Financial Time Series Prediction Using Least Squares Support Vector Machines: an Application Example

Kai Chen

Abstract

This paper reports the work of final project for Econometrics II. In this project, a *Least Squares Support Vector Machine (LS-SVM)* regression model is implemented to predict U.S. Standard & Poor's 500 index (S&P 500) based on 10-year historical data (from 2008 to 2018, 2,429 observations). The performance of out of sample one-step prediction is compared with results from a classical ARIMA model. The Pesaran-Timmermann Test veriefies that LS-SVM model can make significant out of sample sign predictions and the Diebold-Mariano test shows LS-SVM model significantly outperforms ARIMA type models.

Keywords: Financial time series prediction, LS-SVM, ARIMA, model comparison

1. Introduction

The preliminary knowledge on time series prediction and support vector machines can be found in my previous survey paper: Financial Time Series Prediction Using Support Vector Machines.

This paper is organized as follows. The set up and realization of the LS-SVM regression model is illustrated in Section 2. The ARIMA model for comparison is explained in Section 3. The performance analysis, including two statistical tests, are given in Section 4. All the experiments are conducted in Matlab.

2. Least Squares Support Vector Machines (LS-SVM)

2.1. Basic Formulation

An LS-SVM formulation employs the equality constraints and a sum-squared error (SSE) cost function, instead of quadratic program in traditional SVM.

Consider a training data set of N points $D = \{(x_i, y_i)\}_{i=1}^N$ with input data $x_i \in \mathbf{R}^n$ and the response $y_i \in \mathbf{R}$. Then in order to get the LS-SVM model, we need to solve the optimization problem [1]:

$$\min_{w,e} \mathcal{J}(w, e) = \frac{\mu}{2} w^T w + \frac{1}{2} \sum_{i=1}^{N} \zeta_i e_i^2 \tag{1}$$

Email address: kai.chen@rutgers.edu (Kai Chen)

$$\text{s.t.} \quad y_i = w^T\varphi(x_i) + b + e_i, \quad i = 1, \dots, N \tag{2}$$

Here $\varphi(\cdot)$ is the non-linear mapping function to a higher dimensional space. Usually, the RBF kernel is employed, which is also the choice of function in this paper.

Then the Lagrangian for the minimization problem is given by:

$$\mathcal{L}(w, b, e; \alpha) = \mathcal{J}(w, e) - \sum_{i=1}^{N} \alpha_i [w^T\varphi(x_i) + b + e_i - y_i] \tag{3}$$

where $\alpha_i \in \mathbf{R}$ are the Lagrange multipliers (also called support values). The input data points with non-zero Lagrange multipliers α are defined as *Support Vectors*.

Applying the conditions for optimality, one can set the derivatives of L with respect to w, b, e, α as zero:

$$\begin{cases} \frac{\partial \mathcal{L}_1}{\partial w} = 0 \to w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial \mathcal{L}_1}{\partial b} = 0 \to \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial \mathcal{L}_1}{\partial e_i} = 0 \to \alpha_i = \gamma_i e_i, \quad i = 1, \dots, N \\ \frac{\partial \mathcal{L}_1}{\partial \alpha_i} = 0 \to b = y_i - w^T\varphi(x_i) - e_i, \quad i = 1, \dots, N \end{cases} \tag{4}$$

with $\gamma_i = \zeta_i/\mu(i = 1, \dots, N)$ and ζ_i^{-1} is the variance of e_i. Eliminating w and e, one obtains the following linear Karush-Kuhn-Tucker system in α and b:

$$\left[\begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & \Omega + D_\gamma^{-1} \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right] \tag{5}$$

with $y = [y_1; \dots; y_N]$, $1_v = [1; \dots; 1]$, $e = [e_1; \dots; e_N]$, $\alpha = [\alpha_1; \dots; \alpha_N]$ and $D_\gamma = \text{diag}([\gamma_1; \dots; \gamma_N])$. $\Omega(i, j) = K(x_i, x_j)$ is the kernel matrix.

The LS-SVM regression function is thus given by:

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b \tag{6}$$

where α_i and b are the solutions of the linear system (5).

The main benefit of the LS-SVM technique is that it transforms the traditional QP problem to a simultaneous linear system problem, thus ensuring simplicity in computations, fast convergence and high precision.

2.2. *Experiment Result*

We use *holdout* cross-validation technique [2] to split N data points ($N = 2,429$ in our case) into training set and test set. Specifically, the algorithm creates a random nonstratified partition and the size of test set is $p * N$ ($p = 0.1$ in our case). The inputs are simply set as training observations. Readers are encouraged to test the model with more inputs variables, such as some macro-economic explanatory variables, since the codes are written in a style that accepting multi-dimensional inputs. The prediction results are shown in the figure below.

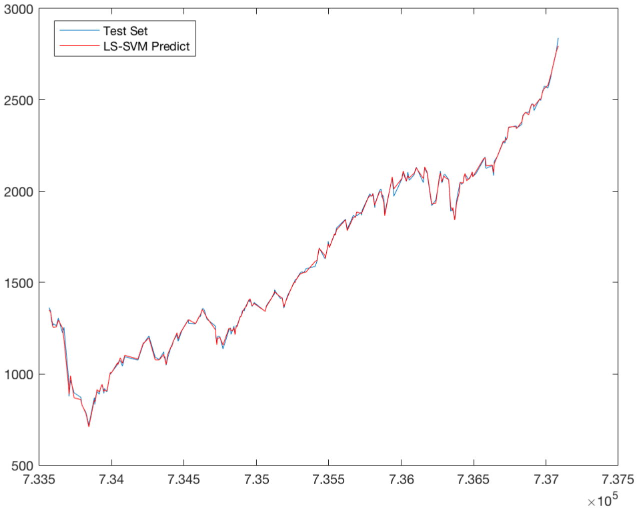

Figure 1: LS-SVM Prediction Result

3. ARIMA Model

3.1. *Basic Formulation*

An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model, which has been widely used in financial time series modeling. The mathematical formulation of ARIMA(p,d,q) using lag polynomials is given by [3]:

$$\varphi(L)(1-L)^d y_t = \theta(L)\varepsilon_t \tag{7}$$

where $\varphi(L) = 1 - \sum_{i=1}^{p} \varphi_i L^i$ and $\theta(L) = 1 + \sum_{j=1}^{q} \theta_j L^j$

Here p, d, and q are integers greater than or equal to zero and refer to the order of autoregressive, integrated, and moving average parts of the model respectively.

3.2. *Experiment Result*

We choose the optimal model structure selected in [4] with lag at 14. In order to make the series stationary, we also employ difference once, which gives us the ARIMA(14,1,0) model. For ARIMA model, we do not apply cross-validation partition, instead we leave out the last $p * N(p = 0.1)$ data points for testing. The out of sample one-step prediction results are shown in the figure below.

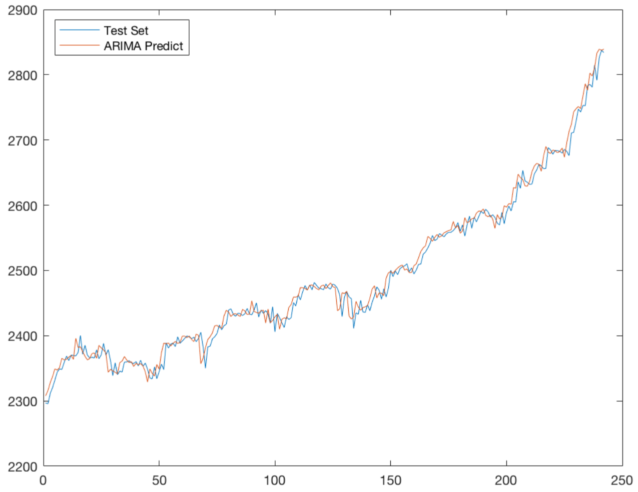

Figure 2: ARIMA Prediction Result

4. Performance Analysis

4.1. Goodness of Prediction

To check the goodness of prediction for our model, we introduce Normalized Mean Square Error (NMSE), which is defined in Equation:

$$NMSE = 1 - \frac{||X_{ref} - X||^2}{||X_{ref} - \bar{X}_{ref}||^2} \tag{8}$$

where X_{ref} is the test data vector and X is the prediction vector. $|| \cdot ||$ indicates the 2-norm of a vector. NMSE varies between $-\infty$ (bad fit) to 1 (perfect fit).

In addition, the Percentage of Correct Sign Predictions (PCSP) is also a good measure of how good a predictor is, which can be simply calculated.

4.2. The Pesaran-Timmermann Test

Pesaran and Timmermann (1992) [5] present a non-parametric test to examine the ability of a forecast to predict the direction of change in a series of interest, i.e. its directional accuracy.

We fist define:

$$I(\cdot) = \begin{cases} 1 & if \cdot > 0 \\ 0 & otherwise \end{cases} \tag{9}$$

Denote the series of interest as y_t and the prediction of it as x_t with n elements. Then we need to calculate:

$$\hat{P}_y = \frac{\sum_{t=1}^{n} I(y_t)}{n}, \quad \hat{P}_x = \frac{\sum_{t=1}^{n} I(x_t)}{n}, \quad \hat{P} = \frac{\sum_{t=1}^{n} I(y_t * x_t)}{n} \tag{10}$$

$$\hat{P}_* = \hat{P}_y \hat{P}_x + (1 - \hat{P}_y)(1 - \hat{P}_x), \quad \hat{V}(\hat{P}) = \frac{\hat{P}_*(1 - \hat{P}_*)}{n} \tag{11}$$

$$\hat{V}(\hat{P}_*) = \frac{(2\hat{P}_y - 1)^2 \hat{P}_x (1 - \hat{P}_x)}{n} + \frac{(2\hat{P}_x - 1)^2 \hat{P}_y (1 - \hat{P}_y)}{n} + \frac{4\hat{P}_y \hat{P}_x (1 - \hat{P}_y)(1 - \hat{P}_x)}{n^2} \tag{12}$$

Under the null hypothesis of x_t not being able to predict y_t, the proposed test is given by:

$$PT = \frac{\hat{P} - \hat{P}_*}{\sqrt{\hat{V}(\hat{P}) - \hat{V}(\hat{P}_*)}} \sim N(0,1) \tag{13}$$

4.3. The Diebold-Mariano Test

Suppose that we have two prediction models for a time series and we want to see which is better in the sense of having better predictive accuracy. The most straightforward approach is to select the model that has smaller error measurement. However, if we want to go one step further and determine whether the difference is significant or simply due to the specific choice of data sets, we then need a statistical test to determine for us.

Diebold and Mariano (1995) [6] formulated this predictive accuracy comparison test. Let e_i and r_i be the residuals for the two prediction series and let d_i be defined as the difference between them. Two possible measurements are:

$$d_i = e_i^2 - r_i^2 \quad \mathit{or} \quad d_i = |e_i| - |r_i| \tag{14}$$

The d_i time series is then called the **loss-differential**. We then define:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i, \quad \mu = E[d_i] \tag{15}$$

For $n > k \leq 1$, define:

$$\gamma_k = \frac{1}{n} \sum_{i=k+1}^{n} (d_i - \bar{d})(d_{i-k} - \bar{d}) \tag{16}$$

Notice that γ_k is the **autocovariance** at lag k.

Under the null hypothesis ($\mu = 0$), for a constant $h \leq 1$, the Diebold-Mariano test is defined as:

$$DM = \frac{\bar{d}}{\sqrt{(\gamma_0 + 2\sum_{k=1}^{h-1} \gamma_k)/n}} \sim N(0,1) \tag{17}$$

It is usually sufficient to use the value $h = n^{1/3} + 1$.

4.4. Comparison Result

Apply the performance measurements to both models. The results are listed in the following table:

Model	NMSE	PCSP	PT(p-value)
RBF-LS-SVM	0.9994	59%	2.1895e-08
ARIMA(14,1,0)	0.9876	50%	0.7742

Table 1: Model Comparison Results

As we can see, LS-SVM model outperforms ARIMA model with respect to all the measurements. In addition, the p-value for Diebold-Mariano test is: 0.0015, which verifies that LS-SVM model is significantly better than ARIMA model in terms of predictive accuracy.

5. Conclusion

In this project, a Least Squares Support Vector Machine (LS-SVM) regression model is implemented to predict S&P 500 index based on 10-year historical data. Then an ARIMA type model is set up to make comparison. The performance measurements and Diebold-Mariano test show that the LS-SVM model outperforms the ARIMA model in the sense of having better predictive accuracy.

References

[1] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (3) (1999) 293–300.

[2] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai, Vol. 14, Montreal, Canada, 1995, pp. 1137–1145.

[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[4] T. Van Gestel, J. A. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, J. Vandewalle, Financial time series prediction using least squares support vector machines within the evidence framework, IEEE Transactions on neural networks 12 (4) (2001) 809–821.

[5] M. H. Pesaran, A. Timmermann, A simple nonparametric test of predictive performance, Journal of Business & Economic Statistics 10 (4) (1992) 461–465.

[6] F. X. Diebold, R. S. Mariano, Comparing predictive accuracy, Journal of Business & economic statistics 20 (1) (2002) 134–144.