International Journal of Computational Intelligence and Informatics, Vol. 2: No. 3, October - December 2012

Detecting Outliers using PAM with Normalization Factor on Yeast Data

P Ashok
Bharathiar University
Coimbatore, Tamilnadu, India
ashokcutee@gmail.com

G M Kadhar Nawaz
Sona College of Technology
Salem, Tamilnadu, India
nawazse@yahoo.co.in

E Elayaraja
Periyar University
Salem, Tamilnadu, India
elayarajaphd.e@gmail.com

***Abstract*- Protein is a macro nutrient composed of amino acids that is essential for the proper growth and function of the human body. While the body can construct several amino acids required for protein production, a set of fundamental amino acids needs to be obtained from animal and/or vegetable protein sources. Cluster analysis is a one of the primary data analysis tools in data mining. Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other cluster. Clustering can be performed on Nominal, Ordinal and Ratio-Scaled variables. The main purpose of clustering is to reduce the size and complexity of the dataset. In this paper, we introduce the method clustering and its type's K-Means and K-Medoids. The clustering algorithms are improved by implementing the three initial centroid selection methods instead of selecting centroids randomly, which is compared by Davies Bouldin index measure. Hence selecting the initial centroids selection by systematic selection (ICSS) algorithm overcomes the drawbacks of selecting initial cluster centers then other methods. In the yeast dataset, the defective proteins (objects) are considered as outliers, which are identified by the clustering methods with ADOC (Average Distance between Object and Centroid) function. The outlier's detection method and computational complexity method is studied and compared, the experimental results shows that the K-Medoids method performs well when compare with the K-Means clustering.**

Keywords- Initial centroid selection, Outliers, Normalization, Computational Complexity, K-Medoids

I. INTRODUCTION

Cluster analysis is a one of the primary data analysis tools in data mining. Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other cluster. Clustering can be performed on Nominal, Ordinal and Ratio-Scaled variables. The main purpose of clustering is to reduce the size and complexity of the dataset.

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification. It represents many data objects by few clusters, and hence, it models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. Therefore, clustering is unsupervised learning of a hidden data concept. Data mining deals with large databases that impose on Clustering analysis additional severe computational requirements. These challenges led to the emergence of powerful broadly applicable data mining clustering methods surveyed below. Types of Clusters are

1. Partitional algorithms: Construct various partitions and then evaluate them by some criterion.
2. Hierarchical algorithms: Create a hierarchical decomposition of the set of objects using some criterion

This paper is organized as follows. Section II presents an overview of Partitional Clustering techniques and its method. Section III describes performance of Experimental analysis and Discussion. Section IV presents conclusion and future work.

II. PARTITIONAL CLUSTERING METHODS

The partitioning methods [22] generally result in a set of M clusters, each object belonging to one cluster. Each cluster may be represented by a centroid or a cluster representative this is some sort of summary description of all the objects contained in a cluster. The precise form of this description will depend on the type of the object which is being clustered. In case where real-valued data is available, the arithmetic mean of the attribute vectors

for all objects within a cluster provides appropriate representative, alternative types of centroid may be required in other cases, and e.g., a cluster of documents can be represented by a list of those keywords that occur in some minimum number of documents within a cluster.

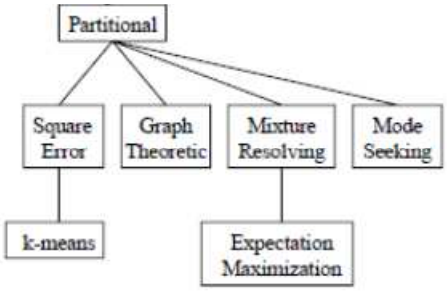

Fig 1. Types of Partitional Clustering Method.

In this section, we discuss about the two types of Partitional clustering algorithms are

- K-Means Algorithm
- K-Medoids Algorithm

1. K-Means Algorithm:

K-Means [7], [8], [16] is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group is done. At this point, it is need to re-calculate k new centroids as centres of the clusters resulting from the previous step. After these k new centroids, a new binding has to be done between the same data points and the nearest new centroid. A loop has been generated. As a result of this loop it may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

The K-Means algorithm is effective in producing clusters for many practical applications. But the computational complexity of the original K-Means algorithm is very high, especially for large Data sets. The K-Means clustering algorithm is a partitioning clustering method that separates data into K groups. For real life problems, the effective clusters centroids cannot be initialized. To overcome the above drawback the current research focused on developing the clustering algorithms instead of selecting the initial centroids randomly.

a. Advantages of K-Means Algorithm

- It is easy to implement and works with any of the standard norms.
- It allows straightforward parallelization.
- It is incentive with respect to data ordering

b. Drawbacks of K-Means Algorithm

- The final clusters do not represent a global optimization result but only the local one, and complete different final clusters can arise from difference in the initial randomly chosen cluster centres.
- We have to know how many clusters we will have at the first

2. K-Medoids Algorithm:

K-Means clustering is sensitive to the outliers and a set of objects closest to a centroid may be empty, in which case centroids cannot be updated. For this reason, K-medoids clustering are sometimes used, where representative objects called medoids are considered instead of centroids.

The K-Medoids algorithm [6], [9], [10], [17] is a clustering algorithm related to the K-Means algorithm. Both the K-Means and K-Medoids algorithms are Partitional technique of clustering that clusters the data set of n objects into k clusters with k known a priori and both attempt to minimize squared error, the distance between points labelled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm, K-Medoids chooses data points as centers (medoids or exemplars)

The basic strategy of K-Medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the Medoids or Exemplars) for each cluster. Each remaining object is clustered with the Medoid to which it is the most similar. K-Medoids method uses representative objects as reference points instead of taking the mean value of the objects in each cluster. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects.

In this paper, these two clustering algorithms are executed, compared and also identify the best clustering algorithm from the observed values from various situation which are explained in the following sections.

III. PROPOSED APPROACH

In this paper, the proposed approach is mainly concentrating on stopping criteria, which is used to stop the clustering algorithm more effectively than other criteria. Second one is selecting initial centroids selection using three centroid selection methods to produce the better results in clustering methods. Third one is outlier detection, which is used to detecting outliers in the given dataset and yields the better results in clustering process. The last one is time complexity which is compared by the k-means and k-medoids. These proposed works are explained briefly in the following section.

A. Stopping Criteria

The K-Means algorithm is said to have converged when recomputing the partitions does not result in a change in the partitioning. In the terminology that we are using, the algorithm has converged completely when $C^{(i)}$ and $C^{(i-1)}$ are identical. Sometimes the convergence of the centroids (i.e. $C^{(i)}$ and $C^{(i+1)}$ being identical) takes several iterations. Also in the last several iterations, the centroids move very little. As running the expensive iterations so many more times might not be efficient, we need a measure of convergence of the centroids so that we stop the iterations when the convergence criteria are met.

The objective function (Mean Square Error) [21] is used as the convergence criteria for the K-Means algorithm, it aims to minimize the sum of squared error and it shows better cluster result.

$$\mathbf{J} = \sum_{\mathbf{j=1}}^{\mathbf{C}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \left|\left| \boldsymbol{x_i} - \boldsymbol{c_j} \right|\right|^{\mathbf{2}} \quad \textbf{(1)}$$

Where J is the objective function, $||\boldsymbol{x_i} - \boldsymbol{c_j}||^{\mathbf{2}}$ is is a chosen distance measure between a data point x_i and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

B. Initial Centroid Selection Methods

The initial centroids play the main role in the clustering process. In the original k-means algorithm the initial centroids are taken just randomly out of the input data set. But this random selection of initial centroids leads the computation of the algorithm into local optima. Say, k is determined to be 3. If from a given data set we select first 3 points as initial centroids and compute the kmeans algorithm. Next time suppose we select the last 3 points as the initial centroids and further third time let we select any 3 arbitrary data points as initial centroids and compute the k-means algorithm. Each time the end clustering results will come out to be different. Then we have to analyze which one is the most appropriate result. Thus with the random selection of initial centroids there is no guarantee that the k-means algorithm will converge into best results. This is the limitation which needs to be dealt with in order to make the k-means algorithm more efficient.

Sometimes the algorithm produce bad clustering results due to selecting centroids randomly in the given data set, to avoid this problem, we introduce three different methods to select the initial centroids for the K-Means algorithm are listed and explained below.

1. Systematic Selection Method
2. systematic with Interval Method
3. Elimination Method

Algorithm 3: Initial Centroid Selection by Systematic Selection (ICSS) method

---

Steps

1. Using Euclidean distance as a dissimilarity measure, compute the distance between every pair of all objects as follows:

$$\boldsymbol{d_{ij}} = \sqrt{\sum_{\boldsymbol{a=}}^{\boldsymbol{p}} (\boldsymbol{X_{ia}} - \boldsymbol{X_{ja}})^2} \quad \boldsymbol{i, j = 1 \ldots . n} \qquad (2)$$

2. Calculate $\mathbf{M_{ij}}$ to make an initial guess at the centers of the clusters

$$\boldsymbol{M_{ij}} = \frac{\boldsymbol{d_{ij}}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{d_{ij}}} \quad \boldsymbol{i = 1 \ldots n; j = 1 \ldots n} \qquad (3)$$

3. Calculate $\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{M_{ij}^2}_{\ (\boldsymbol{j=1 \ldots . n})}$ at each object and sort them in ascending order.

4. Select K objects having the minimum value as initial cluster medoids.

Algorithm 4: Initial Centroid Selection by Elimination (ICSE) Method [1], [14].

Input:

D = {d1, d2,.....,dn} // set of *n* data items

***K* // Number of clusters**

Output: A set of *k* initial centroids.

Step:

1. Set m = 1.
2. Compute the distance between each data point and all other data- points in the set D.
3. Find the closest pair of data points from the set D and form a data-point set Am (1<= m <= k) which contains these two data- points, Delete these two data points from the set D.
4. Find the data point in D that is closest to the data point set Am, Add it to Am and delete it from D.
5. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k);
6. If m<k, then m = m+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Am and delete them from D, Go to step 4.
7. For each data-point set Am (1<=m<=k) find the arithmetic mean of the vectors of data points in Am, these means will be the initial centroids.

Algorithm 5: Initial Centroid Selection by Systematic with Intervals (ICSSI) method

Steps:

1. Using Euclidean distance as a dissimilarity measure, compute the distance between every pair of all objects as follows:

$$\boldsymbol{d_{ij} = \sqrt{\sum_{a=}^{p}(X_{ia} - X_{ja})^2} \;\; i = 1 \ldots . n; j = 1 \ldots . n} \quad (4)$$

2. Calculate $\boldsymbol{M_{ij}}$ to make an initial guess at the centers of the clusters

$$\mathbf{M_{ij} = \frac{d_{ij}}{\sum_{i=1}^{n} d_{ij}} \;\; i = 1 \ldots n; j = 1 \ldots n} \quad (5)$$

3. Calculate $\sum_{i=1}^{n} \mathbf{M_{ij}^2}_{\;(j=1\ldots.n)}$ at each object
4. Sort all objects in the order of values of the chosen variable.
5. Divide the range of the above values into K equal intervals
6. Select one object randomly from each interval as the initial centroid.

C. Outliers Detection

The data objects that do not comply with the general behavior or model of the data, Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers [3], [12], [13]. The outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining or outlier analysis.

The outliers are detected by using various techniques which is listed below.

1. Statistical Tests
2. Deviation-based Approaches
3. Distance statistical model
4. Distance-based Approaches
5. Density-based Approaches

In this paper, the defective protein sequence are considered as outliers which are detected by using two clustering algorithms are K-Medoids and K-Means with Distance based outlier detection method. From the above outlier techniques, we use the Distance Based method to detect the outliers.

1. Outlier Removal Clustering (ORC) and Normalization

The objective of the [23] outlier detection algorithm that we call Outlier Removal Clustering (ORC) is to produce a codebook as close as possible to the mean vector parameters that generated the original data. It consists of two consecutive stages, which are repeated several times. In the first stage, finding the maximum distance between the object and centroid in the cluster, in the second stage, we assign an outlyingness factor for each vector. Factor depends on its distance from the cluster centroid. Finding the object with maximum distance to the centroid

$$\boldsymbol{d_{max}} = \mathbf{max}\{\ ||\boldsymbol{x_i} - \boldsymbol{c_i}||\ \} \quad (6)$$

Outlyingness factors for each vector are then calculated. We define the outlyingness of a vector $\boldsymbol{x_i}$ as follows:

$$\boldsymbol{O_i} = \frac{||\boldsymbol{x_i} - \boldsymbol{C_i}||}{\boldsymbol{d_{max}}} \quad (7)$$

We see that all outlyingness factors of the dataset are normalized to the scale [0, 1]. The greater the value, the more likely the vector is an outlier.

Algorithm 6: Outlier Removal Clustering (ORC)

C←Run K-Means with multiple initial solutions
For j=1 to I do
$\boldsymbol{d_{max}} = \mathbf{max}\{||\boldsymbol{x_i} - \boldsymbol{c_i}||\}$
For I =1 to n do
$\boldsymbol{O_i} = \frac{||\boldsymbol{x_i} - \boldsymbol{C_i}||}{\boldsymbol{d_{max}}}$
If O_i > T then
X=X / {x_i}
endif
endfor

(C, P)=Kmeans(X, C)
endfor

The objects for which $\mathbf{o_i} > \boldsymbol{T}$, are defined as outliers. By setting the threshold to $\boldsymbol{T} < \mathbf{1}$, at least one vector is removed. Thus, increasing the number of iterations and decreasing the threshold.

D. Computational Complexity

The required computational complexity of K-Means algorithm [17], [10] is ***O(nkl)***. Where '**n**' is the number of data points, '**k**' is the number of clusters and '***l***' is the number of iterations. To get the initial clusters the required computational complexity is ***O(nk)***. Here, some data points stay in the cluster itself and some other data points move to other clusters based on their relative distance from old centroid and the new centroid. If the data point stays in the same cluster then the required complexity is ***O(1)***, otherwise ***O(k)***. In each iteration, the moving of data points to other clusters is decreases. Hence the total computational complexity for assigning the data points is to the clusters is ***O(nkl)***.

The computational Complexity of the K-Medoids algorithm [10] is $\boldsymbol{O(lk(n-k)^2)}$. Here '**k**' is the number of medoids and '***l***' is the total number of iterations. The K-Medoid algorithms having less number of iteration other then K-Means algorithm to complete the clustering process. Hence the K-Medoids algorithm has less computational complexity compared to the original K-Means clustering algorithm.

IV. EXPERIMENTAL ANALYSIS AND RESULTS

In this section, we describe the data sets used to analyses the methods studied in sections II and the results are arranged and listed in the Table II to Table VI, number of features are in column wise, and number of items/samples are in row wise.

A. Dataset Description

In this research work, we use the **yeast** dataset [11], which is obtained from the website www.ics.uci.edu.

Yeast Data Set: In this experiment, we use a yeast data set which has been used to find localization site of protein. The data set contains 1400 records (objects), each with attributes (8 real-valued input features). In the

yeast dataset, eight features (attributes) are used: mcg, gvh, alm, mit, erl,pox, vac, nuc. Proteins are classified into various clusters. cytosolic or cytoskeletal (CYT), nuclear (NUC), mitochondrial (MIT), membrane protein without N-terminal signal (ME3), membrane protein with uncleaved signal (ME2), membrane protein with cleaved signal (ME1), Extracellular (EXC), vacuolar (VAC), peroxisomal (POX), Endoplasmic reticulum lumen (ERL). We regard the objects having peroxisomal targeting signal in the C-terminus (POX) value zero as normal data, Whereas POX value greater than 2 as abnormal. In this experiment we use all 1400 records as normal objects and added some abnormal objects as outliers. More related information about this data set can be reached at http://archive.ics.uci.edu/ml/datasets/Yeast.

B. Comparative study and Performance Analysis

1. Stopping Criteria

In this paper we define two kinds of convergence criteria for the K-Means clustering algorithm which are listed and defined as below.

Two types of convergence criteria

- Centroid based criteria
- Objective function based criteria

The K-Means clustering algorithms has converged completely when $C^{(i)}$ and $C^{(i-1)}$ are identical. Where $C^{(i)}$ is the centroid of the i^{th} iteration, by this method the algorithms is executed too many iteration to avoid this situation we use the alternative method called Objective Function which reduce the iteration count, the objective function is described in the equation from Section II.

The K-Means algorithm is executed by the two Stopping criteria methods with the various cluster values and the observed DB index values are listed in the following Table II.

TABLE I. Stopping Criteria For K-Means

S. No	Clusters	Stopping criteria	
		Objective function	*Centroid method*
1	5	1.3800	1.3836
2	10	1.5178	1.5417
3	15	1.5162	1.5276
4	20	1.5414	1.525
5	25	1.5300	1.5292
6	30	1.5110	1.5132
7	35	1.4833	1.5039
8	40	1.4131	1.4856
9	45	1.4661	1.4667
10	50	1.4708	1.4651

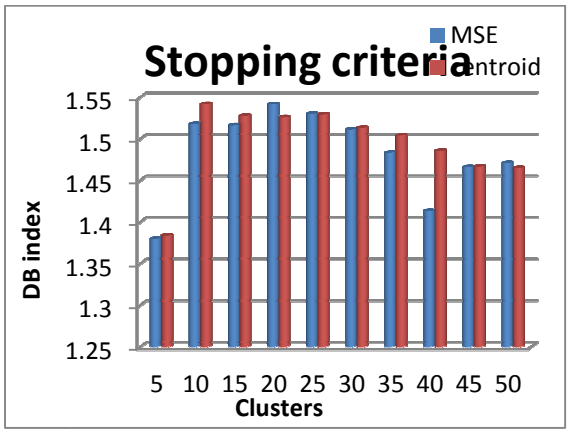

Fig 1. Stopping Criteria for K-Means

From the figure 1, we can able to identify that the objective function based convergence criteria is produce the minimum DB index value for the K-Means algorithm from different cluster values.

2. Initial Centroid Selection Methods

Performance of K-Means clustering algorithms which converges to numerous local minima depends highly on initial cluster centers. Generally initial cluster centers are selected randomly. In this section, the initial centroid selection algorithm is compared with random centroid selection method, the initial centroid selection algorithms [9] is already explained in Algorithm 3, 4 and 5, here the dataset is constantly fixed for the different

clusters, the three methods are executed and the observations are listed in the below Table II and their results shows that the performance of the K-Means algorithm is improved.

TABLE II. Initial Centroids Selection Methods For K-Means

S. No	Clusters	Initial Centroid selection method		
		Systematic (ICSS)	*Systematic with Interval (ICSI)*	*Elimination (ICSE)*
1	2	1.5612	1.5265	1.5801
2	4	1.3097	1.3217	1.5125
3	6	1.3701	1.3323	1.4264
4	8	1.3198	1.4071	1.3878
5	10	1.3612	1.3915	1.4224
6	12	1.4285	1.4042	1.4561
7	14	1.3005	1.3441	1.4012
8	16	1.4154	1.3902	1.3368
9	18	1.3474	1.3999	1.4159
10	20	1.2784	1.3032	1.3875

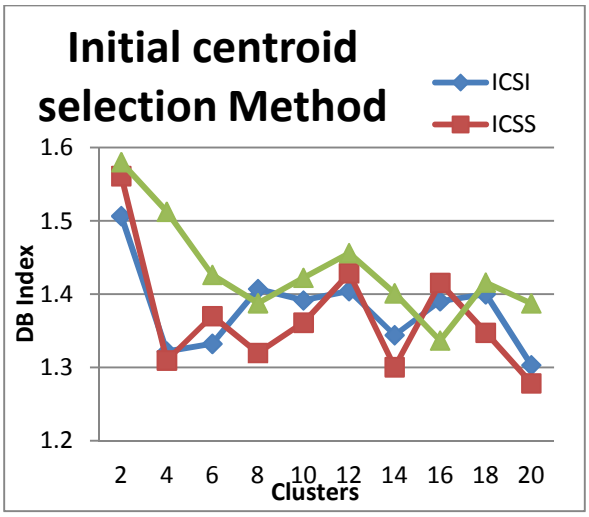

Fig 2. Initial Centroid selection chart for K- Means

From the figure 2 clearly shows that the DB index value for the initial centroid selection by Systematic Selection (ICSS) method achieve better results than the ICSI and Elimination method for most of the different clusters values. Hence the ICSS method for selecting initial centroids proved to improve the performance of the K-Means method.

3. Outliers Detection

The K-Means and K-Medoids clustering algorithms are used to find the outliers, here the outliers detected from various method. In this paper the outliers are identified by the method which is discussed in section II, the both algorithms are able to identify the outliers which are listed in the Table III.

TABLE III. Outliers Detections Methods

S. No	Clusters	Outlyingness parameter	Outliers	
			K-Means	*K-Medoids*
1	10	0.9	6	9
2		0.8	15	19
3		0.7	33	29
4		0.6	51	67
5		0.5	79	71
6		0.4	91	103
7		0.3	133	168
8		0.2	167	204

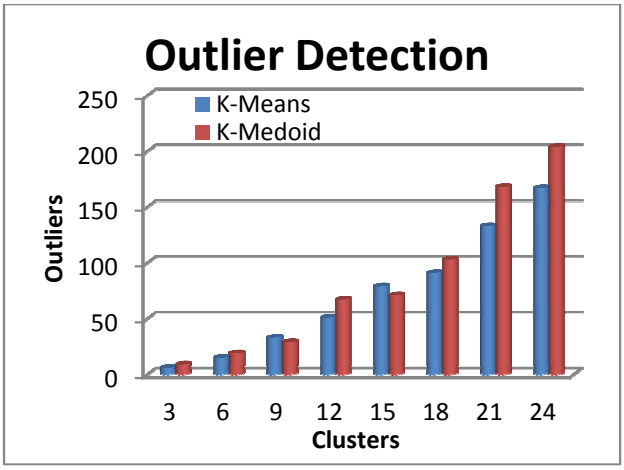

Fig 3. Outlier's Detection chart for K-Means and K-Medoids.

From the Figure 3, the outliers are detected by both clustering algorithms with the help of the outlyingness parameter but the K-Medoids clustering algorithms is detecting more number of outliers then the K-Means clustering algorithm due to k-Medoids clustering having minimum intra cluster distance (distance between the object and centroid) or more compact, hence the K-Medoids clustering algorithm is more efficient to identify the outliers then K-Means clustering algorithm but not effective for large dataset.

4. Computational Complexity

The Computational complexity of the two algorithms are discussed in the section , the computational complexity of the K-Means and K-Medoids clustering algorithms are calculated with various cluster values then the computational complexity values are listed in the Table IV.

TABLE IV. Computational Complexity For Clustering Methods

S. No.	Clusters	COMPUTATIONAL COMPLEXITY (in seconds)	
		K-Means	K-Medoid
1	2	0.0184	0.0071
2	3	0.0457	0.0154
3	4	0.0939	0.0441
4	5	0.1181	0.0475
5	6	0.0534	0.0316
6	7	0.1071	0.0861
7	8	0.0841	0.0478
8	9	0.1122	0.0472
9	10	0.1061	0.0342

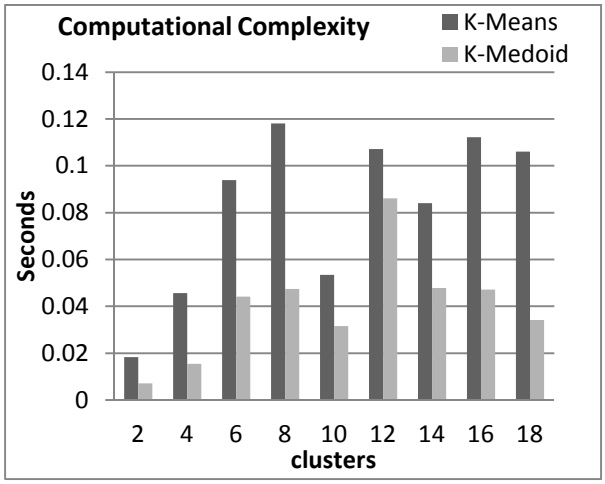

Fig 4. Computational complexity chart for clustering methods

From the figure 4, the Computational complexity of the K-Means and K-Medoids algorithms are clearly shown, the K-Medoid clustering algorithm obtain the minimum computational complexity for all the different cluster

values due to K-Medoid algorithm converged with small number of iterations, Hence the K-Medoids clustering algorithm is better than the K-Means algorithm.

V. CONCLUSION

In this work, the Partitional clustering methods are studied and apply the K-Means algorithm with objective function to find the optimal number of cluster centroids. One of the demerits of K-Means algorithm is random selection of initial seed point of desired clusters. This was overcome with three initial cluster centroid selection methods for finding the initial centroids to avoid the selecting centroids randomly and it produces different better results. Here the outliers (defective protein) are considered as dissimilar objects which are located in the each cluster. The outlier's detection and computational complexity for the both clustering algorithms are studied, tested and detected, but the K-Medoids method perform very well to detect outliers and having less computational complexity then K-Means due to the k medoids clustering algorithm converged within few iteration. Both the algorithms were tested with Yeast dataset and analysis the performance at various cluster values using Davis Bouldin measurement. Therefore, compare different clustering algorithms with various cluster validity measures are used to improve the cluster performance and also improve the K-Medoids algorithm for large dataset is our future work.

REFERENCES

[1] Abdul Nazeer K.A. and M. P. Sebastian, July 2009 "Improving the accuracy and efficiency of the K-Means clustering algorithm", international Conference on Data Mining and Knowledge Engineering (ICDMKE), Proceedings of the World Congress on Engineering (WCE- 2009), London, UK. Vol.1

[2] Bannai H, Tamada Y, Maruyama O, Nakai K, Miyano S: Bioinformatics 2002, 18(2):298-305.

[3] Chiu, A. and A. Fu, 2003. "Enhancement on Local Outlier Detection." 7th International Database Engineering and Application Symposium (IDEAS03)", pp. 298-307.

[4] Chou, K. C.; Shen, H. B. Review: Recent progresses in protein subcellular location prediction. Anal. Biochem., 2007, 370, 1-16.

[5] Davies & Bouldin, 1979. Davies, D.L., Bouldin, D.W., (2000) "A cluster separation measure." IEEE Trans.Pattern Anal. Machine Intell., 1(4), 224-227.

[6] Hae-Sang Park, Jong-Seok Lee, and Chi-Hyuck Jun, "K-means-like Algorithm for K-medoids Clustering and Its Performance".

[7] HARTIGAN, J. and WONG, M. 1979. Algorithm AS136: "A k-means clustering algorithm". Applied Statistics, 28, 100-108.

[8] HEER, J. and CHI, E. 2001. "Identification of Web user traffic composition using multimodal clustering and information scent.", 1st SIAM ICDM, Workshop on Web Mining, 51-58, Chicago, IL.

[9] Horton P, Park KJ, and Obayashi T, Nakai K: Protein subcellular localization prediction with WoLF PSORT. Proceedings of the 4th Annual Asia Pacific Bioinformatics Conference (APBC'06): 13–16 February 2006; Taipei, Taiwan 2006, 39-48.

[10] Kaufman, L. and Rousseeuw, P.J. (1987), "Clustering by means of Medoids, in Statistical Data Analysis Based on the L1–Norm and Related Methods", edited by Y. Dodge, North-Holland, 405–416.

[11] Kenta Nakai & Minoru Kanehisa, "A Knowledge Base for Predicting Protein Localization Sites in Eukaryotic Cells", Genomics 14:897-911, 1992.

[12] Knorr, E. and R. Ng, Algorithms for Mining Distance-based Outliers in Large Data Sets, 1998. Proc. the 24th International Conference on Very Large Databases (VLDB), pp. 392-403.

[13] Loureiro,A., L. Torgo and C. Soares, 2004. Outlier Detection using Clustering Methods: a Data Cleaning Application, in Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector. Bonn, Germany.

[14] Madhu Yedla, Srinivasa Rao Pathakota, T M Srinivasa , 2010 "Enhancing K-Means Clustering Algorithm with Improved Initial Center" , Madhu Yedla et al. / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 1 (2), pp121-125.

[15] Nakai, K. Protein sorting signals and prediction of subcellular localization. Adv. Protein Chem., 2000, 54, 277-344.

[16] Sauravjoyti Sarmah and Dhruba K. Bhattacharyya. May 2010 "An Effective Technique for Clustering Incremental Gene Expression data", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 3.

[17] Velmurugan T and T. Santhanam, "Computational Complexity between K-Means and K-Medoids Clustering Algorithms," Journal of Computer Science, vol. 6, no. 3, 2010.

[18] Yuan F, Z. H. Meng, H. X. Zhangz, C. R. Dong, August 2004 " A New Algorithm to Get the Initial Centroids", proceedings of the 3rdInternational Conference on Machine Learning and Cybernetics, pp. 26-29.

[19] http://en.wikipedia.org/wiki/Organelle

[20] http://en.wikipedia.org/wiki/Protein

[21] Velmurugan T. and Santhanam T, Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points, Journal of Computer Science 6 (3) (2010), 363-368.

[22] Pradeep Rai and Shubha Singh,A "Survey of Clustering Techniques", International Journal of Computer Applications (0975 – 8887) Volume 7– No.12, October 2010.

[23] Hautamaki,V.Cherednichenko, S.,Karkkainen, I.,Kinnunen, T.,and Franti, P.2005. Improving K-Means by Outlier Removal. In: SCIA 2005, pp.978-987.