Expert Systems with Applications 40 (2013) 850–853

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using multiplicative neuron model to establish fuzzy logic relationships

Cagdas Hakan Aladag *

Department of Statistics, Hacettepe University, Ankara 06800, Turkey

ARTICLE INFO

Keywords:
Forecasting accuracy
Fuzzy relations
Fuzzy time series
Multiplicative neuron model

ABSTRACT

Determination of fuzzy logic relationships between observations is quite effective on the forecasting performance of fuzzy time series approaches. In various studies available in the literature, it has been seen that utilizing artificial neural networks for establishing fuzzy relations increase the forecasting accuracy. In this study, a novel high order fuzzy time series forecasting approach in which multiplicative neuron model is used to define fuzzy relations is proposed in order to reach high forecasting level. Also, particle swarm optimization method is utilized to train multiplicative neuron model. In order to show forecasting performance of the proposed method, it is applied to a well-known data Taiwan future exchange and the results produced by the proposed approach is compared to those obtained from other fuzzy time series forecasting models. As a result of the implementation, it is observed that the proposed approach gives the best forecasts for Taiwan future exchange time series.

1. Introduction

In recent years, various fuzzy time series methods have been proposed in the literature. A fuzzy time series can be defined as a time series whose observations are fuzzy sets. Observations of most of real life time series can be expressed as fuzzy sets since these series contain uncertainty. Stock exchange time series can be given an example of such series. Fuzzy time series approaches, which have a wide application range, are generally composed of three main stages such as fuzzification, determination of fuzzy relationships and defuzzification. In the literature, some contributions have been made to these stages by various studies to increase forecasting accuracy of the method.

The description of fuzzy time series and its basic definitions were firstly introduced by Song and Chissom (1993a). In Song and Chissom (1993a), fuzzy time series is split into two subclasses which are time variant and time invariant. Solution methods for time variant and time invariant fuzzy time series were proposed by Song and Chissom (1993b) and Song and Chissom (1994), respectively.

In the literature, methods based on partition of universe of discourse, and fuzzy clustering techniques have been employed for fuzzification phase of fuzzy time series. Besides these methods, heuristic algorithms such as genetic algorithms (Lee, Wang, & Chen, 2007, 2008) and particle swarm optimization (Hsu et al., 2010) have been used in the fuzzification phase.

For the stage of determination of fuzzy relations, matrix operations based on fuzzy relations and fuzzy logic group relation tables have been frequently preferred. Also, artificial neural networks have been used as a good alternative method for this stage. Song and Chissom (1993b, 1994) utilized matrix operations based on fuzzy relations to establish fuzzy relations. Especially when the number of fuzzy sets is large, these matrix operations can be very complex and time consuming. Therefore, Chen (1996) proposed a new approach based on fuzzy group relation tables and using easier operations. However, in case of high order or multivariable models, it can be very hard to use fuzzy group relation tables. Hence, Huarng and Yu (2006) suggested employing feed forward neural networks for first order singe variable fuzzy time series models. Aladag, Basaran, Egrioglu, Yolcu, and Uslu (2009) used artificial neural networks to define fuzzy relations in high order singe variable fuzzy time series. Egrioglu, Uslu, Yolcu, Basaran, and Aladag (2009a) and Egrioglu, Aladag, Yolcu, Uslu, and Basaran (2009b) utilized artificial neural networks for high order multivariable fuzzy time series. In Aladag et al. (2009), Egrioglu, Uslu, et al. (2009a), Egrioglu, Aladag, et al. (2009b), and Huarng and Yu (2006) set indexes are used for both inputs and targets of neural network models. However, differently from these studies, Yu and Huarng (2010) use the membership degrees of previous and next observations for inputs and targets, respectively.

In the phase of defuzzification, there have been some methods such as centroid, feed forward neural networks and adaptive expectation methods. In the literature, the most preferred method is centroid method Aladag, Yolcu, and Egrioglu (2010).

* Tel.: +90 312 2977900.
E-mail address: aladag@hacettepe.edu.tr

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.05.039

In this study, a novel fuzzy time series forecasting approach in which multiplicative neuron model is used to define fuzzy relations is proposed in order to increase forecasting accuracy. In the proposed approach, a modified particle swarm optimization algorithm is also utilized to train multiplicative neuron model, which was firstly introduced in (Yadav, Kalra, & John, 2007). To show the applicability of the proposed method, it is applied to a well-known time series Taiwan future exchange (TAIFEX). For the aim of comparison, the data is also analyzed with some other methods available in the literature. As a result of the application, it is seen that the proposed approach has the best forecasting accuracy for TAIFEX data.

The next section presents basic definitions of fuzzy time series. The modified particle swarm optimization method is given in Section 3. Section 4 gives brief information about multiplicative neuron model. The proposed approach is introduced in Section 5. The implementation is given in Section 6. Finally, Section 7 concludes the paper.

2. Fuzzy time series

The definition of fuzzy time series was firstly introduced by Song and Chissom (1993a, 1993b, 1994). Basic definitions of fuzzy time series are given as follows (Song & Chissom, 1993a, 1993b, 1994):

Let U be the universe of discourse, where $U = \{u_1, u_2, ..., u_b\}$. A fuzzy set A_i of U is defined as $A_i = f_{A_i}(u_1)/u_1 + f_{A_i}(u_2)/u_2 + \cdots + f_{A_i}(u_b)/u_b$, where f_{A_i} is the membership function of the fuzzy set $A_i; f_{A_i} : U \to [0,1]$. u_a is a generic element of fuzzy set $A_i; f_{A_i}(u_a)$ is the degree of belongingness of u_a to $A_i; f_{A_i}(u_a) \in [0,1]$ and $1 \leqslant a \leqslant b$.

Definition 1. Let $Y(t)$ ($t = \ldots,0,1,2,\ldots$) a subset of real numbers, be the universe of discourse on which fuzzy sets $f_j(t)$ are defined. If $F(t)$ is a collection of $f_1(t), f_2(t),\ldots$ then $F(t)$ is called a fuzzy time series defined on $Y(t)$.

Definition 2. Fuzzy time series relationships assume that $F(t)$ is caused only by $F(t-1)$, then the relationship can be expressed as:

$$F(t-1) \to F(t)$$

This model is called as first order fuzzy time series forecasting model.

Definition 3. If $F(t)$ is a caused by $F(t-1), F(t-2),\ldots, F(t-m)$, then this fuzzy logical relationship is represented by

$$F(t-m), \cdots, F(t-2), F(t-1) \to F(t)$$

and it is called the mth order fuzzy time series forecasting model.

3. The modified particle swarm optimization

Particle swarm optimization, which is a population based heuristic algorithm, was firstly proposed by Kennedy and Eberhart (1995). This method is a useful tool for optimization problems Zhao and Yang (2009). Distinguishing feature of this heuristic algorithm is that it simultaneously examines different points in different regions of the solution space to find the global optimum solution. Local optimum traps can be avoided because of this feature. In this study, the modified particle swarm optimization method is used in the training process of multiplicative neuron model. The modified particle swarm optimization method algorithm has time varying inertia weight like in Shi and Eberhart (1999). In a similar way, this algorithm also has time varying acceleration coefficient like in Ma, Jiang, Hou, and Wang (2006). The algorithm of the method is presented below.

Algorithm 1: The modified particle swarm optimization

Step 1. Positions of each kth ($k = 1,2, \ldots, pn$) particles' positions are randomly determined and kept in a vector X_k given as follows:
$X_k = \{x_{k,1}, x_{k,2}, \ldots, x_{k,d}\}$, $k = 1,2, \ldots, pn$
where x_i^k ($i = 1,2,\ldots,d$) represents ith position of kth particle. pn and d represents the number of particles in a swarm and positions, respectively.

Step 2. Velocities are randomly determined and stored in a vector V_k given below.
$V_k = \{v_{k,1}, v_{k,2}, \ldots, v_{k,d}\}$, $k = 1,2,\ldots,pn$

Step 3. According to the evaluation function, *Pbest* and *Gbest* particles given in (1) and (2), respectively, are determined.

$$Pbest_k = (p_{k,1}, p_{k,2}, \ldots, p_{k,d}), \quad k = 1,2,\ldots,pn \tag{1}$$

$$Gbest = (p_{g,1}, p_{g,2}, \ldots, p_{g,d}) \tag{2}$$

where $Pbest_k$ is a vector stores the positions corresponding to the kth particle's best individual performance, and *Gbest* represents the best particle, which has the best evaluation function value, found so far.

Step 4. Let c_1 and c_2 represents cognitive and social coefficients, respectively, and w is the inertia parameter. Let (c_{1i}, c_{1f}), (c_{2i}, c_{2f}), and (w_1, w_2) be the intervals which includes possible values for c_1, c_2 and w, respectively. At each iteration, these parameters are calculated by using the formulas given in (3), (4) and (5).

$$c_1(c_{1f} - c_{1i})\frac{t}{\text{max}t} + c_{1i} \tag{3}$$

$$c_2(c_{2f} - c_{2i})\frac{t}{\text{max}t} + c_{2i} \tag{4}$$

$$w = (w_2 - w_1)\frac{\text{max}t - t}{\text{max}t} + w_1 \tag{5}$$

where *maxt* and t represent maximum iteration number and current iteration number, respectively.

Step 5. Values of velocities and positions are updated by using the formulas given in (6) and (7), respectively.

$$v_{i,d}^{t+1} = [W \times v_{i,d}^t + c_1 \times rand_1 \times (p_{id} - x_{i,d}) + c_2 \times rand_2 \times (p_{g,d} - x_{i,d})] \tag{6}$$

$$x_{i,d}^{t+1} = x_{i,d} + v_{i,d}^{t+1} \tag{7}$$

where $rand_1$ and $rand_2$ are random values from the interval [0 1].

Step 6. Steps 3–5 are repeated until a predetermined maximum iteration number (*maxt*) is reached.

4. Multiplicative neuron model

One type of artificial neural networks is multiplicative neuron model that was presented by Yadav et al. (2007) and it was shown that this model can produce better forecasts. The multiplicative neuron model has only one neuron and the general structure of the model, which includes only five inputs, is given in Fig. 1.

In the model given in Fig. 1, x_i ($i = 1,\ldots,5$) represents the inputs. While inputs are being summed in a feed forward neural network model, inputs are being multiplied in multiplicative neuron model. In other words, instead of using sum function, multiplication function is used in the multiplicative neuron model when input and output values are computed. The function $\Omega(x,\theta)$ is composed of multiplication of weighted inputs. f and y represents activation function and the output of the model, respectively. In the literature, different learning algorithms have been employed to train the