A New Fuzzy Belonging-based Multi-view K-means Clustering Algorithm

Junda Zhao
Department of Computer Science
University of Toronto
Toronto M5S1A1, Canada
drixs2050@gmail.com

Weiling Cai
College of Computer Science and Technology
Nanjing Normal University
Nanjing 210097, China
caiwl@nuaa.edu.cn

Yizhen Ge
College of Electronic and Information Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing 20016, China
geyizhen@nuaa.edu.cn

Lijuan Zhang
College of Electronic and Information Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing 20016, China
lijuan.zhang@nuaa.edu.cn

***Abstract*—Recently, a variety of data acquisition methods leads to the description of the same thing from different angles, which induces a large number of multi-view data. Clustering multi-view data in large-scale data analysis is of great significance and very challenging. In this paper, we present a new robust clustering algorithm, the Multi-view Fuzzy K-means clustering (MFKC) algorithm. In MFKC, a fuzzy membership method is used and optimized to describe the probability of a sample belonging to certain clusters. Compared with the hard partition based multi-view clustering algorithms, MFKC can effectively describe the fuzzy relationship between multi-view data and all clusters, and has better interpretability for the clustering results. In addition, MFKC also sets different weights for different views. By optimizing the multi-view clustering objective function, we obtain the optimal weigh of each view to reflect the importance of the view. The feasibility and effectiveness of the proposed algorithm is validated through various experimental comparisons.**

Keywords—Data Cluster, Fuzzy Belongness, Multi-View Data, K-Means Cluster

I. INTRODUCTION

In the past decades, with the dramatic growth of internet and related business, massive amount of data is generated from various internet activities, such as video websites, social networks, commercials, etc. Multiple types of data were collected, e.g., graphical data, textual data, and video data [1]. During the last year, about video of 300 hours is uploaded to YouTube every minute and 12 trillion photos are uploaded [2]. Such data usually possesses tremendous potential information or trends of internet. To gain useful and applicable information from this data, effective data analysis methods are essential to express the pattern and characteristic of the data. One of the most challenging problems in data analysis is clustering.

Clustering algorithms in [3] group data by analyzing the similarities between samples of the given datasets. However, choosing a suitable standard to measure the similarity can be crucial. In general, there are 5 main types of clustering methods, i.e., the partitioning, hierarchical, density-based, grid-based and model-based methods [4]. Among all the clustering algorithms, K-means clustering algorithm [5] is one of the most widely used clustering methods. As a partition method, K-means was proposed over 50 years ago. The algorithm finds a partition such that the squared error between the empirical mean of a cluster and minimizes the points in the cluster. Due to its low cost, the k-means algorithm is widely applied in large-scale data clustering. However, this algorithm is designed for single-view datasets.

Nowadays, large amount of datasets is collected in different views, which encourages the development of multi-view data analysis algorithms. Bickel and Scheffer [6] had proposed several multi-view modifications of classic algorithms applied in single-view clustering, including multi-view EM clustering, Multi-view spherical k-means, Multi-view Agglomerative clustering, etc. The results reflect a clear improvement on performance compared to the results of their single-view counterparts. Xiao *et al* [7] had also proposed an application of multi-view k-means algorithm on Big Data, which introduces a weight function to show the importance of each view.

The multi-view clustering algorithms usually had a better performance than the single-view clustering algorithms. However, these algorithms share a common problem. They use hard partition method to group the samples, which sometimes can be inaccurate. A hard partition algorithm defines one sample to a single cluster unequivocally. However, the similarity of such sample may not be as high as other samples in this cluster. This results in an non-ideal clustering. Bezdek *et al.* [8] proposed a function to produce a fuzzy partition that characterizes the membership of each sample point in all the clusters by a membership function. The membership ranges between zero and one. Membership close to unity signifies a high degree of similarity between the sample and the cluster. Whereas, memberships close to zero implies little similarity between the sample and the cluster.

This paper proposes a multi-view fuzzy k-means clustering (MFKC) algorithm. To give a clearer grouping of the data, the algorithm introduced a fuzzy membership function to represent the membership of a sample to a cluster. In MFKC, we also introduce a weight function for each view to show the importance of different views. By iteratively optimizing the target function, the MFKC algorithm keeps updating the weight of views, membership matrices of samples and centers of clusters until the algorithm convergent. Through this process we also get the alternately updating equations of these outputs. The experimental results of

978-1-7281-2530-5/19/$31.00 ©2019 IEEE

applying MFKC on the artificial, text, and graphic datasets demonstrate the feasibility and effectiveness of the proposed algorithm.

II. Proposed Multi-view Fuzzy K-means Clustering Algorithm

In this work, we describe the new multi-view fuzzy K-means clustering (MFKC) algorithm in detail. In MFKC, we construct the target function based on the fuzzy membership strategy and optimize the target function by considering several variables.

A. Target Function

Given a multi-view data set X with n samples and w views. We group the n samples into c clusters, i.e., $\{\Omega_1, \Omega_2, \ldots, \Omega_c\}$. Denote the samples of the v^{th} view by $x^{(v)}$, where v=1,2,...,w, $x^{(v)} \in \mathbb{R}^{d_v \times n}$, and d_v is the dimension of the v^{th} view. Let $m_j^{(v)}$, $m_j^{(v)} \in \mathbb{R}^{d_v \times c}$, j=1,2,...,c, be the centers of the clusters; u_{ij} be the membership of the i^{th} sample to the j^{th} cluster; θ be the coefficient that controls the fuzziness; $\alpha^{(v)}$ be the weight of the v^{th} view; and γ be the coefficient which controls the weight of different views.

In order to construct the target function, we consider two main aspects. Firstly, we describe the probability of the sample that belongs to one cluster through the fuzzy membership. Next, we show the importance of different views according to their weights. Thus, the target function can be express as follows,

$$\mathrm{J}_f = \underset{\alpha^{(v)}, m^{(v)}, U}{\operatorname{argmin}} \sum_{v=1}^{w} \sum_{j=1}^{c} \sum_{i=1}^{n} (\alpha^{(v)})^{\gamma} u_{ij}^{\theta} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2 \tag{1}$$

$$\text{s.t.} \begin{cases} \sum_{v=1}^{w} \alpha^{(v)} = 1 & \alpha^{(v)} \in [0,1] \\ \sum_{j=1}^{c} u_{ij} = 1, i=1,2,\ldots,n & u_{ij} \in [0,1] \end{cases} .$$

Based on (1), we know that: 1) if sample $x_i^{(v)}$ is in cluster Ω_j, the distance of $x_i^{(v)}$ to the center of Ω_j must be the shortest among the distances between $x_i^{(v)}$ and all the clusters; 2) the membership u_{ij} of $x_i^{(v)}$ to cluster Ω_j must be the largest among the membership to all the clusters and vice versa; 3) the weight factor should be larger for views that has more distinguishable clusters, and smaller for views containing more overlapping and confounding clusters.

B. Target optimization

Since there are three variables in (1), i.e., $m_j^{(v)}$, $\alpha^{(v)}$, and u_{ij}, we optimize the target function in three steps. We solve one variable each time and consider the other variable as constant.

Step 1: Solving $m_j^{(v)}$ when $\alpha^{(v)}$ and u_{ij} are fixed.

We fix $\alpha^{(v)}$, u_{ij}, and do the partial derivation with respect to $m_j^{(v)}$ of (1), i.e.,

$$\frac{\partial \mathrm{J_f}}{\partial m_j^{(v)}} = 0. \tag{2}$$

With $\sum_{i=1}^{n} u_{ij}^{\theta}(x_i^{(v)} - m_j^{(v)}) = 0$, we have

$$m_j^{(v)} = \frac{\sum_{i=1}^{n} u_{ij}^{\theta} x_i^{(v)}}{\sum_{i=1}^{n} u_{ij}^{\theta}}. \tag{3}$$

Step 2: solving $\alpha^{(v)}$ when $m_j^{(v)}$ and u_{ij} are fixed.

Let

$$\mathrm{H}^{(v)} = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{\theta} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2. \tag{4}$$

Similarly, we apply constraint $\sum_{v=1}^{i} \alpha^{(v)} = 1$, and use Lagrange multiplier to obtain a new target function as

$$\mathrm{J}_{\alpha} = \underset{\alpha^{(v)}, m^{(v)}, U}{\operatorname{argmin}} \sum_{v=1}^{w} (\alpha^{(v)})^{\gamma} \mathrm{H}^{(v)} - \lambda\left(\sum_{v=1}^{w} \alpha^{(v)} - 1 \right). \tag{5}$$

Let

$$\begin{cases} \frac{\partial \mathrm{J}_{\alpha}}{\partial \alpha^{(v)}} = 0 \\ \frac{\partial \mathrm{J}_{\alpha}}{\partial \lambda} = 0 \end{cases} , \tag{6}$$

we have

$$\alpha^{(v)} = \frac{(\mathrm{H}^{(v)})^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{w} (\mathrm{H}^{(v)})^{\frac{1}{1-\gamma}}} . \tag{7}$$

Step 3: Solving u_{ij} when $\alpha^{(v)}$ and $m_j^{(v)}$ are fixed

Applying $\sum_{j=1}^{c} u_{ij} = 1$, we obtain a new target function through Lagrange multiplier：

$$\mathrm{J}_u = \sum_{v=1}^{w} \sum_{j=1}^{c} \sum_{i=1}^{n} (\alpha^{(v)})^{\gamma} u_{ij}^{\theta} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2 - \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{c} u_{ij} - 1 \right). \tag{8}$$

With $\frac{\partial \mathrm{J}_u}{\partial u_{ij}} = 0$, we have

$$u_{ij} = \left(\frac{\lambda_i}{\theta} \right)^{\frac{1}{\theta-1}} \sum_{v=1}^{w} \left(\frac{1}{(\alpha^{(v)})^{\gamma} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2} \right)^{\frac{1}{\theta-1}} . \tag{9}$$

With $\frac{\partial \mathrm{J}_u}{\partial \lambda_i} = 0$, we have

$$\sum_{j=1}^{c} u_{ij} - 1 = 0 \tag{10}$$

Substitute u_{ij} into (10), the following equation obtains.

$$\left(\frac{\lambda_i}{\theta} \right)^{\frac{1}{\theta-1}} = \frac{1}{\sum_{j=1}^{c} \sum_{v=1}^{w} \left(\frac{1}{(\alpha^{(v)})^{\gamma} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2} \right)^{\frac{1}{\theta-1}}} \tag{11}$$

Substitute (11) back into (9), we obtain u_{ij} as follows.

$$u_{ij} = \frac{\sum_{v=1}^{w} \left((\alpha^{(v)})^{\gamma} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2 \right)^{-\frac{1}{\theta-1}}}{\sum_{j=1}^{c} \sum_{v=1}^{w} \left((\alpha^{(v)})^{\gamma} \left\| x_i^{(v)} - m_j^{(v)} \right\|_2^2 \right)^{-\frac{1}{\theta-1}}}. \tag{12}$$

C. Algorithm of MFKC

More specifically, Algorithm 1 gives the pseudocode of the proposed MFKC method. We first obtain $m_j^{(v)}$ via *Step 1*, then update $\alpha^{(v)}$ according to *Step 2*. Next, we update U via *Step 3*. Then, we repeat this process iteratively until convergence.

Algorithm 1. Algorithm of MFKC

Input: Data set X=$\{x^{(1)}, x^{(2)}, \ldots, x^{(w)}\}$ with w views. Let c be the clusters, θ be the coefficient of the fuzziness, γ be the weight coefficient of the views.

Output: Membership matrix U, center of clustering $m_j^{(v)}$ (j=1,2,…,c, v=1,2,…,w), weight of views $\alpha^{(v)}$ (v=1,2,…,w).

Initialization:

1. Generate matrix U with $\sum_{j=1}^{c} u_{ij}=1$ randomly.

2. Initialize the weight of each view as $\alpha^{(v)}=1/w$, v=1, 2, ..., w.

3. **for** v=1:w

4. **for** j=1:c

5. Update the center of the cluster $m_j^{(v)}$ according to (3).

(*to be continued…*)

6. **End for**

7. Update the weight of view $\alpha^{(v)}$ according to (7).

8. **End for**

9. Update the membership matrix U according to (12).

Repeat 3-8 until convergence.

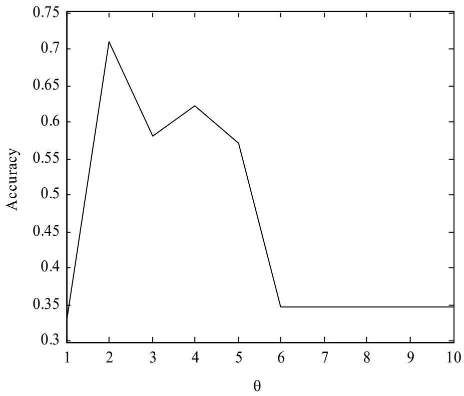
Return U, $m_j^{(v)}$ and $\alpha^{(v)}$.

III. Performance Analysis

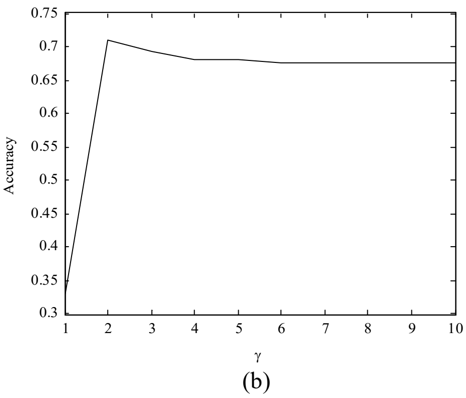
In this section, we first analyze the effect of two coefficients on the accuracy of MFKC, i.e., the fuzziness coefficient θ and the weight of views coefficient γ. Next, the time complexity of MFKC is analyzed theoretically. Then some advantages and limitations of our proposed MFKC algorithm is discussed.

A. Effect of coefficients θ and γ

In MFKC, we use two important coefficients θ and γ. θ controls the fuzziness and γ controls the weight of views. Figs. 1(a) and 1(b) illustrate the effects of these two coefficients on the accuracy of our algorithm, respectively.



(a)



(b)

Fig. 1. Effects of the coefficients on the accuracy of MFKC: (a) effect of θ; (b) effect of γ.

As is shown in Fig. 1(a), when θ>2 the accuracy of MFKC decreases as the fuzziness coefficient θ increases. When θ=1 or θ>6, the accuracy is the lowest, and when θ=2, the accuracy is the highest. In Fig. 1(b), when the weight of view coefficient γ=1, the accuracy of MFKC is the lowest. When γ<2, the accuracy of MFKC slowly decreases. When γ=2, the accuracy is the highest. Therefore, we set θ=2 and γ=2 in the subsequent experiments to obtain the optimal performance.

B. Time complexity

Assume that the data set contains n samples and w views, the number of dimensions of each view is $d^{(v)}$. Let t be the number of times the algorithm had iterated, and c be the number of clusters. Firstly, we analyze the time complexity of updating the center of the clusters. As is given in Section II-B, (3) updates the center of clustering and contains the membership matrix U and the samples $x_i^{(v)}$ in different views. The time complexity of updating the center of cluster $m_j^{(v)}$ in the v^{th} view is $O(ncd^{(v)})$. Since, there are w views, the total time complexity should be $O(ncD)$, $D=\sum_{v=1}^{w} d^{(v)}$.

Secondly, we analyze the time complexity of updating the membership matrix U. Since (12) contains the weight of each views $\alpha^{(v)}$, the sample $x_i^{(v)}$, and the center of cluster $m_j^{(v)}$, the time complexity of updating the membership matrix should be $O(ncD)$. At last, we analyze the time complexity of updating the weight of views $\alpha^{(v)}$. Equation (6), which updates the weight of views, contains the sample $x_i^{(v)}$ in the v^{th} view, the center of clustering $m_j^{(v)}$, and the membership matrix U. So the time complexity of updating the weight of view $\alpha^{(v)}$ should be $O(ncd^{(v)})$. Since there are w views in total, so the total complexity of updating all weights of views should be $O(ncD)$. Though iterating t times, the total time complexity of the algorithm is $O(tncD)$.

C. Advantages and limitations of MFKC

With the fuzzy membership and optimization strategies, MFKC has two main advantages when compares with previous works.

1) MFKC can effectively express the importance of different views by using the weight factor. From (7) we know that the smaller the sum of variances between the samples and the center of the clusters calculated from the membership matrix, the larger the weight of view, otherwise the weight of view is smaller.

2) MFKC can better model the real world. Based on the soft partitioning method, the proposed algorithm is more accurate than the hard partitioning methods. More specifically, we give an example in Fig. 2 to explain this advantage.

In Fig. 2, the green and red dots represent 2 different clusters. The center of the cluster with green dots is (-3,0) and the center of the cluster with red dots are (3,0). The positions of dots X and Y are (0,0) and (0,4), respectively. If we try to put X, Y in to the 2 clusters, we would find that the distance from either of the points to the centers of the clusters with green and red dots are the same. For hard partitioning algorithms, X, Y must be in green cluster or red cluster. Whereas, our fuzzy algorithm sets the probability of X, Y in both clusters to be 0.5, which is more reasonable.

There are also some limits of our algorithm. Similar to K-means and FCM algorithms, the MFKC algorithm requires a

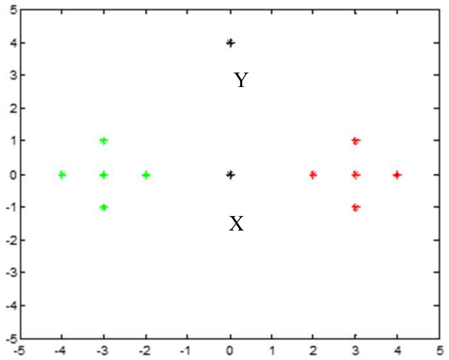


Fig. 2. Fuzzy clustering.

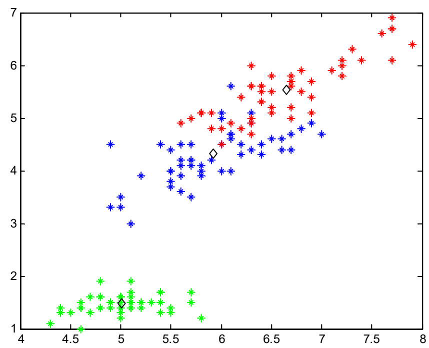
given number of clusters. However, in real life the number of clusters is usually hard to determine. Also, the coefficient of weight and membership must be pre-determined. Finally, MFKC might give a local optimal solution instead of an overall optimal solution. Nevertheless, this can be resolved by running the algorithm multiple times.

IV. EXPERIMENTAL RESULTS

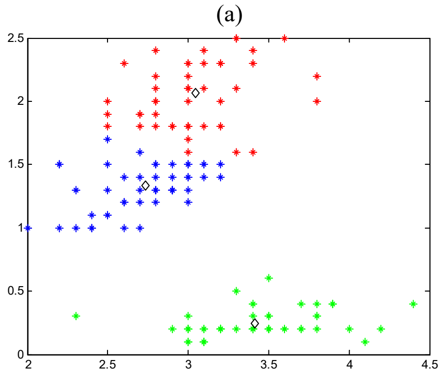
In this section, we evaluate the performance of the proposed MFKC algorithm. In the evaluation, two types of datasets are used: 1) one modified Iris artificial dataset; 2) five multi-view datasets, including 3sources, ads, mfeat, cora, and webKB. We compare the performance of MFKC with four benchmark algorithms, i.e., K-means, Fuzzy C-means (FCM), Multi-view k-means (MKmeans), discriminative embedded clustering (DEKM).

A. Artificial datasets

In this subsection, we evaluate the performance of MFKC with a modified Iris dataset. Collected by the British statistician and biologist Fisher, the Iris dataset consists of 150 samples, 3 types and 4 properties (i.e., the length of calyx, width of calyx, length of petal, and width of petal).



(a)



(b)

Fig. 3. Clustering resluts of the artificial data set: (a) resluts of view 1; (b) resluts of view 2.

We use the lengths of calyx and petal as view 1, and the widths of calyx and petal as view 2. The clustering results of the two views are given in Fig. 3, where the red, blue and green dots represent three different clusters, and the center of each cluster is marked with diamond. As can be observed, the proposed MFKC algorithm can effectively cluster the given data.

Fig. 4 gives the loss of target function when the number of iterations ranges from 1 to 16. As can be observed, the loss of the target function decreases as the number of iterations increases and keeps as 24 when the number of iterations is greater than 6.

Fig. 5 demonstrate clustering accuracy when the number of iterations ranges from 1 to 16. As is shown, the clustering accuracy increases with the number of iterations. The accuracy achieves 0.94 when the number of iterations is 5, which demonstrates the effectiveness of MFKC.

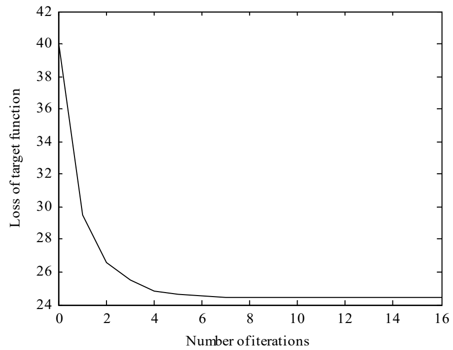


Fig. 4. Loss of target function.

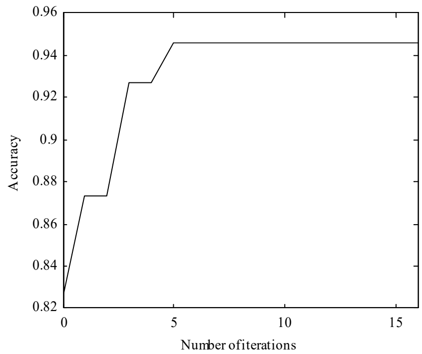


Fig. 5. Clustering accuracy

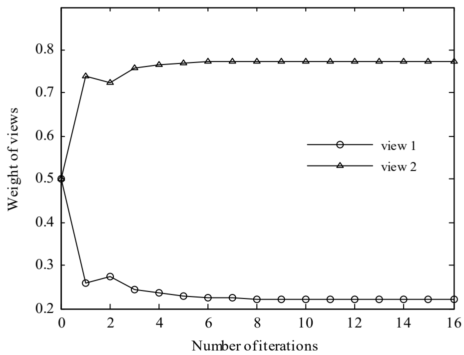


Fig. 6. Weight of each view.

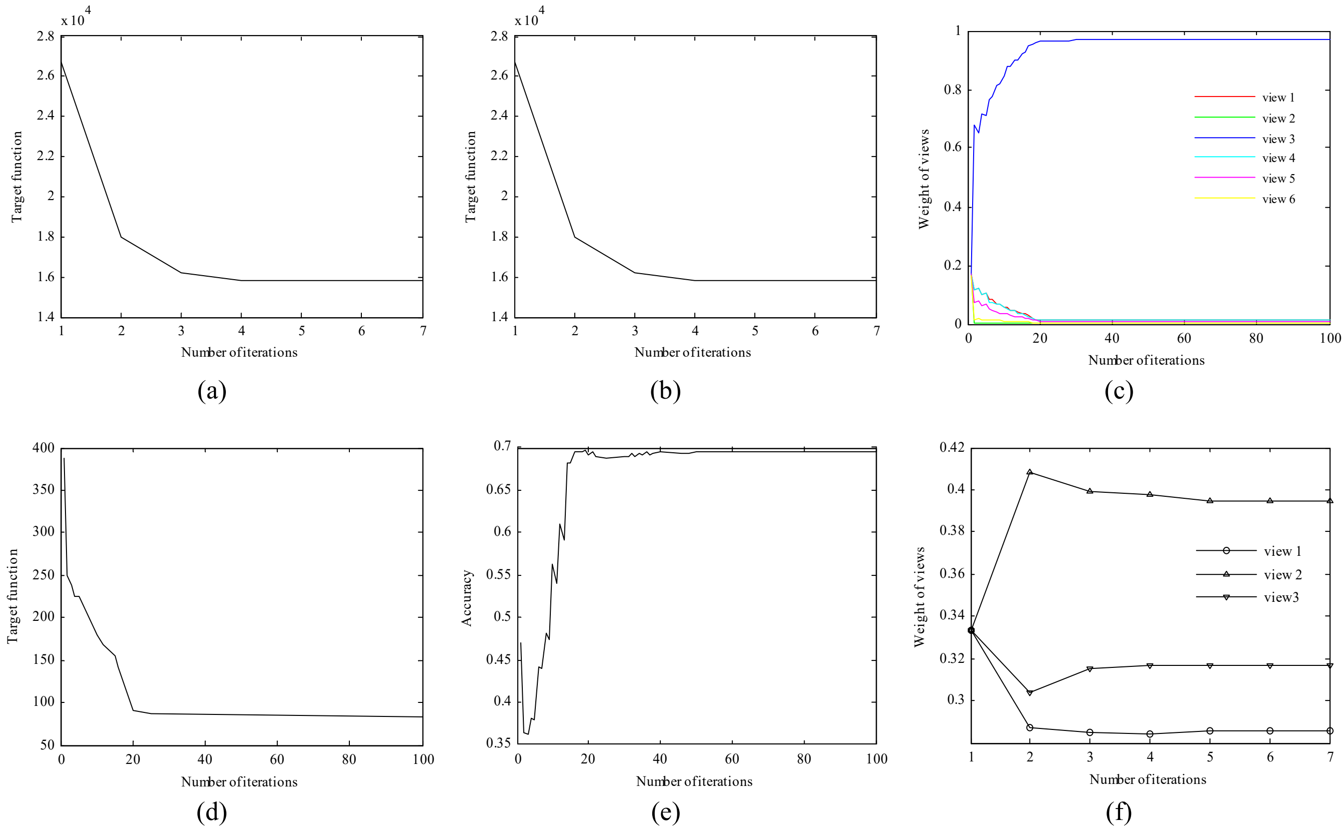


Fig. 7. Data clustering results: (a) Loss of target function on the *3sources* dataset; (b) Clustering accuracy on the *3sources* dataset; (c) Weight of each view on the *3sources* dataset; (d) Loss of target function on the *mfeat* dataset; (e) Clustering accuracy on the *mfeat* dataset; (f) Weight of each view on the *mfeat* dataset.

Moreover, we also illustrate the relationship between the weight of view and the number of iterations in Fig. 6. In the test dataset, we use two views. The sum of the weights of views 1 and 2 should equal to 1, which is validated in Fig. 6. We can observe that the weight of view 1 increases when the weight of view 2 decreases. The weights of views 1 and 2 become stable at around 0.22 and 0.78, respectively. The reason why the weight of view 2 is much larger than view 1 can be found in the cluster results in Fig. 2. As can be observed from Fig. 2, the red and blue clusters overlap in view 1 but not in view 2. Clusters in view 2 are well-separated, which means view 2 is better for clustering and should have a higher weight than view 1.

B. *Multiview datasets*

In this subsection, we first evaluate the performance of MFKC in two typical multi-view datasets, i.e., 3sources and mfeat. In the evaluation, we first consider the loss of target function, clustering accuracy, and weight of views with the number of iterations. Then, we extend the experimental datasets to five typical multi-view datasets, including the 3sources, ads, mfeat, cora, and webKB data sets, and compare the performance of MFKC with four benchmark clustering algorithms: K-means, Fuzzy C-means (FCM), Multiview k-means (MKmeans), discriminative embedded clustering (DEKM). For k-means and FCM, we connect the feature in different views in series and cluster them as one view. The adopted five multi-view datasets are given as follows.

- *3sources dataset*: This is a text dataset collected from 3 sources, i.e., BBC, Reuters and The Guardian. It includes 948 new articles. There are 6 topic labels for the dataset: business, entertainment, health, politics, sport, technology. The dataset contains 3 views which are put in 3 different data formats.
- *ads dataset*: This set contains textual and graphical data collected from internet. There are only 2 types in the dataset, non-ad and ad. This dataset consists of 5 different views.
- *mfeat dataset*: This is a dataset consisting of handwritten digits (0–9) with a total number of 2000 samples in 6 different views. There are a total number of 10 different groups in the dataset. The goal of using this dataset is to predict which number the sample is.
- *cora dataset*: This dataset contains a variety of machine learning related papers. The samples are in 7 different groups: Case Based, Genetic algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, and Rule Learning. It contains 2 different views.
- *webKB dataset*: This dataset includes multiple web pages, which are collected from 4 universities. There are 5 different types of samples: course, faculty, student, project, and staff. The dataset contains 2 different views.

TABLE I. **TABLE 1.** ACCURACY OF DIFFERENT ALGORITHMS ON DIFFERENT DATASETS.

	Kmeans	FCM	DEKM	MKmeans	MFKC
ads	**0.8381**	0.6923	0.7721	<u>0.8106</u>	0.7475
3sources	0.4260	<u>0.6330</u>	0.3550	0.5562	**0.7041**
mfeat	0.5485	0.5475	**0.6980**	0.6285	<u>0.6545</u>
cora	<u>0.4086</u>	0.3177	0.2225	0.3735	**0.4604**
webKB1	<u>0.4395</u>	**0.5096**	0.3758	0.4076	0.4076
webKB2	<u>0.6280</u>	0.4879	0.4686	0.6087	**0.6618**
webKB3	0.5867	0.4133	0.5400	<u>0.5933</u>	**0.6067**
webKB4	**0.6697**	<u>0.6786</u>	0.4706	0.4977	0.5566

1) Clustering results

The performance of the proposed KFMC algorithm on two representative datasets, *3sources* and *mfeat*, is given in Fig 7. The performance is evaluated in three aspects, i.e., the loss of target function, accuracy and weight of views, and the number of iterations ranges from 1 to 7 on the *3sources* dataset and from 1 to 100 on the *mfeat* datasets.

As can be observed from Figs. 7(a) and 7(d), with the increase of the number of iterations, the loss of target function decreases and stabilizes at a certain value. In Figs. 7(b) and 7(e), the accuracy increases as the number of iterations increases. In Figs. 7(c) and 7(f), the weights of different views keep changing until stabilize in the end. For the *3sources* dataset, view 2 has the highest weight and have the highest impact on the cluster result. For the *mfeat* dataset, view 3 has much higher weight than other views, meaning that view 3 affects the cluster result the most.

In general, the performance of the proposed MFKC algorithm approaches convergence. Take the results of the 3scources dataset as an example, the accuracy reaches the highest value after 6 iterations, which means the MFKC algorithm approaches convergent fairly quickly.

2) Clustering accuracy

Finally, the accuracies of MFKC is compared with four benchmark algorithms, i.e., Kmeans, FCM, DEKM and MKmeans, on all the aforementioned datasets. The comparison results are list in Table I, where the bold value in each line refers to the highest accuracy that the comparative algorithms can achieve for the used dataset, the underlined value is the second highest accuracy. As is shown in Table 1, comparing with other algorithms, the proposed MFKC has the highest accuracy in *3sources*, *cora*, *webKB2*, and *webKB3*, and the second highest accuracy on *mfeat*. This demonstrates that MFKC shows better performance than other comparative algorithms in most datasets. This is because k-means and FCM obtain information in a single-view form. DEKM and MKmeans uses hard partition methods.

V. Conclusion

In this paper, we proposed a robust multi-view clustering algorithm, the Multi-view fuzzy K-means clustering algorithm. The algorithm can effectively describe the probability of a sample belonging to certain clusters. Though deep analysis of the coefficients of fuzziness and weight of view, we obtained the optimal parameter settings. The performance of the proposed MFKC algorithm, in terms of the loss of target function, accuracy and weight of views on the artificial and multi-view datasets, is effectively evaluated. We also validated the feasibility and effectiveness of our algorithm through various experimental comparisons.

References

[1] C. Li, S. Wang, L. He, P. S. Yu, Y. Liang, and Z. Li, "SSDMV: Semi-Supervised Deep Social Spammer Detection by Multi-view Data Fusion", in *Proc. 11th Intl. Conf. Data Mining (ICDM)*, Singapore, Singapore Nov. 2018

[2] R. Sowmya and K. R. Suneetha, "Data mining with big data", in *Proc. 11th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, Inda, Jan. 2017, pp. 246-250

[3] I. Zheliznyak, Z. Rybchak, I. Zavuschak, "Analysis of Clustering Algorithms", *Advances in Intelligent Systems and Computing*, pp. 305-314, Springer, Cham (2017)

[4] B. Wu, B.M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise", *IEEE Transactions on Industrial Informatics*, 13(4), pp. 1620-1628, Apr. 2017

[5] A. Boukhdhir, O. Lachiheb, and M. S Gouider, "An improved MapReduce Design of Kmeans for clustering very large datasets," In *Proc. 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, Marrakech, Morocco, Nov. 2015, pp. 1-6.

[6] S. Bickel, T. Scheffer, "Multi-view clustering", In *Porc. IEEE International Conference on Data Mining (ICDM)*, Brighton, UK, Nov. 2004, vol. 4, pp. 19-26

[7] X. Cai, F. Nie, H. Huang, "Multi-view k-means clustering on big data", in *23rd International Joint conference on artificial intelligence*, Beijing China, Aug. 2013, pp. 2598-2604

[8] J.C. Bezdek, M. Moshtaghi, T. Runkler, C. Leckie, "The generalized c index for internal fuzzy cluster validity", *IEEE Transactions on Fuzzy Systems*, 24(6), pp. 1500-1512, 2016