ISSN: 1916-9795
E-ISSN: 1916-9809

Journal of Mathematics Research
Vol. 2, No. 2, May 2010

Predict GARCH Based Volatility of Shanghai Composite Index by Recurrent Relevant Vector Machines and Recurrent Least Square Support Vector Machines

Phichhang Ou (Corresponding author)

School of Business, University of Shanghai for Science and Technology

Rm 101, International Exchange Center, No. 516, Jun Gong Road, Shanghai 200093, China

E-mail: phichhang@gmail.com

Hengshan Wang

School of Business, University of Shanghai for Science and Technology

Box 461, No. 516, Jun Gong Road, Shanghai 200093, China

Tel: 86-21-5527-5971 E-mail: wanghs@usst.edu.cn

This work is supported by Shanghai Leading Academic Discipline Project, Project Number: S30504.

Abstract

A new machine learning method so called Relevant Vector Machine (RVM) is an efficiently learning technique for classification and regression problems, including financial time series forecasting. One of the main advantages is that the model is treated by Bayesian approach and its functional form is identical to a powerful prediction tool Support Vector Machine. In this paper, we propose a new recurrent algorithm of the relevant vector machine to predict GARCH (1,1) based volatility of Shanghai composite index. The recurrent support vector machine, recurrent least square support vector machine and normal GARCH (1,1) models are also employed to make a comparison with the proposed model. Our empirical results show that the proposed approach generates superior forecasting performance.

Keywords: Recurrent relevant vector machine, Recurrent least square support vector machine, Volatility forecasting

1. Introduction

Volatility is important for pricing derivatives, calculating measure of risk and hedging. A large number of time series based volatility models have been developed since the introduction of ARCH model of Engle (1982). See Poon and Granger (2003) for review and references. Ability of predicting volatility accurately is a crucial job for stock market researchers and practitioners. Recently, machine learning approaches have been introduced to predict volatility based on various models of GARCH family since they are expected to generate high accuracy of prediction. The empirical results also show that using machine learning approaches combined with GARCH models yield better results. For instance the improved results of forecasting performances by some machine learning techniques can be found in Donaldson and Kamstra (1997) for Neural Network based GJR model, Perez-Cruz et al (2003): SVM based GARCH, Tang et al (2008, 2009) for SVM based GARCH with wavelet and spline wavelet kernels, and Bildirici and Ersin (2009) for Neural Network based on nine different models of GARCH family.

Chen et al (2008b) applied SVM to model and forecast GARCH(1,1) volatility based on the concept of recurrent SVM in Chen et al (2008a), following from the recurrent algorithm of neural network and least square support vector machine of Suykens and Vandewalle (2000). The model was shown to be a dynamic process and capture longer memory of past information than the feed-forward SVM which is just static.

Based on the recurrent SVM result of Chen et al (2008a, 2008b), in this paper we propose the recurrent algorithm for relevant vector machine (RRVM). The RVM, an alternative method of SVM, is a probabilistic model introduced by Tipping in 2000. The RVM has recently become a powerful tool for prediction problems. One of the main advantages is that the RVM has functional form identical to SVM and hence it enjoys various benefits of SVM based techniques: generalization and sparsity. On the other hand, RVM avoids some disadvantages faced by SVM such as the requirement to obtain optimal value of regularized parameter, C, and epsilon tube; SVM needs to use Mercer's kernel function and it can generate point prediction but not distributional prediction in RVM (Tipping, 2001). Our goal here is to compare the proposed recurrent RVM model with other competitive approaches including recurrent SVM, recurrent LSSVM and normal GARCH(1,1) to forecast volatility of Shanghai composite index. It is important for us to forecast the China stock market volatility more accurately. Recently the potential growth of China stock market has attracted foreign and local investors. Annual rate of return for the Shanghai composite index was 81.7% during 2006 and the rapid growth of the rate of return has led to the increasing volatility of this emerging China stock market.

ISSN: 1916-9795
E-ISSN: 1916-9809

The remainder of the paper is organized as follow. Next section summarizes LSSVM and RVM formulations as well as their recurrent algorithms. Section 3 deals with empirical analysis. The last section of the paper is for conclusion.

2. Literature review

2.1 Least Square Support Vector Machines

LSSVM approximates the data $\{x_i, y_i\}$ of the form $y_i = f(x_i) + e_i$ for $i = 1, \cdots, n$ by a nonlinear function defined as

$$y_i = w^T \phi(x_i) + b + e_i \tag{1}$$

The model parameter w is called weight and e_i is random noise. Output $y_i \in R$ can be referred as volatility, while the input vector $x_i \in R^n$ may consist of lagged volatility. Mapping $\phi(\cdot) : R^n \to F$ is nonlinear function that maps the input vector x into a higher dimensional feature space. Estimating the function by the LSSVM is involved in the optimization problem formulated as, Suykens (2000),

objective function

$$\min_{w,b,e} J(w,\ e) = \frac{1}{2} w^T w + \frac{1}{2} y \sum_{i=1}^{n} e_i^2$$

subject to the constraints

$$e_i = y_i - (w^T \phi(x_i) + b) \quad i = 1, \cdots, n$$

Here the equality constraint is used in LSSVM instead of the inequality constraint in SVM. Lagrangian can be defined to solve the above minimization problem as

$$L(w, b, e; \alpha) = J(w, e) - \sum_{i=1^n} \alpha_i (w^T \phi(x_i) + b + e_i - y_i)$$

where α_i denotes Lagrange multipliers (also called support values). From the Karush-Kuhn-Tucker theory, a system of equations is obtained as the following

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \to w = \sum\limits_{i=1}^{n} \alpha_i \phi(x_i) \\ \frac{\partial L}{\partial b} = 0 \to \sum\limits_{i=1}^{n} \alpha_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \to \alpha_i = y_i e_i \quad i = 1, \cdots, n \\ \frac{\partial L}{\partial \alpha_i} = 0 \to b = y_i - w^T \phi(x_i) - e_i, \quad i = 1, \cdots, n. \end{cases} \tag{2}$$

By eliminating w and e_i, the linear system is written as follow

$$[\frac{0}{1_v} | \frac{1_v^T}{\Omega + D_\gamma^{-1}}][\frac{b}{\alpha}] = [\frac{0}{y}] \tag{3}$$

where $y = [y_1, \cdots, y_n]$, $1_v = [1, \cdots, 1]$, $e = [e_1, \cdots, e_n]$, $\alpha = [\alpha_1, \cdots, \alpha_n]$, $D_\gamma = diag([\gamma_1, \cdots, \gamma_n])$.,

Matrix $\Omega_{ij} = \phi(x_i)^T \phi(x_j) = K(x_i,\ x_j)$ for $i,\ j = 1, \cdots, n$ satisfies Mercer's condition. By solving (3), the LS-SVM model for estimating function is shown to be

$$f(x) = w^T \phi(x) + b = \sum_{i=1}^{n} \alpha_i K(x, x_i) + b. \tag{4}$$

In this case the complexity of computing the nonlinear mapping ϕ is avoided. Gaussian kernel or RBF (radial basis function) $K(x_1, x_2) = exp(-\frac{1}{\sigma^2}\|x_1 - x_2\|^2)$ is used in our experiment as it tends to give a good performance under general smoothing assumptions.

2.2 Relevance Vector Machines

For a given training data $\{x_i, t_i\}_{i=1}^n$, the goal is to seek a function indexed by parameter w:

$$y(x; w) = \sum_{j=1}^{m} \omega_j \phi_j(x) = w^T \phi(x). \tag{5}$$

where $\phi(x) = (\phi_1(x), \cdots,\ \phi_m(x))^T$ is nonlinear basis function and $w = (\omega_1, \cdots,\ \omega_m)^T$ is weight vector.

Note that, the function in (5) is identical to SVM based function and it describes the mapping relation between the input vector x and target t with $t_i = \langle x_i, w\rangle + \varepsilon_i$, where $\varepsilon_1, \cdots, \varepsilon_n$ are assumed to be independent Gaussian distribution with mean zero and variance σ^2.

In notation, $p(\varepsilon) = \prod_{i=1}^{n} N(\varepsilon/0, \sigma^2)$.

Thus the likelihood of the complete dataset can be written as

$$p(t/w, \sigma^2) = 2\pi\sigma^{2^{-\frac{n}{2}}} \exp\{-\frac{1}{2\sigma^2}\|t - \Phi w\|^2\} \tag{6}$$

where $t = (t_1, \cdots, t_n)^T$, $w = (\omega_1, \cdots, \omega_m)^T$ and Φ is $(n \times m)$ design matrix with $\Phi = [\phi(x_1), \cdots, \phi(x_n)]^T$ and $\phi(x_i) = [1, K(x_i, x_1), \cdots, K(x_i, x_n)]^T$.

Maximum likelihood estimation of w and σ^2 from (6) will generally lead to overfitting problem. To avoid this advantage, zero mean Gaussian prior over the weights is introduced,

$$p(w/\alpha) = \prod_{i=1}^{n} N(w_i/o, \alpha_i^{-1}) \tag{7}$$

where α_i is the i^{th} element of vector hyperparameter α assigned to each model parameter w_i.

By Bayes rule,

$$p(w, \alpha, \sigma^2/t) = \frac{p(t/w, \alpha, \sigma^2)p(w, \alpha, \sigma^2)}{p(t)} \tag{8}$$

But $p(w, \alpha, \sigma^2/t) = p(w/t, \alpha, \sigma^2)p(\alpha, \sigma^2/t)$ and $p(w/t, \alpha, \sigma^2) = \frac{p(t/w,\sigma^2)p(w/\alpha)}{p(t/\alpha,\sigma^2)}$ is Gaussian distribution $N(\mu, \sum)$ with covariance

$$\sum = (\sigma^{-2}\Phi^T\Phi + A)^{-1} \tag{9}$$

and mean

$$\mu = \sigma^{-1} \sum \Phi^T t \tag{10}$$

where

$$A = diag(\alpha_0, \alpha_1, \cdots, \alpha_n).$$

To evaluate μ and $\sum$, we need to obtain α and σ^2 which maximize

$$p(\alpha, \sigma^2/t) \propto p(t/\alpha, \sigma^2)p(\alpha)p(\sigma^2)$$

By using uniform prior, the problem is to maximize the term $p(t/\alpha, \sigma^2)$ with respect to α and σ^2: . The hyperparameters are estimated by iterative algorithm and can be obtained as $\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}$ and $(\sigma^2)^{new} = \frac{\|t-\Phi\mu\|^2}{n-\sum_i \gamma_i}$ where μ_i is i^{th} posterior mean weight from (10) and $\gamma_i \equiv 1 - \alpha_i \sum_{ii} \in [0, 1]$ can be interpreted as a measure of well determinedness of each parameter w_i. Whereas $\sum_{ii}$ is i^{th} diagonal element of the posterior weight covariance in (9).

During the re-estimation, many α_i tend to infinity such that w will have a few nonzero weights that will be considered as relevance vectors and analogous to the support vectors of SVM. Thus the resulting model enjoys the properties of SVM such as sparsity and generalization.

The predictive distribution for a new input x^* is $p(t^*/t, \alpha_{MP}. \sigma^2_{MP}) = \int p(t^*/w, \sigma^2_{MP})p(w/t, \alpha_{MP}, \sigma^2_{MP})dw = N(t^*/y^*, \sigma^2_*)$ which is easily computed due to the fact that both integrated terms are Gaussian, implying a Gaussian form too with mean $y^* = \mu^T\phi(x^*)$ and variance $\sigma^2_* = \sigma^2_{MP} + \phi(x^*)^T \sum \phi(x^*)$. So the predictive mean is $y(x^*; \mu)$ and the predictive variance composes of two variance components.

2.3 Recurrent Relevance Vector Machines

The recurrent input/output model which is nonlinear output error model is defined as

$$\tilde{y}_t = f(\tilde{y}_{t-1}, \tilde{y}_{t-2}, \cdots, \tilde{y}_{t-p}, u_{t-1}, u_{t-2}, \cdots, u_{t-p}) \tag{11}$$

where $\tilde{y}_t$ denotes the estimated output and f is a smooth nonlinear mapping. $u_t \in R$ is input of any deterministic nonlinear dynamic system and $y_t \in R$ is output.

The corresponding feed-forward input/output model is represented as

$$\tilde{y}_t = f(y_{t-1}, y_{t-2}, \cdots, y_{t-p}, u_{t-1}, u_{t-2}, \cdots, u_{t-p}) \tag{12}$$

ISSN: 1916-9795
E-ISSN: 1916-9809

The models in (11) and (12) can be trained by algorithm of SVM and LSSVM (Suykens and Vandewalle 2000) and hence they can be further trained by the algorithm of RVM since the RVM is of identical form to SVM (Tipping, 2001). Thus we denote RRVM, RSVM and RLSSVM to be the recurrent vector machines obtained by fitting model in (11) by algorithm of RVM, SVM and LSSVM respectively.

The parameterization of f in (12) by the RVM (or LSSVM) is static because there is no recursion in the variable $\tilde{y}_t$. Hence the recurrent models act as nonlinear dynamic process and capture longer memory of past information than the feed-forward models and the parametric models. See (Suykens and Vandewalle, 2000; and Chen et al, 2008a) for detailed discussion on Dynamic system acted by the recurrent LSSVM and recurrent SVM respectively. For simplicity, ARMA model is illustrated as follow:

Linear ARMA(1,1) model estimated by MLE (Maximum likelihood estimation) is described as

$$y_t = \mu + \phi y_{t-1} + e_t + \theta e_{t-1} \tag{13}$$

The nonlinear ARMA(1,1) model estimated by the RRVM (or RLSSVM) can be expressed

$$y_t = f(y_{t-1},\ e_{t-1}) + e_t \tag{14}$$

Then the feed-forward RVM (or LSSVM) corresponding to nonlinear AR(1) is written as

$$y_t = f(y_{t-1}) + e_t \tag{15}$$

Now we turn to GARCH model which is the volatility modeling for asset return. GARCH(1,1) is the most popular form for modeling and forecasting the conditional variance of return or volatility, (Hansen & Lunde, 2005). Therefore, we consider GARCH(1,1) model throughout our paper.

Let P_t be stock price at time. Then $y_t = 100.(\ln P_t - \ln P_{t-1})$ denotes the continuously compounded daily returns of the underlying assets at time t.

AR(1)-GARCH(1,1) is defined as

$$y_t = \mu + \phi_1 y_{t-1} + \varepsilon_t, \quad \varepsilon_t = \sigma_t z_t \tag{16}$$

$$\sigma_t^2 = \omega + \beta_1 \sigma_{t-1}^2 + \alpha_1 \varepsilon_{t-1}^2 \tag{17}$$

Note that conditional variance of ε_t is given by $\sigma_t^2 = E[\varepsilon_t^2 / F_{t-1}] = \hat{\varepsilon}_{t/t-1}^2$. By Bollerslev (1986), the conditional variance of ε_t^2 is the ARMA process given as

$$\varepsilon_t^2 = \omega + (\alpha_1 + \beta_1)\varepsilon_{t-1}^2 + w_t - \beta w_{t-1} \tag{18}$$

by letting $w_t = \varepsilon_t^2 - \hat{\varepsilon}_{t/t-1}^2 = \varepsilon_t^2 - \sigma_t^2$.

Here w_t can be shown to be white noise (or error). The parameters are assumed to be positive to guarantee positive conditional variance: $\omega > 0$ $\alpha_1 \geq 0$, $\beta_1 \geq 0$ and the stationary condition of the covariance requires $\alpha_1 + \beta_1 < 0$. $\{z_t\}$ is a sequence of (iid) independent identically distributed random variables with mean 0 and variance 1. Its one step ahead forecast is $\sigma_{t+1}^2 = \omega + \alpha_1 + \varepsilon_t^2 + \beta_1 \sigma_t^2$.

From (16) and (18), the corresponding nonlinear GARCH model can be formulated as the following:

$$y_t = h(y_{t-1}) + \varepsilon_t$$

$$\varepsilon_t^2 = f(\varepsilon_{t-1}^2,\ w_{t-1}) + w_t$$

where the functions h(.) and f(.) are estimated by feed-forward RVM and by recurrent RVM respectively. Below is the illustration of recurrent algorithm of RVM (or LSSVM) for modeling and forecasting GARCH model.

Step 1: Fit RVM (or LSSVM) to the return y_t as AR(1) format in the full sample period N,

$y_t = h(y_{t-1}) + \varepsilon_t$ for $t = 1, \cdots,\ N$ to obtain residuals $\varepsilon_1,\ \varepsilon_2, \cdots,\ \varepsilon_N$.

Step 2: recursively run the recurrent RVM (or LSSVM) for squared residuals

$\varepsilon_1^2, \cdots,\ \varepsilon_{N_1}^2 (N_1 < N)$ with updating, $\varepsilon_t^2 = f(\varepsilon_{t-1}^2,\ w_{t-1}) + w_t$

to obtain n one-step-ahead forecasted volatilities:

1^{st} sample: $t = 1, \cdots,\ N_1 \rightarrow \hat{\varepsilon}_{N_1+1}^2$,

2^{nd} sample: $t = 1, \cdots,\ N_1 + 1 \rightarrow \hat{\varepsilon}_{N_1+1+1}^2$,

.....

n^{th} sample: $t = 1, \cdots, N_1 + n - 1 \rightarrow \hat{\varepsilon}^2_{N_1+n}$.

For each of n estimations, set the residuals of w_{t-1} to be zero at the first time in the Step 2, and then run the feed-forward RVM (or LSSVM) to obtain estimated residuals. Using the estimated residuals as new w_{t-1} inputs, this process can be carried out repeatedly until the stopping criterion is satisfied. Unlike the parametric case, by using the proposed approach we don't need any assumption on the model parameters for stationary condition.

3. Empirical results

3.1 Data description

We examine Shanghai Composite Index (SSECI) of China Stock Market in the experiment. The stock index price is collected from Yahoo Finance and is transformed into log return before making analysis. The whole sample of size 1564, spanned from 01 Jan. 2001 to 29 Dec. 2006, is used in the experiment to check the predictive capability and reliability of the proposed models. The sub-sample of size 1305, from 04 Jan. 2001 to 31 Dec. 2005, is taken for the in-sample estimation and full one year of 260 points spanned from 02 Jan. to 29 Dec. 2006 is reserved for out of sample forecasting. Table 1 displays the descriptive statistics of the return series of SSECI. The mean of the return is close to zero. The series is positive skewed though the skewness coefficient is not so large in magnitude. The kurtosis value (5.6896) indicates the return has excess kurtosis than the normal value, 3. The large value of Jarque Bera statistic also claims that the return is non-normally distributed. Finally, Ljung Box test of squared return strongly rejects the hypothesis of no ARCH effect. Based on the diagnosis, we can conclude that the return series exhibit volatility clustering and leptokurtic pattern. Therefore it is very suitable to model and forecast the return series by GARCH(1,1) model. We will in next subsection fit this return series by normal GARCH and nonlinear GARCH models.

3.2 In sample estimation or training results

We first fit the return series to equations (16) and (17) to obtain GARCH(1,1) model. The estimation result obtained from Maximum likelihood estimation on GARCH(1,1) with normal innovation is given below:

$$\begin{array}{rccccc} \sigma_t^2 = & 0.1818 & + & 0.7367\sigma_{t-1}^2 & + & 0.1648\varepsilon_{t-1}^2 \\ & [0.028] & & [0.025] & & [0.015] \end{array}$$

The stationary condition holds and the MLE estimates with their corresponding standard errors (0.028, 0.025, 0.015) are all significant. These imply that the model is appropriate and can be further applied for out-of sample forecasting.

Now we turn to consider our proposed model recurrent relevance vector machine, recurrent support vector machine and recurrent least square support vector machine. The proposed models must be trained using the above algorithm stated in Step 1 and Step 2. Table 2, 3, 4 illustrate the training results by RRVM, RLSSVM and RSVM respectively. From the Table 2 the RRVM produces 0.46203 as smallest training error and 0.50961, the variance, as well as 3.7291 to be the optimal value of RBF kernel parameter. The RVM requires 136 relevant vectors with the same number of alphas while training. Figure 1 plots the values of 136 alphas (left) and the values of 136 relevant vectors (right).

By considering Table 3, RLSSVM needs 108.0387 as the optimal value of the regularized parameter and 6.55708 as the RBF kernel parameter while training. But it just generates 0.2744 as the smallest training error. The value of 8.1306 is the constant term of the estimated function by LSSVM.

Finally, Table 4 visualizes the training process of RSVM. Gridsearch technique is used to select the optimal values of cost, C, and RBF kernel parameter, γ which are in the same range $[2^{-5},\ 2^5]$. The optimal parameters are obtained to be $(C,\ \gamma) = (2^5,\ 2^{-4})$ which corresponds to the smallest training error 1.425. Here the epsilon tube is taken to be 0.005.

3.3 Out of sample forecasting

The following evaluation metrics are used to measure the performance and reliability of the proposed models while they are applied to forecast Shanghai composite index volatility: Mean Absolute Deviation (MAD), and Normalized Mean Square Error (NMSE), and Hit Rate which are defined as the following

$$MAD = \frac{1}{n'}\sum_{t=1}^{n}|a_t - p_t|,\ \ NMSE = \frac{1}{s^2 n'}\sum_{t=1}^{n'}(a_t - p_t)^2\ \ where\ \ s^2 = \frac{1}{n'-1}\sum_{t=1}^{n'}(a_t - \overline{a}_t)^2$$

$$Hit\ Rate = \frac{1}{n'}\sum_{t=1}^{n'} d_t\ \ where\ \ d_t = \begin{cases} 1 & (y_t - y_{t-1})(\hat{y}_t - \hat{y}_{t-1}) \\ 0 & otherwise \end{cases}$$

Here $a_t = y_t^2$ actual values, $p_t = \hat{\sigma}_t^2$ forecasted volatility and n' is out of sample size.

ISSN: 1916-9795
E-ISSN: 1916-9809

Also linear regression technique is employed to evaluate the forecasting performance of the volatility models. We simply regress square return on a constant and the forecasted volatility for out-of-sample time point, $t = 1, 2, \cdots, n'$, $y_t^2 = c_0 + c_1\hat{\sigma}_t^2 + e_t$. The square correlation is a measure of forecasting performance.

Table 4 summarizes the forecasting performance based on four measures defined above, MAD, NMSE, R square and Hit Rate. From the table 4, we can see that recurrent RVM generates smallest values of MAD (1.3422) and NMSE (0.7179) but largest value of R square (0.6696) and Hit Rate (0.8416), hence outperforms the other models. Whereas recurrent LSSVM and SVM, they provide better performance than GARCH(1,1) for all cases. Yet, the two models are still competitive. The recurrent SVM is better than recurrent LSSVM based on MAD and R square only, but in term of NMSE and Hit Rate, the recurrent LSSVM is better than the recurrent SVM. Figure 2 plots one step ahead forecasts by the proposed and normal GARCH(1,1) against actual values (upper plot) and the various forecasts by all forecasting models (lower plot). From the bottom plots we can see that though the RVM approach generates better forecasting performances, the difference among the other machine learning techniques is not large; that means the forecasting lines by the three recurrent approaches are almost overlapped.

4. Conclusion

In this paper, we propose recurrent relevance vector machine based on GARCH to forecast volatility of Shanghai composite index. Other corresponding machine learning approaches including recurrent LSSVM and RSVM, as well as normal GARCH(1,1) are employed to make a comparison with the proposed model. The experimental results suggest that the recurrent RVM yields better predictive capability than the other models since it is a dynamic process and can capture longer memory of past information compactly. Furthermore, the RVM takes more advantages than SVM and LSSVM.

References

Bildirici, M., & Ersin, O.O. (2009). Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange. *Expert Systems with Applications*, 7355-7362.

Bollerslev, T. (1986). Generalized Auto Regressive Conditional Heteroskedasticity. *Journal of Econometrics*, pp. 307-327.

Chen, S, Jeong, K., & Hardle, W. K. (2008a). Recurrent Support Vector Regression for a Nonlinear ARMA Model with Applications to Forecasting Financial Returns. *SFB 649 Discussion Paper of Economic Risk*, Berlin.

Chen, S, Jeong, K., & Hardle, W. K. (2008b). Support Vector Regression Based GARCH Model with Application to Forecasting Financial Volatility of Financial Returns. *SFB 649 Discussion Paper of Economic Risk*, Berlin.

Donaldson, R. G., & Kamstra M. (1997). An artificial neural network-GARCH model for international stock return volatility. *Journal of Empirical Finance*, pp. 17-46.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of variance of UK inflation. *Econometrica*, 50, pp. 987-1008.

Hansen, P.R., & Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics*, John Wiley & Sons; pp. 873-889. Doi:10.1002/jae.800.

Perez-Cruz, F., Afonso-Rodriguez, J.A., & Giner, J. (2003). Estimating GARCH models using support vector machines. *Journal of Quantitative Finance*, pp. 163-172.

Poon, S.H., & Granger, C. (2003). Forecasting volatility in financial markets: a review. *Journal of Economic Literature*, 41: pp. 478-539.

Suykens, J.A.K. (2000). Least squares support vector machines for classification and nonlinear modeling. *Neural Network World*, Vol. 10, pp. 29-48.

Suykens, J.A.K., & Vandewalle, J. (2000). Recurrent Least Squares Support Vector Machines. *IEEE Transactions on Circuits and Systems-I*, V. 47, No. 7, pp. 1109-1114.

Tang, L.B., Sheng, H.Y., & Tang, L.X. (2008). Forecasting volatility based on wavelet support vector machine, Expert Systems with Applications.

Tang, L.B., Sheng, H.Y., Tang, L.X. (2009). GARCH prediction using spline wavelet support vector machine. *Journal of Neural Computing and Application*, Springer-Verlag London.

Tipping, M.E. (2000). Relevance Vector Machine. *Microsoft research*, Cambridge, UK.

Tipping, M.E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*.

Table 1. Descriptive statistics of return series

Sample	Min	Max	Mean	Variance	Skewness	Kurtosis	JB	$Q^2(10)$
1564	-6.543	9.4007	0.0163	1.6977	0.6377	5.6896	2224	63.715

Table 2. Training result from Recurrent RVM

	Smallest Training error	Value of Variance	Number of Relevant vectors	Optimal value of RBF Kernel parameter
Recurrent RVM	0.46203	0.50961	136	3.7291

Table 3. Training result from Recurrent LSSVM

	Smallest Training error	Optimal value of the regularized parameter	Optimal value of RBF Kernel parameter	The constant term of the LSSVM function " b"
Recurrent LSSVM	0.2744	108.0387	6.55708	8.1306

Table 4. Training result from Recurrent SVM

	γ										
C	2^{-5}	2^{-4}	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
2^{-5}	4.382	4.104	4.051	4.130	4.248	4.443	4.655	4.883	5.083	5.262	5.459
2^{-4}	3.370	3.365	3.486	3.595	3.752	3.909	4.141	4.402	4.711	4.938	5.183
2^{-3}	2.588	2.681	2.820	2.989	3.183	3.388	3.606	3.896	4.201	4.565	4.830
2^{-2}	2.204	2.230	2.366	2.564	2.747	2.932	3.145	3.423	3.752	4.081	4.420
2^{-1}	1.938	2.070	2.084	2.156	2.339	2.559	2.779	3.005	3.266	3.591	3.962
2^{0}	1.729	1.796	1.889	1.929	2.101	2.251	2.464	2.697	2.937	3.183	3.441
2^{1}	1.640	1.610	1.619	1.699	1.804	1.984	2.215	2.422	2.631	2.811	3.016
2^{2}	1.576	1.496	1.447	1.493	1.599	1.812	2.109	2.259	2.340	2.507	2.678
2^{3}	1.514	1.465	1.436	1.574	1.691	2.121	2.378	2.269	2.336	2.500	2.725
2^{4}	1.485	1.430	1.446	1.695	1.943	2.466	2.858	2.431	2.452	2.709	2.981
2^{5}	1.468	**1.425**	1.521	1.767	2.395	2.815	3.125	2.577	2.780	3.302	3.470

Note: *The table illustrates the cross-validation error corresponding to the tuning parameters (C , γ).*
Here (C , γ)=(2^5,2^{-4}) which corresponds to the smallest training error = 1.425.

Table 5. Forecasting performance based on evaluation metrics by different models

Models	MAD	NMSE	R square	Hit Rate
GARCH(1,1)	1.7446	0.7297	0.4928	0.7760
Recurrent SVM	1.3447	0.7281	0.6629	0.8223
Recurrent LSSVM	1.3636	0.7202	0.6646	0.8301
Recurrent RVM	1.3422	0.7179	0.6696	0.8416

ISSN: 1916-9795
E-ISSN: 1916-9809

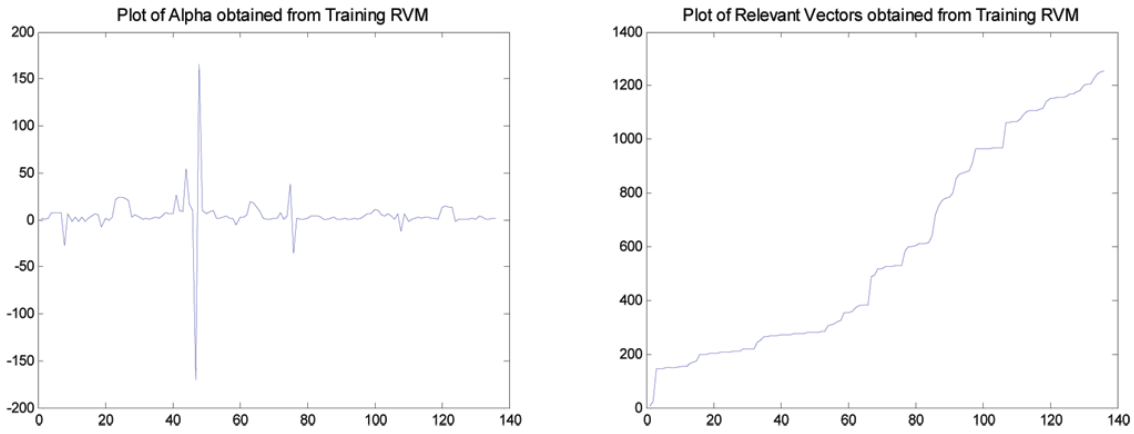

Figure 1. Plot of Alpha (left) and Relevance Vectors (right) obtained from Training RRVM

Note: The horizontal line shows the number of alphas (left figure) and the number of relevance vectors (right figure) while the vertical axis indicates the values of the alpha and relevance vectors.

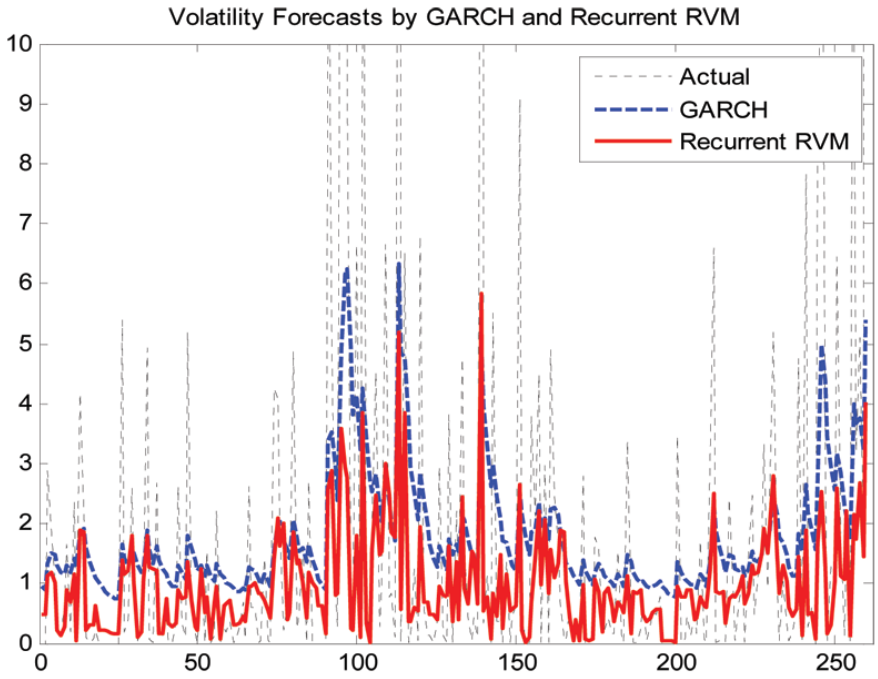

Figure 2. Plots of Volatility Forecasts by GARCH and Recurrent RVM against Actual values

Note: The small dot line is actual value. The dash line is the forecast values by GARCH model and the thick line is the forecasts by recurrent relevance vector machine.

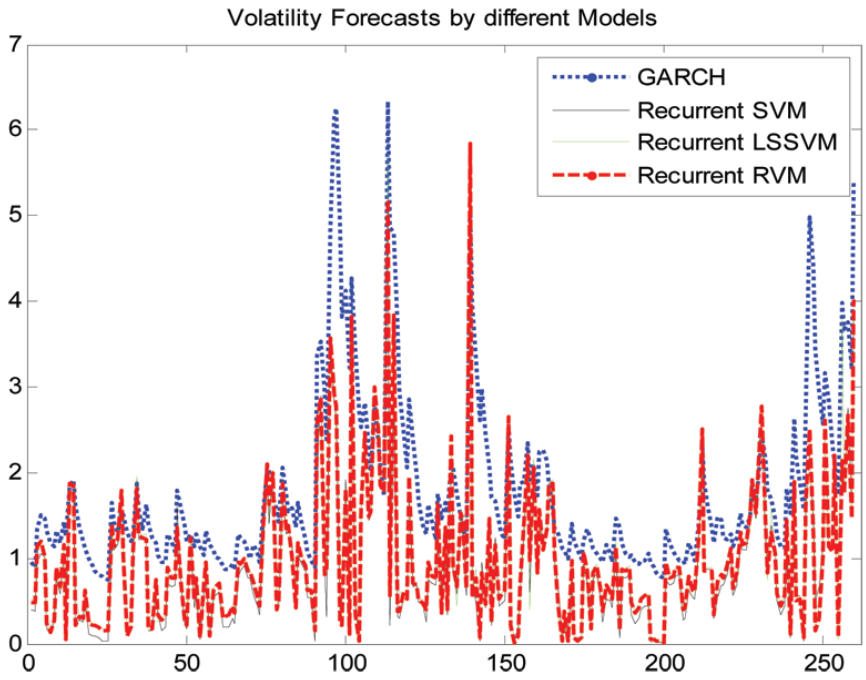

Figure 3. Plots of Volatility Forecasts by GARCH against Recurrent RVM, Recurrent LSSVM, and Recurrent SVM

Note: the Recurrent Models (dash lines) exhibit forecasting points which are closer to the actual values than the parametric GARCH model (dot line). The three recurrent models behave almost the same.