

Research Article

Financial Time Series Forecasting Using Directed-Weighted Chunking SVMs

Yongming Cai,¹ Lei Song,¹ Tingwei Wang,¹ and Qing Chang²

¹ School of Management, University of Jinan, Jinan 250002, China

² School of Management, Inner Mongolia University of Technology, Huhhot 010050, China

Correspondence should be addressed to Yongming Cai; cym2001099@163.com

Received 21 February 2014; Accepted 2 April 2014; Published 24 April 2014

Academic Editor: Wei Chen

Copyright © 2014 Yongming Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Support vector machines (SVMs) are a promising alternative to traditional regression estimation approaches. But, when dealing with massive-scale data set, there exist many problems, such as the long training time and excessive demand of memory space. So, the SVMs algorithm is not suitable to deal with financial time series data. In order to solve these problems, directed-weighted chunking SVMs algorithm is proposed. In this algorithm, the whole training data set is split into several chunks, and then the support vectors are obtained on each subset. Furthermore, the weighted support vector regressions are calculated to obtain the forecast model on the new working data set. Our directed-weighted chunking algorithm provides a new method of support vectors decomposing and combining according to the importance of chunks, which can improve the operation speed without reducing prediction accuracy. Finally, IBM stock daily close prices data are used to verify the validity of the proposed algorithm.

1. Introduction

Financial time series forecasting is an important aspect of financial decisions. Financial practitioners and academic researchers have proposed a lot of methods and techniques to improve the accuracy of predictions. Because it is inherently noisy, nonstationary, and deterministically chaotic [1], financial time series forecasting is regarded as one of the most challenging applications of modern time series forecasting.

Time series analysis of the early years study is focused on the regression model, such as autoregression model like AR and ARMA and volatility model, like ARCH and GARCH. In recent years, studies focused on application of artificial intelligence algorithms (AI), such as artificial neural networks (ANN) [2–4], reasoning neural networks (RNN) [5], genetic algorithms (GA) [6], particle swarm optimization (PSO) [7, 8], and support vector machines (SVMs) [9, 10].

Among these artificial intelligence algorithms, SVMs are an elegant tool for solving pattern recognition and regression problems. According to the research of Vapnik [11], SVMs implement the structural risk minimization principle which

seeks to minimize an upper bound of the generalization error rather than minimize the training error. The regression model of SVMs, called support vector regression (SVR), has also been receiving increasing attention to solve linear or nonlinear estimation problems. For instance, Tay and Cao [9] studied the five real future contracts in Chicago Mercantile Market, Cao and Tay [12] studied the S&P 500 daily price index, and Kim [13] studied the daily Korea composite stock price index (KOSPI). Based on the criteria of normalized mean square error (NMSE), mean absolute error (MAE), directional symmetry (DS), and weighted directional symmetry (WDS), the above researches indicate that the performance of SVMs is better than ARMA, GARCH (ARCH), and ANN.

According to the statistical learning theory (SLT), support vector machine regression is a convex quadratic programming (QP) optimization with linear constraint. However, there is an obvious disadvantage of SVMs that the training time scales somewhere between quadratic and cubic depend on the number of training samples. In order to deal with massive data set and improve the training speed, many improved

support vector machine methods are proposed. One way is to combine SVMs with some other methods, such as active learning [14, 15], multitask learning [16, 17], multiview learning [18], and semisupervised learning [19, 20]. Another way is to develop some optimization techniques of training algorithms in SVMs, such as sequential updating methods, like kernel-Adatron algorithm [21], successive over relaxation algorithm [22], working set methods, like chunking SVMs [23, 24], reduced support vector machine (RSVM) [25], and sequential minimal optimization algorithm (SMO) [26].

Chunking SVMs, reduced support vector machine (RSVM), and SVMs with sequential minimal optimization (SMO) are outstanding methods in dealing with massive data set. For example, Lee and Mangasarian [25] designed a reduced support vector machine algorithm (RSVM), which can greatly reduce the size of the quadratic program to be solved by reducing the data set volume, so the memory usage is much smaller than that of a conventional SVM using the entire data set. Osuna et al. [27] designed a decomposition algorithm that is guaranteed to solve the QP problem and that does not make assumptions on the expected number of support vectors. Platt [26] put forward a sequential minimal optimization algorithm (SMO), which breaks the large QP problem into a series of smallest possible QP problems which are analytically solvable in order to speed up the training time. Tay and Cao [28] proposed to combine support vector machines (SVMs) with self-organizing feature map (SOM) for financial time series forecasting, where SOM is used as a clustering algorithm to partition the whole input space into several disjoint regions. Tay and Cao [29] also put forward C -ascending support vector machines to amend the ε -insensitive errors in the nonstationary financial time series.

Most of the improved support vector machine methods do well in memory requirement and CPU time, but the forecast accuracy has declined more or less. For financial time series prediction, the typical massive-scale data, we should pay more attention to the prediction accuracy while reducing the computational complexity. In this paper, we proposed a directed-weighted chunking SVMs algorithm, which can improve the operation speed without reducing prediction accuracy.

This paper consists of five sections. Section 1 introduces the basic algorithm of SVM. Section 2 proposes a directed-weighted chunking SVMs algorithm. Section 3 designs a series of experiments, and empirical results are summarized and discussed. Section 4 presents the conclusions and limitations of this study.

2. SVMs Regression Theory

In this section, we will briefly introduce the support vectors regression (SVR) theory. Suppose there are a given set of data points $G = \{(x_i, d_i)\}_i^l$ (x_i is the input vector; d_i is the desired value). SVMs approximate the function in the following form:

$$y = \sum_{i=1}^l w_i \phi_i(x) + b, \quad (1)$$

where $\{\phi(x)\}_{i=1}^l$ are the features of inputs and $\{w_i\}_{i=1}^l$, b are coefficients.

According to the structural risk minimization principle, the support vectors regression problem can be expressed as

$$\begin{aligned} \text{Minimize:} \quad & R(w, \xi^{(*)}) = \frac{1}{2} w^T w + C^* \sum_{i=1}^l (\xi_i + \xi_i^*), \\ \text{Subject to:} \quad & w\phi(x_i) + b - d_i \leq \varepsilon + \xi_i, \\ & d_i - w\phi(x_i) - b \leq \varepsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, \\ & i = 1, 2, \dots, l, \end{aligned} \quad (2)$$

where x_i is mapped to a higher dimensional space by the function ϕ . ξ_i and ξ_i^* are slack variables (ξ_i is the upper training error; ξ_i^* is the lower), which are subject to the ε -insensitive tube $|d_i - (w\phi(x_i) + b)| \leq \varepsilon$. The parameters which control the regression quality are the cost of error C , the width of the tube ε , and the mapping function ϕ .

Thus, (1) becomes the following explicit form:

$$f(x, a_i, a_i^*) = \sum_{i=1}^l (a_i - a_i^*) K(x, x_i) + b. \quad (3)$$

Then, we can obtain the following form by maximizing the dual form of function (3):

$$\begin{aligned} \phi(a_i, a_i^*) = & \sum_{i=1}^l d_i (a_i - a_i^*) - \varepsilon \sum_{i=1}^l (a_i + a_i^*) \\ & - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (a_i - a_i^*) (a_j - a_j^*) K(x_i, x_j), \end{aligned} \quad (4)$$

with the following constraints:

$$\begin{aligned} \sum_{i=1}^l (a_i - a_i^*) &= 0, \\ 0 \leq a_i &\leq C, \quad i = 1, 2, \dots, l, \\ 0 \leq a_i^* &\leq C, \quad i = 1, 2, \dots, l, \end{aligned} \quad (5)$$

where the a_i , a_i^* are the Lagrange multipliers introduced and $K(x, x_i)$ is named the kernel function. The value is equal to the inner product of two vectors x_i and x_j in the feature spaces $\phi(x_i)$ and $\phi(x_j)$. There are many choices of the kernel function: common examples are the polynomial kernel $K(x_i, x_j) = (x_i x_j + 1)^d$ and Gaussian kernel $K(x_i, x_j) = \exp(-\delta^2(x_i - x_j)^2)$.

Training SVMs is equivalent to optimizing the Lagrange multipliers a_i , a_i^* with constraints based on (4). Good fitting function can be obtained by choosing appropriate functional space.

There are many different researches using the straightforward approaches to construct and implement SVMs

for financial time series analysis (see, e.g., [9, 13, 30]); other methods like chunking SVMs, reduced support vector machine (RSVM), and SVMs with sequential minimal optimization (SMO) are used to deal with massive-scale data sets. Among these methods, SVM chunking provides an alternative method to running a typical SVM on a data set by breaking up the training data and running the SVM on smaller chunks of data. In previous literature, many decomposition and combination methods are proposed.

We would like to mention that, for financial time series prediction problems, the data set from different periods will have different effects on current forecasts, and the change of direction of past stock price will also affect the current forecasts. Therefore, we proposed directed-weighted chunking SVMs, which can improve the operation speed without reducing prediction accuracy.

3. The Directed-Weighted Chunking SVMs

3.1. Chunking Model in Support Vector Regression. Training SVMs is equivalent to solving a linearly constrained QP problem. Training SVMs depends on QP optimization techniques. Standard QP techniques cannot be directly applied to SVMs problems with massive-scale data set. In training stages of directed-weighted chunking SVMs, the whole training set is decomposed into several chunks, and support vectors are calculated, respectively, in their working subset. In the prediction stage, all these support vectors are combined into new working data set to get the model in accordance with their importance as illustrated in Figure 1, and the progress of directed-weighted chunking SVMs algorithm can be described as follows.

Step 1. Decompose the whole training set G into m subsets G_1, G_2, \dots, G_m , and $G = G_1 \cup G_2, \cup \dots, \cup G_m, G_i \cap G_j = \Phi$.

Step 2. Calculate the support vector regression SVG_i for each subset G_i .

Step 3. Calculate the weight and direction of each subset.

Step 4. Combine all support vectors G_{SVM_i} for each subset into new working data set G_{SVM} .

Step 5. Calculate the weighted support vector regression on the new working data set G_{SVM} , and get the model.

3.2. Directed-Weighted Chunking SVMs. In the time series forecasting, such as stock market, the effect of past stock prices on the future stock prices will be different. Usually, the more recent the period of time, the greater the weight coefficient. According to a certain time interval, the total time series of stock prices will be divided into different chunks, and support vectors in each chunk are calculated, respectively. Furthermore, the weighted support vector regressions are calculated to obtain the forecast model.

The stock market is one of the complex systems [31, 32]. We can treat chunks as the nodes and relationship between chunks as edges. Thus, these nodes and edges will be a complex network. The entire time series of stock price can be regarded as a directed-weighted network with a large number of nodes and edges. The mutual influence between chunks has its direction. For example, in chunks G_1 and G_2 , if there exists a nonzero correlation coefficient from chunk G_1 to G_2 , we will draw a directed edge between G_1 and G_2 . Because the strength of mutual influence between chunks is different, the edge weight, which is called correlation intensity, is also different. So, simply chunking SVMs cannot reflect the influence of respective chunk on final model. Here we will introduce directed-weighted chunking algorithm into SVMs.

In traditional SVM regression optimization problem, the parameter C^* is a constant. In order to show the different influence on the prediction results, we introduce a function $Q()$ and modify the SVMs regression function as follows:

$$R(w, \xi^{(*)}) = \frac{1}{2} w^T w + C^* \sum_{i=1}^l Q(G_i, \Delta t) (\xi_i + \xi_i^*). \quad (6)$$

In (6), $Q(G_i, \Delta t)$ can be understood as the weight of each chunk, and the values of $Q(G_i, \Delta t)$ can be positive or negative. The positive or negative values of $Q(G_i, \Delta t)$ will change parameter C^* in SVMs, which is very similar to that of the positive or negative information impact on the future stock price. According to the definition of network correlation coefficient introduced by Bonanno et al. [33], we can define the correlation intensity $Q(G_i, \Delta t)$ (influence of chunk G_i by the time interval Δt) as follows:

$$Q(G_i, \Delta t) = \frac{\langle R(G_i, t) R(G_i, t - \Delta t) \rangle - \langle R(G_i, t) \rangle \langle R(G_i, t - \Delta t) \rangle}{\sqrt{\langle [R(G_i, t) - \langle R(G_i, t) \rangle]^2 \rangle \langle [R(G_i, t - \Delta t) - \langle R(G_i, t - \Delta t) \rangle]^2 \rangle}}, \quad (7)$$

where $\langle \rangle$ is a temporal average always performed over the investigated time period; t represents a time; Δt represents a time interval; $R(G_i, t)$ is the stock returns in a time interval Δt , and $R(G_i, t) = \ln[p(G_i, t)] - \ln[p(G_i, t - \Delta t)]$ is the logarithmic difference of current stock price and the stock price before the time interval Δt .

According to the data point and time interval Δt , original data can be decomposed into several chunks. The boundary of $Q(G_i, \Delta t)$ is $-1 \leq Q(G_i, \Delta t) \leq 1$. If the $Q(G_i, \Delta t)$ is positive, the influence of chunk G_i on the time interval Δt is positive, and vice versa. We can calculate all the correlation intensity $Q(G_i, \Delta t)$ according to (7) and get a $m \times m$ matrix of

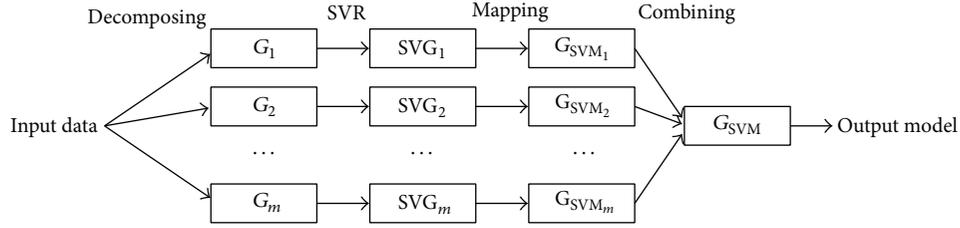
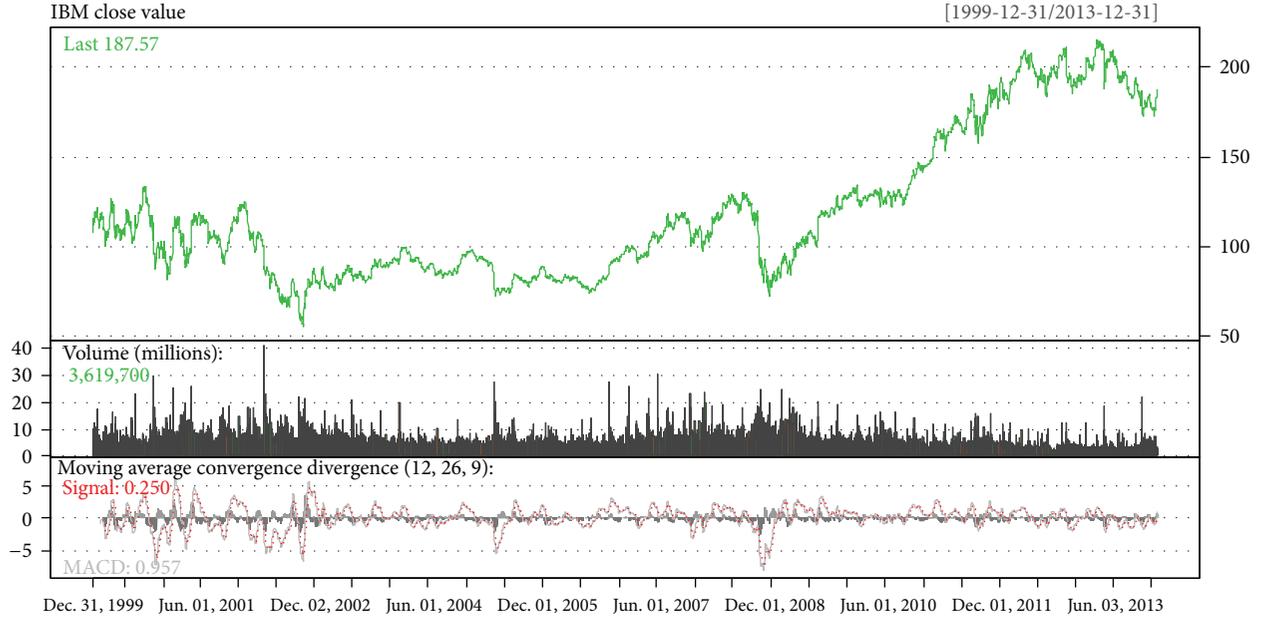


FIGURE 1: Model of directed-weighted chunking SVMs.

FIGURE 2: The IBM stock daily close prices (from December 31, 1999, up to December 31, 2013). Source: Yahoo Finance, <http://finance.yahoo.com/>.

correlation intensity. Using these correlation intensity values, we can get a relationship matrix with direction and weight. Now, dual form of the original optimization problem can be deduced as

$$\begin{aligned} \max \quad & \left\{ -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (a_i - a_i^*) (a_j - a_j^*) K(x_i, x_j) \right. \\ & \left. + \sum_{i=1}^l a_i (d_i - \varepsilon) - a_i^* (d_i + \varepsilon) \right\}, \\ \text{Subject to: } \quad & \sum_{i=1}^l (a_i - a_i^*) = 0, \\ & 0 \leq a_i \leq CQ(G_i, \Delta t), \quad i = 1, 2, \dots, l, \\ & 0 \leq a_i^* \leq CQ(G_i, \Delta t), \quad i = 1, 2, \dots, l. \end{aligned} \quad (8)$$

Now, the solution of this equation is the overall optimal solution of the original optimization problem.

4. Experiments

In order to make a fair and thorough comparative between directed-weighted chunking SVMs and ordinary SVMs, the IBM stock daily close prices are selected as shown in Figure 2. The data points cover the time period from December 31, 1999, up to December 31, 2013, which has 21132 data points.

Data points starting from December 31, 1999, up to December 31, 2007 (12075 data points), are used for training, and data points starting from January 1, 2008, up to the December 31, 2013 (9057 data points), are used for testing. Now we decomposed the training data into 1208 chunks by the time intervals of 10 days and calculated the correlation intensity $Q(G_i, \Delta t)$ according to the function (7). Finally, we obtain a 1208×1208 matrix of correlation intensity.

4.1. Forecast Accuracy Assessment of Directed-Weighted Chunk SVMs. The prediction performance is evaluated by using the following statistical metrics, namely, the normalized mean squared error (NMSE), the mean absolute error (MAE), and the directional symmetry (DS). These criteria are calculated as (9). NMSE and MAE are the measures of the deviation between the actual and predicted values. The values of NMSE

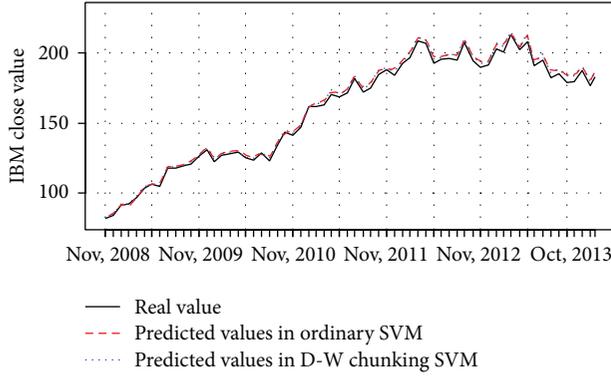


FIGURE 3: The predicted value of IBM stock daily close prices in test data set (from January 01, 2008, up to December 31, 2013).

and MAE denote the accuracy of prediction. A detailed description of performance metrics in financial forecasting can be referred to in Abecasis [34]:

$$\text{NMSE} = \frac{1}{\sigma^2 N} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

$$\text{where, } \sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y}_i)^2, \quad \bar{y}_i = \frac{1}{N} \sum_{i=1}^N y_i,$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|,$$

$$\text{DS} = \frac{100}{N} \sum_{i=1}^N d_i, \quad \text{where } d_i = \begin{cases} 1 & (y_i - y_{i-1}) (\hat{y}_i - \hat{y}_{i-1}) \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

The program of directed-weighted chunking SVMs algorithm is developed using R language. In this paper, the Gaussian function is used as the kernel function of the SVMs. The experiments show that a width value of the Gaussian function of 0.02 is found to produce the best possible results. C and ϵ are arbitrarily chosen to be 8.5 and 10^{-3} , respectively.

We calculate the SVMs on the training data sets (from December 31, 1999, up to December 31, 2007) and then obtain the trained model. Finally, we obtain the predicted result by applying the trained model on the test data set (from January 1, 2008, up to December 31, 2013). In order to compare the differences of various algorithms, real value, predicted value in ordinary SVMs, and the predicted value in directed-weighted chunking SVMs are plotted in Figure 3.

In Figure 3, we can clearly see that these two forecasting methods are very precise, but it is hard to tell which one is more excellent. So we calculated the performance criteria, respectively, as shown in Table 1. By comparing these data, we find that the NMSE, MAE, and DS of directed-weighted chunking SVMs are 0.3760, 0.1325, and 38.29 on the training set and 1.0121, 0.2846, and 43.78 on the test set. It is evident that these values are much smaller than those of ordinary SVMs, which indicates that there is a smaller deviation

TABLE 1: Results of accuracy performance criteria.

	D-W chunking SVMs			Ordinary SVMs		
	NMSE	MAE	DS	NMSE	MAE	DS
Training set	0.3760	0.1325	38.29	0.5625	0.1778	41.67
Test set	1.0121	0.2846	43.78	1.3574	0.2564	48.97

TABLE 2: Performance comparison between the traditional SVMs and directed-weighted chunk SVMs.

SVM method	Sensitivity	Specificity	CPU time (ms)
Traditional SVMs	0.79	0.81	458231
Directed-weighted chunk SVMs.	0.83	0.85	65367

Dataset: IBM stock daily close prices in training data set (12075 data points, from December 31, 1999, up to December 31, 2007).

SVMs parameters: kernel is Gaussian function with $\sigma = 0.02$, $C = 8.5$, $\epsilon = 0.001$, and tolerance = 0.001.

Chunking methods: now we decomposed the training data into 1208 chunks by the time intervals of 10 days.

The CPU time covered the execution of entire algorithm excluding the file I/O time.

between the actual values and predicted values with directed-weighted chunking SVMs.

4.2. Calculation Performance of Directed-Weighted Chunk SVMs. As is well known, the performance of SVM depends on the parameters. But, it is difficult to choose suitable parameters for different problems. Chunk algorithm reused the Hessian matrix elements from one to the next, which can improve the performance sharply.

The calculation performance of all algorithms is measured on an unloaded AMD E-350 1.6 GHz processor running Windows 7 and R 3.0.1. The same experiment will be done on the data set of IBM stock daily close prices. The results of the experiments are shown in Table 2.

The primary purpose of these experiments is to examine differences of training times between two methods. An overall comparison of the SVM methods can be found in Table 2. Compared to the traditional SVMs, directed-weighted chunk SVMs can improve the accuracy and decrease run times sharply. Additionally, the directed-weighted chunk SVMs method allows users to add machines to make the algorithm training even faster.

4.3. Analysis of Optimal Number of Chunks in Directed-Weighted Chunk SVMs. In the experiment described above, we decomposed the training data into 1208 chunks by the time intervals of 10 days arbitrarily and got a satisfactory prediction. Original training data set can be decomposed into 500 chunks or 5000 chunks also. Doing the same experiments on the same training sets by different chunks, we will get a series of performance data. Plotting the curve based on chunks number and NMSE values (in Figure 4), we can intuitively discover relationships between chunks number and errors and get the optimal number of chunks.

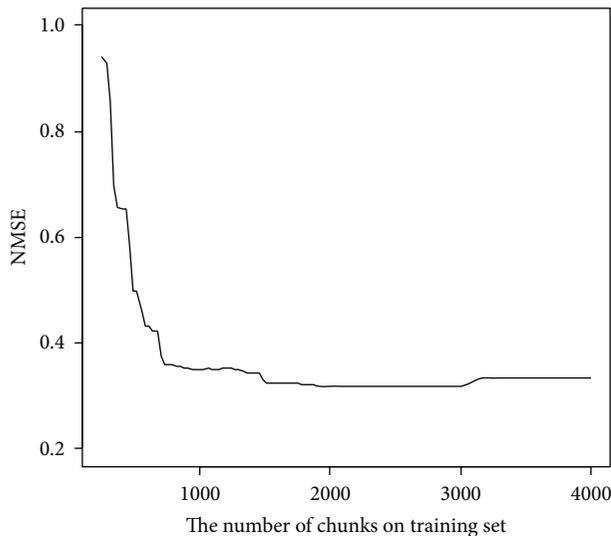


FIGURE 4: NMSE value in different chunks (IBM stock daily close prices data).

According to NMSE criteria, we get the minimum NMSE value 0.3173 on the point of chunks 2350. That means that the best number of chunks is 2350. Under this chunking number, we can get the best performance of prediction.

From Figure 4, when the chunks number is increased, NMSE value is declined rapidly. But, when the decrease reaches a certain value, NMSE value will increase conversely. However, this upward trend is not very large, which indicates that the directed-weighted chunking SVM is not a fundamental transform of SVM, but a limited improvement. But from the perspective of processing massive-scale data, this improvement is very important.

5. Conclusions

In this paper, we proposed a new chunks algorithm in SVMs regression, which combined the support vectors according to their importance. The proposed algorithm can improve the computational speed without reducing prediction accuracy.

In our directed-weighted chunking SVMs, Δt , the criteria for the chunking, is a constant, but, in practice, Δt can be variable or some form of function. In addition, further studies on the different kernel functions and more suitable parameters ε and C can be done in order to improve the performance of directed-weighted chunking SVMs.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (NSFC) under Project Grant (no. 71162015) and the Inner Mongolia Autonomous Region

Higher Education Development Plan of Innovation Teams' Project Grant (no. NMGIRT1404).

References

- [1] Y. S. Abu-Mostafa and A. F. Atiya, "Introduction to financial forecasting," *Applied Intelligence*, vol. 6, no. 3, pp. 205–213, 1996.
- [2] W. Cheng, W. Wagner, and C. H. Lin, "Forecasting the 30-year US treasury bond with a system of neural networks," *Journal of Computational Intelligence in Finance*, vol. 4, no. 1, pp. 10–16, 1996.
- [3] K.-J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.
- [4] H. Ahmadi, "Testability of the arbitrage pricing theory by neural network," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '90)*, pp. 385–393, June 1990.
- [5] R. Tsaih, Y. Hsu, and C. C. Lai, "Forecasting S&P 500 stock index futures with a hybrid AI system," *Decision Support Systems*, vol. 23, no. 2, pp. 161–174, 1998.
- [6] G. G. Szpiro, "Forecasting chaotic time series with genetic algorithms," *Physical Review E*, vol. 55, no. 3, pp. 2557–2568, 1997.
- [7] A. Brabazon and M. O'Neill, *Biologically Inspired Algorithms for Financial Modelling*, Springer, Berlin, Germany, 2006.
- [8] X. Cai, N. Zhang, G. K. Venayagamoorthy, and D. C. Wunsch II, "Time series prediction with recurrent neural networks trained by a hybrid PSO-EA algorithm," *Neurocomputing*, vol. 70, no. 13–15, pp. 2342–2353, 2007.
- [9] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.
- [10] G. Rubio, H. Pomares, I. Rojas, and L. J. Herrera, "A heuristic method for parameter selection in LS-SVM: application to time series prediction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 725–739, 2011.
- [11] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [12] L. Cao and F. E. H. Tay, "Financial forecasting using Support Vector Machines," *Neural Computing and Applications*, vol. 10, no. 2, pp. 184–192, 2001.
- [13] K.-J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.
- [14] S. Sun and D. R. Hardoon, "Active learning with extremely sparse labeled examples," *Neurocomputing*, vol. 73, no. 16–18, pp. 2980–2988, 2010.
- [15] V. Ceperic, G. Gielen, and A. Baric, "Recurrent sparse support vector regression machines trained by active learning in the time-domain," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10933–10942, 2012.
- [16] S. Parameswaran and K. Q. Weinberger, "Large margin multi-task metric learning," in *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS '10)*, Vancouver, Canada, December 2010.
- [17] T. Jebara, "Multi-task feature and kernel selection for SVMs," in *Proceedings of the 21st International Conference on Machine Learning*, pp. 433–440, ACM, July 2004.
- [18] Y. Li, S. Gong, and H. Liddell, "Support vector regression and classification based multi-view face detection and recognition,"

- in *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 300–305, 2000.
- [19] K. Bennett and A. Demiriz, “Semi-supervised support vector machines,” *Advances in Neural Information Processing Systems*, vol. 11, pp. 368–374, 1999.
- [20] C. Brouard, F. D’Alché-Buc, and M. Szafranski, “Semi-supervised penalized output kernel regression for link prediction,” in *Proceedings of the 28th International Conference on Machine Learning*, pp. 593–600, July 2011.
- [21] K. Veropoulos, C. Campbell, and N. Cristianini, “Controlling the sensitivity of support vector machines,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 55–60, Citeseer, 1999.
- [22] O. L. Mangasarian and D. R. Musicant, “Successive overrelaxation for support vector machines,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1032–1037, 1999.
- [23] V. Vapnik, *Estimations of Dependences Based on Statistical Data*, Springer, Berlin, Germany, 1982.
- [24] L. Kaufman, “Solving the quadratic programming problem arising in support vector classification,” in *Advances in Kernel Methods*, pp. 147–167, MIT Press, 1999.
- [25] Y. J. Lee and O. L. Mangasarian, “RSVM: reduced support vector machines,” in *Proceedings of the First SIAM International Conference on Data Mining*, pp. 5–7, Philadelphia, Pa, USA, 2001.
- [26] J. C. Platt, “Using analytic QP and sparseness to speed training of support vector machines,” in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pp. 557–563, Cambridge, UK, 1999.
- [27] E. Osuna, R. Freund, and F. Girosi, “Improved training algorithm for support vector machines,” in *Proceedings of the 7th IEEE Workshop on Neural Networks for Signal Processing (NNSP ’97)*, pp. 276–285, September 1997.
- [28] F. E. H. Tay and L. J. Cao, “Improved financial time series forecasting by combining support vector machines with self-organizing feature map,” *Intelligent Data Analysis*, vol. 5, no. 4, pp. 339–354, 2001.
- [29] F. E. H. Tay and L. J. Cao, “Modified support vector machines in financial time series forecasting,” *Neurocomputing*, vol. 48, pp. 847–861, 2002.
- [30] L. Cao, “Support vector machines experts for time series forecasting,” *Neurocomputing*, vol. 51, pp. 321–339, 2003.
- [31] K. E. Lee, J. W. Lee, and B. H. Hong, “Complex networks in a stock market,” *Computer Physics Communications*, vol. 177, no. 1–2, p. 186, 2007.
- [32] P. Caraiani, “Characterizing emerging European stock markets through complex networks: from local properties to self-similar characteristics,” *Physica A*, vol. 391, no. 13, pp. 3629–3637, 2012.
- [33] G. Bonanno, G. Caldarelli, F. Lillo, S. Micciché, N. Vandewalle, and R. N. Mantegna, “Networks of equities in financial markets,” *European Physical Journal B*, vol. 38, no. 2, pp. 363–371, 2004.
- [34] S. Abecasis, E. Lapenta, and C. Pedreira, “Performance metrics for financial time series forecasting,” *Journal of Computational Intelligence in Finance*, vol. 7, no. 4, pp. 5–22, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

