Effective Performance Evaluation of Cluster Analysis

[I]Neelambike S, [II]NasreenTaj M.B, [III]Amrutha Sheeli, [IV]Asma UL Husna, [V]Deepika J.B

[I, II]Asst. Professor Dept. of ISE GMIT Davangere

[III, IV, V]Research scholar GMIT Davangere

Abstract

In this paper we describes the cluster analysis for gene data and also compare the witch algorithm produces the effective cluster with the minimum amount of time and matches maximum reference sequence. For cluster analysis we have following methods such as partitioning algorithm, fuzzy k-mean and soft fuzzy algorithm and hierarchical algorithm select the DNA sequence and corresponding reference sequence from the NCBI portal then converted into numerical text file then give input to above algorithms then track the time taken to produce the cluster and also track the how much cluster is matched with the corresponding reference sequence that will arrange in the rank based so programmer easily pick the suitable algorithm to implement the their work so, it reduces the prior work of project development.

Keywords

Partitioning algorithm, fuzzy k-mean, soft k-mean, hierarchical algorithm, cluster analysis

I. Introduction

Cluster analysis divides data into meaningful or useful groups (clusters). If meaningful clusters are the goal, then the resulting clusters should capture the “natural” structure of the data. For example, cluster analysis has been used to group related documents for browsing, to find genes and proteins that have similar functionality, and to provide a grouping of spatial locations prone to earthquakes. However, in other cases, cluster analysis is only a useful starting point for other purposes, e.g., data compression or efficiently finding the nearest neighbors of points. Whether for understanding or utility, cluster analysis has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining.

Cluster analysis groups objects (observations, events) based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group will be similar (or related) to one other and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group, and the greater the difference between groups, the “better” or more distinct the clustering.

Thus, a set of objects is represented (at least conceptually) as an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute. This matrix has different names, e.g., pattern matrix or data matrix, depending on the particular field. c

All cluster analysis algorithms are worked based on the 1. data matrix 2. Proximity Matrix

Data Matrix

Objects (samples, measurements, patterns, events) are usually represented as points (vectors) in a multi-dimensional space, where each dimension represents a distinct attribute (variable, measurement) describing the object. For simplicity, it is normally assumed that values are present for all attributes.

$$x'_{ij} = \frac{x_{ij}}{\max|x_{ij}|}$$

divide each attribute value of an object by the maximum observed absolute value of that attribute. This restricts all attribute values to lie between -1 and 1. Often all values are positive. And thus, all transformed values lie between 0 and 1.

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

for each attribute value subtract off the mean. μ_j ,of that attribute and then divide by the attribute’s standard deviation σ_j . if the data are normally distributed. Then most attribute values will lie between -1 and 1. $x'_{ij} = \frac{xij - \mu j}{\sigma_j^a}$

for each attribute value subtracts off the mean μ_j of that attribute and divides by the attribute’s absolute deviation σ_j . Typically, most attribute values will lie between -1 and 1. $\sigma_j^a = \frac{1}{m}\sum_{i=1}^{m}|xij - \mu j|$ (is the absolute standard deviation of the j^th^ feature).

A.Proximity Matrix

While cluster analysis sometimes uses the original data matrix, many clustering algorithms use a similarity matrix, S, or a dissimilarity matrix, D. For convenience, both matrices are commonly referred to as a proximity matrix, P. A proximity matrix, P, is an *m* by *m* matrix containing all the pairwise dissimilarities or similarities between the objects being considered.

B. Proximity Types and Scales

The attributes of the objects (or their pairwise similarities and dissimilarities) can be of different data types and can be measured on different data scales. The different types of attributes are

1. Binary (two values)
2. Discrete (a finite number of values)
3. Continuous (an effectively infinite number of values)

The different data scales are

(a) Qualitative

- Nominal – the values are just different names.
- Ordinal – the values reflect an ordering, nothing more.

(b) Quantitative

- Interval – the difference between values is meaningful, i.e., a unit of measurement exits.
- Ratio – the scale has an absolute zero so that ratios are meaningful.

C. Common Proximity Measures

Distance Measures: The most commonly used proximity measure, at least for ratio scales (scales with an absolute 0) is the Minkowski

metric, which is a generalization of the normal distance between points in Euclidean space. It is defined as $p_{ij} = \left(\sum_{k=1}^{d} |xik - xjk|^{r} \right)^{1/r}$
1) r= 1. City block (Manhattan, taxicab, L1 norm) distance. A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors.
2) r = 2. Euclidean distance.The most common measure of the distance between two points.
3) $r \rightarrow \infty$. "supremum" (Lmax norm, L∞ norm) distance. This is the maximumdifference between any component of the vectors..
Ordinal Measures: Another common type of proximity measure is derived by ranking the distances between pairs of points from 1 to m * (m - 1) / 2.

Types of Clustering

Many different clustering techniques that have been proposed over the years. These techniques can be described using the following criteria [1]:

- Hierarchical vs. partitional (nested and unnested).
- Divisive vs. agglomerative
- Incremental or non-incremental.

Hierarchical vs. partitional (Nested and Unnested).

Hierarchical techniques produce a nested sequence of partitions, with a single, all inclusive cluster at the top 14 and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining (splitting) two clusters from the next lower (next higher) level.

Divisive vs. Agglomerative.

Hierarchical clustering techniques proceed either from the top to the bottom or from the bottom to the top, i.e., a technique starts with one large cluster and splits it, or starts with clusters each containing a point, and then merges them.

Incremental or non-incremental.

Some clustering techniques work with an item at a time and decide how to cluster it given the current set of points that have already been processed. Other techniques use information about all the points at once. Nonincremental clustering algorithms are far more common [6].

II. Clustering Techniques

As described earlier, partitional clustering techniques create a one-level partitioning of the data points. There are a number of such techniques, but we shall only describe two approaches in this K-means and K-medoid

A. K-means Clustering

The K-means clustering technique is very simple and we immediately begin with a description of the basic algorithm. We elaborate in the following.
Basic K-means Algorithm for finding K clusters.

1. Select K points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change.

In the absence of numerical problems, this procedure always converges to a solution, although the solution is typically a local minimum.

B. Hierarchical Clustering

In hierarchical clustering the goal is to produce a hierarchical series of nested clusters, ranging from clusters of individual points at the bottom to an all-inclusive cluster at the top. A diagram called a dendogram graphically represents this hierarchy and is an inverted tree that describes the order in which points are merged (bottom-up view) or clusters are split (top-down view).

C. Density Based Clustering

CLIQUE and MAFIA are also density based clustering schemes, but because they are specifically designed for handling clusters in high-dimensional data.

III. Experimental Results

The following are the experimental results for cluster analysis by using different algorithms.

A. Original Image

From the below Fig 1 shows the original patterns of the given sequence.
Here using the Ruspini dataset for performing the cluster analysis and taken the corresponding reference sequence for matching the given sequences.

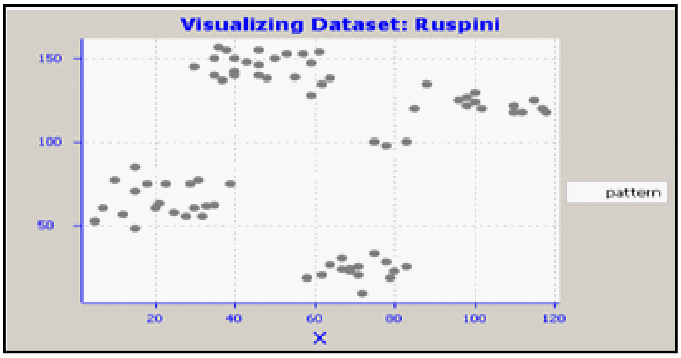

Fig 1: original patterns of Ruspini sequence

The original patterns are distributed as shows above fig1.

B. Partition algorithm using K-mean

Partitional techniques create a one-level (unnested) partitioning of the data points. If K is the desired number of clusters, then partitional approaches typically find all K clusters at once. Contrast this with traditional hierarchical schemes, which bisect a cluster to get two clusters or merge two clusters to get one. Of course, a hierarchical approach can be usedto generate a flat partition of K clusters, and likewise, the repeated application of apartitional scheme can provide a hierarchical clustering.

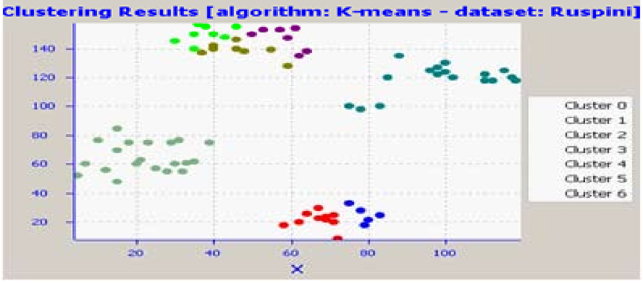

Fig 2: Cluster analysis by using the K-mean

From the above Fig 2 produces the result of the partitional K-mean algorithm for Ruspini dataset it produces the seven diffierent clusters those clusters are indicated by different color as shown above fig 2.

C. Partition algorithm using DHB

The below fig 3 shows the cluster analysis for Ruspini dataset by using partitional DHB algorithm each cluster is identified by the different color .

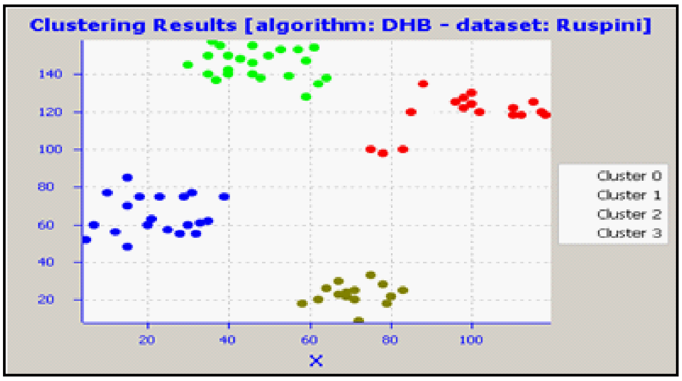

Fig 3: cluster analysis by using the DHB

DHB is mainly worked based on the hierarchical density

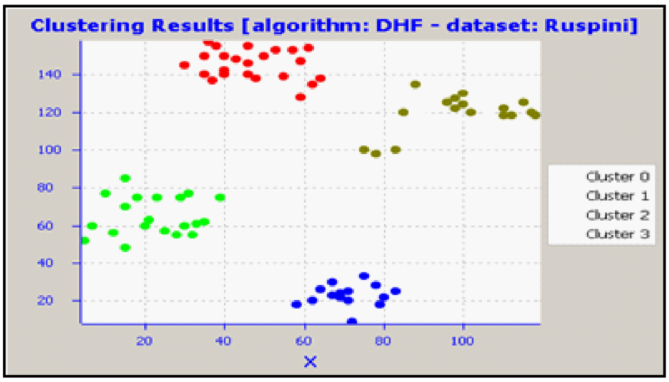

Fig 4: cluster analysis by using the DHF

D. Partition algorithm comparison

The below fig 5 shows the time taken to produce a cluster with all partitioning algorithms.

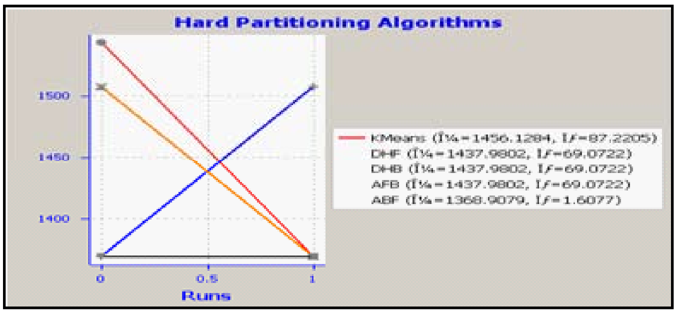

Fig 5: Comparison of hard partitioning

E. Fuzzy K-mean

ThFrom the below fig 6 find the which elements are common for two or more clusters.

Fig 6: Fuzzy K-mean

F. Soft K-mean

Fig 7: Soft Fuzzy K-mean

G. Hierarchical Algorithn

The below fig 8 shows the dendrogram representation of the given input DNA sequence.

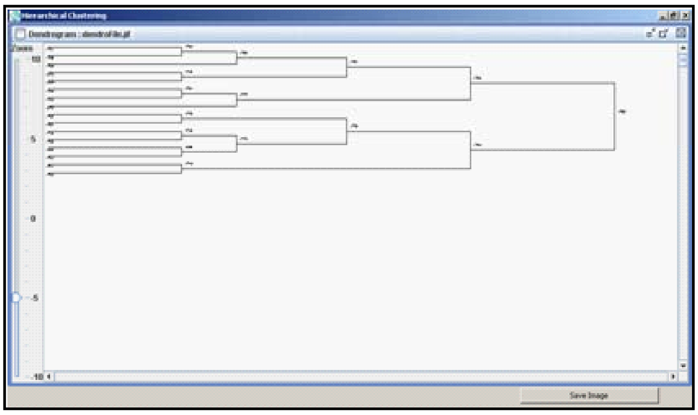

Fig 8: Hierarchical cluster result

From this hierarchical approach first it merges the minimum distance between the data points similarly it done for all the data points in the given sequence finally got the dendogram representation as show above fig 8.

H. Comparison of cluster analysis algorithm

Below fig 9 shows the comparison of all clustering algorithms such as all partional algorithms, Fuzzy K mean soft K-mean algorithms and hierarchical clustering algorithms . here it displays bar charts based on the time constraint and the how much of reference sequence is matched with the reference sequence these reference sequence are taken from the NCBI web site based on this chart programmer or implementer can select their which algorithm is suite their project so they easily save their time for selecting the algorithms for implementation.

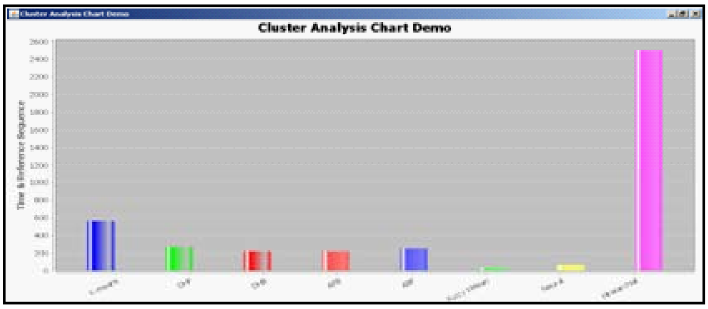

Fig 9: Comparison of all cluster analysis

References

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

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

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

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

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

[6] Chu, S., Derisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O., and Herskowitz, I., Thetranscriptional program of sporulation in budding yeast, Science, 282:699–705, 2003.

[7] Dembele, D. and Kastner, P., Fuzzy C-means method for clustering microarray data, Bioinformatics, 19:973–980, 2002.

[8] Gasch, A. and Eisen, M., Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering, Genome Biology, 3(11):research0059.1–research0059.22, 2002.

[9] Kupiec, M., Ayers, B., Esposito, R.E., and Mitchell, A.P., The molecular and cellular biology of the yeast Saccaromyces, Cold Spring Harbor, 889–1036, 2001.

[10] Tomida, S., Hanai, T., Honda, H., and Kobayashi, T., Gene expression analysis using Fuzzy ART,Genome Informatics, 12:245–246, 2000.

[11] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York, 2000

[12] http://www.ncbi.nlm.nih.gov