COMPARISON OF DIFFERENT PARTITIONAL ALGORITHMS BASED ON EFFICIENCY

[1]Nasreen Taj M.B, [2]Neelambike S

Assistant Professor
Information Science and Engineering,
G M Institute of Technology, Davangere, India

***Abstract* - In this paper we describe cluster analysis and compare different Partitional algorithms based on their efficiency by considering gene data. For the Partitional algorithms we are going to give DNA sequences of one or two animals (like human, rat, monkey) as input. Here we take the input, DNA sequence for different animals, from the website www.ncbi.nlm.nih.gov which is a Government website. The Partitional algorithms compare the similarity between the genes and form clusters. Then find the time taken for clustering and find effective cluster using the above methods. Using the efficiency in time we conclude that which method is best for finding effective cluster.**

***Index Terms* --- X- mean, K-mean, Partitional algorithm, cluster analysis**

I. INTRODUCTION

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem.

II. SYSTEM ARCHITECTURE

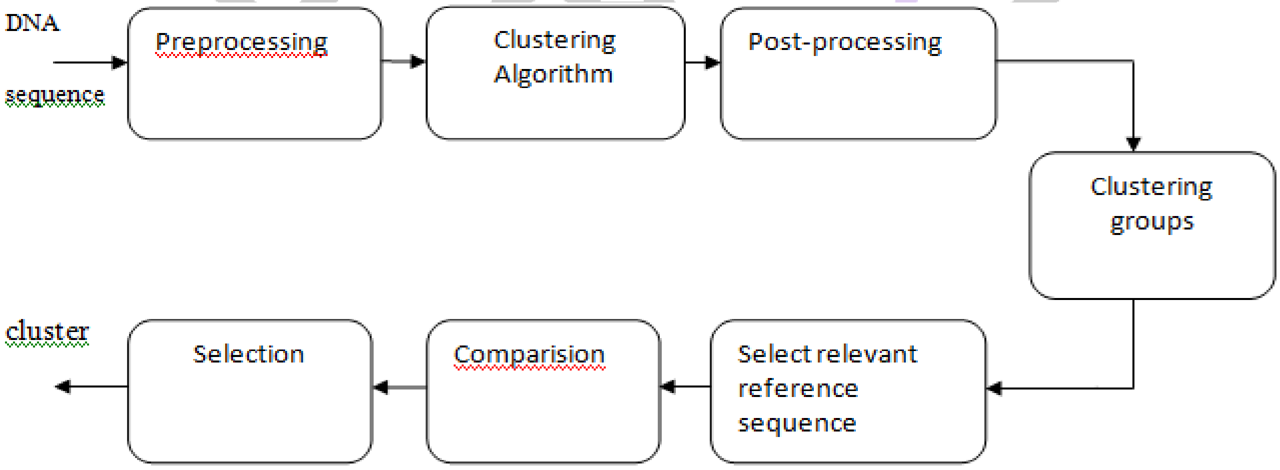


Fig. 1 System architecture

Pre-processing: As shown in fig. 1 Data pre-processing describes any type of processing performed on raw data to prepare it for *another* processing procedure. Commonly used as a preliminary data mining practice, data pre-processing transforms the data into a format that will be more easily and effectively processed for the purpose of the user. There are a number of different tools and methods used for pre-processing, including: *sampling*, *transformation*, *denoising*, normalization, feature *extraction etc*.

Clustering Algorithm: Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). Here K-means and different Partitional algorithms are applied.

Post-processing: After the data analysis, the user can perform some post-processing on the results data. This post-processing allows the analysis of the previous result sets in a formal way. At the moment, only one post-processing algorithm, Meta K-means clustering, is included in the toolbox, but more are due to be added later. To use this feature, one first has to have a collection of result parameters that require analysis.

Clustering groups: Cluster is a group of objects that belongs to the same class. In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster. A cluster of data objects can be treated as one group. While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups. The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

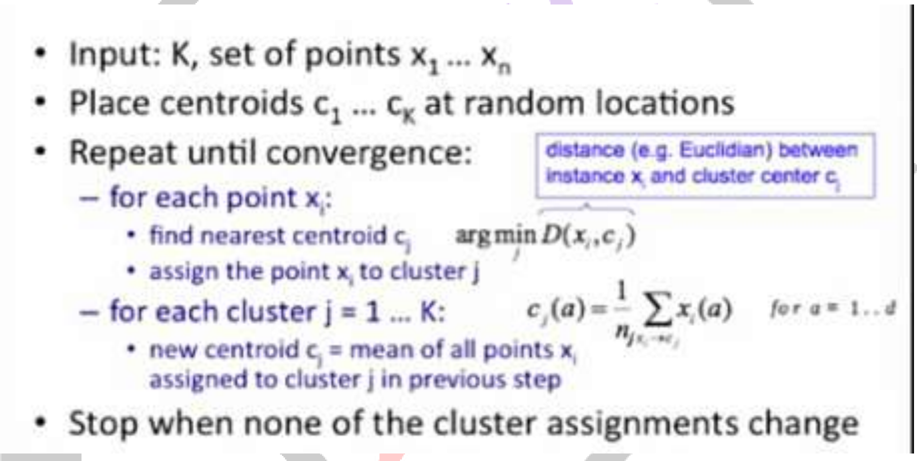
Select relevant reference sequence: While there exist many algorithms for clustering, the important issue of feature selection, that is, what attributes of the data should be used by the clustering algorithms, is rarely touched upon. Feature selection for clustering is difficult because, unlike in supervised learning, there are no class labels for the data and, thus, no obvious criteria to guide the search. Due to the introduction of a minimum message length model selection criterion, the saliency of irrelevant features is driven toward zero, which corresponds to performing feature selection. The criterion and algorithm are then extended to simultaneously estimate the feature saliencies and the number of clusters.

Comparison: The main goal is to identify the comparison of the performance of criterion function in the context of partition clustering approach, k means, and agglomerative hierarchical approach. By comparing all this we establish right clustering algorithm to produce qualitative clustering of real world document. And also modify existing algorithm to establish right algorithm which we try to make more efficient than existing algorithms.

Selection: The purpose of this module is to select efficient Partitional algorithm to produce required number of clusters. This process terminates by producing the required number of clusters.

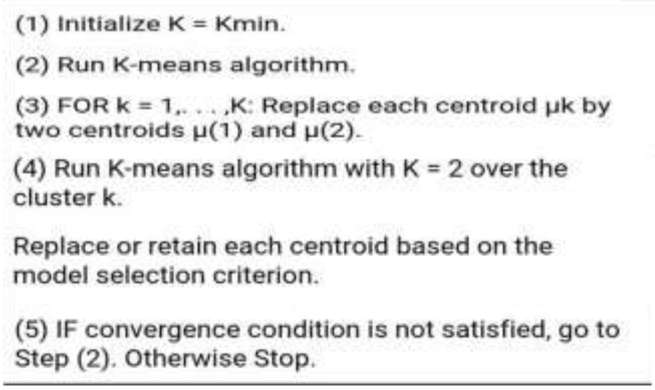
III. CLUSTERING ALGORITHMS

A. K-MEAN:



```
• Input: K, set of points $x_1 … x_n$
• Place centroids $c_1 … c_K$ at random locations
• Repeat until convergence:
    – for each point $x_i$:
        • find nearest centroid $c_j$        $\arg\min_j \overbrace{D(x_i, c_j)}^{\text{distance (e.g. Euclidian) between instance } x_i \text{ and cluster center } c_j}$
        • assign the point $x_i$ to cluster j
    – for each cluster j = 1 … K:        $c_j(a) = \frac{1}{n_j} \sum_{x_i \to c_j} x_i(a)$   for a = 1..d
        • new centroid $c_j$ = mean of all points $x_i$
          assigned to cluster j in previous step
• Stop when none of the cluster assignments change
```

B. X-MEAN:



(1) Initialize K = Kmin.

(2) Run K-means algorithm.

(3) FOR k = 1,. . . ,K: Replace each centroid μk by two centroids μ(1) and μ(2).

(4) Run K-means algorithm with K = 2 over the cluster k.

Replace or retain each centroid based on the model selection criterion.

(5) IF convergence condition is not satisfied, go to Step (2). Otherwise Stop.

.

IV. EXPERIMENTAL RESULTS

The following are the experimental results for the given input file by using different algorithms.

A. Original Image

From the below Fig. 2 shows the original patterns of the given sequence.Here using the Ruspini dataset for performing the cluster analysis

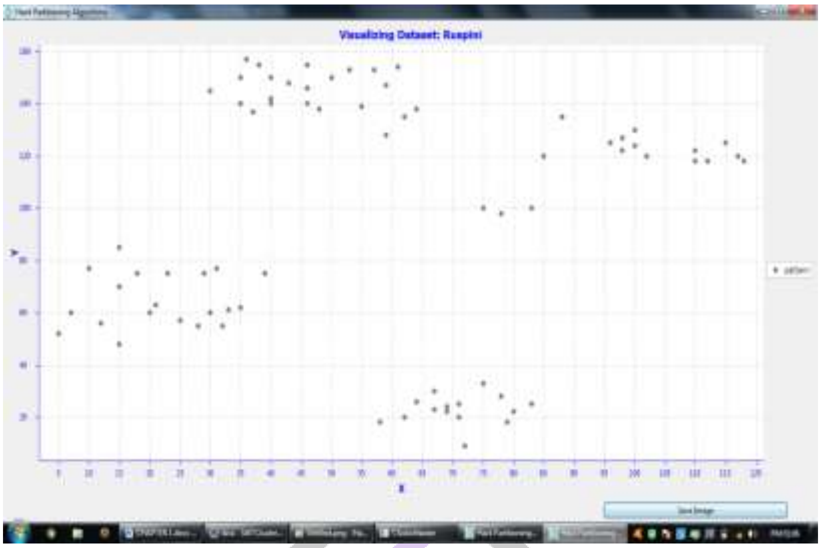


Fig. 2 Pattern display for ruspini input

B. K-mean algorithm

If K is the desired number of clusters, then Partitional approaches typically find all K clusters at once.

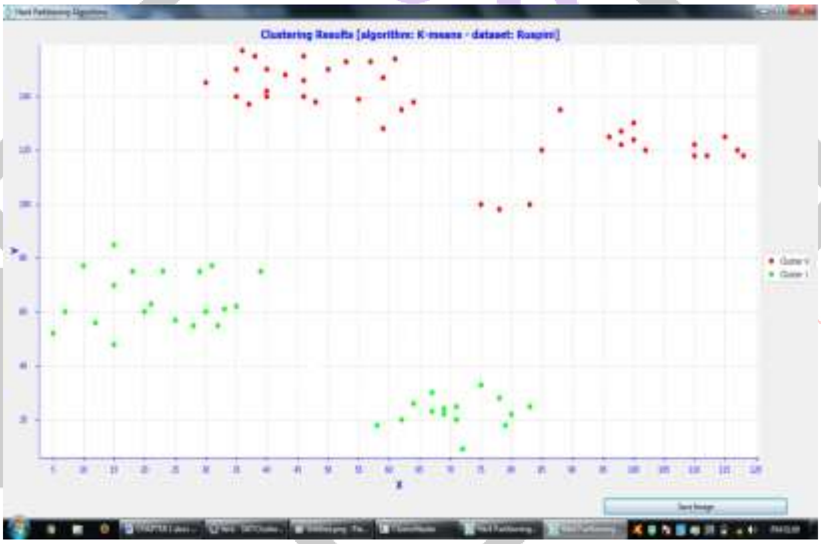


Fig. 3 Clustering results for K-means algorithm

C. Partitional algorithms

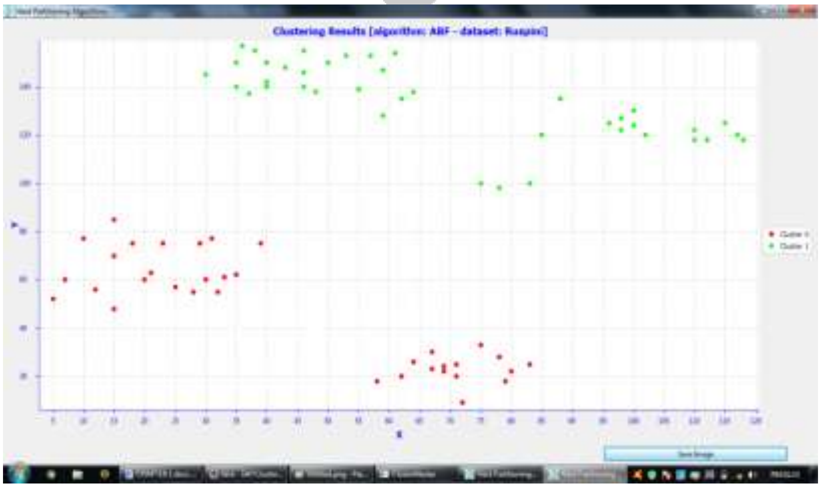


Fig. 4 Clustering results for ABF algorithm

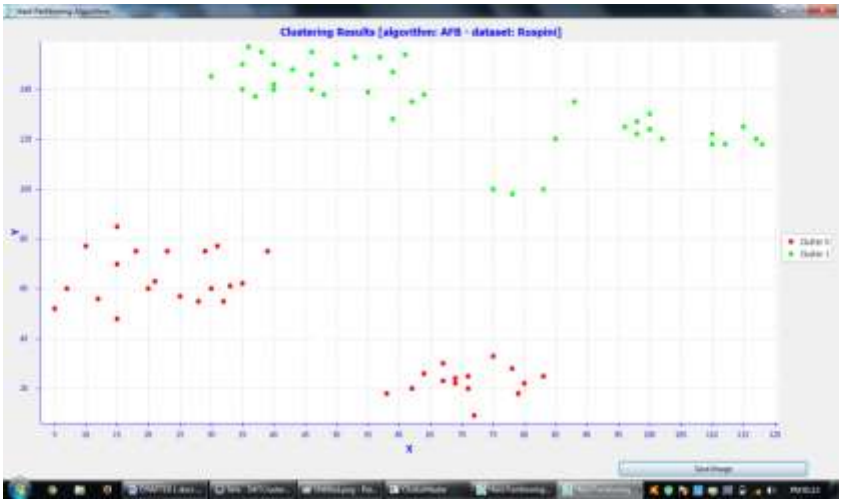


Fig. 5 Clustering results for AFB algorithm

Fig. 6 Cluster memberships for given k and m values for X-mean

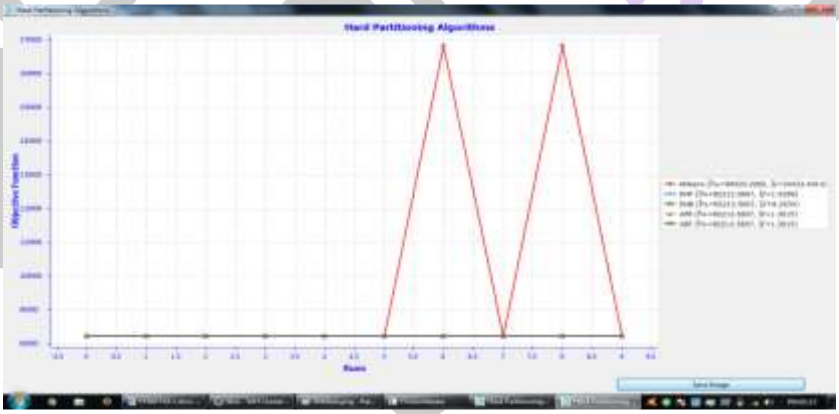


Fig. 7 comparing efficiency among different algorithms

V CONCLUSION

The K-Means and X-Means algorithms are one of the important clustering algorithms in data mining domain. Researchers have explored the usage of these two algorithms and reported satisfactory results in the literature. Generally, the time taken will vary from processor to processor. The performance of the K-Means and X-Means algorithm for both normal and uniform distributions depends on input provided. X-Means produces close results to K-means clustering, yet it requires more computation time than K-means because of the X-Means measures calculations involved in the algorithm. Thus for the data points generated using statistical distributions, the K-Means algorithm seems to be superior to X-Means.

REFERENCES

[1] Neelambike S, NasreenTaj M.B, Amrutha Sheeli, Asma UL Husna, Deepika J.B, "Effective Performance Evaluation of Cluster Analysis", IJARCST, Vol. 2 Issue 1, ISSN : 2347 – 8446. pp. 82-85, 2014.

[2] NeelambikeS , "Effective generation of clusters for gene data" vol II ,ISSUE VIII,ISSN: 2320-0790, 2013.

[3] Y. Shi and M. Mizumoto, An improvement of neuro-fuzzylearning algorithm for tuning fuzzy rules, Fuzzy Sets andSystems, vol.118, no.2, pp.339-350, 2012.

[4] L. O. Hall and I. B. Ozyurt, Clustering with a genetically optimized approach, IEEE Trans, vol.7, no.3, pp.103-112, 2010.

[5] J. Li, X. Gao and L. Jiao, A new feature weighted fuzz clustering algorithm, Acta Electronic Sinica, vol.34, no.1, pp.89-92, 2009.

[6] Y. Lu and X. Fan, Fuzzy weighting distance and its rationality discussing, Journal of Northern Transportation University, Beijing, 2007.

[7] F. Abascal and A. Valencia, “Clustering of proximal sequence space for the identification of protein families,” Bioinformatics, vol. 18, pp. 908–921, 2002.

[8]] C. Aggarwal and P. Yu, “Redefining clustering for high-dimensional applications,” IEEE Trans. Knowl. Data Eng., vol. 14, no. 2, pp. 210–225, Feb. 2002.