Membership Set Based K_MEANS Approach for High Dimensional Data

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Information Security

Submitted By
Varun Kumar Sharma
801233027

Under the supervision of:
Ms. Anju Bala
Asst. Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

July 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Membership Set Based *K-Means Approach for High Dimensional Data*"**, in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Ms. Anju Bala* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature:
Varun Kumar Sharma

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Anju Bala
Assistant Professor,
Computer Science and Engineering
Thapar University,
Patiala

Countersigned by
(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgements

I am very much thankful to my guide, Ms. Anju Bala, Assistant. Professor, Computer Science and Engineering Department, Thapar University, who has been very concerned and has aided for all the material essential for the preparation of this thesis report. She has helped me to explore this vast topic in an organized manner and provided me with all the ideas on how to work towards a research-oriented venture.

I would also like to thank Dr. Deepak Garg, Head, CSED for providing me the opportunity in all the research facilities for conducting my thesis work. I would also like to thank Dr. Jhilik Bhattacharya, Assistant Professor, CSED, for providing me her views on the topic. At last but not the least I would like to thank all my friends specially Yogesh Punia (ME-CSE), Ankit Singh (ME-IS) for motivating me, and Sachendra Singh Chauhan (ME-CSE), and Mandeep Singh (ME-IS) for helping me in my implementation work.

Most importantly, I would like to thank my parents and the Almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Varun Kumar Sharma
801233027
ME-Information
Security

Abstract

With the development of Information technology, computers are helping human in every aspect of their life. In recent years the organizations capacity to produce more and more data has been increased. Be it Gmail, Facebook, YouTube, Twitter, Yahoo, blogging or any other social networking website, these are generating tons of data. These large dataset stores high dimensional data. Data mining is data analysis methodologies, which process this large voluminous data, summarize it so that it can be easily understood. Data analysis can be exploratory, or confirmatory. Exploratory data analysis is used when data analyst has no prior model of analysis, and they want to categorize it on the basis of general features. On the other hand in Confirmatory analysis, the classes of data are known and data needs to be classified into these classes.

The objective of clustering process is to find groups of similar items. The similarity is defined on the basis of characteristics, also known as dimensions of data. The characteristic used in clustering algorithms is distance, Sine most of numeric data is available in Euclidean space; which is calculated for similarity purpose. When data is small and have less number of characteristics, human eye can better categorize it, but once data grown in dimensions (more than 3), it is impossible to categorize it by just seeing, and the solution lies in several clustering algorithms. K-means is most used clustering analysis algorithm. It is an iterative approach of point assignment into k clusters. The standard k-means algorithm has many issues with it such as its high time complexity for high dimensional data. Several improvements have been suggested by research community, but when it is applied on high dimensional data, the complexity becomes infeasible because the computation of distance function takes too much time and becomes a bottleneck. Therefore, membership set based k-means approach has been proposed to reduce the computation. It aims to define a cluster membership set for every data point. The distance function is calculated only for the clusters which are contained in this set. With this membership set of cluster, the complexity of overall algorithm is reduced.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Information Industry generates huge amounts of data every day. These large datasets contains several hidden patterns that can be useful for strategic and competitive advantages. This large voluminous data is useless until it is transformed into some useful information. Knowledge Discovery in Databases (KDD) is the process to achieve this task. Data mining is the crucial part of overall KDD process. Data mining is the process of finding hidden pattern and trends from large databases. These hidden patterns are analyzed and evaluated to extract the knowledge for strategic and competitive advantages. There are many algorithms exists for data mining process, among them, Clustering is most used and powerful data analysis technique. Clustering is an unsupervised learning technique. K-means is the most basic and most applied clustering technique to classify the databases. This thesis work concentrates on k-means clustering algorithm.

1.1 Knowledge Discovery in Databases (KDD)

KDD [1] is the process of discovering knowledge for strategic advantages from large voluminous databases. KDD refers to the overall process of useful knowledge extraction from databases. KDD mainly has five steps. Data mining is the main step in overall KDD process. Data mining is the application of specific algorithms for extracting hidden patterns in the databases. These pattern which are found with help of data mining, are then analyzed and interpreted to gain some knowledge. A typical idea of how knowledge discovery is done in the large databases is shown in Figure 1.1[1]. As shown in Figure 1.1, KDD process has several steps:

- Selection of target datasets,
- Pre-processing the data,
- Transformation,
- Data mining, and
- Evaluation.

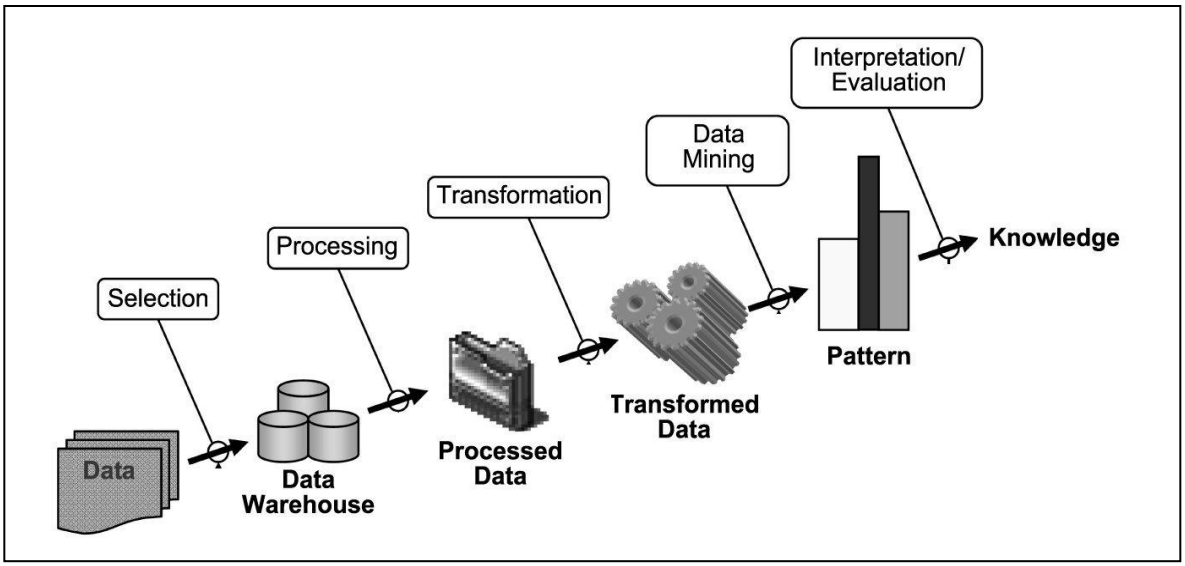

Fig 1.1: Knowledge Discovery in Databases

First step is to select the target application and collect the dataset or a subset of dataset. In the second step, reprocessing of data step, this collected data is refined and the noise or unwanted data is removed from the data. This pre-processed data is transformed into appropriate format in the transformation step. The features that useful are defined and data is defined according to the requirements of data mining. In fourth step, data mining step, the transformed data is analysed to discover hidden features and patterns. In the last step, knowledge is developed by evaluating the mined data. KDD process results in useful knowledge which helps in taking better business decisions [2]. Data mining is a crucial step in KDD process, which finds the hidden patterns in the target dataset. The data mining process only finds the hidden patterns and features of data not the knowledge itself. To gain knowledge from these patterns, the analysis and evaluation is further a step in the KDD process.

1.2 Data Mining

Data mining is a data analysis methodology, which process these large databases automatically, summarize it so that it can be easily understood. Data mining is the process of knowledge discovery from large voluminous databases. Companies generate lots of data

every day. These large databases contain hidden patterns and relationships which are very useful for strategic and competitive importance. Data mining helps in determining these hidden patterns and relationships in databases. These hidden patterns and features when analyzed and interpreted generates knowledge. Data mining is very critical for companies in order to produce strategic information by using their historical data. With the help of data mining tool, large firms can optimize and control their resources, costs and maximize the output. Today, data mining is used in many business applications for many objectives. Business organizations use data mining tool to predict the behaviour of their target customers by analysing the history and pattern of the customers. This way they predict the future behaviour and trend of the customers towards product usage, which helps in gaining the new customers, and retaining the already existing customers.

Organizations manage data warehouses to keep and maintain their data. The large and unmanaged data is pre-processed and stored in data warehouses. The data warehouses work as target dataset for data mining. Data warehouses have some functions which help in continuously retrieving valuable data from business perspective. This collected data is then stored in the data warehouses for further use. Data mining takes this pre-processed data from data warehouses.

1.3 Types of Data Mining Approaches

Data mining algorithms are the collection of techniques in order to perform data mining task. Currently, there are a lot of data mining algorithms for a wide range of data mining tasks. Mainly, these algorithms can be categorized into three groups according to the types of patterns which those algorithms try to discover. These three types of data mining algorithm are presented in the following parts [3].

- Association Rules,
- Classification
- Clustering

1.3.1 Association Rules

Association Rules learning is a method of data mining which is used for un-earthling relations between several items in databases. Association rules are identified by analyzing the

relationships between several data items and their patterns. Usually these relationships are of if-and-than type relationships. This kind of data mining is also called link analysis, because we try to find out the possibilities of two or more events occurring together. This learning is often used in super markets to increase their sale. For example, if a customer is buying cold drinks from a shopping store, he is also likely to purchase chips with that. To increase this possibility, cold drinks and chips can be put together. Association Rules are strong rules to determine the interest of customer towards several products. These strong rules help in decision making in marketing strategies like offer and discount schemes and the placement of the items. These rules are also useful in other applications like web usage patterns, intrusion detection techniques etc.

1.3.2 Classification

Classification is a supervised learning technique, which categorized the data items in already predefined classes. It is used in the case of ladled data. The classes in which the data items are to be classified are known in advance. Classification helps in determining a new data item belongs to which known category. The example of classification can be e-mail filtering system which defines to e-mail category as weather legitimate or spam.

1.3.3 Clustering

While classification is a supervised learning technique of data mining, clustering is considered as unsupervised learning. Clustering is also a process of categorization of given set of points or items into non overlapping groups called clusters such that similar types of items are in the same group and dissimilar types of items in other groups. The difference between classification and clustering is that in clustering the classes in which the objects are to be categorized are not known in advance. The classes or clusters are also determined by the clustering process. For example news clustering, this helps in categorization of news on the basis of their content and presents them in news search result. Also in World Wide Web, the clustering is used for document categorization.

Chapter 2: Literature Survey

2.1 Cluster Analysis

Clustering is a data mining technique which is a pre-processing step of preparing data for further processing [2]. It is a data modelling process of categorization of given set of points or items into non overlapping groups called clusters such that similar types of items are in the same group and dissimilar types of items in other groups [4]. Similarity of points means closeness of points on the bases of some distance measure. Representation of too many objects in fewer clusters necessarily results in loss of certain details, but it helps in achieving simplification. It represents many data objects by fewer clusters, and hence, it is modelling of data by clusters. Although clustering is a classification of data, but the two terms are not same. Clustering is different from classification in the sense that classification is the assignment of all the objects to some predefined classes, whereas in clustering classes in which objects are to categorized are not defined in advance [5]. In clustering algorithm itself clusters classes are defined and each data points are assigned to one of the class.

Clustering is used in many fields and applications like Data mining, pattern recognition, machine learning, market research and image processing [6]. It is also used in the field of biology to group plants and animal taxonomies. With the help of clustering different web documents can be classified for better information discovery. In the field of security, clustering can be used in outlier detection applications to detect fraud customer [7]. The objective of clustering process is to find groups of similar items. Cluster is unsupervised learning procedure of understanding data, which means in clustering process, the data available is unlabeled, and with the help of clustering, it is converted in meaningful labelled clusters. The output of the clustering depends on the number of cluster are being formed. Different number of clusters can be defined for the same data. Figure 2.1 (a) shows a dataset having fifty random data points with 2 clusters in with Figure 2.1 (b), with 4 clusters in Figure 2.1 (c), and with 6 clusters in Figure 2.1 (d).

Clustering is an exploratory data analysis technique, which categorize the dataset into some groups. These groups are formed in a way so that items which have similar features live in same group and those have dissimilar features remain in other. The similarity is defined on

the basis of characteristics, also known as dimensions of data. The characteristic used in clustering algorithms is distance, Sine most of numeric data is available in Euclidean space; Euclidean distance is calculated for similarity purpose. When data is small and have less number of characteristics, human eye can better categorize it, but once data grown in dimensions (more than 3), it is impossible to categorize it by just seeing, and the solution lies in several clustering algorithms. There are many clustering algorithms available. Different kinds of algorithms are best used for different kinds of data.

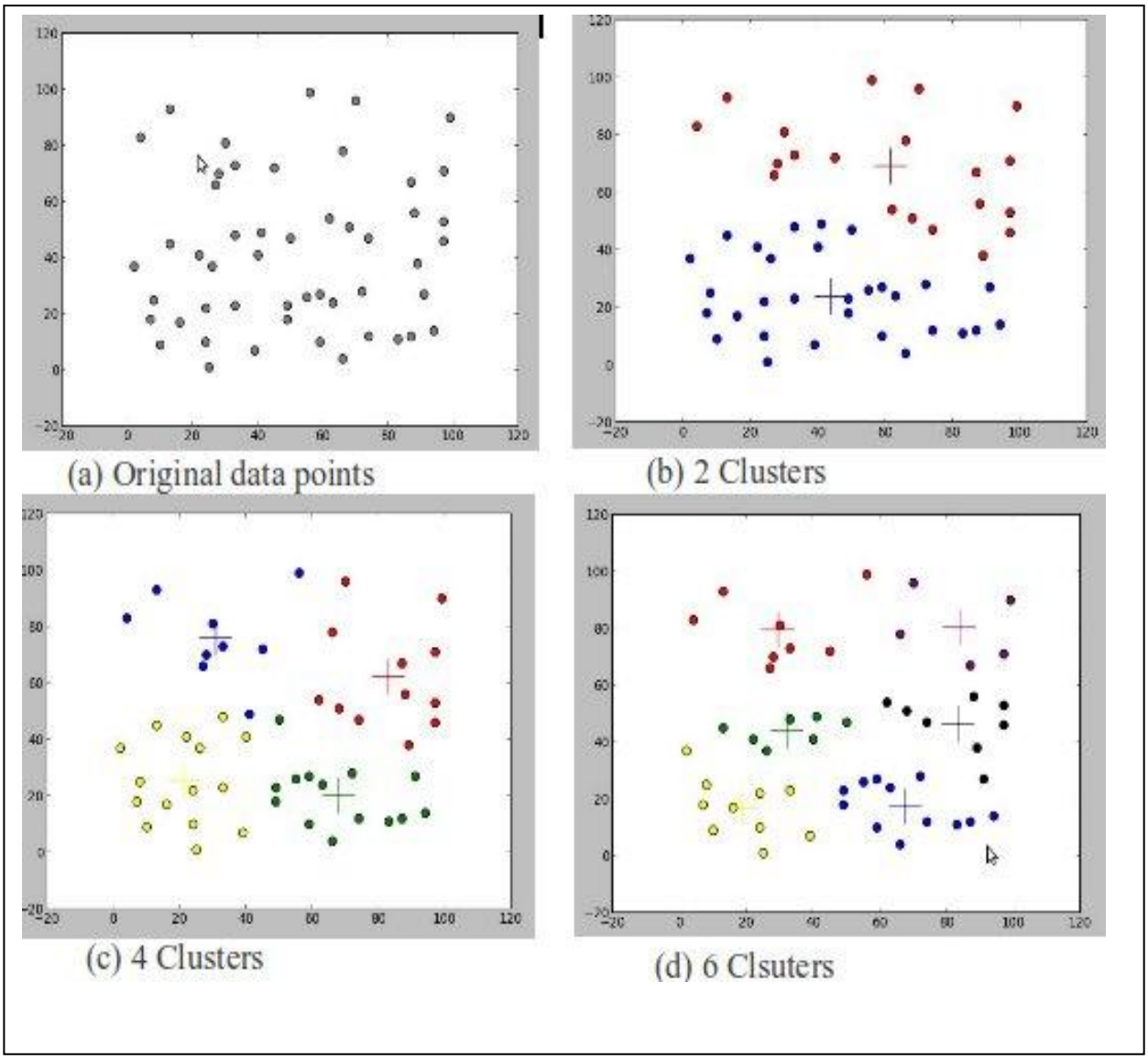

Fig 2.1: Different clustering of a dataset of 50 data points

2.2 Similarity

In clustering, the dataset is categorized into some groups on the basis of the similarity of data points. The clusters are formed in a way that the items which have same features go in same clusters, and the data points having dissimilar features go in different clusters. This similarity can be defined in different ways. Usually it depends on the type dataset we are dealing with. The similarity defined for numerical data points cannot be applied for the textual data points. The general way to define and find similarity of data points is by defining a distance metric between data points. The distance measure chosen depends on the target dataset, but it should always satisfy some conditions. Suppose we have a dataset D having some data points (d_1, d_2, d_3.....), the distance between any two data points (d_i, d_j) is defined as dist(d_i,d_j), than it should satisfy these properties: [8]

- The distances are always non-negative, i. e. distance between any two data points d_i, and d_j, dist(d_i,d_j)>=0
- The distance between a data point di, to itself must always be zero, i. e. dist(d_i,d_i)=0,
- The distance measure chosen should satisfy symmetry feature, means, the distance between two data points di, and dj should be equal to dj and di, i. e. dist(d_i,d_j)=dist(d_j,d_i),
- The distance measure should satisfy the triangular inequality, i. e. for any three data points, d_i, d_j, d_k, dist(d_i,d_k)<=dist(d_i,d_j)+dist(d_j,d_k)

The conditions 1, and 2, ensures that the distance between any two data point is always positive, except when the data point is same, it is considered as zero.

This distance metric (d) is can be chosen differently for different kinds of dataset. The main distance measures which are used are [9]:

- Euclidian Distance
- Manhattan Distance
- Edit Distance
- Jaccard Distance
- Hamming Distance

2.2.1 Euclidian Distance

Euclidian distance is the most basic distance what we could think of. Simply the difference between two points is called as the Euclidian distance. The n-dimensional dataset is one in which every data point has n-features, i. e. n real values associated with it. The Euclidian distance is also called the L_2 Norm. The L_2 norm for any two n-dimensional data points $x(x_1, x_2,....., x_n)$, and $y(y_1, y_2,....., y_n)$ is defined as in Eq. 2.1.

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad \text{(Eq. 2.1)}$$

That is, to say, the distances of respective dimensions are squared, summation is taken for all distances, and then squared root of the summation is calculated.

The usual L_r norm between two data points can be defined as in Eq. 2.2.

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = (\sum_{i=1}^{n} |x_i - y_i|^r)^{1/r} \qquad \text{(Eq. 2.2)}$$

It can be easily verified that all four conditions of distance measures are satisfied. The Euclidian distance between any two data points can never be negative. When calculating distance between same data point, (x_i, x_i) is always zero. The distance between (x, y) and (y, x) is same, because $(x_i-y_i)^2=(y_i-x_i)^2$. Triangular inequality is also satisfied for any L Norm.

2.2.2 Manhattan Distance

Manhattan Distance is also called L_1 norm. It is just the summation of distances in each respective dimension of the two data points. The Manhattan distance is used when travelling between the data points. For The L_1 norm for any two n-dimensional data points $x(x_1, x_2,....., x_n)$, and $y(y_1, y_2,....., y_n)$ is defined as in Eq. 2.3.

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sum_{i=1}^{n} |x_i - y_i| \qquad \text{(Eq. 2.3)}$$

2.2.3 Edit Distance

Edit Distance is used when we have the dataset available is of strings. The Edit distance determines the similarity between two strings by finding out the operations to be performed in conversion of one string to another. The edit distance between two strings $S(s_1,s_2,s_3,....s_m)$, and $T(t_1,t_2,t_3,....t_n)$ is the minimum number of operations required to convert string S into T. Number of operations means is the insertion and deletion.

For example lets S=abcde

And T= adcef

For this, to convert String S to String T, the minimum number of operations required is

- delete b,
- insert d
- delete d,
- insert f

So the Edit distance between S and T, dist(S, T) =4

2.2.4 Hamming Distance

Hamming distance is usually used where the dataset is available in Boolean numbers (0s and 1s). Hamming distance between two Boolean numbers is the differing components between the two. Hamming distance is calculated when both the strings and numbers are of same length. Clearly it is evident that hamming distance satisfies all the distance measure properties. The hamming distance can never be negative; the distance between strings to itself is always zero. The symmetry is satisfied because it does not depend on the order of the strings. The hamming distance the triangle inequality is also met because if difference of components between string a and b is m, and the difference of components between string b and c is n, than the difference of components between a and c cannot be more than Sum(m, n).

2.2.5 Jaccard Distance

Jaccard distance is useful when the data points in considerations are sets. It is defined in the terms of Jaccard similarity. Jaccard similarity between sets A, and B, can be defined as the

ratio of size of intersection of the sets to the size of union of the two sets, i. e. $|A \cap B|/|A \cup B|$. The Jaccard distance between set A and B, is the difference of numeric one and the Jaccard similarity, i. e. (1-SIM(A,B)) [11]. Jaccard distance satisfies all the properties of distance measure. The Jaccard distance can never be negative because the intersection size of two sets can never be more than their union size. The intersection size and union size of a set to itself is always one, so the Jaccard distance of a set to itself is always zero. Also intersection and union are not sensitive to order of the sets, so symmetry is also satisfied. The triangular inequality is also satisfied for the set operations.

2.3 Clustering Algorithm

There are many clustering schemes available, and each of them may give different clusters of the objects. The choice of a particular method depends on the type of input available and type of output desired.

Classification of clustering methods, clustering is mainly divided into four categories:

- Hierarchical clustering methods
- Partition Relocation clustering methods
- Distribution-based clustering methods
- Density-based clustering methods

2.3.1 Hierarchical clustering methods

Hierarchical clustering also called Connectivity based clustering, builds a hierarchy of clusters. These hierarchies are called dendrograms. These hierarchical clusters can be seen as a tree. Every cluster node has children clusters. The siblings of a parent partition their parent. The process of Hierarchical clustering has been shown in Figure 2.2 [10]. Hierarchical clustering methods are further categorized into two categories, agglomerative and divisive [10]. An agglomerative clustering is a bottom-up strategy, it starts treating every point as cluster (singletons) and recursively merges two or more clusters on the basis of similarity. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved. While a divisive clustering, a top-down approach starts with a single cluster containing all objects and recursively splits the more similar items to different clusters is achieved.

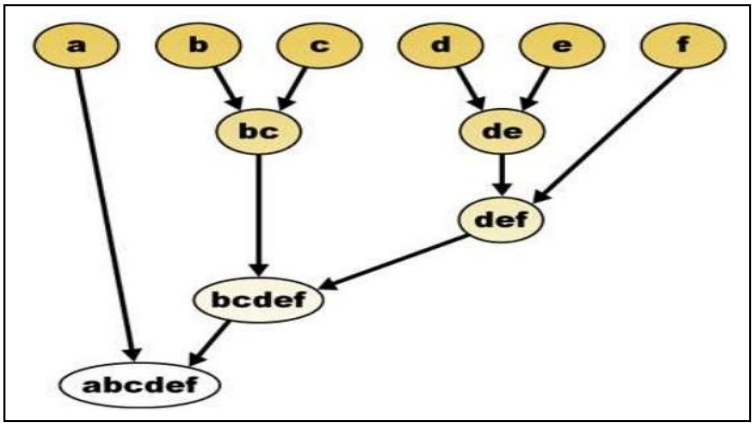

Fig 2.2: Hierarchical Clustering Algorithm

The process continues until a stopping criterion (frequently, the requested number k of clusters). Hierarchical methods are very sensitive to outliers. Outliers are the object or group of objects which do not fit in any of the clusters. These algorithms results in extra clusters. Sometimes these outliers misbalance all the clusters, and algorithm results in less optimized clusters, usually these hierarchical methods are not used directly instead they are used in combination with other clustering schemes.

2.3.2 Partition Relocation clustering methods

Data partitioning algorithms are greedy approaches to find clusters. Partitioning algorithms divide data points into several subsets. To find most appropriate clusters an iterative process is applied. These algorithms results in high quality grouping of data points. For these algorithms, the number of clusters (k) should be known apriori, and is provided as an input to the algorithm. Initially k data points are chosen as the initial clusters with some algorithms. All the data points are assigned to these clusters. These clusters are iteratively refined again and again by reassigning data points and improve the clusters gradually. The process is continued until some optimization is achieved. Figure 2.3 shows the output of partition relocation clustering algorithm with four clusters.

Partitioning algorithms can be categorized into three main categories [11].

- K-medioid clustering,
- K-means clustering

In K-medioid clustering technique, every cluster is represented by the medioid of its data points. The main benefit of this approach is that it is applicable to any type of data.

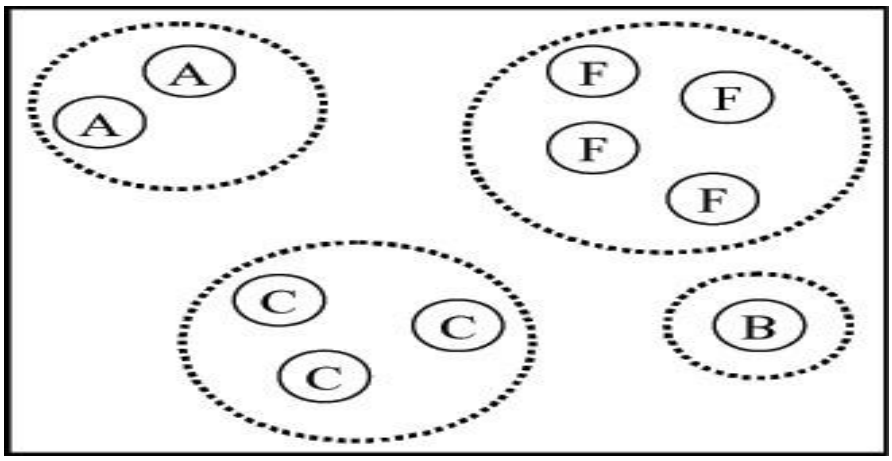

Fig 2.3: Partitioning Clustering

This method is insensitive to `outliers because points at boundary do not affect them [11]. The subset of data points which have proximity to the medioid, are assigned to the respective clusters.

The K-means method of clustering is the most popular clustering algorithm used in real time applications. In this scheme, as name suggests, every cluster is represented by the means of its data points, called centroid of that cluster. This algorithm is not used for categorical data. It performs best when data is numeric and continuous.

2.3.3 Distribution based clustering algorithm

Distribution-based clustering or Probabilistic clustering is based on distribution models [13]. In Probabilistic clustering, it is assumed that the data is taken from different populations whose distributions is to be determined. For each distribution scheme mean and variance are calculated. Usually a cluster is associated by the distribution mean. The area surrounded by the mean a distribution forms a new cluster for that distribution. A data point x is assigned to

a particular cluster on the basis of the probability of that point being in cluster.

2.3.4 Density-Based Clustering Algorithm

Density based clustering scheme is a partitioning based method of clustering [5]. The clusters in this partitioning scheme are formed on the bases of density. The data points are considered to be in Euclidean space. Then in the space by the positions of the points it is determined that how the data points can create a denser group. The points are connected to others on the bases of proximity. This problem is similar to find a connected component. The more connected components are taken as cluster. Since an area in the space can be in any shape and in any direction, so this algorithm results in clusters of any arbitrary pattern. The algorithm is insensitive to outliers.

2.4 K-Means Clustering

K-means is a partitioning relocation based clustering [14]. It is the most famous technique of data clustering. The process of clustering breaks up the dataset in some subsets. Each of the subset is represented by a cluster representative, or cluster centre, which is usually the mean of all the data points in that cluster. The objective is to find as tight clusters as possible. To achieve this, for every cluster, an Error function which is defined as sum of the squares of the distances of all the data points in that cluster is minimized. The problem is NP-Hard problem because to solve it every possible combinations of k subset should be checked for least minimization of Error function.

The greedy algorithm K-means tries to find out the solution to the problem for a fixed cluster size k. This algorithm is usually applied when data is continuous and is available in Euclidean space. A Euclidean Space consists of n-dimensions. Every direction in the space is represented by a vector. Every point in Euclidean space has n real vectors, and each vector is represented by a unique vector. Suppose D is the input dataset to be clustered has m records, and each record has n attributes. Than to consider this dataset D as in Euclidean space, each attribute is mapped to a particular direction. Let there is an n-dimensional Euclidean space. Each point x in this space is represented by n real number vectors. A data point in this Euclidean space is represented by (d_{i1}, d_{i2}, d_{i3}.....d_{in}).

The Dataset D and a number k are provided as input to the k-means clustering algorithm. In K-means method the number k specifies the number of clusters to be formed. The representative of a cluster is the *mean* (average) of all data points in that cluster. The number of cluster (k) is given as an input to the algorithm. The algorithm follows an iterative approach to assign the data points in the clusters and calculation of centroid until some optimum condition is achieved. The similarity of the data points with the centroid is measured by Euclidean distance. Let there are two data points x(d_{x1}, d_{x2}, d_{x3}.....d_{xn}) and y(d_{y1}, d_{y2}, d_{y3}.....d_{yn}) in this space. Than the distance between these two points is given by the formula in Eq. 2.4.

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad \text{(Eq. 2.4)}$$

First, the algorithm selects any k data points from the all available data points by some mechanism. The basic approach to select initial cluster centers is random. Once initial cluster centers, the next step is to assign every data to one of each cluster. The assignment of data point is done to the cluster, if its cluster centre has the minimum distance from the data point. Mean i.e. Average is calculated of all the clusters and the cluster centre is updated with its new calculated mean with the help of the formula given in Eq. 2.5.

$$C_{new} = (1/m) \sum_{x_i \in C_j} (C_{old} - Xi)^2 \qquad \text{(Eq. 2.5)}$$

Where m is the number of data items in the respective cluster. This process of data point assignment to new clusters and calculated new centroids of the clusters is repeated until data points are assigned same clusters for consecutive two steps. It is checked if the means of consecutive loops are same. Instead of matching exact means, some threshold can also be taken, which will define some minimum limit to stop. The minimizing function is called the

error function. Error function E is defined as in Eq. 2.6.

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} \| x_i - v_j \|^2 \quad \text{for } i = 1, \ldots, n; j = 1, \ldots, k. \qquad \text{(Eq. 2.6)}$$

The process of k-means algorithm has been shown in Figure 2.4, step by step in the form of flow chart. The input to the algorithm is the initial cluster centers. The algorithm has usually three steps. In step 1, initialization of cluster centers, in step 2, assignment of data items into cluster centers, and in third step the new cluster centers are calculated by taking average of its data items. Step 2 and 3 are repeated several times until there is a change in cluster centers.

The initialization of cluster centers is done either by random choice or by some other specific cluster algorithm like hierarchical algorithm. After initialization of cluster centers, the data points of the datasets are assigned to its nearest cluster. For this the distance of every data point is measured from each cluster and the data point is assigned to the cluster which has least distance from the data point. In next step, new cluster center is determined. For this the average data point is chosen for each cluster and it is defined as the new cluster center. Now these clusters are validated against the stopping criteria. If satisfied than we stop the algorithm, otherwise we go back to the data point data point assignment step. The Algorithm 2.1 shows the formal k-means algorithm.

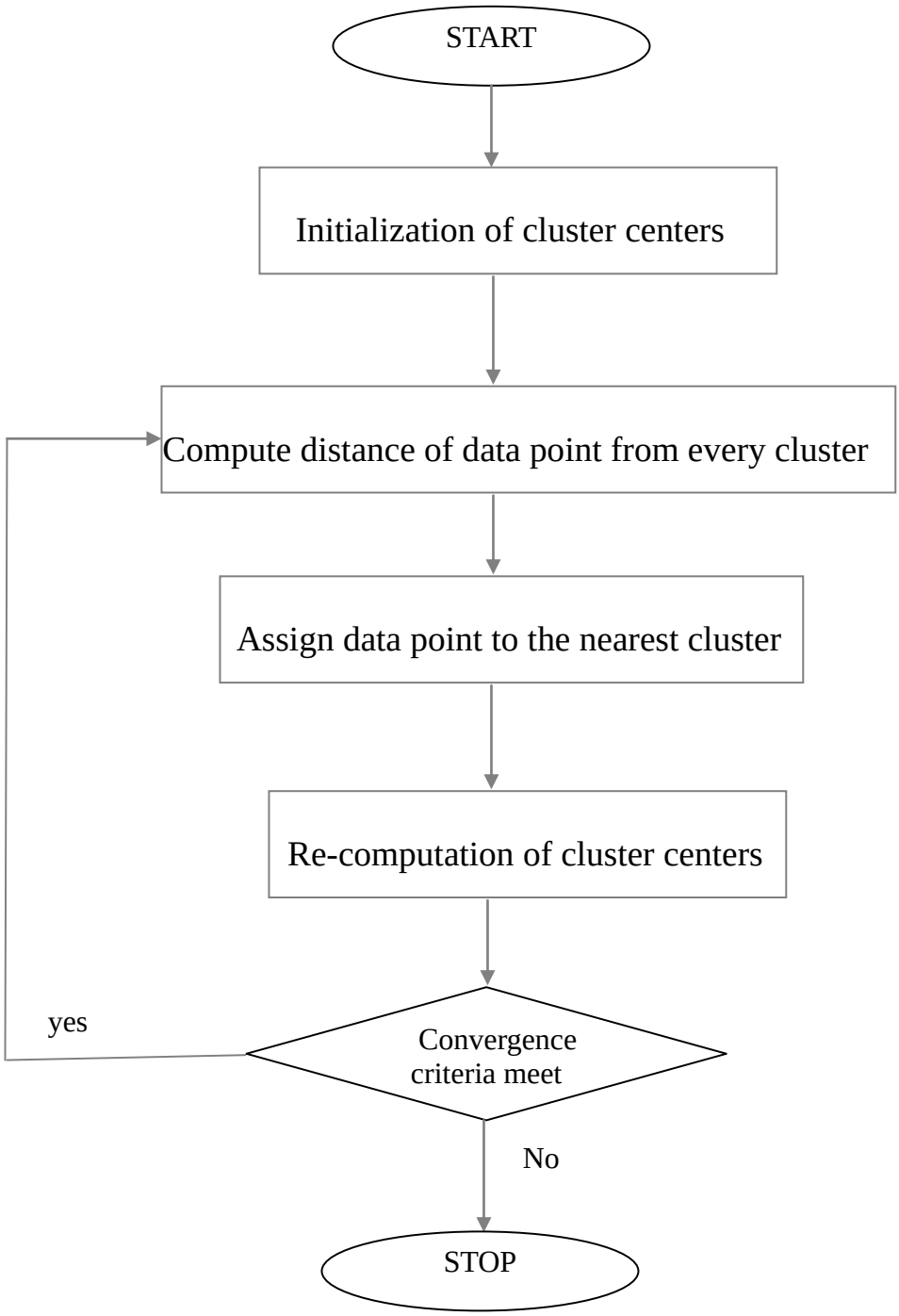

Fig 2.4: Standard K-means Approach

Algorithm 2.1 Standard K-means Algorithm

Input: The Dataset (D), Initial Centroid set(C), number of clusters k

Output: C

Procedure:

1: WHILE any $C_i(1<j<k)$ change location DO
2: FOR D_i {1<i<n} do
3: CLUSTER $(D_i) \leftarrow \min j \, \|D_i - C_j\|$
4: END FOR
5: FOR D_j (1<j<k) DO
6: $C_j \leftarrow \Sigma_i \, I(\text{CLUSTER}(D_i) = j) * D_i / I(\text{CLUSTER}(D_i) = j)$
7: END FOR
8: END WHILE
9: RETURN C

Symbol used:

$D(x_1, x_2, x_3, ..., x_n)$	Dataset
n	number of data points in dataset
x_i	i^{th} data point
$C(c_1, c_2, ..., c_k)$	Set of cluster
Cj	j^{th} Clusters
$d(x_i, C_j)$	Distance of x_i data point from C_j cluster

If the loop in this algorithm is repeated t times before the stopping criteria meets, The complexity of this algorithm comes to be O(ktn). The number of loops taken to converge the error function totally depends on the chosen initial cluster centres. Moreover, initial cluster centers also affect the overall clusters. Different initial cluster centres results in different final clusters. The time when running algorithm also depends on the distance function run k times in every loop. Once the number of dimensions, or better say attributes, go high the distance function becomes more complex, and overall time gets increased. For large datasets having high dimensional data points, this complexity is too high, and not acceptable.

K-means algorithms very useful algorithm, and is widely used in many applications for business intelligence purpose. However the basic k-means algorithm gives high time

complexity for real big data. K-means is a widely research topic for research community and there have been many improvements for the k-means algorithm to faster and improve its performance and accuracy. Some improvements can be given here.

In [15], Vance Faber, has proposed a new approach, called "continuous k-means algorithm", in which the initialization and data point re-assignment procedures have been modified, and done with the help of random sampling method. Cluster centers are not selected at random as in standard algorithm rather, selected as random samples from the available population of dataset in a way that if there is a dense area in the original dataset, than it should also be reflected in the selected random sample. The data point's re-assignment step is also refined with the help of random sampling, instead of that, in every step for every cluster only a random sample of its data points are selected if they need to be reassigned. This approach gives batter results and converges after checking only a little part of the large input dataset.

In [16], Hung et al., presented a new approach of handling very large dataset for clustering by applied a grid based technique. The area surrounded by the all of the data points in input dataset is divided into some equal size of grids. Only non-empty grids are chosen for consideration i.e. which has at least one data point in its area. Representatives are chosen from these grids, and considered to be the dataset for the k-means algorithm. Representatives of these grids are the medioids of the respective grids. The clustering algorithm is run for these representatives, considering as data points. For initialization step, k random representatives are chosen from this new dataset. Clusters are generated from this new dataset, the result of the algorithm happens to be the final cluster result. All the data points contained in one grid are represented by its representative, and move along. The number of grids has been defined empirically for this approach. For the grid which is on boundary of two clusters, separate mechanism is followed and its every data point is individually checked for each of neighbouring clusters. The algorithm is very efficient for very large dataset.

In [17], Malay K. Pakhira, has done a work on the algorithm to avoid empty clusters in the k-mean algorithm, but at the cost of more iteration. The idea is to modify the computation of new cluster centres. In this work, to mitigate the effect of empty clusters, apart from including all the members of that clusters, also include the present cluster means to calculate new cluster mean. This way if in any step the cluster centers happen to be equal

than in the next step, this inclusion of present cluster mean in new centre computation will definitely result in different cluster centers. It delays the convergence condition by adding extra iterations. In paper it has been shown that for large dataset, this overhead of iterations is relatively very small. However the solution does not work if all the cluster centers initially assigned to the median of the dataset.

In [18], Jieming Wu and Wenhu Yu have introduced a pre-processing of input data so as to reduce the number of iterations in the clustering algorithm. The idea is to first filter out those k data points which are most isolated data pints. When selecting initial cluster centers, if these isolated points get selected, the overall computation would be increased, can also distort the shape of clusters. Filtering out k most distant data items will assure that these items will not be selected as initial cluster centers. Leaving these isolated k data points, remaining dataset is considered as the set for initialization step. It has been shown that, Performing this pre-processing steps, the initial cluster centers are chosen to be closer to the final cluster centers.

In [19], Shehroz S. Khan and Amir Ahmad, an algorithm "Cluster centre initialization algorithm for K-means clustering (CCIA)" has been proposed. It aims at computing initial cluster centers for each attribute in the data set, rather than for data points. This way outliers are not getting chosen and initial cluster centers results closer to accurate cluster centers. To achieve this, for dataset is presumed to be normal distributed. The normal curve represented by this attribute for all data point is drawn and parted into k equal parts. Now k-means algorithm is run for this attribute. The output results in k vectors for that attributes, these vectors serves as initial cluster centers for its respective attribute in main k-means algorithm. These outliers do not get chosen as initial cluster centers.

In [20], Rupali Vij and Suresh Kumar, have proposed some modification to the standard k-means algorithm in point re-assignment steps. They have used the fact that in two successive steps there are many points which remain unchanged and are not re-assigned. When assigning a point to a cluster its distance from its cluster is computed and stored. In next iteration if the distance from the new cluster center for that data point is found to be less, the point assignment remain unchanged, entire process for assignment is not run, which results in reduction of computation.

In [21], K. A. Abdul Nazeer and M. P. Sebastian, Have proposed a different cluster centre initialization scheme so as select centers from more densed area and to avoid outliers. In the proposed scheme, iteratively k sets are generated, each containing closets (3/4) of (n/k) items. It is done so that each set can have equal number of items and no item remains at last. Now means of these subsets are calculated which work as seed for initialization step of the algorithm. For point assignment step same procedure as defined in [7] has been followed.

In [22], Kohei Arai and Ali Ridho Barakbah, have used hierarchical clustering algorithm for the initialization step. Standard k-means algorithm result is affected by the choice of initial cluster centers. In the standard algorithm the initial cluster centers are chosen as random data point, which gives different result when run. In this work, the algorithm is run several times(much larger than the number of clusters required) by applying random initial centers, thus collected final cluster centers are reduced to k numbers by the agglomerative hierarchical mean approach. This results in closer final cluster centers.

Chapter 3: Problem Statement

3.1 Gaps in Research

The k-means clustering is the most basic and oldest clustering algorithm. There have been several improvements to the K-means clustering algorithm by the research communities. However, there are still many problems which are faced in the implementation of k-means algorithms.

There are some issues with this algorithm, which can be listed as above

- Initialization of cluster centers [14],
- Sensitive to outliers [24],
- Fixing the size k [13],
- Final clusters may remain empty, that is no data point would be assigned to it [25],
- The time complexity is too high when dataset is very large, since it depends on both, the number of items and the number of clusters [16].

3.1.1 Initialization of cluster centers

The algorithm is sensitive to initial clusters. Different selection of initial cluster centers gives different performance results. The points should be chosen in a way that the distance between the clusters can be maximized. The random scheme of selection gives poor performance.

An approach to select initial cluster centre is first to apply k-means on different small sample data points selected randomly, these resulting clusters are than used as an initialization vectors for union of all these selected data samples [14]. Best cluster centers selected this way results in very good choice as initialization for k-means algorithm on complete dataset. Other clustering algorithms can be used to decide initial cluster centres like hierarchical clustering. In hierarchical clustering, agglomerative approach is followed to find k initial cluster centers. Every data point is considered as a cluster. The iterative approach is followed to merge two clusters which have minimum distance between them. The iterations are repeated until only k-cluster remains. The scheme is feasible only for small dataset, but for large dataset, the scheme results in too much computation, and so cannot be applied. A systematic approach can also be chosen so as to maximize the distance between centres, for

this first any data point is chosen randomly and take the farthest data point as another cluster centre [13]. The next cluster centre is one which is farthest from both of the previous chosen centres. These way k-initial clusters can be found out. But the process takes many comparisons, and still the performance depends on the first chosen centre.

3.1.2 Sensitive to outliers.

The other problem with the k-means algorithm is sensitiveness with outliers. The data points which are very far from other data points are known as outliers or isolated points. The algorithm performance gets affected because of these distant points, in terms of both, time complexity, and the final results. The outliers will change the mean, i. e. cluster centre, and so can affect the overall cluster output, and the data point might go in wrong clusters [24]. The good idea is not to assign them as initial cluster centers. There are different mechanism to deal with outliers, one, they can be left from clustering, two, these can be collected in a different cluster, or three, these can be assigned to nearest available cluster. Figure 3.1 shows the data points in the dataset which are outliers. In the Figure 3.1, the outliers are assigned to different clusters.

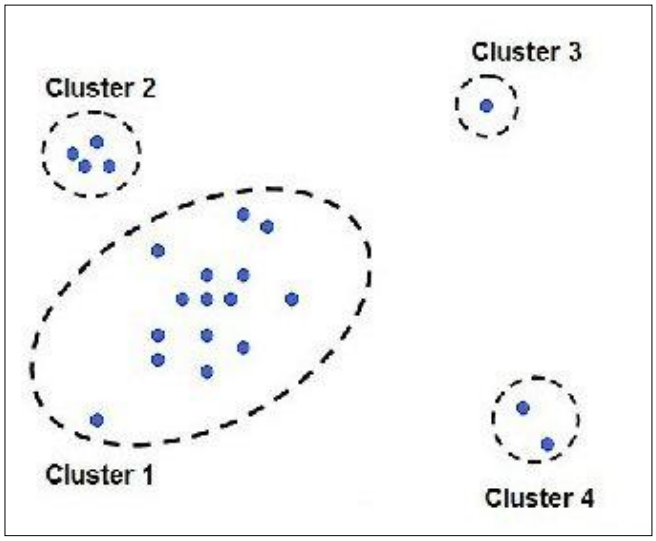

Fig: 3.1 Problems of Outliers in k-means algorithm

3.1.3 Fixing the size k

Another issue with k-means algorithm is selecting the correct number of cluster i. e. K-value. The usual way to choose the value of exact k is trying the clusters for different values of k, and checking for cluster density and intra cluster distance. This process takes long time and depends on user intuition itself. If the number of generators of data points is known in advance than it could be the cluster size. If the data points can be categorized on the basis of some classes, like in the case of grading of the students, or location of different customers, than these classes can be used as the number of clusters. In statistics and numerical data, objects can be categorized on the basis of number of different distribution functions. Some other clustering algorithms are also used to find the appropriate value of k, like hierarchical or density based clustering methods [13]. Once clustering result is given by the algorithm, it can be validated visually. In, Euclidean space, all the data can be drawn on a graph and visual idea about the shape and the number of clusters can be taken. This way, the output of the clustering algorithm can be verified up to a little extent.

3.1.4 Problem of Empty clusters

Empty cluster results totally depend on initialization steps of the algorithm. If the choice of initial cluster centres is bad, it might generate empty clusters [25]. The problem occurs when initial cluster centres are either having same value, or they are too close. In that case the data points, near to the close centres are assigned only to one cluster, and leave the other one empty. The general approach to deal with the problem is running the algorithm until there is some acceptable number of items in each cluster. A re-initialization is also an option if previous choice of initialization results in any empty cluster.

3.1.5 Complexity for High Dimensional Data

In standard algorithm in each iteration, each data point is checked for its membership in every cluster. In each iteration, distance function also runs for each data point and for every cluster, which results in huge calculation. For small dataset, algorithm runs fine, but for large dataset, the algorithm results in high time complexity [16]. If the data is high dimensional, applying standard k-means algorithm becomes infeasible for the data set. Much work has been done to reduce its complexity, but still it is an open issue for big data.

3.2 Research Question

K-means clustering algorithm is very useful in several algorithms to find the hidden pattern of the input data or a part of data. This algorithm is very useful and effective if data to be clustered is less featured. But when data has many features i. e. the data is high dimensional, the algorithm becomes complex in terms of the time and calculation. This thesis work aims to reduce the time complexity consumed by the K-means algorithm for high dimensional data.

Chapter 4: Membership Set Based Proposed K-means algorithm

Most of the existing work has been done to improve the performance of well known clustering algorithm K-means. In this work the complexity of the standard and existing improvements of the algorithm has been refined. There are three main steps of k-means algorithm

- Initialization of initial centroid
- Data points assignment, and
- Checking stopping criteria.

In the k-means algorithm, initialization step is performed just once while point assignment and stopping criteria run in every loop. The second step i.e. Data items assignment is the most time consuming step in the algorithm. For every loop it runs for all data items. Moreover distance calculation is also performed in the step. By applying some modification to the data points assignment step, the overall taken time can be reduced for the algorithm. In standard algorithm, every time, for every data item, its distance from each cluster centre is calculated and minimum of all distances is drawn, and the data item is assigned to the respective cluster set. It means that for every loop distance function runs k times for every data item. It results in redundant calculations. It is because in two successive loops, it is very common that most of the data items are re-assigned to the same clusters. This fact can be used to reduce the computation.

To achieve this reduction in computation, for every data point calculate its distance from its respective cluster's mean. Whenever the new centroids are calculated, also find the difference between the old and new cluster mean. Now when re-assigning the points, see if the its distance from new cluster mean is less than or equal to the difference of the two means (old and new means), if true, the data point stays in the same cluster. Also, after some loops, it can be detected that in what clusters the data item will certainly not be assigned. There is no need to do calculation for these clusters. The idea is to define a membership set of clusters for each data point. This can be determined by setting some number to decide if a cluster is valid for consideration. If the distance from the data point is increasing for some certain

number of iterations, than that cluster should not be considered as a hope for that data point, and must be removed from the membership set. This membership set scheme is very effective when dataset is very large (consists of thousands of records), and the number of clusters to be built is also significantly large. For the initialization step, initial cluster centers are chosen in a way so that the difference between the clusters i. e. Inter-cluster distance can be maximized. For this, in this work, a scheme has been chosen which has been proposed by in [23]. In this scheme, data items are sorted by their position in the Euclidean space. For this the distance from origin point is computed. This sorted dataset is split up in k continuous equal groups. Now these k groups are treated as k initial clusters with their respective medioids as the cluster centers. The step by step approach for the proposed algorithm has been shown in Figure 4.1.

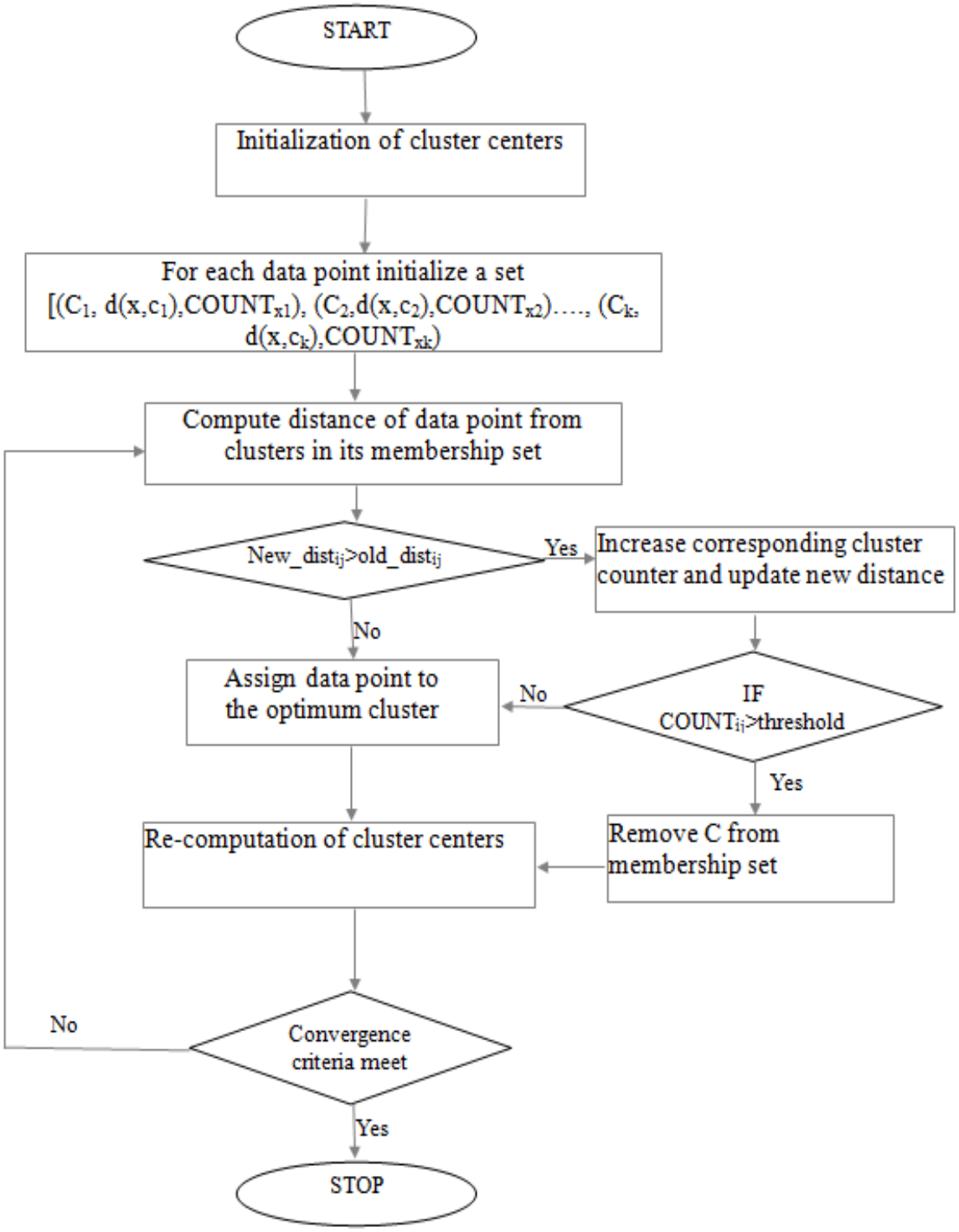

Fig 4.1: MEMBERSHIP-K-MEANS approach

Symbols used:

$D(x_1, x_2, x_3, ..., x_n)$	Dataset
n	number of data points in dataset
x_i	i^{th} data point
Cj	j^{th} Clusters
$d(x_i, C_j)$	Distance of x_i data point from C_j cluster
old_dist_{ij}	Previous distance of i^{th} data point from j^{th} cluster
new_dist_{ij}	Current distance of i^{th} data point from j^{th} cluster
$COUNT_{ij}$	Counter for i^{th} data point and j^{th} cluster
THRESHOLD	Constant threshold value

The explanation of the flowchart can be given as:

- In step 1, the cluster centers are initialized. For initialization purpose the scheme proposed in [23] has been implemented.
- In step 2, for each data point a membership set has been defined which consists of three tuples- cluster, its distance from the data point and a counter initialized with zero. The cardinality of the set is same as the number of clusters.
- In step 3, is data point assignment step. In this step, the data point assignment is done only for those clusters which are in its membership sets.
- In step 4, when calculating distance from each cluster, the new distance (new_d) is compared with the old distance (old_dist). If it is greater, than the new_dist is replaced with the old_dist in the membership set, and the counter is increased by one. If the counter is equal to threshold than the hope for being the data point in that cluster is no more, so we remove it from the membership set.
- In step 5, if the new_dist is less than the old_dist, than the data point is assigned to that cluster.
- In step 6, the new cluster centers are determined for each cluster by taking the average.
- In step 7, the clusters are checked for the stopping criteria. If the criteria does not meet, than we again go back to the step 3, i. e. data point assignment step.

The MEMBESHIP-K-MEANS algorithm has been shown in Algorithm 4.1.

Algorithms 4.1 MEMBERSHIP_K-MEANS

Input: The Dataset (D), number of clusters k, Initial Centroid set(C), a number threshold

Output: C

Procedure

1: FOR i(1<=I <= n) DO

2: DEFINE MSi←{(C_1,d(x,c_1),$COUNT_{i1}$=0),(C_2,d(x,c_2),$COUNT_{i2}$=0) ,.....
.....C_k,d(x,c_k), $COUNT_{ik}$=0)}

3: WHILE any C_j(1<j<k) change location DO

4: FOR D_i {1<i<n} DO

5: CLUSTER (D_i) ← min $||D_i - C_j||$

6: END FOR

7: FOR i (1...n) DO

8: FOR each element in MSij

9: NEW_DISTij←|D_i-MS_{ij}|

10: CLUSTETR (D_i) ←min (NEW_DIST_{ij})

11: CALL RE-SET_SET ($DIST_{ij}$, NEW_DIST_{ij}, $COUNT_i$)

12: END FOR

13: END FOR

14: FOR Dj(<j<k) DO

15: C_j ←Σ_i I(CLUSTER (D_i) = j)* D_i /I(CLUSTER(D_i) = j)

16: END FOR

17: END WHILE

18: RETURN C

Procedure RE-SET_SET ($DIST_{ij}$, NEW_DIST_{ij}, $COUNT_i$)

1: FOR each element in MS_i DO

2: IF (NEW_DIST_{ij}>=$DIST_{ij}$)

3: $COUNT_i$←$COUNT_i$+1

4: $DIST_{ij}$← NEW_DIST_{ij}

5: AND IF ($COUNT_i$=threshold)

6: Remove jth cluster from MS_{ij}

7: ELSE

8: $COUNT_i \leftarrow 0$

9: END FOR

The Algorithm 4.1 shows the overall procedure of the proposed algorithm. The algorithm first, takes input as dataset D, the initial choice of centroids, the number of clusters to be formed (k), and a numerical threshold value. The dataset D has n data points, each data point having m dimensions. The input C, i. e. initial choice of centroid set C is chosen by finding the distances of the data points with the origin, and sorting the data points according to the distance calculated. Now this set can be divided into k parts, and the median data point median of each part can be taken as the initial choice of centroids. The input threshold is chosen according to the data points available. How close the data points are determines the value of threshold. Threshold taken depends on the number clusters to be created and the data points into consideration. It has been observed that as the number of clusters increase, the threshold decrease. In the algorithm a membership set for each data point is created initially having all the cluster centroids as its element, along with it's distance with the data point, and a counter initialized with zero. The data points are assigned to one of the cluster according to the least distance criteria. In next step the new centroid of each cluster is chosen by taking the median of all data pints in the respective cluster. Next, when assigning the elements to the clusters, first the new distance from the clusters centroid is calculated and compared with the old distance which is in the set. If the distance comes out to be more, than the counter with regard to that centroid is increased. The idea to have counter is that to keep track how many times the data point is continuously far from that cluster. This counter value is compared with the threshold, if it is equal to the threshold. Than this cluster centroid is removed from the membership set of that cluster. This overall procedure has been shown in the PROCEDURE RE-SET_SET. In next loop, the assignment of that data point is considered only for the remaining cluster centroids. This way the calculation gets reduced in each next loop.

Chapter 5: Experimental Results

The proposed algorithm MEMBERSHIP-K-MEANS has been tested for multiple real datasets. The result of the outcomes has been compared with the traditional k-means algorithm, and the enhanced method proposed by K. A. Abdul Nazeer [21]. The analysis of algorithm has been done in terms of number of distance function execution, elapsed time and the accuracy and quality of the clusters.

5.1 About the Dataset [26]

The dataset on which the experiment has done are

- Ruspindi (2-dimensional),
- Iris(4-dimensional),
- Environmental(4-dimensional),
- Production (8-dimensional).

5.1.1 Ruspindi

Ruspindi is a dataset of 75 data points which are in 2 dimensions. These data points are consist of four categories. This data set is widely used for illustrating clustering methods and algorithms.

5.1.2 Iris

The dataset Iris, is a dataset of flowers species contain 150 data points, each having 4-dimensions. Each data points contain the 4 variables, sepal length, sepal width, pedal length, and pedal width. All measurements are in centimetres. The dataset is divided into three species, setosa, versicolor, and virginica.

5.1.3 Environmental

The Environmental dataset is the measurement environment factors, average ozone concentration hourly, average wind speed between (7:00 AM to 11:00 AM), maximum temperature of the day measured in Fahrenheit, and solar radiation in the city Langley

between (8:00 AM to 12:00 PM) of the new York city from May 1973 to September 1973. The dataset consist of 111 records. There are three groups found in the dataset.

5.1.2 Production

Production dataset is a dataset of 816 observations of US States productions. Every record is 8-dimensional having the pcap as private capital stock, hwy (highway and streets), water (water and sewage facilities), until (public buildings), pc (public capital), gsp (gross state products), emp (employment) and the unemp (unemployment in the states).

5.2 Test and Results

Three implementations have been made for k-means algorithm, the basic standard k-means algorithm, the one alternative enhanced method proposed by K. A. Abdul Nazeer [21], and the one proposed in this thesis, MEMBERSHIP-K-MEANS. Each of four datasets is clustered by all three methods and the results have been studied. Figure 5.1 shows the original data points in the Ruspindi dataset.

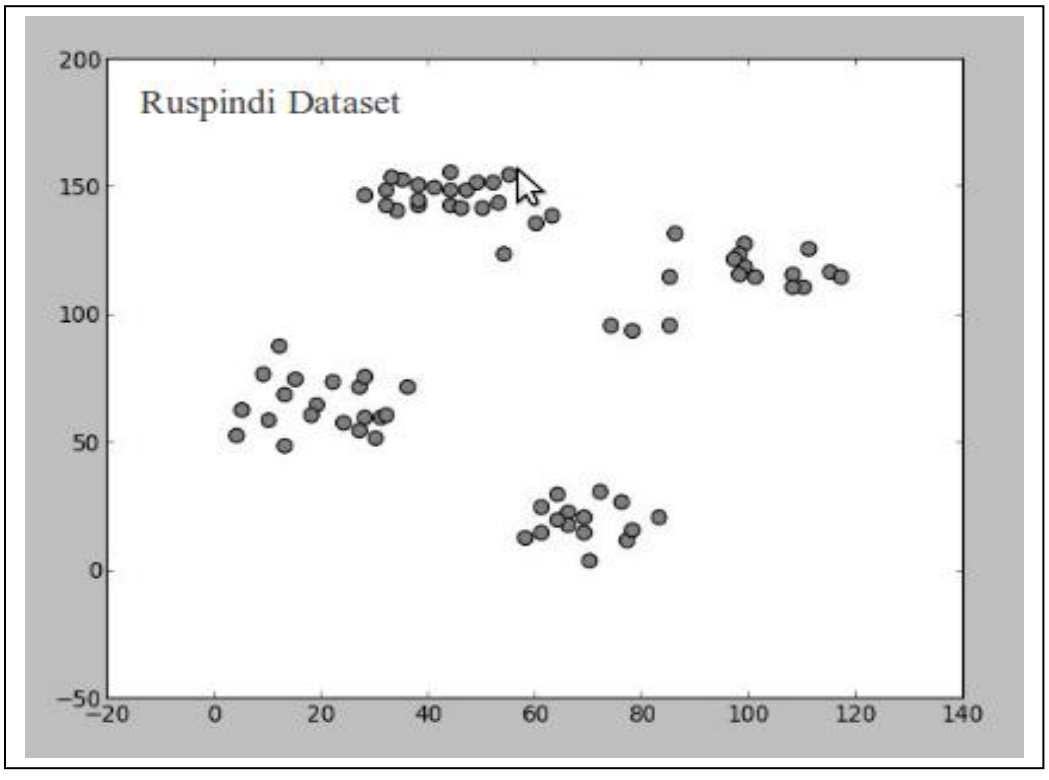

Fig 5.1: Original Ruspindi Dataset

Figure 5.1 shows all the 75 data points. The algorithms are run for the dataset Ruspindi and the result is shown in Figure 5.2. The figure shows the dataset with three clusters. The resulting 4 clusters have 23, 17, 15 and 20 data points respectively. The threshold value given as the input to the MEMBERSHIP-K-MEANS algorithm is 3.

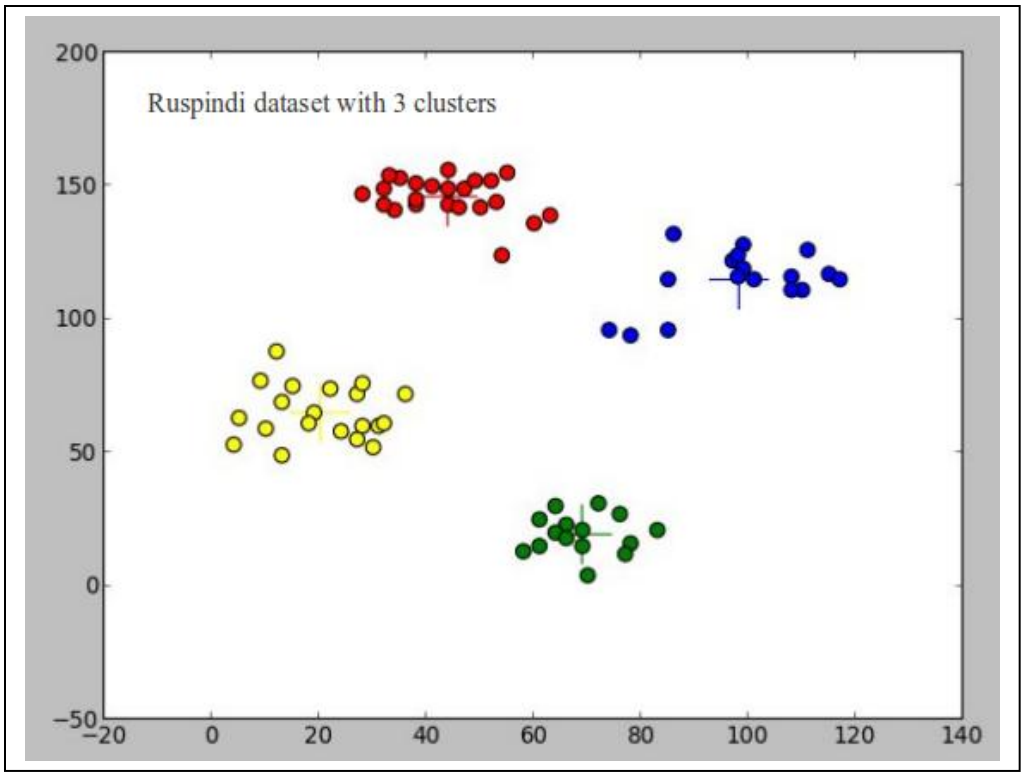

Fig 5.2: Ruspindi dataset with 3 clusters

The original Iris dataset having 150 data points has been shown in Figure 5.3. The algorithm is run for the iris dataset with input as k=3, i. e. for 3 clusters. The output of the algorithm has been shown in Figure 5.4. The threshold value given as input is 2. The algorithm produces 3 clusters having 43, 34 and 34 data points respectively.

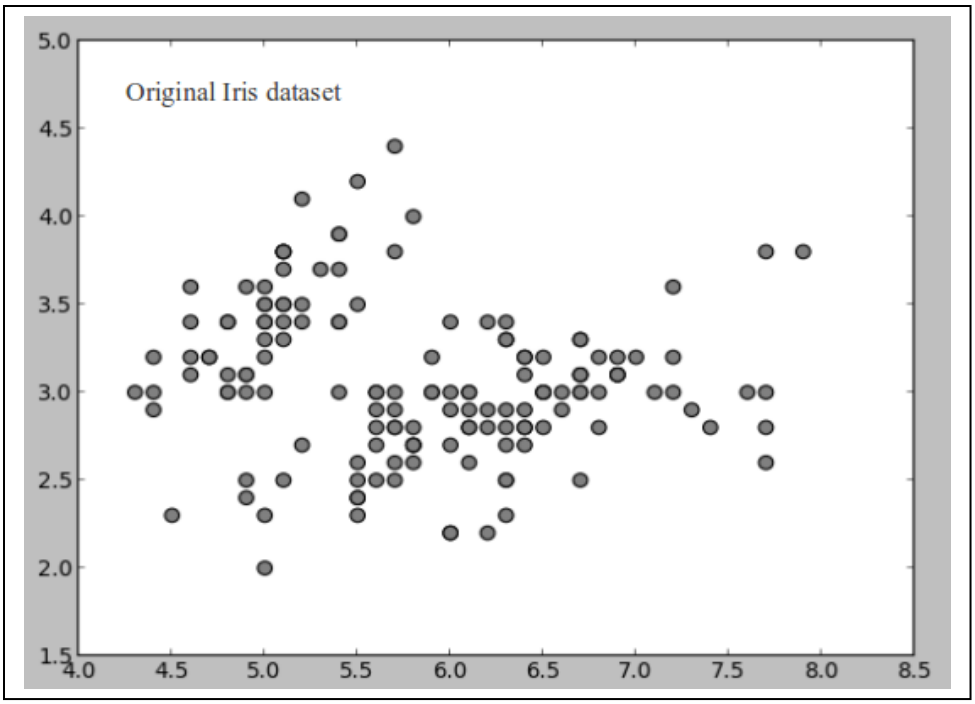

Fig 5.3: original Iris Dataset

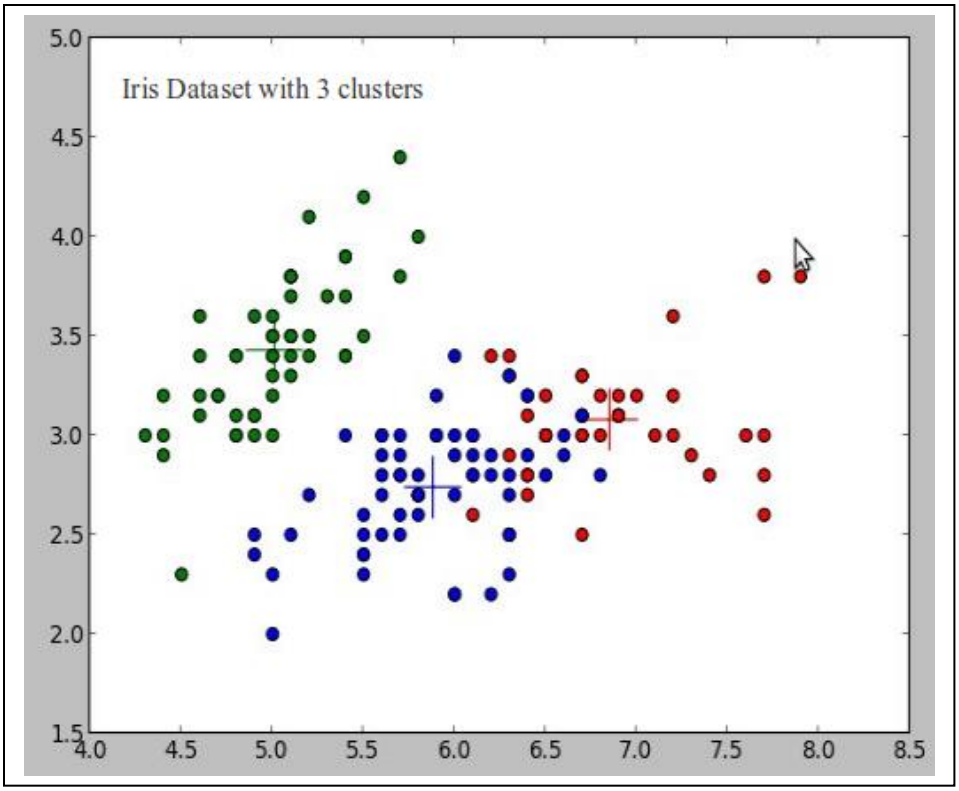

Fig 5.4: Iris Dataset with 3 clusters

The Environmental dataset having 111 data points has been shown in Figure 5.5. The algorithm is run for the Environmental dataset with input as k=3, i. e. for 3 clusters. The output of the algorithm has been shown in Figure 5.6. The threshold value given as input is 5. The algorithm produces 3 clusters having 39, 61 and 50 data points respectively.

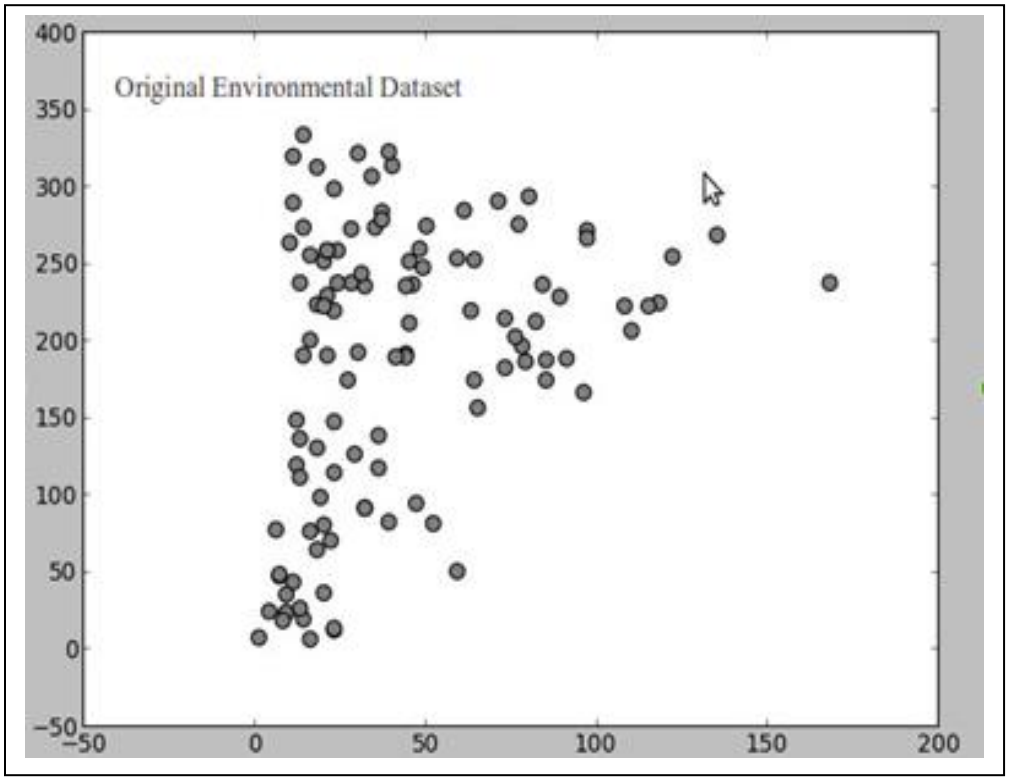

Fig 5.5: Original Environmental Dataset

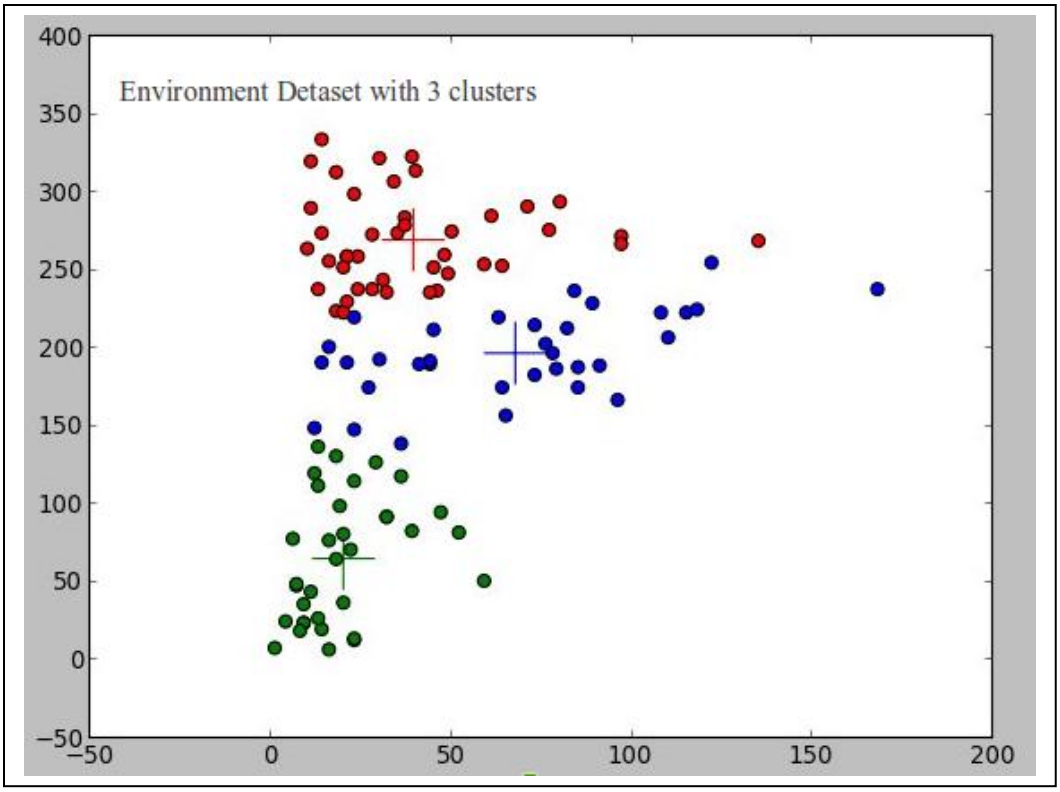

Fig 5.6: Environment Dataset with 3 clusters

The result of the program has been shown in the Table 5.1. The table shows the number of data points in dataset, the number of clusters to be made, i.e. size of k, and the number of calculations for distance function, and the threshold value taken for MEMBERSHIP-K-MEANS algorithm.

Table 5.1: Result of K-MEANS, Enhance Method & MEMBERSHIP_K-MEANS Algorithm

Dataset	Cluster	no. of time distance function called			
		Original k-means	Enhance Method [21]	Proposed Method (MEMBERSHIP_K-MEANS)	Threshold value
Ruspindi	4	1200	564	224	3
Iris	3	1350	714	317	4
Environmental	3	2664	2289	1123	5
Production	7	4200	3353	1233	6

Table 5.1 determines that the number of computation for distance function is less for the proposed algorithm than the k-means algorithm. In K-means algorithm in every iteration, each data point is checked for its membership in every cluster. However in the proposed algorithm, each data point has its membership set of cluster, which contains only those clusters which are hopeful to be the cluster of that data point. In case of large dataset and significantly large number of clusters, this algorithm results in much computation reduction. In k-means algorithm, in each iteration, distance function also runs for each data point and for every cluster, which results in huge calculation. In the proposed algorithm, this calculation is reduced by reusing the fact that if the distance of the data point from new cluster mean is less than the distance between old and new cluster mean, than the data point will be assigned to the same cluster, so there is no need to run any test for that data point. From these facts it can be seen that the proposed algorithm results in computation reduction. Table 5.2 represents the Efficiency of the three algorithms in terms of time complexity. It can be easily seen that the proposed algorithm MEMBERSHIP_K-MEANS shows significant improvements in time and accuracy.

Table 5.2: Time of K-MEANS, Enhance Method & MEMBERSHIP_K-MEANS

Dataset	Time(sec.)		
	Original k-means	Enhance method [21]	Proposed Method (MEMBERSHIP_K-MEANS)
Ruspindi	.082	.062	.058
Iris	.096	.086	.057
Environment	.072	.043	.023
Production	.092	.076	.053

The comparison of time taken by the all the three algorithms have been shown by graph in Figure 5.7. The graph shows that the time taken for all the four datasets in the clustering is least for the Proposed Method, MEMBERSHIPK-MEANS.

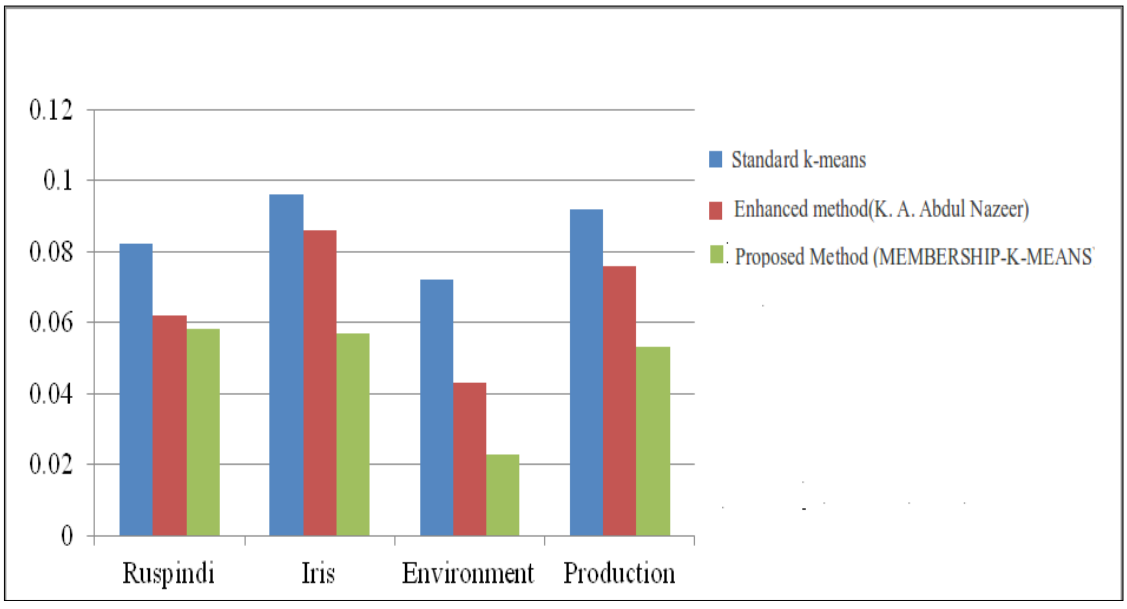

Fig 5.7: Time Comparison of Standard, Enhanced and proposed K-means Algorithm

Threshold taken depends on the number clusters to be created and the dataset into consideration. The study of threshold is done for Ruspindi dataset, Iris dataset, Environmental dataset, and Production dataset. For the cluster value 3, 4, 5, 6, and 7, the correct value of threshold is found out. The threshold values for different clusters for all four datasets have been shown in graph in Figure 5.8. It has been observed that as the number of clusters increase, the threshold decrease. In Figure 5.8 (a) For Ruspindi dataset, If the algorithm is run for 3 clusters than the required value of threshold should be set as 7. For others values of threshold less than 7, it generates flawed clusters. Similarly for 4, 5, 6 and 7 clusters, the threshold values which gives desired clusters are 5, 5, 4 and 3 respectively.

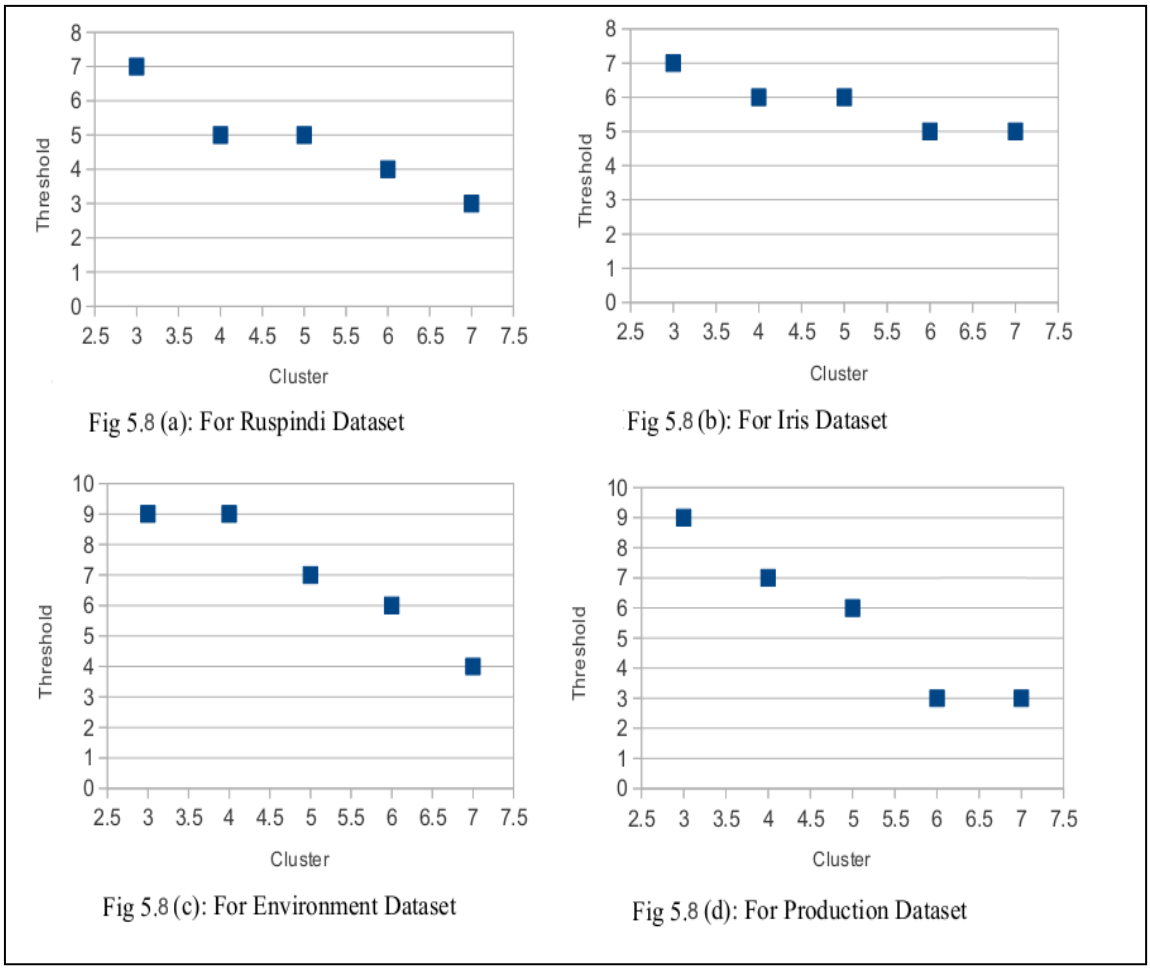

Fig 5.8: Threshold values for different clusters for different datasets

In Figure 5.8 (b), for Iris dataset, to make 3, 4, 5, 6 and 7 clusters, the threshold values which gives desired clusters are 7, 6, 6, 5, and 5 respectively. In Figure 5.8 (c) for Environment dataset, to make 3, 4, 5, 6 and 7 clusters, the threshold values which gives desired clusters are 9, 9, 7, 6, and 4 respectively. In Figure 5.8 (d) for Production dataset, to make 3, 4, 5, 6 and 7 clusters, the threshold values which gives desired clusters are 9, 7, 6, 3, and 3 respectively. From all these results we can notice that the value of the threshold has a relationship with the dataset, and the number of clusters to be formed. The experimental results reflects that as the number of clusters to be formed is increased the threshold value decreases.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

Time complexity is the main issue with K-means when dataset are very large and is high dimensional. To address this problem, in this thesis, a modified membership set based K-means algorithm MEMBERSHIP_K-MEANS has been proposed to reduce the time complexity. It maintains a membership set for every data point. The membership set contain those clusters, for which there is a hope that it might be the cluster for the data point. The membership set has three tuples- cluster, is distance from data point, and a counter initialized with zero. The counter maintains for how many times the particular data point has not been assigned to that cluster. After the counter exceeds the predefined threshold, that cluster can be removed from the hope set for that data point.

The proposed algorithm has been validated and run for real dataset like Ruspindi, Iris, Environmental, and Production and compared with the standard k-means algorithm, and one K-means Enhanced Method [21]. It has been observed from the output that the proposed algorithm is more efficient than the other two methods. The results clearly demonstrate that the proposed algorithm is efficient in terms of time complexity. The experimental results also show the behaviour of the threshold. It results shows that as the number of clusters increases, the threshold value decreases as it depends on the range and types of data points and the number of clusters.

6.2 Future Scope

The proposed MEMBERSHIP_K-MEANS algorithm can be validated for multiple datasets. The idea of the membership hope set can be applied to other variants of k-means clustering. Dynamic threshold value for particular dataset can also be implemented in near future.

References

[1] U. Fayyad, "Data Mining and Knowledge Discovery in Databases: Implications for Scientific Databases," In Proceedings of Ninth International Conference on Scientific and Statistical Database Management, pp. 2-11, 1997.

[2] Ganti, Gehrke, and Ramakrishnan, "Mining very large databases," IEEE Computer Society, vol.32, no.8, pp. 38-45, 1999.

[3] Charu C. Aggarwal , and Philip S. Yu, "Data Mining Techniques for Associations, Clustering and Classification," In Proceedings of The Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, pp.13-23, 1999.

[4] A. K. Jain , M. N. Murty , P. J. Flynn, "Data clustering: a review," ACM Computing Surveys (CSUR), v.31 n.3, pp. 264-323, 1999.

[5] Anil K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, v.31 n.8, pp. 651-666, 2010.

[6] C. Apte, "Data mining: an Industrial Research Perspective," Computational Science & Engineering, IEEE , vol.4, no.2, pp. 6,9, 1997.

[7] Loureiro, Antonio, Luis Torgo, and Carlos Soares, "Outlier Detection using Clustering Methods: a Data Ceaning Application," In Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector, 2004.

[8] C. Hennig, and B. Hausdorf, "Design of Dissimilarity Measures: A New dissimilarity Measure between Species Distribution Ranges," Data Science and Classification, Studies in Classification, Data Analysis, and Knowledge Organization, Springer-Verlag GmbH, Berlin, German, pp. 29–38, 2006.

[9] Schaeffer, and Satu Elisa, "Graph clustering," Computer Science Review, pp. 27-64, 2007.

[10] A. K. Jain, and R. C. Dubes, "Algorithms for Clustering Data," Prentice-Hall, Englewood Cliffs, NJ, 1988.

[11] T. VELMURUGAN, "Efficiency of k-Means and K-Medoids Algorithms for Clustering Arbitrary Data Points ", International Journal of Computer Technology & Applications , 2012.

[12] Anand Rajaraman, Jure Leskovec, and Jeffrey D. Ullman, “Clustering,” in Mining of Massive Datasets, 4th ed., 2013, ch. 7, pp. 239-278.

[13] G. McLachlan and K. Basford, “Mixture Models: Inference and Applications to Clustering”, Marcel Dekker, New York, NY, 1988.

[14] J. Hartigan and M. Wong., “A k-means clustering algorithm.”, Applied Statistics, 28:100–108, 1979

[15] Vance Faber, “Clustering and the Continuous k-Means Algorithm”, Los Alamos Science, 1994.

[16] Ming-Chuan Hung, Jungpin Wu, Jin -Hua Chag, and Don-Lin Yang, “An Efficient k-Means Clustering Algorithm Using Simple Partitioning”, Journal Information Science and Engineering, 2005

[17] Malay K. Pakhira, “A Modified k-means Algorithm to Avoid Empty Clusters”, International Journal of Recent Trends in Engineering, 2009.

[18] Jieming Wu, and Wenhu Yu, “Optimization and Improvement based on K-Means Cluster Algorithm ”, Second International Symposium on Knowledge Acquisition and Modeling , 2009.

[19] Shehroz S. Khan , and Amir Ahmad , “Cluster center initialization algorithm for K-means clustering,” Pattern Recognition Letters, vol. 25, pp. 1293-1302, 2004

[20] Rupali Vij, and Suresh Kumar, “Improved k- means clustering algorithm for two dimensional data,” In Proceedings of The Second International Conference on Computational Science, Engineering and Information Technology, pp. 665-670, 2013

[21] K. A. Abdul Nazeer, and M. P. Sebastian , “Improving the Accuracy and Efficiency of the k-means Clustering Algorithm, ” In Proceedings of the World Congress on Engineering, 2009.

[22] Kohei Arai, and Ali Ridho Barakbah, “Hierarchical K-means: an algorithm for centroids initialization for K-means ”, Department of Information Science , 2007.

[23] S. Sujatha, and A. Shanthi Sona, “New Fast K-Means Clustering Algorithm using Modified Centroid Selection Method” , International Journal of Engineering Research & Technology , 2013.

[24] A. Sridhar, and S. Sowndarya, “Efficiency of K-Means Clustering Algorithm in Mining Outliers from Large Data Sets ”, International Journal on Computer Science

and Engineering , 2010.

[25] Parul Agarwal, M. Afshar Alam, and Ranjit Biswas, "Issues, Challenges and Tools of Clustering Algorithms," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011

[26] Source: github [Online]. Available: http://vincentarelbundock.github.io/Rdatasets/datasets.html

List of Publications

[1] Varun Kumar Sharma, Anju Bala, “Clustering for High Dimensional Data,” ICNSC- 2014 Confrence, IEEE Explore, 19 August 2014, (Accepted).