SVR with Nonlinear Conditional Heteroscedasticity

Bao Rong Chang[1] and Shiou Fen Tsai[2]
[1]Department of Computer Science and Information Engineering, National Taitung University, Taiwan
brchang@cc.nttu.edu.tw
[2]Department of International Business, Shu Te University, Taiwan
soenfen@mail.stu.edu.tw

Abstract - A support vector regression (SVR) is very useful on modeling the predictor for forecasting complex time series. However, SVR cannot avoid volatility clustering effect and thus worst its prediction accuracy. NGARCH(p,q) is utilized for dealing with the problem of volatility clustering or fat-tail effect to best fit the modeling. Therefore, SVR with nonlinear conditional heteroscedasticity is introduced herein in order for dealing with volatility clustering phenomenon while it is applied to forecasting complex financial indexes, e.g. stock price indexes or future trading indexes. This proposed method can get the satisfactory results because of improving its generalization.

Keywords: Support vector regression, NGARCH(p,q), Heteroscedasticity.

1 Introduction

A support vector regression with nonlinear conditional heteroscedasticity is schemed for smoothing the volatility clustering effect in complex time series. This scheme is to take nonlinear time-varying conditional variance of residual into account [1]. The evaluation of nonlinear time-varying conditional variance of residual may be realized from NGARCH(p,q) [2]. A weighted nonlinear conditional heteroscedasticity is added into SVR and this weight is 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. Support vector regression (SVR) employing the support vectors for regression modeling [7] can improve the smoothness of fitted model because of highly reducing the occurrence of possibility of over-fitting or under-fitting situation. However, SVR cannot avoid volatility clustering effect and thus worst its fitness. Generally, NGARCH(p,q) has been utilized to resolve volatility clustering or fat tail effect [8] in the complex time series. For instance, 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 as shown in Fig. 2. Volatility clustering has occurred at several intervals and this phenomenon can be viewed as nonlinear time-varying conditional variance of residuals.

1.2 Motivation

NGARCH(p,q) is well-known for dealing with the issue of nonlinear conditional heteroscedasticity to smooth data fitting if required. The proposed scheme is to incorporate embedded nonlinear time-varying conditional variance of residual into SVR. In other words, a weighted nonlinear conditional heteroscedasticity is added into SVR and this weight is tuned by a back-propagation neural network (BPNN).

2 Support vector regression

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, here there are real-valued functions [10]. We consider approximating functions solved by support vector regression (SVR) as the form of

$$f(x,w)=\sum_{i=1}^{l} w_i \phi(x), \tag{1}$$

where $\phi(x)$ are denoted by features. In order to introduce all relevant and necessary concept of SV regression in a gradual way, linear regression is considered first.

$$f(x,w)=w^T x+b \tag{2}$$

Furthermore, Vapnik introduced a general type of loss function, namely, error, the linear loss function with ε-insensitivity zone [8]:

$$\left|y-f(x,w)\right|_\varepsilon=\begin{cases}0 & if\left|y-f(x,w)\right|\le\varepsilon,\\ \left|y-f(x,w)\right|-\varepsilon & otherwise\end{cases} \tag{3}$$

A new empirical risk is introduced for performing SVM regression.

$$R_{emp}(w,b)_\varepsilon=\frac{1}{l}\sum_{i=1}^{l}\left|y_i-w^T x_i-b\right|_\varepsilon \tag{4}$$

According to the learning theory of SVMs, the objective is to minimize 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

$$R(w,\xi,\xi^*)=\frac{1}{2}\|w\|^2+C\left(\sum_{i=1}^{l}\xi_i+\sum_{i=1}^{l}\xi_i^*\right), \tag{5}$$

under constrains

$$y_i - w^T x_i - b \le \varepsilon + \xi_i,\ i = 1,...,l \quad (6)$$

$$w^T x_i + b - y_i \le \varepsilon + \xi_i^*,\ i = 1,...,l \quad (7)$$

$$\xi_i \ge 0,\ i = 1,...,l \quad (8)$$

$$\xi_i^* \ge 0,\ i = 1,...,l \quad (9)$$

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. ξ_i and ξ_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 $L_d(\alpha, \alpha^*)$:

$$L_d(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i^T x_j - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)y_i, \quad (10)$$

subject to constraints

$$\sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^*, \quad (11)$$

$$0 \le \alpha_i \le C,\quad i = 1,...,l \quad (12)$$

$$0 \le \alpha_i^* \le C,\quad i = 1,...,l \quad (13)$$

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

$$w_0 = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i \quad (14)$$

and an optimal bias of regression hyberplane as

$$b_0 = \frac{1}{l}\left(\sum_{i=1}^{l}(y_i - x_i^T w_0)\right). \quad (15)$$

In non-linear cases for regression, the kernal 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. (10).

If the term $\beta_i = (\alpha_i - \alpha_i^*)$ is defined in training data set, the output of SVR can be obtained with new input pattern z_i [9].

$$y = g\beta + b_0 \quad (16)$$

where the vector g is constructed by

$$g = z_i^T x, \quad (17)$$

and matrix x stands for patterns in training data set as well as vector z_i represents new input pattern.

$$x = [x_1 x_2 ... x_l] \quad (18)$$

$$z_i = [z_{i1}, z_{i2}, ..., z_{iN}]^T \quad (19)$$

3 ARMAX/NGARCH composite model

ARMAX/NGARCH composite model allows the description of the conditional mean using ARMAX and the conditional variance employing NGARCH(p,q) [9]. In the presence of conditional heteroscedasticity, this composite model can perform the estimation, simulation, and prediction for the univariate time series, especially in financial time series applications like asset return problem. The ARMAX encompass autoregressive (AR), moving-average (MA), and regression (X) models, in any combinations as expressed below.

$$y(t) = C + \sum_{i=1}^{r}R_i y(t-i) + \varepsilon(t) + \sum_{j=1}^{m}M_j \varepsilon(t-j) + \sum_{k=1}^{N_X}\beta_k X(t,k) \quad (19)$$

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

The NGARCH(p,q) consists of nonlinear time-varying conditional variances and Gaussian innovations. Its mathematical formula is shown as follows.

$$\sigma^2(t) = K + \sum_{i=1}^{p}G_i\sigma^2(t-i) + \sum_{j=1}^{q}A_j\sigma^2(t-i)\left[\frac{\varepsilon(t-j)}{\sqrt{\sigma^2(t-i)}} - C_j\right]^2 \quad (20)$$

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

4 SVR with nonlinear time-varying conditional variance

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).

4.1 Combining SVR with nonlinear time-varying conditional variance of residual

In order to achieve higher generalization in SVR modeling, a nonlinear time-varying conditional variance of residual is consider as a part of SVR results. We denote the output of SVR and nonlinear time-varying conditional variance of residual as $\hat{y}_{svr}(k+1)$ and $\hat{\varepsilon}_{ngarch}(k+1)$, respectively. $\hat{\varepsilon}_{ngarch}(k+1)$ is employed to improve the generalization capability of SVR modeling in such this way that can highly reduce the occurrence of possibility of over-fitting or under-fitting. Therefore, the overall result $\hat{y}_{svrnch}(k+1)$ will be proposed to be a combination of the output of SVR and nonlinear time-varying conditional variance of residual as formulated below.

$$\hat{y}_{svrnch}(k+1) = \hat{y}_{svr}(k+1) + \varpi(k+1)\cdot\hat{\varepsilon}_{ngarch}(k+1)\cdot rand(1), \quad \varpi(k+1) \in [0,1] \quad (21)$$

where the weight ϖ is determined by applying back-propagation neural network (BPNN). $rand(1)$ represents a random number ranged from 0 to 1.

4.2 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 $\hat{y}_{svrnch}(k+1)$ and $\hat{\varepsilon}_{ngarch}(k+1)$ in Eq. (21). 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 ϖ) as shown in Fig. 1. We arrange the input pattern in the following: a single-step-ahead predicted output from grey model $\hat{y}_{svrnch}(k+1)$, a single-step-ahead predicted output from C3LSP model $\hat{\varepsilon}_{ngarch}(k+1)$, 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, ϖ , 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.

5 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 meyhod. 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 (0.3715 in average) 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 (0.1278 in average) greater than level of significance (0.05).

6 Conclusions

SVR cannot avoid volatility clustering effect and thus worst its generalization capability. NGARCH(p,q) is utilized for dealing with the problem of volatility clustering or fat-tail effect to best fit the modeling. Therefore, NGARCH(p,q) provides nonlinear time-varying conditional variance of residual to combine SVR for dealing with volatility clustering phenomenon while SVR is applied to forecasting complex financial indexes, e.g. stock price indexes or future trading indexes. This proposed method can get the satisfactory results because its generalization is much improved.

7 Acknowledgements

This work is fully supported by the National Science Council, Taiwan, Republic of China, under grant number **NSC 93-2218-E-143-001**.

References

[1] B. R. Chang, “Grey Model with Support Vector Regression Learning- SVRGM(1,1),” *Journal of Grey System*, Vol. 16, No. 2, pp. 103-114, 2004.
[2] T. Bellerslve, “Generalized Autoregressive Conditional Heteroscedasticity”, *Journal of Econometrics*, Vol. 31, pp. 307-327, 1986.
[3] O. Castillo and P. Melin, “Simulation and Forecasting Complex Economic Time Series Using Neural Network and Fuzzy Logic,” *Proc. of IEEE-IJCNN*, pp.1805-1810, 2001.
[4] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting & Control*, New Jersey: Prentice-Hall, 1994.
[5] J. L. Deng, Control Problems of Grey System,” *System and Control Letter*, Vol. 1, No. 5, pp. 288-94, 1982.
[6] B. R. Chang, “A Study of Non-Periodic Short-Term Random Walk Forecasting Based on RBFNN, ARMA, or SVR-GM(1,1,tau) Approach,” *Proc. of IJCNN 2003*, pp. 254-259, 2003.
[7] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.
[8] Gourieroux, C., *ARCH Models and Financial Applications*, New York: Springer-Verlag, 1997.
[9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and other kernel-based learning methods)*, London: Cambridge University Press, 2000.
[10]V. Kecman, *Learning and Soft Computing*, Massachusetts: MIT Press, 2001.
[11]E. Kreyszig, *Advanced Engineering Mathematics*, 8th Edition, Wiley, New York, 1999.
[12]L. Smith, “An Introduction to Neural Network,” http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html, 2001.
[13]International Stock Price Index, FIBV FOCUS MONTHLY STATISTICS, 2002.
[14]G. M. Ljung, and G. E. P. Box, “On a Measure of Lack of Fit in Time Series Models,” *Biometrika*, Vol. 65, pp. 67-72, 1978.
[15]London International Financial Futures and Options Exchange (LIFFE), http://www.liffe.com/, 2002.

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)

Methods	N Y- D.J. Indus. Index	Taiwan TAIEX Index	Japan Nikkei Index	Korea Composite Index	Average of MSE
GM	4.0577	2.8018	4.7121	0.0481	2.9049
ARMA	7.4955	5.5694	7.1935	0.0855	5.0860
SVRGM	3.1782	2.5189	3.9918	0.0455	2.4336
GARCH	2.8931	2.1325	3.8290	0.0346	2.2223
SVR	2.3300	2.0803	3.6498	0.0298	2.0225
SVRNCH	2.2119	1.8611	3.5425	0.0287	1.9111

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.

Methods	Equity Products on Futures Index	Equity Products on Options Index	Average of MSE
GM	0.0945	0.0138	0.0542
ARMA	0.0547	0.0114	0.0331
SVRGM	0.0830	0.0088	0.0459
GARCH	0.0648	0.0091	0.0370
SVR	0.0097	0.0065	0.0081
SVRNCH	0.0095	0.0063	0.0079

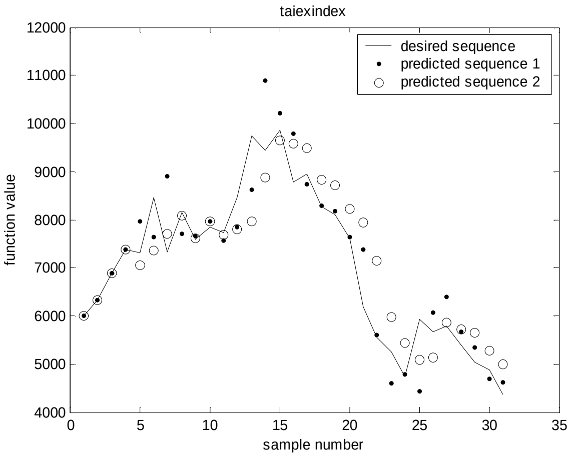

Figure 1. The desired sequence represents the monthly Taiwan's stock price index for a period of 31 months from January 1999 to July 2001. The outputs of GM(1,1) model indicted by the predicted sequence 1 denoted by "•" reveals a crucial problem, an overshooting effect might happen around the turning 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 "o" around the turning points at sample number 6, 12,13, 25 and 26.

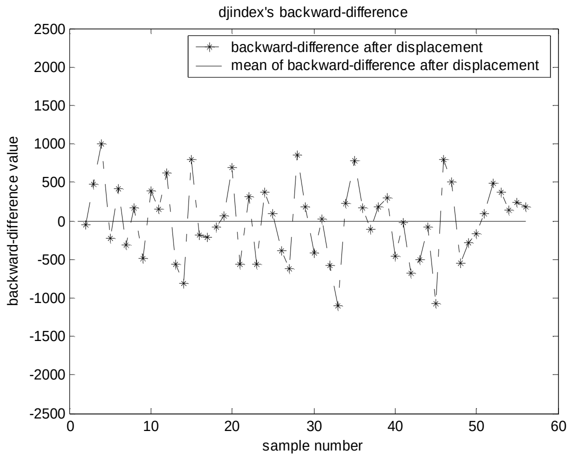

Figure 2. A sequence of backward-difference from New York D.J. Industry Index has displaced to its mean value dated from January 1999 to August 2003 for a period of 56 months. This plot showing the volatility clustering as big changes happened around sample number 2-4, 12-16, 27-35, and 45-47 as well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

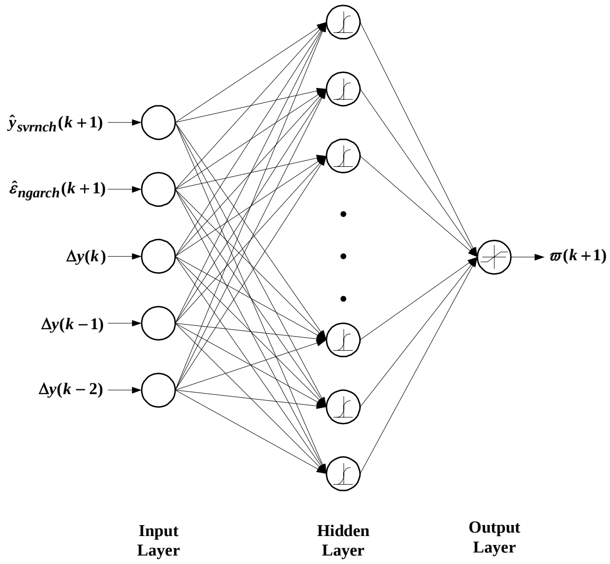

Figure 3. A structure of 5×16×1 BPNN is used to compute weight ϖ during training phase. Activation function in neurons is set by log-sigmoid function in hidden layer, and symmetric saturating linear function in output layer.

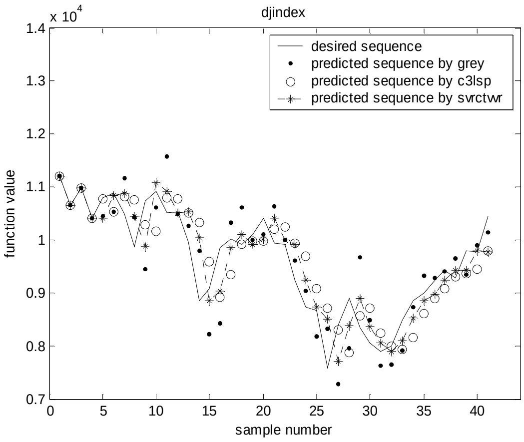

Figure 4. Forecasts of N. Y. -D. J. Indus. monthly index for 41 months from Aug. 2000 to Dec. 2003.

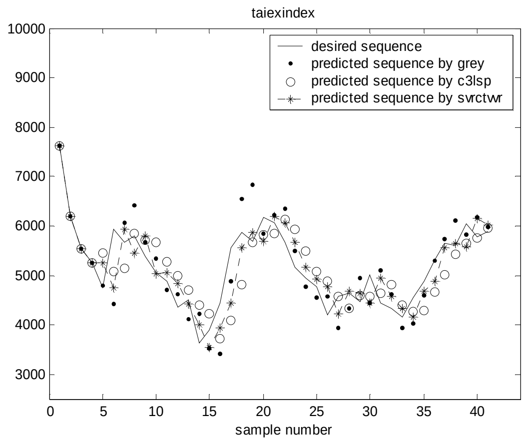

Figure 5. Forecasts of Taiwan TAIEX monthly index for 41 months from Aug. 2000 to Dec. 2003.

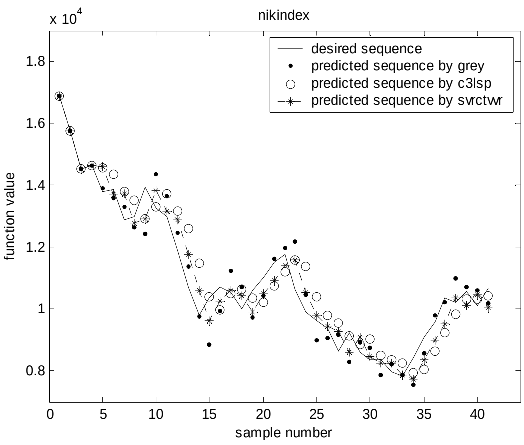

Figure 6. Forecasts of Japan Nikkei monthly index for 41 months from Aug. 2000 to Dec. 2003.

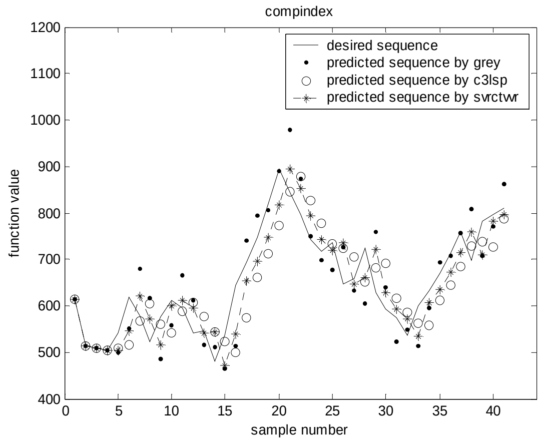

Figure 7. Forecasts of Korea Comp. monthly index for 41 months from Aug. 2000 to Dec. 2003.

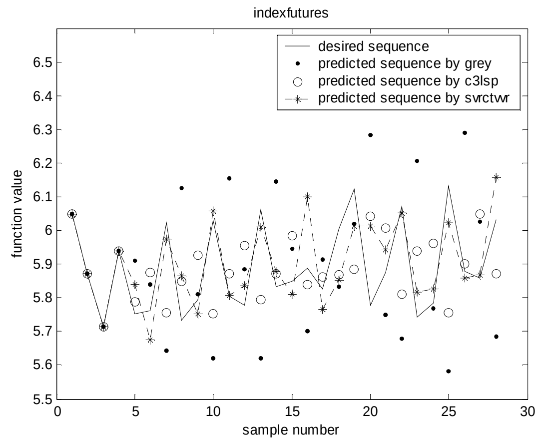

Figure 8. Forecasts of equity products on futures monthly index for 24 months from Jan. 2001 to Dec. 2002.

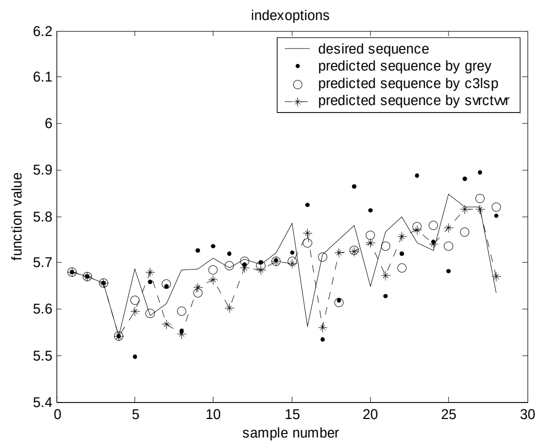

Figure 9. Forecasts of equity products on options monthly index for 24 months from Jan. 2001 to Dec. 2002.