A new approach for efficient clustering using fuzzy prototypes with varying neighborhoods

K.Mrudula[1*] and T. Hitendra Sarma[2]

[1] G. Narayanamma Institute of Technology and Science, Hyderabad, India
[2] Vasavi College of Engineering, Hyderabad, India,
mrudulasarma22207@gmail.com

Abstract. It is highly desirable to perform the clustering for large datasets more efficiently by finding the approximate clustering results in a reduced time. PFCM, PKFCM-F, and PKFCM-K are recent attempts to improve the efficiency of the traditional FCM, KFCM-F, and KFCM-K algorithms using fuzzy prototypes. Here each prototype represents all the data items in its ϵ neighborhood and the parameter ϵ highly influences the overall performance. Further, it is not possible to determine the optimal value of ϵ beforehand. This article presents a simple and practical approach to finding the ϵ-neighborhood of each prototype on the fly. Empirical results are presented to establish the efficiency of the proposed approach on several publicly available data sets.

Keywords: Prototype, Epsilon, Neighborhood, Fuzzy C-Means, Kernel FCM-F, Kernel FCM-K

1 Introduction

Clustering deals with the process of identifying groups in the data. In fuzzy clustering, each data item attains a membership value associated with each cluster, called *the fuzzy membership* [2][5]. The fuzzy memberships are essential in case of handling the natural phenomenon having smooth overlap of class distributions where some data items belong to one or more classes *i.e., with multi-class belongingness* [4]. The objective of the popular Fuzzy C-Means *FCM* clustering algorithm is to assign each data item to closest cluster center considering its fuzzy memberships in each cluster. The traditional FCM performs well if the clusters are spherical or convex in shape and there is no overlap of the clusters in the data. But, in most of the cases, the clusters in the data are non-convex shaped and non-linearly separable. Kernel versions of FCM are able to handle the later case effectively by exploring the advantage of linear separability in the high-dimensional feature space by the implicit non-linear mapping of data items in the input-space [13]. Fuzzy clustering methods have been widely applied in many areas including networks, [16], medical image analysis [18], [7], remote sensing [17], and climate analytics [19] and many others [6]. There are two kernel based fuzzy clustering approaches *viz.,* Kernel FCM-F and Kernel FCM-K

© The Author(s) 2023
C. Kiran Mai et al. (eds.), *Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACECS 2023)*, Atlantis Highlights in Computer Sciences 18,
https://doi.org/10.2991/978-94-6463-314-6_26

[3]. The major difference between these two approaches is the process of computing the cluster centers. Kernel FCM-F identifies the center of each cluster in the input space. On the other hand, the Kernel FCM-K identifies the centers in the induced space [3]. FCM and Kernel FCM-F algorithms have linear time complexity whereas the Kernel FCM-K has the quadratic time complexity w.r.t the size of the data set.

The key idea of the article is to use strategically selected leaders or prototypes from the given data and use them in the process of clustering. The clustering results on the sample data are then generalized to the whole data to get the final clustering result. Experimentally, it is proved that the prototype-based methods of FCM, KFCM-F [9] and KFCM-K [10] achieved relatively better accuracy in less running time. In all these approaches prototype is a data item representing a set of data items within its ϵ-neighborhood and the parameter ϵ has a high impact on the overall performance. A larger value of ϵ results in a great reduction in running time, but poor clustering quality; on the other hand, smaller the value of ϵ results less reduction in running time, but clustering results are very close to them that are achieved with the whole data. Hence it is crucial to fix the value of ϵ, but there is no straightforward approach to find the optimal value for ϵ. We call this problem as *neighborhood fixing problem.*

The objective of this article is to present a simple and efficient approach to overcome the *neighborhood fixing problem*, by finding the ϵ-neighborhood of each prototype on the fly. Empirical results are presented to establish the efficiency of the proposed method on several benchmark data sets from popular open source data sets *viz.,* Iris, Pendigits, OCR, LIR, and Shuttle.

The paper is organized as follows: Section 2 provides the details of the contemporary works in the literature. The main contribution of the article *i.e.,* finding the fuzzy prototypes with varying neighbourhoods is presented in section 3, experimental study is presented in section 4 and the conclusions are discussed in 5.

2 Fuzzy prototypes and related work

This section provides an overview of selecting prototypes and the recent improvements over the conventional clustering methods using these prototypes.

Prototype: Let C be the number of clusters to partition a data set $\mathcal{D} = \{x_1, x_2, ..., x_N\}$. A prototype Q_j is a representative of a small group-let of x_is within a neighborhood, say ϵ, such that

$$Q_j = \{x_i / ||Q_j - x_i|| \leq \epsilon\} \tag{1}$$

Fuzzy Prototypes: In order to speed-up the fuzzy clustering methods, fuzzy prototypes have been introduced [10]. The set of Fuzzy prototypes, say s, denoted by $\mathcal{Q} = \{Q_1, Q_2, ..., Q_s\}$, be selected over the data set using eq(1). In fuzzy prototypes, every data item x_i may be lying in the neighborhood of one

or more prototypes (Q_j). The membership of x_i in Q_j is denoted by $\mu_{Q_j x_i}$ and it is calculated using the below formula

$$\mu_{Q_j x_i} = \frac{1}{\sum_{k=1}^{s} \left(\frac{d(Q_j, x_i)}{d(Q_k, x_i)}\right)^{\frac{2}{m-1}}} \tag{2}$$

Here m is called the fuzzyness index.

The total of memberships of each data item in all prototypes (it belongs to) is always equal to 1 *i.e.,* $\sum_j \mu_{Q_j x_i} = 1$.

Prototype based approaches have been used to speed up the clustering process of hard k-means [11,14] and kernel k-means [15,13,12] clustering methods.

Mrudula*et.al.,* [8] applied the fuzzy prototypes to speed-up the conventional FCM approach, the kernel versions of FCM, *viz.,* Kernel FCM-F and Kernel FCM-K [3] and achieved relatively better accuracy in less running time as presented in [9] and KFCM-K [10]. In all these algorithms, the parameter ϵ highly influenced the overall performance, irrespective of hard clustering or soft clustering. Larger the value of ϵ results in great reduction in running time, but poor clustering quality; on the other hand, smaller the value of ϵ results in less reduction in running time, but clustering results are very close to that achieved with the whole data. Hence it is crucial to fix the value of ϵ, but there is no candid approach to find the optimal value ϵ. Further, it is also proved that these prototype-based clustering processes will converge, but are not guaranteed to reach global optimum, similar to the case of clustering with whole data.

3 Prototypes with varying neighbourhoods

This section presents a simple and efficient approach to address the *neighborhood fixing problem*, where the neighborhood size is identified on the fly based on the data under processing.

Let $\mathcal{D}$ be the given data set. Following the prototype generation as described in Section 2, we put a randomly selected data item x_i as the initial prototype, say Q_1 and identify a small cluster, $G(l_1)$.

$$G(Q_1) = \{x_j / ||(x_j) - (Q_1)|| \leq \epsilon_1\} for 1 \leq j \leq N. \tag{3}$$

such that Q_1 is a representative of few data items that lies within ϵ_1 neighbourhood.

Second prototype will be identified if we encounter a data item x_k such that $||(Q_1) - (x_k)|| > \epsilon_1$. x_k becomes the new prototype and the neighborhood of x_k denoted by ϵ_2 which is to be determined as follows.

$$\epsilon_2 = \min_{dist}\{dist(x_m, x_k)\} \tag{4}$$

for all x_m in $G(Q_i)$. It can be easily observed that, $\epsilon_2 \lneq \epsilon_1$.

The process continues until to get all the data items in $\mathcal{D}$ partitioned into $\{G(Q_1), G(Q_2), \ldots, G(Q_s)\}$. The set of prototypes obtained is $\{Q_1, Q_2, , , , , Q_s\}$ with neighbourhoods $\{\epsilon_1, \epsilon_2, ..., \epsilon_s\}$ such that $\epsilon_1 \gtrsim \epsilon_2 \gtrsim \epsilon_3 \gtrsim ..... \gtrsim \epsilon_s$.

An illustration of the prototypes with varying neighborhoods is presented in Figure 1.

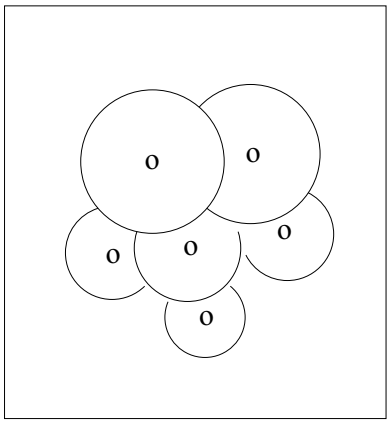

Fig. 1. Prototypes with varying neighborhoods

Finally, the membership of each data item x_i for each prototype Q_j is calculated as follows.

$$\mu_{Q_j x_i} = \frac{1}{\sum_{j=1}^{s} \left(\frac{d(Q_j, x_i)}{d(Q_k, x_i)}\right)^{\frac{2}{m-1}}} \tag{5}$$

where m is called the fuzziness index and i varies from 1 to N and j varies from 1 to s.

3.1 FCM, Kernel FC Means-F and Kernel FC Means-K using prototypes of varying neighborhoods

The prototype based FCM and Kernel FCM-F [3] are presented in [9] and Kernel FCM-K in [10] with fixed neighborhoods. As an improvement, here we use fuzzy prototypes with varying neighborhoods as generated using the above process.

In the above process, there is a two-level membership; the membership value, μx_i for each i in each Q_j (as shown in Equation 5) and the membership of each prototype Q_j in each cluster C_k which can be computed as follows.

$$\mu_{C_k Q_j} = \frac{1}{\sum_{k=1}^{K} \left(\frac{d(C_k, Q_j)}{d(C_i, Q_j)}\right)^{\frac{2}{m-1}}} \tag{6}$$

where j varies from 1 to s and k varies from 1 to K.

The final membership value of every data item *i.e.,* μx_i in a cluster C_k, $k = 1, 2, \ldots, K$ is approximated using $\mu_{Q_j x_i}$ and $\mu_{C_k Q_j}$ as given in the Equation 7.

$$\mu^*_{C_k x_i} = \frac{\mu_{Q_j x_i} + \mu_{C_k Q_j}}{2} \tag{7}$$

where Q_j is the prototype, at which the data item x_i has highest membership among all prototypes.

Experimentally, it is proved that the values of $\mu^*_{c_k x_i}$ are very close to the exact membership $\mu_{c_k x_i}$ that are calculated using 8.

$$\mu_{C_k x_i} = \frac{1}{\sum_{k=1}^{K} \left(\frac{d(c_k, x_i)}{d(c_j, x_i)}\right)^{\frac{2}{m-1}}} \tag{8}$$

The fuzzy clustering methods with prototypes of varying neighborhoods are presented in the following algorithm 1.

Algorithm 1 Fuzzy clustering with prototype of varying neighborhoods

Input: $\mathcal{D}$, $\mathcal{C}$, m
Output: Partition of $\mathcal{D}$
Stage 1:
1. Find a set of fuzzy prototypes $\mathcal{Q} = \{Q_1, Q_2, ...Q_s\}$, from $\mathcal{D}$ in single scan.
2. Calculate the memberships of x_i in Q_j, that is $\mu_{Q_j x_i}$ using 5.
Stage 2:
3. Apply FCM (or Kernel FCM-F or Kernel FCM-K) algorithm on $\mathcal{Q}$ and calculate the memberships, $\mu_{C_k Q_j}$ using 6
6. Compute $\mu^*_{C_k x_i}$ of each data item x_i in every cluster C_k using 7

4 Experimental Study

The experimental study has been done on the benchmark data sets such as Pendigits, Optical Character Recognition (OCR), Letter Image Recognition (LIR) and Shuttle data sets available on UCI Machine Learning Repository [1].

Pendigits data set has 10992 data items with 16 dimensions and 10 classes. OCR data set contains 10003 items with 192 and 10 classes. The size of LIR data set is 20000 with 16 dimensions and 26 classes. Shuttle data set contains 58000 items with 9 dimensions and 7 classes.

The proposed fuzzy prototypes with varying neighborhoods are tested with FCM, Kernel FC Means-F and Kernel FC Means-K clustering methods on the above data sets. The values of clustering accuracy(CA) and running time(RT) of the proposed approaches are compared with the existing contemporary methods of FCM, Kernel FCM-F and Kernel FCM-K using fixed ϵ-neighborhoods, called PFCM, PKFCM-F, PKFCM-K [9,10]. The proposed prototype based methods are highlighted with * symbol. Adjusted Rand Index is used to measure the clustering accuracy and the running time is measured in seconds. Experiments are conducted using Google Colab. The average results over 20 runs are presented

in tables 1 and 3. The reduction in clustering accuracy and running time are presented in tables 2 and 4 respectively. It is observed that using the proposed approach with prototypes of varying neighborhoods, a significant reduction in the running time from 15% to 25% is achieved with a small compromise in clustering accuracy from 1% to 4%.

Table 1. Comparing the proposed methods with other approaches *w.r.t* CA values

Data set	FCM	P-FCM	P-FCM*	KFCM-F	P-KFCM-F	P-KFCM-F*	P-KFCM-K	P-KFCM-K	P-KFCM-K*
Pendigits	88.9	84.7	87.2	89.2	82.6	87.9	88.4	81.2	84.9
OCR	86.2	82.5	85.3	89.6	82.3	87.3	86.2	80.9	84.6
LIR	78.9	72.8	74.3	89.9	86.5	88.8	83.4	74.9	80.7
Shuttle	84.2	80.1	82.2	90.1	82.4	87.4	84.2	76.8	81.9

Table 2. Comparing the reduction in CA obtained using proposed methods with other approaches

Dataset	Reduction in Clustering Accuracy (%)					
	P-FCM	P-FCM*	P-kernel FCM-F	P-kernel FCM-F*	P-kernel FCM-K	P-kernel FCM-K*
Pendidits	2.9	1.9	6.0	1.5	4.3	3.9
OCR	3.2	1.1	5.8	2.5	4.4	1.9
LIR	4.6	3.2	2.6	1.2	7.2	3.2
Shuttle	2.6	2.3	5.8	2.9	6.2	2.7

Table 3. Comparing the proposed methods with other approaches *w.r.t* RT values

Data set	FCM	P-FCM	P-FCM*	KFCM-F	P-KFCM-F	P-KFCM-F*	KFCM-K	P-KFCM-K	P-KFCM-K*
Pendigits	513.2	336.4	375.3	918.3	619.2	715.2	1014.5	681.5	813.4
OCR	994.2	752.3	778.5	1252.1	918.2	999.3	1401.6	992.5	1101.5
LIR	1723.3	1032.7	1321.3	2541.7	1854.2	2013.2	2721.7	91819.1	2108.3
Shuttle	3624.7	2698.2	2894.5	3921.4	2631.4	3041.5	3987.4	2832.7	3124.9

Table 4. Comparing the reduction in RT obtained using proposed methods with other approaches

Dataset	Reduction in Running Time (%)					
	P-FCM	P-FCM*	P-KernelFCM-F	P-KernelFCM-F*	P-KernelFCM-K	P-KernelFCM-K*
Pendidits	10.37	26.87	13.43	22.12	16.22	19.83
OCR	6.84	21.69	8.12	20.19	9.89	21.41
LIR	21.84	23.73	7.89	20.79	13.72	22.54
Shuttle	6.78	20.54	13.49	22.44	9.36	21.64

5 Conclusions

This article presented a new fuzzy prototype based clustering approach to speed up the existing fuzzy clustering methods like FCM, Kernel FC Means-F and Kernel FC Means-K. There are some improvements in recent times using the prototypes, where each prototype represents a set of data items in a neighborhood, say ϵ. The key challenge of finding the optimal value of ϵ is addressed in the current work, by finding the neighborhood of each prototype based on the data streaming in. Empirically, it is observed that the membership of every data item in the given data can be easily approximated using its membership in each prototype and the prototype's membership in each cluster. The proposed approach results in a great reduction in running time from 15% to 25% with very little compromise in clustering accuracy from 1% to 4%.

References

1. Arthur Asuncion and David Newman. Uci machine learning repository, 2007.
2. James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.
3. Daniel Graves and Witold Pedrycz. Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy sets and systems*, 161(4):522–543, 2010.
4. Richard J Hathaway and James C Bezdek. Extending fuzzy and probabilistic clustering to very large data sets. *Computational Statistics & Data Analysis*, 51(1):215–234, 2006.
5. Timothy C Havens, James C Bezdek, Christopher Leckie, Lawrence O Hall, and Marimuthu Palaniswami. Fuzzy c-means algorithms for very large data. *IEEE Transactions on Fuzzy Systems*, 20(6):1130–1146, 2012.
6. SR Kannan, R Devi, S Ramathilagam, TP Hong, and A Ravikumar. Effective kernel fcm: Finding appropriate structure in cancer database. *International Journal of Biomathematics*, 9(02):1650018, 2016.
7. Lujia Lei, Chengmao Wu, and Xiaoping Tian. Robust deep kernel-based fuzzy clustering with spatial information for image segmentation. *Applied Intelligence*, pages 1–26, 2022.
8. K. Mrudula. *Some studies on improvements over fuzzy kernel based clustering methods*. PhD thesis, Jawaharlal Nehru Technological University, Anantapuram, 30-08-2018.

9. K Mrudula and E Keshava Reddy. Improving kfcm-f algorithm using prototypes. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*, pages 695–701. Springer, 2019.
10. K Mrudula and T Hitendra Sarma. A prototype based hybrid approach to speed-up kernel fcm-k. In *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pages 1–8. IEEE, 2019.
11. T. Hitendra Sarma and P. Viswanath. Speeding-up the k-means clustering method: A prototype based approach. In *Proc. 3rd Int. Conf. on Pattern Recognition and Machine Intelligence(PReMI)LNCS 5909*, page 56–61, Berlin Heidelberg, 2009. Springer-Verlag.
12. T Hitendra Sarma, P Viswanath, and Atul Negi. Speeding-up the prototype based kernel k-means clustering method for large data sets. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 1903–1910. IEEE, 2016.
13. T Hitendra Sarma, P Viswanath, and B Eswara Reddy. A fast approximate kernel k-means clustering method for large data sets. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, pages 545–550. IEEE, 2011.
14. T Hitendra Sarma, P Viswanath, and B Eswara Reddy. A hybrid approach to speed-up the k-means clustering method. *International Journal of Machine Learning and Cybernetics*, 4(2):107–117, 2013.
15. T Hitendra Sarma, P Viswanath, and B Eswara Reddy. Speeding-up the kernel k-means clustering method: A prototype based hybrid approach. *Pattern Recognition Letters*, 34(5):564–573, 2013.
16. Seyyit Alper Sert and Adnan Yazici. Increasing energy efficiency of rule-based fuzzy clustering algorithms using clonalg-m for wireless sensor networks. *Applied Soft Computing*, 109:107510, 2021.
17. Chengmao Wu and Zeren Wang. Robust fuzzy dual-local information clustering with kernel metric and quadratic surface prototype for image segmentation. *Applied Intelligence*, pages 1–30, 2022.
18. Chengmao Wu and Xue Zhang. Total bregman divergence-driven possibilistic fuzzy clustering with kernel metric and local information for grayscale image segmentation. *Pattern Recognition*, 128:108686, 2022.
19. Shenghui Zhang, Chen Wang, Peng Liao, Ling Xiao, and Tonglin Fu. Wind speed forecasting based on model selection, fuzzy cluster, and multi-objective algorithm and wind energy simulation by betz's theory. *Expert Systems with Applications*, 193:116509, 2022.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.