2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A Hybrid Type-2 Fuzzy Clustering Technique for Input Data Preprocessing of Classification Algorithms

Vahid Nouri, Mohammad-R. Akbarzadeh-T.
Departments of Computer Engineering,
Islamic Azad University, Mashhad Branch,
Mashhad, Iran
Emails: vahid.nouri@mshdiau.ac.ir, akbarzadeh@ieee.org

Alireza Rowhanimanesh
Department of Electrical Engineering,
University of Neyshabur
Neyshabur, Iran
rowhanimanesh@ieee.org

Abstract— Recently, clustering has been used for preprocessing datasets before applying classification algorithms in order to enhance classification efficiency. A strong clustered dataset as input to classification algorithms can significantly improve computation time. This can be particularly useful in Big Data where computation time is equally or more important than accuracy. However, there is a trade-off between speed and accuracy among clustering algorithms. Specifically, general type-2 fuzzy c-means (GT2 FCM) is considered to be a highly accurate clustering approach, but it is computationally intensive. To improve its computation time we propose here a hybrid clustering algorithm called KGT2FCM that combines GT2 FCM with a fast k-means algorithm for input data preprocessing of classification algorithms. The proposed algorithm shows improved computation time when compared with GT2 FCM as well as KFGT2FCM on five benchmarks from UCI library.

Keywords- General tye-2 fuzzy, k-means, clustering, input data preprocessing, classification

I. INTRODUCTION

Classification is a common problem data mining [2] where datasets are mapped into predefined groups called classes. Classes are defined according to the similarity of characteristics or features of data [1]. Because, the classes are determined before applying the real data, this method is known as a supervised learning algorithm. Classification is used in many fields and sciences such as, bio-informatics [15], genetics [14], biology [16] and healthcare [17].

Several researches have shown that the computational efficiency of classification is enhanced if the input data is first clustered before for classification. This is particularly applicable when handling big data, where low computation time is equally or more important than classification accuracy. The class information also improves the accuracy of clustering [12]. To have the advantages of both clustering and classification, many existing algorithms have been developed in a sequential hybrid way [12]. For example in both [11] and [12], first the criterion is preprocessed and optimized by a clustering algorithm and then in the next step the classification criterion is connected with the achieved clustering results to enhance the accuracy of classification algorithms. Generally, there is a trade-off between accuracy and computation time of clustering algorithms, i.e. the higher the accuracy, the more the computation time. One of the better known clustering algorithms is k-means. K-means is fast but has low accuracy [1]. On the other hand, general type-2 fuzzy clustering (GT2 FCM) is a new method that has high accuracy but is computationally intensive. In [4], a general type-2 fuzzy clustering algorithm is introduced that is based on α-planes. This algorithm has high accuracy and can deal with the uncertainty in datasets, while k-means and FCM, which are fast clustering algorithms, cannot handle the uncertainty in a dataset.

There are several works that focus on enhancing the speed issue of type-2 fuzzy clustering. A modified version of type-2 fuzzy system was proposed in [6] to improve the speed (computational time) of type-2 fuzzy clustering. Also, in [7]-[9] interval type-2 fuzzy is used instead of general type-2 fuzzy for clustering, because interval type-2 is faster than general type-2 fuzzy.

In addition Yang worked on similarity measurements of type-2 fuzzy clustering algorithms on fuzzy datasets [19]-[22]. In these work, they redefined new similarity measurements based on union, maximum. These new similarity measures affect type-2 fuzzy clustering performance.

In this paper, we propose a hybrid clustering algorithm for data input preprocessing of classification algorithms to address the high computation time of general type-2 fuzzy clustering algorithm. The proposed hybrid method is based on a combination of general type-2 fuzzy, which is an accurate algorithm and k-means, which is a fast algorithm. We call the proposed approach KGT2FCM. KGT2FCM has the advantages of both general type-2 fuzzy and k-means clustering algorithms, i.e. it has high accuracy and low computational time. The results are compared with GT2 FCM and KFGT2FCM clustering algorithms for different datasets. Unlabeled datasets are used for clustering algorithms; however, labeled datasets are used for classification algorithms. Since, we use classification datasets in our experiments; we can measure the accuracy of our clustering algorithm. The paper is organized as follows: section II discusses the proposed hybrid algorithm. The results and conclusion are presented in sections III and IV, respectively.

II. PROPOSED METHOD

Our method is based on k-means and general type-2 fuzzy clustering. General type-2 fuzzy clustering was presented in [4]. First, a general overview of type-2 fuzzy is given, and then the proposed method is described.

A. General Type-2 Fuzzy Clustering

There are two kinds of type-2 fuzzy sets which are often used in clustering algorithms: 1) interval and 2) general. In interval type-2 fuzzy, the secondary membership function is always one, while in general type-2 fuzzy it is a value in range of [0,1].

General type-2 fuzzy clustering is based on FCM (Fuzzy C-Means) algorithm. So, the same as FCM, it initializes the centers randomly. The FCM algorithm uses linguistic terms such as "Small", "Medium" or "High", modeled by type-1 fuzzy sets for the fuzzifier parameter M (Fig.1). The FCM algorithm is used by the GT2 FCM cluster membership functions. The general type-2 fuzzy clustering proposed in [4] uses α-planes. α-planes manage the uncertainty of general type-2 fuzzy sets. The GT2 FCM algorithm exploits the linguistic fuzzifier M for its secondary membership functions of the general type-2 fuzzy partition matrix $\tilde{u}_j$ as shown in (1). Equation (2), that is a membership grade $\tilde{u}_j(x_i)$ expressed as type-1 fuzzy sets, is used to describe the membership degree of pattern x_i to cluster v_j.

$$\tilde{u}_j = \sum_{x_i \in X} \tilde{u}_j(x_i) \tag{1}$$

$$\begin{aligned} \tilde{u}_j(x_i) &= \bigcup_{\alpha\in[0,1]} \alpha \Big/ S_{\tilde{u}_j}(x_i \mid \alpha) \\ &= \bigcup_{\alpha\in[0,1]} \alpha \Big/ [s^L_{\tilde{u}_j}(x_i \mid \alpha), s^R_{\tilde{u}_j}(x_i \mid \alpha)] \end{aligned} \tag{2}$$

Where $s^R_{\tilde{u}_j}(x_i \mid \alpha)$ and $s^L_{\tilde{u}_j}(x_i \mid \alpha)$ are calculated by (3) and (4):

$$s^R_{\tilde{u}_j}(x_i \mid \alpha) = \max\left(\frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}}{d_{il}}\right)^{\frac{2}{s^L_M(\alpha)-1}}}, \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}}{d_{il}}\right)^{\frac{2}{s^R_M(\alpha)-1}}} \right) \tag{3}$$

$$s^L_{\tilde{u}_j}(x_i \mid \alpha) = \min\left(\frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}}{d_{il}}\right)^{\frac{2}{s^L_M(\alpha)-1}}}, \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}}{d_{il}}\right)^{\frac{2}{s^R_M(\alpha)-1}}} \right) \tag{4}$$

According to [4], centroid $C_{\tilde{u}_j}$ can be calculated as a weighted composition of the interval centroids of individual α-planes using (5). The input of (5) is $\tilde{u}_j$. Here, d_{ij} is the distance of ith data from jth centroid. Initial centroids are used for the first iteration. s^R_M and s^L_M are obtained as shown in Fig.2 for each α-planes and c is the number of clusters.

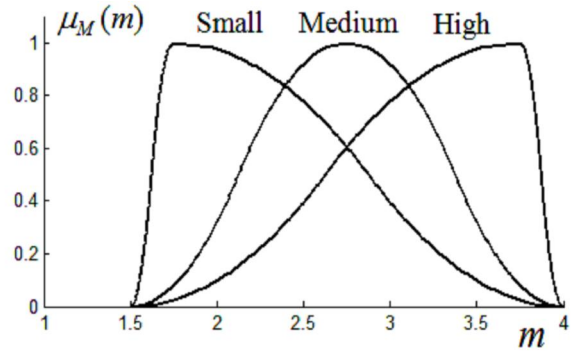

Fig. 1. Linguistic variables for initializing the membership functions [4]

To compute the precise cluster position, (6) is used to defuzzify the cluster centroid $C_{\tilde{u}_j}$.

$$C_{\tilde{u}_j} = \bigcup_{\alpha\in[0,1]} \alpha \Big/ [c^L_{\tilde{u}_j}(\alpha), c^R_{\tilde{u}_j}(\alpha)] \tag{5}$$

$$v_j = \frac{\sum_{i=1}^{K} y_i C_{\tilde{u}_j}(y_i)}{\sum_{i=1}^{K} C_{\tilde{u}_j}(y_i)} \tag{6}$$

In (6), K is the number of steps that the domain of the centroid has been discretized into and y_i is the position vector of ith discretized step. In this algorithm the hard-partitioning is done based on the defuzzified value of the type-1 fuzzy membership grade. So, the following rule is used for hard-partitioning:

$$\begin{aligned} &If(\tilde{u}_j(x_i) > \tilde{u}_k(x_i)), k = 1,\ldots,c, k \neq j \\ &Then\ \ x_i \in Cluster\ \ j \end{aligned} \tag{7}$$

But in [4] formula (8) is used for hard-partitioning instead of (7). In (7), since the Euclidian distance norm is used to calculate the membership of pattern x_i to cluster j in the multidimensional space, it seems redundant to separately aggregate identical membership values for each dimension. So in [4] the authors use (8) for hard-partitioning:

$$\begin{aligned} &If(c(\tilde{u}_j(x_i)) > c(\tilde{u}_k(x_i))), k = 1,\ldots,c, k \neq j \\ &Then\ \ x_i \in Cluster\ \ j \end{aligned} \tag{8}$$

The centroid of the type-1 fuzzy membership grade $c(\tilde{u}_j(x_i))$ can be calculated using (9):

$$c(\tilde{u}_j(x_i)) = \frac{\sum_{i=1}^{K} y_i \tilde{u}_j(y_i)}{\sum_{i=1}^{K} \tilde{u}_j(y_i)} \tag{9}$$

In this equation, K and y_i have the same definitions as in (6), where $c(\tilde{u}_j(x_i))$ is the centroid of the jth cluster.

Schematic view of GT2 FCM is depicted in Fig.2. Random selection of initial centroids causes the algorithm to have more iterations, hence, more computation time (lower speed). So, if a clustering algorithm, such as k-means, finds the centers one step before GT2 FCM and passes them to GT2 FCM, the computation time of GT2 FCM will be reduced.

B. K-means

K-means is one of the most common algorithms in clustering. In this method, k denotes the number of clusters. K-means algorithm has three steps including:

Step 1) k cluster centers are specified, randomly i.e. one center for each cluster, step 2) for each input, distance from each cluster center is calculated. The data belongs to the cluster which has the closest distance from the center. This step is repeated for all input data, and step 3) the barycenters of clusters (which are generated in step 2) are calculated and considered as new cluster centers and then the algorithm goes to step 2 [1]. These steps are repeated until centers do not change for the two consecutive iterations. The goal of this algorithm is to minimize its cost function denoted as (10) [1],[3].

$$J = \sum_{i=1}^{n}\sum_{j=1}^{K}\left\|x_i - c_j\right\|^2 \qquad (10)$$

Here, n is the number of samples, K is the number of clusters, c_j shows the j[th] cluster and x_i shows i[th] sample of pattern.

In this paper we use Euclidian distance which is a traditional metric for distance measurement of k-means. Following equation presents the Euclidian distance [10]:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (11)$$

In this equation $d(x,y)$ is the Euclidean distance between sample x with cluster y.

C. Previous work

To improve the computation time of GT2 FCM, the algorithm at [10] uses the output of k-means algorithm as the input for FCM. Then, the outputs of FCM are used as the initial cluster centroids for GT2 FCM. In the following the details are discussed. In the conventional FCM, the initial values of membership functions are random numbers in the range of [0, 1]. However, if the initial values of membership functions are selected more wisely, it is expected to need less number of iterations by FCM, hence, the computation time is improved. To do so, the authors of [10] use k-means to determine the centroids of input data and then calculate the distances of each data from all centroids.

The normalized distances are assumed as initial values of membership functions of input data of FCM. By doing so, FCM would have a better starting point and it helps to reduce the execution time and iterations of FCM. This algorithm is called KFGT2FCM. However, using both k-means and FCM clustering algorithms for initializing GT2 FCM causes almost a large overhead.

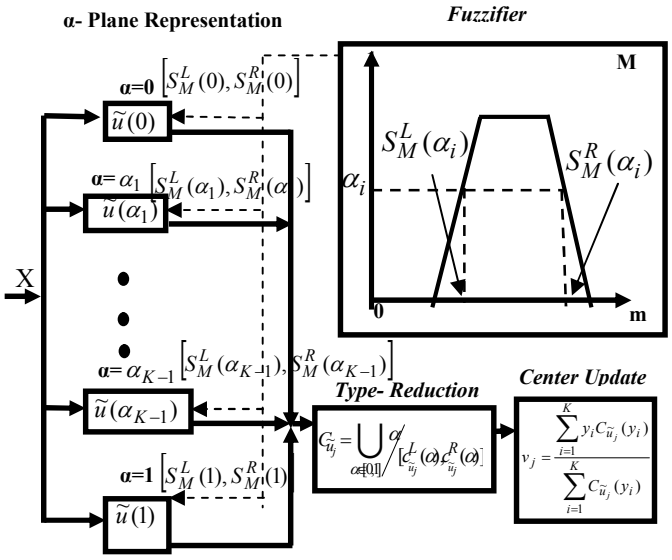

Fig. 2. Schematic view of GT2 FCM [4]

D. Proposed Method

To make the algorithm even faster, we can omit one of the k-means or FCM algorithms which generate initial centroids for GT2 FCM. We omit FCM algorithm from the flow because k-means is faster than FCM. Also, clusters' centroids detected by FCM and k-means clustering algorithms are close to each other. However, the complexity time of k-means and FCM are O(ncdi) and O(ndc2i),respectively [23], where, n denotes number of dimensions, i is the number of iterations, n shows the number of sample of dataset and c is the number of clusters. According to [23], for n=100, d=3, i=20 and for a constant number of dataset's samples, the elapsed time for k-means and FCM are 0.443755 and 0.781679 seconds, respectively. Also, if we assume the number of clusters as a constant and assume n=150, d=2, c=2 and i=20, then the time complexity of k-means and FCM are 12000 and 24000, respectively [23]. So, FCM needs more computation time than k-means. Hence, in the revised version just k-means is employed to find the centroids of GT2 FCM. This change improves the computation time of the algorithm and consequently, the new algorithm is executed much faster than KFGT2FCM. This new algorithm is called KGT2FCM.

The flowchart of KGT2FCM is depicted in Fig.3. As depicted in Fig.3, first of all the input dataset are applied to the k-means algorithm. Then, k-means clusters input dataset and finds their centroids. After that, these centroids are used by GT2 FCM as the initial centroids. In the next step, the type-2 fuzzifier function calculates the secondary membership functions based on α-planes and using (3), (4) and "Medium" linguistic term for secondary membership function as depicted in Fig.1. We use 10 α-planes. Then, EKM[1] algorithm [13] is used for type reduction and finding the centroids of α-planes. EKM introduced by Mendel and Wu to enhance the computation time of KM. EKM is 39% faster than KM algorithm and can save about two iterations while KM find the answer usually between two to six iterations [13]. In our proposed method, for type reduction we use (5) which are based on EKM algorithm, to find the centroids of 10 α-planes.

[1]Enhancement Karnik Mendel

Doing so, the type-2 fuzzy membership function reduces to type-1 fuzzy which is a primary membership function. To find the precise center of each cluster, the centroids should be determined using (6). The centroids calculated by (6) are checked with previous centroids of each cluster. If they are different, the algorithm recalculates the secondary membership function using new centroids and then the following steps are repeated. Otherwise, the algorithm finishes (Fig. 3).

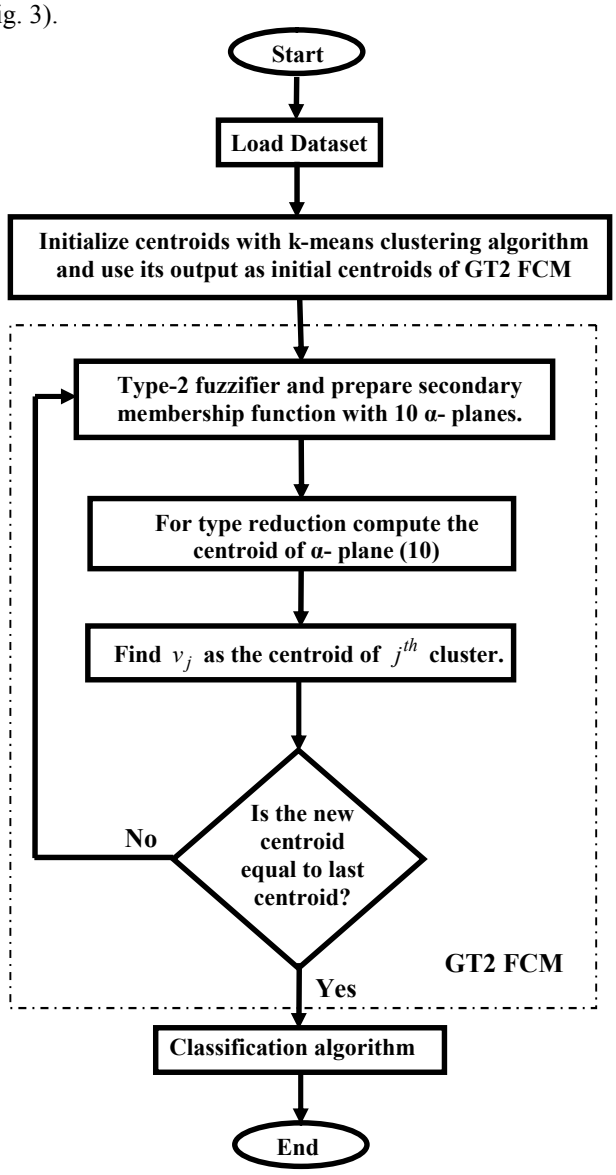

Fig. 3. Flowchart of proposed KGT2FCM

III. SIMULATION RESULTS

In this section the experimental setup and simulation results are presented.

A. Experimental Setup

In this paper, five standard datasets of UCI are selected, including Iris, Wine, Pima Indians, Shuttle and Magic which have been listed in table I [18]. The Shuttle data has been divided into two classes. One class (class 1) includes the most numerous data class which is 80% of data and the second class (class 2) contains the remaining less numerous data classes which is the 20% of data. All of the datasets of table I are applied to GT2 FCM, KFGT2FCM and KGT2FCM 50 times. The machine used for doing the experiments and simulations is an Acer 5750G system with an Intel Core i7-2630QM@2.00GHz and 6.00 GB RAM and running Windows 7. MATLAB software has been used for implementing the algorithms. For fair comparisons of computation time of the three algorithms, the target accuracy has been assumed the same for all the three algorithms in all the experiments. All of the three algorithms (i.e. GT2 FCM, KFGT2FCM and KGT2FCM) are based on GT2 FCM [4], and use the same membership functions (i.e. the same initial conditions).

TABLE I
LIST OF DATASETS THAT USED FOR EXPERIMENTS

Dataset	Clusters	Size	Attributes
Iris	3	150	4
Wine	3	178	13
Pima Indians	2	768	8
Shuttle	2	43,500	9
Magic	2	19,020	10

Since initial centroids of k-means and GT2 FCM are selected randomly, we run each algorithm for 50 iterations, i.e. with 50 sets of random initial centroids, to show that the random initial centroids have trivial effects on the results. The computation time improvement (speedup) of KGT2FCM compared to GT2 FCM and KFGT2FCM is calculated using equation (12). Therefore, when the speedup is greater than one the computation time of KGT2FCM is less than GT2 FCM or KFGT2FCM and when the speedup is less than one the computation time of KGT2FCM is greater than GT2 FCM or KFGT2FCM.

Computation time improvement (speedup) =

$$\frac{Run\ time\ of\ (GT2FCM\ or\ KFGT2FCM)}{Run\ time\ of\ KGT2FCM} \quad (12)$$

B. Experimental Results

The 30% of Wine dataset which have been selected randomly is applied to the three algorithms while the target accuracy is assumed to be 66% for all three. The computation time of three algorithms are shown in Fig.4, Fig.5, Fig.10 and Fig.11.

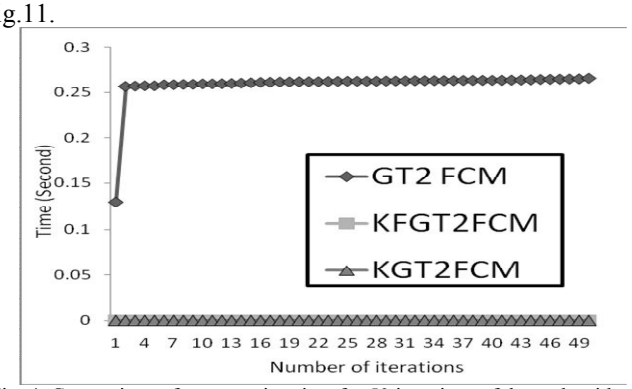

Fig. 4. Comparison of computation time for 50 iterations of three algorithms with a target accuracy of 66% for 53 data of Wine dataset

In another experiment, 70% of Iris dataset which have been selected randomly is applied to the three algorithms while the target accuracy is assumed to be 70% for all three algorithms. Comparing table II and table III, reveals that KGT2FCM outperforms GT2 FCM and KFGT2FCM significantly for low target accuracies. For the experiments performed for generating results of table II and III, 70% of each dataset which selected randomly, were used. However, for the experiments done for generating results of Fig.4 and 5, 30% of each dataset, which selected randomly, were exploited.

Since, in this paper we focus on computation time reduction and not accuracy, the target accuracies are selected close to the maximum accuracy.

TABLE II
COMPARING COMPUTATION TIME (IN SECONDS) OF THREE ALGORITHMS WITH LOW TARGET ACCURACY

Method	Iris Acc [a]: 50%	Wine Acc: 50%	Pima Indians Acc: 65%	Shuttle Acc: 70%
GT2FCM (in seconds)	0.2138	0.51	1.68e-5	2.32e-4
KFGT2FCM (in seconds)	9.76e-5	1.7e-5	1.72e-5	2.37e-4
KGT2FCM (in seconds)	**2.51e-5**	**1.5e-5**	**1.67e-5**	**2.31e-4**
Speedup of KGT2- FCM vs. GT2 – FCM	8518×	**34000×**	1.005×	1.004×
Speedup of KGT2- FCM vs. KFGT2FCM	**3.888×**	1.13×	1.03×	1.026×

[a] Target accuracy

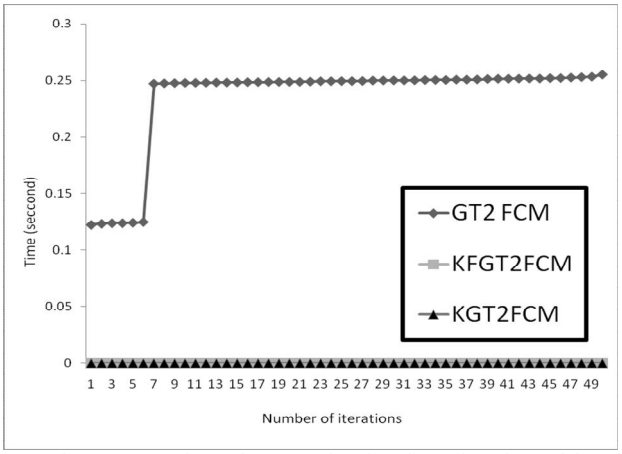

Fig. 5. Comparison of computation time for 50 iterations of three algorithms with a target accuracy of 70% for 105 data of Iris dataset

As illustrated in Fig.4 and Fig.5, the computation time of KFGT2FCM and KGT2FCM enhanced significantly compared to GT2 FCM. The average speedup of KGT2FCM compared to GT2 FCM and KFGT2FCM are 15880× and 1.004× respectively for Fig. 4 and are 15029× and 1.04×, for Fig. 5, correspondingly. Because the computation time of GT2 FCM is much larger than the two others, the differences of KFGT2FCM and KGT2FCM computation time is not seen in these figure. Therefore, two other Fig. 6 and 7, which are zoomed in of Fig. 4 and 5, have been added for comparing only KFGT2FCM against KGT2FCM.

The same as Fig.4 and 5, in Fig. 6 and 7 the results of KFGT2FCM and KGT2FCM are very similar. Therefore to clarify the difference of these clustering algorithms, in Fig. 7 and Fig.8, the same results are redrawn just for 35 iterations.

TABLE III
COMPARING COMPUTATION TIME OF THREE ALGORITHMS (IN SECONDS) ON DIFFERENT DATASETS WITH DIFFERENT TARGET ACCURACY

Method	Iris Acc [a]: 86%	Wine Acc: 71%	Pima Indians Acc: 68%	Shuttle Acc: 74.7%
GT2FCM (in seconds)	1.1	3.55	2.32e-4	**1.63e-5**
KFGT2FCM (in seconds)	**0.04**	0.278	2.33e-4	1.75e-5
KGT2FCM (in seconds)	0.11	**1.53e-5**	**2.30e-4**	1.67e-5
Speedup of KGT2- FCM vs. GT2 – FCM	10×	**232026×**	1.008×	0.97×
Speedup of KGT2- FCM vs. KFGT2FCM	0.36×	**18170×**	1.013×	1.05×

[a] Target accuracy

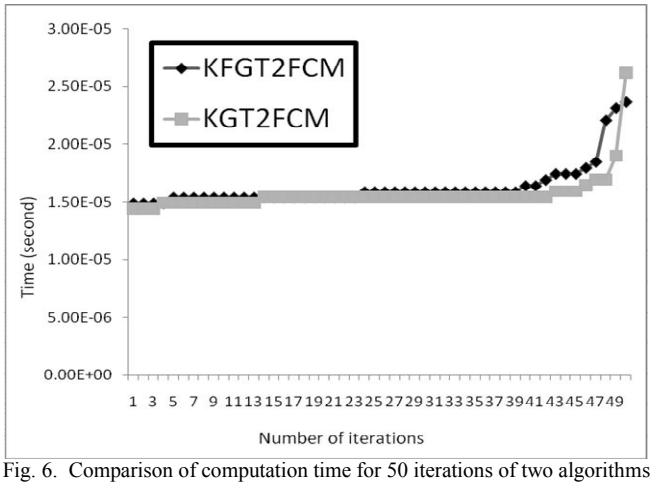

Fig. 6. Comparison of computation time for 50 iterations of two algorithms with a target accuracy of 70% for 105 data of Iris dataset

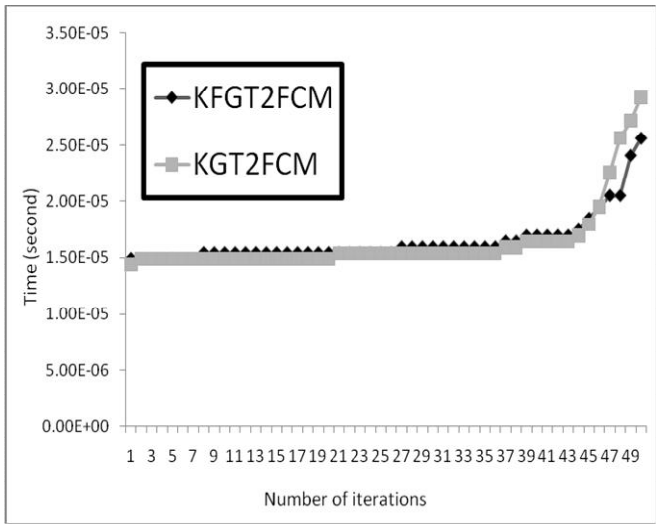

Fig. 7. Comparison of computation time between two algorithms to reach to the 66% accuracy in 50 times iterations for 53 data of Wine dataset

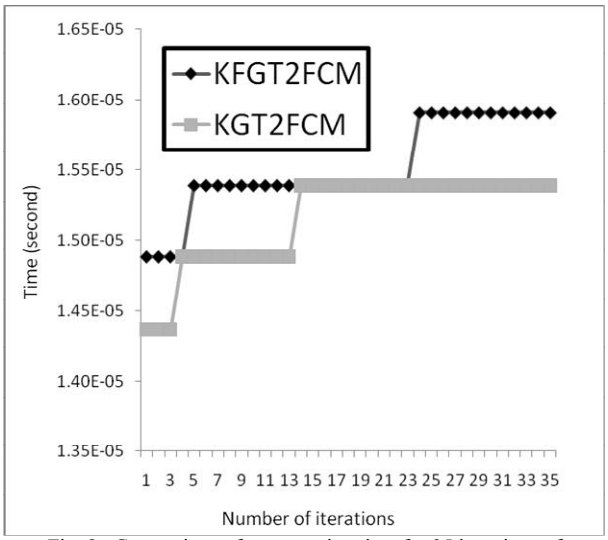

Fig. 8. Comparison of computation time for 35 iterations of two algorithms with a target accuracy of 70% for 105 data of Iris dataset

Fig. 10 shows the computation time for GT2 FCM, KFGT2FCM and KGT2FCM, when the accuracy is 60% and 30% of Pima Indians dataset is selected. The average computation time improvement of KGT2FCM compared to GT2 FCM and KFGT2FCM for this dataset are 32244× and 12911×, respectively.

The same experiments have been done using 70% of Iris dataset assuming 75% target accuracy. The results have been depicted in Fig. 11. The average computation time improvement of KGT2FCM compared to GT2 FCM and KFGT2FCM for this dataset are 11.68× and 0.55×, respectively. The reason for performance degradation of KGT2FCM compared to KFGT2FCM is due to the last five runs of KFGT2FCM which have long runtime.

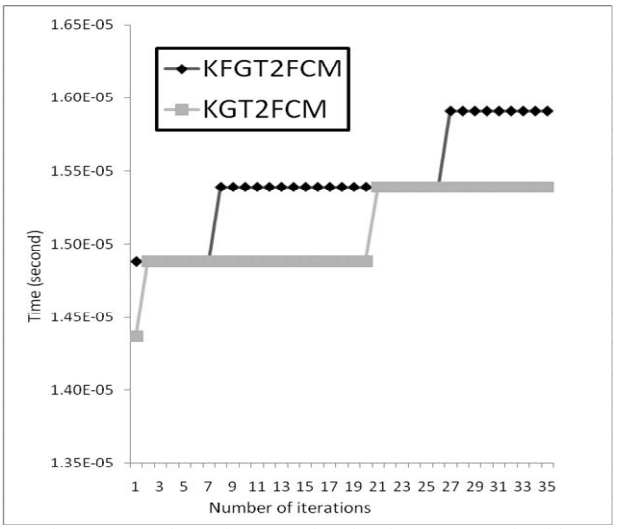

Fig. 9. Comparison of computation time between two algorithms to reach to the 66% accuracy in 35 times iterations for 53 data of Wine dataset

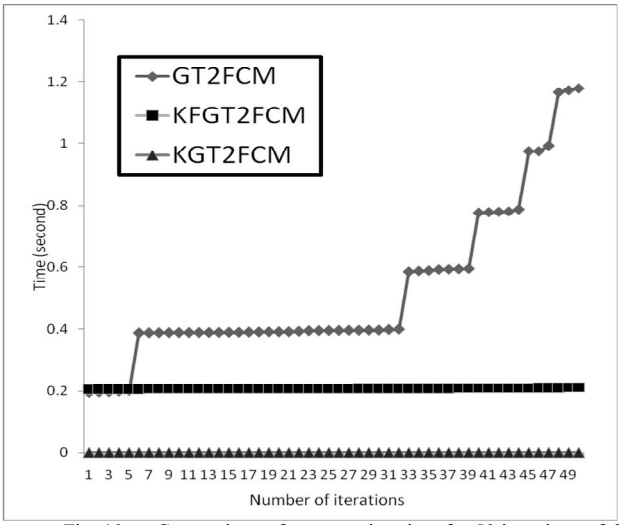

Fig. 10. Comparison of computation time for 50 iterations of three algorithms with a target accuracy of 60% for 230 data of Pima Indians dataset

Due to the large differences between execution time of the last 15 runs of KFGT2FCM and the other 35 runs, it seems that curves of KFGT2FCM and KGT2FCM are very close to each other for the first 35 runs. However, the real differences of KFGT2FCM and KGT2FCM computation time are not illustrated in Fig. 11, clearly.

To clarify this, Fig. 12, which is zoomed in of Fig. 11, has been added for comparing KFGT2FCM and KGT2FCM for the first 35 runs. Considering only the first 35 runs, the average execution time improvement of KGT2FCM compared to KFGT2FCM for this dataset is 1.05×.

The averages of computation time (in seconds) of 50 iterations of each algorithm on a specific dataset with specific target accuracy are available in table II and table III. Also, table II and III show the speedup of KGT2FCM compared to KFGT2FCM and GT2 FCM for each specific dataset with specific target accuracy. The input data used for these tables are 70% of data of each dataset that were selected randomly.

According to table II the maximum speedup for KGT2FCM vs. KFGT2FCM is 3.888× and for KGT2FCM vs. GT2 FCM is 34000× and for table III are 18170× and 232026×, respectively. Totally, we tried all of these datasets for five different accuracies. But, due to page limitation we only presented two groups of the results in tables II and III and the rest of the results were not presented. The average speedup for all of the results obtained for KGT2FCM vs. KFGT2FCM and GT2 FCM, are 1818× and 27456×, respectively.

According to table II and table III, KFGT2FCM obtains better results for Iris dataset. The reason is that the combination of k-means and FCM produces better initial centroids compared to k-means, which shortens GT2 FCM execution time. However, the combination of k-means and FCM produces worse initial centroids compared to the proposed method for Wine dataset which lengthens GT2 FCM execution time. The proposed method obtains better results for 80% of case studies used for tables II and III. The best results are bolded in table II and table III.

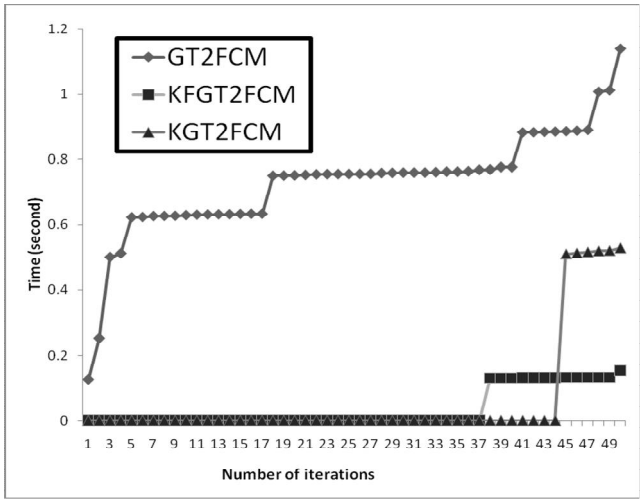

Fig. 11. Comparison of computation time for 50 iterations of three algorithms with a target accuracy of 75% for 105 data of Iris dataset

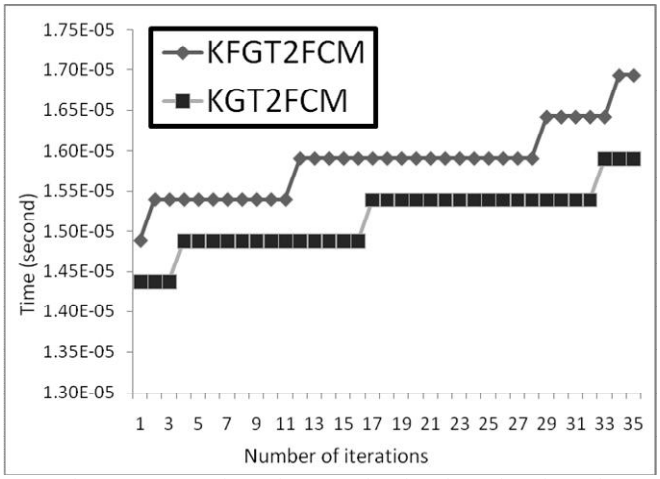

Fig. 12. Comparison of computation time for 35 iterations of two algorithms with a target accuracy of 75 % for 105 data of Iris dataset

In table III, for the Shuttle dataset, the GT2 FCM performs better compared to KFGT2FCM and KGT2FCM. The reason is that GT2 FCM reaches to the target accuracy, in the first iteration. So, in both KFGT2FCM and KGT2FCM algorithms, k-means and FCM algorithms which are used to initialize the centroid of GT2 FCM, results in longer computation time for KFGT2FCM and KGT2FCM. Therefore, these algorithms run slower than GT2 FCM. Also, in table III, for Iris dataset, KFGT2FCM is faster than KGT2FCM. The reason of this phenomenon is that four iterations from 50 iterations of KGT2FCM are very time consuming. The effects of these slow iterations causes the average time of KGT2FCM algorithm to be lower than KFGT2FCM. Considering k-means, in which its initial centroids are selected randomly, reveals that if the random centroids are not selected properly, it results in poor accuracy and therefore causes GT2 FCM to need more iteration and hence, makes KGT2FCM slower. The same problem happens in figures 6 and 7.

According to table II and III, although, in several cases KGT2FCM is slower than GT2 FCM and KFGT2FCM, the computation time differences of KGT2FCM in these cases is very small and are approximately 4.0e-7 seconds compared to GT2 FCM and KFGT2FCM.

IV. CONCLUSION

Recently, several works have used clustering and classification in sequential structures to improve the performance of classification algorithms. As they indicated, the efficiency of classification learning is enhanced if the input data is first clustered and then used for classification. However, there is a trade-off between computation time and accuracy of clustering algorithms. In this paper, a new clustering method is introduced to improve the computation time of a classification algorithm by preprocessing classification dataset. To address the conflict of high computation time and high accuracy of clustering algorithm, we propose a hybrid clustering algorithm called KGT2FCM.

This algorithm is a combination of high accuracy general type-2 fuzzy C-means (GT2 FCM) that can deal with uncertainty via using α-planes with low computation time k-means algorithm for input data preprocessing of classification algorithms. The proposed algorithm improves the speed of GT2 FCM and run on five datasets of UCI for clustering with different target accuracy.

KGT2FCM has better efficiency compared to GT2 FCM for almost all cases. The reason is that KGT2FCM produces better initial centroids for GT2 FCM. The average speedup of KGT2FCM compared to GT2 FCM on five datasets including Iris, Wine, Pima Indians, Shuttle and Magic, is 27456×.

In 86% of case studies; KGT2FCM is faster than KFGT2FCM. The reason is that for these cases combination of k-means and FCM takes long time to converge to good initial centroids for GT2 FCM, however KGT2FCM does not have the overhead of FCM. In the remaining 14% cases, because the combination of k-means and FCM produces better initial centroids for GT2 FCM, KFGT2FCM has better speedup compared to KGT2FCM. Finally, the average speedup of KGT2FCM compared to KFGT2FCM on five datasets including Iris, Wine, Pima Indians, Shuttle and Magic, is 1818×. In conclusion, the proposed method

(KGT2FCM) is faster than GT2 FCM and KFGT2FCM by 27456× and 1818×, respectively.

ACKNOWLEDGMENT

We thank Dr. Ondrej Linda from Idaho University, for his generosity and supporting us.

REFERENCES

[1] K. Funatsu and K. Hasegawa," New fundamental technologies in data mining". First published January, 2011.Printed in India. ISBN 978-953-307-547-1.

[2] R. Athauda, M. Tissera, C. Fernando. " Data Mining Applications: Promise and Challenges". Data Mining and Knowledge Discovery in Real Life Applications, ISBN 978-3-902613-53-0, pp. 438, February 2009, I-Tech, Vienna, Austria.

[3] R. Xu, D. Wunsch II," Survey of Clustering Algorithms". IEEE Transactions On Neural Networks, Vol. 16, No. 3, MAY 2005.

[4] O. Linda, M. Manic," General Type-2 Fuzzy C-Means Algorithm for Uncertain Fuzzy Clustering". Fuzzy Systems, IEEE Transactions. Feb 13 2012, ISSN : 1063-6706.

[5] A. K. Jain," Data clustering: 50 years beyond k-means". Pattern Recognition Letters 31 (2010) 651–666.

[6] M. H. Fazel Zarandi, I. B. Turksen, O. Torabi Kasbi," Type-2 fuzzy modeling for desulphurization of steel process". Expert Systems with Applications 32 (2007) 157–171.

[7] A. Shahi, R. Binti Atan and M-D. Nasir Sulaiman," An effective fuzzy c-mean and type-2 fuzzy logic for weather forecasting". Journal of Theoretical and Applied Information Technology 2005 - 2009 JATIT. Malaysia.

[8] Q. Liang and J. Mendel," Decision Feedback Equalizer for Nonlinear Time-Varying Channels Using Type-2 fizzy Adaptive Filters". Fuzzy Systems, 2000. FUZZ IEEE 2000.

[9] G. Zheng; *Jing Wang; Wengwei Zhou; Yong Zhang*, "A Similarity Measure between Interval Type-2 Fuzzy Sets". Proceedings of the 2010 IEEE, International Conference on Mechatronics and Automation, August 4-7, 2010, Xi'an, China. International Conference on 26-28 July 2011.

[10] V. Nouri, Mohammad-R. Akbarzadeh-T., Alireza Rowhanimanesh."General type-2 fuzzy clustering using hybrid of k-means and type-1 fuzzy clustering for data preprocessing" in Persian. 8th Symposium on Advances in Science and Technology. Dec 19, 2013.

[11] E. R. ;Pfahringer, B. ; Holmes, G. "Clustering for classification". Information Technology in Asia (CITA 11), 2011 7th International Conference on Digital Object Identifier: 10.1109/CITA.2011.5998839. Publication Year: 2011 , Page(s): 1 – 8.

[12] W Cai ;S. Chen ;D. Zhang. "A Multi-objective Simultaneous Learning Framework for Clustering and Classification". IEEE Transactions on Neural Networks, Volume: 21 , Issue: 2.Published in 2010.

[13] D. Wu , J. Mendel." Enhanced Karnik-Mendel Algorithms for Interval Type-2 Fuzzy Sets and Systems". Fuzzy Information Processing Society, 2007. NAFIPS '07. Annual Meeting of the North American.

[14] Yuvaraj, N.; Vivekanandan, P.;" An efficient SVM based tumor classification with symmetry Non-negative Matrix Factorization using gene expression data". International Conference on Information Communication and Embedded Systems (ICICES), 2013. 21-22 Feb. 2013.

[15] Borges, H.B. ; Nievola, J.C." Hierarchical classification using a Competitive Neural Network". Eighth International Conference on Natural Computation (ICNC), 2012. 29-31 May 2012.

[16] W. Yang ; K. Wang ; W. Zuo. "Prediction of protein secondary structure using large margin nearest neighbor classification". Advanced Computer Control (ICACC), 2011 3rd International Conference. 18-20 Jan. 2011.

[17] Swangnetr, M.; Kaber, D.B.;" Emotional State Classification in Patient–Robot Interaction Using Wavelet Analysis and Statistics-Based Feature Selection". IEEE Transactions on Human-Machine Systems (Volume: 43, Issue: 1). Jan. 2013.

[18] A. Frank, A. Asuncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Informatics and Computer Science.

[19] Der-Chen Lin, Miin-Shen Yang," A similarity measure between type-2 fuzzy sets with its application to clustering". Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007.

[20] Wen-liang Hung, Miin-shen Yang ," Similarity Measures Between Type-2 Fuzzy Sets". International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. Vol. 12, No. 6 (2004) 827-841 World Scientific Publishing Company.

[21] Miin-Shen Yang, Der-Chen Lin. "On similarity and inclusion measures between type-2 fuzzy sets with an application to clustering". Computers and Mathematics with Applications 57 (2009) 896_907.

[22] Hwang C.-M., Yang M.-S., Hung W.-L., "On similarity, inclusion measure and entropy between type-2 fuzzy sets". International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems 2012.

[23] Soumi Ghosh , Sanjay Kumar Dubey . "Comparative Analysis of K-Means and Fuzzy CMeans Algorithms". ((IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.4, 2013.