A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets

Madjid Khalilian, Norwati Mustapha, MD Nasir Suliman, MD Ali Mamat

Abstract- **Data clustering is an unsupervised method for extraction hidden pattern from huge data sets. Having both accuracy and efficiency for high dimensional data sets with enormous number of samples is a challenging arena. For really large and high dimensional data sets, vertical data reduction should be performed prior to applying the clustering techniques which is performing dimension reduction, and the main disadvantage is sacrificing the quality of results. However, because dimensionality reduction methods inevitably cause some loss of information or may damage the interpretability of the results, even distorting the real clusters, extra caution is advised. Furthermore, in some applications data reduction is not possible e.g. personal similarity, customer preferences profiles clustering in customer recommender system or data by which is generated sensor networks. Existing clustering techniques would normally apply in a large space with high dimension; dividing big space into subspaces horizontally can lead us to high efficiency and accuracy. In this study we propose a method that uses divide and conquer technique with equivalency and compatible relation concepts to improve the performance of the K-Means clustering method for using in high dimensional datasets. Experiment results demonstrate appropriate accuracy and speed up.**

Index Term-Clustering, K-Means Algorithm, Personality Similarity, High Dimensional Data, Equivalency Relation

I. Introduction

Many applications need to use unsupervised techniques where there is no previous knowledge about patterns inside samples and its grouping, so clustering can be useful. Clustering is grouping samples base on their similarity as samples in different groups should be dissimilar. Both similarity and dissimilarity need to be elucidated in clear way. High dimensionality is one of the major causes in data complexity. Technology makes it possible to automatically obtain a huge amount of measurements. However, they often do not precisely identify the relevance of the measured features to the specific phenomena of interest. Data observations with thousands of features or more are now common, such as profiles clustering in recommender systems, personality similarity, genomic data, financial data, web document data and sensor data. However, high-dimensional data poses different challenges for clustering algorithms that require specialized solutions. Recently, some researches have given solutions on high-dimensional problem. In the well-known survey [1] the problem is introduced in a very illustrative way and some approaches are sketched. There is no clear distinction between different sub problems (axis-parallel or arbitrarily oriented) and the corresponding algorithms are discussed without pointing out the underlying differences in the respective problem definitions.

Our main objective is proposing a framework to combine relational definition of clustering space and divide and conquer method to overcome aforementioned difficulties and improving efficiency and accuracy in K-Means algorithm to apply in high dimensional datasets. Despite previous study in dividing whole space into subspaces vertically based on object's features [1-6] we apply a horizontal method to divide entire space into subspaces based on objects. We conducted some experiments on real world dataset (personality similarity) as an example of special groups of applications that are silent other methods for presenting suitable solution.

In section 2 preliminaries for research are described. Proposed method has been explained in section 3, in section 4 we demonstrate our methodology, datasets description, experiment procedure. Finally we have conclusion and future works.

II. Preliminaries

Clustering is grouping samples into some classes unsupervised with unknown label. Let consider space with samples which are defined with vectors: $S = \{V_1, V_2, ..., V_n\}$ as each vector has own properties that is shown with v_1, v_2, so $V_{ij} \in R^d$ means j^th property of V_i and R^d is a space with d dimensions.

A. K-Means Clustering Algorithm

K - Means is regarded as a staple of clustering methods, due to its ease of implementation. It works well for many practical problems, particularly when the resulting clusters are compact and hyper spherical in shape. The

M. KHALILIAN is with the Islamic Azad University, Karaj Branch; Iran. He is now PhD candidate in Faculty of Computer Science and Information Technoloy, University Putra Malaysia (E-mail: khalilian@ieee.org).

Dr Norwati Mustapha is with the Computer Science Department, Faculty of Computer Science and Information Technoloy, University Putra Malaysia (E-mail: Norwati@fsktm.upm.edu.my).

Dr Nasir Suliman is with the Computer Science Department. He is the Associated Prof. in Faculty of Computer Science and Information Technoloy, University Putra Malaysia (E-mail: Nasir@fsktm.upm.edu.my).

Dr Ali Mamat is with the Computer Science Department. He is the Associated Prof. in Faculty of Computer Science and Information Technoloy, University Putra Malaysia (E-mail: Ali@fsktm.upm.edu.my).

time complexity of *K* - means is $O(N.K.d.T)$ where *T* is the number of iterations. Since *K*, *d*, and *T* are usually much less than *N*, the time complexity of *K* - means is approximately linear. Therefore, *K* - means is a good selection for clustering large - scale data sets [7].

B. Divide AND Conquer

When the size of a data set is too large to be stored in the main memory, it is possible to divide the data into different subsets that can fit the main memory and to use the selected cluster algorithm separately to these subsets. The final clustering result is obtained by merging the previously formed clusters. This approach is known as divide and conquer [8, 9]. Specifically, given a data set with *N* points stored in a secondary memory, the divide - and - conquer algorithm first divides the entire data set into *r* subsets with approximately similar sizes. Each of the subsets is then loaded into the main memory and is divided into a certain number of clusters with a clustering algorithm. Representative points of these clusters, such as the centers of the clusters, are then picked for further clustering. These representatives may be weighted based on some rule, e.g., the centers of the clusters could be weighted by the number of points belonging to them [8]. The algorithm repeatedly clusters the representatives obtained from the clusters in the previous level until the highest level is reached. The data points are then put into corresponding clusters formed at the highest level based on the representatives at different levels.

C. Equivalency AND Similarity

We are going to describe a mathematic base for our method and definition of similarity base on relation definition. In previous studies on divide and conquer[8, 9], dividing data has been done without any prior knowledge about data but in this study a criterion is used to divide data into partitions. This criterion can be selected base on the following discussion.

Theorem.1. Each equivalence relation R can divide S into some partitions (classes) $s_1, s_2, \ldots, s_n$ where

$a)\bigcup_{i=1}^{n} s_i = S$

$b) s_i \neq \emptyset$ proof is in context of Discrete

$c) s_i \bigcap s_j = \emptyset, i \neq j$

Mathematics Structure [10, 11].

Theorem.2. Length of vector is an equivalence relation where length is defined by $L(V) = \sqrt{\sum_{i=1}^{d} v_i^2}$, where d and v_i are for number of dimensions and value of i^th^ feature respectively.

Proof

$a) reflexive$

$\therefore \forall V_i \in S : L(V_i) = L(V_i)$

$b) symmetric$

$\therefore \forall V_i, V_j \in S : L(V_i) = L(V_j) \Rightarrow L(V_j) = L(V_{ji})$

$c) transitive$

$\therefore \forall V_i, V_j, V_k \in S : L(V_i) = L(V_j) \wedge L(V_j) = L(V_k) \Rightarrow L(V_i) = L(V_k)$

Outcome of theorems 1, 2 is the important result that implies length of vector can divide our problem space into subspaces with equivalency property. Although, vectors in one subspace are in the same level but they might be in different directions. The radius of subspace is L (V_i) for all vectors inside of that subspace as V_i belongs to subspace.

Theorem.3. Similarity is a compatible relation where similarity is defined by Minkowski distance [12]:

$D_n(V_i, V_j) = \left(\sum_{k=1}^{d} \left| V_{ik} - V_{jk} \right|^n \right)^{1/n}$ or cosine measure [13-15] that is used inner product for similarity among vectors: $\cos(V_i, V_j) = \frac{V_i^T \cdot V_j}{|V_i||V_j|}$

Proof

$a) reflexive$

$\therefore \forall V_i \in S : D_n(V_i, V_i) = 0, \cos(V_i, V_i) = 1$

$b) symmetric$

$\therefore \forall V_i, V_j \in S : D_n(V_i, V_j) = D_n(V_j, V_i), \cos(V_i, V_j) = \cos(V_j, V_i)$

Similarity is not an equivalence relation because it doesn't have transitive property. For example consider 3 vectors in space V1, V2, V3. If we define threshold of angle α for being similarity, it is clear that V_1 is not similar with V_3 base on definition because the angle is 2α more than threshold value.

Result of theorems 2, 3 $\therefore D_n \subset L$. We are going to device an algorithm in two steps. First it divides entire space into some subspaces based on length of vectors. As it was mentioned, samples in each subspace are equivalent but not necessarily similar. In second phase K-Means algorithm is applied in each subspace. It is suitable to overcome clustering problem for large datasets with high dimensions instead of using clustering on entire space

- Space Dividing: base on length of vector which is equivalency relation, we divide space of problem into some subspaces. This has been developed by K-Means algorithm. Length of vectors are input for K-Means algorithm and output will be some subspaces or partitions which elements inside them are equivalent and ready to clustering. In other word all samples in one partition have almost same size but might be dissimilar.

- Subspaces Clustering: after finding subspaces, clustering algorithm is applied on each subspace and outcomes final group. Although samples in different subspaces may be similar base on cosin criterion but they are in different levels [16]
- Clusters Validity: quality of clusters should be demonstrated. K-Means uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well-separated as possible. You can control the details of the minimization using several optional input parameters to K-Means, including ones for the initial values of the cluster centroids, and for the maximum number of iterations. To get an idea of how well-separated the resulting clusters are, we can make a silhouette plot using the cluster indices output from K-Means. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters. This measure ranges from +1, indicating points that are very distant from neighboring clusters, to 0, indicating points that are not distinctly in one cluster or -1, indicating points that are probably assigned to the wrong cluster. A more quantitative way to compare the two solutions is to look at the average silhouette values for the two cases. Average silhouette width values can be interpreted as follows: [17]

0.70—1.0	A strong structure has been found
0.50—0.70	A reasonable structure has been found
0.25—0.50	The structure is weak and could be artificial
<0.25	No substantial structure has been found

III. EXPERIMENTAL RESULTS

We compare our method according to its quality and speed up using a specific data set. In this section data set is described and proposed algorithm is compared with K-Means algorithm and Hierarchical Clustering considering quality of clustering and speed up.

A. Data set description

We conduct our experiments on a data set which data is gathered from PEIVAND[1] web site. This web sit is for finding suitable partners who are very similar from point of personality's view for a person. Based on 8 pages of psychiatric questions personality of people for different aspects is extracted. Each group of questions is related to one dimension of personality. To trust of user some questions is considered and caused reliability of answers are increased. Data are organized in a table with 90 columns for attributes of people and 704 rows which are for samples. There are missing values in this table because some questions have not been answered, so we replaced them with 0. On the other hand we need to calculate length of each vector base on its dimensions for further process. All attributes value in this table is ordinal and we arranged them with value from 1 to 5, therefore normalizing has not been done. There is not any correlation among attributes and it concretes an orthogonal space for using Euclidean distance. All samples are included same number of attributes.

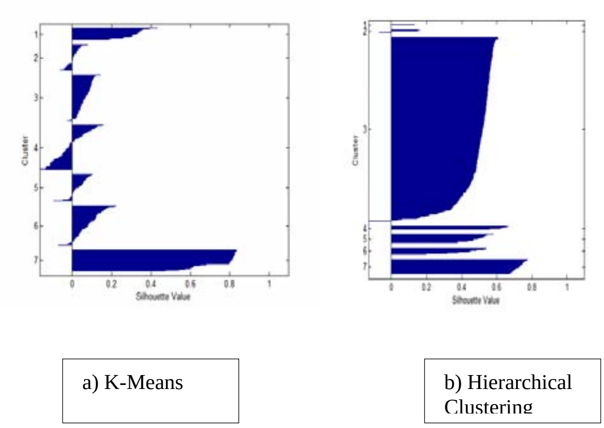
B. Experimental setup

In all experiments we use MATLAB software as a powerful tool to compute clusters and windows XP with Pentium 2.1 GHZ. The K-Means, hierarchical clustering and proposed algorithms are applied on the above mentioned data set. As a similarity metric, Euclidean distance has been used in each algorithm. Having a fare compression we consider 7 clusters as a final result in all algorithms, so number of clusters for K-Means and max number of clusters for HC is considered 7 and in proposed algorithm we consider 2 subspaces where one of them has been grouped into 4 clusters and another 3 clusters. Two groups of experiment have been conducted for quality and efficiency.

C. Quality compression

In this section we demonstrate our experiments result for clusters quality. As it was mentioned we use silhouette value for quality measuring. Negative value means wrong clustering. As shown in Figure. 3 although quality of clustering is much better in HC method but there are still having some wrong clustering objects.

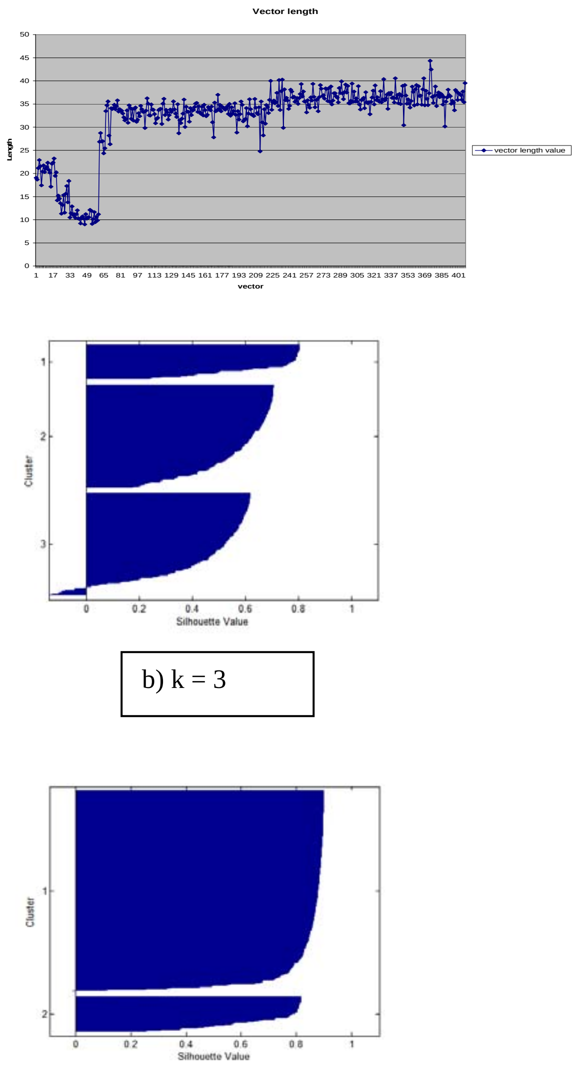
[1] www.peivand.org



a) K-Means

b) Hierarchical Clustering

Figure 1 a) K-Means silhouette value with mean=0.1242
b) Hierarchical Clustering silhouette value with mean=0.502



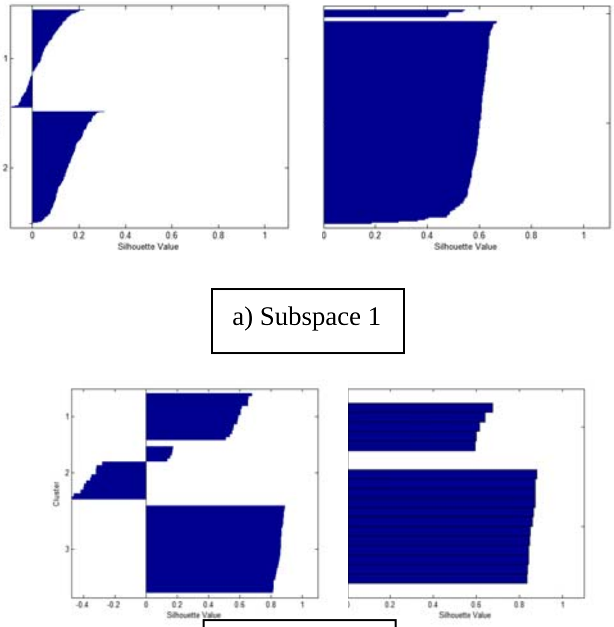
b) k = 3

c) k = 2

Figure 2 Vector length clustering: a) vector length values, b) 3 subspaces, c) 2 subspaces

Dividing entire space into subspaces needs to have a criterion where we use length of vector for this purpose. It is possible to have others criteria for dividing e.g. distance from source or angle value between vector and one of axis. Equivalency is a necessary condition for chosen criterion to divide space into equivalent subspaces as it described before and it should be proved. We divide our problem space into 2 subspaces base on experiment. Finding a suitable value for K is easy because vectors length data is one dimension data (figure.4.a.).

In next step we should apply K-Means algorithm in each subspace and achieve final results. To compare with other algorithms, first subspace's samples are grouped into 3 clusters and second grouped 4 clusters. We select K-Means for this step because this algorithm is easy to implement, scalable and its complexity is linear. Quality for clusters has been demonstrated in figure.5.



a) Subspace 1

b) Subspace 2

Figure 3 subspaces clustering.

Nested clustering is possible in proposed method, means each subspace is included some sub clusters which can be considered a new space. In this fashion we are able to concrete structured clustering like hierarchical method and achieve high quality clusters. In subspace 1 we have 2 clusters where cluster 1 doesn't have good quality and it should be clustered again. We have same problem in subspace 2 and we behave like previous subspace. In figure.6 we have shown the quality of two clusters' samples for instance.

D. Speed up compression

This set of experiments is about the runtime of K-Means, HC and the proposed algorithm with different initializations. Figure 7 shows the evaluation results using our data set. The evaluations were conducted for 10 test runs. It is clear that the proposed algorithm has the lowest time consuming most of the time, only HC is competitively. Another advantage is nearly linear trend means stability in execute of time (figure.8.). The proposed algorithm is scalable because it follows K-Means complexity that is linear with n number of samples, T number of iteration, d number of dimensions and k number of clusters: O (T.K.N.D). Both HC and K-Means use pair's similarity measurement that is very time consuming. By dividing space into subspaces, run time would be reduced because pair's similarity measurement doesn't need to apply for whole samples. Each sample will be compared only with samples in its subspace.

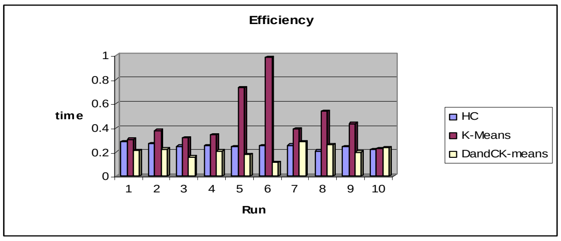


Figure 6 Execution time for DANDC K-Means comparing HC and K-Means.

IV. Conclusions and Future Works

Combining advantages of K-Means and divide and conquer strategy can help us in both efficiency and quality. Besides simulating of HC is possible with recursive intrinsic divide and conquer method and creating nested clusters. HC algorithm can construct structured clusters. Although HC yields high quality clusters but its complexity is quadratic and is not suitable for huge datasets and high dimension data. In contrast K-Means is linear with size of data set and dimension and can be used for big datasets that yields low quality. In this paper we present a method to use both advantages of HC and K-Means by introducing equivalency and compatible relation concepts. By these two concepts we defined similarity and our space and could divide our space by a specific criterion. Many directions exist to improve and extend the proposed method. Different applications can be used and examined the framework. Text mining is an interesting arena. Based on this method data stream processing can be improved. Data type is another direction to examine this method. In this study K-Means has been used for second phase whereas we can use other clustering algorithms e.g. genetic algorithm, HC algorithm, Ant clustering[18], Self Organizing Maps [19], etc. Determining number of sub spaces can be studied as important direction for the proposed method.

References

[1] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *ACM SIGKDD Explorations Newsletter*, vol. 6, pp. 90-105, 2004.

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, p. 863.

[3] R. Agrawal, J. E. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," Google Patents, 1999.

[4] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma, "Subspace clustering of high dimensional data," 2004.

[5] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," 2003, p. 186.

[6] H. P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," 2009.

[7] R. XU and I. DONALD C. WUNSCH, *clustering*: A JOHN WILEY & SONS, INC., PUBLICATION, 2008.

[8] Guha, Meyerson, A. Mishra, N. Motwani, and O. C. . "Clustering data streams: Theory and practice ." *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 515-528, 2003.

[9] A. Jain , M. Murty , and p. Flynn " Data clustering: A review.," *ACM Computing Surveys*, vol. 31, pp. 264-323, 1999.

[10] P. C. Biswal, *Discrete Mathematics and Graph Theory*. New Delhi: Prentice Hall of India, 2005.

[11] M. Khalilian, *Discrete Mathematics Structure*. Karaj: Sarafraz, 2004.

[12] K. J. Cios, W. Pedrycz, and R. M. Swiniarsk, "Data mining methods for knowledge discovery," *IEEE Transactions on Neural Networks*, vol. 9, pp. 1533-1534, 1998.

[13] V. V. Raghavan and K. Birchard, "A clustering strategy based on a formalism of the reproductive process in natural systems," 1979, pp. 10-22.

[14] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Inf Process Manage*, vol. 24, pp. 513-523, 1987.

[15] G. Salton, "Automatic text processing: the transformation," *Analysis and Retrieval of Information by Computer*, 1989.

[16] M. Khalilian, N. Mustapha, M. N. Sulaiman, and F. Z. Boroujeni, "K-Means Divide and Conquer Clustering," in *ICCAE*, Thiland, Bangkok, 2009, pp. 306-309.

[17] I. Kononenko and M. Kukar, *machin learning and data mining*. Chichester, UK: Horwood Publishing, 2007.

[18] U. Boryczka, "Finding groups in data: Cluster analysis with ants," *Applied Soft Computing Journal*, vol. 9, pp. 61-70, 2009.

[19] D. Isa, V. P. Kallimani, and L. H. Lee, "Using the self organizing map for clustering of text documents," *Expert Systems With Applications*, vol. 36, pp. 9584-9591, 2009.