2014 IEEE 17th International Conference on Computational Science and Engineering

Predicting Stock Trend Using Fourier Transform And Support Vector Regression

Haiying Huang [1], Wuyi Zhang [2], Gaochao Deng [3], James Chen [4]

[1] Department of Marketing, University of Kentucky, Lexington, USA

[2] Department of Computer Science, Hohai University, Nanjing, China

[3] School of International Trade and Economics, Central University of Finance and Economics, Beijing, China

[4] Department of Computer Science, University of Southern California, Los Angeles, USA

E-mail: jasonusa@163.com

***Abstract*— Predicting stock price is an important task as well as difficult problem. Stock price prediction depends on various factors and their complex relationships, which is the act of trying to determine the future value of a company stock. The successful prediction of a stock future price could yield significant profit. This paper demonstrates the applicability of a framework that combines support vector regression and Fourier transform, for predicting the stock price by learning the historic data. Fourier transform is used for noise filtering, and the support vector regression is for model training. Our results suggest that the proposed framework is a powerful predictive tool for stock predictions in the financial market.**

Keywords-stock prediction; data mining; support vector regression; noise filtering; Fourier transform

I. INTRODUCTION

A time series is a collection of observations that measures some activities over time [1]. It records some historical activities, with consistent measurement method, where the measurement is taken at equally spaced intervals, for example, day, week, month, etc. There are a lot of time series in our daily life. For example, the water consumption in Los Angeles recoded in successive days or months is a time series. Among all the different possible time series, the financial time series (stock price) is unusual because it is noisy, non-stationary and uncertain.

For possible monetary gain, currently a number of different supervised machine learning methods have been applied to predict future stock prices in the past decade. K. G. Srinivasa et al. [2] used a neuro-genetic algorithm for stock market prediction, M. A. Kaboudan [3] used genetic programming to predict the stock price, and M. T. Farrell et al. [4] tried Gaussian process regression models for predicting stock trends. However, some of above methods have difficulty in explaining the prediction results, thus suffering from trouble with generalization due to overfitting. In addition, historical stock price data is a massive amount of time-series data with noise. Some stock price prediction frameworks proposed in the above papers do not have explicit solution to deal with the noise in the collected data.

In this paper, we will address the above issues by using support vector regression (SVR) model to capture changes in stock price prediction. It is very natural to think of the price of a stock as some function over time. We will learn this function by supervised learning method SVR. Different to other traditional regression models, SVR not only minimizes the empirical risk (training error), but also minimizes a term which makes the objective function as flat as possible. Additionally, before the SVR is used, Fourier transformation is used for noise filtering, which is necessary to help improve the subsequent learning. This two-step learning framework shown in Figure 1 is reasonable to model the stock price trend.

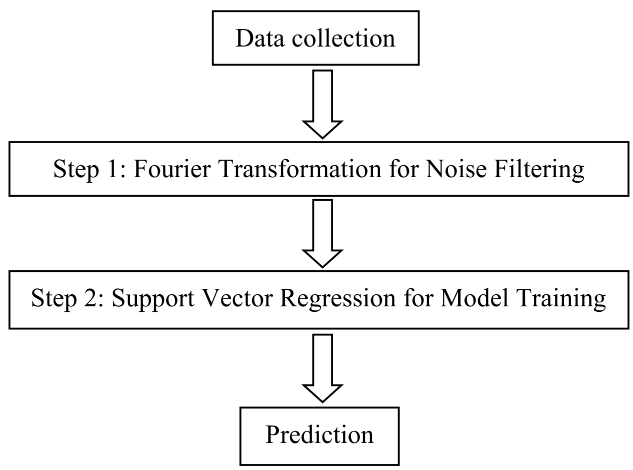

Figure 1. Framework of stock trend prediction using support vector regression

The rest of this article is organized as follows. In Section 2, we will show how to use Fourier transformation to perform noise filtering. In Section 3, we will give brief theoretical overviews of SVR. In Section 4, we focus on the application of SVR on the stocks market. Section 5 presents a discussion.

II. FOURIER TRANSFORM FOR NOISE FILTERING

It is inevitable for a dataset to include some noise that may influence the whole information of the dataset. In the stock market, the prices of stocks fluctuate every day and do not show any signs for forecasting the stock market, thus

978-1-4799-7981-3/14 $31.00 © 2014 IEEE
DOI 10.1109/CSE.2014.70

resulting in difficulty to understand the trend of the change in it. However, from a macroscopical perspective of the stock market, the long-term trend should be predicted and discovered. Although this long-term trend can't give you an obvious indication which specific stock will rise tomorrow, at least it reveals the performance of the whole investment environment and to a certain extend give you an important hints helping you make decisions on the stock market. To better understand and analyze the trend, noise filtering is very necessary. Now, we will use Fourier Transform to analyze the data and realize noise filtering with it.

As what we know, the Fourier Transform is a mathematical operation that decomposes a signal into its constituent frequencies and named after the famed French mathematician Jean Baptiste Joseph Fourier 1768-1830. Today, the Fourier Transform is widely used in science and engineering in digital signal processing and it is a good tool to analyze the frequency in time-series data. The original signal depends on time, and therefore is called the time domain representation of the signal, whereas the Fourier transform depends on frequency and is called the frequency domain representation of the signal. Fourier transform can convert a time domain representation of the signal to its frequency domain representation, which allows people to analyze the data from another perspective. The idea of nose filtering with Fourier Transform is that we firstly remove the frequencies component corresponding to the noise in the perspective of frequency domain representation, and then inverse the frequency domain representation back to the original time domain representation.

Historical data of NASDAQ-100 index will be used for analyzing the trend, which are gathered from the Yahoo's financial web site. The NASDAQ-100 Index includes 100 of the largest domestic and international non-financial securities listed on The Nasdaq Stock Market based on market capitalization. The Index reflects companies across major industry groups including computer hardware and software, telecommunications, retail/wholesale trade and biotechnology. It does not contain securities of financial companies including investment companies. To examine the NASDAQ-100 index we will start with 24 years of daily prices from 2/5/1990 to 1/5/2014. This creates a series of 6027 data points. There is no magic attached to the selection of 21 years of daily data. We just selected this time period for example purposes. We only care about the NASDAQ-100 daily closing price series. Figure 2 is the power spectrum of the original data after doing the Fourier transform. We can see that composition of frequencies in these data. The high peaks with high power represent the important frequency components in these data. Some frequency component with power below certain threshold is considered the frequency component of noise. So we can simply reduce the noise by constructing a new frequency transform without any noise frequencies. Suppose the original data is $x(i), i = 1,2,..,n$, its Fourier Transform can be denoted by $fftx(j), j = 1,2,...,m$. And a new frequency transform we need to construct is represented by $fft2(j)$, $j = 1,2,...,m$. The method for constructing a new frequency transform is as follows:

if $|fftx(j)| > threshold$ then $fftx2(j) = fftx(j)$
if $|fftx(j)| <= threshold$ then $fftx2(j) = 0$

After obtaining a new frequency domain representation of signal, we use inverse Fourier transform to get noise filtered data. In our case, Figure 3 shows a new constructed Fourier transform which is obtained by only keeping the frequency above a certain threshold.

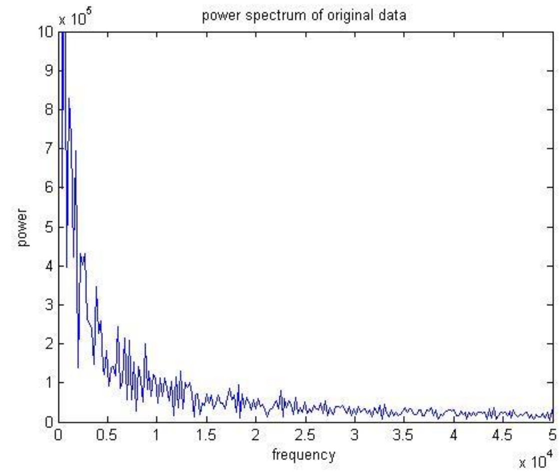

Figure 2. power spectrum of original data

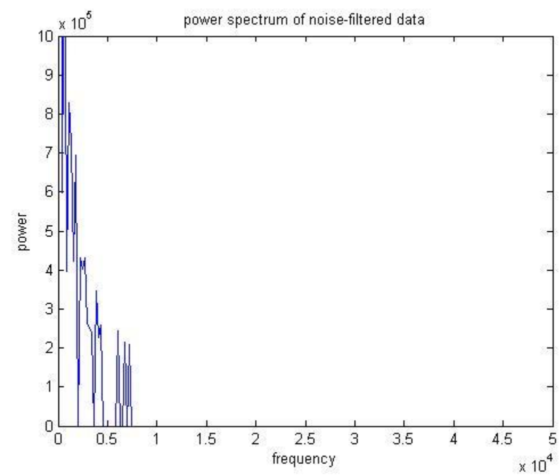

Figure 3. Reconstructed power spectrum

We next perform an inverse FFT on the filtered frequencies, and get the results shown in Figure 4. The blue line shows the original data curve, while the red smooth line is the noise-filtered data curve, which fits the trend of the original data very well.

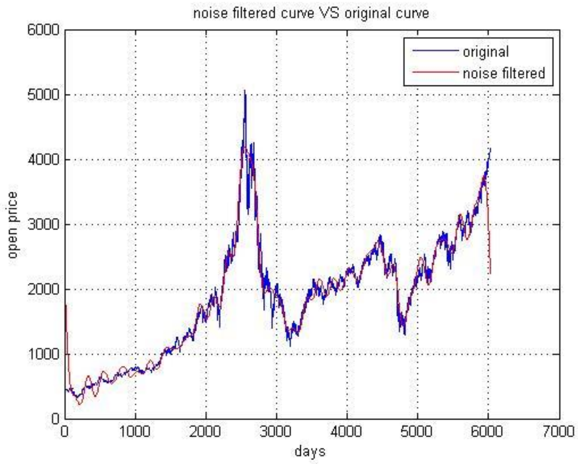

Figure 4. Noise-filtered curve

III. Support Vector Regression

A. Support Vector Regression

Support Vector Machines (SVM) are grounded on the statistical learning theory, or VC theory, which was first developed by Vapnik [5]. The advantages of SVM, such as theoretical background, geometric interpretation, unique solution, and mathematical tractability has made SVM attract researchers' interest and be applied to many applications in various fields with impressive performance, such as computer vision, speech recognition, and text classification. The idea of SVM is to constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class, because in general the larger the margin the lower the generalization error of the classifier. [6,7]

The margin concept used in SVM is also used in the regression problem. In this case, the goal is to construct a hyperplane that lies "close" to as many of the data points as possible. Therefore, the objective is to choose a hyperplane with small norm while simultaneously minimizing the sum of the distances from the data points to the hyperplane [8].

When SVMs were used to solve the regression problem, they were usually called Support Vector Regression (SVR) and the aim of SVR is to find a function f with parameters $\mathbf{w}$ and b by minimizing the following regression risk:

$$R(f) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} l(f(\mathbf{x}_i), y_i), \qquad (1)$$

where C is a tradeoff term, the first term is the margin in SVMs and used to measure VC-dimension [9]. l is a loss function which can be defined as one needs. The function f is defined by

$$f(\mathbf{x}, \mathbf{w}, b) = \langle \mathbf{w}, \emptyset(\mathbf{x}) \rangle + b,$$

where $\emptyset(\mathbf{x})\colon \mathbf{x} \to \Omega$ is kernel function, mapping $\mathbf{x}$ into a high dimensional space.

B. Loss function

We use the ε-insensitive loss function proposed in [9] as follows in SVR,

$$l(y, f(\mathbf{x})) = \begin{cases} 0, & \text{if} \quad |y - f(\mathbf{x})| < \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon, & \text{otherwise} \end{cases}$$

The minimization of Eq. (1) is equivalent to the following constrained minimization problem:

$$\min \gamma(\mathbf{w}, b, \boldsymbol{\xi}^{(*)}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} (\xi_i + \xi_i^*),$$

Subject to

$$y_i - (\langle \mathbf{w}, \phi(\mathbf{x_i}) \rangle + b) \le \varepsilon + \xi_i,$$

$$(\langle \mathbf{w}, \phi(\mathbf{x_i}) \rangle + b) - y_i \le \varepsilon + \xi_i^*,$$

$$\xi_i^{(*)} \ge 0$$

ξ_i and ξ_i^* measure the up error and down error for sample $(\mathbf{x_i}, y_i)$ respectively. A standard method to solve the above minimization problem is to construct the dual problem of this optimization problem (primal problem) by Lagrange Method and to translate it to maximize its dual function.

C. Kernel fuction

Kernel function should satisfy the Mercer's Theorem. Four common kernel functions include:

(1) **Linear function**: $K(\mathbf{x}_k, \mathbf{x}_l) = \langle \mathbf{x}_k, \mathbf{x}_l \rangle$;

(2) **Polynomial function** with parameter d: $K(\mathbf{x}_k, \mathbf{x}_l) = (\langle \mathbf{x}_k, \mathbf{x}_l \rangle + 1)^d$;

(3) **Radial Basis Function** (RBF) with parameters β: $K(\mathbf{x}_k, \mathbf{x}_l) = \exp(-\beta \|\mathbf{x}_k - \mathbf{x}_l\|)^2$;

(4) **Hyperbolic tangent**: $K(\mathbf{x}_k, \mathbf{x}_l) = \tanh(2\langle \mathbf{x}_k, \mathbf{x}_l \rangle + 1)$.

In this paper, we will use Radial Basis Function (RBF) as kernel function.

D. Implemented Algorithm

There are some packages for SVR available on the internet. For example, the package SVM$^{\text{light}}$ of Joachims [10], the libSVM, which is prepared by Chih-Jen Lin [11], the Matlab SVM Toolbox, by Steve Gunn [12]. We will use libSVM in our project.

IV. Prediction Of Stock Trend

Our data set has 6 attributes. They are Open price, High price, Low price, Closing price, Volume, and Adjusted closing price. The regression analysis focus on the Open price on the (t+1)-th day changes when the Open price, High price, Low price, Volume, and Adjusted close price on the *i-th* day vary. Our goal is to fit the following relationship by regression analysis:

$$Open_{t+1} = f(Open_t, High_t, Low_t, Volumn_t, Adj\_Close_t)$$

Before performing the regression, we need to use Fourier Transform to filter noise and normalize the data value on each attribute separately. The regression performances vary with different selection of two important parameters: (1) punishment parameter c and (2) kernel parameter g. Two-stage grid search is used to find the best pair of parameters. A rough and fast Grid search is performed at the first stage (see Figure 5), and then a refined Grid search is performed to shrink the region to locate the optimum solution (see Figure 6). Both Figure 5 and 6 show contour map for SVR parameters selection, where x-axis represents the range of parameter c, y-axis represents the range of parameter g. and the numbers displayed on the contour are the cross validation mean squared error (CVmse). The final parameters selected for this case are c=0.0625 and g=1 with the CVmse 0.00087676. Figure 7 shows the results of our regression by plotting the original data and regressive data together. Mean squared error (MSE) for regression is 3.26706e-05, and correlation R = 99.9423%.

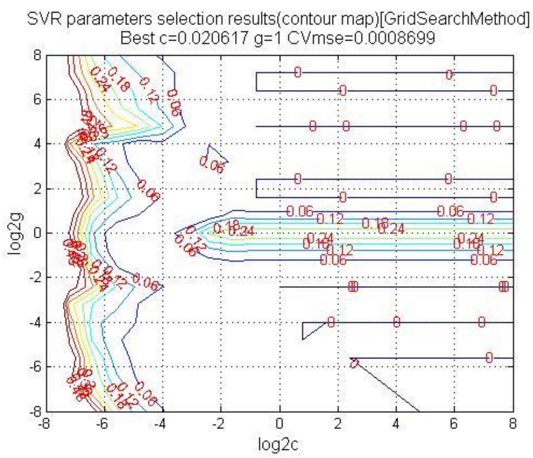

Figure 5. SVR parameters rough selection results (contour map)

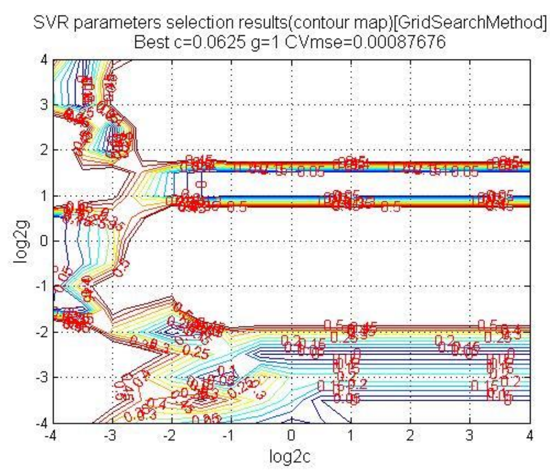

Figure 6. SVR parameters refined selection results (contour map)

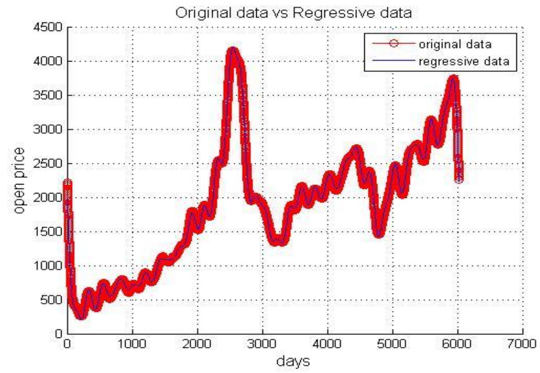

Figure 7. Regression result

V. CONCLUSIONS

In this paper, we study the prediction of stock price by a framework that combine support vector regression and Fourier transform. SVR is a powerful tool for modeling the stock movement and Fourier transform is an efficient approach to address the noise issue. Experiment suggests that the proposed framework is a powerful predictive tool for stock predictions in the financial market.

REFERENCES

[1] C. Chatfield. “The Analysis of Time Series: An Introduction”. Chapman and Hall, fifth edition, 1996.

[2] Srinivasa, K. G., K. R. Venugopal, and Lalit M. Patnaik. "An efficient fuzzy based neuro-genetic algorithm for stock market prediction." International Journal of Hybrid Intelligent Systems 3.2 (2006): 63-81.

[3] Kaboudan, Mark A. "Genetic programming prediction of stock prices." Computational Economics 16.3 (2000): 207-236.

[4] Farrell, M. Todd, and Andrew Correa. "Gaussian process regression models for predicting stock trends." Relation 10 (2007): 3414.

[5] C. Cortes and V. Vapnik, “Support Vector Networks”, Machine Learning, 20, 273-297, 1995.

[6] M.Pontil and A. Verri, “Properties of Support Vector Machines”, Technical Report, Massachusetts Institute of Technology, 1997.

[7] E.E. Osuna, R. Freund and F. Girosi, “Support Vector Machines: Training and Applications”, Technical Report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, AI Memo No. 1602, , 1997.

[8] Xia, Yaqing, Yulong Liu, and Zhiqian Chen. "Support Vector Regression for prediction of stock trend." Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on. Vol. 2. IEEE, 2013.

[9] V. N. Vapnik. The Nature of Statistical Learning Theory. Springer, New York, 1995.

[10] Joachims, Thorsten. "Making large scale SVM learning practical." (1999).

[11] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology (TIST) 2.3 (2011): 27.

[12] Gunn, Steve R. "Support vector machines for classification and regression." ISIS technical report 14 (1998).