World Academy of Science, Engineering and Technology
International Journal of Economics and Management Engineering
Vol:5, No:6, 2011

Support Vector Fuzzy Based Neural Networks For Exchange Rate Modeling

Prof. Chokri SLIM
ISCAE, Manouba University
2030 Manouba
Tunisia
e-mail : chokri.slim@iscae.rnu.tn

Abstract**—**A Novel fuzzy neural network combining with support vector learning mechanism called support-vector-based fuzzy neural networks (SVBFNN) is proposed. The SVBFNN combine the capability of minimizing the empirical risk (training error) and expected risk (testing error) of support vector learning in high dimensional data spaces and the efficient human-like reasoning of FNN.

Keywords**—** Neural network; Fuzzy inference; Machine learning; Fuzzy modeling and rule extraction; Support Vector Regression

I. Introduction

In modeling some systems where available information is uncertain, we must deal with a fuzzy structure of the system considered. This structure is represented as a fuzzy function whose parameters are given by fuzzy sets. The fuzzy functions are defined by Zadeh's extension principle [1], [2], [3].

The Support Vector Machines (SVMs), developed at AT&T Bell Laboratories by Vapnik and co-works [4], [5], have been very successful in pattern classification and function estimation problems for crisp data. It is based on the idea of structural risk minimization, which shows that the generalization error is bounded by the sum of the training error and a term depending on the Vapnik–Chervonenkis (VC) dimension. By minimizing this bound, high generalization performance can be achieved. A comprehensive tutorial on SVM classifier has been published by Burges [6]. In regression and time series prediction applications, excellent performances were also obtained in. Lin et al. [7] and Huang et al. [8] first introduced the use of fuzzy set theory for SVM classification problems.

Using hybrid models or combining several models has become a common practice to improve forecasting accuracy.The literature on this topic has expanded dramatically since the early work of Reid [9]. The basic idea of the model combination in forecasting is to use each model's unique feature to capture different patterns in the data. Both theoretical and empirical findings suggest that combining different methods can be an effective and efficient way to improve forecasts [11]. Slim [11] and [12] proposed an hybrid approach based on neural network and fuzzy system for financial time series forecasting and bankruptcy prediction to use the advantages and to fulfill the limitations of the two systems.

In this paper, we incorporated the concept of fuzzy set theory into the SVM regression and Neural ntworks modeling. A novel fuzzy neural network combining with support vector learning mechanism called support-vector-based fuzzy neural networks (SVBFNN) is proposed. The SVBFNN combine the capability of minimizing the empirical risk (training error) and expected risk (testing error) of support vector learning in high dimensional data spaces and the efficient human-like reasoning of FNN.

A learning algorithm consisting of three learning phases is developed to construct the SVBFNN and train the parameters. In the first phase, the fuzzy rules and membership functions are automatically determined by the clustering principle. In the second phase, the parameters of FNN are calculated by the SVR with the proposed adaptive fuzzy kernel function for time series prediction. In the third phase, the relevant fuzzy rules are selected by the proposed fuzzy rule reduction method.

The paper is organized as follows: the structure and the learning algorithme behavior of the proposed SVBFNN are described in Section 2. The proposed model is used to predict currency exchange rate between EURO/TND, USD/TND, GBP/TND and JPY/TND. and is compared with some parametric models such as ARIMA, VAR and nonparametric techniques such as SVR and BNN in Section 3. Conclusion is summarized in the last section.

II. Methodology : Structure of the SVBFNN and the Learning Algorithm

The proposed SVBFNN is a four-layered FNN that is comprised of the input, membership function, fuzzy rules, and the output layer.

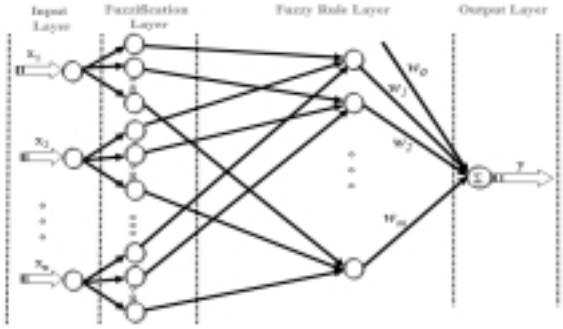

Fig. 1. Structure of the SVBFNN

- Layer 1 accepts input variables, whose nodes represent input linguistic variables. No computation is done in this layer. Each node in this layer, which corresponds to one input variable, only transmits input values to the next layer directly. That is :

$$y^1 = x_i \tag{1}$$

where x_i, $i = 1, 2, \ldots, n$ are the input variables of the network

- Layer 2 is to calculate the membership values, whose nodes represent the terms of the respective linguistic variables. In other words, the membership value which specifies the degree to which an input value belongs to a fuzzy set is calculated in Layer 2:

$$y^2 = e^{-\frac{(x_i - \mu_{ij})^2}{\sigma_{ij}^2}} \tag{2}$$

where $e(.)$ is a gaussian membership function

- Nodes at Layer 3 represent fuzzy rules. The links before Layer 3 represent the preconditions of fuzzy rules, and the links after Layer 3 represent the consequences of fuzzy rules

Here we use the AND operation for each Layer 2 node :

$$y^3 = e^{-[\ _j(\ -\boldsymbol{\mu}_j)^T][\ _j(\ -\boldsymbol{\mu}_j)]} \tag{3}$$

where $_j$=diag[$\frac{1}{\sigma_{1j}}, \ldots, \frac{1}{\sigma_{nj}}$], $\mu_j = [\mu_{1j}, \mu_{2j}, \ldots, \mu_{nj}]^T$, $= [x_1, x_2, x_3, \ldots, x_n]^T$ is the input vector of the network

- Layer 4 is the output layer : The single node y^4 in this layer is labeled with , which computes the overall output and can be computed as:

$$y^4 = \sum_{j=1} w_j\ y^3 + w_0 = \sum_{j=1} w_j\ e^{-[\ _j(\ -\boldsymbol{\mu}_j)^T][\ _j(\ -\boldsymbol{\mu}_j)]} + w_0 \tag{4}$$

The proposed learning algorithm of SVBFNN consists of three phases :

In the first phase, the initial fuzzy rule and membership of network structure are automatically established based on the fuzzy clustering method. In the second phase, the means of membership functions and the connecting weights between layer 3 and layer 4 of SVBFNN are optimized by using the result of the support vector learning method with the fuzzy kernels for time series prediction. In the third phase, unnecessary fuzzy rules are recognized and eliminated and the relevant fuzzy rules are determined.

First the input datasets are partitioned. For each incoming pattern $b = [x, y]^T$ the strength a rule is fired. We can use the firing strength as this degree measure :

$$F^j(b) = e^{-[\ (\ -\boldsymbol{\mu}_j)^T][\ _j(\ -\boldsymbol{\mu}_j)]} \in [0\ 1] \tag{5}$$

We can obtain the following criterion for the generation of a new fuzzy rule :

$$H = \max_{1 \le j \le c(t)} F^j(b) \tag{6}$$

where $c(t)$ is the number of existing rules at time t.

The optimal parameters of SVBFNN are trained by using the ξ insensitivity loss function SVR based on the fuzzy kernels.

The dual quadratic optimization of SVR is solved in order to obtain an optimal hyperplane for any linear or nonlinea space:

$$\max\ \mathcal{L}(a, a^*) = -\xi \sum_{i=1} (a_i^* + a_i) + \sum_{i=1} (a_i^* + a_i) y_i - \frac{1}{2} \sum_{i,j=1} (a_i^* - a_i)(a_j^* - a_j)\ K(\ _i, \ _j) \tag{7}$$

subject to contraints :

$$\sum_{i=1} a_i^* = \sum_{i=1} a_i;\ \ 0 \le a_i^* \le C; \qquad 0 \le a_i \le C;\quad i = 1, 2, ..., q \tag{8}$$

where K($_i$ $_j$) is the fuzzy kernel, ξ is a previously chosen nonnegative number for insensitive loss function, and C is a user-specified positive parameter to control the tradeoff between complexity of SVR and the number of nonseparable points.

A solution = $(a_1, a_2, ..., a_{ns})$ and * = $(a_1^*, a_2^*, ..., a_{ns}^*)$ can be obtained, where a_i and a_i^* are Lagrange coefficients

The corresponding support vectors =$[s_1, s_2, ..., s_i, ..., s_{ns}]$ can be obtained, and the constant (threshold) w_0 in (4) is :

$$w_0 = \frac{1}{q}\left(\sum_{i=1} (y_i - x_i^T \theta_0)\right) \text{ with } \theta_0 = \sum_{i=1} (a_i^* - a_i)\ x_i \tag{9}$$

The coefficients w_j in (4) can be calculated by :

$$w_j = y_j(a_j^* - a_j). \tag{10}$$

In this phase, the number of fuzzy rules learning in Phases 1 and 2 is reduced by removing some irrelevant fuzzy rules and the consequent parameters

The method reduces the number of fuzzy rules by minimizing the distance measure between original fuzzy rules and reduced fuzzy rules

To achieve this goal, we rewrite (4) as :

$$y^4 = \sum_{j=1} w_j\ y^3 + w_0 = \sum_{j=1} w_j \prod_{i=1} e^{-\frac{(x_i - \mu_{ik})^2}{\sigma_{ik}^2}} + w_0 \tag{11}$$

The reduced set is given by :

$$y^{re(4)} = \sum_{k=1} \gamma_k \prod_{i=1} e^{-\frac{(x_i - \mu_{ij}^{(r)})^2}{\sigma_{ij}^{2(r)}}} + w_0 \tag{12}$$

γ_k is the consequent parameters of the remaining fuzzy rules.

The whole learning scheme is iterated until the new rules are sufficiently sparse.

III. Empirical Study : Data and Results

Daily values of exchange rates for EURO/TND, USD/TND, GBP/TND and JPY/TND were used from Januray 02, 2000 to july 20, 2009. The data set was randomly divided into three groups, training, cross validation, and testing set.

First, we conduct a time series analysis to determine the number of inputs by using ARIMA method, VAR for EURO/TND, USD/TND, GBP/TND and JPY/TND rates. Second, the SVBFNN and SVM and BNN are applied. The performances of these models are compared in terms of mean square error (MSE), mean absolute error (MAE), Normalized mean square error NMSE,.Mean Absolute Percentage Error (MAPE) and Correlation between actual and predicted (CBAP).

The ADF unit root test revealed that EURO/TND, USD/TND, GBP/TND and JPY/TND time series are not stationary. Based on autocorrelation and partial autocorrelation, the following models are selected for each exchange rate datasets :

Table 1: Selected Models

Exchange rates	ARIMA models
EURO/TND	ARIMA(1,1,0)
USD/TND	ARIMA(2,1,0)
GBP/TND	ARIMA(1,1,0)
JPY/TND	ARIMA(2,1,0)

We found that time series are $I(1)$ by using Augmented Dickey Fuller(ADF) test. We chose the Lag length as k =4 using AIC A VAR model was assumed and the parameters of the model were estimated.

$$\begin{aligned} x_t &= f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}) \\ y_t &= f(y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}, x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}) \\ z_t &= f(z_{t-1}, z_{t-2}, z_{t-3}, z_{t-4}, x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}) \\ v_t &= f(v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}) \end{aligned} \quad (13)$$

x_t =(Euro/TND), y_t =(USD/TND), z_t =(GBP/TND), v_t = (JPY/TND)

Next, we will use this information on SVBFNN and SVM and BNN methods. We have used the same experimental design. An BNN network with one hidden layer is used for each series. The number of hidden units and other parameters are determined by using a cross-validation data.

Table 2 shows the MSE, MAE, NMSE,.MAPE and CBAP. of ARIMA and VAR selection input for SVBFNN and SVM and BNN methods.

Table 2: Criteria performances of ARIMA and VAR input selection procedure for testing set

Exchange rate	Criteria	SVBFNN		SVM		BNN	
		ARIMA	VAR	ARIMA	VAR	ARIMA	VAR
EURO/TND	MSE	0.0000047	**0.0000013**	0.0000045	0.0000036	0.0000048	0.0000089
	MAE	**0.0015284**	**0.0007451**	0.0015397	0.0011958	0.0015411	0.0013473
	NMSE	**0.0008050**	**0.0002150**	0.0008120	0.0006080	0.0008160	0.0006680
	MAPE	**0.2350144**	**0.1133051**	0.2367995	0.1827616	0.2370115	0.2066587
	CBAP	**0.9995980**	**0.9998920**	0.9995970	0.9996960	0.9995920	0.9996570

We see that SVBFNN outperforms the SVM and BNN networks for ARIMA and VAR input selection methods.

IV. Conclusion

We try to combine parametric and nonparametric techniques in order to obtain a better performance for exchange rate forecasting. Comparison of three nonparametric models, SVBFNN, BNN and SVM, is given with two input selection techniques, ARIMA and VAR, respectively. Experiments showed that SVBFNN SVM methods outperforms the BNN networks for each input selection algorithm. This can be explained by the formulation of the SVBFNN, SVM and BNN networks. SVBFNN and SVM methods uses a quadratic programming problem which is convex and has a global optimum solution. BNN networks use the backpropagation algorithm to minimize the network error, the problem is nonconvex and it is hard to find the global optimum.

References

[1] L. A. Zadeh, “The concept of linguistic variable and its application to approximate reasoning,” Inform. Sci., vol. 8, pp. 199-249, 1975.
[2] H. Tanaka, S. Uejima, and K. Asai, “Linear regression analysis with fuzzy model,” IEEE. Trans. On Syst., Man, and Cyber., vol. 12, no. 6, pp. 903-907, 1982.
[3] H. Tanaka and H. Lee, “Interval regression analysis by quadratic programming approach,” IEEE Trans. on Fuzzy Systems, vol. 6, no. 4, pp. 473-481, 1998.
[4] Vapnik, The Nature of Statistical Learning Theory. Springer-Verlag, New York, 1995.
[5] C. Cortes, and V. N. Vapnik, “Support vector network,” Machine learning, vol. 20, pp. 1-25, 1995.
[6] C.J.C. Burges, “A tutorial on support vector machines for pattern recognition,” Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 955-974, 1998.
[7] C. -F. Lin and S. -D. Wang, “Fuzzy support vector machines,” IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 464-471, Mar. 2002.
[8] H.-P. Huang and Y.-H. Liu, “Fuzzy support vector machines for pattern recognition and data mining,” International Journal of Fuzzy Systems, vol. 4, no. 3, pp. 826-835, Sep. 2002.
[9] M.J. Reid, Combining three estimates of gross domestic product, Economica 35 (1968) 431–444.
[10] C Slim (2009) : Hybrid Approach in Neural Network design Applied to Financial Time Series Forecasting, The Journal of American Academy of Business Cambridge, Vol. 15, Num. 1,September 2009, 294-300.
[11] C. Slim (2007) : Fuzzy Neural Model for Bankruptcy Prediction, The Journal of Business Review Cambridge, Vol. 8, Num. 2, December 2007, 117-122.