A Hybrid Method Composed of SVRGM(1,1|C,ε) and GARCH(p,q) for Forecasting Complex Time Series

Bao Rong Chang¹ and Shiou Fen Tsai²

¹Department of Computer Science and Information Engineering, National Taitung University, Taiwan
brchang@cc.nttu.edu.tw

²Department of International Business, Shu Te University, Taiwan
soenfen@mail.stu.edu.tw

Abstract - In order to achieve higher accuracy, a hybrid method composed of SVRGM(1,1|C,ε) and GARCH(p,q) applied to forecasting complex financial indexes, e.g. stock price indexes or future trading indexes, is introduced in this study. SVRGM(1,1|C,ε) model is employed to improve the control and environment parameters in grey model GM(1,1) in such a way that can highly reduce the problem of overshooting results in prediction. Moreover, GARCH(p,q) is utilized for dealing with the problem of volatility clustering or fat-tail effect to best fit the prediction model.

Keywords: SVRGM(1,1|C,ε), GM(1,1), GARCH(p,q).

1 Introduction

A hybrid method proposed herein for forecasting the complex time series with higher accuracy. This predictor is composed of SVRGM(1,1|C,ε) [1] and GARCH(p,q) [2] and the predicted output is based on the weighted-average between the predicted outputs of SVRGM(1,1|C,ε) and GARCH(p,q) tuned by a back-propagation neural network [3].

1.1 Background

ARMA [4] has been employed into many scientific or economic applications required a lot of observed data for fitting model but cannot resolve volatility clustering in financial time series. GM(1,1) [5] acts contrary to the aforementioned just acquiring a few data for modeling without any training process. However, it always encounters the overshooting problem [6], for example, a grey prediction for the monthly Taiwan’s stock price index for a period of 31 months from January 1999 to July 2001 as shown in Fig. 1. SVRGM(1,1|C,ε) model employs the support vector regression (SVR) [7] learning algorithm to improve the control and environment parameters in grey model GM(1,1) in such a way that can highly reduce the overshooting results in prediction. Generally, GARCH(p,q) has been utilized to resolve volatility clustering or fat tail effect for the complex time series. Several financial related time series has frequently taken place a severe phenomenon called volatility clustering or fat tail effect [8] that causes ARMA forecasting being not work very well due to the occurrence of time-varying variance. For instance, a sequence of backward-difference from New York D.J.

2 Modeling and learning algorithm

Models are listed in the following subsection such as grey model (GM), support vector regression grey model (SVRGM), and generalized autoregressive conditional heteroscedasticity (GARCH) for constituting the proposed hybrid method.

2.1 Grey Model, GM(1,1)

A prototype of grey model prediction GM(1,1) is introduced in the grey system theory [5].

Step 1: starting with first-order accumulated generating operation (1-AGO)

\[ x^{(1)}(k) = \sum_{j=1}^{k} x^{(0)}(j), \quad k = 1, 2, ..., n \]  (1)

\[ x^{(0)}(k) : \text{the original sampled data that is a nonnegative sequence} \]

Step 2: finding developing coefficient and control coefficient by using grey difference equation

\[ x^{(0)}(k) + \alpha x^{(1)}(k) = b, \quad k = 2, 3, ..., n \]  (2)

\[ \varepsilon^{(0)}(k) = \tau x^{(0)}(k) + (1 - \tau) x^{(0)}(k-1), 0 \leq \tau \leq 1 \]  (3)

\[ \varepsilon^{(1)}(k) : \text{the background value} \]
are denoted by features. In order to obtain $\hat{y}(0)$, first-order accumulated generating operation (1-IAGO) grey differential equation and performing the inverse of regression (SVR) as the form of approximating functions solved by support vector regression learning algorithm.

### Step 3: solving the predicted value $\hat{x}(0)(k)$ through the grey differential equation and performing the inverse of first-order accumulated generating operation (1-IAGO) to obtain $\hat{x}(0)(k)$

\[
\frac{d\hat{x}(0)(k)}{dk} + a\hat{x}(0)(k) = b
\]

\[
\hat{x}(0)(k) = \hat{x}(0)(k) - \hat{x}(0)(k-1)
\]

\[
= (x^{(0)}(1) - b \frac{b}{a} e^{-at(k-1)} - e^{-at(k-2)}), k = 2,3,...
\]

### 2.2 Support vector regression learning algorithm

Initially developed for solving classification problems, SV technology [9] can also be successfully applied in regression, i.e. functional approximation, problems. Unlike pattern recognition problems, where the desired outputs are discrete values like Booleans, problems, SV technology [9] can also be successfully applied in regression, i.e. functional approximation, cases.

According to the learning theory of SVMs, the objective is to maximize the empirical risk and norm-squared of weight vector simultaneously. Thus, estimate a linear regression hyperplane $f(x, w) = w^T x + b$ by minimizing

\[
R(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \xi_i^* \right)
\]

under constrains

\[
y_i - w^T x_i - b \leq \varepsilon, i = 1, 2, ..., l
\]

\[
y_i - w^T x_i + b = \varepsilon + \xi_i, i = 1, 2, ..., l
\]

\[
\xi_i \geq 0, i = 1, 2, ..., l
\]

\[
\xi_i^* \geq 0, i = 1, 2, ..., l
\]

where the constant C influences a trade-off between an approximation error and an estimation error decided by the weight vector norm $\|w\|$, and this design parameter is chosen by the user. $\xi_i$ and $\xi_i^*$ are slack variables as the measurement upper bound and lower bound of outputs.

This quadratic optimization is equivalence to apply Karush-Kuhn-Tucker (KKT) [10] conditions for regression in which maximizing dual variables Lagrangian $L_\alpha(\alpha, \alpha^*)$:

\[
L_\alpha(\alpha, \alpha^*) = \frac{1}{2} \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) x_i^T x_j
\]

\[
- \varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i,
\]

subject to constraints

\[
\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \alpha_i^*,
\]

\[
0 \leq \alpha_i \leq C, i = 1, l
\]

\[
0 \leq \alpha_i^* \leq C, i = 1, l
\]

After calculating Lagrange multipliers $\alpha_i$ and $\alpha_i^*$, find an optimal desired weights vector of the regression hyperplane as

\[
w_0 = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) x_i
\]

and an optimal bias of regression hyperplane as

\[
b_0 = \frac{1}{l} \left( \sum_{i=1}^{n} (y_i - x_i^T w_0) \right)
\]

In non-linear cases for regression, the kernel function, for typical instances, polynomial, RBF, or sigmoid function, will be adopt to replace the scale product $x_i^T x_j$ with $K(x_i, x_j)$ [6] in Eq. (17). If the term $b_i = (\alpha_i - \alpha_i^*)$ is defined in training data set, the output of SVR can be obtained with new input pattern $z$, [9].

\[
y = H \ast b + b_0
\]

where the Hessian matrix $H$ [11] is constructed by

\[
H = \begin{bmatrix}
2 x_i^T x_i & -x_i^T x_j \\
-x_i^T x_j & 2 x_j^T x_j
\end{bmatrix}
\]

and matrix $x$ stands for patterns in training data set as well as matrix $z$ represents new input patterns

\[
x = [x_1, x_2, ..., x_l]^T
\]

\[
z = [z_1, z_2, ..., z_p]^T
\]

Support vector regression (SVR) is herein applied to tune the parameters, $a$ and $b$ in Eq.(2), in grey prediction model for optimizing its generalization capability over the forecasting task. That is, using trained $w_0$ and $b_0$ on Eq. (21-22) as the parameter’s value for $a$ and $b$ in Eq.(2), respectively, is accomplished according to appropriately first arranging Eq. (5) to be the form of Eq. (8) and then training Eq. (5) through SVR learning algorithm to obtain the optimal values for $a$ and $b$ in Eq.(2) instead of the method of least-squared-error in grey model.

### 2.3 ARMAX/GARCH composite model

A GARCH prediction model allows a flexible model description of conditional mean, using a general...
ARMAX form, and conditional variance, employing a GARCH(p,q) process [9]. This ARMAX/GARCH composite model can perform simulation, forecasting, and parameter estimation of univariate time series in the presence of conditional heteroscedasticity, especially in financial time series applications like asset return problem. ARMAX models encompass autoregressive (AR), moving average (MA), and regression (X) models, in any combinations as described below.

\[
j(t) = C + \sum_{i=1}^{p} \alpha_i j(t-i) + \sum_{j=1}^{q} \beta_j X(t-j) + \sum_{k=1}^{n} \gamma_k X(t-k)
\]  

(27)

where \( X \) is an explanatory regression matrix in which each column is a time series and \( X(t,k) \) denotes the \( i \)th row and \( k \)th column.

The GARCH(p,q) models the conditional variance as a standard GARCH process with Gaussian innovations and its mathematical expression is shown as follows.

\[
\sigma^2(t) = k + \sum_{j=1}^{p} \alpha_j \sigma^2(t-j) + \sum_{j=1}^{q} \beta_j \epsilon^2(t-j)
\]

(28)

where \( \sigma^2(t) \) represents the conditional variance and \( \epsilon^2(t-j) \) stands for the \( j \)-lag residual from ARMAX modeling with Gaussian distribution.

3 Hybrid method using BPNN adaptation

The predicted result from proposed hybrid method on the non-period short-term forecasting task is based on the weighted-average between the outputs of SVRGM(1,1|C,ε) and GARCH(p,q). The technique about adjusting the outputs of SVRGM(1,1|C,ε) and GARCH(p,q) is proposed in this study to employ an intelligent computation, back-propagation neural network (BPNN).

3.1 Hybrid method for forecasting

In order to achieve higher accuracy, a hybrid method composed of SVRGM(1,1|C,ε) and GARCH(p,q) applied to forecasting the complex time series is proposed in this study. We denote the predicted outputs of SVRGM(1,1|C,ε) and GARCH(p,q) as \( \hat{y}_{svrgm}(k+1) \) and \( \hat{y}_{garch}(k+1) \), respectively. SVRGM(1,1|C,ε) model is employed to improve the control and environment parameters in grey model GM(1,1) in such a way that can highly reduce the problem of overshooting results in prediction. Moreover, GARCH(p,q) is utilized for dealing with the problem of volatility clustering or fat-tail effect to best fit the prediction model. Therefore, the predicted result \( \hat{y}_{hybrid}(k+1) \) will be proposed to be a combination of the predicted outputs of SVRGM(1,1|C,ε) and GARCH(p,q) as formulated below.

\[
\begin{align*}
\hat{y}_{hybrid}(k+1) &= F(\hat{y}_{svrgm}(k+1), \hat{y}_{garch}(k+1)) \\
&= \alpha(k+1) \cdot \hat{y}_{svrgm}(k+1) + (1 - \alpha(k+1)) \cdot \hat{y}_{garch}(k+1), \\
&= \alpha(k+1) \in [0,1]
\end{align*}
\]

(29)

where the weight \( \alpha \) is determined via the back-propagation neural network (BPNN).

3.2 BPNN weighting mechanism

A well-known intelligent computing machine, three-layer back-propagation neural net (BPNN) [12] is used in this hybrid prediction for tuning the appropriate weights for the forecast \( \hat{y}_{svrgm}(k+1) \) and \( \hat{y}_{garch}(k+1) \) in Eq. (29). For a three-layer BPNN, a structure of 5×16×1 multilayer-perceptron is used that the input layer has 5 input neurons to catch the input patterns, the hidden layer has 16 neurons to propagate the intermediate signals, and the output layer has 1 neuron to display the computed results (weight \( \alpha \) ) as shown in Fig. 1. We arrange the input pattern in the following: a single-step-ahead predicted output from grey model \( \hat{y}_{svrgm}(k+1) \), a single-step-ahead predicted output from C3LSP model \( \hat{y}_{garch}(k+1) \), and three most recent differential values of the true observations denoted by \( \Delta y(k) \), \( \Delta y(k-1) \), and \( \Delta y(k-2) \). Only one appropriate weight, \( \alpha \) applied to the hybrid prediction, is designed as the output. For more training assignments in this three-layer BPNN, the log-sigmoid transfer function is applied as the activations in the hidden layer, the symmetric saturating linear transfer function is employed to the output layer as the activations, and Bayesian regulation involved Levenberg-Marquardt training method is set as the learning algorithm for this three-layer BPNN.

4 Experimental results

As shown in Fig. 4 to Fig. 7 or Fig. 8 to Fig. 9, the predicted sequences indicate the predicted results of GM(1,1), ARMA, and hybrid method. In these experiments, the most recent four actual values is considered as a set of input data used for modeling to predict the next desired output. As the next desired value is obtained, the first value in the current input data set is discarded and joins the latest desired (observed) value to form a new input data set for the use of next prediction. The international stock price indexes prediction for four areas (U.S.A. New York Dow Jones, Taiwan TAIEX, Japan Nikkei Index, and Korea Comp. Index) [13] have been experimented as shown in Fig. 4 to Fig. 7. The accuracy of predicted result, GM, ARMA, SVRGM, GARCH, and the proposed method, is also compared and the summarized first experiment is listed in Table I. The goodness of model fitting on the first experiment is tested by Q-test successfully due to p-value (0.4158 averaged value) greater than level of significance (0.05) [14]. London International Financial Futures and Options Exchange (LIFFE) [15] provides the indices of volumes of equity products on futures and options, and further their indices forecasting tasks are done as shown in Fig. 8 to Fig. 9, and Table II lists the forecasting summary and the comparison between methods are also accomplished. As for the goodness of model fitting, this experiment is also tested by Q-test successfully due to p-value (0.1833 averaged value) greater than level of significance (0.05).
5 Conclusions

Grey model, GM(1,1), might encounter the problem of overshooting phenomenon. ARMA applied to time series forecasting sometimes gets worst accuracy in the short-term non-periodic prediction due to the volatility clustering or fat tail effect. SVRGM(1,1|C, ε) model employing the support vector regression (SVR) learning to improve the control and environment parameters in GM(1,1) achieves the better accuracy because of reducing the overshooting results. Moreover, GARCH(p,q) is utilized for dealing with the problem of volatility clustering or fat-tail effect to best fit the prediction model. In order to achieve higher accuracy on the non-periodic short-term prediction, hybrid method composed of SVRGM(1,1|C, ε) and GARCH(p,q) is proposed herein for forecasting the complex time series and it can definitely obtain the satisfactory predicted results with least mean square error as referring TABLES. This is because the one we proposed can actually smoothing and lessening the phenomena for both overshooting and volatility clustering so as to enhance its generalization capability and bootstrap the prediction accuracy highly.

6 Acknowledgements

This work is fully supported by the National Science Council, Taiwan, Republic of China, under grant number NSC 92-2213-E-143-001.

References


TABLE I.
The mean squared error (MSE) between the desired values and the predicted results for international stock price monthly indices is up to 41 months from Aug. 2000 to Dec. 2003. (unit=10^5)

<table>
<thead>
<tr>
<th>Methods</th>
<th>NY-D.J. Index</th>
<th>Taiwan TAIEX Index</th>
<th>Japan Nikkei Index</th>
<th>Korea Composite Index</th>
<th>Average of MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>4.0577</td>
<td>2.8018</td>
<td>4.7121</td>
<td>0.0481</td>
<td>2.9049</td>
</tr>
<tr>
<td>ARMA</td>
<td>7.4955</td>
<td>5.5694</td>
<td>7.1935</td>
<td>0.0855</td>
<td>5.0860</td>
</tr>
<tr>
<td>SVRGM</td>
<td>3.1782</td>
<td>2.5189</td>
<td>3.9918</td>
<td>0.0455</td>
<td>2.4336</td>
</tr>
<tr>
<td>GARCH</td>
<td>2.8931</td>
<td>2.1325</td>
<td>3.8290</td>
<td>0.0346</td>
<td>2.2223</td>
</tr>
<tr>
<td>HYBRID</td>
<td>2.5483</td>
<td>1.9972</td>
<td>3.7015</td>
<td>0.0298</td>
<td>2.0692</td>
</tr>
</tbody>
</table>

TABLE II.
The mean squared error (MSE) between the desired values and the predicted results for the volumes of equity products on futures monthly index and options monthly index is up to 24 months from Jan. 2001 to Dec. 2002.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Equity Products on Futures Index</th>
<th>Equity Products on Options Index</th>
<th>Average of MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>0.0945</td>
<td>0.0138</td>
<td>0.0542</td>
</tr>
<tr>
<td>ARMA</td>
<td>0.0547</td>
<td>0.0114</td>
<td>0.0331</td>
</tr>
<tr>
<td>SVRGM</td>
<td>0.0830</td>
<td>0.0088</td>
<td>0.0459</td>
</tr>
<tr>
<td>GARCH</td>
<td>0.0648</td>
<td>0.0091</td>
<td>0.0370</td>
</tr>
<tr>
<td>HYBRID</td>
<td>0.0508</td>
<td>0.0083</td>
<td>0.0295</td>
</tr>
</tbody>
</table>

Figure 1. The desired sequence represents the monthly Taiwan's stock price index for a period of 31 months from January 1999 to July 2001. The outputs of GM(1,1) model indicated by the predicted sequence 1 denoted by “•” reveals a crucial problem, an overshooting effect might happen around the turning points at sample number 7, 14, 25, 26, and 27. The underestimated predicted outputs may be resulted from a cumulated 3-point least squared linear model indicated by the predicted sequence 2 denoted by “○” around the turning points at sample number 6, 12, 13, 25 and 26.

Figure 2. A sequence of backward-difference from New York D.J. Industry Index has displaced to its mean value dated from January 1999 to August 2003 for a period of 56 months. This plot showing the volatility clustering as big changes happened around sample number 2-4, 12-16, 27-35, and 45-47 as well as small changes revealed around sample number 5-11, 17-19, 21-26, 36-44, and 48-56.

Figure 3. A structure of 5×16×1 BPNN is used to compute weight ω during training phase. Activation function in neurons is set by log-sigmoid function in hidden layer, and symmetric saturating linear function in output layer.
Figure 4. Forecasts of N. Y. -D. J. Indus. monthly index for 41 months from Aug. 2000 to Dec. 2003.

Figure 5. Forecasts of Taiwan TAIEX monthly index for 41 months from Aug. 2000 to Dec. 2003.

Figure 6. Forecasts of Japan Nikkei monthly index for 41 months from Aug. 2000 to Dec. 2003.

Figure 7. Forecasts of Korea Comp. monthly index for 41 months from Aug. 2000 to Dec. 2003.

Figure 8. Forecasts of equity products on futures monthly index for 24 months from Jan. 2001 to Dec. 2002.

Figure 9. Forecasts of equity products on options monthly index for 24 months from Jan. 2001 to Dec. 2002.