International Journal of Information Science and Management
Vol. 18, No. 2, 2020, 215-228

A Distributed Clustering Approach for Heterogeneous Environments Using Fuzzy Rough Set Theory

Niloofar Mozafari
Department of Designing & System Operation, Regional Information Center for Science and Technology, RICeST, Shiraz, IRAN
Corresponding Author: mozafari@ricest.ac.ir

Mohammad-Ali Nikouei Mahani
Institute for Physiology, University of Tuebingen, Germany
nikouei@ut.ac.ir

Sattar Hashemi
Department of Computer Science and Engineering and Information Technology
School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
s_hashemi@shirazu.ac.ir

∗ Received: 01 May 2020
Accepted: 23 June 2020

Abstract

Vast majority of data mining algorithms have been designed to work on centralized data, unfortunately however, almost all of nowadays data sets are distributed both geographically and conceptually. Due to privacy and computation cost, centralizing distributed data sets before analyzing them is undoubtedly impractical. In this paper, we present a framework for clustering distributed data which takes into account privacy and computation cost. To do that, we remove uncertain instances and just send the label of the other instances to the central location. To remove the uncertain instances, we develop a new instance weighting method based on fuzzy and rough set theory. The achieved results on well-known data verify effectiveness of the proposed method compared to previous works.

Keywords: Distributed Clustering, Fuzzy Rough Set Theory, Data Distributed Mining.

Introduction

Today Data Distributed Mining which, abbreviated as DDM has received great deal of attention by researchers. There are many factors which led to the evolution of DDM: privacy, transmission and memory cost. The goal of DDM is to extract useful information from data located at heterogeneous sites (Clifton, Kantarcioglu, Vaidya, Lin & Zhu, 2002).

One of the most important techniques in data mining is data clustering (Jain, Murty & Flynn, 1999). In the data distributed clustering, whole data (which is distributed in several sites) is partitioned into different groups or clusters, so that data which are in same cluster have most similarity and data in different clusters have most dissimilarity. This (dis)similarity depends on the application domain.

In distributed environments, data can be distributed into two aspects (Strehl & Ghosh, 2002):

1- Homogeneous: data is distributed horizontally across the sites and each site has access

to a subset of instances. In data distributed clustering, it is also denoted as Object Distributed Clustering (ODC).

2- Heterogeneous: data is distributed vertically across the sites and therefore each site has access to a subset of features that in data distributed clustering, it is known as Feature Distributed Clustering (FDC).

The problem of clustering in a distributed environment, is explored in (Kergupta, Hamzaoglu & Stafford, 1997) for the first time. In that paper, authors proposed a hierarchical clustering algorithm that composes of three components, namely a user interface, a facilitator and independent agents with their own storage. A local clustering is performed on each agent and the results are sent to a client.

Samatova, Ostrouchov, Geist & Melechko (2005) presented another hierarchical clustering in distributed environments that send a representative from each cluster to a central location. This representative composes from some statistical information such as the number of data points in the cluster, the square norm of the centroid, the radius of the cluster, the sum of the components and their minimum and maximum value. In the central location, different clusterings are merged using just these representatives.

In (Johnson & Kargupta, 2000), authors proposed a method for clustering in heterogeneous distributed data and called their method CHC (Collective Hierarchical Clustering). Each site which has access to subset of all features performs a local hierarchical clustering and then sent the obtained dendograms to central location. In the central location, the global model is computed by statistical bounds. Although the similarity between the aggregated results and the centralized clustering results make CHC a good distributed clustering algorithm but it does not specifically address privacy of data. There are also some methods for distributed clustering in homogeneous data that work well in distributed environments but they do not specifically address the privacy issues (Tasoulis & Vrahatis, 2004), (Dhillon & Modha, 2002).

Strehl & Ghosh (2002) proposed an ensemble clustering method. The method works in either heterogeneous or homogenous environments. Each node performs a clustering algorithm which may be different from the other nodes and then the clustering result is sent to the central location. In central location the received results from different nodes are combined using a combiner. They have proposed three combiners. Due to the low complexity of these combiners, it is feasible to run all the proposed combiners and the best result is chosen.

Zhao & Sayed (2015) have proposed an adaptive clustering method that allows instances to learn which neighbors they should cooperate with and which others should be ignored. A clustering technique for large spatial datasets in heterogeneous environments has been proposed in (Bendechache & Kechadi, 2015). That method is based on k-means algorithm and then aggregates the result in elaborated aggregation phase. Awatshi et al. (2017) proposes a general framework for designing distributed clustering algorithms. They provide conceptually simple distributed algorithms, combined with a new analysis, to paint a unifying picture for distributed clustering. A density based clustering in distributed environments was proposed in (Santos, Syed, Naldi, Campello & Sander, 2019).

In this paper, we propose an algorithm for data clustering in heterogeneous distributed environments. Our algorithm takes into account privacy of data and computation cost. Each node partitions instances into different clusters and assigns a label to each instance, so that all instances in one cluster have the same label. In the next step, the proposed algorithm selects a portion of labels and sent them to central location. For selecting the appropriate labels, we

propose a new instance weighting method based on fuzzy and rough set theory. To the best of our knowledge, most of instance weighting algorithms used distance between instances as a criterion for weighting which is not suitable for high dimensional datasets. To address this issue, we use a new modification of this theory called Fuzzy Rough set Instance Weighting (FRIW). After weighting the instances based on FRIW, instead of sending the entire data, the label of instances with higher weights are just sent to the central location. In the central location, the selected labels of instances are combined using clustering ensemble technique. We compare our method with ensemble clustering and also fuzzy clustering methods. The achieved results on well-known data like Ecoli, Pendig, and segmentation verify effectiveness of the proposed method compared to previous works.

The reminder of this paper is organized as follow. Theoretical background on rough set theory will be discussed in Section 2. In Section 3, the proposed algorithm is presented. Our experimental results are given in Section 4 and Section 5 concludes the paper by a conclusion part and presents the future work.

Theoretical background on rough set

Rough set theory as a methodology of database mining in the relational databases, was first introduced by Pawlak in 1982. It can be used for discovering structural relationship within imprecise and noisy data.

Rough set theory is closely related to fuzzy theory and both of them are complementary generalization of classical sets. The approximation spaces of rough set theory are sets with multiple memberships, while fuzzy sets are concerned with partial memberships. The basic problem in data analysis solved by Rough set theory is finding dependency between the features (Pawlak, 1982).

In Rough set, data model information is stored in a table. Each row shows a fact which are not consistent with each other. In Rough set terminology, a data table is called Information System (*IS*) (Cornelis, Medina & Verbiest, 2014). *IS* can represent as a pair of instances and features, $IS = (U, F)$ where U is a set of instances and F is a set of features or attributes. In many applications, there exists a feature which is called the class label. It is the outcome of the classification algorithm which is always known in train dataset. This feature is called decision feature. If an information system table contains the decision feature (class label), it sometimes called decision system table (*DS*) instead of *IS*.

Table 1 shows an example of a *DS* for covid-19 with four features and a decision feature.

Table 1
An example of DS for covid-19

	Features				Decision
Case	Temperature	Headache	Nausea	Cough	Covid-19
1	Very High	Yes	No	Yes	Yes
2	High	Yes	Yes	Yes	No
3	High	Yes	Yes	Yes	Yes
4	Normal	Yes	No	No	No
5	Normal	No	No	Yes	No
6	Normal	Yes	Yes	Yes	Yes

Most times, a decision table expresses all information about the system and has redundant data.

The indiscernible or resemble instances may represent several times in the Table. Also, this redundancy may be seen in the features. Let $P \subset F$ be some features of information table, *U* be set of all instances, then the equivalence relation, *IND (P), is as follows* (Thilagavathy & Rajesh, 2011):

$$IND(P) = [x]_P = \{(x, y) \in U^2 | \forall a \in P, a(x) = a(y)\} \quad (1)$$

Where $IND(P)$ is called indiscernibility of the relation. It means x and y are indiscernible from each other by features P. In covid-19 example, if $p = \{Tempreture, Headache\}$, then $IND(P) = \{\{1\}, \{2,3\}, \{4,6\}, \{5\}\}$. It means instances 2 and 3 are indiscernible from each other with these two features. Instances 4 and 6 are also indiscernible from each other by using these two features.

As another example of calculating indiscernibility is shown below:

$p = \{Tempreture, Headache, \text{Nausea}, \text{Cough}\}$

$IND(P) = [x]_P = \{\{1\}, \{2,3\}, \{4\}, \{5\}, \{6\}\}$

It means instances 2 and 3 are indiscernible from each other by using all features. This redundancy helps us to detect similar instances.

Rough set concept can be defined quite generally by means of topological operations called approximations. Any subset of objects like $X \subseteq U$ can be approximated using any subset of features like $P \subseteq F$ by defining *P-lower bound* and *P-upper bound* of *X* (Thilagavathy & Rajesh, 2011).

$$\underline{P}X = \{x] | [x]_P \subset X\} \quad (2)$$
$$\overline{P}X = \{x | [x]_P \cap X \neq \emptyset\}$$

Where $\underline{P}X$ is P-lower bound for X, and $\overline{P}X$ is P-upper bound for *X*. Pair of $\underline{P}X$ and $\overline{P}X$ is called rough set (Zhao, Wang, Hu & Zhu, 2019). Difference between P-upper and P-lower bound is boundary region of *X* denoted by *BNDP (X)* (Thilagavathy & Rajesh, 2011).

$$BND_p(X) = \overline{P}X - \underline{P}X \quad (3)$$

Following is as example which shows how upper, lower and boundary regions are calculated. Suppose that X is the set of instances who are infected with covid-19 ($X = \{1,3,6\}$).

$p = \{Tempreture, Headache, \text{Nausea}, \text{Cough}\}$

$IND(P) = \{\{1\}, \{2,3\}, \{4\}, \{5\}, \{6\}\}$

$\underline{P}X = \{1,6\}$

$\overline{P}X = \{1,2,3,6\}$

$BND_p(X) = \overline{P}X - \underline{P}X = \{2,3\}$

A set is rough if its boundary region is non-empty. Upper bound, lower bound and boundary region can help us to know the instances better. Instances in P-lower bound of *X* can certainly classify according to features in *P* but instances in P-upper bound of *X* possibly can be classified with features in *P*. The instances in *BNDP(X)* cannot be classified with features in *P*. We use three regions to specify the importance of instances for difference classes.

The proposed algorithm

Suppose that there are *r* sites which each one has access to a subset of features. In the first step, instances are partitioned with any clustering algorithm in each node and then all instances in the same cluster take one label. Instead of sending all instances to the central location, a portion of label of instances are sent. The proposed algorithm removes two types of instances and sends the label of the others to the central location. Those instances that our algorithm tries to remove them are uncertain instances, namely instances in the border of clusters and outliers. In order to remove the uncertain instances, we develop a new instance weighting method based on fuzzy and rough set theory. In the following, we describe how our method removes the uncertain instances in each site.

Our method uses rough set theory to weight the instances. As it is apparent, rough set theory can be applied only on nominal features, therefore to use it for continues features, a discretization method must be applied on continues features to prepare them for next processes. Despite of the fact that many discretization methods were proposed (Fayyad & Irani, 1993), (De Sá, Soares & Knobbe, 2016), (Nojavan, Qian & Stow, 2017), most of them had used the same criteria, decrease entropy and increase information gain. Even though the discretization made by these methods increase the information gain, they cannot handle uncertainly as fuzzy methods, in view of the fact that no instance can be discretized in multiple sets.

In this paper, we use from fuzzy sets to change continues features to nominal because it can consider dependency degree of each instance to its discretized set. Our Fuzzy Rough set Instance Weighting (FRIW) gives weight to each instance and instead of sending the entire data, the label of instances with higher weights are just sent to the central location.

Orthogonal triangular fuzzy membership functions discretize each feature. Weight of the instance x in *U/IND (a)* denoted by $W_{IND(a)}(x)$ and defined by Equation 4. It is obtained according to dependency of instance *x* to μ_{a_i} which is the degree of *x* in i^{th} membership function on feature A.

$$W_{IND(A)}(x) = \{\mu_{A_i}(x) \,|\, \mu_{A_i}(x) > 0\} \tag{4}$$

Definition (4) can be extended to a subset of features. In case, *P* is any subset of features, the weights of *x* in *U/IND (P)* would be defined as follow:

$$W_{IND(P)}(x) = \Theta\{W_{IND(A)}(X) \,|\, A \in P\} \tag{5}$$

$$A\Theta B = \{W_a(x) * W_b(x) \,|\, a \in A, b \in B, W_a(x) * W_b(x) \neq 0\}$$

Our method defines the weight of instances in each upper bound according to $W_{IND(P)}(x)$ as follow:

$$W_{\overline{PC}_i}(x) = \begin{cases} \dfrac{\sum_{y \in [x]\overline{PC}_i, label(y)=i} W(y)}{\sum_{y \in [x]\overline{PC}_i} W(y)} & if \quad x \in [x]\overline{PC}_i, label(x) = i \\ \dfrac{\sum_{y \in [x]\overline{PC}_i, label(y)\neq i} W(y)}{\sum_{y \in [x]\overline{PC}_i} W(y)} & if \quad x \in [x]\overline{PC}_i, label(x) \neq i \end{cases} \tag{6}$$

In equation (6), $W_{\bar{P}C_i}(x)$ is the weight of x in upper bound of class i. $[x]_{\bar{P}C_i}$ are equivalence classes which make $\bar{P}C_i$ or upper bound of class i.
The final weight of each instance in our method is determined by equation 7 as follows:

$$W(x) = 1 - \min_{1<i<N} W_{\bar{PC}_i}(x) \quad (7)$$

Where N is the number of clusters and $W(x)$ is the final weight of x. instances can be selected by cutting them according to a proper threshold value on their weights.

For combining the different clustering results, meta-graph is constructed which is an undirected graph that each vertex of it is a hyper-edge and the number of vertices is as the number of hyper-edges (Strehl & Ghosh, 2002).

The edge weights are proportional to the similarity between vertices or hyper-edges. We use Jaccard measure in order to get similarity between vertices. It is the ratio of the intersection to the union of the sets of instances. Let $w_{a,b}$ be the edge weight between two vertices h_a and h_b. Then Jaccard index between them is calculated as follows (Lee, 2017):

$$w_{a,b} = \frac{|h_a \cap h_b|}{|h_a \cup h_b|} \quad (8)$$

This similarity metric considers relative number of common items. In the second step, the matching labels are found by partitioning the meta-graph into a number of balanced meta-clusters (ibid). As it is stated before, each vertex in the meta-graph is a hyper-edge. On the other hand, each cluster transforms to one hyper-edge. Hence, each vertex in the meta-graph represents a distinct cluster label and a meta-cluster represents a group of corresponding labels. In the third step, meta-clusters are collapsed. For each of the *k* meta-clusters, the hyper-edges are collapsed into a single meta-hyper-edge. Each meta-hyper-edge has an association vector which contains an entry for each object describing its level of association with the corresponding meta-cluster. In the last step, each object is assigned to its most associated meta-cluster.

Experimental Results and Discussion

This section is composed of two subsections, precisely covering our observation and analysis. The first subsection presents experimental setup and evaluation measures. The latter one presents and analyses the obtained results.

Experimental Setup

Data sets

We examine our proposed algorithm on five data sets in different domains with different number of instances and features. Table 2 gives the names and characteristics of the used data sets. Iris, Ecoli, Pendig and Segmentation are publicly available from the UCI Machine Learning Repository (Asuncion & Newman, 2007). 8d5k data set is available for download at http://strehl.com. It involves 1000 instances from five Gaussian distribution on eight dimensions. All clusters have same variance (0.1), but different means. Means were given

randomly from a uniform distribution.

Table 2
The used data sets and their characteristics, data sets select in different domains with different number of features, instances and classes.

Data Set	#Instances	#Features	# Class
Iris	150	4	3
Ecoli	336	7	8
8d5k	1000	8	5
Segmentation	2310	19	7
Pendig	7494	16	10

Evaluation Measure

To evaluate the accuracy of proposed method, we focus on Normalized Mutual Information (NMI) which is an information theoretic measure (Estévez, Tesmer, Perez & Zurada, 2009).

$$NMI(\lambda^a, \lambda^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{l=1}^{k^{(b)}} n_{h,l} \log(\frac{n.n_{h,l}}{n_h^{(a)} n_l^{(b)}})}{\sqrt{(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^{(a)}}{n})(\sum_{l=1}^{k^{(b)}} n_l^{(b)} \log \frac{n_l^{(b)}}{n})}} \quad (9)$$

In this formula, λ^a is true the label of instances and λ^b is the result of clustering using the proposed method. $k^{(a)}$ and $k^{(b)}$ are the number of clusters in λ^a and λ^b respectively. $n_h^{(a)}$ is the number of data in cluster h^{th}. $n_{h,l}$ defines the set of common instances in cluster h and l^{th}.

Results and Discussion

In heterogeneous distributed environment, data is distributed vertically across the sites and therefore each site has access to a subset of features. Each node partitions data into different clusters based on any clustering algorithm and assigns a label to each data, so that all data in one cluster have same label. If two objects have same label, we can conclude that they have been in one cluster. To simulate such a scenario in our experiments, we have six sites that each one has access to a subset of features. In each site, we run k-means clustering algorithm on data with available features. Note that our algorithm is independent of used clustering algorithm. Each site partitions data and sends the clustering results (labels) to the central location. In next step, a portion of labels is selected and sent to the central location. We use FRIW to weight instances in each site and select the labels of instances with higher weight. Instead of sending the entire instances with all of their features to central location, the labels of selected data are just sent. Thus the privacy is considered with sending the labels of instances instead of data with all their features and also with selecting a portion of labels using FRIW.As a Result, the computation cost in the central location is decreased.

The proposed method removes two types of instances. These two types involve instances which each site has the lowest uncertainty about them. The first type involves outliers. Outliers are the observations that deviated from the rest of data (Tang & He, 2017), (Domingues, Filippone, Michiardi & Zouaoui, 2018). So in FRIW, they belong to separate membership function in each feature. If each site has access to F_i features, they may belong to 2^{F_i}

indiscernibilities. Since outliers are deviated from the rest of data, they belong to same cluster in all indiscernibilities and are upper bound of same cluster. Therefore, they have lower weight rather than the other instances. The second type of removed instances in FRIW involves instances which are in the boundary of clusters.

We use a synthetic data with 22 instances and 2 features to illustratively show how our method gives the weight to each instance. Figure 1 illustrates the weight of instances in FRIW for this test data set. As it is obvious, in this data set outlier instances which are far from the core (central) of the cluster, have the minimum weights. These instances are ignored and removed in the first place. In the second place, the boundary instances are omitted on account of the fact that they have lower weight in comparison to the core (central) instances. In FRIW approach, the central instances are the worthiest, in contrast to the outlier and boundary instances. According to Section 4, Equation 5 and 6, since the numbers of outlier instances are less than the other types and also they are far from the central of the cluster, their dependency degree in the final equivalence cluster will be less than central and boundary instances. Consequently, they will be omitted in the first step. Boundary instances are the next which are omitted in FRIW approach. Due to the fact that boundary instances mostly belongs to the equivalence clusters which have many different cluster instances, their weight are less than central ones. As a consequence, the boundary instances will be omitted in the second step. This weighting order is obvious in the Figure 1.

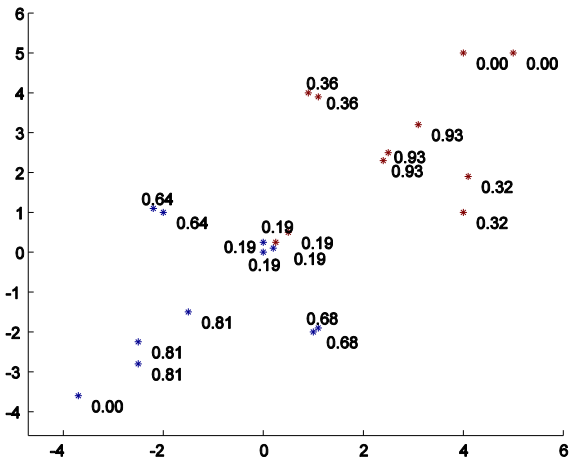


Figure 1. The weight of instances in FRIW. Outlier and boundary instances take low weight. Because outliers are less than other types and also they are far from the central of the cluster, their dependency degree in the final equivalence cluster will be less than central and boundary instances. Boundary instances mostly belong to the equivalence clusters which have many different cluster instances, their weight are less than central instances.

We examine the proposed algorithm on Iris, Ecoli, 8D5K, Segmentation and Pendig data sets. As we discussed previously, outliers have lowest weight in our algorithm. The proposed method avoids sending label of outliers to the central location. Table 3 illustrates the performance of our algorithm compared to Strehl & Ghosh's method and also fuzzy clustering method (Silva Filho, Pimentel, Souza & Oliveira, 2015).

Table 3
The performance of the proposed method in comparison of the others.

Data Set	Strehl& Ghosh method	Fuzzy clustering	The proposed method
Iris	0.7743	0.75	0.8226
8D5K	0.8699	0.88	0.9056
Ecoli	0.4678	0.56	0.5112
Pendig	0.5008	0.5714	0.5207
Segmentation	0.5465	0.5101	0.5867

Figure 2 is an illustrated example that shows how the accuracy improves with removing the outlier instances in each site. In this figure, there are three sites that each one has access to a subset of features. Each site has access to two features. Based on available features in each site, X_1 is outlier in sites 1, 2. But it is similar to X_5, X_6 and X_7 in site 3. Each site votes to X_1 based on its available features. Sites 1 and 2 vote X_1 to cluster that involves X_2, X_3 and X_4. But site 3 votes X_1 to cluster that involves X_5, X_6 and X_7. Without removing outlier in sites 1 and 2, the vote of sites 3 is dominated by incorrect vote of site 1 and 2. So with removing outliers in distributed clustering, the accuracy is increased.

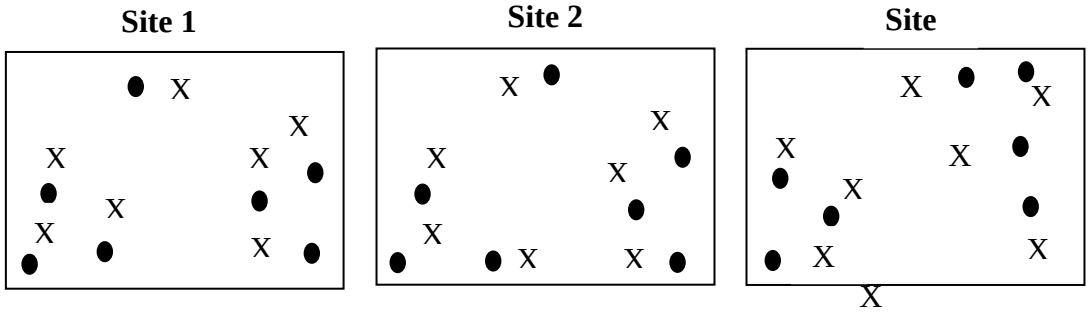
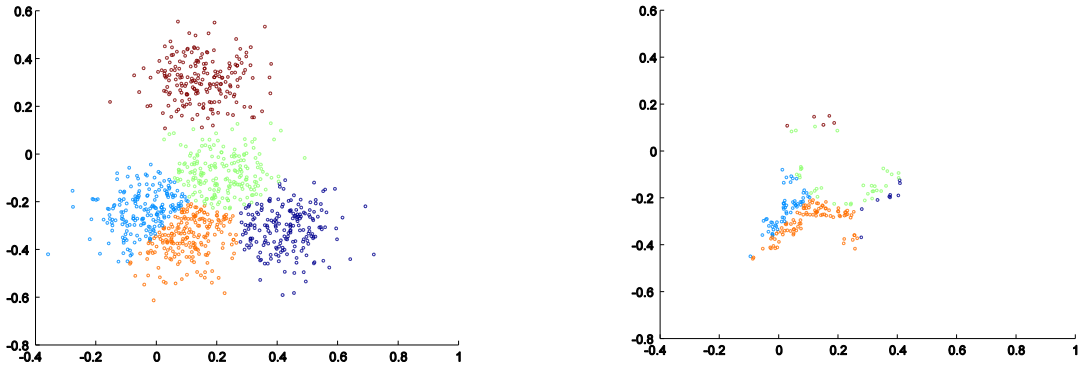


Figure 2. An illustration example that shows removing outlier instances in the sites can improve the accuracy of clustering in the central location

The second type of instances that our algorithm avoids sending them to the central location is boundary instances. Boundary instances are in the boundary of clusters where the clustering algorithm has the low uncertainty about true cluster that instances should belong to. We omit the label of boundary instances to central location. In order to show how FRIW is able to select boundary instances, we examine FRIW on 8d5k data set. 8d5k has eight dimensions and five clusters. We project instances on two principal dimensions. The left side of Figure 3 illustrates the scatter view of instances in two dimensions. The right side of this figure shows the scatter view of instances after selecting boundary instances with FRIW. It is apparent that FRIW can find the boundary instances according to their weights. This promising experiment shows that FRIW can be used as standalone boundary detection algorithm for other purposes in future.



(a) (b)

Figure 3. The power of FRIW in finding the boundary instances in 8D5K data sets. All instances are projected into two principal dimensions. It is apparent FRIW can find the boundary instances according to their weights.

Table 4 shows the percentage of reduced data when the boundary instances are removed. As this table indicates with removing the label of boundary instances, too many instances in the central location are removed and the accuracy does not change. For example, in Pendig data set; with removing the boundary instances; the number of label of instances in the central location decreases from 44964 to 25526. It is obvious that removing 56% of instances in the central location saves the computation cost and storage.

Table 4
The performance of the proposed method in comparison of previous work.

Data Set	# instances in the central location in Streh's method	Accuracy of clustering in Strehl's method	# instances in the central location in the proposed method	Accuracy of clustering in the proposed method	Percentage of reduction data
Iris	900	0.7842	480	0.7848	46.67%
8D5K	6000	0.8889	3300	0.8102	45.00%
Ecoli	2000	0.4634	900	0.4664	55.00%
Pendig	44964	0.4256	25526	0.4273	43.23%
Sengmentation	13860	0.5167	8000	0.5014	42.27%

Figure 4 (a-d) illustrates the accuracy of the proposed method on the used data sets. The horizontal axis shows the number of selected instances and the vertical axis illustrates the accuracy. In Figure 4-a, we examine the proposed method on Iris data set. There are six sites that each one has 150 instances. If each site sends all data, there would be 900 instances with all their features in central location. In the proposed method, each site sends the label of each data instead of data with all features and also it sends a portion of labels to central location, so the computation cost and privacy is considered. The accuracy with all data is 77%. As the number of selected data achieves near 650, the accuracy gets near 85%. The reduced instances in this step are outliers. Selecting near 200 instances leads to 66% accuracy. These types of instances involve instances which are in the boundary of clusters. So removing these instances has little effect on the boundary of clusters. After removing the two types of instances which we called them outliers and boundary instances, the just remaining instances are central instances. In this data set, with selecting the central instances, it just remains 200 labels in central location instead of 900 labels.

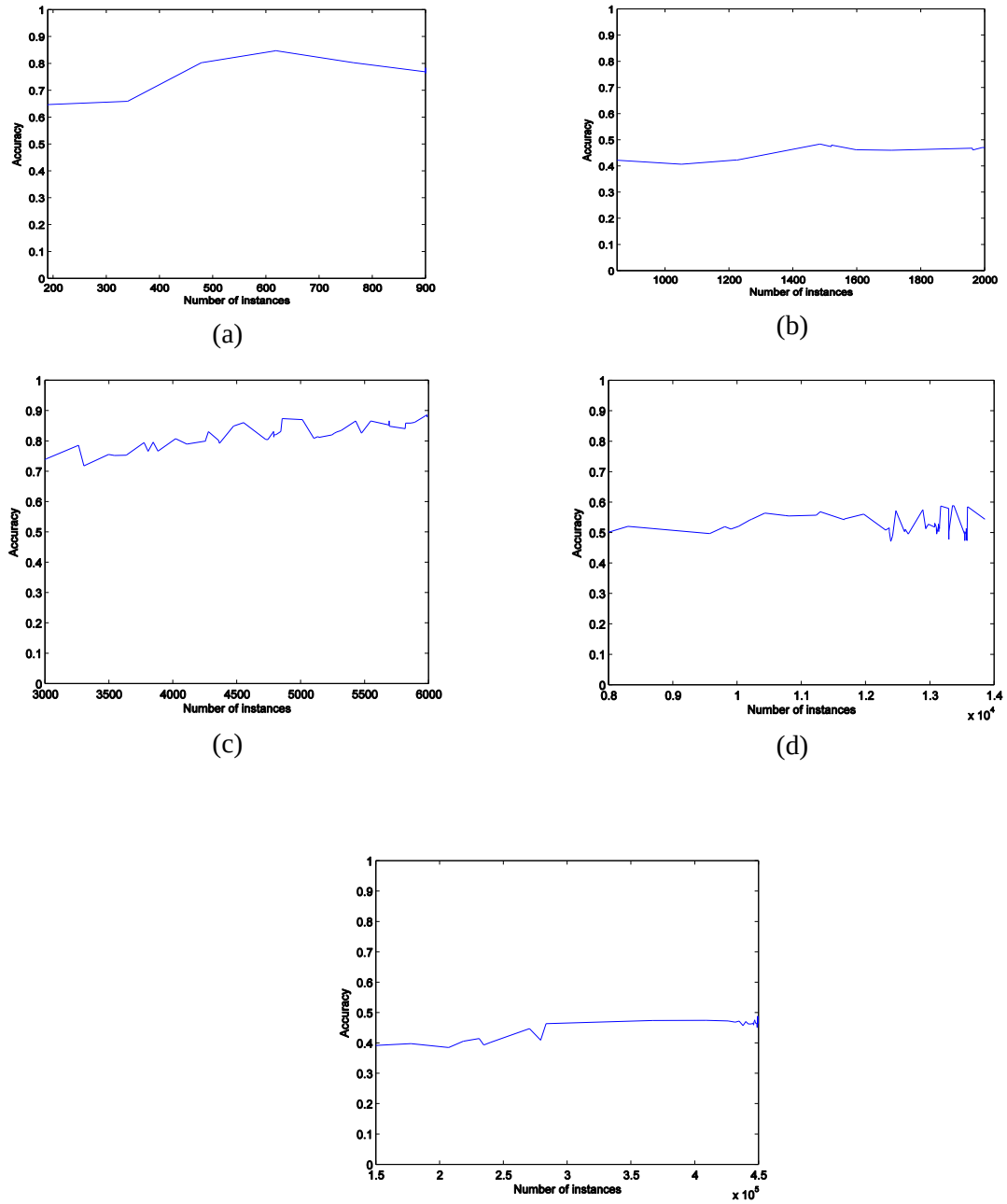


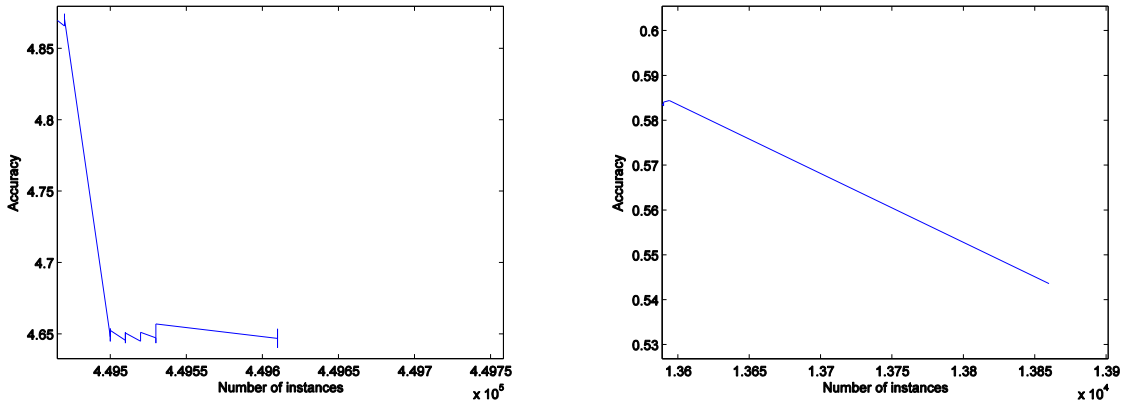
Figure 4. The accuracy of clustering result in the central location in different data sets a) Iris, b) Ecoli, c) 8D5K, d) Segmentation, e) Pendig. With removing the large number of instances in the central location, the accuracy nearly remains fixing. The removing instances that does not effect on the clustering accuracy in the central location are uncertain instances, namely outliers and boundary instances.

Figure 4-b illustrates the accuracy of the proposed method on Ecoli data set. The accuracy increased by selecting near 1500 labels. These instances are the outliers. If each site sends label of non-boundary instances to central location instead of all instances, the accuracy decreases 0.01%. So the proposed method decreases communication cost. Figure 4-c shows the accuracy of the proposed method on 8d5k data set. The trend is similar to the other data sets, but duo to the property of this data set, it has some fluctuations. As it can be seen in Figure 4-c, the accuracy of clustering drops a little where about 50% of instances were reduced. Figure 4-d

shows the accuracy of proposed method on Segmentation data set. In this experiment, each site has access to a subset of features and 2310 instances. If each site sends all data, there exist 13860 instances with all features in central location. In the proposed method, each site sends the labels of each data instead of data with all features and also it sends a portion of labels to central location, thus the computation cost and privacy is considered. As the number of selected data achieves near 13600, the accuracy is increased. It is notable that, in order to select the appropriated label of instances, the proposed algorithm takes the same threshold for all sites. Due to that, some data bases such as 8D5K and Segmentation have some fluctuations.

The next experiment that is illustrated in Figure 4-e, we apply our algorithm on Pendig data set. This data set has 7494 instances. If each site sends all data, it exist 44964 instances with all features in central location. But each site sends the labels of each data instead of data with all features and also it sends a portion of labels to central location. As the number of selected data achieves near 44940, the accuracy is increased. With selecting near 25000 instances in comparison of 44964 (50% of instances), the accuracy changes a little. These types of instances are boundary instances.

Figure 5 (a, b) shows the trend of increasing the accuracy with removing the outlier instances in Pendig and Segmentation data sets.



(a) (b)

Figure 5. The effect of removing outlier instances in increasing the accuracy of clustering in a) Pendig, b) Segmentation data sets.

Remarks on privacy

In the proposed algorithm for clustering in heterogeneous environments, each site that has access to a subset of features runs clustering algorithm independently and assign a label to each instance. So that all instances in the same cluster have one label. Instead of sending all instances with all of their features, each site just sends a portion of label of instances to the central location. Each site has not any knowledge about the features of the other site or the clustering algorithm that the others used. Thus in the proposed algorithm, privacy is considered.

Conclusion and future work

In the recent years, Data Distributed Mining is an attractive research in data mining. Clustering as the most important technique in the data mining has been interested in distributed environment. Data distributed clustering is partition whole data which is distributed in several sites into different groups or clusters, so that data which are in same cluster have most similarity

and data in different clusters have most dissimilarity.

In this paper we present a framework for clustering data in heterogeneous distributed environments which takes into account privacy of data and computation cost. Each node partitions data into different clusters based on any clustering algorithm and assigns a label to each data, so that all data in one cluster have same label. In next step, the proposed algorithm selects a portion of labels and sent them to central location.

In order to select the appropriate labels, we propose a new instance weighting method based on fuzzy and rough set theory. The proposed method removes two types of instances. These two types involve instances which each site has the lowest uncertainty about them. The first type involves outliers. In FRIW, they belong to separate membership function in each feature. If each site has access to F_i features, they may belong to 2^{F_i} indiscernibilities. Since outliers are deviated from the rest of data, they belong to same cluster in all indiscernibilities and are upper bound of same cluster. Therefore, they have lower weight rather than the other instances. The second type of removed instances in FRIW involves instances which are in the boundary of clusters.

After weighting the instances based on the proposed method, instead of sending the entire data, the label of instances with lower weight are just sent to central location. As a Result, the computation cost in the central location is decreased. Also the privacy is considered with sending the labels of instances instead of data with all their features and also with selecting a portion of labels using FRIW. In other words, the proposed method, each site has not any knowledge about the features of the other site or the clustering algorithm that the others used.

As the experimental results indicated the proposed algorithm for Fuzzy Rough set Instance Weighting (FRIW) will be able to applicable to other related instance weighting environments. For the future work, we will investigate applying the proposed method on dynamic environments.

References

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. Retrieved from http://www.ics.uci.edu/~mlearn/MLRepository.html

Awasthi, P., Balcan, M. F. & White, C. (2017). General and robust communication efficient algorithms for distributed clustering. arXiv preprint arXiv:1703.00830, 1.

Bendechache, M &, Kechadi, M. T. (2015, July). Distributed clustering algorithm for spatial data mining. In 2015 2nd *IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM)* (pp. 60-65). IEEE.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. & Zhu, M. Y. (2002). Tools for privacy preserving distributed data mining. *ACM Sigkdd Explorations Newsletter*, 4(2), 28-34.

Cornelis, C., Medina, J. & Verbiest, N. (2014). Multi-adjoint fuzzy rough sets: Definition, properties and attribute selection. *International Journal of Approximate Reasoning*, 55(1), 412-426.

De Sá, C. R., Soares, C. & Knobbe, A. (2016). Entropy-based discretization methods for ranking data. *Information Sciences*, 329, 921-936.

Dhillon, I. S. & Modha, D. S. (2002). A data-clustering algorithm on distributed memory multiprocessors. In *Large-scale parallel data mining* (pp. 245-260). Springer, Berlin, Heidelberg.

Domingues, R., Filippone, M., Michiardi, P. & Zouaoui, J. (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, 74, 406-421.

Estévez, P. A., Tesmer, M., Perez, C. A. & Zurada, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2), 189-201.

Fayyad, U. & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999). Data clustering: a review. A*CM computing surveys* (CSUR), 31(3), 264-323.

Johnson, E. L., & Kargupta, H. (2000). Collective, hierarchical clustering from distributed, heterogeneous data. In *Large-Scale Parallel Data Mining* (pp. 221-244). Springer, Berlin, Heidelberg.

Kergupta, H., Hamzaoglu, I. & Stafford, B. (1997). Scalable, Distributed Data Mining Using An Agent Based Architecture (PADMA). *In Proceeding* of *High Performance Computing* (Vol. 97).

Lee, S. (2017). Improving jaccard index for measuring similarity in collaborative filtering. In International Conference on Information Science and Applications (pp. 799-806). Springer, Singapore.

Nojavan, F., Qian, S. S., Stow, C. A. (2017). Comparative analysis of discretization methods in Bayesian networks. *Environmental Modelling & Software*, 87, 64-71.

Pawlak, Z. (1982). Rough sets. *International journal of computer & information sciences*, 11(5), 341-356.

Santos, J., Syed, T., Naldi, M. C., Campello, R. J. & Sander, J. (2019). Hierarchical Density-Based Clustering using MapReduce. *IEEE Transactions on Big Data*. doi: 10.1109/TBDATA.2019.2907624

Samatova, N. F., Ostrouchov, G., Geist, A. & Melechko, A. V. (2002). RACHET: An efficient cover-based merging of clustering hierarchies from distributed datasets. *Distributed and Parallel Databases*, 11(2), 157-180.

Silva Filho, T. M., Pimentel, B. A., Souza, R. M. & Oliveira, A. L. (2015). Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. Expert Systems with Applications, 42(17-18), 6315-6328.

Strehl, A. & Ghosh, J. (2002). Cluster ensembles--a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3, 583-617.

Tang, B. & He, H. (2017). A local density-based approach for outlier detection. *Neurocomputing*, 241, 171-180.

Tasoulis, D. K. & Vrahatis, M. N. (2004). Unsupervised distributed clustering. In *Parallel and distributed computing and networks* (pp. 347-351).

Thilagavathy, C. & Rajesh, R. (2011, April). A note on rough set theory. In *2011 3rd International Conference on Electronics Computer Technology* (Vol. 6, pp. 39-41). IEEE.

Zhao, X. & Sayed, A. H. (2015). Distributed clustering and learning over networks. *IEEE Transactions on Signal Processing*, 63(13), 3285-3300.

Zhao, H., Wang, P., Hu, Q. & Zhu, P. (2019). Fuzzy Rough Set Based Feature Selection for Large-Scale Hierarchical Classification. *IEEE Transactions on Fuzzy Systems*, 27(10), 1891-1903.