



An Improvement to the K-Means and K-MEDOID using D-M Clustering Approach

Amaneeep Jhakhhar

M.Tech. (CSE) Student,

ITM University, Gurgaon, Haryana, India

Abstract: Clustering is an effective data mining operation used to divide the available data set in small segments in which each segment contain similar data. The effectiveness of clustering algorithm can be measured in terms of similarity in the elements of same cluster and the difference with other cluster elements. Clustering is used to generate the sample space for some major applications. It is effective in data organization and information retrieval. In this paper the exploration of two main clustering approaches is defined called KMeans Clustering, K-Memoid Clustering. A new approach called DM approach is suggested in this work. These approaches are implemented on a sample dataet and the analysis is performed under different parameters. The results show that the DM Approach provided more effective Results than K-Means and K-Memoid Appraoch Clustering Approach.

Keywords: Clustering, KMeans, K-Memoid, Segmentation.

I. INTRODUCTION

Clustering is the process of grouping a set of objects into clusters, with the objective of maximizing intra-cluster similarity and minimizing inter-cluster similarity. According to Han and Kamber [1], clustering has its roots in many areas, including data mining, statistics, biology, and machine learning. The clustering problem has been addressed in many contexts and by researchers in many disciplines ranging from sociology and psychology, to commerce, biology, and computer science. This reflects its wide appeal and usefulness as an important step in exploratory data analysis, grouping, decision making, data mining, Data retrieval, image segmentation, and pattern classification[3][4]. Application of clustering algorithms include fraud detection in the telecommunications industry, IDS, Web Data clustering, Wireless sensor networks, Web Mining, Data Mining, Information Retrieval etc. In business, clustering can help marketers to categorize their customers based upon their purchasing behaviour and find their target customer group. In biology, it can be applied to derive plant and animal taxonomies categorize genes with similar functionality, and gain insight into structures inherent in populations. It can also be used to help cluster related Data on the Web. Clustering is an unsupervised learning (unlike classification) where no class labels are provided in advance, in some cases clustering can be done in a semi-supervised fashion where use of some background knowledge is made. As stated by Han and Kamber [1] clustering algorithms can be categorized as follows:

A. Partitioning Methods

A partitioning algorithm partitions a dataset of n objects into clusters ($k \leq n$). The K-Means and K-medoids methods are well known partitioning algorithms. The K-Means algorithm is a centroid based technique in which the cluster similarity is measured in regard to the mean value of the objects in a cluster (i.e. each cluster is represented by the centre of the cluster. Strength of the k-means is that it is relatively efficient with a complexity $O(nkt)$, where n is number of objects, k is number of clusters, and t is number of iterations. It often terminates at a local optimum [5]. The weakness is that number of clusters 'k' needs to be predefined which makes it unsuitable for discovering clusters with non-convex shapes or clusters with different sizes. It is also sensitive to noise and presence of outliers. Other variants of K-Means viz. Expectation-Maximization and K-modes can be studied in [2]. Unlike K-means in K-medoids each cluster is represented by one of the objects in the cluster. K-medoid algorithms include PAM (Kaufman and Rousseeuw, 1987), CLARA (Kaufmann and Rousseeuw, 1990) and CLARANS (Ng, R.T and Jiawie Han, 1994)

B. Hierarchical Methods

Unlike partitioning algorithms in which the number of clusters need to be defined in advance, this is not required in hierarchical clustering methods. These methods provide a tree view of clusters also called dendograms. These methods can be categorized as follows:

C. Agglomerative (bottom up approach)

Agglomerative clustering methods begin with each item in its own cluster, and then, in a bottom-up fashion, repeatedly merge the two closest groups to form a new cluster.

D. Divisive (top down approach)

Split a cluster iteratively. It starts with all objects in one cluster and subdivides them into smaller pieces. Some more useful clustering algorithms produced as a result of integration of hierarchical and distance-based

algorithms are: BIRCH [6], CURE [7] and CHAMELEON [8]. ROCK [9] is a hierarchical clustering algorithm for categorical data.

E. Density Based Methods:

Developed to discover clusters with arbitrary shapes. Clustering is based on density (local cluster criterion), such as density-connected points. Some interesting studies include DBSCAN, CLIQUE, DENCLUE and OPTICS [1].

F. Grid Based Methods

The grid-based clustering approach makes use of a multi-resolution grid data structure. All clustering operations are performed on a grid structure which is produced by quantizing object space into a finite number of cells. Its main advantage is fast processing time, which is typically independent of the number of data objects, yet dependent on only the number of cells in each dimension in the quantized space. Some typical algorithms are STING (Wang, Yang and Mutz in 1997), WaveCluster (Sheikholeslami, Chatterjee and Zhang in 1998), CLIQUE (Agrawal, Gehrke, Gunopulos, Raghavan in 1998) and GRIDCLUST (Schikuta 1997).

G. Machine Learning Methods

Grouping of data is based on probability density models (i.e. based on how many features are the same). Unlike conventional clustering, which primarily identifies groups of similar objects, conceptual clustering (a form of clustering in machine learning) goes one step further by also finding characteristic descriptions for each group, where each group represents a concept or class. Hence, conceptual clustering is a two-step approach: clustering is performed first, followed by characterization. COBWEB [1] is a popular conceptual clustering algorithm.

II. Data Clustering

Clustering of Data is a difficult task in Data data mining owing to the high-dimensionality and sparse nature of Data. It requires efficient algorithms which can address this high dimensional clustering problem. Data clustering plays an important role in web based applications and Data data mining. Major applications of Data clustering include [2].

Effective Search results: By search results we mean the Data that were returned in response to a query. Data clustering is applied in web search engines (The automatic generation of a taxonomy of Web Data like that provided by Yahoo, Google etc.) to improve search results. Its benefit is more effective information presentation to the users.

Cluster-based effective navigation: This is an interesting alternative to keyword searching, the standard information retrieval paradigm. This is extremely useful in cases where users prefer browsing over searching when they are unsure about which search terms to use. Its benefit is provision of alternate user interface i.e. 'search without typing'. The result of a query is now matched to a cluster rather than to each Data thus reducing the search space.

Collection clustering: As an alternative to the user-mediated iterative clustering [2], we can also compute a static hierarchical clustering of a collection that is not influenced by user interactions e.g. Google News. In case of web news articles, we frequently need to recompute the clustering to make sure that users can access the latest breaking events. Data clustering is well suited for access to a collection of news stories since news reading is not really search, but rather a process of selecting a subset of stories about recent events. It is useful for effective information browsing for exploratory browsing.

The standard Data clustering process consists of the following steps

A. Preprocessing

The Data to be clustered are in an unstructured format therefore some pre-pre-processing steps need to be performed before the actual clustering begins. The pre-processing includes Tokenization, Stemming of Data words, and Stopword removal.

Tokenization means tagging of words where each token refers to a word in the data.

Stemming involves conversion of various forms of a word to the base word. E.g. 'computing' and 'computed' will be stemmed to the base word 'compute'. Similarly 'sarcastically' is stemmed to the word 'sarcasm'. The Porter's Algorithm [10] is the most popular stemming technique for English Language Data. Snowball is a popular tool using this stemming algorithm. [11]

Stop word removal: Stop words are the words present in Data which do not contribute in differentiating a collection of Data hence, are removed from the Data. These are basically articles, prepositions, and pronouns. Standard stoplists are available but they can be modified depending upon the kind of dataet to be clustered.

Dimensionality Reduction is sometimes done where high-dimensionality becomes a curse at times. Techniques useful for this process are Principal component analysis (PCA) [12], Latent Semantic Indexing (LSI) [13].

III. Proposed Work

Data clustering is a fundamental operation used in unsupervised Data organization, automatic topic extraction, and information retrieval. Fast and high-quality Data clustering algorithms play an important role in effectively navigating, summarization and organization of information. The Data to be clustered can be web news articles, abstracts of research papers etc. This thesis provides an approach to Data clustering problem by dividing a set of Data into clusters of related Data on basis of *feature sensitive clustering* where a ‘feature’ is the concept contained in the Data. The Data to be clustered are chosen to be a collection web news articles as they are in abundance on the web and need to be categorized accurately. A hybridized approach involving Swarm intelligence based algorithm *Particle Swarm Optimization (PSO)* with traditional partitioning clustering algorithms *K-means* and Fuzzy-CMeans has been applied to address such high-dimensional clustering. The problem statement adopted is the one provided by Christopher et al. [2] as a formalized statement of Data clustering:

(i) a set of Data $D = \{d_1, d_2, \dots, d_n\}$

(ii) a desired number of clusters k

(iii) an objective function f that evaluates the quality of clustering.

We want to compute a mapping $\gamma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, k\}$, that minimizes (or in other cases, maximizes) the objective function f subject to some constraints.

The proposed clustering approaches KPSO selected for implementation and comparison are hybrids of traditional partitioning K-Means algorithm algorithm with Swarm Intelligence based Particle Swarm Optimization (PSO) technique.

A. K-Means Algorithm

K-Means algorithm is the most popular traditional partitioning clustering algorithm. It is the simplest unsupervised learning algorithms [11] used to solve well-known clustering problems.

For Data clustering problem this algorithm assigns every Data to one of the K clusters. Ideal cluster in Kmeans will a sphere with the centroid as its center of gravity [70]. The goal of K-means is to minimize the average distance of Data from their cluster centers, where a cluster center is taken as the mean (or centroid) of the Data in a cluster. The centroid μ of the Data in a cluster ω is computed as below:

$$\mu(\omega) = \frac{1}{|\omega|} \sum_{x \in \omega} x$$

The K-means algorithm is composed of the following steps:

Table 1: K-Means Algorithm

Input: //Set of n items to cluster $D_n = \{d_1, d_2, d_3, \dots, d_n\}$ //No. of clusters (temporary cluster) randomly chosen i.e. k //So below, K is set of subsets of D as temporary clusters and C is set of centroids of those clusters $K = \{k_1, k_2, k_3, \dots, k_k\}$ $C = \{c_1, c_2, c_3, \dots, c_k\}$ Where $k_1 = \{d_1\}$, $k_2 = \{d_2\}$, $k_3 = \{d_3\}$ and $\dots, k_k = \{d_k\}$ $c_1 = d_1, c_2 = d_2, c_3 = d_3, \dots, c_k = d_k$ //Here $k \leq n$
Out put: //K is set of subsets of D as final clusters and C is set of centroids of these clusters $K = \{k_1, k_2, k_3, \dots, k_k\}$ $C = \{c_1, c_2, c_3, \dots, c_k\}$
Algorithm: <i>k-Means (D, K, C)</i> 1. Repeat until (No change in centroid) 2 for $i = 1$ to n do 3. Determine distance (m) between d_i and each centroid c_j of any k_j in K such that m is minimum. ($1 \leq j \leq k$) 4. Assign d_i to cluster k_j 5. Calculate new mean (centroid) for each cluster k_j . ($1 \leq j \leq k$).

B. K-Memoid Algorithm

K-Memoid algorithm clusters n data objects in to k number of clusters. It asks from the user to enter data set and number of clusters to be formed. Initially, assign first k objects as k individual clusters. As each cluster have single data object so medoid of cluster is data object itself. In next step compute the distance of each data object to each medoid, store the total distance and then cluster the data objects according to minimum distance.

Now, clusters are updated by randomly selecting the new non-medoid objects as medoid and compute the total distance w.r.t. new medoids. If the new distance is less then the old distance then replace the non-medoid object with medoid object.

Table 2: K-Memoid Algorithm

Input: //Set of n items to cluster
 $D_n = \{d_1, d_2, d_3, \dots, d_n\}$
 //No. of clusters (temporary cluster) randomly chosen i.e. k
 //So below, K is set of subsets of D as temporary clusters and C is set of Medoids of those clusters
 $K = \{k_1, k_2, k_3, \dots, k_k\}$
 $M = \{m_1, m_2, m_3, \dots, m_k\}$
 Where $k_1 = \{d_1\}$, $k_2 = \{d_2\}$, $k_3 = \{d_3\}$ $k_k = \{d_k\}$ and $m_1 = d_1$, $m_2 = d_2$, $m_3 = d_3$ $m_k = d_k$
 //Here $k \leq n$
Out put: // K is set of subsets of D as final clusters and M is set of Medoids of these clusters
 $K = \{k_1, k_2, k_3, \dots, k_k\}$
 $M = \{m_1, m_2, m_3, \dots, m_k\}$
Algorithm:
 k -Medoids (D, K, M)
 1. Repeat
 2. for $i = 1$ to n do
 3. Assign each remaining object to the cluster with the nearest representative object;
 4. randomly select a non-medoid object, m_{rand}
 5. Compute the Distance, after swapping any medoid m_j with m_{rand}
 6. **If NewDistance < OldDistance then**
 swap m_j with m_{rand} and form the new clusters
 7. until (No change in Medoids).

C. D-M Clustering

Density Means clustering approach is the new methodology to cluster the data objects into number of groups, which is unknown initially. Number of groups (clusters) is some positive integer. The grouping is done by measuring the distance between object and centroid. Objects are iteratively grouped into the existing clusters or a new cluster formation is done with those objects based up on the Distance Determination Factor. Thus the purpose of this clustering is to classify the data on the basis of distance dynamically. It could improve the chances of finding the global optima with careful selection of initial cluster. In this algorithm data objects are stored in secondary memory and transferred to main memory one at a time. Only the cluster representatives are stored permanently in main memory to alleviate space limitations. In D-M Clustering, the input is Data Set and Distance Determination Factor (DDF) and on the basis of DDF it is decided that any particular data objects will lie in which cluster. Firstly, measures the distance between the data object and centroids and do the followings with the centroid having minimum distance with that data object:

If ($Distance > DDF$) then

Design the New Cluster taking these objects as centroid

Else

Combine the object with cluster, having minimum centroid distance from the data object. Thus, in this way we divide the data points in different set of clusters and there is no prior specification of any k to decide the cluster beforehand.

Table 3: D-M Clustering

Input: //Set of n objects to cluster
 $D = \{d_1, d_2, d_3, \dots, d_n\}$
Output: // K is set of subsets of D as final clusters and C is set of centroids of those clusters
 $K = \{k_1, k_2, k_3, \dots, k_k\}$
 $C = \{c_1, c_2, c_3, \dots, c_k\}$
Algorithm:
 Density Means cluster algorithm(D)
 1. let $k=1$
 2. $k_k = \{d_k\}$
 3. $K = \{k_k\}$
 4. $C_k = d_i$
 5. Assign some constant value to DDF
 6. for $i = 2$ to n do

7.	Determine distance (m) between d_i and each centroid C_j of any k_j in K such that m is minimum.
($1 \leq j \leq k$)	
8.	if ($m \leq DDF$) then //DDF-Distance Determination Factor
	Maximum distance allowed
9.	$k_j = k_j \cup d_i$
10.	Calculate new mean (centroid c_j) for cluster k_j ;
11.	else $k = k+1$
12.	$k_k = d_i$
13.	$K = K \cup k_k$

IV. RESULTS

The presented work is implemented in matlab environment. The results obtained from the work are given here under

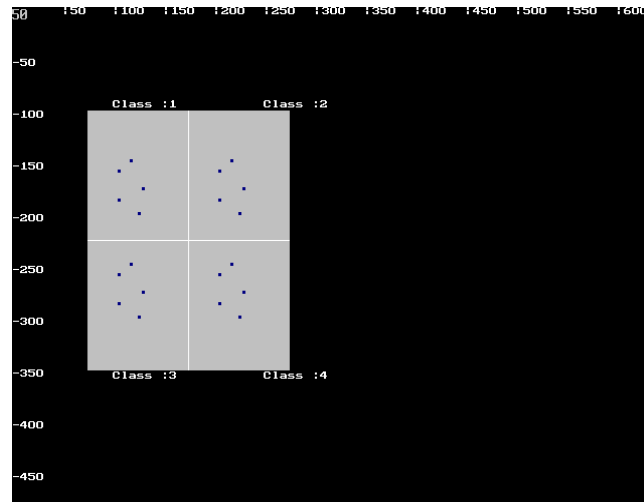


Figure: 1

Here figure 1 is showing the clustering process. Here the clustering segments with relative members. The distance based match is shown in figure 1.

Table 3: Comparison Table of Clustering Approaches

Name of algorithm	Coefficient of Variation
k-means	0.1360
k-medoids	0.2990
D-M	0.2990

The coefficient of variation is shown in table 3. The results show that the proposed D-M approach reduced the variation over data elements.

Table 4: Complexity Comparison

Name of algorithm	Computational Complexity
k-means	$O(nkt)$
k-medoids	$O(k(n-k)^2)$
D-M	$O(n^1)$

The comparison of proposed approach with existing K-Means and K-Method approaches is shown in table 4. The results show that the presented approach is effective in terms of accuracy and the efficiency.

V. CONCLUSION

In this present work, A new clustering approach is presented called D-M approach. This proposed approach is distance match based on determination method. The results show that the presented approach is effective to provide the effective outcome so that more accuracy and efficiency is obtained. The results are compared with two existing approaches called KMeans and K-Memoid Algorithms. The results show, the presented approach is more effective than existing.

REFERENCES

- [1] Michael Steinbach, "A Comparison of Data Clustering Techniques".
- [2] Khaled Hammouda, "Collaborative Data Clustering".
- [3] Benjamin C.M. Fung, "Hierarchical Data Clustering Using Frequent Itemsets".
- [4] Bader Aljaber, "Data Clustering of Scientific Data Using Citation ConData".
- [5] Oren Zamir, "Web Data Clustering: A Feasibility Demonstration".
- [6] Wei Xu, "Data Clustering Based On Non-negative Matrix Factorization".
- [7] Alan F. Smeaton, "An Architecture for Efficient Data Clustering and Retrieval on a Dynamic Collection of Newspaper Data".
- [8] Ye-Hang Zhu, "Data Clustering Method Based on Frequent Co-occurring Words".
- [9] Andreas Hotho, "Wordnet improves Data Clustering".
- [10] M. Shahriar Hossain, "GDClust: A Graph-Based Data Clustering Technique".
- [11] Mihai Surdeanu, "A Hybrid Unsupervised Approach for Data Clustering".
- [12] Anna Huang, "Similarity Measures for Data Clustering".
- [13] Md Maruf HASAN, "Data Clustering: Before and After the Singular Value Decomposition".
- [14] Y. HE, "MINING A WEB CITATION DATABASE FOR DATA CLUSTERING", Applied Artificial Intelligence.
- [15] Rekha Baghel, "A Frequent Concepts Based Data Clustering Algorithm", International Journal of Computer Applications (0975– 8887).