MCOKE: Multi-Cluster Overlapping K-Means Extension Algorithm

Said Baadel, Fadi Thabtah, Joan Lu

***Abstract*—**Clustering involves the partitioning of *n* objects into *k* clusters. Many clustering algorithms use hard-partitioning techniques where each object is assigned to one cluster. In this paper we propose an overlapping algorithm MCOKE which allows objects to belong to one or more clusters. The algorithm is different from fuzzy clustering techniques because objects that overlap are assigned a membership value of 1 (one) as opposed to a fuzzy membership degree. The algorithm is also different from other overlapping algorithms that require a similarity threshold be defined a priori which can be difficult to determine by novice users.

***Keywords*—**Data mining, k-means, MCOKE, overlapping.

I. Introduction

DATA clustering, also known as unsupervised classification is a research field widely studied in data mining and machine learning domains due to its applications to segmentation, summarization, learning, and target marketing [1]. Clustering involves the partitioning of a set of objects or data into clusters or subsets such that the objects or data in each subset or cluster contains similar traits based on measured similarities [2] and data from different clusters are dissimilar [13]. There are many techniques that have been explored for clustering processes such as distance-based, probabilistic, and density/grid-based etc. with the distance based being very popular in research fields. Two major types of distance-based algorithms are hierarchical and partitioning. The first type typically represents clusters hierarchically through a dendo-gram and using a similarity criterion either splits or merges the partitions to create a tree-like structure [3]. The second type divides data into several initial clusters partitions and iteratively data is assigned to their closest cluster partition or centroid using a dissimilarity criterion. According to [4] hierarchical algorithms are more suitable for small data sets and partitioning algorithms for large data sets.

Algorithms that perform these two types of clustering use mostly crisp or hard-clustering techniques where each object is assigned to a single cluster. Fuzzy clustering techniques allow objects to belong to multiple clusters with different degrees [5] by assigning membership degrees to the objects and assigning the object to the cluster that has the highest degree. Overlapping clustering techniques allows an object to belong to one or more clusters [6]. This has several applications such as dynamic system identification, document categorization (document belonging to different clusters) etc. among others.

K-means is one of the most frequently used partitioning clustering algorithms and also considered one of the simplest methods for clustering [7]. This study proposes modified K-means to produce overlapping clusters by using the maximum Euclidean distance assigned to all the objects as the global threshold to reassign data to overlapping clusters.

The paper is organized as follows: Section II will give a review of related work on overlapping cluster algorithms. Section III presents MCOKE: Multi-Cluster Overlapping K-Means Extension (MCOKE) algorithm that we propose. After that the experiments and results are discussed. Finally Section V mentions the conclusions and future work.

II. Related Work on Overlapping Clustering

Most research on overlapping clustering has focused on algorithms that evolve fuzzy partitions of data [8] and based around the Fuzzy C-means and many of its variants [5], [9]. Data objects are assigned membership degrees (values between 0 and 1) to a particular cluster where the total sum of all membership values must add up to 1 hence generating soft partitions. Objects are eventually assigned to clusters that have the highest degree of membership. If the highest degree of membership is not unique, then an object is assigned to an arbitrary cluster that achieves the maximum. By adding a constraint where the data object must belong to a cluster with the highest membership degree, a “1” is imposed on every object in the matrix thus degenerating it to hard-partitioning.

Other overlapping algorithms such as Overlapping Partitioning Cluster (OPC) [10] that does not consider the similarity of objects in the same cluster but rather use a similarity threshold that determines whether the object will belong to a cluster or not. An object that has a distance of less than the threshold can belong to multiple clusters this way. The Overlapping K-means (OKM) [11] has a similar threshold to determine the belonging of an object. The drawback of such algorithms is that the threshold is determined a *priori* and may not be easy to determine the right threshold for different data samples.

The MCOKE algorithm assigns the global threshold to determine the belonging of a data object to a cluster once the K-means algorithm finishes its iterations and picks the maximum distance (*maxdist*) of all objects that were assigned to the clusters. In this case, the threshold maxdist is not a priori and can change depending on the data.

Said Baadel is with the Canadian University of Dubai, Dubai, UAE (e-mail: baadel@cud.ac.ae).

Fadi Thabtah is with the Canadian University of Dubai, Dubai, UAE.

Joan Lu is with The University of Huddersfield, UK.

III. MCOKE: Multi-Cluster Overlapping K-Means Extension Algorithm

The MCOKE algorithm consists of two procedures. The standard K-means clustering that iterates through the data objects in order to attain a distinct partitioning of the data points given a priori number of C clusters by minimizing the distance between the objects and the cluster centroids.

```
Input: Number of clusters K, A set of data vectors
Output: Membership Matrix
Step One:
    • Draw randomly k initial cluster prototypes
Step Two:
    • For each data point, compute the centroid it is closest to
      using Euclidean distance measure
    • Assign the data point to the cluster
    • Re-compute and update the centroids
Step Three:
    • If not converged, repeat Step Two; otherwise return
      assignment vector, final centroids vector, and maximum
      Euclidean distance (maxdist) used in the assignment
Step Four:
    • Draw initial membership matrix MT
    • For each data point, compare with final centroid vector
      distance with maxdist
    • Add new member in MT if distance is shorter than
      maxdist
```

Fig. 1 MCOKE Algorithm

The goal of K-means is essentially optimizing the objective function provided as

$$J = \sum_{i=1}^{C} \sum_{xi\in\mu i} d(x_i, v_i) \quad (1)$$

where v_i is the center of luster μ_i and d(x_i, v_i) is the Euclidean distance between a point x_i and v_i. The objective J will therefore decrease with every iteration until it converges to a local minimum. Hence we know that each object assigned to a particular cluster centroid will have the minimum distance to it compared to other clusters. After the distinct partitioning of the data set, each object is assigned to one cluster. The maximum distance allowed by K-means in assigning the data objects to a cluster is saved as *maxdist* and is used in our algorithm as the global threshold of belonging to a cluster to be used in procedure two below when assigning objects to multi-clusters.

The clustering algorithm starts by comparing the cluster centroids to a random generated initial cluster centers and iteratively re-computes the cluster centroids to a more sensible location until the centroids do not change.

The second procedure in our algorithm uses the results produced from the K-means algorithm to generate a membership table MT (of dimension N x C) such that MT(*i*,*j*) denotes a member of object *i* to cluster *j* where *i* = 1,…,N and *j* = 1,...,C. Each object in MT(*i*,*j*) is assigned a 1 (one) to denote membership to that cluster and a 0 (zero) for non-membership of a cluster.

The algorithm then iterates through the table MT and compares the distance of the objects assigned to their respective clusters with the other final centroids in the table. If the object distance is less than the *maxdist* (used as the threshold for belonging to a cluster) generated from K-means algorithm then that object is also assigned to that cluster centroid and the membership table is updated with a 1 (one). The complexity of the MCOKE is similar to that of K-means procedure of *O*(*n*) where *n* is the number of data points in the dataset.

IV. Experiments

We run three experiments to highlight and test the algorithm. Experiment 1 uses a sample data to demonstrate the application of MCOKE as compared to the standard K-means algorithm.

TABLE I
Sample Experimental Data

Book	Price, Chapter
ENGLISH	40,4
INTRO COMPUTER	37,4
STRATEGY	45,4
E-BUSINESS	34,2
ECON	38,5
STATISTICS	38,6
MATHS	38,3

We set k, the number of clusters to 4 and the following membership matrix is returned where each data set belongs to one cluster. The algorithm labels the clusters as C0, C1, C2, C3, and a 1or 0 is assigned depending on whether they belong to the cluster.

TABLE II
K-Means Membership Table

Vector	Cluster Centroid	C0	C1	C2	C3
40,4	42.5,4	0	1	0	0
37,4	37.5,3.5	1	0	0	0
45,4	42.5,4	0	1	0	0
34,2	34,2	0	0	0	1
38,5	38,5.5	0	0	1	0
38,6	38,5.5	0	0	1	0
38,3	37.5,3.5	1	0	0	0

We then run the same data set through MCOKE and a membership matrix with the same cluster centroids returned as follows.

TABLE III
IMPLEMENTATION RESULT 1 OF MCOKE ALGORITHM

Vector	Cluster Centroid	C0	C1	C2	C3
40,4	42.5,4	1	0	1	0
37,4	37.5,3.5	0	0	1	1
45,4	42.5,4	1	0	0	0
34,2	34,2	0	1	0	0
38,5	38,5.5	0	0	1	1
38,6	38,5.5	0	0	0	1
38,3	37.5,3.5	0	0	1	1

TABLE IV
IMPLEMENTATION RESULT II OF MCOKE ALGORITHM ON UCI DATA

Vector	C0	C1	C2
14.23,1.71,2.43,15.6,127,2.8,3.06,0.28,2.29,5.64,1.04,3.92,1065	0	1	1
13.2,1.78,2.14,11.2,100,2.65,2.76,0.26,1.28,4.38,1.05,3.4,1050	0	1	1
13.16,2.36,2.67,18.6,101,2.8,3.24,0.3,2.81,5.68,1.03,3.17,1185	0	1	1
13.16,2.36,2.67,18.6,101,2.8,3.24,0.3,2.81,5.68,1.03,3.17,1185	0	1	1
14.37,1.95,2.5,16.8,113,3.85,3.49,0.24,2.18,7.8,0.86,3.45,1480	1	1	0
13.24,2.59,2.87,21,118,2.8,2.69,0.39,1.82,4.32,1.04,2.93,735	0	1	1
14.2,1.76,2.45,15.2,112,3.27,3.39,0.34,1.97,6.75,1.05,2.85,1450	1	1	0
14.39,1.87,2.45,14.6,96,2.5,2.52,0.3,1.98,5.25,1.02,3.58,1290	1	1	0
14.06,2.15,2.61,17.6,121,2.6,2.51,0.31,1.25,5.05,1.06,3.58,1295	1	1	0
14.83,1.64,2.17,14,97,2.8,2.98,0.29,1.98,5.2,1.08,2.85,1045	0	1	1
13.86,1.35,2.27,16,98,2.98,3.15,0.22,1.85,7.22,1.01,3.55,1045	0	1	1
14.1,2.16,2.3,18,105,2.95,3.32,0.22,2.38,5.75,1.25,3.17,1510	1	1	0
14.12,1.48,2.32,16.8,95,2.2,2.43,0.26,1.57,5,1.17,2.82,1280	1	1	0
13.75,1.73,2.41,16,89,2.6,2.76,0.29,1.81,5.6,1.15,2.9,1320	1	1	0
14.75,1.73,2.39,11.4,91,3.1,3.69,0.43,2.81,5.4,1.25,2.73,1150	0	1	1
14.38,1.87,2.38,12,102,3.3,3.64,0.29,2.96,7.5,1.2,3,1547	1	0	1
13.63,1.81,2.7,17.2,112,2.85,2.91,0.3,1.46,7.3,1.28,2.88,1310	1	1	0
14.3,1.92,2.72,20,120,2.8,3.14,0.33,1.97,6.2,1.07,2.65,1280	1	1	0
13.83,1.57,2.62,20,115,2.95,3.4,0.4,1.72,6.6,1.13,2.57,1130	0	1	1
14.19,1.59,2.48,16.5,108,3.3,3.93,0.32,1.86,8.7,1.23,2.82,1680	1	0	1
3.64,3.1,2.56,15.2,116,2.7,3.03,0.17,1.66,5.1,0.96,3.36,845	0	0	1
14.06,1.63,2.28,16,126,3,3.17,0.24,2.1,5.65,1.09,3.71,780	0	0	1
12.93,3.8,2.65,18.6,102,2.41,2.41,0.25,1.98,4.5,1.03,3.52,770	0	0	1
13.71,1.86,2.36,16.6,101,2.61,2.88,0.27,1.69,3.8,1.11,4,1035	0	1	1
12.85,1.6,2.52,17.8,95,2.48,2.37,0.26,1.46,3.93,1.09,3.63,1015	0	1	1
13.5,1.81,2.61,20,96,2.53,2.61,0.28,1.66,3.52,1.12,3.82,845	0	1	1
13.05,2.05,3.22,25,124,2.63,2.68,0.47,1.92,3.58,1.13,3.2,830	0	1	1
13.39,1.77,2.62,16.1,93,2.85,2.94,0.34,1.45,4.8,0.92,3.22,1195	0	1	1
13.3,1.72,2.14,17,94,2.4,2.19,0.27,1.35,3.95,1.02,2.77,1285	1	1	0
13.87,1.9,2.8,19.4,107,2.95,2.97,0.37,1.76,4.5,1.25,3.4,915	1	0	1
14.02,1.68,2.21,16,96,2.65,2.33,0.26,1.98,4.7,1.04,3.59,1035	0	1	1
13.73,1.5,2.7,22.5,101,3,3.25,0.29,2.38,5.7,1.19,2.71,1285	1	1	0
13.58,1.66,2.36,19.1,106,2.86,3.19,0.22,1.95,6.9,1.09,2.88,1515	1	0	0
13.68,1.83,2.36,17.2,104,2.42,2.69,0.42,1.97,3.84,1.23,2.87,990	0	1	1
13.76,1.53,2.7,19.5,132,2.95,2.74,0.5,1.35,5.4,1.25,3,1235	0	1	0
13.51,1.8,2.65,19,110,2.35,2.53,0.29,1.54,4.2,1.1,2.87,1095	0	1	1
13.48,1.81,2.41,20.5,100,2.7,2.98,0.26,1.86,5.1,1.04,3.47,920	0	0	1
13.28,1.64,2.84,15.5,110,2.6,2.68,0.34,1.36,4.6,1.09,2.78,880	0	0	1
13.05,1.65,2.55,18,98,2.45,2.43,0.29,1.44,4.25,1.12,2.51,1105	0	1	1
13.07,1.5,2.1,15.5,98,2.4,2.64,0.28,1.37,3.7,1.18,2.69,1020	0	1	1
14.22,3.99,2.51,13.2,128,3,3.04,0.2,2.08,5.1,0.89,3.53,760	0	1	1
13.56,1.71,2.31,16.2,117,3.15,3.29,0.34,2.34,6.13,0.95,3.38,795	0	1	1
13.41,3.84,2.12,18.8,90,2.45,2.68,0.27,1.48,4.28,0.91,3,1035	0	1	1
13.88,1.89,2.59,15,101,3.25,3.56,0.17,1.7,5.43,0.88,3.56,1095	0	1	1
13.24,3.98,2.29,17.5,103,2.64,2.63,0.32,1.66,4.36,0.82,3,680	0	0	1
13.05,1.77,2.1,17,107,3,3,0.28,2.03,5.04,0.88,3.35,885	0	0	1
14.21,4.04,2.44,18.9,111,2.85,2.65,0.3,1.25,5.24,0.87,3.33,1080	0	1	1
14.38,3.59,2.28,16,102,3.25,3.17,0.27,2.19,4.9,1.04,3.44,1065	0	1	1
14.1,2.02,2.4,18.8,103,2.75,2.92,0.32,2.38,6.2,1.07,2.75,1060	0	1	1
13.94,1.73,2.27,17.4,108,2.88,3.54,0.32,2.08,8.9,1.12,3.1,1260	0	1	0
13.05,1.73,2.04,12.4,92,2.72,3.27,0.17,2.91,7.2,1.12,2.91,1150	0	1	1
13.83,1.65,2.6,17.2,94,2.45,2.99,0.22,2.29,5.6,1.24,3.37,1265	0	1	1
13.82,1.75,2.42,14,111,3.88,3.74,0.32,1.87,7.05,1.01,3.26,1190	0	1	1
13.77,1.9,2.68,17.1,115,3,2.79,0.39,1.68,6.3,1.13,2.93,1375	1	1	0
13.74,1.67,2.25,16.4,118,2.6,2.9,0.21,1.62,5.85,0.92,3.2,1060	0	1	1
13.56,1.73,2.46,20.5,116,2.96,2.78,0.2,2.45,6.25,0.98,3.03,1120	0	1	1
13.29,1.97,2.68,16.8,102,3,3.23,0.31,1.66,6,1.07,2.84,1270	0	1	0
13.72,1.43,2.5,16.7,108,3.4,3.67,0.19,2.04,6.8,0.89,2.87,1285	1	1	0

Here we note that while the cluster centroids are maintained in this instance, the membership matrix is updated each time the distance from the vector to the cluster centroid is less than *maxdist*.

We then conduct an experiment on real data that strongly show overlapping clustering from the UCI Machine Learning Repository domain on wine [12]. The data contains 13 attributes {alcohol, malic acid, ash, magnesium,…} that describe the constituents found in three types of wines. Class 1 of the wines has 59 instances depicted below.

The results suggest that out of the 59 instances of class 1 with 13 different attributes, only 10 instances can be uniquely identified with only 1 cluster while the rest overlap to multiple clusters. The test data is run 3 more times returning the following results.

TABLE V
IMPLEMENTATION RESULT III OF MCOKE ALGORITHM ON UCI DATAWINE

No. of Run	Objects in C1	Objects in C1	Objects in C2
1	17	48	42
2	26	52	36
3	42	17	48

A third experiment is conducted on UCI Repository on the Iris Plants dataset [12]. The data is classified into 3 classes of plant species with 50 instances on each class. There are 4 attributes {sepal length, sepal width, petal length, and petal width} on the set. We run the data through MCOKE with different *k* clusters and the following result on Table VI is reported with the number of objects in each cluster.

TABLE VI
IMPLEMENTATION RESULT I OF MCOKE ALGORITHM ON UCI DATA IRIS

K	C1	C2	C3	C4	C5	Mindist	Maxdist	Iter
3	50	98	92	-	-	0.05993	1.66064	5
4	92	50	50	98	-	0.10377	1.6468	7
5	50	58	26	66	45	0.05993	1.23935	10

The table shows the overlapping objects in different cluster sizes and the minimum and maximum Euclidean distance with the iterations taken to achieve the results.

V. CONCLUSION AND FUTURE WORK

In this paper we introduced MCOKE algorithm, an extension of K-means algorithm which is to partition *n* objects into *C* clusters that may overlap with each other. The algorithm differs from other algorithms in that it does not require a similarity threshold to be defined a *priori* which may be difficult to set depending on the data samples but rather uses the maximum distance (*maxdist*) allowed in K-means to assign objects to a given cluster as the global threshold. An initial run of the standard K-means algorithm produces a membership matrix table where each object is assigned to one cluster with the minimum Euclidean distance from the centroid. At this stage all objects are assigned to at least one cluster hence the objective function of K-means is optimized. A second run is done through the membership table comparing the vectors with the final cluster centroids determined by the K-means algorithm and a new membership is assigned to the dataset object and the table updated if the distance between the two is less than *maxdist*.

Finally, while we acknowledge that more tests need to be done, MCOKE algorithm returns desirable overlapping cluster results each time its run. However, the algorithm suffers the same drawbacks of the standard K-means algorithm in that it only works with numerical data and that the objective function of K-means is designed to optimize while under constrain of assigning the data objects to hard-partition. Future work can be done to address these issues. We also propose extending the algorithm to be able to combine together every multi-cluster (with object mapping) that are very close to each other and assigning them a new cluster name on the fly while still maintaining the original clusters.

References

[1] C.C. Aggarwal, C.K. Reddy. Data Clustering: Algorithms and Applications. CRC Press, 2014.

[2] A.K. Jain, R.C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.

[3] E. Boundaillier, G. Hebrail. Interactive interpretation of hierarchical clustering. Intell. Data Anal. 1998.

[4] O.A. Abbas. Comparisons between Data Clustering Algorithms. The International Arab Journal of Information Technology, Vol 5. No. 3. 2008.

[5] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition, Wiley, 1999.

[6] B. S. Everitt, S. Landau, M. Leese, “Cluster Analysis”, Arnold Publishers, 2001

[7] A. Jaini. Data Clustering: 50 years beyond k-means. Pattern Recognition Letters, 31(8): pp. 651-666, 2010.

[8] E.R. Hruschkaet. al. A survey of Evolutionary Algorithms for Clustering. IEEE Trans. Vol. 39, pp. 133-155, 2009.

[9] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithm, Plenum Press, 1981.

[10] Y. Chen, H. Hu. An overlapping Cluster algorithm to provide non-exhaustive clustering. Presented at European Journal of Operational Research. pp. 762-780, 2006

[11] G. Cleuzious. An extended version of the k-means method for overlapping clustering. IEEE International Conference on Pattern Recognition. 2008

[12] K. Bache, M. Lichman. UCI Machine Learning Repository (http://archive.ics.uci.edu/ml). Irvine, CA: University of California, School of Information and Computer Science. 2013

[13] N. Abdelhamid, A. Ayesh, F. Thabtah. Phishing detection based Associative Classification data mining. Expert Systems with Applications Journal. Vol. 41 (13). 2014