ON THE S&P INDEX

ε	AWSVR	SVR
0.0	1.2124	1.2186
0.2	0.9582	1.2456
0.4	**0.9189**	1.2133
0.6	0.9592	1.1651
0.8	0.9609	1.0571
1.0	1.0283	1.0127
2.0	1.1531	1.1543

TABLE III
MSE OF AWSVR AND SVR ON THE FSTE INDEX

ε	AWSVR	SVR
0.0	1.7037	1.6488
0.2	1.6847	1.6897
0.4	1.7198	1.7030
0.6	1.6536	1.7042
0.8	1.6010	1.7347
1.0	1.5902	1.7482
2.0	**1.5750**	1.6585

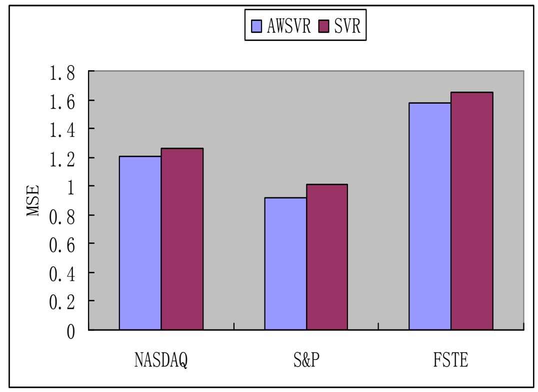

Fig. 2. MSE comparison among the AWSVR and SVR

In order to investigate how the trade-off parameter C influences the performance of the proposed algorithm, we plot the MSEs against C on S&P500 index in Fig. 3. One can find that the performance of the ε-SVR is very sensitive to the choice of C in these high-volatile data, while the performance of the AWSVR almost remains unchanged against different Cs. If one looks back into the optimization problem (8) in the AWSVR, the loss is weighted by w_i around each data point x_i. The non-fixed weight would "absorb" the influence caused by the noise points, and we can even remove the loss term $C\sum_{i=1}^{N} w_i l_\varepsilon(y_i - f(x_i))$ from (8). This phenomenon actually makes our model more appealing. In comparison, in the ε-SVR, the impact caused by outliers are only mitigated slightly by the slack variables ζ_i and ζ_i^* in (4).

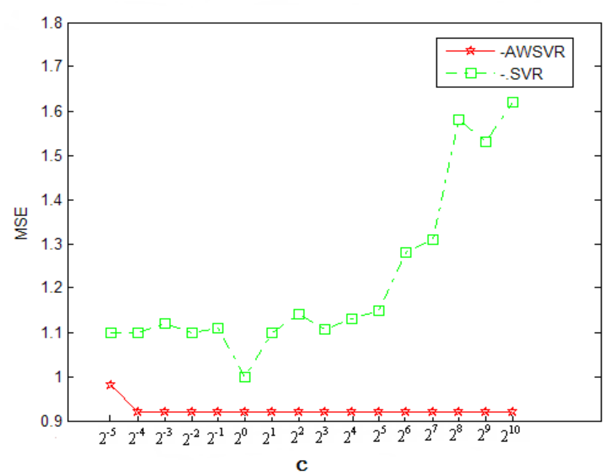

Fig. 3. The influence of C on the performance S&P500

The disadvantage of the proposed AWSVR is its relatively long training time. Though the AWSVR algorithm needs the second phase to solve the adaptive weight problem, its time complexity is polynomial. In the experiments, we observe that the AWSVR costs less than 1s in training all three indices, and it is worth to adopt the weighted learning for financial time series prediction.

V. CONCLUSION

In this paper, a novel "two-phase" SVR training procedure is proposed to mitigate the effect of outliers. This idea motivates from the phenomenon that the weighting and adaptive learning around each data point will be able to refine its loss function with robustness to outliers. The experimental results on three indices indicate that our proposed AWSVR algorithm has improvement on the prediction.

Some remaining works are worth considering. For example, the choice of weights is defined based on the first phase solution and it may not be reasonable anymore to assign weight using the second phase solution. How to consider another choice of weights and construct an iterative procedure using corresponding weights to obtain better prediction, it remains to be a highly challenging yet interesting topic. These are our future works.

REFERENCES

[1] M. Panella, "Advances in biological time series prediction by neural networks," *Biomedical Signal Processing and Control*, vol. 6, pp. 112-120, Apr. 2011.

[2] M. Panella, F. Barcellona, and R. D'Ecclesia, "Subband prediction of energy commodity prices", in *Proc. of 13th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2012),* Çeşme, Turkey, 2012.

[3] H. Yang, L. Chan, I. King, "Support vector machine regression for volatile stock market prediction," in *IDEAL 2002*, 2002, pp. 391-396.

[4] M. Panella, F. Barcellona, V. Santucci, and R. D'Ecclesia, "Neural networks to model energy commodity price dynamics," in *Proc. of the 30th USAEE/IAEE North American Conference USAEE/IAEE,* Washington D.C., U.S.A., 2011.

[5] H. Yang, I. King, L. Chan, K. Huang, "Financial time series prediction using non-fixed and asymmetrical margin setting with momentum in support vector regression," in *J. C. Rajapakse, L. Wang (Eds.), Neural Information Processing: Research and Development, Studies in Fuzziness and Soft Computing,* Springer, Berlin, 2004, vol. 152, pp. 334-350.

[6] H. Q. Yang, K. Z. Huang, L. W. Chan, I. King, and M. R. Lyu, "Outliers treatment in support vector regression for financial time series prediction," in *N. R. Pal et al. (Eds.): ICONIP 2004, LNCS 3316,* Springer-Verlag Berlin Heidelberg 2004, pp. 1260-1265.

[7] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[8] V. N. Vapnik, *The Nature of Statistical Learning Theory, second ed.*, Springer, New York, 1998.

[9] C. -C. Chuang, S. F. Su, J. T. Jeng, and C. C. Hsiao, "Robust support vector regression network for function approximation with outliers,"*IEEE Transtractions on Neural Networks*, vol. 13, pp.1322-1330, Mar. 2002.

[10] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocompt*. ,2001.

[11] H. Q. Yang, K. Z. Huang, I. King, and M. R. Lyn, "Localized support vector regression for time series prediction," *Neurocomputing*, vol. 72, pp. 2659-2669, Sep. 2009.

[12] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods,* Cambridge Univerity Press, Cambridge, 2000.

[13] B. Scholkopf, A. Smola, Learning With Kernels, MIT Press, Cambridge, MA, 2002.

[14] D. P. Bertsekas, *Nonlinear Programming, second ed.*, Athena Scientific, Belmont, MA, 1999.

[15] C. -C. Chang and C. –J. Lin, *LIBSVM: a Library for Support Vector Machines*, 2004.

[16] D. C. Montgomery, G. C. Runger, *Applied Statistics and Probability for Engineers, second ed.*, Wiley, New York, 1999.