Journal of AI and Data Mining
Vol 7, No 4, 2019, 551-561
DOI: 10.22044/JADM.2018.4237.1514

Entropy-based Consensus for Distributed Data Clustering

M. Owhadi-Kareshk and M.-R. Akbarzadeh-T.*

Department of Computer Engineering, Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran.

Received 02 May 2016; Revised 07 May 2017; Accepted 03 July 2018
Corresponding author: akbazar@um.ac.ir (M.-R. Akbarzadeh-T.).

Abstract
The increasingly larger scale of available data and the more restrictive concerns on their privacy are some of the challenging aspects of data mining today. In this paper, Entropy-based Consensus on Cluster Centers (EC3) is introduced for clustering in distributed systems with consideration for the confidentiality of data; i.e. it is the negotiations among local cluster centers that are used in the consensus process, hence no private data is transferred. With the proposed use of entropy as an internal measure of consensus clustering validation at each machine, the cluster centers of the local machines with higher expected clustering validity have more influence on the final consensus centers. We also employ the relative cost function of the local Fuzzy C-Means (FCM) and the number of data points in each machine as measures of relative machine validity as compared to other machines and its reliability, respectively. The utility of the proposed consensus strategy is examined on 18 datasets from the UCI repository in terms of clustering accuracy and speed-up against the centralized version of FCM. Several experiments confirm that the proposed approach yields to higher speed-up and accuracy, while maintaining data security due to its protected and distributed processing approach.

Keywords: *Consensus Clustering; Distributed Clustering; Fuzzy C-Means; Ensemble Learning; Entropy.*

1. Introduction

Connections in social networks, sensors of mobile and wearable gadgets, as well as data from space probes are only a few instances of many applications that produce large-scale data nowadays. In addition to volume, variety and velocity are the other main attributes of such applications that are often categorized as 'Big Data.' These databases also often come with inherently distributed availability, raising new concerns on data security and privacy. In other words, we may be reluctant to gather all data in one place and wish to avoid centralized processing, not only due to lack of sufficient memory or processing power but also out of concern for database privacy. Accordingly, distributed clustering becomes an attractive venue. They are more scalable as compared to the competing centralized strategies. They maintain a more distributed processing by taking advantage of the inherently distributed availability of their database. For the same reason, they are able to better address its privacy and security.

From a broader perspective, clustering is also becoming more popular, among other data processing approaches, since most modern datasets are not labeled. In clustering, data is divided based on their natural partitions since there is no label or external knowledge about the data. K-means and spectral clustering are some of the most effective algorithms amongst the various approaches to clustering. However, these algorithms associate each data point to only one specific cluster without considering the uncertainty in data association. This is to the contrary to the real-world nature of the problems where each data point may in actuality belong to more than one cluster with different degrees.

Bezdek was the first to address this problem by combining fuzzy logic with clustering and introducing Fuzzy C-Means (FCM) [1]. In various studies, this combined approach has shown to

better handle data uncertainty. Some of these approaches such as Gustafson-Kessel FCM (GK-FCM) [2] focus on the definition of distance measures. Others such as the Possibilistic C-Means (PCM) [3] aim to reduce the FCM sensitivity to noisy data by changing the constraints of the objective function. Furthermore, some approaches aim to improve the FCM performance by adding a regularization term to the objective function of FCM [4-7]. Recently, in [8] and [9], relative entropy with considering the fuzzy membership values has been proposed. An effective fuzzy clustering method with Particle Swarm Optimization (PSO) has also been presented in [10] for time-series data.

Clustering of local machines in a distributed system may yield to different results due to the possibility of having different available input data, environmental uncertainty, clustering structure, and randomness in the learning process. Here, consensus is defined as the procedure of combining these multiple partitions in order to one 'optimal' partition. Ideally, the machines should reach this consensus by sharing their information with each other only indirectly, i.e. without transferring any actual data points. Transferring the actual data points is avoided due to the privacy concerns in the real-world problems. It also helps reduce network congestion. Furthermore, by taking advantage of such 'ensemble' learning, the result of consensus may become more stable and robust. There are a number of works in the literature on this related topic. For instance, a linear consensus has been presented in [11] and has been recently used for consensus in distributed artificial neural networks [12]. Wu, et. al. [13] have presented a strong theoretical framework of consensus in K-means. The use of kernel functions has been suggested for clustering consensus in [14]. In [15], a weighted consensus has been applied with a diversity criterion. For a further review of these and related techniques, the interested reader is also referred to the work of Pons and Shulcloper in [16].

There are two approaches to consensus-based clustering: object co-occurrence and median partitioning. Those that are based up on the object co-occurrence count the number of times that two data points occur concurrently and then apply a voting procedure to reach consensus. In this process, the data points are relabeled (since there is no relation between the true labels and the generated labels in each clustering) and then the voting procedure is executed. Voting-merging [17], plurality voting [18], cumulative voting [19], and fuzzy clustering voting [20] are amongst the most popular voting-based methods. Relabeling in voting-based methods is a challenging problem; and therefore, in the co-association matrix methods [21], an intermediate representation is produced based on a new similarity measure. Also in [22], the voting-K-Means algorithm has been proposed, which aims to find the consistent clusters by employing the majority voting. The normalized edges can be used to measure the similarity of clusters in a hierarchical fashion [23].

Median partitioning in consensus-based clustering aims to minimize the distance between different partitions by a similarity measure. Non-negative matrix factorization [24] and kernel methods [14] are two effective examples of median partitioning approaches. Counting pairs such as in Rand index [25] and Jaccard coefficient [26], information-theoretic measures such as mutual information [26] and entropy [27], and kernel measures such as graph kernel [28] and positive semi-definite kernels [29]) are amongst the most well-known similarity measures. Recently, using a sparse graph representation, an effective ensemble clustering method has been proposed [30]. In [31], the consensus clustering has been employed for community detection. Furthermore, a divide-and-conquer strategy has been proposed for consensus clustering on Big Data in [32]. An efficient clustering technique for multi-core systems has been also proposed in [33], which is especially designed for high-dimensional data such as microarrays. Furthermore, Olgierd et al. [34] have proposed a stability-based consensus clustering algorithm with the same viewpoint.

More recently, Thanh Ngo et al. [35] have proposed interval-valued fuzzy sets to better address data uncertainty in consensus clustering. The work by Lyu et al. [36] has aimed to preserve the data privacy by using feature reduction before consensus clustering. Wang et al. [37] have introduced a new consensus clustering technique for multi-view data, which can address the 'variety' property of Big Data very well. In another effort to employ fuzzy consensus clustering for Big Data, a new objective function has been defined in [38] based on a novel fuzzified contingency matrix. Then, a set of utility functions have been derived. The fuzzy consensus clustering is defined as a weighted piecewise fuzzy c-means clustering (piFCM) problem at the end. After comparing twelve consensus clustering algorithms, Wu et al. [39] have concluded that average-linkage agglomerative clustering and K-means form the best combination for consensus clustering.

However, the above consensus techniques are often designed for centralized data on one multi-core processor system and use shared memory.

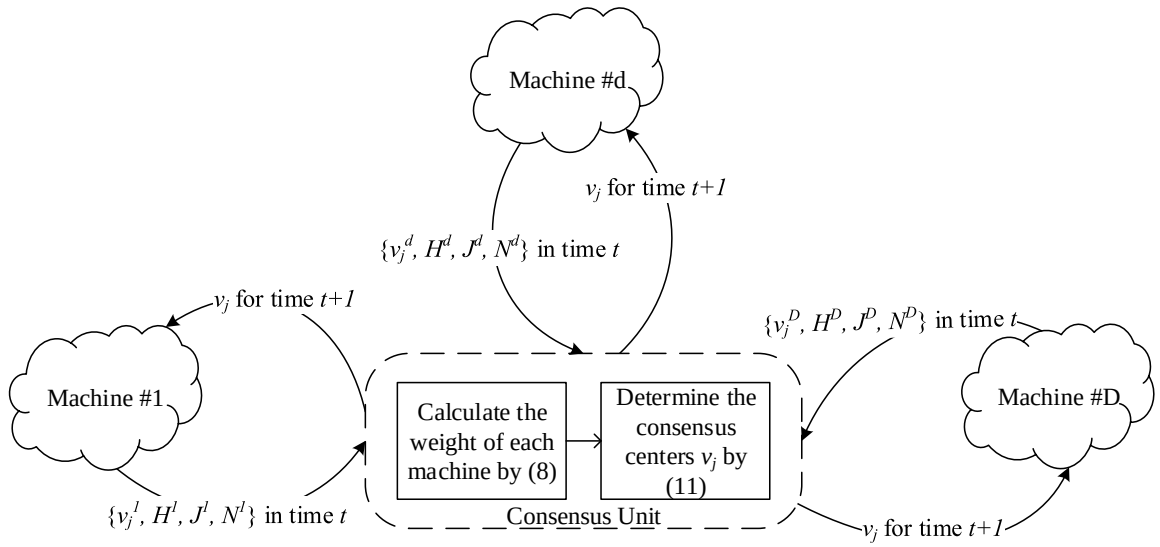

Figure 1. The proposed Entropy-based Consensus Clustering Centers (EC3).

This is while, data privacy and security can be best maintained if data is kept at its origin, i.e. by taking advantage of the distributed nature of the sensory system and reaching consensus among the individual processing elements without sharing any data/memory of the dataset.

Accordingly, we introduce here an entropy-based weighted averaging for reaching consensus from cluster centers that are reported from multiple machines. The basic challenge here is to determine the appropriate weight of each machine from 'internal' measures of clustering rather than the 'external' measures of validation, and without sharing individual data. For this purpose, three consensus measures are proposed here. The first measure uses the relative FCM cost function as a measure of relative clustering validity. This measure is based up on the clustering status of all machines in the distributed system. The second measure uses the sum of the entropies over all clusters on each machine, in contrast to the other works on this topic 'within' a given machine as a measure of its 'internal' validity. Both of these mentioned measures are for clustering validation but they imply two different perspectives, local view and relative global view. And the third is the number of data points in each machine as an estimate of clustering reliability, i.e. more data points observed by a machine can be attributed to a more accurate estimation of data distribution.

The main idea behind the present work is hence that the 'internal' measures of clustering such as the entropy at each machine, instead of the external measures of validation, should be used in the consensus process on the centers of clusters in distributed systems. This is especially relevant to the Big Data problems, where there is no trivial knowledge about the true labels. Accordingly, individual machines should have different effects on the consensus process. The basic root of the variation in the performance of local machines may be in the inherent imbalance in the distribution of data or the random initialization of the clusters at different machines.

The rest of this paper is organized as follows. Section 2 explains the details of the proposed algorithm and its elements. Results of implementing the proposed strategy on 18 datasets from the UCI repository [40] are provided in Section 3. Analysis of these results is also provided in terms of clustering accuracy and speed-up. Lastly, conclusions are drawn and future work is discussed in Section 4.

2. Proposed entropy-based consensus on cluster centers

The main goal of the proposed EC3 is to reach consensus among clustering machines by sharing only cluster centers rather than transferring actual data points among machines, hence preserving privacy. The data is inherently distributed and large; therefore, there is no prior knowledge about its overall shape or distribution. These two steps of repeated clustering and consensus are as follow:

1. **Clustering:** FCM is employed here as an effective clustering method to illustrate the partitioning process of datasets with uncertainty. However, the strategy is general and can be used along with any other clustering approach. Furthermore, the standard version of FCM algorithm is chosen to keep the focus on the consensus strategy. In the standard version of FCM, the main loop is executed iteratively until the stopping criteria are met. Since these

conditions are checked in EC3's external consensus loop, the FCM here is executed for a fewer number of $\mathcal{L}$ iterations. Each machine calculates the cluster centers for its local data and sends the cluster centers and clustering measures to the consensus unit. This partitioning process of local data is further described in Section 2-1.

2. **Consensus:** The consensus process is applied as a weighted averaging method. The main challenge here is to define the appropriate weight of each machine, i.e. its effect in determining the final cluster centers. The weight of each machine is defined here based on the sum of the entropy of all of its clusters and the relative value of its local FCM objective function as two measures of clustering validity and the number of its data points available to each machine as a measure of its reliability. A lower FCM cost function and the sum of entropies over all clusters in a machine are the sign of a "good" clustering. Furthermore, more data points can better describe the whole structure of data, and hence, the number of data points can be an estimate of the clustering reliability factor in the clustering. These three terms can determine the influence of partitioning of each machine on the final consensus result. This process is explained further in Section 2-2.

As illustrated in figure 1, after each $\mathcal{L}$ iteration of the FCM algorithm, each machine submits its cluster centers and the information that is required for determining its weight to the consensus unit. This information consists of its clustering validation (the sum of entropy over all clusters and the value of FCM cost function) and reliability (the number of data points stored on the machine) measures. Consensus unit then calculates the new centers by weighted averaging. These new consensus centers are the new initial points for executing the new FCM algorithm on each machine. This procedure is repeated until the stopping conditions are satisfied. In fact, only the first execution of local FCMs are initialized with random centers; and in the subsequent iterations, consensus centers are the starting points of local FCM algorithms. In the following, the main modules of the proposed method are introduced. It should be mentioned that the use of the standard FCM in the proposed approach can be easily generalized to other clustering algorithms.

2.1. Fuzzy C-Means at each machine

FCM is the first functional stage of EC3 that is executed on each machine $d = 1, \dots, D$ in the distributed architecture. It aims to find the optimal cluster centers from the local data available to it, such that the sum of distances between all data points of each cluster and their cluster center are minimized, as shown by the following cost function:

$$J_d\left(u^d, v^d\right) = min \sum_{j=1}^{c^d} \sum_{i=1}^{N^d} u_{ij}^{d\,m} x_i^d - v_j^{d\,2} \tag{1}$$

such that the following conditions are satisfied:

$$0 < \sum_{i=1}^{N^d} u_{ij}^d < N^d \;; \forall j = 1, \dots, c^d, \tag{2}$$

$$\sum_{j=1}^{c^d} u_{ij}^d = 1 \;; \forall i = 1, \dots, N^d, \tag{3}$$

$$0 \le u_{ij}^d \le 1 \;; \forall i = 1, \dots, N^d \;, \quad j = 1, \dots, c^d. \tag{4}$$

Here, x_i^d is the i^{th} data point in machine d, v_j^d is the center of j^{th} cluster, u_{ij}^d is the membership degree of data point i belonging to fuzzy cluster j, m is the fuzziness value (here, *2* is used), N^d is the total number of data points, c^d is the pre-defined number of clusters that is known apriori, and $\|.\|$ is the Euclidean norm (any other metric can be used). Using the Lagrangian multipliers, (5) and (6) are found for determining the membership values and cluster centers.

$$u_{ij}^d = \frac{1}{\sum_{k=1}^{c} \left(\frac{x_i^d - v_j^{d\,2}}{x_i^d - v_k^{d\,2}} \right)^{\frac{2}{m-1}}} \tag{5}$$

$$v_j = \frac{\sum_{i=1}^{N^d} u_{ij}^{d\,m}\; x_i^d}{\sum_{i=1}^{N^d} u_{ij}^{d\,m}}. \tag{6}$$

Algorithm 1: (Modified) Fuzzy C-means

1. Start
2. Set: Number of clusters and m
3. l=1
4. while l< $\mathcal{L}$:
    a. Calculate the membership values (u_{ij}^d) by Eq. 5
    b. Calculate the center of clusters (v_j^d) by Eq. 6
    c. l=l+1
5. Finish

Table 1. Datasets for experiments.

No.	Dataset	# Data	# Features	# Classes	$\mathcal{L}$ (Section 3-2)
1	Balance	625	4	3	22
2	Breastcancer	569	30	2	4
3	Gamma	19020	10	2	6
4	Glass	214	9	6	5
5	Haberman	306	3	2	4
6	Heart	270	13	2	4
7	Ionosphere	351	34	2	3
8	Iris	150	4	3	3
9	Isolet	7797	617	26	5
10	Pendigits	10992	16	10	12
11	Pima	768	8	2	5
12	Seeds	210	7	3	3
13	Segment	2310	19	7	8
14	Sonar	208	60	2	6
15	Tictactoe	958	9	2	2
16	Waveform	5000	21	3	5
17	Waveform noise	5000	40	3	13
18	Wine	178	13	3	5

The modified FCM algorithm is shown in Algorithm 1. In the standard version of FCM, stopping conditions such as reaching the maximum number of iterations or convergence of cluster centers to an optimally acceptable performance are controlled after each iteration. However, in this work, FCM is itself a part of the external consensus loop; and hence, the stopping condition in the modified FCM algorithm is only reaching a pre-defined number of $\mathcal{L}$ iterations. In other words, $\mathcal{L}$ is the number of FCM iterations that is performed between any two consecutive consensus steps. The performance criterion is then checked at the external consensus steps. A simple heuristic is proposed in this paper to determine $\mathcal{L}$, as defined below:

$$\mathrm{L} = \sqrt{L \diagup 2}, \tag{7}$$

where L is the estimated number of steps required for a conventional 'centralized' version of FCM algorithm to reach the optimal solution for a particular application. Intuitively, L represents the complexity of the clustering task for a given dataset.

2.2. Entropy-based consensus on cluster centers (EC3)

Although the same clustering algorithm with the same parameters is executed on each local machine, each machine reaches a different set of cluster centers due to the differences in their local input data and their random initialization of centers. For the consensus process, these cluster centers are then shared among machines after each $\mathcal{L}$ iterations of local FCMs without transferring any actual data points. For this purpose, we propose a weighted sum heuristic that uses only the cluster centers at each machine (rather than transfer of actual data), and instead, is based up on measures of the internal validity, relative clustering performance, and reliability for machine d. More specifically, the weight of machine d in the consensus process is:

$$w^d = \frac{N^d}{H^d \tilde{J}^d}, \tag{8}$$

where H^d is the sum of entropies over all clusters, $\tilde{J}^d$ is the relative FCM cost function, and N^d is the total number of data points at machine d.

More specifically, the sum of entropies H^d is defined as:

$$H^d = -\frac{1}{c}\sum_{j=1}^{c}\sum_{x_i \in c_j} u_{ij}^d \log_2 u_{ij}^d, \tag{9}$$

and is inversely proportional to the *internal validity* of the machine's clustering process.

Table 2. Average accuracy over three independent runs (bolded numbers indicate superior results).

Dataset	Proposed EC3 with different number of machines					Centralized FCM
	2	*4*	*6*	*8*	*10*	
Balance	**64.96**	57.28	48.00	51.52	56.96	64.00
Breastcancer	85.06	85.59	**85.41**	82.25	**85.41**	**85.41**
Gamma	**61.31**	61.19	61.16	61.28	61.29	61.23
Glass	76.17	**87.38**	86.92	64.95	59.35	53.74
Haberman	54.90	51.63	52.61	**55.88**	53.59	52.61
Heart	59.26	**60.74**	59.63	59.63	60.37	59.26
Ionosphere	**71.23**	70.94	70.66	70.66	70.94	70.94
Iris	**92.67**	**92.67**	89.33	82.67	92.00	89.33
Isolet	91.55	91.62	91.54	91.20	91.55	**91.65**
Pendigits	68.84	69.46	69.55	68.82	74.06	**75.01**
Pima	74.22	**77.08**	73.96	74.22	74.74	74.22
Seeds	77.62	83.81	84.76	88.57	**89.52**	**89.52**
Segment	**61.82**	55.41	56.19	56.10	51.60	55.19
Sonar	54.33	54.33	54.33	54.33	54.81	**55.29**
Tictactoe	52.51	**55.43**	53.44	52.09	52.09	51.04
Waveform	**60.48**	55.58	55.58	55.58	55.58	55.58
Waveform noise	**64.08**	64.00	63.98	63.88	63.86	64.02
Wine	67.98	**69.10**	67.42	68.54	68.54	68.54

Since transferring u_{ij}^{d} is resource consuming, the entropy H^{d} is computed at each machine and then shared with the consensus unit. $\tilde{J}^{d}$ represents the relative FCM cost (reverse of goodness) of the clustering process in machine *d* in comparison with other machines, and is defined as:

$$\tilde{J}^{d} = \frac{J^{d}}{\sum_{k=1}^{D} J^{k}}. \quad (10)$$

This measure is also inversely proportional to clustering validity, although it provides a *relative measure of performance among machines* in the consensus process. Finally, increasing N^{d} implies that more data points at machine d is directly proportional to a higher *relative reliability* of the machine.

At the end, the weights of all machines are normalized by dividing each weight by the sum of all weights. Accordingly, the consensus process replaces the calculated cluster centers in each machine with their weighted average:

$$v_j = \sum_{d=1}^{D} w^{d} v_j^{d}. \quad (11)$$

The nearest-neighbor strategy is employed to determine the associated cluster centers at each machine for the above consensus process. In other words, each consensus cluster center v_j is aligned with the closest cluster center of the other machines. After this consensus step, each machine repeats the above clustering and consensus procedure by executing the FCM algorithm again from these consensus centers as initial points. Algorithm 2 summarizes the proposed consensus algorithm.

Algorithm 2: Entropy-based Consensus on Cluster Centers (EC3)

1. Start
2. While $\delta(t) < \varepsilon$ (here 10^{-5})
   a. Run FCM in each machine simultaneously using Algorithm 1, start from the random cluster centers in the first iteration, and from the output of 2.c in the next iterations.
   b. Determine the weight w^{d} of each machine using (8)
   c. Perform clustering consensus using (11)
3. Finish

where $\delta(t) = \sum_{d=1}^{D}(v_j^{d}(t) - v_j^{d}(t-1))/D$ is the average difference between the local cluster centers in two subsequent iterations.

With the consensus on the outputs of FCM (i.e. cluster centers), it is possible to use synergism in partitions without transferring any actual data point and risk security hazards. As shown in the subsequent experiments, this heuristic produces a stable and robust answer for clustering problems and covers the main drawbacks of FCM such as its

Table 3. Average speed up of the proposed EC3 approach over three independent runs.

Dataset	EC3 Speed Up with different number of machines				
	2	***4***	***6***	***8***	***10***
Balance	2.68	5.53	6.50	8.73	20.23
Breastcancer	3.47	7.09	10.36	13.21	17.10
Gamma	2.87	5.53	8.26	10.34	13.05
Glass	18.93	34.87	57.25	74.52	74.92
Haberman	2.30	4.67	7.10	8.59	11.35
Heart	3.03	6.03	9.80	11.71	14.89
Ionosphere	3.36	7.32	11.05	12.95	13.31
Iris	4.66	9.23	12.82	17.19	21.62
Isolet	2.39	5.96	12.53	19.86	30.39
Pendigits	5.40	9.08	11.45	20.50	22.58
Pima	3.43	6.84	10.13	13.54	17.10
Seeds	3.97	6.31	9.72	14.66	15.28
Segment	3.59	6.51	38.75	21.90	23.52
Sonar	2.89	9.98	6.20	11.36	13.29
Tictactoe	2.78	5.56	8.97	11.81	14.04
Waveform	2.93	5.78	8.95	11.07	13.32
Waveform noise	3.37	6.45	9.36	10.68	9.08
Wine	3.79	7.19	10.42	13.48	16.79

sensitivity to noisy data, outliers, and instability.
The present work uses the standard version of FCM to only emphasize the utility of the basic concept in the proposed consensus strategy. Obviously, more recent and effective FCM variations [8, 9] can be used for earning better results.

3. Performance evaluation

The results of implementing the proposed EC3 on 18 different UCI datasets are presented here. After a brief introduction of the employed datasets, the experimental conditions are described. Subsequently, the analysis of the results are presented, and the sensitivity of EC3 to its main hyper-parameter $\mathcal{L}$ is investigated.

3.1. Datasets

For an appropriate comparison against the centralized version of the algorithm, a set of 18 datasets from the UCI repository are considered here. In order to take advantage of the EC3's distributed construction, the original datasets are randomly partitioned over the D machines, forming D separate partitions. Different numbers of machines/partitions, from 2 to 10, are considered in order to study the general applicability of the algorithm under distributed computation. Table 1 shows the datasets and their properties.

3.2. Experimental setup

The proposed algorithm is implemented in MATLAB, using a computer with Quad-Core Core i-7 4.00 GHz CPU, 8GB of RAM and 128GB SSD. In the proposed method, machines are connected only to the consensus unit, i.e. there is no direct communication link among the machines. Also, the connection lines only report the cluster centers from each machine to the consensus center and return the weighted clusters from the consensus process, rather than actual data points.

3.3. Results

Here, we study the clustering algorithm in terms of accuracy and speed-up, in comparison with its centralized FCM counterpart. Specifically, table 2 shows the results in terms of percent accuracy with different numbers of machines. To ensure the robustness of the results, each experiment is repeated three times, and the presented results are the average of these executions. Percent accuracy is computed here as:

$$\%Accuracy = \frac{\#\ of\ \ Correctly\ \ Clusterd\ \ Data}{\#\ of\ \ All\ \ Data} \times 100 \qquad (12)$$

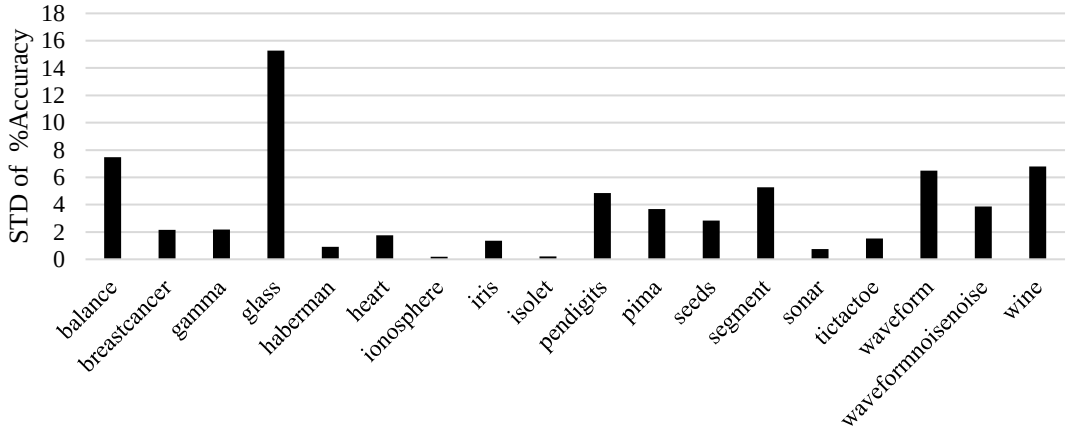

Figure 2. $\mathcal{L}$-Sensitivity of the percent accuracy of the proposed method for various datasets for $\mathcal{L}$={1, …, 10}.

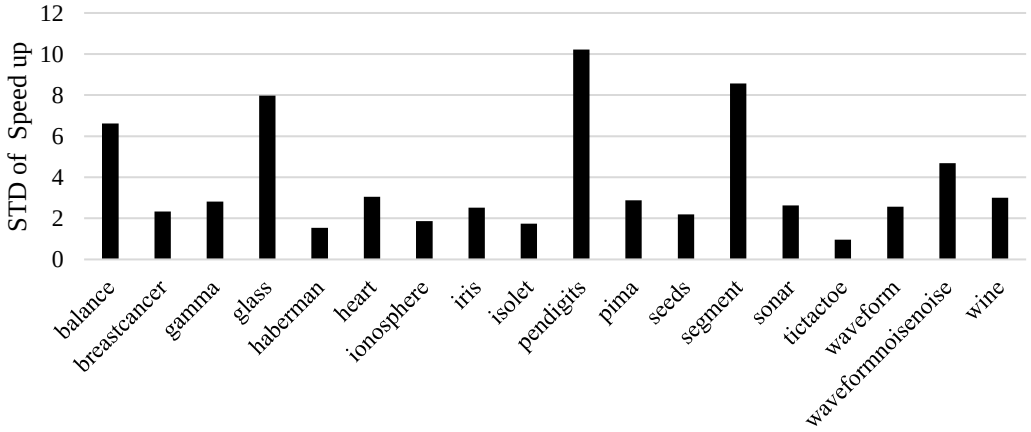

Figure 3. $\mathcal{L}$-Sensitivity of the speed up of the proposed method for various datasets for $\mathcal{L}$={1, …, 10}.

As indicated in table 2, EC3 often has a better average performance in comparison with the centralized FCM. The reason behind this result could be in the synergism, which is created by the data fusion process. Since the information of the data on each machine is incomplete and based on this lacking of knowledge, the variables of local FCMs converge to their local optimal values. However, the weighted average of these locally calculated centers in the consensus center has a better overall performance in comparison with each one of these centers individually or the centralized clustering. The results also demonstrate that the number of machines has no special effect on the overall performance of the EC3 algorithm. Since the number of machines is usually constrained by the specific application, this issue can be considered as the advantage of EC3 in terms of its general applicability.

Execution time is one of the key measures in the distributed systems. At first, it seems that the EC3 execution time should be approximately equal to the centralized FCM divided by the number of machines. However, for two antithetical reasons, this assumption is not entirely correct, i.e. the distribution overhead and synergism in ensemble learning. Distribution overhead is the additional processing that is done only for handling the distributed data processing system, e.g. task scheduling and connection management, while the ensemble learning could reach a faster and more robust convergence by building a synergism in its parallel processing of locally available and smaller set of expertise. Speed-up is employed here to analyze the difference between the execution time of the proposed method and the centralized FCM. This measure is defined as below:

$$speed-up=\frac{Run\ Time_{Proposed\ EC3}}{Run\ Time_{Centralized\ FCM}}. \qquad (13)$$

As the results illustrate here, ensemble synergism is more effective here than the distribution overhead since the execution time is less than the expected value (i.e. the run time of centralized FCM divided by the number of machines). In particular, table 3 shows that the speed-up is considerably higher than the number of processing machines, as examined on the 18 datasets in this

work. These results illustrate the scalability property of the proposed EC3 in terms of higher speed-up with increasing number of machines.

3.4. $\mathcal{L}$-sensitivity analysis

$\mathcal{L}$ is an interesting hyperparameter in that it reflects the number of iterations (self-reflections) that each processor takes before attempting to share the gained insight (consensus) with its network of other processors. Here, we study the sensitivity of the consensus performance and speed-up to $\mathcal{L}$. Fig. 2 presents the standard deviation of clustering accuracy for the various datasets with different values of $\mathcal{L}$ ($\mathcal{L} = \{1, \dots, 10\}$). As it can be seen in this figure, the value of $\mathcal{L}$ has different effects on the clustering accuracy (as high as 15 for the Glass dataset and nearly zero for the Ionosphere and Isolet datasets).

This effect is more consistent in speed-up, as illustrated in Fig. 3. On the 18 datasets in this work, the standard deviation of speedup ranges from 1 to nearly 10. There are few datasets for which $\mathcal{L}$ could be an important performance measure (exceeding 8 for the Pendigits and Segment). Hence, choosing an appropriate $\mathcal{L}$ could be a determining factor when considering various competing techniques on consensus clustering.

To effectively deal with choosing the appropriate $\mathcal{L}$, in this paper, the heuristic in (7) is proposed that is applicable across the various database platforms. This observation was possible since the independent parameter L was already known in the benchmark problems in this work. In general, however, $\mathcal{L}$ is not known a priori. But this fact does not change the conclusions of this paper since, even though significant, variations due to $\mathcal{L}$ settings are considerably smaller than the synergism in the speed-up that is reported.

4. Conclusions and future works

In this work, Entropy-based Consensus on Cluster Centers (EC3) has been proposed for distributed data clustering based on weighted averaging of the cluster centers of all machines. Specifically, we proposed a weighting mechanism that assigned a higher weight to machines with higher internal validity, higher relative validity against other machines, and access to more data samples. Accordingly, three measures were used. First is the sum of entropies of all clusters that is inversely proportional to the internal clustering validation at each machine. Second is the relative values of local FCM cost function that is inversely proportional to the relative clustering validation of one machine with others. And third is the number of data points available to each local machine as its reliability estimate. Hence, the machines with higher estimated validation and reliability are more effective in the consensus procedure.

Application to the 18 benchmarks indicates the consistency of the improvement in terms of speed-up across problems of various sizes. Hence, the approach is scalable. This is while the accuracy in comparison with the centralized FCM algorithm is maintained. In fact, the proposed approach reaches better results in 13 of the 18 benchmark problems. In short, the final cluster centers more stable and robust due to the ensemble learning approach. The main advantage, however we believe, is in its privacy preservation. In other words, the proposed approach reaches these results by only sharing cluster centers, rather than individual data instances.

Finally, we proposed a heuristic to determine the number of iterations $\mathcal{L}$ at each machine before each consensus step. This is a typical problem in consensus strategies since these algorithms must create a balance between their machines' self-reflections and their social consensus. While this problem generally remains unsolved, the formulation here illustrates a relation between the number of iterations in centralized versus decentralized strategies. We believe that further studies of this relation could be an interesting study on the aspects of individual (one machine) vs. social (many machines) processing of information. Overall, the final conclusions of this work remain true regardless of this heuristic, since the speed-up improvement is considerable and the sensitivity study of performance against variations of $\mathcal{L}$ shows a lower affinity towards this value.

It should be mentioned that the proposed method is a consensus algorithm, and the local clustering technique is simply employed only to illustrate its overall value in distributed algorithms. Accordingly, FCM is chosen as a standard and widely-used clustering benchmark algorithm that allows us to keep focus on the main proposed consensus technique. As such, we believe that the impact of using other clustering algorithms could be investigated in future research works. We also aim to study the relative impact of the data set size and distribution versus the number of features. The current dataset sizes are considerably bigger than the number of features. While this is the norm for most clustering problems, there are some uprising application domains such as microarray gene expression databases in which the number of features far exceeds the number of samples. It would be interesting to determine how the proposed clustering measures behave under these different conditions and if they maintain or exceed

their current relative performance levels. As future work, we plan to apply our consensus algorithm to a multi-objective clustering technique [41] to improve the accuracy of clustering on each machine.

5. References

[1] Bezdek, J. C. (2013). Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media.

[2] Gustafson, D. E., & Kessel, W. C. (1979, January). Fuzzy clustering with a fuzzy covariance matrix. In 1978 IEEE conference on decision and control including the 17th symposium on adaptive processes (pp. 761-766). IEEE.

[3] Krishnapuram, R., & Keller, J. M. (1993). A possibilistic approach to clustering. IEEE transactions on fuzzy systems, vol. 1, no. 2, pp. 98-110.

[4] Miyamoto, S., & Umayahara, K. (1998, May). Fuzzy clustering by quadratic regularization. In 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228) (Vol. 2, pp. 1394-1399). IEEE.

[5] Tran, D., & Wagner, M. (2000, May). Fuzzy entropy clustering. In Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063) (Vol. 1, pp. 152-157). IEEE.

[6] Pham, D. L. (2001). Spatial models for fuzzy clustering. Computer vision and image understanding, vol. 84, no. 2, pp. 285-297.

[7] Yang, D. R., Lan, L. S., & Pao, W. C. (2006, August). A new fuzzy entropy clustering method with controllable membership characteristics. In 2006 49th IEEE International Midwest Symposium on Circuits and Systems (Vol. 1, pp. 187-191). IEEE.

[8] Zarinbal, M., Zarandi, M. F., & Turksen, I. B. (2014). Relative entropy fuzzy c-means clustering. Information Sciences, vol. 260, pp. 74-97.

[9] Zarinbal, M., Zarandi, M. F., & Turksen, I. B. (2014). Interval type-2 relative entropy fuzzy C-means clustering. Information Sciences, vol. 272, pp. 49-72.

[10] Izakian, Z., & Mesgari, M. (2015). Fuzzy clustering of time series data: A particle swarm optimization approach. Journal of AI and Data Mining, vol. 3, no. 1, pp. 39-46.

[11] Tsitsiklis, J. N. (1984). Problems in decentralized decision making and computation (No. LIDS-TH-1424). Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems.

[12] Georgopoulos, L., & Hasler, M. (2014). Distributed machine learning in networks by consensus. Neurocomputing, vol. 124, pp. 2-12.

[13] Wu, J., Liu, H., Xiong, H., Cao, J., & Chen, J. (2015). K-means-based consensus clustering: A unified view. IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 1, pp. 155-169.

[14] Vega-Pons, S., Correa-Morris, J., & Ruiz-Shulcloper, J. (2010). Weighted partition consensus via kernels. Pattern Recognition, vol. 43, no. 8, pp. 2712-2724.

[15] Domeniconi, C., & Al-Razgan, M. (2009). Weighted cluster ensembles: Methods and analysis. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 2, no. 4, 17.

[16] Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. International Journal of Pattern Recognition and Artificial Intelligence, vol. 25, no. 03, pp. 337-372.

[17] Dimitriadou, E., Weingessel, A., & Hornik, K. (2001). Voting-merging: An ensemble method for clustering. Artificial Neural Networks—ICANN 2001, 217-224.

[18] Fischer, B., & Buhmann, J. M. (2003). Bagging for path-based clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 11, pp. 1411-1415.

[19] Ayad, H. G., & Kamel, M. S. (2008). Cumulative voting consensus method for partitions with variable number of clusters. IEEE transactions on pattern analysis and machine intelligence, vol. 30, no. 1, pp. 160-173.

[20] Dimitriadou, E., Weingessel, A., & Hornik, K. (2002). A combination scheme for fuzzy clustering. International Journal of Pattern Recognition and Artificial Intelligence, vol. 16, no. 07, pp. 901-912.

[21] Fred, A. L., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 6, pp. 835-850.

[22] Fred, A. (2001). Finding consistent clusters in data partitions. In International Workshop on Multiple Classifier Systems (pp. 309-318). Springer Berlin Heidelberg.

[23] Li, Y., Yu, J., Hao, P., & Li, Z. (2007). Clustering ensembles based on normalized edges. Advances in Knowledge Discovery and Data Mining, pp. 664-671.

[24] Li, T., Ding, C., & Jordan, M. I. (2007). Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In Data Mining. ICDM. Seventh IEEE International Conference on (pp. 577-582). IEEE.

[25] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, vol. 66, no. 336, pp. 846-850.

[26] Ben-Hur, A., Elisseeff, A., & Guyon, I. (2001). A stability based method for discovering structure in clustered data. In Pacific symposium on biocomputing (Vol. 7, pp. 6-17).

[27] Bakus, J., Hussin, M. F., & Kamel, M. (2002). A SOM-based document clustering using phrases. In Neural Information Processing. ICONIP'02. Proceedings of the 9th International Conference on (Vol. 5, pp. 2212-2216). IEEE.

[28] Vega-Pons, S., Correa-Morris, J., & Ruiz-Shulcloper, J. (2008). Weighted cluster ensemble using a kernel consensus function. Progress in Pattern Recognition, Image Analysis and Applications, 195-202.

[29] Scholkopf, B. & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA.

[30] Huang, D., Lai, J. H., & Wang, C. D. (2016). Robust ensemble clustering using probability trajectories. IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 5, pp. 1312-1326.

[31] Wu, Z., Gao, G., Bu, Z., & Cao, J. (2016). SIMPLE: a simplifying-ensembling framework for parallel community detection from large networks. Cluster Computing, vol. 19, no. 1, pp. 211-221.

[32] Liu, H., Cheng, G., & Wu, J. (2015). Consensus Clustering on big data. In Service Systems and Service Management (ICSSSM), 12th International Conference on (pp. 1-6). IEEE.

[33] Kraus, J. M., & Kestler, H. A. (2010). A highly efficient multi-core algorithm for clustering extremely large datasets. BMC bioinformatics, vol. 11, no. 1, pp. 169.

[34] Unold, O., & Tagowski, T. (2015, July). A Parallel Consensus Clustering Algorithm. In International Workshop on Machine Learning, Optimization and Big Data (pp. 318-324). Springer, Cham.

[35] Ngo, L. T., Dang, T. H., & Pedrycz, W. (2018). Towards interval-valued fuzzy set-based collaborative fuzzy clustering algorithms. Pattern Recognition, vol. 81, pp. 404-416.

[36] Lyu, L., Bezdek, J. C., Law, Y. W., He, X., & Palaniswami, M. (2018). Privacy-preserving collaborative fuzzy clustering. Data & Knowledge Engineering.

[37] Wang, Y., & Chen, L. (2017). Multi-view fuzzy clustering with minimax optimization for effective clustering of data from multiple sources. Expert Systems with Applications, vol. 72, pp. 457-466.

[38] Wu, J., Wu, Z., Cao, J., Liu, H., Chen, G., & Zhang, Y. (2017). Fuzzy Consensus Clustering with Applications on Big Data. IEEE Transactions on Fuzzy Systems, vol. 25, no. 6, pp. 1430-1445.

[39] Wu, X., Ma, T., Cao, J., Tian, Y., & Alabdulkarim, A. (2018). A comparative study of clustering ensemble algorithms. Computers & Electrical Engineering, vol. 68, pp. 603-615.

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

[41] Shahsamandi Esfahani, P., & Saghaei, A. (2017). A multi-objective approach to fuzzy clustering using ITLBO algorithm. Journal of AI and Data Mining, vol. 5, no. 2, pp. 307-317.

نشریه هوش مصنوعی و داده‌کاوی

اجماع مبتنی بر آنتروپی در خوشه‌بندی توزیع‌شده فازی

معین اوحدی کارشک و محمدرضا اکبرزاده توتونچی*

گروه مهندسی کامپیوتر، قطب علمی رایانش نرم و پردازش هوشمند اطلاعات، دانشگاه فردوسی مشهد، مشهد، ایران.

ارسال ۲۰۱۶/۰۵/۰۲؛ بازنگری ۲۰۱۷/۰۵/۰۷؛ پذیرش ۲۰۱۸/۰۷/۰۳

چکیده:

رشد سریع حجم و توجه بیش از پیش به حفظ حریم شخصی داده‌ها از چالش‌های دنیای امروز داده‌کاوی است. در این مقاله، اجماع مبتنی بر آنتروپی مراکز خوشه برای خوشه‌بندی داده‌ها در محیط توزیع‌شده، با حفظ محرمانگی داده‌ها، پیشنهاد می‌شود. به بیان دیگر، در روش پیشنهادی، مراکز خوشه در ماشین‌های محلی به جای اصل داده‌ها در فرآیند خوشه‌بندی مورد توافق قرار می‌گیرند. ما از آنتروپی به عنوان معیاری از درستی خوشه‌بندی استفاده می‌کنیم تا ماشین محلی با خوشه‌بندی بهتر، تاثیر بیش‌تری در خوشه‌بندی نهایی داشته‌باشد. ما هم‌چنین از مقدار نسبی تابع هزینه خوشه‌بندی فازی به عنوان معیار معتبربودن و از تعداد داده‌های هر ماشین به عنوان معیار اتکاپذیری استفاده می‌کنیم تا میزان تاثیر هر ماشین محلی را مشخص کنیم. کارایی روش پیشنهادی در مقایسه با روش خوشه‌بندی فازی متمرکز با استفاده از ۱۸ پایگاه داده UCI با معیارهای صحت خوشه‌بندی و نرخ افزایش سرعت مورد بررسی قرا گرفت. آزمایش‌های متعدّد نشان می‌دهد که استفاده از روش پیشنهادی، ضمن حفظ امنیّت داده‌ها، باعث بهبود هر دو معیار کارایی ذکرشده می‌شود.

کلمات کلیدی: آنتروپی، اجماع در خوشه‌بندی، خوشه‌بندی توزیع‌شده، خوشه‌بندی فازی، یادگیری مبتنی بر اجتماع.