DE GRUYTER OPEN

JAISCR, 2016, Vol. 6, No. 1, pp. 23 – 33
10.1515/jaiscr-2016-0003

CLUSTERING LARGE-SCALE DATA BASED ON MODIFIED AFFINITY PROPAGATION ALGORITHM

Ahmed M. Serdah, Wesam M. Ashour

Computer Engineering Department, Islamic University of Gaza
108 Gaza, Palestine

Abstract

Traditional clustering algorithms are no longer suitable for use in data mining applications that make use of large-scale data. There have been many large-scale data clustering algorithms proposed in recent years, but most of them do not achieve clustering with high quality. Despite that Affinity Propagation (AP) is effective and accurate in normal data clustering, but it is not effective for large-scale data. This paper proposes two methods for large-scale data clustering that depend on a modified version of AP algorithm. The proposed methods are set to ensure both low time complexity and good accuracy of the clustering method. Firstly, a data set is divided into several subsets using one of two methods random fragmentation or K-means. Secondly, subsets are clustered into K clusters using K-Affinity Propagation (KAP) algorithm to select local cluster exemplars in each subset. Thirdly, the inverse weighted clustering algorithm is performed on all local cluster exemplars to select well-suited global exemplars of the whole data set. Finally, all the data points are clustered by the similarity between all global exemplars and each data point. Results show that the proposed clustering method can significantly reduce the clustering time and produce better clustering result in a way that is more effective and accurate than AP, KAP, and HAP algorithms.

1 Introduction

Clustering is one of the most popular research trends in data mining field. It is called unsupervised learning because no data, or statistical assumptions are needed. Currently, many applications such as text retrieval, medical image processing, and network intrusion detection are based on the clustering analysis. The goal of the clustering is to divide the data into sub-groups, whereas the data points in each group are relatively similar to each other and differ from the data points in the other groups. Consequently, many clustering methods have been developed, each of which has a different induction principle. Some of the most popular methods are:

Hierarchical-Based Clustering Methods [1 - 3]: These methods construct the clusters by recursively partitioning the instances in either agglomerative or divisive hierarchical. The result of these clustering methods is a dendrogram, representing the objects groups of data and similarity levels at which groupings change. Clusters are obtained by cutting the dendrogram at the desired similarity level.

Partitioning-Based Clustering Methods [4, 5]: Partitioning methods relocate move instances from one cluster to another, starting from an initial partition. In these methods, the number of clusters must be specified by the user. To achieve the global optimum in partitioning-based clustering, an exhaustive enumeration process of all possible partitions is required. Because this is not feasible, certain greedy heuristics approaches are used in the form of iterative optimization. That is; an iterative relocation method relocates points between the k clusters.

Unauthenticated
Download Date | 10/7/18 3:03 PM

Graph-Based Clustering Methods [6, 7]: Graph-based clustering methods produce clusters via graphs. The edges of the graph connect node instances in one class. Menendez et al. [7] proposed the Co-Evolutionary Multi-Objective Genetic Graph-based Clustering (CEMOG) algorithm, which is based on the Multi-Objective Genetic Graph-based Clustering (MOGGC).

Grid-Based Clustering Methods [8, 9]: These methods divide space into a finite number of cells that form a grid structure in which all operations performed for clustering. Fastest processing time is the main advantage of these methods.

Density-Based Clustering Methods [10, 11]: Density-based clustering methods assume that the points belong to each cluster drawn from a specific probability distribution. The overall distribution of data is assumed to be a mix of various distributions. The goal of density-based methods is to detect the clusters and their distribution parameters. These methods are designed to discover clusters of arbitrary shape.

Model-Based Clustering Methods [12, 13]: These methods try to optimize the fitting between the input data and some mathematical models. Unlike other clustering methods, which detects groups of objects, model-based clustering methods detect characteristic descriptions for each group, whereas each group designates a class or a concept. The most frequently used induction methods are neural networks and decision trees.

Affinity Propagation (AP) clustering algorithm is one of best and most recent partitioning clustering algorithms. Frey and Dueck [14] were the first to propose Affinity Propagation clustering algorithm (AP). The algorithm selects exemplar of each data point by passing two types of messages between data points iteratively. These messages are called responsibility and availability. The advantages of AP over other clustering algorithms are: performance, speed, initialization independence, and the similarity matrix between pairs of data points do not need to be symmetrical [15].

However, AP clustering has its drawbacks; it cannot generate a specific number of clusters out of K clusters, and the number of ultimate clusters is affected by a user- defined parameter ‘preference’. To obtain the desired number of clusters using AP, optimal value of ‘preference’ must be set. The bisection method suggests using AP to find a suitable preference for specified cluster number [20]. The process of finding the parameters is very time consuming because each change of any parameter will require re-running the algorithm. KAP [20] was developed to address AP drawback. KAP is able to generate K clusters according to what a user specifies by adding one constraint in the message passing process to restrict the number of clusters to K.

Another drawback in AP is that it consumes a significant amount of time and memory while clustering large-scale data, because it build three similarity matrices with size $n*n$ for n point data set. Although there are many algorithms proposed to improvement AP preference and initialization parameter problems [15 - 18], HAP [21] is the only one that tries to improve the efficiency of the AP algorithm in large-scale data clustering.

Inspired by the above ideas, this paper proposes a new clustering algorithm called IWC-KAP for large-scale data sets. IWC-KAP can directly generate K clusters, as specified by the user. It retains the advantages of KAP and inverse weighted clustering algorithm IWC [22]. Experiments on IWC-KAP show that it can generate K clusters directly without any parameter tuning, and can cluster large-scale data more efficiently than other related algorithms. Moreover, given a specified cluster number, IWC-KAP is much more efficient than HAP. IWC-KAP suggests two mechanisms for splitting data into subsets and then applies the KAP algorithm on each subset to find the local exemplars. IWC algorithm is applied on all local exemplars to find a specific number of global exemplars; then all the data points are re- clustered into new clusters by the global exemplars.

The rest of the paper is organized as follows: section II presents the related work. Section III, describes in details the proposed clustering algorithm IWC-KAP. Experiments on artificial and real data sets are conducted, and results are presented and analyzed in section IV. Finally, section V concludes the paper.

Unauthenticated
Download Date | 10/7/18 3:03 PM

2 Related Work

Many papers proposed clustering algorithms that depend on AP to increase the performance of AP using different ways. This part reviews some of these papers.

2.1 Affinity Propagation Clustering Algorithms

Affinity Propagation clustering algorithm [14, 18, 19] is based on passing messages between data points. Each data point receives availability *a(i,j)* message from the exemplars (members of the input set that are representative of clusters) and sends a responsibility *r(i,j)* to the example. The AP messages take into account different kind of competition. The availability message that is sent from candidate exemplar j to point i, reflects the accumulated evidence of how close point i to point j while taking into account that point j may be an exemplar for other points. The responsibility message that is sent from data point i to candidate exemplar point j, reflects the accumulated evidence of how well-suited data point j is to serve as the exemplar for data point i, and taking into consideration other possible exemplars for point i. The input of the AP is a matrix of similarities between pairs of data points *S= (s(i,j))*, and the output is cluster exemplars of all data points and relationships between each point and its cluster's exemplar. The similarity function *s(i,j)* in the similarity matrix indicates how well the data point j is suitable to be the exemplar of a data point with index i. The diagonal element of the similarity matrix *S(k,k)* indicates the 'preference' of data point with index **k**, so exemplars are that data points with larger values of *S(k,k)*.

The steps of the AP algorithm are as follows

Algorithm 1 AP algorithm steps

1: Initialization the availabilities matrix to zero
$a(i,k) = 0$
k is the number of exemplars
$i \in \{1,2,\ldots..,n\}$
n is the number of data points

2: Update the responsibilities by the following equation.
$r(i,k) = s(i,k) + \max_{k' \neq k}\{a(i,k') + s(i,k')\}$
where s is the similarity matrix between data points.

3: Update the availabilities matrix $a(i,k)$ $=\min\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\}$
Update self-availability by
$a(k,k) = \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}$

4: Find $sum = a(i,k) + r(i,k)$ for each point i and the exemplar k that maximize the sum.

5: For fixed number of iterations: If exemplars do not change go to step (6) else go to Step (1)

6: Assign the data points to its exemplars on the based on the maximum similarity to find clusters

2.2 Multi-exemplar Affinity Propagation (MEAP)

MEAP [23] proposed an extension of the single-exemplar model to a multi-exemplar one. MEAP can identify exemplars and a superexemplar for each cluster automatically. Each data point assigned to the most suitable exemplar and each exemplar assigned to the most suitable superexemplar. The superexemplar is defined as an exemplar that best represents the exemplars belonging to the corresponding cluster. The objective of the MEAP is to maximize the sum of all similarities between data points and the corresponding exemplars, plus the sum of all linkages between exemplars and the corresponding superexemplars. However, if the cluster number is prior knowledge, MEAP would not be able to make a use of such knowledge directly

Unauthenticated
Download Date | 10/7/18 3:03 PM

in its learning process. Instead, it has to rely on re-running the process as many times as it takes by tuning parameters until it generates the desired number of clusters. It also consumes a large amount of time and memory while processing large-scale data.

2.3 Adaptive Affinity Propagation

Adaptive Affinity Propagation (AAP) [24] was proposed as a model to overcome the drawback of AP that is related to knowing the value of the parameter preference, and tries to produce an optimal clustering solution. AAP firstly finds out a range of preference, then searches the space of preference to find a good value, which can optimize the clustering result.

2.4 Generating Specified K Clusters by Efficient Affinity Propagation

KAP [20] was modified to generate a given number of an optimal set of exemplars through Affinity Propagation. KAP can generate K clusters as the user specifies by adding one constraint in the process of message passing to confine the number of clusters to be K while keeping all AP advantages in clustering. Another advantage of KAP over AP is the confidence in one data item to be an exemplar is automatically self-adapted by KAP while the confidence in AP is a parameter specified by a user. Moreover, the computational cost overhead compared to AP is negligible. However, the limitations of clustering large-scale data are still existing as in AP. It still consumes time and memory while processing large-scale data.

2.5 An Improved Affinity Propagation Clustering Algorithm for Large-scale Data Sets

Hierarchical Affinity Propagation (HAP) was the first algorithm to use AP algorithm on large-scale data [21]. HAP proposed an improved hierarchical AP clustering algorithm. The algorithm achieves efficient, accurate and no predefined parameter large-scale data clustering by applying hierarchical selection and partitioned clustering. In a hierarchical selection, AP algorithm is executed for each subgroup according to: firstly finding well suited local exemplars for clusters in each subgroup. Secondly, AP is executed on all the local exemplars to find the global exemplars for all the data. In partitioned clustering, all of the data points are partitioned once again into new clusters by the global exemplars. One of the drawbacks of HAP is that when the number of clusters K is available, HAP, just like AP, cannot generate specified number of clusters directly. The second drawback of HAP is the time and memory consumption that comes as a result of using AP in finding the global exemplars.

3 Proposed Clustering Algorithm

3.1 The basic idea of the proposed algorithm

The proposed clustering algorithm depends on KAP Clustering algorithm to achieve efficiently, and accurate clustering result for large-scale data where the number of clusters is known. The basic idea of the algorithm is that, data set is divided into several subsets, each of which can be efficiently clustered by the KAP algorithm. The resulting subsets exemplars are clustered through the proposed IWC algorithm [22] to get a specific number of global exemplars, and then each data point is clustered to its exemplars. This process is divided into four steps:

Step 1: data partition

In this step, the entire data points of the data set are split into several small subsets that can be efficiently clustered using KAP. Two methods have been used in this step to divide the data. In the first method, data was divided randomly into n subset; in the second method, data was divided using the K-means algorithm. Through this step, the KAP clustering algorithm was directly applied on a large-scale data set.

Step 2: using KAP algorithm

In this step, the KAP algorithm is executed on each subset to select well-suited specific number of exemplars for each subset. The selection depends on the known clusters number of the data set and leads to obtaining the local optimal exemplars.

Step 3: find global exemplars and grounping data

In this step, IWC algorithm is used to find cluster centers from the all-local optimal exemplars. The selected cluster centers of the entire data set are called global exemplars.

Unauthenticated
Download Date | 10/7/18 3:03 PM

Step 4:

In this step, each data point is grouped into its cluster by finding its global exemplar using similarities between each data point and all global exemplars as in K-means clustering. Each data point will fit into its cluster as indicated by its maximal similarity.

3.2 Proposed method 1 Procedure

The procedures of the proposed method 1 in the algorithm are described in Algorithm 2.

Algorithm 2 The procedures used in the proposed method 1

1: The data set D divided into k partitions randomly, denoted as $D1, D2, \ldots, Dk$
where k is the number of partitions
$D1 \cap D2 \cap \ldots \cap Dk = \emptyset$
$D1 \cup D2 \cup \ldots \cup Dk = D$

2: For each partition Di, the KAP algorithm is performed to select the n number of local exemplar set of this partition, denoted as Ei. Where n is the number of clusters

3: Exemplars of all the partitions create a new data set, denoted as: $E = E1 \cup E2 \cup \ldots \cup Ek$.
IWC will be used on the data set E to select the global exemplars of the entire data set, denoted as $Eg1, Eg2, \ldots, Egn$. Each exemplar Egi $(1 \leq i \leq n)$ will be regarded as a centroid of cluster Ci, which is $Ci = \{ Egi \}$.

4: For each point di in data set D the similarities between di and each exemplar Egi, denoted as sim(di; Egi), are compared to find the exemplar point m with the maximal similarity.
$m = \max_j sim(di; cj)$
$then\ Cm = Cm \cup \{di\}$

5: Return the clustering result
$D = C1 \cup C2 \cup \ ::: \cup Cn$

3.3 Proposed method 2 Procedure

The procedures used in the proposed method 2 in the algorithm are described in Algorithm 3.

Algorithm 3 The procedures used in the proposed method 1

1: The data set D divided into k partitions using K-means algorithm to cluster data to initial k clusters, k clusters denoted as $D1, D2, \ldots, Dk$
where k is the number of partitions
$D1 \cap D2 \cap \ldots \cap Dk = \emptyset$
$D1 \cup D2 \cup \ldots \cup Dk = D$

2: For each partition Di, the KAP algorithm is performed to select the n number of local exemplar set of this partition, denoted as Ei. Where n is the number of clusters.

3: Exemplars of all the partitions create a new data set, denoted as: $E = E1 \cup E2 \cup \ldots \cup Ek$.
IWC will be used on the data set E to select the global exemplars of the entire data set, denoted as $Eg1, Eg2, \ldots, Egn$. Each exemplar Egi $(1 \leq i \leq n)$ will be regarded as a centroid of cluster Ci, which is $Ci = \{ Egi \}$.

4: For each point di in data set D the similarities between di and each exemplar Egi, denoted as sim(di; Egi), are compared to find the exemplar point m with the maximal similarity.
$m = \max_j sim(di; cj)$
$then\ Cm = Cm \cup \{di\}$

5: Return the clustering result
$D = C1 \cup C2 \cup \ ::: \cup Cn$

3.4 Key issues in the proposed algorithm

The Choosing partition size should address for the implementation of the algorithm.

The Choosing partition size: due to the limitations in time and memory consumption, the partition size should be decided by both the executing efficiency and the clustering result of the algorithm.

To get better results, the partitions of the data set should be a representative subset of the real data

Unauthenticated
Download Date | 10/7/18 3:03 PM

set. Usually, the bigger the partition size is, the better the representation of data will be, as well as the result of KAP. However, due to the memory and time consumption in KAP, small partition size is preferred. Therefore, for proposed method 1 neither big nor small partition size would lead to better the results. The choice of the partition size should be based on; application's specific demands for the clustering result, the efficiency of the clustering algorithm, and the characteristics of the data set that will be clustered. Furthermore, well-divided partitions ensure that each partition is a good representation of the entire data set. For proposed method 2, the K-means algorithm can produce well-divided partitions that can be a good representation of the entire data set.

4 Experimental Results And Analysis

An experiment was conducted on three clustering algorithms using a computer with 8G memory and 2.5GHz frequency. The algorithms were; the proposed algorithm (i.e. IWC-KAP), traditional AP, and KAP algorithm. The experiment is set to illustrate whether the proposed algorithm is more suitable for the large-scale data set clustering problem than the other algorithms.

4.1 Data sets and generating methods

The data sets and their characteristics of each data set for the experiments are as shown in Table 1.

Artificial data set

Three artificial two-dimensional data set S-Data1, S-Data2, and S-Data3, were generated using the random function in Matlab. As shown in table 1, S-Data1 contains 3500 samples, S-Data2 contains 1800 samples and S-Data1 contains 1400 samples. The data points of the artificial data sets are described in two-dimensional attributes to simplify the computation without loss of generality.

Real data set

As described in Table 1, five real data set were used:

Iris: 150 samples with 4 dimensions and 3 clusters.

Yeast: 1448 samples with 8 dimensions and 10 clusters.

Wine: 178 samples with 13 dimensions and 3 clusters.

Ionosphere: 151 samples with 34 dimensions and 2 clusters.

Heart: 302 samples with 13 dimensions and 5 clusters.

These data sets were used in most clustering algorithm experiments, and can be obtained from the UCI machine learning knowledge base website [25].

This work used the previous real data set to compare the result with [11] who used the same data set.

Table 1. Characteristics of data sets

Data sets	Data size	Number of classes	Attribute dimension
IRIS	150	3	4
Wine	178	3	13
Yeast	1484	10	8
Ionosphere	151	2	34
Heart	302	5	13
S-Data1	3500	7	2
S-Data2	1800	6	2
S-Data3	1400	7	2

Euclidean distance method was used to find the similarity between data points *pi* and *pj* in the data set.

Distance is described as in the following expression.

$$s(pi, pj) = -||pi - pj||^2 \quad (1)$$

The partition size for the proposed method 1 for the proposed algorithm was selected as 0.25 times of the data set size that will cluster. However, in the proposed method 2 depend on the size of initial clusters result from K-means algorithm. For the partition step in the proposed method 1, the code was run 50 times. The results were recorded, and the average value was calculated.

Unauthenticated
Download Date | 10/7/18 3:03 PM

4.2 Evaluation Methods

The Normalized Mutual Information (NMI) index [26] was used to evaluate the results of the four algorithms; AP, KAP, HAP and the proposed algorithm. NMI was used to measure the similarity between the result of the clustering algorithm and the standard division of the data set. The calculation of the NMI index can be described as follows

For any partition of the data set, denoted as P^a, the information entropy (which is the expected value (average) of the information) of this partition is:

$$H(P^a) = -\sum_{i=1}^{ka} \frac{n_i^a}{n} \log(\frac{n_i^a}{n}) \tag{2}$$

Where n is the data size, ka is the number of clusters in the partition, $n^a{}_i$ is the number of points in the i-th cluster in the partition.

The mutual information (which is a measure of the variables' mutual dependence) for two partitions of the same data set Pa and Pb, is calculated by the following formula:

$$I(p^a,p^b) = \sum_{i=1}^{ka}\sum_{j=1}^{kb} \frac{n_{ij}^{ab}}{n} \log(\frac{\frac{n_{ij}^{ab}}{n}}{\frac{n_i^a}{n} \times \frac{n_j^b}{n}}) \tag{3}$$

Where n_{ij}^{ab} is the number of the points both in the i-th cluster of P^a and in the j-th cluster of P^b.

The lack of information between two vectors is defined as:

$$I(p^a|p^b) = -\sum_{i=1}^{ka}\sum_{j=1}^{kb} \frac{n_{ij}^{ab}}{n} \log(\frac{\frac{n_{ij}^{ab}}{n}}{\frac{n_j^b}{n}}) \tag{4}$$

From the communication theory point of view, the above-defined quantities can be interpreted as follows. Suppose we need to transmit all the cluster labels in P^a on a communication channel, then H(P^a) can be interpreted as the average amount of information, for example, in bits, needed to encode the cluster label of each data point according to P^a. Now suppose that P^b is made available to the receiver, and then H(P^a | P^b) denotes the average number of bits needed to transmit each label in p^a if P^b is already known. We are interested to see how much H(P^a | P^b) is smaller than H(P^a), that is, how much the knowledge of P^b helps us to reduce the number of bits needed to encode P^a. This can be quantified in terms of the mutual information H(P^a)-H(P^a | P^b) = I(P^a, P^b). The knowledge of P^b thus helps us to reduce the number of bits needed to encode each cluster label in P^a by an amount of I(P^a, P^b) bits. In the reverse direction, we also have I(P^a, P^b) = H(P^b)-H(P^b P^a). Clearly, the higher the MI, the more useful the information in P^b helps us to predict the cluster labels in P^a and vice-versa.

The similarity of two partitions Pa and Pb for the same data set is calculated using the NMI index as follows.

$$NMI(p^a,p^b) = \frac{2 \times I(pa,pb)}{H(Pa) + H(Pa)} \tag{5}$$

The NMI measures the information that P^a and P^b share: it tells us how much each one of these clusterings reduces our uncertainty about the other. The value of the NMI index for two partitions of any data set is [0..1]. The bigger the NMI index is, the more similarity the two partition are.

4.3 Results and Analysis

Table 2, Figure 1 and Figure 2, show the clustering time of the real and the artificial data sets, and compare the two methods of proposed algorithm with the AP algorithm and KAP algorithm.

Table 3, Figure 3 and Figure 4, show the NMI index of the real and the artificial data sets, and compare the two methods of proposed algorithm with the AP algorithm and KAP algorithm.

Table 4, Figure 5 and Figure 6, show the clustering time of the real and the artificial data sets, and compare the two methods of proposed algorithm with the HAP algorithm.

Table 5, Figure 7 and Figure 8, show the NMI index of the real and the artificial data sets, and compare the two methods of proposed algorithm with the HAP algorithm.

Through Table 2, Table 3, Figure 1 till Figure 4, it can be seen that the proposed algorithms achieves the lowest time consumption compared to AP and KAP algorithms for almost all data sets. Furthermore, the proposed algorithms gave better results than AP with the NMI index and gave an almost

Unauthenticated
Download Date | 10/7/18 3:03 PM

result as close as possible to KAP algorithm. However, time consumption increases rapidly with the growth of the data size in the AP and KAP algorithm, which makes those algorithms not suitable for solving the clustering problem of large-scale data. In the proposed algorithms, the clustering results are significantly better than AP and KAP on all the data sets except the Ionosphere and Yeast data. Also, the time consumption of the proposed algorithms is much lower than AP and KAP on all data. The great time consumption of AP and KAP is due to the computation of matrixes of similarities between pairs of data points in the entire data set while the input to the proposed algorithms is the matrix of similarities between pairs of data points only in each partition, and the input to the selection of global exemplars is the matrix of its data record only. The proposed algorithms have advantages over others in term of memory consumption comparing with AP and KAP. It is also more efficient and accurate more than the others.

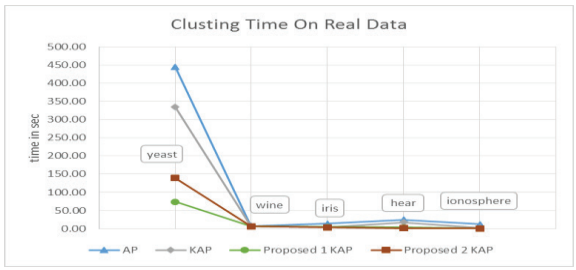

Figure 1. Clustering time our proposed v. AP and KAP in real data set

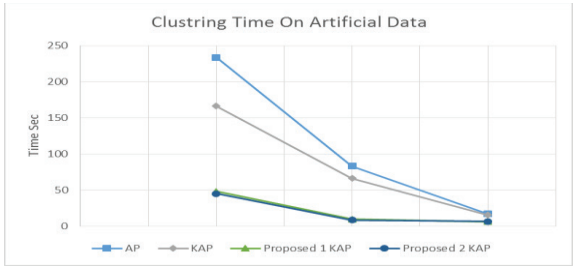

Figure 2. Clustering time our proposed v. AP and KAP in Artificial data set

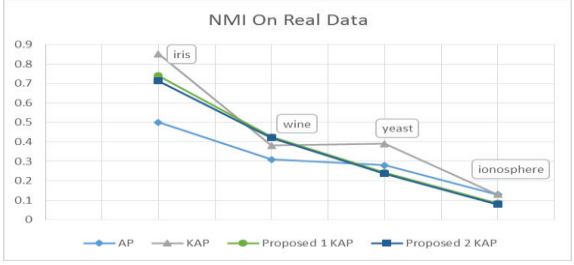

Figure 3. NMI index our proposed v. AP and KAP in a real data set

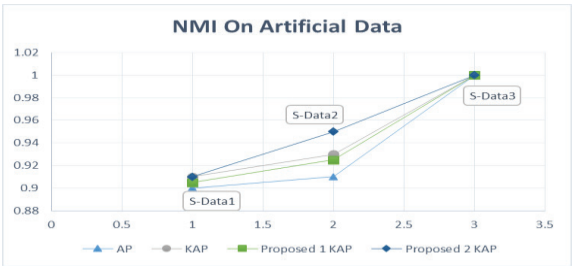

Figure 4. NMI index our proposed v. AP and KAP in Artificial data set

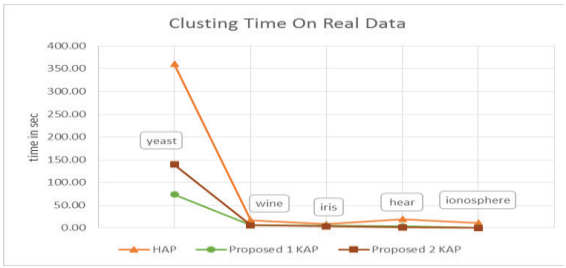

Figure 5. Clustering time our proposed v. HAP in a real data set

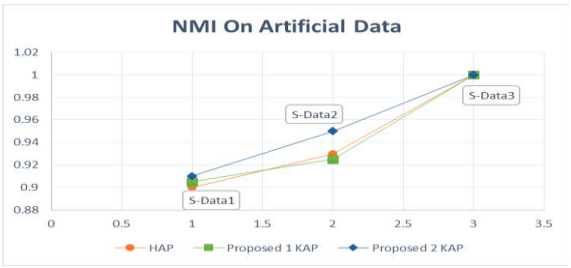

Figure 6. Clustering time our proposed v. HAP in Artificial data set

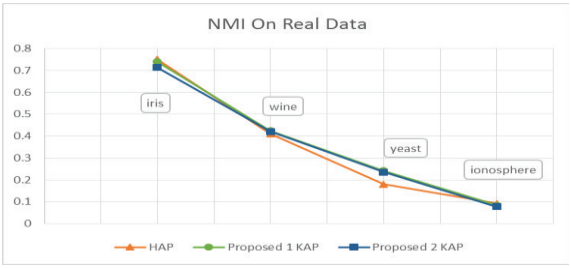

Figure 7. NMI index our proposed v. HAP real in a data sett

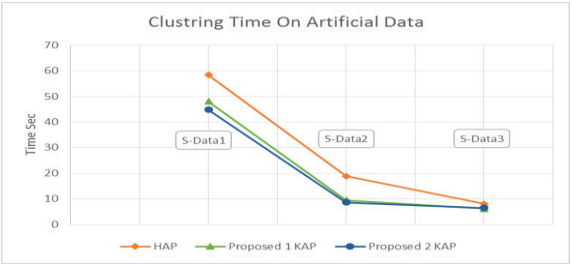

Figure 8. NMI index our proposed v. HAP in Artificial data set

This work was also compared with the HAP algorithm, which is the first algorithm that addressed the AP problem in the large-scale data set. As it

Unauthenticated
Download Date | 10/7/18 3:03 PM

Table 2. Clustering time our proposed V. AP and KAP

	AP	KAP	Proposed 1 KAP best	Proposed 1 KAP worst	Proposed 1 KAP	Proposed 2 KAP
yeast	444.71	334.81	57.16	90.23	73.70	139.09
wine	7.33	6.58	5.05	7.09	6.07	6.20
iris	13.36	4.06	4.66	5.37	5.01	3.68
heart	24.81	17.66	2.36	4.63	3.49	1.11
ionosphere	12.34	2.78	0.39	0.72	0.56	0.29
S-Data1	233.65	165.96	41.49	54.47	47.98	44.75
S-Data2	82.83	65.97	8.13	10.68	9.41	8.54
S-Data3	17.01	15.22	5.53	6.69	6.11	6.40

Table 3. MNI index our proposed V. AP and KAP

	AP	KAP	Proposed 1 KAP best	Proposed 1 KAP worst	Proposed 1 KAP	Proposed 2 KAP
iris	0.5	0.8512	0.6561	0.8245	0.7403	0.7131
wine	0.31	0.3813	0.4561	0.3945	0.4253	0.4214
yeast	0.28	0.3896	0.2834	0.2018	0.2426	0.2361
ionosphere	0.129	0.1296	0.1347	0.0349	0.0848	0.0784
S-Data1	0.9	0.91	0.91	0.9	0.905	0.91
S-Data2	0.91	0.93	0.95	0.9	0.925	0.95
S-Data3	1	1	1	1	1	1

Table 4. Clustering time our proposed V. HAP

	HAP	Proposed 1 KAP best	Proposed 1 KAP worst	Proposed 1 KAP	Proposed 2 KAP
yeast	360.48	57.16	90.23	73.70	139.09
wine	17.35	5.05	7.09	6.07	6.20
iris	8.27	4.66	5.37	5.01	3.68
heart	19.27	2.36	4.63	3.49	1.11
ionosphere	10.99	0.39	0.72	0.56	0.29
S-Data1	58.41	41.49	54.47	47.98	44.75
S-Data2	18.83	8.13	10.68	9.41	8.54
S-Data3	8.04	5.53	6.69	6.11	6.40

Unauthenticated
Download Date | 10/7/18 3:03 PM

Table 5. MNI index our proposed V. HAP

	HAP	Proposed 1 KAP best	Proposed 1 KAP worst	Proposed 1 KAP	Proposed 2 KAP
iris	0.75	0.6561	0.8245	0.7403	0.7131
wine	0.41	0.4561	0.3945	0.4253	0.4214
yeast	0.18	0.2834	0.2018	0.2426	0.2361
ionosphere	0.09296	0.1347	0.0349	0.0848	0.0784
S-Data1	0.9	0.91	0.9	0.905	0.91
S-Data2	0.93	0.95	0.9	0.925	0.95
S-Data3	1	1	1	1	1

can be seen in Table 4, Figure 5 and Figure 6 the time consumption of the proposed algorithm is not only lower than AP and KAP algorithms but also lower than HAP algorithm. However, as seen in Table5, Figure 7 and Figure 8, the clustering results by the NMI index is almost close to HAP algorithm. However, the time consumption increases rapidly with the growth of the data size in the HAP, which makes the proposed algorithm more efficient as it consumes time less than HAP when the size of data grow. The increase in time consumption of HAP results is due to the computation of the matrices of similarities between pairs of data points in the entire data set when using the AP algorithms again to get the global exemplars from the local exemplars. However, the proposed algorithms use modified K-means algorithms to find global exemplars that decrease the time because the K-means is faster than AP algorithms. The proposed algorithms have advantages over HAP when it comes to less memory consumption. It is also more efficient and accurate.

5 Conclusion

Two methods depend on KAP and IWC algorithms were proposed in this paper. The proposed methods achieve eficient, accurate, and time-saving clustering for large-scale data sets. Data points are clustered by splitting the data into small groups, then applying KAP algorithm on each subset of the data. Then, IWC is applied to find the global exemplars for original finally well-suite clusters. These clusters are obtained by setting the points to its similar exemplars due to similarity function.

The algorithms were tested using real and simulated data sets. The results show that the proposed algorithms are more effective and efficient in term of clustering time consumption and memory space consumption than AP, KAP, and HAP. This is due to the proposed novel techniques. In the future, other modified versions of AP will be used in clustering subsets instead of KAP. Furthermore, other partitioning algorithms are going to be used instead of IWC to find out whether better results can be achieved.

References

[1] L. Kaufan, P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York, 1990.

[2] R. Lior, and O. Maimon, Clustering Methods, Data mining and knowledge discovery handbook. Springer US, 2005, pp. 321-352

[3] S. Patel, S. Sihmar and A. Jatain, A Study of Hierarchical Clustering Algorithms, Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on, 2005 pp. 537-541

[4] J.W. Han and M. Kambr, Data Mining Concepts and Techniques, Higher Education Press, Beijing, 2001.

[5] Y. Kang and Y. B. PARK, The Performance Evaluation of K-means by Two MapReduce Frameworks, Hadoop vs. Twister, Information Networking (ICOIN), 2015 International Conference on, 2015, pp. 405-406

[6] A. Y. Ng, M. I. Jordan, and Y. Weiss, On spectral clustering: Analysis and an algorithm, in Advances in Neural Information Processing Systems, 2001, pp. 849-856

[7] H. D. Menendez, D. F. Barrero and D. Camacho, A Co-Evolutionary Multi-Objective Approach for a K-Adaptive Graph-based Clustering Algorithm, IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 2724-2731

Unauthenticated
Download Date | 10/7/18 3:03 PM

[8] Han, J., and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001, pp. 450-479

[9] C. Tsai; Y. Hu, Enhancement of efficiency by thrifty search of interlocking neighbor grids approach for grid-based data clustering, Machine Learning and Cybernetics (ICMLC), 2013 International Conference on, 2013, pp. 1279-1284

[10] M. Ester, H. P. Kriegel, J. S, X. W. Xu, A density based algorithm for discovering clusters in large spatial databases with noise, in Proc. 2nd International Conference on, 1993, pp. 2-11

[11] S.T.Mai, He. Xiao, N. Hubig, C. Plant and C. Bohm, Active Density-Based Clustering, Data Mining (ICDM), 2013 IEEE 13th International Conference on, 2013, pp. 508–517

[12] Zahn, C. T., Graph-theoretical methods for detecting and describing gestalt clusters. IEEE trans. Comput. C-20 (Apr.), 1971, pp. 68-86

[13] F. Chamroukhi, Robust EM algorithm for model-based curve clustering, Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1-8

[14] B. J. Frey, D. Dueck, Clustering by Passing Messages Between Data Points, in Science, vol. 315, 2007, pp. 972-976

[15] Wang Kai-jun, Zhang Jun-ying, Li Dan, et al, Adaptive Affinity Propagation Clustering, J. Acta Automatica Sinica, vol. 33(12), 2007, pp. 1242-1246

[16] Wang Kai-jun, Li Jian, Zhang Jun-ying, et al, Semi-supervised Affinity Propagation Clustering, J. Computer Engineering, vol. 33(23), 2007, pp. 197-201

[17] Yancheng He, Qingcai Chen, Xiaolong, et al, An Adaptive Affinity Propagation Docu-ment Clustering, Proceedings of the 7th International Conference on Informatics and Sys-tems, 2010, pp. 1-7

[18] Yangqing Jiay, Jingdong Wangz, Changshui Zhangy, Xian-Sheng Hua, Finding Image Exemplars Using Fast Sparse Affinity Propagation, Proceedings of the 16th ACM International conference on Multimedia, 2006, pp. 113 -118

[19] Yasuhiro Fujiwara, Go Irie and Tomoe Kitahara, Fast Algorithm for Affinity Propagation, International Joint Conference on Artificial Intelligence (IJCAI), 2011, pp. 2238-2243

[20] Xiangliang Zhang, Wei Wang, Kjetil Nrvag and Michele Sebag, K-AP: Generating Specified K Clusters by Efficient Affinity Propagation, Data Mining (ICDM), 2010 IEEE 10th International Conference on, 2010, pp. 1187 -1192

[21] Xiaonan Liu, Meijuan Yin, Junyong Luo and Wuping Chen, An Improved Affinity Propagation Clustering Algorithm for Large-scale Data Sets, 2013 Ninth International Conference on Natural Computation (ICNC), IEEE, 2013, pp. 894 - 899

[22] W. Barbakh and C. Fyfe. Inverse weighted clustering algorithm, Computing and InformationSystems, 11(2)10-18, May 2007. ISSN 1352-9404.

[23] C.-D. Wang, J.-H. Lai, C. Suen, and J.-Y. Zhu, Multi-exemplar affinity propagation, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 35, 2013 pp. 2223–2237

[24] K.J. Wang, J.Y. Zhang, D. Li, X.N. Zhang, and T. Guo, Adaptive Affinity Propagation Clustering, Acta Automatica Sinica, vol. 33, no. 12, 2007, pp. 1242-1246

[25] C. L. Blake, C. J. Merz, "UCI repository of machine learning databases," 2012, http://archive.ics.uci.edu/ml/.

[26] L. N. Ana, Fred, K. J. Anil, Robust Data Clustering, Computer Vision and Pattern Recognition, 2003, Proceedings, 2003 IEEE Computer Society Conference on, 2003, pp. 128 – 133

Ahmed Serdah is a Master Student at Islamic University of the Gaza. He has graduated in 2005 with B.Sc. in Computer Engineering from Islamic University of Gaza. He is interested in research on neural networks, artificial intelligence, data mining and intrusion detection.

Wesam Ashour is an Assistant Professor at Islamic University of Gaza. He is an active researcher at the Applied Computational Intelligence Research Unit in the University of the West of Scotland. He obtained his Master and Doctorate degrees from UK. His research interests include data mining, artificial intelligence, reinforcement learning and neural networks.

Unauthenticated
Download Date | 10/7/18 3:03 PM