Analysis of Some Algorithms for Clustering Data Objects

Mohamed Nour Elsayed and Monzer Mohamed Qasem

Abstract—The main objective of clustering is to partition a set of objects into groups or clusters. The objects within a cluster are more similar to one another than those of the others clusters. This work analyzes, discusses and compares three clustering algorithms. The algorithms are based on partitioning, hierarchical, and swarm intelligence approaches. The three algorithms are k-means clustering, hierarchical agglomerative clustering, and ant clustering respectively. The algorithms are tested using three different datasets. Some measurable criteria are used for evaluating the performance of such algorithms. The criteria are: intra-cluster distance, inter-cluster distance, and clustering time. The experimental results showed that the k-means algorithm is faster and easily understandable than the other two algorithms. The k-means algorithm is not capable of determining the appropriate number of clusters and depends upon the user to identify this in advance. The ease of handling of any forms of similarity or distance is one of the advantages of the hierarchical clustering algorithm. The disadvantage involves the embedded flexibility regarding the granularity level.

The ant-clustering algorithm can detect the more similar data for larger values of swarm coefficients. The performance of the ant clustering algorithm outperforms the other two algorithms. This occurs only for the better choice of the swarm parameters; otherwise the agglomerative hierarchical clustering is the best.

Index Terms—Datasets, hierarchical clustering, partitioning clustering, swarm intelligence.

I. INTRODUCTION

Clustering analysis is a primary tool for data mining and/or analysis. Clustering in [1] is considered a necessary unsupervised learning problem (H. Wang et al., 2012). A cluster is briefly defined as [2] a collection of data objects which are similar between them and are dissimilar to the objects belonging to other clusters.

The clustering methods can be classified into five categories mainly: [3] partitioning, hierarchical, density-based, grid-based, and model-based methods. Several attempts were presented discussing the different clustering algorithms. Examples of such clustering works are briefly mentioned as follows: [3] mentioned that hierarchical algorithms find successive clusters using previously established clusters, whereas partitioning algorithms determine all clusters at a time. Hierarchical algorithms may be agglomerative or divisive. Agglomerative algorithms begin with each element as a separate cluster and merge them successively to form larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

In [4] mentioned that clustering with swarm-based algorithms is emerging as an alternative to more conventional clustering techniques. The authors briefly presented a survey on ant—based clustering algorithms as well as some applications on such clustering algorithms.

In [5] presented a survey of density-based clustering algorithms. Such algorithms can detect the clusters of different shapes and sizes from large amount of data which contains outliers. The authors analyzed two clustering methods in terms of the parameters essential for creating meaningful clusters.

In [6] mentioned that clustering techniques are used to put similar data items in the same group. k-means clustering is a common approach which is based on initial centroids selected randomly. The authors calculated initial centroids instead of random selection. This improved the elapsed time consumed in the clustering process.

In [7] discussed the concepts of inductive and non–inductive clustering. When the clustering results naturally induce functions for classification on the whole space of interest such method is called inductive clustering. A method is called non-inductive if it does not induce such a function. Examples of inductive clustering are crisp and fuzzy c-means while the agglomerative hierarchical clustering is an example of non-inductive clustering.

In [8] mentioned that the k-means is one of the unsupervised learning algorithms that solve the well known clustering problem. The algorithm can classify a given data set through a certain number of clusters fixed a priori (k-clusters for example). The k-means algorithm can cluster data faster than k-medoids when tested with large datasets and the results were satisfactory.

The organization of this work is as follows: Section II, Section III, and Section IV present the clustering algorithms based on partitioning, hierarchical and swarm approaches respectively. The implementation is presented in Section V. Finally, the discussion of results and conclusion are presented in Section VI and Section VII respectively.

II. CLUSTERING USING THE K-MEANS ALGORITHM

The k-means clustering algorithm can partition a dataset into a predetermined number of clusters. The elements of the dataset are assigned to the nearest cluster according to the distance metric [9], [10]. Clusters are fully dependent on the selection of the initial clusters’ centroids. k-data elements are chosen as initial centers then distances of all
data elements are calculated. Data elements with less
distance to centroids are moved to the appropriate cluster.
The operation continues till no more changes in the
obtained clusters. The words data points or objects can be
used interchangeably. This algorithm was taken from [6],
obtained clusters. The words data points or objects can be
The updating formula can be written as:
\[ d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \] (5)
The average linkage involves the group average where the
dissimilarity between clusters can be written as:
\[ d(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x, y)}{|C_i||C_j|} \] (7)
where \(|C_i|\) is the number of elements in \(C_i\). The updating formula can be written as:
\[ d(C_i, C_j) = \frac{|C_p|d(C_p, C_j) + |C_q|d(C_q, C_j)}{|C_p| + |C_q|} \] (8)

III. CLUSTERING USING THE BOTTOM-UP HIERARCHICAL ALGORITHM

The hierarchical clustering approach can make
decomposition of the dataset or data objects. The hierarchical clustering method may be bottom-up
(agglomerative) or top-down (divisive). The bottom-up hierarchical clustering algorithm was taken from [7], [12], [13], and its steps can be briefly mentioned as shown below.

Algorithm: Agglomerative hierarchical clustering
Inputs: The set of objects \(X = \{x_1, x_2, \ldots, x_n\}\) \(*n* is the number of objects!*/
Output: A set of clusters \(C = \{C_1, C_2, \ldots, C_k\}\) \(*k* is the number of clusters */
Steps:
1) Assume that each object represents an initial cluster \(C_i\) \([x_i]\), \(i=1, 2, \ldots, n, *n* is the number of data objects *)/ \(k=n\) \(*k* is the number of clusters *)/
2) Find the cluster-pair with minimum distance \((C_p, C_q)\) where
\[ (C_p, C_q) = \arg \min_{i,j} d(C_i, C_j) \] (2)

Merge \(C_i = C_p \cup C_q\)
Reduce the number of clusters \(k = k-1\)
3) IF arrived at the number of clusters THEN stop ELSE update distance \(d(C_i, C_j)\) for all other clusters \(C_i\);
4) Go to step 2.
Moreover, the distance \(d(C_i, C_j)\) can take different forms.

1) The single linkage involves the nearest neighbor where the dissimilarity between clusters is defined as:
\[ d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \] (3)
The updating formula can be considered as:
\[ d(C_i, C_j) = \min \{d(C_p, C_j), d(C_q, C_j)\} \] (4)

2) The complete linkage involves the farthest neighbor where the dissimilarity between clusters is defined as:

The updating formula can be written as:
\[ d(C_i, C_j) = \max \{d(C_p, C_j), d(C_q, C_j)\} \] (6)

3) The average linkage involves the group average where the dissimilarity between clusters can be written as:

where \(|C_i|\) is the number of elements in \(C_i\). The updating formula can be written as:
\[ d(C_i, C_j) = \frac{|C_p|d(C_p, C_j) + |C_q|d(C_q, C_j)}{|C_p| + |C_q|} \] (8)
IV. ANALYSIS OF CLUSTERING BASED ON THE ANT BEHAVIOR

Clustering of data items or objects can be done using the behavior of ants. The data items are randomly scattered into a two-dimensional grid. The clustering algorithm based on ant behavior is briefly mentioned below and it was taken from [3], [4].

Ants randomly move around such grid picking and dropping the data items. Picking-up or dropping a data item is based on the ant’s immediate neighborhood. The probability to pick a data item increases if such item is surrounded by similar data in the neighborhood. The probability to drop a data item increases if ants are surrounded by dissimilar data or when there is no data in its neighborhood. The probability to drop a data item increases if such item is unloaded. When an ant encounters a data object and its state is unloaded, a swarm similarity and picking-up probability are computed for deciding whether or not to pick-up the object. The swarm similarity and dropping probability are computed. This occurred for two, four, six, eight, and ten clusters respectively. Each has different values of the swarm similarity and the user defined values.

Regardless the adopted clustering algorithms, the clustering problem can be easily formulated. Assume that \( D = \{x_1, x_2, \ldots, x_n\} \) where \( n \) is the number of data points/ objects and \( x_i \) is a data point of \( p \)-dimensional where \( x_i \in \mathbb{R}^p \). Clustering aims at decomposing the data set \( D \) into \( k \) cluster \( C_1, C_2, \ldots, C_k \) such that:

\[
C_i \neq \emptyset \quad \text{for } i = 1, 2, \ldots, k
\]

\[
C_i \cap C_j = \emptyset \quad \text{for } i, j = 1, 2, \ldots, k \text{ and } i \neq j
\]

\[
\bigcup_{i=1}^{k} C_i = D
\]

The three algorithms are run several times to evaluate the performance of each one. The experiments are operated on three different datasets. In each experiment, the intra-cluster distance, inter-cluster distance, and clustering time are computed. This occurred for two, four, six, eight, and ten clusters respectively. Each has different values of the swarm similarity and the user defined values.

The test-beds/datasets were chosen from the machine learning repository at UCI. The test-beds are: seeds dataset, blood transfusion dataset, and hepatitis dataset. The main attributes for such datasets are briefly written in Table I. For more details, the reader can refer to [14].

\[
f(o_i) = \frac{1}{S} \sum_{o_j \in \text{Neigh}(o_i)} \left[ \frac{1 - d(o_i, o_j)}{\alpha} \right]
\]

where \( S \) is the number of objects \( o_i \), \( \alpha \) is the swarm similarity coefficient, \( d(o_i, o_j) \) is the distance between two objects \( o_i \) and \( o_j \) in the space of attributes measured with Euclidean distance. For more details, the reader can refer to [3], [4].

V. IMPLEMENTATION WORK

TABLE I: THE CHOSEN DATASETS AS TEST-BEDS [14]

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Instances</th>
<th>Area</th>
<th>No. of Attributes</th>
<th>No. of Web Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeds Dataset</td>
<td>210</td>
<td>life</td>
<td>7</td>
<td>20736</td>
</tr>
<tr>
<td>Blood Transfusion Dataset</td>
<td>748</td>
<td>Business</td>
<td>5</td>
<td>60961</td>
</tr>
<tr>
<td>Hepatitis Dataset</td>
<td>155</td>
<td>life</td>
<td>19</td>
<td>53370</td>
</tr>
</tbody>
</table>

A. Implementation of the K-Means Algorithm

The Euclidean metric or distance is used to measure the distance between any two objects or between a cluster center and any of its objects \( x \) where any object is of dimensional \( p \). Such distance can be written as follows: [6], [9], [10]

\[
d = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2}
\]

where \( c_1, c_2, \ldots, c_k \) are the cluster centers and

\[
c_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \quad \text{for } k=1, 2, \ldots, k
\]

This means that any object should be assigned to the cluster corresponding to the nearest center. The calculation of distances among all cluster centers and the data objects are iteratively computed till no change in the clusters. The standard deviation (SD) is used to check the closeness of the
data objects in each cluster as follows:

\[ SD_k = \sqrt{\frac{\sum_{i=1}^{n} (x_{ij} - c_{ik})^2}{m}} \]  

(16)

where \( m \) is the number of objects in the cluster \( c_k \). The intra-cluster similarity or distance (\( \text{inter-CD} \)), and inter-cluster distance/similarity (\( \text{inter-CD} \)) are computed as follows:

\[ \text{Intra-CD} = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_{ij} - c_{jk} \|^2 \]  

(17)

The \( \text{intra-CD} \) is used to measure the compactness of the clusters. The \( \text{inter-CD} \) is the minimum distance between the cluster centroids which can be written as:

\[ \text{Inter-CD} = \min \{ \| m_k - m_{kk} \| \} \]  

\[ \forall k = 1, 2, \ldots, k-1, \text{and } kk = k+1, \ldots, k \]  

(18)

The \( \text{inter-CD} \) is used to measure the separation of the clusters. The cluster distance or cluster similarity can be used interchangeably.

B. Implementation of the Agglomerative Hierarchical Algorithm

In agglomerative hierarchical algorithm, the data objects are assumed to be in a list. The clustering process is done as follows:

1) Each object is considered as a cluster. i.e. there are \( n \) clusters \( C_1, C_2, \ldots, C_n \).
2) A similar matrix is constructed by computing the similarity value between all data objects. The similarity or the cost function between two clusters \( C_i, C_j \) is computed by the cosine angle as follows:

\[ \text{Cosine}(\theta) = \frac{C_i \cdot C_j}{\| C_i \| \| C_j \|} \]  

(19)

3) The most similar pair of clusters \( C_i \) and \( C_j \) are merged in one cluster.
4) The two clusters \( C_i \) and \( C_j \) are removed from the list and replaced by \( C_i \cup C_j \).
5) The similarity matrix is updated after merging \( C_i \) and \( C_j \).
6) If the number of required clustered is obtained stop otherwise continue merging the most two similar objects till the number of clusters are constructed [7], [8], [12].

C. Implementation of the Ant Algorithm

The ant clustering algorithm is run several times and the picking-up probability and the dropping probability are calculated. The values of such probabilities are affected by both the values of the user defined parameters \( (u_1, u_2) \) and the swarm similarity coefficient value \( (\alpha) \). The values \( (u_1, u_2) \) are considered as threshold items where \( \alpha \) is considered as the scale for dissimilarity. The experimental results change by changing such parameters. By the good choice of such parameters, promising clusters are obtained by this ant algorithm. Ten different values of \( \alpha \) are adopted for evaluating the performance of the ant clustering process.

After applying the above clustering algorithms on the chosen datasets, some results were reported as shown in Fig. 1- Fig. 18. The obtained results are: the intra-cluster distance (\( \text{Intra-CD} \)), inter-cluster distance (\( \text{Inter-CD} \)), standard deviation, picking-up and dropping probabilities versus the swarm similarity, and number of data points or objects for each dataset.

![Fig. 1. Cluster distance (CD).](image1)

![Fig. 2. Cluster distance (CD).](image2)

![Fig. 3. Cluster distance (CD).](image3)

![Fig. 4. Cluster distance (CD).](image4)
Fig. 5. Cluster distance (CD).

Fig. 6. Cluster distance (CD).

Fig. 7. Standard deviation.

Fig. 8. Standard deviation.

Fig. 9. Standard deviation.

Fig. 10. Picking-up and dropping probability.

Fig. 11. Picking-up and dropping probabilities.

Fig. 12. Average picking and dropping probabilities.

Fig. 13. Picking and dropping probabilities.

Fig. 14. Average picking and dropping probabilities.
The adopted clustering algorithms were tested using the three datasets mentioned above. Fig. 1 to Fig. 6 show the intra-cluster distance (Intra-CD) and inter-cluster distance (Inter-CD) for both the K-means and agglomerative hierarchical algorithms respectively. Both the Intra-CD and Inter-CD change by changing the number of clusters. The Intra-CD increases by increasing the number of clusters while the Inter-CD decreases. This is valid for the two algorithms for two, four, six, eight, and ten clusters respectively. For the same number of clusters, the Intra-CD and Inter-CD were better for the agglomerative hierarchical algorithm than the K-means one. Moreover, the standard deviation values (in Fig. 7 to Fig. 9) for the hierarchical algorithm are smaller than their corresponding values of the K-means algorithm for the same number of clusters. This occurred for the three chosen datasets. The time consumed for clustering the datasets was higher for the agglomerative hierarchical algorithm than the K-means one.

The performance of the ant clustering algorithm was affected by some important parameters, i.e. the probability of picking-up an ant ($P_p$) and the probability of dropping an ant ($P_d$) are affected by a set of parameters. Such parameters are: the swarm similarity coefficient, the two user defined parameters (the threshold constants) for computing both the picking-up and dropping probabilities, the distance between objects $o_i$ and $o_j$ in the dataset measured with Euclidean distance, the number of objects, the features and characterization of the chosen dataset. Ants can pick-up data objects that are either isolated or surrounded by dissimilar objects and then drop the picked ones in the similar types of objects. Fig. 10 to Fig. 12 show the relationship between picking-up and dropping probabilities versus the swarm similarity coefficient. For all the experiments, the probability of dropping an ant is higher than the corresponding values of the picking-up probability. This occurs for the different values of the swarm similarity coefficient. Both the picking-up and dropping probabilities for two of the chosen test-beds decrease by increasing the value of the swarm coefficient. The adopted ten values of the swarm similarity coefficient change from ten to hundred in all experiments. The idea of such choice is to detect the number of dissimilar and similar objects. Small values of swarm similarity coefficient cause larger number of dissimilar data points and vice versa. That choice aims at reducing the number of dissimilar data points to make more data points falling into clusters. The average $P_p$ and average $P_d$ are different for the same swarm similarity coefficient for the chosen datasets. Fig. 13 to Fig. 15 show the relationship between each $P_p$ and $P_d$ and the number of data points for each dataset. The dropping probability is always greater than the picking-up probability for the same data points. This is due to the chosen values of the threshold constants. The clustering time for the algorithms for the three datasets is reported as shown in Fig. 16 to Fig. 18.

The clustering time was smaller for the $k$-means algorithm for the same number of clusters, i.e. the time complexity is higher for the agglomerative hierarchical algorithm. The time complexity for the agglomerative hierarchical clustering algorithm is $O(n \log^3 n)$ where $n$ is the number of data objects in the dataset. The time complexity for the $k$-means algorithm is $O(nkl)$ where $n$, $k$, and $l$ are the number of data objects/points, the number of clusters, and the number of iterations consumed by the algorithm respectively. The clustering time for both the agglomerative hierarchical algorithm and the ant algorithm are non-linear while it seems to be linear for the $k$-means one.
VII. CONCLUSION

This work presented an analysis of three algorithms for clustering objects or data points. Two of such algorithms are considered conventional mainly the K-means and agglomerative hierarchical algorithms. The third algorithm is based on swarm intelligence. The performance of the agglomerative hierarchical algorithm is better than that of the K-means one. The Intra-cluster and Inter-cluster distances were better for the agglomerative hierarchical algorithm.

The ant clustering; based on the swarm intelligence; is promising compared with the other two algorithms. Several parameters have significant effects on the performance of the ant algorithm. This involves the swarm similarity coefficient, the probability to pick-up or drop ants, the distance between objects, the number of data set or objects, the nature and characterization of the datasets.

Moreover, the clustering time has the highest value for the ant clustering algorithm while the smallest time is for the K-means algorithm. This is valid for the three test-beds datasets.

REFERENCES


Mohamed Nour Elsayed received his M.Sc. and Ph.D. degrees in computer engineering from the Faculty of Engineering Ain Shams University, Cairo in 1987 and 1993 respectively. He taught more than twenty years at the American University in Cairo. He was a professor of Computers at Princess Nourah University, KSA. Now, he is the vice-president of the Electronics Institute, Cairo. He teaches also several computer courses at the Faculty of Engineering at MUST University, Egypt. He served as a program committee member of several computer international conferences. His research interests include high performance computing, performance analysis and artificial intelligence applications. He is a member of the IEEE, IEEE Computer Society, Computational Intelligence Society, and Language Engineering Society.

Monzer Mohamed Qasem received his M.Sc. and Ph.D. degrees from AlNeelain University and Faculty of Information Systems & Technology- Arab Academy for BANKING and Financial Sciences in 2003 and 2009 respectively. Now, he is working as an assistant professor at the Information Systems Department- Princess Nourah University, Riyadh, KSA. His area of interests includes but not limited to: artificial intelligence, data warehousing, data mining, and information systems security.