American. Jr. of Mathematics and Sciences
Vol. 1, No.1 ,(January 2012)

www.journalshub.com

Neural Network based Support vector machines and Genetic algorithm for predicting trading signals of stock indices

Lect J.Emmanuel Robin MCA[1],V.Abimaniya B.Tech(final year)[2],
[1]Department of Computer applications [2]Department of Information Technology/ [1,2] Jayaram College of engineering & technology, Trichy, India
Email: [1]robin_ej@rediffmail.com [2]abimaniya4001@gmail.com

Abstract

The aim of this paper is to present neural network algorithms along with the genetic algorithm to predict whether it is best to buy, hold, or sell shares and trading signals of stock market indices. Neural network algorithms are based on the structure of feed forward neural networks and an Ordinary Least Squares OLSs error function. An adjustment relating to the contribution from the historical data used for training the networks and penalization of incorrectly classified trading signals were accounted for, when modifying the OLS function. A global optimization algorithm was employed to train these networks. In addition, an evolving least squares support vector machine (LSSVM) learning paradigm is used to explore stock market trends. In this proposed learning paradigm, a genetic algorithm (GA), one of the most popular evolutionary algorithms (EAs), is first used to select input features for LSSVM learning, i.e., evolution of input features. Then another GA is used for parameters optimization of LSSVM, i.e., evolution of algorithmic parameters. Finally, the evolving LSSVM learning paradigm with best feature subset, optimal parameters along with the genetic algorithm is used to predict stock market movement direction in terms of historical data series.

Index Terms— ***Least squares support vector machine, evolutionary algorithms, genetic algorithm, feature selection, parameter optimization, genetic algorithm, statistical models***

I. INTRODUCTION

In this paper, we used modified neural network algorithms for forecasting the trading signals of stock market indices. We used the standard FNN algorithm as the basis of these modified algorithms and later the proposed LSSVM paradigm is carried out.

A. Neural network algorithm

A standard FNN is a fully connected network with every node in the lower layer linked to every node in the next higher layer. These linkages are attached with some weights, $w = (w_1 ..., w_M)$, where M is the number of all possible linkages. Given weight, w, the network produces an output for each input vector. The output corresponding to the ith input vector will be denoted by $o_i \equiv o_i(w)$. FNNs adopt the backpropagation learning that finds optimal weights w by minimizing an error between the network outputs and given targets. The most commonly used error function is the Ordinary Least Squares function (OLS):

$$E_{\mathrm{OLS}} = \frac{1}{N}\sum_{i=1}^{N}(a_i - o_i)^2,$$

where N is the total number of observations in the training set, while a_i and o_i are the target and the output corresponding to the ith observation in the training set.

B. Modified neural network algorithm

Modifications to neural network algorithms were done by (i) using the OLS error function as well as the modified least squares error functions; (ii) employing a global optimization algorithm to train the networks. In this paper, we applied the global optimization algorithm, AGOP, for training the proposed network algorithms. As the error function to be minimised, we considered E_{OLS} and E_{DLS} together with the two modified error functions E_{CC} and E_{TCC} Based on these four error functions, we proposed the following algorithms:

(i)NN_{OLS}—neural network algorithm based on the Ordinary Least Squares error function, E_{OLS}

$$E_{OLS} = \frac{1}{N}\sum_{i=1}^{N}(a_i - o_i)^2,$$

(ii) NN_{DLS}—neural network algorithm based on the Discounted Least Squares error function, E_{DLS}

$$E_{DLS} = \frac{1}{N}\sum_{i=1}^{N} w_b(i)(a_i - o_i)^2,$$

(iii) NN_{CC}—neural network algorithm based on the newly proposed error function 1, E_{CC}

$$E_{CC} = \frac{1}{N}\sum_{i=1}^{N} w_d(i)(a_i - o_i)^2.$$

(iv) NN_{TCC}—neural network algorithm based on the newly proposed error function 2, E_{TCC}

$$E_{TCC} = \frac{1}{N}\sum_{i=1}^{N} w_b(i) \times w_d(i)(a_i - o_i)^2,$$

The layers are connected in the same structure as the FNN. A tan-sigmoid function was used as the transfer function between the input layer and the hidden layer, while the linear transformation function was employed between the hidden and the output layers. Algorithm NN_{OLS} differs from the standard FNN algorithm since it employs a new global optimization algorithm for training. Similarly, NN_{DLS} also differs from the respective algorithm used in due to the same reason. In addition to the use of new training algorithm, NN_{CC} and NN_{TCC} are based on two different modified error functions. The only way to examine whether these new modified neural network algorithms perform better than the existing ones is to conduct numerical experiments.

II. EVOLVING LSSVM LEARNING PARADIGM

In this section, the proposed evolving LSSVM learning paradigm is described in detail. First of all, a general framework of the evolving LSSVM learning paradigm is presented. Then each component of the proposed evolving LSSVM learning paradigm is described in detail.

A. General framework of evolving LSSVM learning paradigm

On the one hand, when the input space dimensions, i.e., input features, are rather large, the interpretability of the LSSVM-based predictive model will be poor. It is therefore necessary for LSSVM to preprocess the input features. On the other hand, LSSVM generalization ability is often controlled by kernel type, kernel parameters and upper bound parameter. Although this study uses a mixed kernel function to overcome the influence of kernel types, the choice of many parameters, such as convex combination coefficients (λ_1, λ_2, and λ_3), kernel parameters (d, σ, ρ, and θ) and upper bound parameter C, depends in many aspects on the art of the researchers. For these two problems, evolutionary algorithm (EA) is used. Particularly, for the first problem, a standard genetic algorithm (GA) is used for input feature selection to increase the model interpretability and avoid "curse of dimension". For the second problem, another GA is used to optimize the parameters of LSSVM to improve the generalization ability.

GA searches the exponential space of feature variable subsets and passes one subset of features to a LSSVM model. the GA biases its search direction to optimize the evaluation objective. GA is used to optimize the eight undetermined parameters, λ_1, λ_2, λ_3, d, σ, ρ, θ, and C, as listed above. In this component, the eight undetermined parameters are first defined as a chromosome. The fitness of each chromosome is determined by the GA fitness function, i.e., predictive performance of LSSVM using the chromosome as its parameters. The chromosomes are processed by several evolution procedures, i.e., crossover, mutation and selection, to produce the optimal solution.

B. GA-based input features evolution

To date, GA, the most popular type of evolutionary algorithm (EA), has become an important stochastic optimization method as they often succeed in finding the best optimum in contrast to most common optimization algorithms. Our goal is to find a small subset of input variables from many candidate variables, the evaluation of the fitness starts with the encoding of the chromosomes into LSSVM model whereby "1" indicates that a specific variable is selected and "0" that a variable is not selected by the LSSVM modelFinally, these two evaluation criteria are combined into one so-called fitness function f and used to evaluate the quality of each string. For a classification problem, for example, our fitness function for the GA variable selection can use the following form:

$$f = E_{accuracy} - \alpha E_{complexity}$$

where $E_{accuracy}$ is the classification accuracy or hit ratio, representing the model predictive power, $E_{complexity}$ is the model complexity, α is a parameter. Parameter α is aimed at adjusting the number of variables used by the evolved LSSVM in

terms of users' preference. Usually, a high value of the fitness function results in only few variables selected for each LSSVM model whereas a small value of fitness function results in more variables being selected. We expect that lower complexity will lead to easier interpretability of solutions as well as better generalization.
Usually, model accuracy of classification problems can be represented by hit ratio with the following form:

$$E_{accuracy} = \frac{\text{The number of correct classifica tion}}{\text{The number of all evaluation sample}}$$

the aim of the first part is to favor feature variable subsets with higher discriminative power to classification problems, while the second part is aimed at finding parsimonious solutions by minimizing the number of selected features as follows.

$$E_{complexity} = \frac{n_v}{N_{tot}}$$

where n_v is the number of variables used by the LSSVM models, N_{tot} is the total number of variables.

C. GA-based parameter evolution

As earlier noted, the LSSVM generalization ability is often affected by kernel types, kernel parameters and upper bound parameters. As a result it brings three additional coefficients, λ_1, λ_2, and λ_3. These three coefficients along with kernel parameters (d, σ, ρ, θ) and upper bound parameter C constitute the objects of evolution because the choice of these parameters depends heavily on the experience of researchers. Therefore a general representation of parameters using in a LSSVM training process are describe as a vector as follows:

$$\Theta = (\lambda_1, \lambda_2, \lambda_3, d, \sigma, \rho, \theta, C)$$

To perform parameter search process using GA, the model generalization performance is used as fitness function. That is, the average model performance is used as the fitness function. For a regression or classification problem, we define the fitness function f by the following steps.

1) Randomly split the samples data into $D_{training}$ and $D_{validation}$ using k-fold cross validation technique
2) Use $D_{training}$ to train the LSSVM model with the parameters P and obtain a predictor or classifier
3) Use the predictor or classifier to predict or classify the samples in $D_{validation}$
4) Compute the average prediction or classification performance of the k-fold cross validation, $\bar{E}_{accuracy}$
5) The value of the fitness function f is model generalization performance, it is the average value, i.e., $f(\Theta) = \bar{E}_{accuracy}$.

Now the goal is to maximize the fitness function $f(\Theta)$. Together with the previous parameter constraints, we can formulate the following optimization problem:

$$\begin{cases} \max & f(\Theta) = \overline{E}_{accuracy} \\ \text{s.t.} & \lambda_1 + \lambda_2 + \lambda_3 = 1 \\ & 0 \le \lambda_1, \lambda_2, \lambda_3 \le 1 \\ & \lambda_1, \lambda_2, \lambda_3, d, \sigma, \rho, \theta, C \ge 0 \end{cases}$$

where Θ is a vector representing the LSSVM parameters. The GA is used to solve the above optimization problem. First of all, the parameter vector Θ □ (λ_1 , λ_2 , λ_3 , d,σ, ρ,θ,C) of LSSVM is defined as a chromosome. The detailed evolutionary procedure can be written as following:

1) Perform the basic genetic operations: select parents based on their fitness values, produce children from the parents by crossover and mutation. Replace the current chromosome with the children to formulate a new chromosome;
2) Repeat the second and third steps until the stop criteria are met.
3) Report the best chromosome with the largest fitness value as the optimal solution.This best chromosome corresponds to the optimal prameters by GA process. Return these parameters.

III. COMPARISON RESULTS

First of all, the differences between the different models are very significant. For example, for the S&P 500 testing case, the hit ratio for the ARIMA model is 55.78%, for the OLS and FFNN model the hit ratios are 61.43% and 67.56%, respectively, and for the standard SVM model it is only 72.61%; while for the proposed evolving LSSVM forecasting model, the hit ratio reaches 82.66%, which is significantly higher than the individual SVM, FFNN and other statistical models, implying that the proposed evolving LSSVM learning has a significant improvement on SVM model in mining and exploring stock market trend.

IV. CONCLUSIONS

In this study, the modified neural network algorithm with an evolving LSSVM model integrating LSSVM and evolutionary algorithms (EAs) is proposed to predict stock market tendency. In the proposed evolving LSSVM learning paradigm, a standard genetic algorithm (GA) is first used to select possible input feature combination and optimize parameters of LSSVM and then the evolved LSSVM is used to predict the stock market trend. There are two distinct strengths for the evolving LSSVM model. The ability is to build an optimal prediction model because all model parameters are optimized. These indicate that the proposed evolving LSSVM model can be used as a viable alternative solution to stock market trend mining and exploration. It is worth noting, however, that the proposed evolving LSSVM learning paradigm could be further improved in the future, such as in areas of ensemble learning and ensemble evolution with LSSVM. Furthermore, this proposed method can not only work in the stock market mining but also work in general classification/regression problems. Future studies will look into these important issues.

ACKNOWLEDGMENT

This research was supported by Mr.A.Venkata Subramanian, Head of the department of Information Technology, Jayaram College of engineering and technology.

REFERENCES

[1] A.S. Chen and M.T. Leung, "Regression neural network for error correction in foreign exchange forecasting and trading", *Computers & Operations Research*, vol. 31, no. 7, pp. 1049-1068, 2004.

[2] W. Huang, Y. Nakamori and S.Y. Wang, "Forecasting stock market movement direction with support vector machine", *Computers & Operations Research*, vol. 32, no.10, pp. 2513-2522, 2005.

[3] Lean Yu, Huanhuan Chen, Shouyang Wang,Kin Keung Lai, "Evolving Least Squares Support Vector Machines for Stock Market Trend Mining".

[4] Tilakaratne, M. A. Mammadov, and S. A. Morris, "Modified Neural Network Algorithms for Predicting Trading Signals of Stock Market Indices"

[5] B. Egeli, M. Ozturan, and B. Badur, "Stock market prediction using artificial neural networks," in Proceedings of the 3rd Hawaii International Conference on Business, pp. 1–8, Honolulu, Hawaii, USA, June 2003.

[6] . Genc¸ay and T. Stengos, "Moving average rules, volume and the predictability of security returns with feedforward networks," *Journal of Forecasting*, vol. 17, no. 5-6, pp. 401–414, 1998.