Applied Soft Computing 35 (2015) 151–160

Contents lists available at ScienceDirect

Applied Soft Computing

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

A clustering based forecasting algorithm for multivariable fuzzy time series using linear combinations of independent variables

S. Askari [a], N. Montazerin [a,*], M.H. Fazel Zarandi [b,1]

[a] Mechanical Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, Tehran 1591634311, Iran
[b] Industrial Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, Tehran 1591634311, Iran

ARTICLE INFO

Article history:
Received 30 January 2015
Received in revised form 1 June 2015
Accepted 11 June 2015
Available online 30 June 2015

Keywords:
Fuzzy time series
Fuzzy clustering
Fuzzy C-Means (FCM)
Least Square Estimate (LSE)
Forecasting

ABSTRACT

There are two popular types of forecasting algorithms for fuzzy time series (FTS). One is based on intervals of universal sets of independent variables and the other is based on fuzzy clustering algorithms. Clustering based FTS algorithms are preferred since role and optimal length of intervals are not clearly understood. Therefore data of each variable are individually clustered which requires higher computational time. Fuzzy Logical Relationships (FLRs) are used in existing FTS algorithms to relate input and output data. High number of clusters and FLRs are required to establish precise input/output relations which incur high computational time. This article presents a forecasting algorithm based on fuzzy clustering (CFTS) which clusters vectors of input data instead of clustering data of each variable separately and uses linear combinations of the input variables instead of the FLRs. The cluster centers handle fuzziness and ambiguity of the data and the linear parts allow the algorithm to learn more from the available information. It is shown that CFTS outperforms existing FTS algorithms with considerably lower testing error and running time.

1. Introduction

Fuzzy time series (FTS) is a universal forecasting method in a fuzzy environment [1–3]. FTS is used in various areas such as forecasting electricity load demand [4], stock exchange [5–10], rainfall and temperature forecasting [11], pollution [12], enrollments [13–15], etc. There are two major categories of FTS algorithms: FTS algorithms based on intervals of the universal set [3,16] and FTS algorithms based on fuzzy clustering [17–22]. The main problem with the interval based algorithms is the length of the intervals which is not clear how to be chosen. Many attempts are made to find optimal intervals but the problem is still unsolved [23–26]. Clustering based algorithms are preferred since they are interval independent.

A high-order multi-variable algorithm for FTS (HMV-FTS) was presented based on fuzzy clustering to improve forecasting accuracy and handle fuzzy time series with high order and multi-dimensional input space simultaneously [17]. HMV-FTS outperforms existing FTS algorithms as examined by various data sets of different contexts. Data of each variable of the FTS are clustered individually in HMV-FTS and other clustering based algorithms, which demands higher running time. The objective of the present work is to establish a fast and precise forecasting algorithm for FTS, based on fuzzy clustering and linear combinations of the input variables (CFTS). In contrast to the existing clustering based FTS algorithms, which cluster data of each variable separately, CFTS clusters the input data vectors in the clustering section of the algorithm.

This paper is organized as follows: mathematical framework of CFTS algorithm is discussed in Section 2. The algorithm is evaluated in Section 3. Computational cost of the algorithm is investigated in Section 4. CFTS is compared with recent FTS algorithms in Section 5 and concluding remarks are drawn in Section 6.

2. Mathematical formulation of CFTS algorithm

We introduce FTS briefly and then propose CFTS algorithm. Let $Y(t) \in \Re$, $t = 0, 1, 2, \ldots$ be the universe of discourse on which fuzzy sets $f_i(t)$, $i = 1, 2, \ldots$ are defined and $F(t)$ be a collection of $f_i(t)$s, then $F(t)$ is defined as a fuzzy time series on $Y(t)$. In general, $F(t)$ is a linguistic variable with linguistic values, $f_i(t)$. If $F(t)$ is related to $F(t-1)$, the Fuzzy Logical Relationship (FLR) between them is represented by $F(t-1) \rightarrow F(t)$ which is a first order FLR. In this FLR, $F(t-1)$ and $F(t)$ are called current state and next state and denoted by A_i and A_j, respectively, and their FLR is shown as $A_i \rightarrow A_j$. FLRs

* Corresponding author. Tel.: +98 216454 3415; fax: +98 21 6454 3415.
E-mail addresses: s_askari@aut.ac.ir (S. Askari), mntzrn@aut.ac.ir, mntzrn@hotmail.com (N. Montazerin), arandi@aut.ac.ir (M.H.F. Zarandi).
1 Tel.: +98 2164545378.

http://dx.doi.org/10.1016/j.asoc.2015.06.028
1568-4946/

with the same current states are grouped as a Fuzzy Logical Relationship Group (FLRG). For forecasting, current state of the FLR of the forecast time is constructed and then the FLRG with current state identical with that of the forecast FLR is found. Next state of the forecast FLR is taken the same as the next state of this FLRG. Finally, crisp value of the forecast is computed from defuzzification of the fuzzy value(s) of the forecast obtained from the next state of the forecast FLR [2,16].

In the CFTS algorithm, the input data are clustered as in the clustering based FTS algorithms but no FLR is used. Instead of FLRs, CFTS uses combinations of input variables to map the input data into the output space. For high order FTS, one can simply apply CFTS algorithm on the lagged variables of the FTS to forecast future values of the dependent variable.

Consider $X_{r\times N}$ and $\vec{y}_{1\times N}$ as the input and output data of the FTS where N is number of observations and r is number of the FTS variables. jth input data vector is $\vec{x}_j = [\,x_{1j} \quad x_{2j} \quad \ldots \quad x_{rj}\,]^T$ and its corresponding output is y_j. We group X matrix into c clusters using Fuzzy C-Means algorithm, FCM [27]. For this purpose, following index is minimized with the given constraint [27]:

$$J_1 = \sum_{j=1}^{N}\sum_{i=1}^{c} u_{ij}^{m} \|\vec{x}_j - \vec{v}_i\|_A^2, \quad \sum_{i=1}^{c} u_{ij} - 1 = 0 \tag{1}$$

where c is the number of clusters, $\vec{v}_i$ is center of the ith cluster (ith row of cluster centers matrix, $V_{r\times c}$), u_{ij} is membership grade of the jth data vector in the ith cluster (element of partition matrix, $U_{c\times N}$), $\|\vec{x}_j - \vec{v}_i\|_A^2 = (\vec{x}_j - \vec{v}_i)^T A(\vec{x}_j - \vec{v}_i)$ is distance, $m \in (1, \infty]$ is degree of fuzziness and $A_{r\times r}$ is the covariance norm matrix, defined as:

$$A = \left(\frac{1}{N}\sum_{j=1}^{N}(\vec{x}_j - \bar{\vec{v}})(\vec{x}_j - \bar{\vec{v}})^T\right)^{-1}, \quad \bar{\vec{v}} = \frac{1}{N}\sum_{j=1}^{N}\vec{x}_j \tag{2}$$

Since A is a symmetric matrix, $A = A^T$. Using Lagrange Multipliers Method (LMM), J_1 is written as:

$$J_1^* = \sum_{j=1}^{N}\sum_{i=1}^{c} {u_{ij}}^{m} \|\vec{x}_j - \vec{v}_i\|_A^2 + \sum_{j=1}^{N}\lambda_j\left(\sum_{i=1}^{c} u_{ij} - 1\right)$$

Zeroing derivatives of J_1^* with respect to $\vec{v}_i$, u_{ij} and λ_j yields:

$$\frac{\partial J_1^*}{\partial \vec{v}_i} = \sum_{j=1}^{N} {u_{ij}}^{m}(A + A^T)(\vec{x}_j - \vec{v}_i) = 0, A = A^T \Rightarrow \vec{v}_i = \frac{\sum_{j=1}^{N} u_{ij}^m \vec{x}_j}{\sum_{j=1}^{N} u_{ij}^m}$$

$$\frac{\partial J_1^*}{\partial u_{ij}} = m u_{ij}^{m-1}\|\vec{x}_j - \vec{v}_i\|_A^2 + \lambda_j = 0, \frac{\partial J_1^*}{\partial \lambda_j} = \sum_{i=1}^{c} u_{ij} - 1 = 0 \Rightarrow u_{ij} = \left[\sum_{k=1}^{c}\left(\frac{\|\vec{x}_j - \vec{v}_i\|_A^2}{\|\vec{x}_j - \vec{v}_k\|_A^2}\right)^{\frac{1}{m-1}}\right]^{-1}$$

Therefore,

$$\vec{v}_i = \frac{\sum_{j=1}^{N} u_{ij}^m \vec{x}_j}{\sum_{j=1}^{N} u_{ij}^m}, u_{ij} = \left[\sum_{k=1}^{c}\left(\frac{\|\vec{x}_j - \vec{v}_i\|_A^2}{\|\vec{x}_j - \vec{v}_k\|_A^2}\right)^{\frac{1}{m-1}}\right]^{-1} \tag{3}$$

These equations are repeatedly updated until changes in U and V become negligible. After deciding on cluster centers, Membership Function (MF) of the qth variable in the ith cluster is computed as:

$$u_{qij} = \left[\sum_{k=1}^{c}\left(\frac{\|x_{qj} - v_{qi}\|^2}{\|x_{qj} - v_{qk}\|^2}\right)^{\frac{1}{m-1}}\right]^{-1}, \quad \|x_{qj} - v_{qi}\|^2 = (x_{qj} - v_{qi})^2 \quad \forall j \in [1, N] \tag{4}$$

Weighted contribution of each cluster in the calculation of the output associated with $\vec{x}_j$ is computed as:

$$\tau_{ij} = \frac{\prod_{q=1}^{r} u_{qij}}{\sum_{i=1}^{c}\prod_{q=1}^{r} u_{qij}} \tag{5}$$

Consider the matrix $X^*_{(r+1)\times N}$ such that: $x^*_{1j} = 1, x^*_{(q+1)j} = x_{qj} \quad \forall q \in [1, r], j \in [1, N]$. Then output of CFTS algorithm for $\vec{x}_j^* = [\,1 \quad \vec{x}_j\,]$ is considered as the weighted linear combinations of the input variables.

$$y_j = \sum_{i=1}^{c}\tau_{ij}\sum_{q=1}^{r+1} p_{iq}x^*_{qj} \tag{6}$$

The coefficients p_{iq} are obtained by minimizing the following index:

$$J_2 = \left(\sum_{i=1}^{c}\tau_{ij}\sum_{q=1}^{r+1} p_{iq}x^*_{qj} - y_j\right)^2 \tag{7}$$

Zeroing derivative of J_2 with respect to p_{iq} yields:

$$\begin{aligned} &\frac{\partial J_2}{\partial p_{iq}} = \left(\sum_{i=1}^{c}\tau_{ij}\sum_{q=1}^{r+1} p_{iq}x^*_{qj} - y_j\right)\tau_{ij}x^*_{qj} = 0 \Rightarrow \\ &\sum_{i=1}^{c}\tau_{ij}\sum_{q=1}^{r+1} p_{iq}x^*_{qj} - y_j = 0 \quad \forall j \in [1, N] \end{aligned} \tag{8}$$

Using $x^*_{1j} = 1, x^*_{(q+1)j} = x_{qj}\, \forall q \in [1, r], j \in [1, N]$, (8) is written as the following set of N equations:

$$H\vec{P} = \vec{y} \therefore \vec{P} = [\,\vec{p}_1 \quad \vec{p}_2 \quad \cdots \quad \vec{p}_i \quad \cdots \quad \vec{p}_c\,]^T, \vec{p}_i = [\,p_{i1} \quad p_{i2} \quad \cdots \quad p_{iq} \quad \cdots \quad p_{i(r+1)}\,]$$

$$H = \begin{bmatrix} \tau_{11} & \ldots & \tau_{c1} & \tau_{11}x_{11} & \ldots & \tau_{c1}x_{11} & \ldots & \tau_{11}x_{r1} & \ldots & \tau_{c1}x_{r1} \\ \tau_{12} & \ldots & \tau_{c2} & \tau_{12}x_{12} & \ldots & \tau_{c2}x_{12} & \ldots & \tau_{12}x_{r2} & \ldots & \tau_{c2}x_{r2} \\ \vdots & \ldots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\ \tau_{1j} & \ldots & \tau_{cj} & \tau_{1j}x_{1j} & \ldots & \tau_{cj}x_{1j} & \ldots & \tau_{1j}x_{rj} & \ldots & \tau_{cj}x_{rj} \\ \vdots & \ldots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\ \tau_{1N} & \ldots & \tau_{cN} & \tau_{1N}x_{1N} & \ldots & \tau_{cN}x_{1N} & \ldots & \tau_{1N}x_{rN} & \ldots & \tau_{cN}x_{rN} \end{bmatrix} \tag{9}$$

Since number of equations in $H\vec{P} = \vec{y}$ is usually higher than number of unknowns, it is solved by Least Square Estimate method (LSE) where the error $e = (\vec{y} - H\vec{P})^T(\vec{y} - H\vec{P})$ is minimized.

$$e = \vec{y}^T\vec{y} - 2\vec{y}^T H\vec{P} + \vec{P}^T H^T H\vec{P}$$

$$\frac{\partial e}{\partial \vec{P}} = -2H^T\vec{y} + 2H^T H\vec{P} = 0 \Rightarrow \vec{P} = (H^T H)^{-1}H^T\vec{y}$$

However, sometimes $H^T H$ is ill-conditioned and pseudo-inverse of this matrix should be used. So, $\vec{P}$ is calculated from:

$$\vec{P} = (H^T H)^{+}H^T\vec{y} \tag{10}$$

where $(H^T H)^{+}$ is pseudo-inverse of $H^T H$.

3. Test cases

Results of the CFTS algorithm are compared with those of the other FTS algorithms and popular forecasting methods. Four test cases are studied to evaluate CFTS algorithm. Degree of fuzziness, m, is taken 2 for all cases but one can choose an optimal value of $m \in (1, \infty]$ which minimizes the testing error. We use Root Mean