**Introduction**

In the field of data mining and pattern recognition, we have to deal with many different kind of datasets. In most of the cases these datasets, which we encounter are unlabeled and therefore unsupervised learning is used to segment this data.

Clustering is widely used in Knowledge Discovery in Databases (KDD) and K-means algorithm is one of the well-known clustering algorithms. K-means is highly dependent on the selection of initial cluster centers, hence it is important to properly initialize the cluster centers to get accurate results. Due to random selection of initial cluster centers, K-means doesn’t give accurate and unique results always. The number of iterations in K-means algorithm is also significantly influenced by the initial cluster centers so the computational complexity is very high for large datasets.

There have been many papers published on this topic with various enhancements to the original K-means algorithm. Every method has its own advantages and shortcomings. A few of those methods have been described in the papers mentioned below:

Fang Yuen [1] has given a systematic method for finding the initial centers. The centers calculated by this method are consistent with the distribution of the dataset. Hence the accuracy is better than the initial K-means algorithm but this method is unable to provide any improvement to the time complexity of the k-means algorithm.

A.M. Fahim [2] has given a method to assign data points to suitable clusters. This method has reduced the computational time by assigning the data elements to the appropriate clusters but the drawback of this method is that the initial centers are selected randomly so the resulting clusters are not that accurate.

A. Bhattacharya [3] has also given a modified clustering algorithm called Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes. DCCA method can produce clusters without taking the values of initial cluster centers and the value of K (number of desired clusters) as an input. But the time complexity of the algorithm is high and the cost for repairing for any misplacement is also very high.

Our research issue will be based on comparing the results for three different cluster center initialization methods on the basis of accuracy and efficiency.

The first method is based on selecting random initial cluster centers from the given dataset which is the original k-means clustering method. The k-means algorithm starts by initializing the K cluster centers. The input data points are then allocated to one of the existing clusters according to the square of the Euclidean distance from the clusters, choosing the closest. The mean of each cluster is then computed so as to update the cluster center. This update occurs as a result of the change in the membership of each cluster. The process of re-assigning the input data points and the update of cluster center is repeated until there is no more change in the value of any of the cluster centers.

The second method is the ‘Distance Sorting’ method which is based on the research paper [4] whose algorithm is, sorting the dataset on the basis of distance of each point from origin, dividing it into K equal parts and then finding the median point for each part, these median points are used as initial cluster centers for the k-means algorithm.

The third method is the ‘Variance Sorting’ method which is made by combining the methods mentioned in the two research papers [4] & [5]. The algorithm proposed here is based on finding the dimension of the dataset which has maximum variance, sorting it, dividing it into K sets then finding the median for each set and then using the corresponding data points to initialize the cluster centers for the k-means algorithm.

At the end of our research and tests we will compare the results obtained using the three methods described above. The results would be compared on two different sizes of data sets which are taken from the UCI Machine Learning Repository. The accuracy of the three methods will be checked by purity test and percentage error values.

**Datasets**

1. Haberman’s Survival Data Set:

The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. The Data set and attribute characteristics are multivariate and integer respectively. It contains total 306 number of instances and 3 number of attributes. The three attributes in this data are- age of patient at time of operation, patient’s year of operation, number of positive axillary nodes detected. The two known classes which are 1-‘the patient 5 years or longer’ and 2-‘the patient died within 5 year’.

1. Tamilnadu Electricity Board Hourly Readings Data Set:

The dataset contains the real time readings for residential, commercial, industrial, agriculture, to find the accuracy consumption in Tamil Nadu around Thanajvur. The Data set and attribute characteristics are multivariate and real respectively. It contains total 45780 number of instances and 2 number of attributes. There are seven known classes in the dataset. The two attributes in this data are- meter readings (kilo volt ampere) and meter readings (kilo watt hour).

**Methods**

1. K-means Clustering Algorithm

The first method which we are using to cluster the two datasets is the original k-means method. This method generates K random initial cluster centers, where K is number of clusters specified by the user. At first the Euclidean distance between each data point and each cluster is computed, the data points are assigned to the cluster which gives the smallest distance. After assigning all the data points to different clusters, the mean of each cluster is updated. These new means act as the new cluster centers and again the distance of each data point is calculated with each cluster center. This process is repeated until the convergence criteria is met i.e. the centers remain the same.

Algorithm**:**

**Given**: S= {S1, S2, S3, S4,…….,Sn} where n is the total number of data points.

K= number of clusters

**Require**: A set of K clusters

**Steps**:

1. Randomly choose K data points from S as initial cluster centers;
2. Assign each data point Si to the cluster which has the closest center;
3. Calculate the new center for each center.
4. Repeat step 2 and step 3 until convergence criteria is met;

**End**

1. Distance Sorting Method

In this method first it is checked if all the data point attributes are positive values. If there is a negative value in any of the data point then the whole dataset set is subtracted by the minimum negative value. This is done to make all the points positive and to transform all the data points into the positive space. After doing this the Euclidean distance of each transformed data point is calculated from the origin, then all the data points are sorted according to the Euclidean distance. This sorted dataset is now divided into K equal sets where K is the number of clusters. After this the mid-point of all the K sets are taken as the initial cluster center for the k-means algorithm.

Algorithm:

**Given**: S= {S1, S2, S3, S4,…….,Sn} where n is the total number of data points.

K= number of clusters

**Require**: A set of K clusters

**Steps**:

1. If the data points in S contain any negative then go to step 2 otherwise go to step 4;
2. Find the minimum attribute value in the given dataset S;
3. Subtract the minimum attribute value from each data point in the data set;
4. Calculate the Euclidean between distance origin and each data point;
5. Sort the data point according to the Euclidean distance calculated above;
6. Divide the sorted dataset into K equal sets;
7. Make the middle point in each K sets as the initial cluster center;
8. Assign each data point Si to the cluster which has the closest center;
9. Calculate the new center for each center;
10. Repeat step 8 and step 9 until convergence criteria is met;

**End**

1. Variance Sorting Method

In this method first the variance of all the dimension of the dataset is calculated. The dimension that gives the maximum variance is selected and sorted. The sorted dimension is divided into k sets. The median of each set is calculated and the corresponding values are taken as the initial cluster center for k-means algorithm. Unlike k-means after assignment of all data points to its nearest cluster and finding new mean, the distance of each data point from the current nearest cluster center and from the previous cluster center is compared, if the current distance is less, then the data point stays in the same cluster otherwise distance of all clusters are calculated from that data point and assigned to nearest cluster center.

Algorithm:

**Given**: S= {S1, S2, S3, S4,…….,Sn} where n is the total number of data points.

K= number of clusters

**Require**: A set of K clusters

**Steps**:

1. Calculate the variance of each dimension of the dataset;
2. The dimension with maximum variance call it vmax is selected and is sorted;
3. The data points of vmax is then divided into k equal sets;
4. Find the median of each subset;
5. The corresponding data points for each median is selected to initialize the cluster centers;
6. Assign each data point Si to the cluster which has the closest center;
7. Calculate the new center for each center;
8. **For** each data point Si

Calculate the distance from the present nearest cluster;

If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster;

**Else**

For every centroid the distance is calculated again.

**End**

1. Repeat step 7, step 8 and step 9 until convergence criteria is met;

**End**

**RESULTS**

Dataset 1: Haberman’s Survival Data Set

The clustering is done using all the three methods described above and the clustering results are as follows: