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

**S. Askaria, N. Montazerina,[[1]](#footnote-2), M.H. Fazel Zarandib**

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.

**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.

***Keywords:*** Fuzzy time series, Fuzzy clustering, Fuzzy C-Means (FCM), Least square estimate (LSE), Forecasting

**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  be the universe of discourse on which fuzzy sets  are defined and  be a collection of s, then  is defined as a fuzzy time series on . In general,  is a linguistic variable with linguistic values, . If  is related to , the Fuzzy Logical Relationship (FLR) between them is represented by  which is a first order FLR. In this FLR,  and  are called current state and next state and denoted by  and , respectively, and their FLR is shown as . FLRs 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 [16,2].

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  and  as the input and output data of the FTS where  is number of observations and  is number of the FTS variables.  input data vector is  and its corresponding output is . We group  matrix into  clusters using Fuzzy C-Means algorithm, FCM [27]. For this purpose, following index is minimized with the given constraint [27]:

|  |  |
| --- | --- |
| (1) |  |

Where,  is the number of clusters,  is center of the  cluster ( row of cluster centers matrix, ),  is membership grade of the  data vector in the  cluster (element of partition matrix, ),  is distance,  is degree of fuzziness and  is the covariance norm matrix, defined as:

|  |  |
| --- | --- |
| (2) |  |

Since  is a symmetric matrix, . Using Lagrange Multipliers Method (LMM),  is written as:



Zeroing derivatives of  with respect to ,  and  yields:



Therefore,

|  |  |
| --- | --- |
| (3) |  |

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

|  |  |
| --- | --- |
| (4) |  |

Weighted contribution of each cluster in the calculation of the output associated with  is computed as:

|  |  |
| --- | --- |
| (5) |  |

Consider the matrix  such that: . Then output of CFTS algorithm for  is considered as the weighted linear combinations of the input variables.

|  |  |
| --- | --- |
| (6) |  |

The coefficients s are obtained by minimizing the following index:

|  |  |
| --- | --- |
| (7) |  |

Zeroing derivative of  with respect to  yields:



|  |  |
| --- | --- |
| (8) |  |

Using , (8) is written as the following set of  equations:

|  |  |
| --- | --- |
| (9) |  |

Since number of equations in  is usually higher than number of unknowns, it is solved by Least Square Estimate method (LSE) where the error  is minimized.



However, sometimes  is ill-conditioned and pseudo-inverse of this matrix should be used. So,  is calculated from:

|  |  |
| --- | --- |
| (10) |  |

Where,  is pseudo-inverse of .

**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, , is taken 2 for all cases but one can choose an optimal value of  which minimizes the testing error. We use Root Mean Square Error (RMSE) to compare the algorithms performances for both training and testing data sets.

|  |  |
| --- | --- |
| (11) |  |

Where,  is output of the model and  is number of data points used for training or testing. For all the cases, errors are reported as the average of thirty runs.

The first test case presents a detailed example of applying CFTS on the real data sets. The other test cases follow the same procedure.

**3.1. TAIEX data**

TAIEX data are frequently used for evaluation of the FTS algorithms. There are two variables and 226 data vectors in these data where 208 of them are used for training and the remaining data are used for testing as in the FTS algorithm of [19]. For all the TAIEX data sets studied in this work, the close price  is considered as a function of the high price  and low price . We use seven clusters as [19]. The following step-by-step procedure is used to apply CFTS algorithm on these data.

**Step 1:** Number of clusters , and degree of fuzziness , are assigned which are  and  in this case.

**Step 2:** The training input data are clustered using (3). Fuzzy clustering given in (3) is an iterative method which starts with a randomly assigned partition matrix  from which cluster centers  are computed. Then  is calculated from . If  where  is a predefined threshold (We use for all test cases.) and  is norm of , the solution is converged. If , this procedure is continued to  iterations until so that the clustering algorithm converges.  is final partition matrix from which final cluster centers matrix  is computed using (3). Final of these data is:



**Step 3**:  and  are calculated from (4) and (5), respectively. Then,  is computed using  and the training input data matrix .

**Step 4**: Parameters of the linear combinations, , are calculated using (10) where  is output of the training data set. Then, vector of parameters of the linear equation, , is extracted from . For this case,  is:



The following matrix includes parameters of the linear parts where the  row is parameters of the  linear equation, .



**Step 5**: For any training or testing input data vector ,  and  are calculated from (4) and (5), respectively. Then, output of the CFTS algorithm for  is computed by (6) as following:



Note that number of clusters , is not optional and should be computed. Optimal number of clusters is simply found by running the algorithm for  and choosing  corresponding to the minimum testing error. Although we take degree of fuzziness  but one can find the optimal  by the method used for . We use this method for all cases studied in this paper.

For this data set, training and testing errors of the CFTS algorithm are 35.58 and 34.38 and those of the clustering based FTS algorithm of [19] are 142.93 and 91.1, respectively. So, testing RMSE of the CFTS algorithm is 62% less than testing error of the FTS algorithm of [19]. Output of the algorithm for training and testing data is shown in Fig. 1.



Fig. 1. Comparison of the CFTS algorithm with the FTS algorithm of [19] for TAIEX data of 2004.

Moreover, TAIEX data of 1999 are simulated with five clusters and results of the CFTS algorithm are compared with those of the other FTS algorithms. Output of the CFTS algorithm is shown in Fig. 2.



Fig. 2. Output of the CFTS algorithm versus actual data for the TAIEX data of 1999.

These data were also studied using an FTS algorithm combined with Genetic Algorithms (FTSGA) [28]. It was shown that FTSGA is more accurate than other FTS algorithms. Testing RMSE of the CFTS algorithm is compared with those of the FTSGA and other algorithms given in [28] in Table 1. It is observed that testing RMSE of the CFTS algorithm is 47% less than that of the FTSGA algorithm.

Table 1. Testing RMSE of various forecasting algorithms for the TAIEX data of 1999.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FTSGA | Cheng | Huarng | Yu | Chen | CFTS |
| 103 | 109 | 109 | 142 | 149 | 54.78 |

Cluster centers and parameters of the CFTS algorithm for the TAIEX data of 1999 are:

,

One can assign linguistic terms to the cluster centers and express CFTS results as conditional fuzzy rules. For example, CFTS parameters of the TAIEX data of 1999 are written as:



There are two groups of FTS forecasting algorithms one based on the intervals of the universe of discourse and the other is based on the clustering. The differences between the CFTS algorithm and the other FTS algorithms are summarized as followings:

1. The interval based FTS algorithms need to find optimal length of each interval which is not fully understood and researchers are still trying to solve the problem. The CFTS algorithm is based on fuzzy clustering and does not use any interval.
2. The clustering based FTS algorithms deal with fuzziness, uncertainty and ambiguity of the data using fuzzy clusters instead of the intervals. They use FLRGs to establish relations between inputs and outputs but these relations are not strong enough to model behavior of the data with high precision. The interval based algorithms suffer from the same problem. These algorithms usually require high number of clusters or intervals to model the data with the desired testing error which incur high computational time. The CFTS algorithm handles fuzziness of the data by fuzzy clusters as the clustering based FTS algorithms but establishes input/output relations by linear combinations of the input variables instead of the FLRs. These linear relations provide higher degrees of freedom for learning from the training data set and make CFTS more accurate than the other FTS algorithms. Also, number of clusters needed for the CFTS algorithm to achieve the desired testing error is very smaller than those of the other FTS algorithms because of the linear relations. This makes CFTS computationally efficient and fast.
3. Both interval and clustering based FTS algorithms use FLRs as input/output relations. Using the forecasting FLR and the FLRGs, fuzzy value(s) of the forecast is computed which requires defuzzification. However, CFTS gives crisp value of the output and does not require any defuzzification.

**3.2. Enrollment problem**

Enrollment problem is the basic data set studied by the FTS algorithms where all the data are used for training. Training errors of various FTS algorithms are given in Table 2 which are taken from [17]. Seven clusters are used in this case since other algorithms in the table also use seven intervals or clusters.

Table 2. Training RMSE of various FTS forecasting algorithms for the enrollment problem, all collected from [17].

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Song and Chissom | Sullivan and Woodall | Chen | Huarng | Cheng et al. TFA | Cheng et al. | CFTS |
| 605.4 | 621.3 | 638.36 | 476 | 511 | 478.5 | 319.56 |

It is observed that RMSE of the CFTS algorithm is 33% lower than those of the other popular FTS algorithms. Cluster centers and parameters of the linear combination functions are:

,

**3.3. Box-Jenkins data**

This is a data set with five input variables and 294 data vectors. The first 270 data vectors are used for training and the rest for testing using four clusters. Training and testing RMSE of the CFTS algorithm for these data are 0.2264 and 0.4583, respectively. These data are also modeled using ANFIS (Adaptive Network Based Fuzzy Inference System) [29]. Training and testing RMSE of the ANFIS with triangular MFs, 32 rules and linear THEN part are obtained as 0.1408 and 0.4484, respectively. It is concluded that testing error of the CFTS algorithm with four clusters is comparable with that of the ANFIS with 32 rules. Output of the CFTS algorithm versus actual output is shown in Fig. 3 for the training and testing data sets.



Fig. 3. Actual output of the Box-Jenkins data versus output of the CFTS algorithm.

Cluster centers and parameters of the CFTS algorithm for the Box-Jenkins data are:

,

**3.4. Gas data**

Gas consumption data [17] have three variables and 1238 data vectors. 867 data vectors are used for training and the rest for testing the algorithm with ten clusters. Training and testing RMSE of the CFTS algorithm are 1.6272 and 1.6196 and those of the ANFIS with triangular MFs, 27 rules and linear THEN part are 1.4483 and 1.597, respectively. So, testing error of the CFTS algorithm is comparable with that of the ANFIS. Output of CFTS algorithm for gas data is shown in Fig. 4. Cluster centers and parameters of the linear functions of the CFTS algorithm for the gas data are:







Fig. 4. Output of the CFTS algorithm compared with actual gas data, (a) Training data (b) Testing data.

CFTS algorithm allows dealing with fuzziness, ambiguity, vagueness and uncertainty of the data using cluster centers as the other FTS algorithms but provides higher forecasting precision due to learning using parameters of the linear combinations which make it superior to the other FTS algorithms.

**4. Computational costs of the CFTS algorithm**

Sometimes, existing FTS algorithms need high number of clusters or intervals to learn behavior of the data which demands high computational time. Running time of the CFTS algorithm is compared to those of HMV-FTS and another clustering based FTS algorithm given in [17]. Results are given in Table 3.

Table 3. Comparison of running time (seconds) of the CFTS algorithm with other FTS algorithms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data set | No. of clusters | HMV-FTS algorithm | Cluster based algorithm | CFTS algorithm |
| Box-Jenkins data | 10 | 52.3777 | 51.2010 | 6.5019 |
| Enrollment data | 7 | 0.5989 | 0.5697 | 0.3227 |
| Gas data | 30 | 607.8671 | 550.5880 | 371.0061 |
| TAIEX data | 50 | 506.3641 | 476.2675 | 314.6091 |

It is observed that running time of the CFTS algorithm is considerably lower than that of other FTS algorithms especially when high number of clusters is used. CFTS algorithm learns behavior of the data with lower number of clusters than those required for the other FTS algorithms because of parameters of the linear parts. Although high number of clusters considerably reduces training error of the CFTS algorithm but increases testing error because of overtraining due to high number of parameters of the linear parts. Overtraining causes the algorithm to learn unimportant details of the data which reduces its generalization ability and increases testing error. So, number of clusters should be chosen such that the desired training or testing error acquired and excessive number of clusters should be avoided. Moreover, computational cost of the algorithm increases with number of clusters. As an instance, variation of the training and testing errors of CFTS for the Box-Jenkins data is shown in Fig. 5. It is observed that although CFTS algorithm learns more details of the data with higher cluster numbers which causes lower training error, but its generalization ability decreases because of the overtraining which is reflected by higher testing error.



Fig. 5. Variation of the CFTS algorithm performance with number of clusters, (a) Training error, (b) Testing error.

**5. Comparison of CFTS with recent FTS algorithms**

TAIEX data from 1997 to 2003 are studied in [30]. Now, the data for 1997 and 1998 are not available on the Internet. We study these data from 1999 to 2004 to compare CFTS with other FTS algorithms. We use data vectors of November and December of each of the data sets for testing and the rest for training CFTS as in [30]. Two FTS algorithms are presented in [30]. The first is based on Entropy Discretization (ED) and the second is combination of Entropy Discretization and Fast Fourier Transform (EDFFT) [30]. These algorithms are compared with 9 other methods including Chen’s algorithm, Yu’s algorithm, Chang’s algorithm, Hsieh’s algorithm, AR(1), AR(2), GARCH-M, GARCH-AR(1) and GARCH-AR(2) [30]. Testing error of CFTS is compared with those of the above algorithms as given in Table 4. Errors of these algorithms are taken from Table 9 of [30] and not computed in this work.

Table 4. Performances of various FTS algorithms, all collected from [30].

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1999 | 2000 | 2001 | 2002 | 2003 | Average |
| Chen | 120 | 176 | 148 | 101 | 74 | 124 |
| Yu | 145 | 191 | 167 | 75 | 66 | 129 |
| Chang et al. | 100 | 173 | 119 | 61 | 53 | 101 |
| Hsieh et al. | 86 | 135 | 93 | 62 | 59 | 87 |
| AR(1) | 244 | 375 | 515 | 98 | 97 | 266 |
| AR(2) | 246 | 376 | 506 | 99 | 92 | 264 |
| GARCH-M | 185 | 382 | 611 | 116 | 133 | 285 |
| GARCH-AR(1) | 103 | 151 | 102 | 72 | 64 | 98 |
| GARCH-AR(2) | 316 | 368 | 115 | 62 | 221 | 216 |
| ED | 89 | 116 | 61 | 67 | 46 | 76 |
| EDFFT | 83 | 115 | 62 | 60 | 46 | 73 |
| CFTS | 40 | 61 | 55 | 37 | 25 | 44 |

ED and EDFFT obviously outperform other algorithms. Based on the average testing error, EDFFT is the best among all the algorithms studied in [30]. But testing error of CFTS is 39.73% less than that of EDFFT algorithm.

TAIEX data of 2004 are studied in [31] where the last 45 data vectors are used for testing and the rest for training. Results of modeling this data set with various FTS algorithms are presented in [31]. We model these data using CFTS by taking training and testing data the same as those of [31].Training and testing RMSE of CFTS is compared to those of other FTS algorithms in Table 5. Note that results of the other FTS algorithms are taken from Table 2 of [31] and are not computed here.

Table 5. Performances of various FTS algorithms for the TAIEX data of 2004, all collected from [31].

|  |  |  |
| --- | --- | --- |
| Method | RMSE (Train) | RMSE (Test) |
| Song and Chissom | 102.11 | 77.86 |
| Chen (1996) | 92.79 | 77.18 |
| Chen (2002) | 19.71 | 71.98 |
| Huarng and Yu | 90.43 | 63.57 |
| Huarng et al. | - | 72.35 |
| Yu and Huarng | - | 67.00 |
| Aladag et al. | 45.83 | 69.80 |
| Chen and Chen | - | 57.73 |
| Aladag et al. | 44.48 | 47.95 |
| CFTS | 38.14 | 27.62 |

Testing error of [31] is the smallest one among those of the eight FTS algorithms. However, testing error of CFTS is 42.4% less than testing error of [31]. Results of modeling the TAIEX data from 1999 to 2004 are shown in Fig. 6.



Fig. 6. Actual and forecasted TAIEX data from 1999 to 2004.

Finally, TAIEX data from 2005 to 2014 are forecasted using CFTS. The last 45 data vectors of each data set are used for testing and the rest for training. Actual data and CFTS outputs for each year are shown in Figs. 7 and 8.



Fig. 7. Actual and forecasted TAIEX data from 2005 to 2010.



Fig. 8. Actual and forecasted TAIEX data from 2011 to 2014.

Training and testing errors of CFTS for the TAIEX data from 1999 to 2014 are given in Table 6. Also, optimal number of clusters for each data set is in the table which is chosen such that the testing error of CFTS becomes minimum.

Table 6. Training and testing errors and optimal number of clusters of CFTS for the TAIEX data from 1999 to 2014.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Year | RMSE (Train) | RMSE (Test) |  | Year | RMSE (Train) | RMSE (Test) |  |
| 1999 | 57.9444 | 39.8984 | 3 | 2007 | 33.7431 | 64.9964 | 3 |
| 2000 | 83.7910 | 60.7800 | 2 | 2008 | 54.8430 | 45.0709 | 6 |
| 2001 | 43.3371 | 54.7053 | 5 | 2009 | 40.5057 | 31.9392 | 2 |
| 2002 | 43.7570 | 36.7702 | 2 | 2010 | 31.9017 | 19.6151 | 5 |
| 2003 | 30.8842 | 24.5705 | 2 | 2011 | 36.9288 | 37.3153 | 5 |
| 2004 | 38.1382 | 27.6108 | 4 | 2012 | 29.0916 | 28.0484 | 3 |
| 2005 | 22.1847 | 25.1428 | 2 | 2013 | 25.3076 | 20.0725 | 2 |
| 2006 | 29.7853 | 24.8367 | 6 | 2014 | 24.5502 | 31.5219 | 3 |

**6. Conclusions**

A clustering based forecasting algorithm for fuzzy time series (CFTS) is proposed. CFTS algorithm deals with ambiguity, vagueness and uncertainty of the Fuzzy Time Series (FTS) using fuzzy clusters and replaces FLRs of the conventional FTS algorithms with linear combinations of the input variables. Parameters of the linear combinations are estimated by the Least Square Estimate (LSE) which enable the algorithm to learn behavior of the data more precisely compared to the existing FTS algorithms. The algorithm is evaluated using various data sets. Modeling the enrollment data as the basic problem of FTS shows that error of the CFTS algorithm is 33% less than that of the best algorithm among five popular FTS algorithms. Forecasting TAIEX data demonstrates that testing error of CFTS algorithm is 62% less than testing error of FTS algorithm. Moreover, it is shown that CFTS algorithm is 47% more accurate than the FTS algorithm combined with Genetic Algorithms (FTSGA). Although CFTS algorithm outperforms existing FTS algorithms but its effectiveness is confirmed if it provides satisfactory simulation of the FTS compared with other popular forecasting methods. For this purpose, Box-Jenkins and natural gas consumption data are forecasted using the CFTS algorithm and it is observed that testing errors of the algorithm are comparable with those of the ANFIS. It is also shown that CFTS algorithm is faster than the other FTS algorithms. High accuracy and low running time makes CFTS an efficient algorithm for forecasting fuzzy time series.

**References**

1. Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part I, Fuzzy Sets and Systems, 54 (1) (1993) 1-9.
2. Q. Song, B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets and Systems, 54 (3) (1993) 269-277.
3. Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part II, Fuzzy Sets and Systems, 62 (1) (1994) 1-8.
4. R. Efendi, Z. Ismail, M.M. Deris, A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand, Applied Soft Computing 28 (2015) 422–430.
5. K.H. Huarng, T.H.K. Yu, Ratio-based lengths of intervals to improve fuzzy time series forecasting, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 36 (2) (2006) 328-340.
6. J. Park, D.J. Lee, C.K. Song, M.G. Chun, TAIFEX and KOSPI 200 forecasting based on two-factor high-order fuzzy time series and particle swarm optimization, Expert Systems with Applications, 37 (2) (2010) 959-967.
7. W. Qiu, X. Liu, L. Wang, Forecasting shanghai composite index based on fuzzy time series and improved C-fuzzy decision trees, Expert Systems with Applications, 39 (9) (2012) 7680-7689.
8. H.L. Wong, Y.H. Tu, C.C. Wang, Application of fuzzy time series models for forecasting the amount of Taiwan export, Expert Systems with Applications, 37 (2) (2010) 1465-1470.
9. T.H.K. Yu, K.H. Huarng, A bivariate fuzzy time series model to forecast the TAIEX, Expert Systems with Applications, 34 (4) (2008) 2945-2952.
10. H.J. Teoh, T.L. Chen, C.H. Cheng, H.H. Chu, A hybrid multi-order fuzzy time series for forecasting stock markets, Expert Systems with Applications, 36 (4) (2009) 7888-7897.
11. S.T. Li, S.C. Kuo, Y.C. Cheng, C.C. Chen, A vector forecasting model for fuzzy time series, Applied Soft Computing 11 (3) (2011) 3125–3134.
12. D. Domanska, M. Wojtylak, Application of fuzzy time series models for forecasting pollution concentrations, Expert Systems with Applications, 39 (9) (2012) 7673-7679.
13. S.M. Chen, N.Y. Chung, Forecasting Enrollments of Students by Using Fuzzy Time Series and Genetic Algorithms, Information and Management Sciences, 17 (3) (2006) 1-17.
14. I.H. Kuo, S.J. Horng, T.W. Kao, T.L. Lin, C.L. Lee, Y. Pan, An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization, Expert Systems with Applications, 36 (3) (2009) 6108-6117.
15. Y.L. Huang, S.J. Horng, M. He, P. Fan, T.W. Kao, M.K. Khan, J.L. Lai, I.H. Kuo, A hybrid forecasting model for enrollments based on aggregated fuzzy time series and particle swarm optimization, Expert Systems with Applications, 38 (7) (2011) 8014-8023.
16. S.M. Chen, Forecasting enrollments based on fuzzy time series, Fuzzy Sets and Systems, 81 (3) (1996) 311-319.
17. S. Askari, N. Montazerin, A high-order multi-variable Fuzzy Time Series forecasting algorithm based on fuzzy clustering, Expert Systems with Applications, 42 (4) (2015) 2121-2135.
18. S.M. Chen, K. Tanuwijaya, Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques, Expert Systems with Applications, 38 (8) (2011) 10594-10605.
19. C.H. Cheng, G.W. Cheng, J.W. Wang, Multi-attribute fuzzy time series method based on fuzzy clustering, Expert Systems with Applications, 34 (2) (2008) 1235-1242.
20. E. Egrioglu, C.H. Aladag, U. Yolcu, V.R. Uslu, N.A. Erilli, Fuzzy time series forecasting method based on Gustafson-Kessel fuzzy clustering, Expert Systems with Applications, 38 (8) (2011) 10355-10357.
21. C.H. Aladag, U. Yolcu, E. Egrioglu, A.Z. Dalar, A new time invariant fuzzy time series forecasting method based on particle swarm optimization, Applied Soft Computing, 12 (10) (2012) 3291–3299.
22. O. Duru, E. Bulut, A non-linear clustering method for fuzzy time series: Histogram damping partition under the optimized cluster paradox, Applied Soft Computing, 24 (2014) 742–748.
23. U. Yolcu, E. Egrioglu, V.R. Uslu, M.A. Basaran, C.H. Aladag, A new approach for determining the length of intervals for fuzzy time series, Applied Soft Computing, 9 (2) (2009) 647–651.
24. M. Avazbeigi, S.H. Hashemi Doulabi, B. Karimi, Choosing the appropriate order in fuzzy time series: A new N-factor fuzzy time series for prediction of the auto industry production, Expert Systems with Applications, 37 (8) (2010) 5630-5639.
25. E. Egrioglu, C.H. Aladag, U. Yolcu, V.R. Uslu, M.A. Basaran, Finding an optimal interval length in high order fuzzy time series, Expert Systems with Applications, 37 (7) (2010) 5052-5055.
26. L. Wang, X. Liu, W. Pedrycz, Effective intervals determined by information granules to improve forecasting in fuzzy time series, Expert Systems with Applications, 40 (14) (2013) 5673-5679.
27. N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A Possibilistic Fuzzy c-Means Clustering Algorithm, IEEE Transactions on Fuzzy Systems, 13 (4) (2005) 517-530.
28. Q.S. Cai, D. Zhang, B. Wu, S.C.H Leung, A novel stock forecasting model based on fuzzy time series and genetic algorithm, Procedia Computer Science, 18 (2013) 1155–1162.
29. J.S.R. Jang, ANFIS: Adaptive-Network-based Fuzzy Inference Systems, IEEE Transactions on Systems, Man, and Cybernetics, 23 (3) (1993) 665-685.
30. M.Y. Chen, B.T. Chen, Online fuzzy time series analysis based on entropy discretization and a Fast Fourier Transform, Applied Soft Computing, 14 (2014) 156–166.
31. C.H. Aladag, U. Yolcu, E. Egrioglu, E. Bas, Fuzzy lagged variable selection in fuzzy time series with genetic algorithms, Applied Soft Computing, 22 (2014) 465–473.
1. Corresponding Author: mntzrn@aut.ac.ir, mntzrn@hotmail.com [↑](#footnote-ref-2)