IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. III (Jan. 2014), PP 22-26
www.iosrjournals.org

Comparison Between Clustering Algorithms for Microarray Data Analysis

Makhfudzah bt. Mokhtar[1],Ahmed Abbas Abdulwahhab[1],Siti Mariam Shafie[1]
[1](*Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia, Malaysia)*

Abstract: *Currently, there are two techniques used for large-scale gene-expression profiling; microarray and RNA-Sequence (RNA-Seq).This paper is intended to study and compare different clustering algorithms that used in microarray data analysis. Microarray is a DNA molecules array which allows multiple hybridization experiments to be carried out simultaneously and trace expression levels of thousands of genes. It is a high-throughput technology for gene expression analysis and becomes an effective tool for biomedical research. Microarray analysis aims to interpret the data produced from experiments on DNA, RNA, and protein microarrays, which enable researchers to investigate the expression state of a large number of genes. Data clustering represents the first and main process in microarray data analysis. The k-means, fuzzy c-mean, self-organizing map, and hierarchical clustering algorithms are under investigation in this paper. These algorithms are compared based on their clustering model.*

Keywords: *Microarray, Clustering analysis, k-means algorithm, Fuzzy c-mean algorithm, Self-organizing map algorithm, Hierarchical algorithm.*

I. Introduction

The technology of DNA microarray has risen to be proved as an essential instrument for studying gene expression. The need for massive analytical instruments is crucial and the data accumulated from this technology is able to measure the relative abundance of mRNA of thousands of genes through tens or hundreds samples. Clustering is a practical exploratory method for analyzing these data because of the huge number of genes and complex gene control networks. Clustering or clustering analysis is defined as the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. The resulted groups will be meaningful or useful, or both. Clustering can be done in some sense or another based on required parameters according to the field where it is used. It represents the first and main process in many fields; for instances, data mining, machine learning, statistical data analysis, and bioinformatics [1]. The method or algorithm to do clustering analysis is not limited to one approach. It can be accomplished by various algorithms that significantly differ in their concept in terms of factors that constitutes a cluster and their efficiency. General concept of clustering comprises groups with small distance among the cluster's members, areas with high density in its space, and particular statistical distribution. The suitable algorithm for clustering and parameters settings which include distance function, density threshold, or expected number of clusters depend on the individual dataset and intended use of the results. One of the reasons that stand behind why there are so many clustering algorithms is that the notion of "cluster" cannot be precisely defined. Thus, different researchers employ different cluster models, and again for each of them, different algorithms can be exploited. As found by different algorithms, the concept of cluster varies in its characteristics. In order to understand the differences between various algorithms for microarray data analysis, clustering models should be understood. Typical clustering model include:

(i) Centroid model: Each cluster is represented by a single mean vector. Among the examples are k-means and fuzzy c-mean algorithms.
(ii) Connectivity model: Models are built according to distance[2] connectivity. Among the examples are self-organizing map (SOM) and hierarchical clustering.

II. Clustering Algorithms

Various clustering algorithms have been used for analyzing gene expression data gathered by microarray. For a given gene expression dataset, several indices should be applied to evaluate the clustering algorithm performance. Clustering algorithms can be categorized based on their cluster model. In the next review, only the most prominent examples of these algorithms will be mentioned as there are also other published clustering

algorithms. Objectively, there is no perfect clustering algorithm, but it can be considered as the observer's eye. Unless if there are mathematical reasons for preferring one cluster model over others, the most proper algorithm for a specific issue need to be chosen experimentally.

K-means clustering

K-means is a method of vector quantization[3] and one of the simplest unsupervised learning algorithms which is used to solve familiar clustering problems. K-means algorithm follows an easy and simple way to classify a given dataset via a specific number of clusters (assume *k* clusters). Defining *k* centroids[4] (one for each cluster) is the main idea for this algorithm. As different location causes different result, these centroids should be placed in a cunning way. Thus, placing these centroids far away from each other as much as possible is the best choice. Next step is to take each point under a given dataset and associate it to the nearest centroid. An early grouping is achieved when there is no pending point and recalculation of new *k* centroids are needed as a barycenters of the clusters resulting in previous step. New binding has to be done between the same dataset points and the nearest new centroid after having these *k* new centroids. Once a loop has been generated, *k* centroids alter their location gradually until no more changes are done. In other words, centroids do not move any more. Finally, this algorithm aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest centroid, serving as a prototype of the cluster as in Eq. (1).

$$\arg min \sum_{i=1}^{k} \sum_{x_j \in s_i} | x_j - \mu_i |^2 \qquad \text{Eq. (1)}$$

Where μ_i is the mean of points in space s_i.

The algorithm is composed of the following steps:

(i) *Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
(ii) *Assign each object to the group that has the closest centroid.*
(iii) *When all objects have been assigned, recalculate the positions of the k centroids.*
(iv) *Repeat steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

Fuzzy c- means clustering

Most clustering algorithms do not rely on assumptions common to conventional statistical methods, such as the underlying statistical distribution of data, and therefore they are useful in situations where little prior knowledge exists. Fuzzy clustering is a class of algorithms for cluster analysis in which the allocation of data points to clusters is not "hard" but "fuzzy" in the same sense as fuzzy logic. Unlike in hard clustering where each data element belongs to exactly one cluster, in fuzzy clustering data elements can belong to more than one cluster, and associated with each element is a set of membership levels. Fuzzy c-means (FCM) is one of the most widely used fuzzy clustering algorithms. FCM tries to divide a dataset of *N* elements,$X=\{x_1,x_2,\ldots\ldots,x_n\}$ into a group of *C* fuzzy clusters according to some given criteria. For a given dataset, the algorithm outputs a record of *C* cluster centers,$C=\{c_1,c_2,\ldots\ldots,c_c\}$ and a partition matrix $W=w_{i,j}\in[0,1],\ i=1,2,\ldots.,n,\ j=1,2,\ldots,c$, where each element $w_{i,j}$ tells the degree to which element x_i belongs to cluster c_j. As in k-means, FCM aims to minimize an objective function. The standard function is:

$$W_k(x) = \frac{1}{\sum_j \left(\frac{d(center\ k,x)}{d(center\ j,x)}\right)^{2/(m-1)}} \qquad \text{Eq. (2)}$$

It differs from the k-means objective function due to the addition of the membership values, u_{ij} and the fuzzifier, *m*. Larger *m* leads to smaller memberships (w_{ij}) and hence, fuzzier clusters. Every point in fuzzy clustering has a degree of belonging to clusters rather than belonging completely just one cluster. Thus, points on the edge of a cluster may be in the cluster to a lesser degree than points in the center of cluster.

The FCM algorithm is quite similar to k-means algorithm and composed of following steps:

(i) Chose the number of clusters.
(ii) Assign randomly to each point coefficients for being in the clusters.
(iii) Repeat until the algorithm has converged (change between two iterations is no more than the given sensitivity threshold).

(iv) Compute the centroid for each cluster, using the formula in Eq.(3)

$$C_k = \frac{\sum_x W_k(x)^m \, x}{\sum_x W_k(x)^m} \qquad \text{Eq. (3)}$$

(v) For each point, compute its coefficients of being in the clusters, using the formula above.

Although algorithm minimizes intra-cluster variance as well, it has the same problems as in k-means where the results depend on the initial choice of weights.

Self-Organizing-Map Clustering

Self-organizing-map, SOM is a kind of artificial neural network that is trained using unsupervised learning[5] to construct a low-dimensional (typically two-dimensional)representation of the input space of the training samples, called a map. Neighbourhood function is used to preserve the topological properties of the input space and it is the vital difference between self organizing maps and the other neural networks. Close to multidimensional scaling[6],SOMs are useful for visualizing low-dimensional views of high-dimensional data. SOMs operate in two modes, same as the most artificial neural networks operate, training and mapping. The first mode, training, constructs the map using input examples which is a competitive process[7],while the mapping automatically classifies a new input vector. Nodes or neurons represent the main content of SOM, weight vector with the same dimension as the input data vectors and a position in the map space is related with each node. Two dimensional regular spacing in a hexagonal or rectangular grid is the normal arrangement of neurons. The self-organizing map outlines a mapping from a higher dimensional input space to a lower dimensional map space. The process for placing a vector from data space on the map is to discover the node with the smallest distance weight vector to the data space vector. This type of architecture is mainly different in arrangement and motivation although it is typically (based on its architecture) considered as a type of feed forward networks where the nodes are visualized as being attached. SOM aims to learn an attribute map from the spatially continuous input space, in which our input vectors live, to the low dimensional spatially discrete output space, which is formed by ordering the computational neurons into a grid.

Phases of the SOM's algorithm can be summarized as follows:

(i) Initialization, select random values for the initial weight vectors wj.
(ii) Sampling, sketch a sample training input vector x from the input space.
(iii) Matching, find the winning neuron I(x) that has weight vector nearest to the input vector, i.e. the minimum value of dj(x)=$\sum_{i=1}^{D}(xi - wji)\text{^}2$.
(iv) Updating, apply the weight update equation $\Delta wij = \eta(t)Tj,l(x)\,(t)\,(xi - wij)$*whereTj,l(x) (t) is a Gaussian neighbourhood and η(t) is the learning rate.*
(v) Continuation, keep returning to step 2 until the feature map stops changing.

Hierarchical clustering

Hierarchical clustering is defined as a process of cluster analysis which searches for building a hierarchy of clusters. In general, there are two major types of hierarchical clustering strategies; agglomeration and divisive. The approach in agglomeration strategy is bottom up where each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy while the approach in divisive strategy is top down where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. Commonly, combining and separating are regulated based on Greedy algorithm[8], results are normally presented by dendrogram[9].

In agglomeration clustering approach, each point in the data is considered as a cluster. After that, the closest clusters are merged in repetition manner to construct larger and fewer clusters. Finally, with continuous execution, only one cluster will be resulted. Of course, stopping criterion can be introduced when the appropriate number of clusters or maximum cluster size is reached.

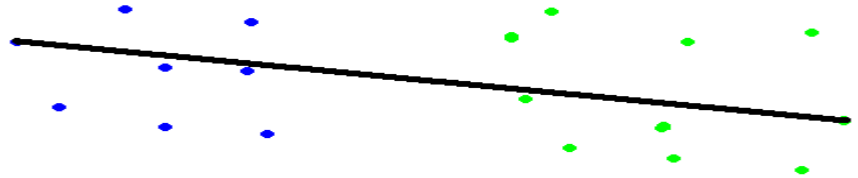
In divisive clustering approach, the entire dataset is assumed as a one cluster. Then, data is separated repeatedly. Eventually, the result ends up with *S* clusters where each point is itself a cluster. In brief, this approach includes calculating a boundary weight defined as the distance between two points within the same cluster and an edge cut that is calculated for all possible partitions such we proceed to divide the cluster for the maximum boundary cut.

III. Closeness Measurement

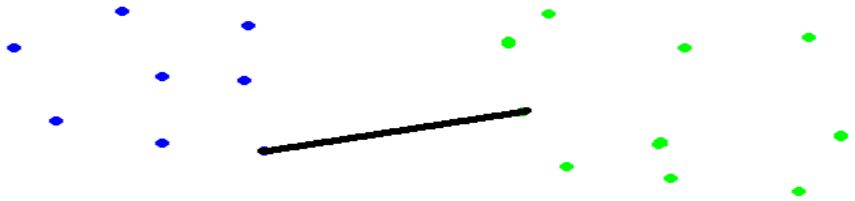
It is important to know that there are three mostly known ways to measure the closeness of two clusters; this importance comes from which strategy should be chosen . These three methods can be explained as follow:

(i) The average-linkage clustering: The distance between the mean of each cluster represents the distance between two clusters as shown below (black line).

(ii) The complete-linkage clustering: The maximum distance between elements of each cluster represents the distance between two clusters as shown below (black line).



(iii) The single-linkage clustering: The minimum distance between elements of each cluster represents the distance between two clusters as shown below (black line).



IV. Complexity

The brute force[10] approach involves computing the distance of each cluster with respect of all the other clusters, for instance, for *N* clusters, the complexity is $O(N^2)$ at this step. Constructing new cluster by combining two clusters need to compute the distance again for this new cluster with respect to all other clusters such that the complexity for this step is O(N). Therefore, the total complexity is $O(N^3)$. Generally, agglomeration clustering has the complexity of $O(N^3)$, which makes them too slow for huge datasets while in divisive clustering with an exhaustive search is $O(N^2)$, which is even worse.

V. Discussion

One of the main benefits of cluster analysis is to track certain biological relations among the genes by clusters interpretation . Clustering results appeared to reflect these elation ; different algorithms show different properties. The k-means generates clusters with slightly better structural quality. The k-means and self-organizing map appear more consistent results with the biological information. However, k-means is relatively sensitive to noise perturbation in the data. Meanwhile, the SOM produces relatively stable clusters when neighbourhood interaction is maintained, but of relatively low structural quality. Neighbourhood constraint on SOM seems to have a double-effect; it helps to improve the clustering stability but prevents further optimization in the clustering structure. Because of the unique feature of SOM is the topographic relation between the

mapping nodes, calculation of the topographic error 'TE' can be done for the measure the topology preservation of the map units which appeared to be correlated to the performance of this algorithm. In theory, if the neighbourhood radius is set to zero, the SOM algorithm closes to k-means which is confirmed by some studies[5].The quality of clusters resulted by SOM r=0 are very similar to that of k-means, especially when evaluated with homogeneity, separation, silhouette width and redundant scores. In general, with relatively small k, SOM is more stable than k-means, i.e., as *k* increased, the resulted clusters from k-means and SOM will be very close to each other's. The hierarchical clustering is the worst among the four algorithms in this particular comparison . Greediness of the algorithm is the cause of relatively low quality of agglomerative hierarchical clustering. As an example, average linkage is not possible to do any refinement or correction after merging two similar clusters.

There is no single perfect clustering algorithm for any feasible datasets, or for all quality measures, different features and properties for different clustering algorithms. The nature of the data determines the suitability of a particular algorithm. For instance, when the data itself contain a hierarchical structure, hierarchical clustering algorithms will be more useful than partition algorithms, such as k-means which will not be able to capture this type of information. SOM has a good feature where clusters are represented by notes ranked in a topological order correlated to the similarity of the clusters. Hence, it is easy to trace relations between clusters.

VI. Conclusion

In conclusion, knowledge and practice about the conduct of clustering algorithm are required in cluster analysis where a priori knowledge about the data and biological processes can also be benefited. In case of there is no prior knowledge about the data or insufficient, it may be eligible to try different algorithms to explore the data and obtain meaningful clustering results through comparisons in addition to that caution is required, as different algorithms tend to result somewhat different clusters.

1- *Bioinformatics is an interdisciplinary field that developed and improves on methods for storing, retrieving, organizing and analyzing biological data.*

2- *mathematically, a distance function is a function that defines a distance between elements of a set.*

3- *Is a classical quantization technique which allows the modeling of probability density functions by the distribution of prototype vectors.*

4- *The Centroid can be said as the centre of mass of the object of uniform density, in multivariate vector, Centroid is used instead of mean.*

5 *-In machine learning, the problem of unsupervised learning is that of trying to find hidden structure in unlabeled data.*

6- *Multidimensional scaling is a means of visualizing the level of similarity of individual cases of a dataset.*

7- *Competitive learning is a shape of unsupervised learning in artificial neural networks, where the nodes compete for the right to respond to a subset of the input data.*

8- *A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage.*

9- *A dendrogram (Greek word "dendron") is a tree diagram frequently used to explicate the arrangement of the clusters produced by hierarchical clustering.*

10- *Brute-force search or exhaustive search, also known as generate and test, is a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.*

References

[1] Abu Abbas O. "Comparisons Between Data Clustering Algorithms", The International Arab Journal of Information Technology, Vol.5. No. 3, July 2008.

[2] Song M., Wang H.," Detecting Low Complexity Clusters by Skewness and Kurtosis in Data Stream Clustering" , Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics. Florida: Proceedings of AIM, 2006. pp,1-8.

[3] Smyth G.K. "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments", Statistical Applications in Genetics and Molecular Biology, 3, No.1, article 3, 2004.

[4] Venet D., " Matarray: a Matlab toolbox for microarray data", "Bioinformatics Application Notes", Vol.19, No. 5, pp.659-660, 2003.

[5] Chen G., Jaradat S., Banerjeen N., Tanaka T., Ko M., and Zhang M., "Evaluation and Comparison of Clustering Algorithms in Analyzing ES cell Gene Expression Data, "Statistica Sinica", Vol. 12, pp 241-262, 2002.

[6] Yeung K.Y., Haynor D.R., and Ruzzo W.L.,"Validating Clustering for gene expression Data", Bioinformatics, Vol. 17, No.4, pp 309-318, 2001.

[7] Keogh E., Chakabarti K., Pazzania M., and Mehrotra S., "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases, "Knowledge and Information Systems", Vol. 3, pp. 263-286, 2001.

[8] Benjamini B.Y. and Yekutieli D., "The Control of the False Discovery rate in multiple testing under dependency", The Annals of Statistics, p 1165-1188, 2001.