Efficient Bisecting *k*-Medoids and Its Application in Gene Expression Analysis

Rasha Kashef and Mohamed S. Kamel

University of Waterloo
Electrical and Computer Engineering Department
{rkashef,mkamel}@pami.uwaterloo.ca

Abstract. The medoid-based clustering algorithm, Partition Around Medoids (PAM), is better than the centroid-based *k*-means because of its robustness to noisy data and outliers. PAM cannot recognize relatively small clusters in situations where good partitions around medoids clearly exist. Also PAM needs $O(k(n-k)^2)$ operations to cluster a given dataset, which is computationally prohibited for large n and k. In this paper, we propose a new bisecting *k*-medoids algorithm that is capable of grouping the co-expressed genes together with better clustering quality and time performances. The proposed algorithm is evaluated over three gene expression datasets in which noise components are involved. The proposed algorithm takes less computation time with comparable performance relative to the Partitioning Around Medoids algorithm.

Keywords: Clustering, Bisecting Clustering, and Quality Measures.

1 Introduction

Clustering is the process of finding a *structure* in a collection of unlabeled objects (in gene expression clustering, objects are usually genes and the vector of each gene is its expression pattern). *Co-expressed* genes in the same cluster are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates co-regulation. Clustering techniques have been proven to be helpful to understand gene function, gene regulation, cellular processes, and subtypes of cells. A variety of clustering algorithms has been proposed in the literature to analyze the gene-expression profiles [1]-[13]. Centroid-based clustering approaches such as *k*-means (KM) [6], fuzzy c-means (FCM) [7], and Bisecting *k*-means (BKM) [8] try to assign objects to clusters such that the sum of square distances of objects to the centroids of the assigned cluster is minimized. PAM [13] (Partitioning Around Medoids) algorithm is an iterative medoid-based clustering technique, which operates on a distance Matrix. For a pre-specified number of clusters *k*, PAM searches for *k* medoids $m_1, m_2, \ldots, m_k$, among the set of objects that minimize the sum of the distances of the objects to their closest medoid. PAM is more robust than *k*-means in the presence of noise and outliers because medoid is less influence by outliers and other extreme values than a mean. In addition, PAM does not require random seeding as *k*-means clustering does, or user-derived thresholds typically required by hierarchical clustering [14]. Also it can be applied to datasets of different types (e.g. gene

A. Campilho and M. Kamel (Eds.): ICIAR 2008, LNCS 5112, pp. 423–434, 2008.
© Springer-Verlag Berlin Heidelberg 2008

expression datasets). However, PAM is inefficient for large and high dimensional data sets due to its complexity and cannot recognize relatively small clusters in situations where good partitions around medoids clearly exist. In this paper, a new divisive Bisecting *k*-medoids (BPAM) clustering method is presented. BPAM recursively applies the basic *k*-medoids method. The main idea is based on a selection of a cluster to split at each level of the hierarchical tree generated by the application of PAM using a homogeneity criterion. The bisecting *k*-medoids technique forms more uniformly sized clusters than the PAM algorithm. Experimental results show that BPAM provides better clustering quality and time performances than the standard PAM algorithm.

The rest of this paper is organized as follows: in section 2, the *k*-medoids algorithm is given. The proposed BPAM is presented in section 3. Complexity analysis of the adopted *k*-medoids and the proposed algorithm is presented in section 4. Section 5 provides external and internal quality measures used to assess the clustering quality. Experimental results are presented and discussed in section 6. Finally, we discuss some conclusions and outline future work in section 7.

2 *k*-Medoids (PAM) Clustering Algorithm

Rather than calculating the mean of the objects in each cluster as in the *k*-means (KM) clustering algorithm, the Partition Around Medoids (PAM) algorithm [13] chooses a representative object, or medoid, for each cluster *at each iteration*. Medoids for each cluster are calculated by finding an object m_i within the cluster that minimizes the objective function defined as the sum of distances of all objects within the clusters to the cluster medoid. The PAM algorithm has the advantage of its robustness to noisy data and outliers compared to *k*-means. PAM works well for small datasets but cannot scale for large datasets. Thus, [13] also present *clustering large applications* (CLARA), which draws one or more random samples from the whole data setand runs PAM on the samples. Ng and Han [15] propose *clustering large applications based on randomized search* (CLARANS) as an extension to PAM. Although CLARA and CLARANS are more scalable than PAM, they are inefficient for disk-resident datasets as they require multiple scans of the entire database and also a good clustering of a sample does not mean good clustering for the whole dataset.

Table 1. Symbols and Notations

Symbol	Definition
X	The entire dataset
$\boldsymbol{x}$	Object(or gene) represented as a vector of features (expression patterns)
d	Dimensionality of object $\boldsymbol{x}$
n	Number of objects
k	Number of clusters
S_j	The j^{th} cluster
$\|S_j\|$	Number of objects in the j^{th} cluster
R_j	The j^{th} class (external labeling for the data)
m_j	The medoid of cluster S_j
ζ	Splitting criterion of the BPAM algorithm

Algorithm: *k*-medoids (PAM) Clustering (X,*k*)
Input: The dataset of objects X and number of clusters *k*
Output: set of *k* clusters, S={S_1,S_2,..,S_k}
Initialization: Select *k* objects randomly as medoids (m_1,m_2,..,m_k)
Begin
  Repeat
    Step1: Assign each remaining non-medoid object to the cluster with the nearest medoid and compute *Total Distances* of cluster S_i, $TD(S_i)$, i=1,..,k, as the sum of distances from all objects to their nearest medoid m_i.
    Step2: For each pair of medoid m_i , i=1,2,..,k, and non-medoid $\boldsymbol{x}_j$, j=1,2,…,$|S_i|$,
      - Compute the value $TD(S_i)(m_i \leftrightarrow \boldsymbol{x}_j)$; i.e. the value of the compactness of cluster S_i that results when swapping m_i with $\boldsymbol{x}_j$
      - Select the non-medoid object $\boldsymbol{x} \in S_i$ for which $TD(S_i)(m_i \leftrightarrow \boldsymbol{x})$ is minimal
      - If $TD(S_i)(m_i \leftrightarrow \boldsymbol{x})$ is smaller than the current $TD(S_i)$,
        Then swap m_i with $\boldsymbol{x}$ *and s*et $TD(S_i) = TD(S_i)(m_i \leftrightarrow \boldsymbol{x})$
  Until (No Change in Medoids)
Return S
End

Fig. 1. *k*-medoids (PAM) Clustering Algorithm

Some terminologies and notations are best presented at this point to pave the way for discussion of the different concepts and strategies for data clustering and also for the proposed algorithm. Table 1 summarizes the notations and symbols that are used throughout this paper. The iterative PAM clustering algorithm is illustrated in Fig .1.

3 Bisecting *k*-Medoids (BPAM) Clustering Algorithm

In this section, a bisecting *k*-medoids algorithm (BPAM) as a variant of *k*-medoids is proposed that can produce either a partitional or a hierarchical clustering by recursively applying the basic *k*-medoids method. The BPAM also works on a distance matrix that is pre-calculated before the start of the BPAM. The main idea of the BPAM is similar to that of the centroid-based bisecting *k*-means clustering [8]. It starts by considering the whole dataset to be one cluster. At each step, one cluster is selected and bisected further into two sub-clusters using the basic *k*-medoids algorithm (*Bisecting Step*). The clustering solution of the BPAM is mainly based on the selection criterion for splitting. This criterion enables the BPAM to uniformly partition the dataset into homogenous clusters. There are a number of different criteria to choose which cluster to split. For example, we can choose:

1) The largest cluster at each step,
2) The one with the least overall similarity
3) A criterion based on both size and overall similarity.

In this paper, the *average cluster compactness* as a measure of *homogeneity* of each cluster is proposed, it is defined as:

$$Average\ Compactness(S_i) = \frac{\sum_{\forall x \in S_i} \| \boldsymbol{x} - m_i \|_2}{| S_i |} \quad (1)$$

Where $\|\boldsymbol{x}\text{-}m_i\|_2$ is the Euclidian distance between object $\boldsymbol{x}$ and the cluster medoids m_i. A small value of this measure indicates a high homogeneity (less scattering) of objects within cluster). Then, the *Bisecting step* is repeated for a number of Iterations (*ITER*) (a user-defined parameter) to assure the choice of good medoids for the two selected partitions. This clustering step continues until the desired number of clusters k or some other specified stopping condition is reached. The bisecting approach forms more uniformly sized clusters. BPAM is also robust to the existence of noise data and outliers because it selects medoids as representatives to each cluster as well. The BPAM is utilized with lower running time than the PAM algorithm as will be shown in the experimental results, as it selects only one cluster for splitting at each level of the hierarchical tree rather than the whole dataset. The proposed BPAM is shown in Fig.2.

Algorithm: Bisecting *k*-Medoids Clustering (BPAM) (X, ITER, *k*, ζ)
Input: The dataset X, Number of iterations *ITER*, the number of clusters *k,* and a homogeneity criterion ζ
Output: The set of *k* clusters S={S_1,S_2,..,S_k}
Initialization: - Let V = X , S={ }
Begin
For number of partitions *l* =2 to *k* (Clustering Step)
Step1: **For *i*=1 to *ITER* (Bisecting Step)**
-Select randomly two initial medoids m_1 and m_2 from the set V.
-Find 2 sub-clusters from the set V using the basic *k*-medoid algorithm.
End
Step2: Take the best of these splits as V_1 and V_2 with the corresponding medoids m_1 and m_2, respectively.
Step3: Select the cluster that satisfies a given homogeneity criterion ζ as V_1
Step4: Assign V to the remaining partition, V=V_2
Step5: Add V_1 to the set of desired clusters $S=S \cup V_1$
End
Add V_2 to the set of clusters $S=S \cup V_2$
Return S
End

Fig. 2. Bisecting *k*-medoids Clustering (BPAM) Algorithm

4 Complexity Analysis

The computational complexity of PAM algorithm is determined by: the number of objects (n), the number of clusters (k), and the number of iterations (*npass*). We

assume that all the basic operations have the same unit operation time. The time complexity of the PAM algorithm is:

$$T^{PAM} = O(npass * k(n-k)^2) \quad (2)$$

For the BPAM algorithm, let *ITER* be the number of iterations for each bisecting step, which is usually specified in advance. The lower and upper bounds of ITER are *log k* and *k*log(*k*) (we used different values for ITER and the best results were obtained at $log\ k \leq ITER \leq klog(k)$; proving the effectiveness of these values was only done empirically; proving it formally has not yet been accomplished, but it is planned at a future point of this paper). The computation complexity of BPAM at each level of the hierarchical tree is determined by the size of the cluster S_j at each bisecting step $|S_j|$, the number of clusters (*k*), the number of iterations of *k*-medoids in each bisecting step (*npass*), and the number of iterations for each bisecting step (*ITER)*. In the bisecting step, $O(npass*(|S_i|-2)^2)$ operations are required. The time complexity of BPAM at each level can be represented as:

$$T^{BPAM} = O(\ npass*\ ITER*\ (|S_i|-2)^2) \quad (3)$$

The best case of the BPAM complexity is occurred for datasets of uniformly equal sized-clusters. Thus, the time complexity of the BPAM is lower compared to that of the PAM algorithm, as it is of order $|S_i|^2$ at each level of the tree rather than $O(n^2)$ as in the standard PAM algorithm.

5 Quality Measures

External and internal informative quality measures are used to evaluate the clustering results of any clustering algorithm. These measures reflect the "goodness" of the resulting clusters. Three external quality measures (F-measure, Entropy, and Purity) are used, which assume that a prior knowledge about the data objects (i.e. class labels) is given [16][17]. The Separation Index (*SI*) [18] is used as an internal quality measure, which does not require prior knowledge about the labels of the objects.

5.1 F-Measure

F-measure combines the *precision* and *recall* ideas from the information retrieval literature. The precision and recall of a cluster S_j with respect to a class R_i, $i, j=1,2,..,k$, are defined as:

$$recall(R_i,S_j) = L_{ij} / |R_i|\ ,\ precision(R_i,S_j) = L_{ij} / |S_j| \quad (4)$$

Where L_{ij} is the number of objects of class R_i in cluster S_j, $|R_i|$ is the number of objects in class R_i , and $|S_j|$ is the number of objects in cluster S_j. The F-measure of a class R_i is defined as:

$$F(R_i) = \max_j \frac{2 * precision(R_i,S_j) * recall(R_i,S_j)}{precision(R_i,S_j) + recall(R_i,S_j)} \tag{5}$$

With respect to class R_i we consider the cluster with the highest F-measure to be the cluster S_j that is mapped to class R_i, and that F-measure becomes the score for class R_i. The overall F-measure for the clustering result of k clusters is the weighted average of the F-measure for each class R_i:

$$Overall\ F\text{ - }measure = \sum_{i=1}^{k} (|R_i| \times F(R_i)) / \sum_{i=1}^{k} |R_i| \tag{6}$$

The higher the overall F-measure, the better the clustering due to the higher accuracy of the resulting clusters mapping to the original classes.

5.2 Entropy

Entropy tells us how *homogenous* a cluster is. Assume a partitioning result of a clustering algorithm consisting of k clusters. For every cluster S_j we compute pr_{ij}, the probability that a member of cluster S_j belongs to class R_i. The entropy of each cluster S_j is calculated as:

$$E(S_j) = -\sum_{i=1}^{k} pr_{ij} \log(pr_{ij}), j=1,\ldots,k \tag{7}$$

The overall entropy for a set of k clusters is the sum of entropies for each cluster weighted by the size of each cluster:

$$Overall\ Entropy = \sum_{j=1}^{k} \left(\frac{|S_j|}{n} \right) \times E(S_j) \tag{8}$$

The lower the overall Entropy, the more *homogeneity* of objects within clusters.

5.3 Purity

The Purity of a clustering solution is the average precision of the clusters relative to their best matching classes. For a single cluster S_i, purity is defined as the ratio of the number of objects in the dominant cluster to the total number of objects:

$$Purity\ (S_j) = \frac{1}{|S_j|} \max_{i=1,..k,\ j \neq i} (L_{ij}) \tag{9}$$

Where L_{ij} is the number of objects from class R_i into cluster S_j, and $|S_j|$ is the number of objects in cluster S_j. To evaluate the total purity for the entire k clustering, the average of the cluster-wise purities are weighted by cluster size as:

$$Overall\ Purity = \frac{1}{|n|} \sum_{j=1}^{k} \max_{i=1,..k,\ j \neq i} (L_{ij}) \tag{10}$$

5.4 Separation Index (SI)

Separation Index is a cluster validity measure that utilizes cluster prototypes to measure the dissimilarity between clusters, as well as between objects in a cluster to their respective cluster prototypes. The smaller the SI the more separate the clusters.

$$SI(k) = \frac{\sum_{i=1}^{k} \sum_{\forall x_j \in S_i} \| x_j - z_i \|_2}{n^* \min_{r,l=1,\ldots,k, r \neq l} \{ \| z_r - z_l \|_2 \}} \tag{11}$$

For centroid-based clustering, z_i is the corresponding centroid of the cluster S_i while in medoid-based clustering, z_i is the medoid of the cluster S_i.

6 Experimental Results

The proposed BPAM has been evaluated on both time series and non-temporal gene expression datasets. The main measures of evaluation are: the external and internal quality of the output clusters and the total running time.

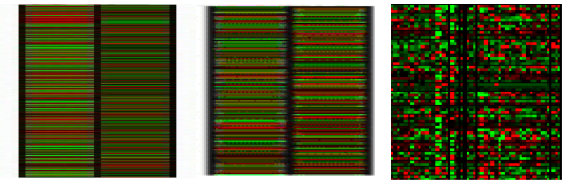
6.1 Datasets

The three data sets that are used throughout this paper are: leukemia dataset [12], yeast gene expression dataset [19], and breast cancer data set [20]. Leukemia dataset is an example of a non-temporal gene expression set. The dataset contains the expression of 999 genes along 38 samples obtained from ALL (27 samples) and AML (11 samples). The yeast cell cycle time series dataset contains the expression levels of 6,218 gene transcripts (identified as ORFs) measured at 10-minutes intervals over two cell cycles (160 minutes). The same filtering and data normalization procedures of [21] are applied resulting in a set of 703 genes. Based on the analysis conducted by Spellman et. al. [22], the five main clusters are: G1-peaking genes, S-peaking genes, G2-peaking genes, M-peaking genes, and M/G1-peaking genes. The breast cancer dataset was primary used in the work done by [20]. The dataset contains 7129 gene expression and 49 tumors. The tumor samples are given labels ER+/LN+, ER+/LN-, ER-/LN+, and ER-/LN-. The classification model for both the leukemia and the breast cancer datasets is discovered using the same approach as in [11].

Table 2. Summary of the datasets

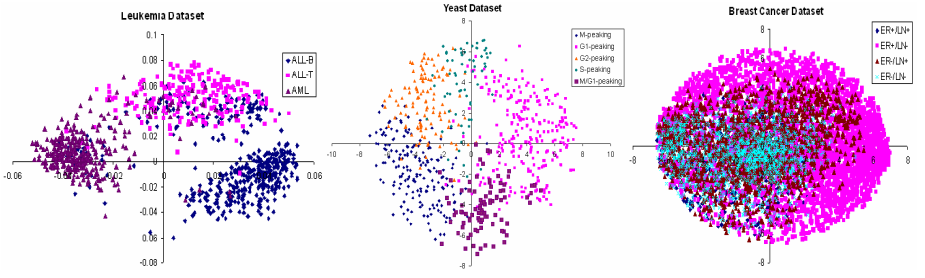
Dataset	*n*	*k*	*d*
Leukemia	999	3	38
Yeast	703	5	73
Breast Cancer	7129	4	49

The color-gradient representation and SVD (Singular Value Decomposition) modes of datasets shown in figures 3 and 4, respectively, reveal the distribution in gene expression along the whole data set.



(a) (b) (c)

Fig. 3. The color-gradient representation for (a) leukemia data set (b) Yeast data set, and (c) Breast Cancer data set



(a) (b) (c)

Fig. 4. Coefficients of SVD modes for (a) leukemia data set (b) Yeast dataset, and (c) Breast Cancer dataset

6.2 Significance of Results

In order to illustrate the significance of the obtained results, the standard deviation of the 20 runs of KM, PAM, and BPAM is calculated. In addition, the *t-test* is used to determine if there are significant differences among the mean values of each measure. The *Null hypothesis* H_0 is "no significance difference". Within 95% confidence interval, the critical value of t, $t_{critical}$, is 2.02 at degree of freedom (df=39). If the calculated $t < t_{critical}$ then the null hypothesis is accepted otherwise it is rejected and that means there is a significance difference between the results. t_1 refers to the *t-test* values between KM and BPAM results and t_2 refers to the *t-test* values between PAM and BPAM results.

6.3 Bisecting *k*-Medoids Performance Evaluation

The clustering quality of the BPAM algorithm is compared with those of the *k*-means (KM) and the PAM algorithms. Tables 3, 4, and 5 show the average performance of 20 runs of KM, PAM algorithm as well as the BPAM algorithm over the high dimensional *leukemia*, *yeast*, and *breast cancer* datasets, respectively, measured by the *F-measure*, *Entropy*, *Purity*, and *SI* quality measures for ITER equals 3.

As both PAM and BPAM are robust to the existence of outliers, they achieve better clustering results than the KM algorithm measured by both internal and external quality measures. It can be shown that BPAM outperforms both the KM and PAM algorithms. It produces clustering solutions with higher values for F-measure (up to 30%

Table3. Average performance of KM, PAM, and BPAM for the *Leukemia* dataset (k=3)

	KM	PAM	BPAM	Improvement
F-measure↑	0.8366 ±(0.01) t_1= 2.23, H_0:Rejected	0.8954 ±(0.01) t_2=2.97, H_0:Rejected	0.9457 ±(0.01)	(**+5.62%**)
Entropy↓	0.4086 ±(0.02) t_1=3.21, H_0:Rejected	0.3971 ±(0.01) t_2=2.54, H_0:Rejected	0.3129 ±(0.01)	(**+21.2%**)
Purity↑	0.4954 ±(0.03) t_1=2.94, H_0:Rejected	0.5877 ±(0.02) t_2=3.21, H_0:Rejected	0.6418 ±(0.02)	(**+9.2%**)
SI↓	0.6854 ±(0.03) t_1=3.11, H_0:Rejected	0.6059 ±(0.03) t_2=3.56, H_0:Rejected	0.4876 ±(0.02)	(**+19.52%**)

Table 4. Average performance of KM, PAM, and BPAM for the *Yeast* dataset (k=5)

	KM	PAM	BPAM	Improvement
F-measure↑	0.6301 ±(0.02) t_1=3.64, H_0:Rejected	0.6422 ±(0.03) t_2=2.64. H_0: Rejected	0.7758 ±(0.02)	(**+20.81%**)
Entropy↓	0.4351 ±(0.03) t_1=2.98, H_0:Rejected	0.4232 ±(0.04) t_2=3.15, H_0:Rejected	0.3687 ±(0.03)	(**+12.87%**)
Purity↑	0.6715 ±(0.02) t_1=3.14, H_0:Rejected	0.7067 ±(0.03) t_2=4.11, H_0:Rejected	0.7985 ±(0.02)	(**+12.98%**)
SI↓	2.8356 ±(0.02) t_1=3.42, H_0:Rejected	2.1591 ±(0.04) t_2=3.64, H_0:Rejected	1.6364 ±(0.03)	(**+24.2%**)

Table 5. Average performance of KM, PAM, and BPAM for the *Breast Cancer* dataset (k=4)

	KM	PAM	BPAM	Improvement
F-measure↑	0.4271 ±(0.03) t_1=3.65, H_0:Rejected	0.5055 ±(0.02) t_2=4.32, H_0:Rejected	0.6587 ±(0.01)	(**+30.31%**)
Entropy↓	0.7948 ±(0.01) t_1=3.66, H_0:Rejected	0.7031 ±(0.03) t_2=2.96, H_0:Rejected	0.5682 ±(0.02)	(**+19.19%**)
Purity↑	0.5734 ±(0.02) t_1=3.46, H_0:Rejected	0.6107 ±(0.02) t_2=4.15, H_0:Rejected	0.7154 ±(0.01)	(**+17.14%**)
SI↓	0.6897 ±(0.03) t_1=3.67, H_0:Rejected	0.6163 ±(0.03) t_2=2.93, H_0:Rejected	0.5523 ±(0.02)	(**+10.38%**)

improvement in *Breast Cancer* dataset), *Purity* (up to 17% improvement in *Breast cancer* dataset) and lower values for *Entropy* (up to 21% improvement in *Leukemia* dataset), and *SI* index (up to 24% improvement in *Yeast* dataset) than those of the PAM algorithm. The BPAM algorithm is able to group the co-expressed genes together into more homogenous groups with better clustering quality than that of the KM and PAM algorithms. In addition, using the *average compactness* as a homogeneity criterion, the BPAM was able to discover the small size clusters rather than the PAM algorithm as in the *Yeast* dataset.

6.4 Variable Number of Clusters

In this sub section, the quality and time performances of the Bisecting *k*-medoids (BPAM) is examined with variable number of clusters. As clustering is defined as unsupervised classification of objects, the *SI* measure (as internal quality measure) is used to evaluate the clustering quality of both the PAM and BPAM along with their corresponding time performances. In Fig.5, the SI trajectories (lines) for the BPAM are always lower compared to those of the PAM algorithm for the three data sets. In addition, the running time values (columns bars) are always lower than those of the PAM algorithm. For large data set as Breast Cancer, the BPAM achieves up to 50% reduction of time with increasing number of clusters to 10 clusters.

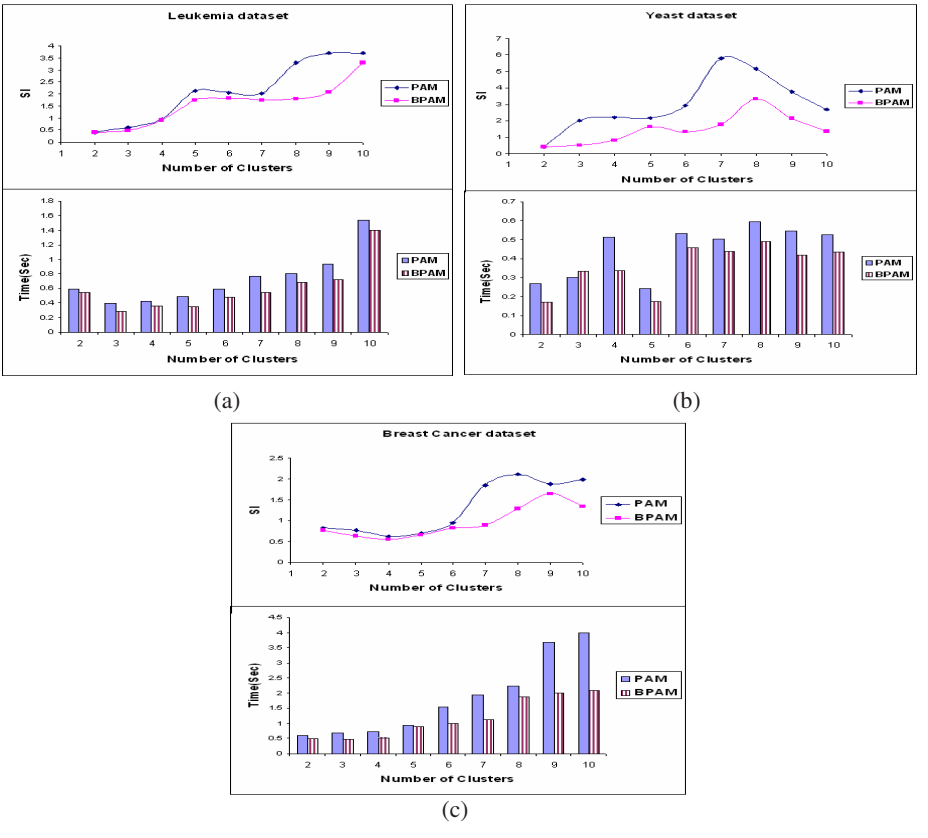


Fig. 5. SI trajectory and the running time values for the BPAM compared to those of PAM for (a) Leukemia dataset, (b) Yeast data set, and (c) Breast Cancer data set

7 Conclusions and Future Works

In this paper, a new bisecting *k*-medoids (BPAM) was presented. The BPAM was utilized with better clustering results than those produced by the traditional *k*-medoids

and *k*-means approaches. The BPAM was effectively applied for a number of gene-expression datasets and it achieves higher values for F-measure (up to 30% improvement), *Purity* (up to 17% improvement) and lower values for the *Entropy* (up to 21% improvement), and *SI* index (up to 24% improvement) than those of the PAM algorithm. The main reason for this enhancement is that the BPAM partitions the dataset based on a homogeneity criterion. The proposed BPAM was able to group the co-expressed genes together into more homogenous groups with lower time complexity. Future work includes applying the BPAM using different splitting criteria and developing a distributed version of the proposed algorithm to be applied on distributed datasets.

References

1. Hartigan, J.: Clustering Algorithms, Wiley Series in Probability and Mathematical Statistics (1975)
2. Jain, A., Murty, M., Flynn, P.: Data Clustering: A Review. ACM computing surveys 31, 264–323 (1999)
3. Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. Journal of Computational Biology 6(3/4), 281–297 (1999)
4. Shamir, R., Sharan, R.: Algorithmic Approaches to Clustering Gene Expression Data. In: Current Topics in Computational Biology, pp. 269–299. MIT Press, Cambridge (2002)
5. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. Information Processing Letters 76(200), 175–181 (2000)
6. Hartigan, J., Wong, M.: A k-means Clustering Algorithm. Applied Statistics 28, 100–108 (1979)
7. Bezdek, J., Ehrlich, R., Full, W.: The Fuzzy C-Means Clustering Algorithm. Computers and Geosciences 10, 191–203 (1984)
8. Savaresi, S., Boley, D.: On the Performance of Bisecting K-means and PDDP. In: Proc. of the 1st SIAM Int. Conf. on Data Mining, pp. 1–14 (2001)
9. Yousri, N.A., Ismail, M.A., Kamel, M.S.: Discovering Connected Patterns in Gene Expression Arrays. In: IEEE Symposium on Computational intelligence and Bioinformatics and Computational Biology (CIBCB), pp. 113–120 (2007)
10. Lu, Y., Lu, S., Fotouhi, F., Deng, Y., Brown, S.: Incremental Genetic K-means Algorithm and its Application in Gene Expression Data Analysis. BMC Bioinformatics 5(172) (2004)
11. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M.A., Bloomfield, C., Lander, E.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286(5439), 531–537 (1999)
12. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Kluwer Academic Publishers, Dordrecht (2003)
13. Kaufmann, L., Rousseeuw, P.: Finding groups in data. Wiley, Chichester (1990)
14. Kustra, R., Zaganski, A.: Incorporating Gene Ontology in Clustering Gene Expression Data. In: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, pp. 555–563 (2006)
15. Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: VLDB, pp. 144–155 (1994)

16. Hammouda, K., Kamel, M.: Collaborative Document Clustering. In: SIAM Conference on Data Mining (SDM 2006), pp. 453–463 (2006)
17. Zhao, Y., Karypis, G.: Criterion Functions for Document Clustering: Experiments and Analysis, Technical report, Department of Computer Science,University of Minnesota, Minneapolis, MN (2002)
18. Bensaid, A., Hall, L.O., Bezdek, J., Clarke, L., Silbiger, M., Arrington, J., Murtagh, R.: Validity-guided (Re)Clustering with applications to imige segmentation. IEEE Transactions on Fuzzy Systems, 112–123 (1996)
19. Tavazoie, S., Hughes, J., Campbell, M., Cho, R., Church, G.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
20. West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J., Marks, J., Nevins, J.: Predicting the clinical status of human breast cancer by using gene expression profiles. Proc. Natl. Acad. Sci. USA 98, 11462–11467 (2001)
21. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., Golub, T.: Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. PNAS 96, 2907–2912 (1999)
22. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster Analysis and Display of Genome-wide Expression Patterns. Proceedings of the National Academy of Sciences of the United States of America 95(25), 14863–14868 (1998)