Journal of Optimization in Industrial Engineering Vol.12, Issue 1, Winter and Spring 2019, 53-61
DOI: 10.22094/joie.2018.621.1399

Classification of Streaming Fuzzy DEA Using Self-Organizing Map

Alireza Alinezhad [a,*] , Mohammad Amin Adibi [a], Amineh Tohidi [a]

[a] *Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Received 11 May 2016; Revised 23 May 2017; Accepted 03 March 2018

Abstract

The classification of fuzzy data is considered as the most challenging areas of data analysis and the complexity of the procedures has been obstacle to the development of new methods for fuzzy data analysis. However, there are significant advances in modeling systems in which fuzzy data are available in the field of mathematical programming. In order to exploit the results of the researches on fuzzy mathematical programming, in this study, a new fuzzy data classification method based on data envelopment analysis (DEA) is provided when fuzzy data are imported as a stream. The proposed method can classify data that changes are created in their behavioral pattern over time using updating the criteria of predicting fuzzy data class. To reduce computational time, fuzzy self-organizing map (SOM) is used to compress incoming data. The new method was tested by simulated data and the results indicated the feasibility of this technique in the face of uncertain and variable conditions.

Keywords: Data Envelopment Analysis; Mathematical Programming; Classification; Streaming Fuzzy Data; Self-Organizing Map.

1. Introduction

Classification is to assign a class or category to data based on a predetermined function or model. Such a model is obtained through comparing the class and characteristics of a training dataset by various methods such as decision tree, artificial neural network (ANN), support vector machine (SVM), logistic regression, etc (Guerrero-Enamorado et al., 2016). With such a model, it is determined whether a data belongs to a particular class or not. The classification is one of the most common issues raised in the data analysis used in areas such as predicting the class of objects and situations, factors affecting the phenomenon, identifying abnormalities (abnormal conditions), etc.

One of the challenges of classification is to create classifying models for fuzzy data because in many cases, the collected data are not certain for various reasons. Recent researches conducted in this area are done by Colubi et al. (2011), Forghani et al. (2013), Shahraki and Zahiri (2015), and Quost et al. (2016).

However, in many cases, data which should be classified are not ready together but they are observed over time and so-called streaming data (Mena-Torres & Aguilar-Ruiz, 2014). In such situations, the question is how a model can be achieved to determine the class of data observed as a stream as well as having fuzzy aspects and changes in the conditions of studied system over time?

In such circumstances, using the capabilities of linear programming (LP) in terms of fuzzy data on one hand and capability of quickly updating the optimal solution, the classification based on data envelopment analysis (DEA) is become to an appropriate method (Pendharkar, 2011; Yan& Wei, 2011;Pendharkar&Troutt, 2014; Wei et al., 2014; Zhang et al., 2015).

Although DEA is used to calculate relative efficiency of a number of the decision making units (DMUs) which transform inputs to outputs, by changing the interpretation of the units, it can be applied in other applications such as classification of data. In this way, each data is considered as a DMU whose inputs and outputs are data characteristics and class number, respectively. It is shown that the space of the same data is the same area covered by the efficiency frontier of solving a DEA problem (Yan& Wei, 2011). In recent years, classification based on DEA has been used to some side applications of classification; for example Jiang and Lin (2015) used DEA for feature ranking and selection. Yang et al. (2017) proposed a rule reduction mechanism using DEA.

Using DEA method for classification, a classifier function by solving a linear programming problem can be achieved. This results in the widely use of the characteristics of linear programming including sensitivity analysis of answer in the case of changing the coefficients in classification which is exploited in this paper. Using DEA to data classification has attracted auspices recently.

Given that the computational time required solving linear programming model greatly rises via increasing the number of variables and constraints, using a mechanism controlling the aspects of a problem is required over time. This mechanism should be able to provide the suitable ground of optimization through controlling entry and exit of effective data on the efficiency frontier over time which is also provided in this paper. Fuzzy self-organizing map (SOM) is also accommodated to compress incoming data stream in the proposed mechanism. As far as we know, the proposed method is the first attempt to develop a classification tool for

*Corresponding author Email address: alinezhad@qiau.ac.ir

streaming fuzzy data

The rest of the paper is organized as follows; Classification using DEA is presented in Section 2. Classification of fuzzy data based on a linear mathematical programming is explained in Section 3. Fuzzy SOM is explained in Section 4 and the proposed method for streaming fuzzy data classification is provided in Section 5 as an integrated framework. Experiment is presented in Section 6 followed by results in Section 7.

2. Classification of Data Using DEA

It is supposed that we seek to identify a border (model) with the aim of data classification. If any data is considered as a DMU so that the values of characteristics of each data and 1 are as input of DMU and its only output, respectively, by solving a problem in DEA, data known as frontier point (FP) in DEA can be used to illustrate the range of category and then, these ranges can be used to predict the category or class of new data. This means if all $x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{im})$ in where $i = 1,2,\dots,n$ are in a same category or class, through solving a set of linear programming (LP) problems shown in Eq. (1) in the form of a DEA problem, the range of the category can be determined.

$$\begin{aligned} & Minimize\,\theta^t \\ & \textit{Subject to:} \\ & \sum_{i=1}^{n} \lambda_i x_{ij} - \xi^t x_{tj} \le 0, \quad j = 1, \dots, m \\ & \sum_{i=1}^{n} \lambda_i = 1 \\ & \lambda_i \ge 0\,, \; i = 1, \dots, n. \end{aligned} \qquad (1)$$

.

For example, consider dataset in Table 1 which is in two classes 1 and 2 (Pendharkar&Troutt, 2014).To identify the ranges of the classes, two series of DEA problems introduced in Eq. (2) and Eq.(3) are solved. The range obtained for the two classes 1 and 2 has been indicated in Fig 1 based on the frontier points. In DEA, frontier points are those in which the objective value (θ) is 1. In addition, dependent variable of θ in the solutions related to the LP problems for the first and second classes have been given in Tables 2 and 3 based on which the identification of the frontier points is possible.

$$\begin{aligned} & Minimize\,\theta^t \\ & \textit{Subject to:} \\ & \sum_{i=1}^{16} \lambda_i x_{ij} - \theta^t x_{tj} \le 0, \quad j = 1,2 \\ & \sum_{i=1}^{16} \lambda_i = 1 \\ & \lambda_i \ge 0\,, i = 1, \dots, 16. \end{aligned} \qquad (2)$$

$$\begin{aligned} & Maximize\,\theta^t \\ & \textit{Subject to:} \\ & \sum_{i=1}^{13} \lambda_i x_{ij} - \theta^t x_{tj} \ge 0, \quad j = 1,2 \\ & \sum_{i=1}^{13} \lambda_i = 1 \\ & \lambda_i \ge 0\,, i = 1, \dots, 13. \end{aligned} \qquad (3)$$

Table 1
Example data

Data	1st feature	2nd feature	Class	Data	1st feature	2nd feature	Class
1	640	6.02	1	17	310	4.7	2
2	550	6.09	1	18	350	4.5	2
3	510	5.67	1	19	400	4.7	2
4	420	5.54	1	20	370	4.8	2
5	560	6.75	1	21	450	4.7	2
6	550	6.60	1	22	500	4.5	2
7	580	5.87	1	23	520	4.6	2
8	420	6.20	1	24	550	4.3	2
9	450	6.77	1	25	570	4.5	2
10	520	5.67	1	26	450	4.9	2
11	440	5.33	1	27	320	4.6	2
12	480	5.96	1	28	400	4.6	2
13	520	6.13	1	29	310	5.1	2
14	570	6.26	1				
15	400	5.95	1				
16	580	5.2	1				

Table 2
Value of θ related to the data of class 1 in the LP model (for data of Table 1)

Data (Class1)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
θ	0.87	0.87	0.93	1.00	0.79	0.81	0.90	0.96	0.89	0.93	1.00	0.90	0.87	0.85	1.00	1.00

Table 3
Value of θ related to the data of class 2 in LP model (for data of Table 1)

Data (Class:2)	1	2	3	4	5	6	7	8	9	10	11	12	13
θ	108	1.11	1.05	1.04	1.03	1.04	1.01	1.04	1.00	1.00	1.10	1.07	1.00

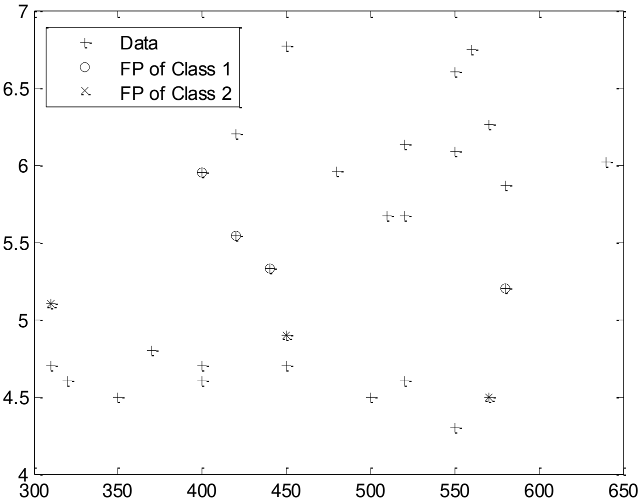

Fig. 1. The border of classes 1 and 2 based on frontier points relatedto data of Table 1

3. Classification of Fuzzy Data Using Fuzzy DEA

Assuming that the value of j th ($j = 1,2,\dots,m$) characteristic related to i th data ($i = 1,2,\dots,n$) is as a trapezoidal fuzzy number in the form of $\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij})$ that its membership function has been given in Fig 2, the pattern of linear programming model appropriate to a DEA problem will be in the form of Eq.(4).

$$\begin{aligned}
&\textbf{minimize } \boldsymbol{\theta^t}\\
&\textbf{Subject to:}\\
&\sum_{i=1}^{n} \lambda_i \tilde{x}_{ji} \le \theta^t \tilde{x}_{jt}, \quad j = 1,\dots,m\\
&\sum_{i=1}^{n} \lambda_i \tilde{y}_{ri} \ge \tilde{y}_{rt}, \quad r = 1,\dots,s\\
&\sum_{i=1}^{n} \lambda_i = 1\\
&\lambda_i \ge 0, \quad i = 1,\dots.,n
\end{aligned} \tag{4}$$

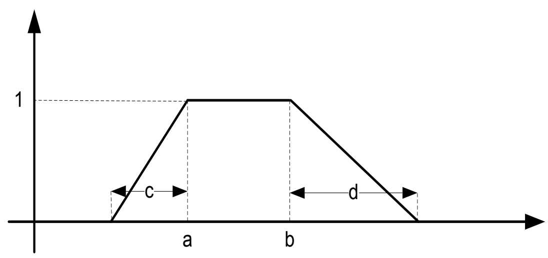

Fig. 2. Membership function of a trapezoidal fuzzy number as $\tilde{x} = (a, b, c, d)$

On the other hand, León et al. (2003) have provided a method for solving the problem shown in Eq.(4) that by using it, a certain amount is achieved for values of θ.The proposed model i.e. equivalent certain LP model has been given in Eq.(5) where $x^L = a$ ،$x^R = b$ ،$\alpha^L = c$,$\alpha^R = d$ and also $s$is the number of outputs of each DMU.

Table 4
Fuzzy numbers related to the example fuzzy data

Data	x_1^L	x_1^R	α_1^L	α_1^R	x_2^L	x_2^R	α_2^R	α_2^L	Class
1	3.25	4.00	0.50	0.50	3.50	4.25	0.50	0.50	1
2	3.00	4.00	0.50	0.50	4.75	5.25	0.50	0.50	1
3	3.00	4.00	0.50	0.50	5.75	6.25	0.50	0.50	1
4	2.00	3.00	0.50	0.50	5.75	6.25	0.50	0.50	1
5	4.00	5.00	0.50	0.50	4.75	5.25	0.50	0.50	1
6	5.00	6.00	0.50	0.50	3.00	3.50	0.50	0.50	1
7	5.00	6.00	0.50	0.50	6.00	6.50	0.50	0.50	1
8	4.00	5.00	0.50	0.50	3.75	4.25	0.50	0.50	1
9	4.96	5.96	0.50	0.50	5.31	5.81	0.50	0.50	1
10	5.26	6.26	0.50	0.50	5.38	5.88	0.50	0.50	1
11	1.50	2.50	0.50	0.50	0.50	1.00	0.50	0.50	2
12	1.25	2.25	0.50	0.50	1.75	2.25	0.50	0.50	2
13	1.25	2.25	0.50	0.50	2.75	3.25	0.50	0.50	2
14	0.25	1.25	0.50	0.50	2.75	3.25	0.50	0.50	2
15	2.25	3.25	0.50	0.50	1.75	2.25	0.50	0.50	2
16	3.25	4.25	0.50	0.50	2.00	2.50	0.50	0.50	2
17	3.25	4.25	0.50	0.50	2.00	2.50	0.50	0.50	2
18	2.25	3.25	0.50	0.50	0.75	1.25	0.50	0.50	2
19	3.21	4.21	0.50	0.50	2.31	2.81	0.50	0.50	2
20	3.51	4.51	0.50	0.50	2.38	2.88	0.50	0.50	2

For example, consider the data in Table 4 corresponded to 20 trapezoidal fuzzy data in two classes 1 and 2. For these values, two classes have been distinguished using Eq. (5) and the results are shown in Fig 3. It should be noted that only to plot fuzzy data in this figure; defuzzified value is considered as maximum-average of membership degree i.e. $(a+b)/2$. The value of θ for each LP problem related to DEA is presented in Table 5 and Table 6.

$$
\begin{aligned}
& minimize\ \theta^t \\
& Subject\ to: \\
& \sum_{i=1}^{n} \lambda_i x_{ji}^L \le \theta^t x_{jt}^L, \quad j = 1, \dots, m \\
& \sum_{i=1}^{n} \lambda_i x_{ji}^R \le \theta^t x_{jt}^R, \quad j = 1, \dots, m \\
& \sum_{i=1}^{n} \lambda_i x_{ji}^L - \sum_{i=1}^{n} \lambda_i \propto_{ji}^L \le \theta^t x_{jt}^L - \theta^t \propto_{jt}^L, j = 1, \dots, m \\
& \sum_{i=1}^{n} \lambda_i x_{ji}^R + \sum_{i=1}^{n} \lambda_i \propto_{ji}^R \le \theta^t x_{jt}^R - \theta^t x_{jt}^R, \ j = 1, \dots, m \\
& \sum_{i=1}^{n} \lambda_i y_{ri}^L \ge y_{rt}^L, \quad r = 1, \dots, s \\
& \sum_{i=1}^{n} \lambda_i y_{ri}^R \ge y_{rt}^R, \quad r = 1, \dots, s, \\
& \sum_{i=1}^{n} \lambda_i y_{ri}^L - \sum_{i=1}^{n} \lambda_i \beta_{ri}^L \le y_{rt}^L - \beta_{rt}^L, \quad r = 1, \dots, s \\
& \sum_{i=1}^{n} \lambda_i y_{ri}^R + \sum_{i=1}^{n} \lambda_i \beta_{ri}^R \ge y_{rt}^R - \beta_{rt}^R, \quad r = 1, \dots, s \\
& \sum_{i=1}^{n} \lambda_i = 1, \\
& \lambda_i \ge 0, \quad i = 1, \dots., n
\end{aligned}
\tag{5}
$$

Table 5
Value of θ related to the data of class 1 in fuzzy LP model (for data of Table 4)

Data (Class 1)	1	2	3	4	5	6	7	8	9	10
θ	1.00	0.94	0.87	1.00	0.82	1.00	0.69	0.95	0.74	0.72

Table 6
Value of θ related to the data of class 2 in fuzzy LP model (for data of Table 4)

Data (Class 2)	1	2	3	4	5	6	7	8	9	10
θ	1.67	1.31	1.00	1.00	1.25	1.06	1.06	1.34	1.03	1.00

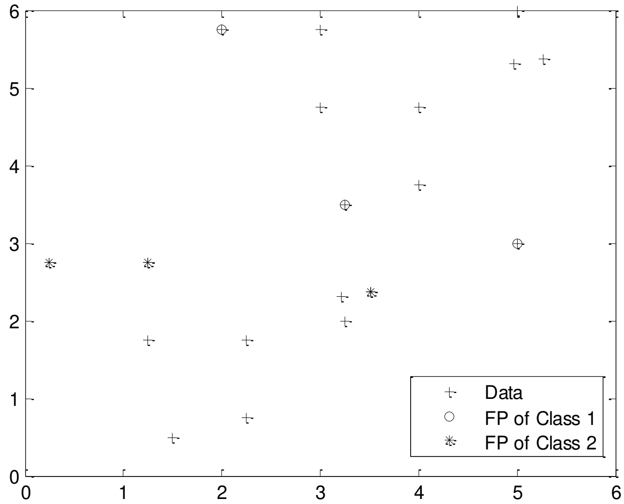

Fig. 3. The border of classes 1 and 2 based on frontier points for data of Table 4

4. Fuzzy Self-Organizing Map

As mentioned in introduction section, fuzzy SOM is used to compress the input fuzzy data in this paper. In its basic form, SOM consists of nodes arranged in a lower dimensional (commonly 2-dimansional) lattice to find a good mapping from the high dimensional input data to the lower dimensional representation of the nodes. Each node is associated with a weight vector with the same dimension (d) as the input data (Adibi & Shahrabi, 2017). In a training process of fuzzy SOM, each instance in the input fuzzy data is presented to the map and the winning unit or the best matching unit (BMU) is identified according to a distance measure. Consider $\tilde{x}$, $\widetilde{w}_{ij}$, and N_c as the input fuzzy data, fuzzy weight vector connecting the input fuzzy data to an output node with coordinates provided by indices i and j, and the neighborhood around the BMU respectively. So, training process of fuzzy SOM can be summarized in Fig. 4 (Aliahmadipour et al., 2017; Adibi & Shahrabi, 2015). Weight vector of the BMUs in a trained fuzzy SOM are used to be classified by mathematical model.

Step 1: Initialize all $\widetilde{w}_{ij}$, Choose a value for the neighborhood N_c and an initial learning rate α.
Step 2: Choose an instance $\tilde{x}$ from the input data set.
Step 3: Select the winning unit, c, so that:

$$\|\tilde{x} - \widetilde{w}_c\| = \min_{ij} \|\tilde{x} - \widetilde{w}_{ij}\|$$

Step 4: Update the weights as:

$$\widetilde{w}_{ij}(t+1) = \begin{cases} \widetilde{w}_{ij}(t) + \alpha(t)\left[\tilde{x} - \widetilde{w}_{ij}(t)\right] & if\ (i,j) \in N_c(t) \\ \widetilde{w}_{ij}(t) & if\ (i,j) \notin N_c(t) \end{cases}$$

Step 5: Decrease the learning rate and the neighborhood according to an appropriate scheme.
Step 6: Repeat steps (2)–(5) until the convergence criterion is satisfied

Fig. 4. Training process of the fuzzy SOM

5. Streaming Fuzzy Data Classification Using DEA

The proposed method in this study has been shown in Fig. 4 as a new framework in order to classify streaming fuzzy data using DEA. As it has been shown in this figure, DEA problems are first solved after compressing in order to identify frontier points based on training data that some or all of their properties have been valued as fuzzy numbers and the label of each data is specified. Borders obtained in each DEA problem will be used to label the training data. Before starting the classification of data stream, variable D is defined and is equal to zero. The variable will be applied in the process of classification of streaming data for saving the distance of data which do not confirm the model of identified classes. Also, S_{new} will be defined and used to hold data which do not confirm the model of recognized classes. This set is empty at the onset.

In the classification of the streaming data, if a new datum y is located in one of the identified ranges, it will be labeled according to that category.

To calculate the distance between two points equivalent to two trapezoidal fuzzy numbers, Eq. (6) is used (Chen et al., 2008). This equation is also used in the fuzzy SOM training process.

$$d(\tilde{x}_1, \tilde{x}_2) = (\frac{1}{6}[\left((a_1 - c_1) - (a_2 - c_2)\right)^2 + 2(a_1 - a_2)^2 + 2(b_1 - b_2)^2 + \left((b_1 + d_1) - (b_2 + d_2)\right)^2])^{\frac{1}{2}} \quad (6)$$

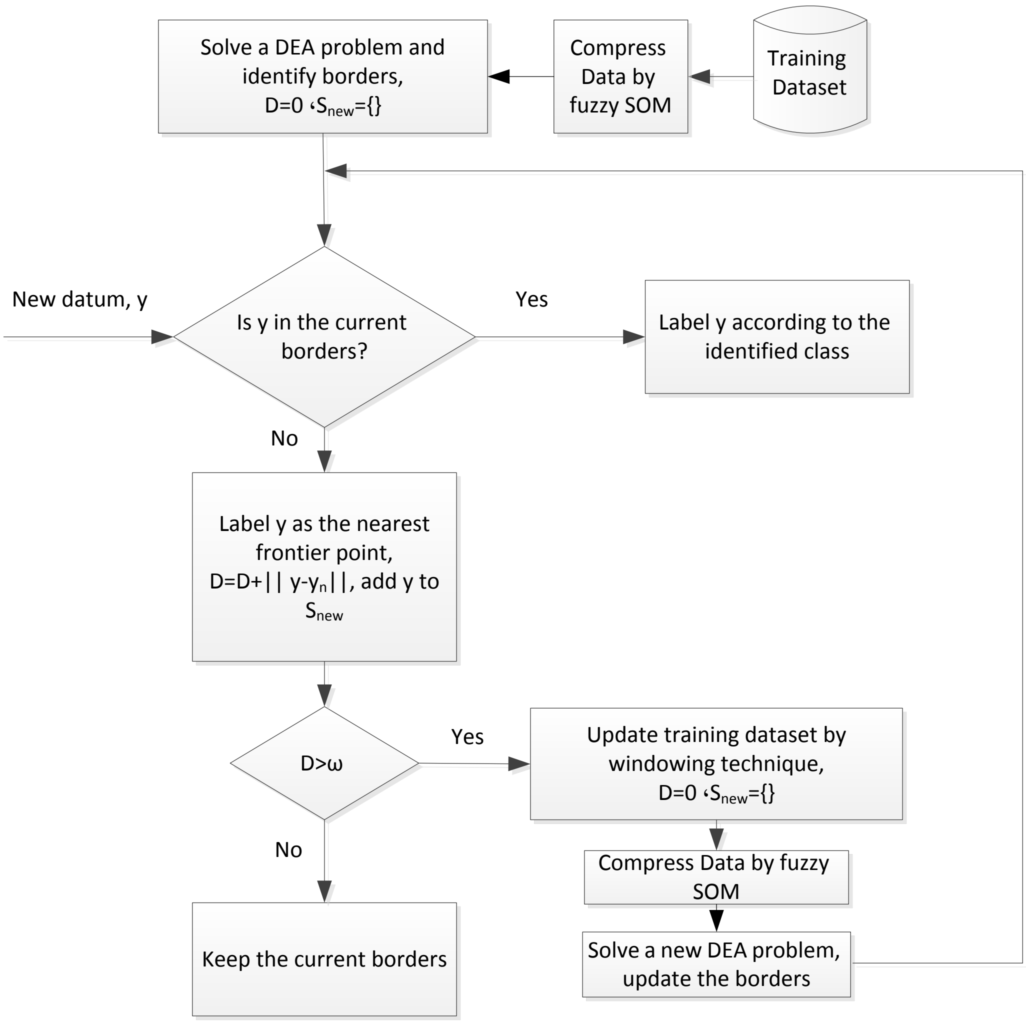

Fig. 4. The proposed method for classification of streaming fuzzy data

Otherwise, the label of data equals to the label of the nearest vector, y_n. Meanwhile, the distance between y and y_n is added to D and y is added to S_{new}. If D is lower than the predetermined threshold limit of ω, the range of classes will be determined for the next data based on the current models. Otherwise, it can be deduced that the current models are not significantly able to cover all data i.e. the system has changed in time and the change in the system has been manifested in the form of change in the behavior pattern of data. Therefore, the modification of the previous model is required.

To modify the models, given that the basis of areas is to solve DEA problems, new DEA problems should be solved based on new compressed data. The set of new data is achieved via adding the members of S_{new} to a set of the current training data and removing the same number of older members of current training set. Through this measurement called Windowing Technique, the number of the members of the training data set remains constant and the opportunity is created to set new boundaries that can do a correct classification based on the status of the system. After these measurements and modifications, a set of new frontier points is obtained and the modified model can be identifiable. At this stage, value of D turns into zero and S_{new} becomes empty. Then, the next new data can be classified with the updated model. The procedure will continue until there is a data stream.

6. Experiments

To generate initial training data as well as data that are fed to the proposed model as streaming data, the study by Yazdi et al. (2009) is adapted. In their study, only classification of fuzzy data in static mode has been considered. So, a modified method is used to generate streaming fuzzy data which is accompanied with drift. For this purpose, to produce 400 (two-dimensional) initial training data, we use random distribution as presented in Tables 7 and 8. To create data entered as a stream, after creation of 400 new data using the method outlined in Tables 9 and 10 in the second phase, 1800 fuzzy data is randomly created between these two groups of data. In fact, a total of 400 initial training data and 2200 data as a stream are produced. By doing this, a drift is induced in the streaming data. Fig 5 shows the first and second group of compressed data which 1800 intermediate data will be distributed among them.

Table 7
Random values distribution for creating fuzzy data for the class 1 in order to be used as the initial training data and also initial state of the data

1st feature	$a \sim N(7,1.5)$, $b' \sim U(0,1)$, $b = a + b'$, $c, d \sim U(0.2,0.7)$
2nd feature	$a \sim N(9,1.5)$, $b' \sim U(0,1)$, $b = a + b'$, $c, d \sim U(0.2,0.7)$

Table 8
Random values distribution for creating fuzzy data for the class 2 in order to be used as the initial training data and also initial state of the data

1st feature	$a \sim N(3,1.5)$, $b' \sim U(0,1)$, $b = a + b'$, $c, d \sim U(0.2,0.7)$
2nd feature	$a \sim N(3,1.5)$, $b' \sim U(0,1)$, $b = a + b'$,$c, d \sim U(0.2,0.7)$

Table 9
Random value distribution for creating fuzzy data for the class 1 in order to be used as final state of the data

1st feature	$a \sim N(8,1.5)$ ، $b' \sim U(0,1)$ ، $b = a + b'$ ، $c, d \sim U(0.2,0.7)$
2nd feature	$a \sim N(9,1.5)$ ، $b' \sim U(0,1)$ ، $b = a + b'$ ، $c, d \sim U(0.2,0.7)$

Table 10
Random value distribution for creating fuzzy data for the class 2 in order to be used as final state of the data

1st feature	$a \sim N(2,1.5)$ ، $b' \sim U(0,1)$ ، $b = a + b'$ ، $c, d \sim U(0.2,0.7)$
2nd feature	$a \sim N(4,1.5)$ ، $b' \sim U(0,1)$ ، $b = a + b'$ ، $c, d \sim U(0.2,0.7)$

After running the new method of classification of fuzzy data stream using DEA in the environment of MATLAB® software, F1-score which is defined as Eq. (7) is used to assess the proposed method. So that precision of the model includes what proportion of the data in a class are from the considered category and recall is also a ratio of the particular category data belonged to a class. The results from applying the proposed method have been given in Table 11.It is worth mentioning that to solve linear programming problems with which we encounter in DEA solving problems, YALMIP(Lofberg, 2010) software implemented in MATLAB® has been used.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

Table 11
The value of Recall, Precision and F1-measure for the simulated streaming fuzzy data for online and offline classification

	Online Classification			Offline Classification		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Class 1	0.9243	0.9711	0.9504	0.8036	1.0000	0.8911
Class 2	0.9655	0.9112	0.9336	1	0.7556	0.8608
Total	-	-	**0.9419**			**0.8759**

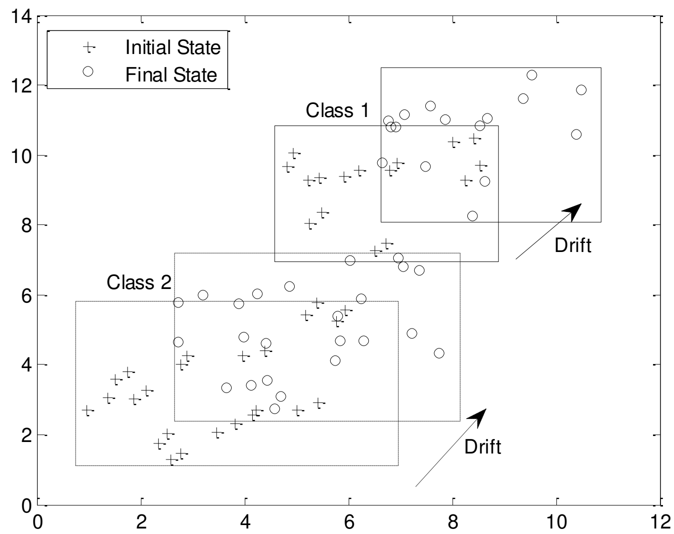

Fig. 5. Initial and final state of the streaming data used in the experiment

In Table 11, F1-score is also presented for situation in which no modification in classification criteria is performed or offline classification. As it is obvious, overall F1-score is 0.8759 in this situation. In fact, the proposed method has enhanced the measure from 0.8759 to 0.9419 which shows its effectiveness.

7. Conclusion

In this research a method for classification of streaming fuzzy data is proposed. The proposed method is based on a fuzzy linear programming model which is inspired by data classification using DEA. For effective use of the proposed model, a general framework is also proposed. The proposed framework permits effective classification of streaming fuzzy data which includes drifts over time as well as compressing incoming fuzzy data by fuzzy SOM. As the first streaming fuzzy data classification tool, the proposed method for the simulated streaming fuzzy data has been examined by F1-score measure. The results show the effectiveness of the proposed method for the test conditions in which F1-score reached to 0.948. Since DEA is a nonparametric method, the proposed method stays away from parameter setting challenge so that only the parameter ω should be determined by the user. All parameters related to the fuzzy SOM affects only computational time. Sensitivity analysis for the linear programming instead resolving it can be a future research idea which reduces computational time.

References

Adibi, M. A., & Shahrabi, J. (2015). Online Anomaly Detection Based on Support Vector Clustering. International Journal of Computational Intelligence Systems, 8(4), 735-746.

Adibi, M. A., & Shahrabi, J. A time-varying quadratic programming for online clustering of streaming data. Pattern Analysis and Applications, 1-10. doi:10.1007/s10044-017-0608-9

Aliahmadipour, L., Torra, V., & Eslami, E. (2017). On Hesitant Fuzzy Clustering and Clustering of Hesitant Fuzzy Data. In Fuzzy Sets, Rough Sets, Multisets and Clustering, 157-168 Springer International Publishing.

Chen, T. Y., Ku, T. C., & Tsui, C. W. (2008). Determining attribute importance based on triangular and trapezoidal fuzzy numbers in (z) fuzzy measures. In *The 19th international conference on multiple criteria decision making,* 75-76

Colubi, A., González- Rodríguez, G., Gil, M. Á., &Trutschnig, W. (2011). Nonparametric criteria for supervised classification of fuzzy data.*International Journal of Approximate Reasoning*, *52*(9), 1272-1282.

Forghani, Y., SadoghiYazdi, H., &Effati, S. (2013). Classification of fuzzy data based on the support vector machines. *Expert Systems*, *30*(5), 403-417.

Guerrero-Enamorado, A., Morell, C., Noaman, A. Y., & Ventura, S. (2016). An algorithm evaluation for discovering classification rules with gene expression programming. *International Journal of Computational Intelligence Systems*, *9*(2), 263-280.

Jiang, C., & Lin, W. (2015). DEARank: a data-envelopment-analysis-based ranking method. Machine Learning, 101(1-3), 415-435.

León, T., Liern, V., Ruiz, J. L., &Sirvent, I. (2003). A fuzzy mathematical programming approach to the assessment of efficiency with DEA models.*Fuzzy sets and systems*, *139*(2), 407-419.

Lofberg, J. (2010). Yalmip.

Mena-Torres, D., & Aguilar-Ruiz, J. S. (2014). A similarity-based approach for data stream classification.*Expert Systems with Applications*, *41*(9), 4224-4234.

Pendharkar, P. C. (2011). A hybrid radial basis function and data envelopment analysis neural network for classification.*Computers & Operations Research*, *38*(1), 256-266.

Pendharkar, P. C., &Troutt, M. D. (2014). Interactive classification using data envelopment analysis.*Annals of Operations Research*, *214*(1), 125-141.

Quost, B., &Denœux, T. (2016). Clustering and classification of fuzzy data using the fuzzy EM algorithm.*Fuzzy Sets and Systems*, *286*, 134-156.

Shahraki, H., &Zahiri, S. H. (2015, March). Particle swarm classifier for fuzzy data sets. In *Artificial Intelligence and Signal Processing (AISP), 2015 International Symposium on,* 295-299. IEEE.

Wei, Q., Chang, T. S., & Han, S. (2014). Quantile–DEA classifiers with interval data.*Annals of Operations Research*, *217*(1), 535-563.

Yan, H., & Wei, Q. (2011). Data envelopment analysis classification machine.*Information Sciences*, *181*(22), 5029-5041.

Yang, L. H., Wang, Y. M., Lan, Y. X., Chen, L., & Fu, Y. G. (2017). A data envelopment analysis (DEA)-based method for rule reduction in extended belief-rule-based systems. Knowledge-Based Systems, 123, 174-187.

Yazdi, H. S., Yazdi, M. S., &Vahedian, A. (2009). Fuzzy Bayesian classification of LR Fuzzy numbers. *International Journal of Engineering and Technology*, *1*(5), 415.

Zhang, Y., Yang, C., Yang, A., Xiong, C., Zhou, X., & Zhang, Z. (2015). Feature selection for classification with class-separability strategy and data envelopment analysis. *Neurocomputing*, *166*, 172-184.

This article can be cited: Alinezhad A., Adibi M.A. & Tohidi .A. (2019). Classification of Streaming Fuzzy DEA using Self-Organizing Map. *Journal of Optimization in Industrial Engineering*. 12 (1), 53-61.

http://www.qjie.ir/article_539596.html
DOI: 10.22094/JOIE.2018.621.1399