International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:4, 2010

An Efficient and Generic Hybrid Framework for High Dimensional Data Clustering

Dharmveer Singh Rajput, P. K. Singh, Mahua Bhattacharya

***Abstract*—**Clustering in high dimensional space is a difficult problem which is recurrent in many fields of science and engineering, e.g., bioinformatics, image processing, pattern reorganization and data mining. In high dimensional space some of the dimensions are likely to be irrelevant, thus hiding the possible clustering. In very high dimensions it is common for all the objects in a dataset to be nearly equidistant from each other, completely masking the clusters. Hence, performance of the clustering algorithm decreases.

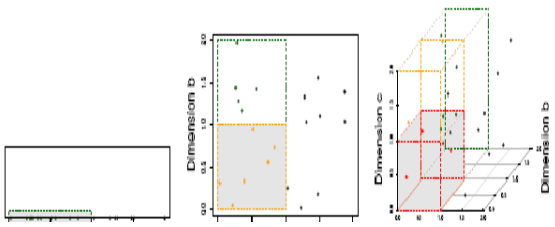
In this paper, we propose an algorithmic framework which combines the (reduct) concept of rough set theory with the k-means algorithm to remove the irrelevant dimensions in a high dimensional space and obtain appropriate clusters. Our experiment on test data shows that this framework increases efficiency of the clustering process and accuracy of the results.

***Keywords*—**High dimensional clustering, sub-space, k-means, rough set, discernibility matrix.

I. Introduction

CLUSTERING is an effort to classify similar objects in the same groups. The obtained clusters are good when the members of a cluster have high degree of similarity with each other (internal homogeneity) and have high degree of dissimilarity with members of other clusters (external homogeneity) [1], [2]. The similarity between objects is often determined using a *distance measure*, e.g., Euclidean distance, over the various dimensions in the dataset [13]. The rapid development of data collection technology has resulted in many high dimensional and large scale datasets in different areas such as biology, geographic, finance and telecommunication [17]. An increase in dimension brings in a lot of difficulties to traditional clustering methods. Most clustering methods encounter difficulties when the dimensionality of the dataset grows high. This is because in high dimensional datasets, only a small number of dimensions are usually relevant to clusters and data in the irrelevant dimensions often produce much noise and mask the real clusters from being discovered. Moreover, an increase in dimensionality brings in *curse of dimensionality* which makes clustering more challenging in high dimensional spaces [4].

As the dimensionality increases the points tend to get scattered away from the center towards the boundaries of the enclosing hyper sphere. Furthermore, as dimensionality increases distance measure becomes increasingly meaningless; there may not be much difference between the closest and farthest neighbor for a given point. This curse of dimensionality is illustrated in Fig. 1 which shows how additional dimensions spread out the points in a sample dataset. The sample dataset contain 20 points randomly placed between 0 and 2 in each of three dimensions. Figure 1(a) shows the data projected onto one axis. The points are very close together with approximately half of them in a one unit sized bin. Figure 1(b) shows the same dataset stretched into two dimensions. The addition of another dimension spread the points out along another axis, pulling them further apart. Now only about a quarter of the points fall into a unit sized bin. In Fig. 1(c) addition of third dimension spreads the data further apart. A one unit sized bin now holds only about one eighth of the points. If we continue to add dimensions, the points will continue to spread out until they are all almost equally far apart and distance is no longer very meaningful.



Dimension a ***Dimension a*** ***Dimension a***

a) 11 Objects in One Unit Bin (b) 6 Objects in One Unit Bin (c) 4 Objects in One Unit Bin

Fig. 1 The curse of dimensionality. Data in only one dimension is relatively tightly packed. Adding a dimension stretches the points across that dimension, pushing them further apart. Additional dimensions spread the data even further making high dimensional data extremely sparse [13].

Feature selection attempts to discover the attributes of a dataset that are most relevant to the data mining task at hand. It is a commonly used and powerful technique for reducing the dimensionality of a problem to more manageable levels. Feature selection involves searching through various feature subsets and evaluating each of these subsets using some

Dharmveer Singh Rajput is a research scholar at ABV-Indian Institute of Information Technology and Management, Gwalior – 4740101, India. (phone: +91 751 2449827; fax: +91 751 2449313; e-mail: dharmveer@students.iiitm.ac.in).

P. K. Singh is associated with ABV-Indian Institute of Information Technology and Management, Gwalior – 474010, India as Associated Professor (e-mail: pksingh@iiitm.ac.in).

Mahua Bhattacharya is associated with ABV-Indian Institute of Information Technology and Management, Gwalior – 474010, India as Associated Professor (e-mail: mb@iiitm.ac.in).

criterion [3], [19]. Subspace clustering is an extension of feature selection that attempts to find clusters in different subspaces of the same dataset. Just as with feature selection, subspace clustering requires a search method and evaluation criteria. In addition, subspace clustering must somehow limit the scope of the evaluation criteria so as to consider different subspaces for each different cluster [4], [20].

There are several issues while determining a subspace cluster. The first concerns the determination of the proper subspace, that is, the set of dimensions that yields a good cluster in terms of the quality or semantic meaning. The second issue deals with the shape of the cluster. We may assume that the cluster is generated from a multivariate uniform or normal distribution, which yield a hyper rectangle or hyper-ellipsoid shape, respectively. The third issue is whether we assume the dimensions are independent or not. If we assume independence, then the clusters will be axis-aligned. Many recently proposed subspace clustering methods suffer from following two problems. First, the algorithms typically scale exponentially with the data dimensionality or the subspace dimensionality of clusters. Second, the clustering results are often sensitive to input parameters [4].

In this paper, we propose an algorithmic framework which combines the (reduct) concept of rough set theory with the k-means algorithm to remove the irrelevant dimensions in a high dimensional space and obtain appropriate clusters. Rest of the paper is organized as follows. Section II summarizes the previous relevant work whereas preliminaries of k-means algorithm and rough set theory are described in Section III. A and III. B respectively. Afterwards the proposed methodology is presented in Section III. C. We analyze performance of the proposed algorithm and compare the results with standard k-means algorithm in Section IV. Finally, Section V summarizes the conclusion and further scope of the work.

II. LITERATURE REVIEW

Clustering has been widely studied by the researchers. It is evident from the various survey papers available in the literatures e.g., Berkhin [11], Jain [15], Jain and Dubes [5], Jain et al. [1], and Zait and Messatfa [6]. Recent data mining texts, e.g., [2], [7], [8], and [9] include chapter on clustering. Though subspace clustering also finds its space in the literature, e.g., [10], [12], and [13] but there is a little work which deals with the subspace clustering in a comprehensive and comparative manner. Jahirabadkar and Kulkarni [19] proposed an approach ISC (Intelligent Subspace Clustering) which uses the density based clustering to find Subspace Clusters embedded in higher dimensional clusters. Domeniconi et al. [3] propose an algorithm which discovers clusters in subspaces spanned by different combinations of dimensions via local weightings of features. Niu et al. [4] discuss the problem of automatic subspace clustering. The proposed solution, SCA, has been designed to find clusters embedded in subspaces of high dimensional datasets. It employs a relation function to evaluate the relevance of every two attributes. Kriegel et al. [20] propose a new filter refinement subspace clustering algorithm FIRES, which efficiently computes maximum dimensional cluster approximation from ID clusters which can be refined to obtain the true clusters.

III. PROPOSED SOLUTION

Our proposed technique is a combination of the concept of rough set theory (reduct and core) and K-Means algorithm. Initially it uses rough set theory to find the reducts and core of high dimensional data sets by removing the irrelevant attributes and then applies the K-Means algorithm on the reducts for determining the optimum clusters. Hence, we describe the basic approach of the K-Means algorithm below followed by the basic concepts of rough set theory, before proposing hybrid algorithm.

A. K-Means Algorithm

Let $X = \{x_i \mid i = 1,..., n\}$ be the set of n d-dimensional points to be clustered into a set of K clusters, $C = \{c_k, k = 1,..., K\}$. K-Means algorithm [21] finds the partitions such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_k be the mean of cluster c_k. The squared error $J(c_k)$ between μ_k and the points in cluster c_k is defined as

$$J(c_k) = \sum_{x_i \in c_k} \| x_i - \mu_k \|^2$$

The goal of K-Means is to minimize the sum of the squared error $J(C)$, which is defined below, over all K clusters.

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \| xi - \mu k \|^2$$

Minimizing this objective function is an NP-hard problem even for K = 2 [15], [16]. Thus K-Means, which is a greedy algorithm, can only converge to a local minimum. However, a recent study [18] has shown that with a large probability K-Means could converge to the global optimum when clusters are well separated. The K-Means starts with an initial partition with K clusters and assign patterns to clusters so as to reduce the squared error. Since the squared error always decreases with an increase in the number of clusters K ($J(C) = 0$ when $K = n$), it can be minimized only for a fixed number of clusters. The main steps of K-Means algorithm are as follows:

1. Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.
2. Generate a new partition by assigning each pattern to its closest cluster centres.
3. Compute new cluster centres.

There are numerous applications of the K-mean clustering, e.g., unsupervised learning of neural network, pattern recognitions, classification analysis, artificial intelligence, image processing, machine vision. In principle, this algorithm may be applied when there are several objects and each object has several attributes and it is required to classify the objects based on the attributes. The figure 2 shows an illustration of

K-Means algorithm on a two-dimensional dataset with three clusters.

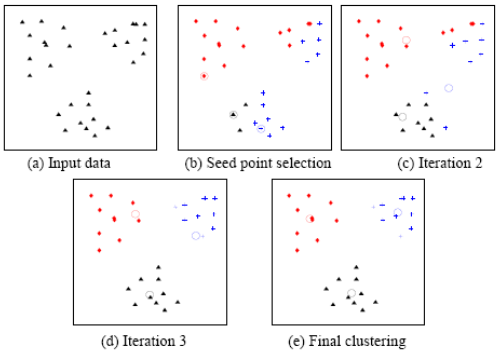


Fig. 2 Illustration of K-Means algorithm. (a) Two-dimensional input data with three clusters; (b) three seed points selected as cluster centers and initial assignment of the points to clusters; (c) & (d) intermediate iterations updating cluster labels and their centers; (e) final clustering obtained by K-Means algorithm at convergence [15]

B. Rough Set Theory

Rough set theory is able to process uncertain and incomplete information by effectively dealing with imprecise, uncertain and vague knowledge which is usually the case in real-world data sets. In the Rough set theory membership is not the primary concept. It represents a different mathematical approach to vagueness and uncertainty [14]. Rough set theory works on Information system $IS = (U, A)$ and Va. Here, U is the universe (a finite set of objects, $U = \{x_i \mid i = 1, 2,..., n\}$), A is the set of attributes (features, variables) and V_a is a set of values which is domain of attribute ***a.*** The rough set theory is explained below with an example. Consider a data set containing the results of three measurements performed on five objects as shown in table I.

TABLE I
EXAMPLE DATA SET

Object	a_1	a_2	a_3
X_1	2	1	3
X_2	3	2	1
X_3	2	2	3
X_4	1	1	4
X_5	1	1	2

Here, $U = (X_1, X_2, X_3, X_4, X_5)$, $A = (a_1, a_2, a_3)$ and the domains of attributes are $V_1 = (1, 2, 3)$, $V_2 = (1, 2)$, $V_3 = (1, 2, 3, 4)$. If the set of attributes are dependent, one can be interested in finding all possible minimal subsets of attributes. The concepts of reducts and core are two fundamental concepts of the rough set theory. To compute reducts and core, the discernibility matrix is used. The discernibility matrix has the dimension N x N. Here N is the number of elementary sets and its elements are defined as the set of all attributes which discern elementary sets X_i and X_j. If they have different values for same attribute then that attribute will be the member of discernibility matrix. For example to compute the value of (row, column) = (Set_2, Set_1) in the discernibility matrix table, we compare each attribute of X_1 to the corresponding attribute of X_2. As the values of attribute a_1, a_2, and a_3 all are different for X_1 and X_2 the attributes a_1, a_2, and a_3 are the member of discernibility matrix as shown in table II.

TABLE II
DISCERNIBILITY MATRIX

	Set_1	Set_2	Set_3	Set_4	Set_5
Set_1					
Set_2	a_1,a_2,a_3				
Set_3	a_2	a_1,a_3			
Set_4	a_1,a_3	a_1,a_2,a_3	a_1,a_2,a_3		
Set_5	a_1,a_3	a_1,a_2,a_3	a_1,a_2,a_3	a_3	

The discernibility matrix is used to compute the discernibility function. Initially discernibility function is the equation of product of sum of all the elements of the discernibility matrix. Then, it is converted in the form of sum of products. Accordingly, the discernibility function $F(A)$ of the discernibility matrix shown in table II is $(a_1+a_2+a_3)\ a_2\ (a_1+a_3)\ (a_1+a_3)\ (a_1+a_2+a_3)\ (a_1+a_2+a_3)\ (a_1+a_2+a_3)\ (a_1+a_2+a_3)\ a_3$. Finally, on solving we obtain $F(A) = a_2a_3$. Hence, the reduct of the example dataset is a set of attribute a_2 and a_3, i.e., reduct = $\{a_2, a_3\}$. It indicates that the attribute a_1 is redundant and only attribute a_2 and a_3 are sufficient for computations.

C. Propose Methodology

Our proposed solution is a combination of concepts of rough set theory (reduct & core) and basic K-Means algorithm. It works as follows:

1. Create discernibility matrix of n x n from given n x d data sets. (Here n denotes the number of objects and d is denotes the dimensions).
2. Calculate discernibility function and find out the reduct.
3. Select K initial partitions randomly from the reduct data set.
4. Generate new partition by assigning each object to its closest cluster centres.
5. Compute new cluster centres.
6. Repeat step 4 and 5 until cluster membership stabilizes.

First two steps in the algorithm uses rough set theoretic approach as attribute subset selection method and removes the irrelevant attributes. Further in step 3-6, it uses standard k-means algorithm to obtain appropriate clusters with minimal set of relevant attributes.

IV. EXPERIMENTAL RESULT

We used the standard dataset available at the Cologne University, Germany [22]. It contains the ingredients of mammal's milk of 25 animals. The ingredients of mammal's milk are water, protein, fat, lactose and ash. Every animal has different percentage of ingredients in their milk as shown in

table III. Here, all the ingredients of mammal's milk are considered as the dimensions (feature or attribute) and mammals are considered as the objects of the data set. First, we obtain the results using standard K-means algorithm. As we wish to obtain four clusters we choose K = 4 and start with first four mammals of the dataset, i.e., Horse, Orangutan, Monkey, and Donkey as the initial centroids of the clusters.

TABLE III
TEST DATA SET

S. NO.	Mammal's	Water	Protein	Fat	Lactose	Ash
1.	HORSE	90.1	2.6	1.0	6.9	0.35
2.	ORANGUTAN	88.5	1.4	3.5	6.0	0.24
3.	MONKEY	88.4	2.2	2.7	6.4	0.18
4.	DONKEY	90.3	1.7	1.4	6.2	0.40
5.	HIPPO	90.4	0.6	4.5	4.4	0.10
6.	CAMEL	87.7	3.5	3.4	4.8	0.71
7.	BISON	86.9	4.8	1.7	5.7	0.90
8.	BUFFALO	82.1	5.9	7.9	4.7	0.78
9.	GUINEA PIG	81.9	7.4	7.2	2.7	0.85
10.	CAT	81.6	10.1	6.3	4.4	0.75
11.	FOX	81.6	6.6	5.9	4.9	0.93
12.	LLAMA	86.5	3.9	3.2	5.6	0.80
13.	MULE	90.0	2.0	1.8	5.5	0.47
14.	PIG	82.8	7.1	5.1	3.7	1.10
15.	ZEBRA	86.2	3.0	4.8	5.3	0.70
16.	SHEEP	82.0	5.6	6.4	4.7	0.91
17.	DOG	76.3	9.3	9.5	3.0	1.20
18.	ELEPHANT	70.7	3.6	17.6	5.6	0.63
19.	RABBIT	71.3	12.3	13.1	1.9	2.30
20.	RAT	72.5	9.2	12.6	3.3	1.40
21.	DEER	65.9	10.4	19.7	2.6	1.40
22.	REINDEER	64.8	10.7	20.3	2.5	1.40
23.	WHALE	64.8	11.1	21.2	1.6	1.70
24.	SEAL	46.4	9.7	42.0	0.0	0.85
25	DOLPHIN	44.9	10.6	34.9	0.9	0.53

It takes 12 iterations for stabilization. The final results (clusters) are shown in table IV. The centroids of the four clusters are as follows: C_1 = (81.88, 7.42, 6.9, 4.01, and 0.93), C_2 = (68.33, 9.55, 17.41, 2.91, and 1.47), C_3 = (45.65, 10.15, 38.45, 0.45, and 0.69), and C_4 = (88.50, 2.57, 2.8, 5.68, and 0.48). The objects contained in these clusters are (O_8, O_9, O_{10}, O_{11}, O_{14}, O_{16}, O_{17}), (O_{18}, O_{19}, O_{20}, O_{21}, O_{22}, O_{23}), (O_{24}, O_{25}), and (O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_{12}, O_{13}, O_{15}) respectively, where O_i denotes the serial number of the object in the data set, e.g., O_9 represents Guinea Pig.

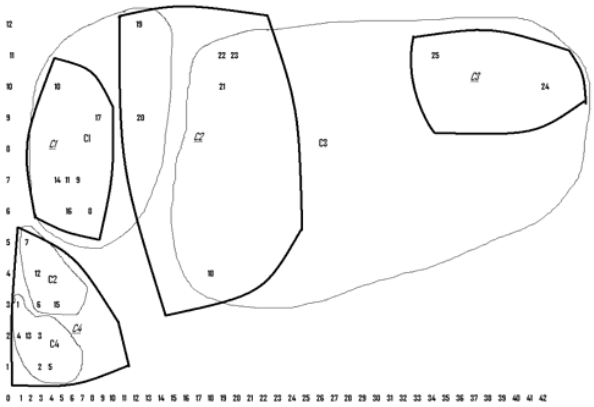


Fig. 3 Clustering of k-means algorithm (thick line) and propose algorithm (thin line)

TABLE IV
CLUSTERS OF DATA SET BY USING K-MEANS ALGORITHM

S. NO.	Mammal's	Water	Protein	Fat	Lactose	Ash	Clusters
1.	HORSE	90.1	2.6	1.0	6.9	0.35	4
2.	ORANGUTAN	88.5	1.4	3.5	6.0	0.24	4
3.	MONKEY	88.4	2.2	2.7	6.4	0.18	4
4.	DONKEY	90.3	1.7	1.4	6.2	0.40	4
5.	HIPPO	90.4	0.6	4.5	4.4	0.10	4
6.	CAMEL	87.7	3.5	3.4	4.8	0.71	4
7.	BISON	86.9	4.8	1.7	5.7	0.90	4
8.	BUFFALO	82.1	5.9	7.9	4.7	0.78	1
9.	GUINEA PIG	81.9	7.4	7.2	2.7	0.85	1
10.	CAT	81.6	10.1	6.3	4.4	0.75	1
11.	FOX	81.6	6.6	5.9	4.9	0.93	1
12.	LLAMA	86.5	3.9	3.2	5.6	0.80	4
13.	MULE	90.0	2.0	1.8	5.5	0.47	4
14.	PIG	82.8	7.1	5.1	3.7	1.10	1
15.	ZEBRA	86.2	3.0	4.8	5.3	0.70	4
16.	SHEEP	82.0	5.6	6.4	4.7	0.91	1
17.	DOG	76.3	9.3	9.5	3.0	1.20	1
18.	ELEPHANT	70.7	3.6	17.6	5.6	0.63	2
19.	RABBIT	71.3	12.3	13.1	1.9	2.30	2
20.	RAT	72.5	9.2	12.6	3.3	1.40	2
21.	DEER	65.9	10.4	19.7	2.6	1.40	2
22.	REINDEER	64.8	10.7	20.3	2.5	1.40	2
23.	WHALE	64.8	11.1	21.2	1.6	1.70	2
24.	SEAL	46.4	9.7	42.0	0.0	0.85	3
25	DOLPHIN	44.9	10.6	34.9	0.9	0.53	3

Afterwards, we solve the problem using the proposed method. Here, we denote the attributes Water, Protein, Fat, Lactose and Ash by a_1, a_2, a_3, a_4 and a_5 respectively for simplicity and ease of use. First, we compute the discernibility matrix which yields following discernibility function:

$$F(A) = (a_1+a_2+a_3+a_4+a_5)\ (a_1+a_2+a_3+a_4)\ (a_2+a_4)\ (a_2+a_3+a_4)\ (a_1+a_3+a_4+a_5)\ (a_1+a_2+a_3)\ (a_1+a_3+a_4)\ (a_2+a_4+a_5)\ (a_1+a_2+a_3+a_5)\ (a_1+a_2+a_5)\ (a_1+a_3)\ (a_3+a_4)\ (a_1+a_2+a_4)\ (a_1+a_2+a_4+a_5)\ (a_2+a_3)\ (a_2+a_4)\ (a_3+a_5)\ (a_2)\ (a_3).$$

Solving this discernibility function we obtain $F(A) = a_2a_3$, i.e., the minimal subset of discernibility function is the attributes a_2a_3 (protein, fat). It renders that water, lactose and ash are irrelevant attributes which are not useful in the clustering of mammal's milk. Hence, only two attributes - protein and fat - are applied in the next steps of the method. Now, it takes six iterations for convergence. The final results (clusters) are shown in table V. The centroids of the four clusters are as follows: C_1 = (8.16, 8.22), C_2 = (3.8, 3.27), C_3 = (9.35, 25.95), and C_4 = (1.75, 2.48). The objects contained in these clusters are (O_8, O_9, O_{10}, O_{11}, O_{14}, O_{16}, O_{17}, O_{19}, O_{20}), (O_6, O_7, O_{12}, O_{15}), (O_{18}, O_{21}, O_{22}, O_{23}, O_{24}, O_{25}), and (O_1, O_2, O_3, O_4, O_5, O_{13}) respectively. As it is difficult to visualize the obtained results in tabular form, the clusters have been depicted in pictorial form in Fig. 3. Here, thick lines indicate the clusters obtained by K-means algorithm and thin lines show the clusters obtained using the proposed method. Our proposed method increases the efficiency of the clustering process by removing the irrelevant attributes from the high dimensional data set and in turn, reducing the number of iterations in the following K-means algorithm.

TABLE V
CLUSTERS OF DATA SET USING PROPOSED ALGORITHM

S. NO.	Mammal's	Protein	Fat	Clusters
1.	HORSE	2.6	1.0	4
2.	ORANGUTAN	1.4	3.5	4
3.	MONKEY	2.2	2.7	4
4.	DONKEY	1.7	1.4	4
5.	HIPPO	0.6	4.5	4
6.	CAMEL	3.5	3.4	2
7.	BISON	4.8	1.7	2
8.	BUFFALO	5.9	7.9	1
9.	GUINEA PIG	7.4	7.2	1
10.	CAT	10.1	6.3	1
11.	FOX	6.6	5.9	1
12.	LLAMA	3.9	3.2	2
13.	MULE	2.0	1.8	4
14.	PIG	7.1	5.1	1
15.	ZEBRA	3.0	4.8	2
16.	SHEEP	5.6	6.4	1
17.	DOG	9.3	9.5	1
18.	ELEPHANT	3.6	17.6	3
19.	RABBIT	12.3	13.1	1
20.	RAT	9.2	12.6	1
21.	DEER	10.4	19.7	3
22.	REINDEER	10.7	20.3	3
23.	WHALE	11.1	21.2	3
24.	SEAL	9.7	42.0	3
25	DOLPHIN	10.6	34.9	3

However, it is not possible to determine the quality of results obtained by both the methods; hence, we use the following quality measures:

Dunn index [23]: It defines the ratio between the minimal intracluster distances to maximal intercluster distance. The index is given by:

$$D = \frac{dmin}{dmax}$$

Here d_{min} denotes the smallest distance between two objects from different clusters, and d_{max} denotes the largest distance of two objects from the same cluster. The Dunn index is limited to the interval [0, 1] and should be maximized.

Davies-Bouldin index [24]: It is defined as:

$$DB = \frac{1}{n}\sum_{i=1,i\neq j}^{n} \max\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

Here, n is the number of clusters, σ_i is the average distance of all patterns in cluster i to their cluster center c_i, σ_j is the average distance of all patterns in cluster j to their cluster center c_j, and $d(c_i, c_j)$ is the distance of cluster centers c_i and c_j. Small values of DB correspond to clusters that are compact, and whose centers are far away from each other. Consequently, the number of clusters that minimizes DB is taken as the optimal number of clusters.

Jagota index [25]: It measures the tightness or homogeneity of the objects within the cluster and is defined as:

$$Q = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\in C_i} d(\ \ , \mu_i)$$

Here, $|C_i|$ is the number of data points in cluster i, k is number of clusters, μ_i is the centroied of i^{th} cluster, x is a point in the cluster and $d(x, \mu_i)$ is the distance between point x and the cluster centroied. Q will be small if (on average) the data points in each cluster are close.

TABLE VI
QUALITATIVE ANALYSIS OF RESULTS

Methods	*Dunn index*	*Davies-Bouldin index*	*Jagota index*
K-Means	0.55	0.225	13.87
Proposed Method	0.50	0.355	14.16

It is evident from table VI that the quality of clusters by both the methods is same as there is no statistical difference in the measure values. The difference in fractional values may be attributed to the computation error in division.

V. CONCLUSION AND FUTURE WORK

In high dimensional data, the general performance of the traditional clustering algorithms decreases. This is partly because the similarity criterion used by these algorithms becomes inadequate in high dimensional space. Another reason is that some dimensions are likely to be irrelevant or contain noisy data, thus hiding a possible clustering. In this paper, we propose a generic framework for efficient clustering of high dimensional data, which is the combination of the concept of rough set theory (reduct) and k-means algorithm. Initially, it finds the low dimensional space in the high dimensional data set by removing the redundant attributes using (reduct) concept of rough set theory. Then k-means algorithm is applied on this low dimensional data (reduct) to find the appropriate clusters. Our experiment on test data set shown that, this framework increases efficiency of clustering process and accuracy of the resultant clustering.

Our future work is to find out the suitable method for determining the initial centroids and the optimum value of k to obtain global optimum clusters in the high- dimensional data set.

REFERENCES

[1] K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering: a review". ACM Computing Surveys (CSUR) 31(3):264-323, 1999.
[2] M. K. Jiawei Han. "Data Mining: Concepts and Techniques", Chapter 8, pages 335-393. Morgan Kaufmann Publishers, 2001.
[3] Carlotta Domeniconi, Dimitris Papadopoulos, Dimitrios Gunopulos, Sheng Ma, "Subspace Clustering of High Dimensional Data", 517-521, SIAM, 1998.
[4] Kun Niu, Shubo Zhang, and Junliang Chen, "Subspace clustering through attribute clustering", Front. Electr. Electron. Eng. China 2008, 3(1): 44–48.
[5] A. K. Jain and R. C. Dubes. "Algorithms for clustering data". Prentice-Hall, Inc., 1988.

[6] M. Zait and H. Messatfa. “A comparative study of clustering methods”. Future Generation Computer Systems, 13(2-3):149-159, November 1997.
[7] J. Ghosh. Handbook of Data Mining, chapter Scalable Clustering Methods for Data Mining. Lawrence Erlbaum Assoc, 2003.
[8] J. Han, M. Kamber, and A. K. H. Tung. “Geographic Data Mining and Knowledge Discovery”, chapter Spatial clustering methods in data mining: A survey, pages 188-217. Taylor and Francis, 2001.
[9] I. H. Witten and E. Frank. Data Mining: Practical Machine Leaning Tools and Techniques with Java Implementations, chapter 6.6, pages 210-228. Morgan Kaufmann, 2000.
[10] C. C. Aggarwal and P. S. Yu. “Finding generalized projected clusters in high dimensional spaces”. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pages 70-81. ACM Press, 2000.
[11] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.
[12] L. ErtÄoz, M. Steinbach, and V. Kumar. “Finding clusters of deferent sizes, shapes, and densities in noisy, high dimensional data”. In Proceedings of the 2003 SIAM International Conference on Data Mining, 2003.
[13] Lance Parsons, Ehtesham Haque, Huan Liu. “Subspace Clustering for High Dimensional Data: A Review”; Supported in part by grants from Prop 301 (No. ECR A601) and CEINT 2004.
[14] Skowron, A., Pawlak, Z., Komorowski, J., & Polkowski, L. “A Rough set perspective on data and knowledge”. Handbook of data mining and knowledge discovery, pp. 134-149, Oxford University Press, 2002.
[15] Anil K. Jain; “Data Clustering: 50 Years Beyond K-Means”; To appear in Pattern Recognition Letters, 2009.
[16] Drineas, P., Frieze, A., Kannan, R., Vempala, S., & Vinay, V. Clustering large graphs via the singular value decomposition. Machine Learning, 56(1-3), 9–33, 1999.
[17] R.C. Belwal, J. Varshney, S. Ahmed Khan, A. Sharma & M. Bhattacharya, “ Hiding Sensitive Association Rules Efficiently By Introducing New Variable Hiding Counter IEEE International Conference on Service, Operations & Logistics and Informatics, proceedings, Vol. I, pp: 130-134, 12th – 15th Oct. Beijing, China 2008.
[18] G. Liu, J. Li, K. Sim, and L. Wong. “Distance based subspace clustering with flexible dimension partitioning”. In Proc. IEEE ICDE, pages 1250-1254, 2007.
[19] Sunita Jahirabadkar, and Parag Kulkarni, ISC – Intelligent Subspace Clustering, A Density based Clustering approach for High Dimensional Dataset, World Academy of Science, Engineering and Technology 55 2009.
[20] Hans-Peter Kriegel, Peer Krger, Matthias Renz, Sebastian Wurst, “A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data ”, In proc. 5th IEEE International Conference of Data Mining (ICDM), Houston, TX, 2005.
[21] McQueen J (1967) some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp 281–297.
[22] http://www.uni-koeln.de/themen/statistik/data/cluster/milk.dat.
[23] Dunn, 1974. Dunn, J. (1974) well separated clusters and optimal fuzzy partitions. Journal of Cybernetics ,4, 95-104.
[24] Davies & Bouldin, 1979. Davies, D.L., Bouldin, D.W., (2000) A cluster separation measure. IEEE Trans. Pattern Anal. Machine Intell., 1(4), 224-227.
[25] Arun Jagota. Novelty detection on a very large number of memories stored in a Hopfield-style network. In Proceedings of the International Joint Conference on Neural Networks (IJCNN’91), 1991.