BIOINFORMATION
Discovery at the interface of physical and biological sciences

open access www.bioinformation.net Hypothesis

Volume 9(2)

A novel harmony search-K means hybrid algorithm for clustering gene expression data

KA Abdul Nazeer[1]*, MP Sebastian[2] & SD Madhu Kumar[1]

[1]Department of Computer Science and Engineering, National Institute of Technology Calicut, India- 673 601; [2]Information Technology and Systems Area, Indian Institute of Management Kozhikode, India- 673 570; KA Abdul Nazeer – Email: nazeer@nitc.ac.in; Phone: 91-495-2286818; *Corresponding author

Received December 18, 2012; Accepted December 21, 2012; Published January 18, 2013

Abstract:
Recent progress in bioinformatics research has led to the accumulation of huge quantities of biological data at various data sources. The DNA microarray technology makes it possible to simultaneously analyze large number of genes across different samples. Clustering of microarray data can reveal the hidden gene expression patterns from large quantities of expression data that in turn offers tremendous possibilities in functional genomics, comparative genomics, disease diagnosis and drug development. The k-means clustering algorithm is widely used for many practical applications. But the original k-¬means algorithm has several drawbacks. It is computationally expensive and generates locally optimal solutions based on the random choice of the initial centroids. Several methods have been proposed in the literature for improving the performance of the k-¬means algorithm. A meta-heuristic optimization algorithm named harmony search helps find out near-global optimal solutions by searching the entire solution space. Low clustering accuracy of the existing algorithms limits their use in many crucial applications of life sciences. In this paper we propose a novel Harmony Search-K means Hybrid (HSKH) algorithm for clustering the gene expression data. Experimental results show that the proposed algorithm produces clusters with better accuracy in comparison with the existing algorithms.

Background:
The advancements in scientific data collection methods have led to the piling up of large quantities of data at various biology and life sciences databases. The advent of DNA microarrays facilitated the simultaneous monitoring of the expression levels of thousands of genes across a multitude of samples. Unraveling the hidden patterns from the gene expression data offers a distinct opportunity for understanding the important biological processes. The information thus obtained can be effectively utilized for important applications like early diagnosis and classification of various diseases. However, the complexity of the enormous amount of the genomic data poses a big challenge to the task of unearthing the hidden patterns from the massive data banks. Cluster analysis **[1]** is a first step towards addressing this challenge.

Cluster analysis is one of the major data mining methods which help identify the natural grouping in a set of data items. Clustering is the process of partitioning a given set of objects into disjoint clusters. This is done in such a way that the objects in the same cluster are similar while objects belonging to different clusters differ considerably, with respect to their attributes. Clustering of microarray Gene Expression (GE) data helps understand the gene functions, gene regulation and cellular processes **[2].** The genes in the same cluster exhibit similar expression patterns and are likely to be co-regulated. Clustering of tissue samples collected from various people including healthy persons and those infected with diseases like cancer, based on the similarity in expression patterns, can help in effective classification of unknown samples which in turn can lead to early diagnosis of diseases **[3].**

The k-means algorithm **[4]** 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. Moreover, this algorithm results in locally

optimum solutions and produces different types of clusters depending upon the random choice of the initial centroids. Several methods were proposed in the literature for improving the performance of the k-means clustering algorithm **[5-8].** However, none of them could find wide acceptance among the users. Clustering can be thought of as an optimization problem that maximizes the intra-cluster similarity and minimizes the inter-cluster similarity among the data points. Attempts were made to develop hybrid algorithms by combining the k-means algorithm with meta-heuristic techniques such as Harmony Search optimization **[9-11]**, in an attempt to obtain globally optimum solutions. This technique makes use of a harmony memory which is initialized with randomly generated feasible solutions to provide a global solution space. The harmony memory is then improvised by generating a new solution from the existing candidate solutions and replacing the current worst solution by the new solution, if it has a better fitness value. This process is repeated for several times and the best solution in the harmony memory is selected as the final solution.

The inadequate clustering accuracy of the existing methods is a bottleneck in crucial applications in the areas of Biology and Life Sciences. This paper presents a novel and effective Harmony Search-K means Hybrid (HSKH) algorithm for clustering the microarray gene expression data **[12-14].**

Methodology:

The proposed HSKH clustering algorithm consists of two phases. In the first phase, the initial centroids of the clusters are determined using an improved Harmony Search optimization technique. The centroids thus determined are used in the second phase to form the final clusters by repetitively assigning the data points to the clusters with the nearest centroids.

The Improved Harmony Search Algorithm

In this method the harmony memory is initialized with *n* number of candidate solutions in a systematic way, where *n* is the number of columns in the GE matrix. Each row in the harmony memory corresponds to a specific clustering solution represented by *k* columns. Each column represents a cluster and the value corresponding to the column indicates the index of the data-point that represents the centroid of the corresponding cluster.

First of all, the first column of the GE matrix (*Sample-IDs* or *Gene-IDs,* as the case may be) is sorted based on the value of the data items present in the second column (expression levels *of the gene in the samples* or *in the sample of the genes*). The sorted data items are then divided into *k* equal sets. The index of the data point which comes at the middle of each set is selected as the centroid of that cluster. A set of *k* such centroids will constitute a candidate solution. The above procedure is repeated for all the *n* columns of the Gene Expression matrix and the harmony memory is initialized with *n* candidate solutions thus obtained. Then the fitness values of all the solutions are evaluated as the Average Distance of Data points to Cluster centroids (ADDC) **(Please see supplementary material for equation and explanation).**

Assigning Data Points to Clusters

After determining the initial centroids using the Improved Harmony Search algorithm, the data items are assigned to the clusters with the nearest centroids using a variant of the method followed in **[6] (Please see supplementary material).**

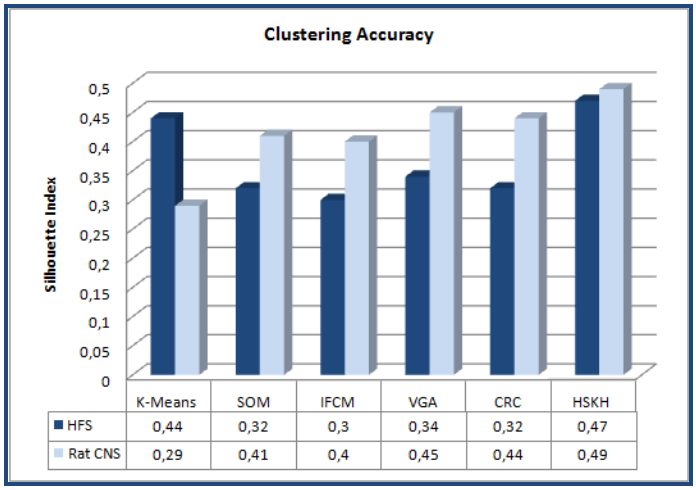

Figure 1: Performance Comparison of the Algorithms

Result and Discussion:

For evaluating the clustering accuracy of the proposed HSKH algorithm, two standard Gene Expression data sets, namely, the Human Fibroblast Serum data **[15]** and the Rat CNS data **[16]** were given as inputs to the original K-means algorithm and the proposed HSKH algorithm. The accuracy of clustering is determined in terms of the *Silhouette Index* **[17]** metric. Its value ranges from -1 to 1, with higher value indicating better solution.

The results of the experiment are compared with the clustering accuracy of four other methods available in the literature **[18].** The methods considered are Self Organizing Maps (SOM) **[19],** Iterative Fuzzy C Means (IFCM) **[20],** Variable string length Genetic Algorithm (VGA) **[21]** and Chinese Restaurant Clustering (CRC) **[22].** The clustering accuracies of the algorithms for the two data sets are compared in **(Figure 1).**

It can be seen from the above experiments that the proposed HSKH algorithm performs better than the existing methods. The significant improvement in the clustering accuracy can be attributed to the improved harmony search method used to determine the initial centroids of the clusters. As a result of the refined initial centroids and the efficient assignment of the data points to the different clusters, our algorithm yields better clusters in less number of computational steps.

Conclusion:

Clustering is a first step towards retrieving useful knowledge from microarray gene expression data. Accurate clustering is very much needed in Biology and Life Science applications as the resulting clusters are used for making crucial inferences on disease diagnosis and drug development. The conventional clustering techniques do not generally give sufficiently good results. The HSKH algorithm proposed in this paper combines the merits of the Harmony Search Optimization and the Enhanced K means algorithms, producing a globally optimal solution with significant improvement in the accuracy of clustering compared to the original k-means and other known clustering techniques.

The proposed HSKH algorithm may not be effective in its

present form for certain clustering applications where the number of clusters *k* is not known in advance. But it is not a bottleneck in the case of clustering gene expression data where the concern is to classify the data into a specified number of clusters. Development of a clustering technique with automatic computation of *k* is suggested as a topic for further research.

References:

[1] Ben-Dor A *et al. J ComputI Biol.* 1999 **6**: 281 [PMID: 10582567]
[2] Daxin Jiang *et al. IEEE Transactions on Data and Knowledge Engineering*. 2004 **16**: 1370
[3] Marcilio CP de Suoto *et al. BMC Bioinformatics.* 2008 **9**: 497 [PMID: 19038021]
[4] http://www.nt.tuwien.ac.at/fileadmin/courses/389075/Least_Squares_Quantization_in_PCM.pdf
[5] http://dataclustering.cse.msu.edu/papers/JainDataClusteringPRL09.pdf
[6] http://www.zju.edu.cn/jzus/article.php?doi=10.1631/jzus.2006.A1626
[7] http://dl.acm.org/citation.cfm?id=593469
[8] http://link.springer.com/article/10.1007%2Fs00357-001-0004-3?LI=true#page-1
[9] http://link.springer.com/chapter/10.1007%2F978-3-642-04317-8_1?LI=true#page-1
[10] http://folk.uib.no/ssu029/Pdf_file/Lee05.pdf
[11] http://www.citeulike.org/user/lop/article/1044500
[12] Berrar D *et al. BMC Bioinformatics.* 2006 **7**: 73 [PMID: 16483361]
[13] http://dl.acm.org/citation.cfm?doid=381371.381378
[14] Kerr G *et al. Comput Biol Med.* 2008 **38**: 283 [PMID: 18061589]
[15] http://genome-www.stanford.edu/serum/
[16] http://faculty.washington.edu/kayee/cluster/
[17] http://dl.acm.org/citation.cfm?id=38772
[18] Sanghamitra Bandyopadhyay *et al. Bioinformatics.* 2007 [PMID: 17720981]
[19] http://www.pnas.org/content/96/6/2907.long
[20] Tomida S *et al. Bioinformatics* 2002 **18**: 1073 [PMID: 12176830]
[21] http://library.isical.ac.in/jspui/bitstream/10263/3069/1/fuzzy%20partiotion.pdf
[22] Quin ZS, *Bioinformatics*. 2006 22: 1988 [PMID: 16766561]

Edited by P Kangueane
Citation: Nazeer *et al.* Bioinformation 9(2): 084-088 (2013)

BIOINFORMATION open access

Supplementary material:

Methodology:

The Improved Harmony Search Algorithm

In this method the harmony memory is initialized with *n* number of candidate solutions in a systematic way, where *n* is the number of columns in the GE matrix. Each row in the harmony memory corresponds to a specific clustering solution represented by *k* columns. Each column represents a cluster and the value corresponding to the column indicates the index of the data-point that represents the centroid of the corresponding cluster.

First of all, the first column of the GE matrix (*Sample-IDs* or *Gene-IDs,* as the case may be) is sorted based on the value of the data items present in the second column (expression levels *of the gene in the samples* or *in the sample of the genes*). The sorted data items are then divided into *k* equal sets. The index of the data point which comes at the middle of each set is selected as the centroid of that cluster. A set of *k* such centroids will constitute a candidate solution. The above procedure is repeated for all the *n* columns of the Gene Expression matrix and the harmony memory is initialized with *n* candidate solutions thus obtained. Then the fitness values of all the solutions are evaluated as the Average Distance of Data points to Cluster centroids (ADDC). This value is measured as

$$ADDC = \frac{1}{k}\{\sum_{i=1}^{k}\left(\frac{\sum_{j=1}^{ni} D(ci, dij)}{ni}\right)\}$$

where *k* is the number of clusters, *ni* is the number of data items in each cluster and *D(ci, dij)* is the distance of each data point to the corresponding cluster centroid. A new solution is then improvised from the harmony memory using the approach described in [10]. The fitness of the new solution is determined by computing the ADDC value. If this solution is better than the worst solution in the harmony memory, then the harmony memory is updated by replacing the worst solution with the newly generated solution. This improvisation step is repeated for MI (Maximum Iterations) number of iterations. Finally the best solution in the harmony memory is returned as the initial centroids of the clusters.

Phase 1: Improved Harmony Search Algorithm for finding the Initial Centroids

Input:

Gene Expression data matrix, D (Samples * Genes) / / *m+1* rows * *n+1* columns

k / / Number of desired clusters.

HMCR, PAR, MI / / Harmony search optimization parameters

Output:

A set of *k* initial centroids

Steps:

1. For each column of the data matrix, *i = 2* to *n+1*
   - 1.1 Sort the first column of the data matrix based on column *i*;
   - 1.2 Divide the sorted column into *k* equal parts;
   - 1.3 For each part *j =1* to *k*, determine the index of the data item at the middle.
     - 1.3.1 Initialize the harmony memory *HM[i-1][j]* with the middle index.

   Endfor

   Endfor
2. Calculate the fitness value of all the candidate solutions in the harmony memory.
3. For *p = 1* to *MI*

   3.1 Improvise a new solution from the harmony memory.

   3.2 If the new solution is better than the worst solution in the harmony memory, replace the worst solution with the new solution.

   Endfor
4. Return the best solution in the harmony memory as the set of initial centroids.

Assigning Data Points to Clusters

After determining the initial centroids using the Improved Harmony Search algorithm, the data items are assigned to the clusters with the nearest centroids using a variant of the method followed in [6].

Phase 2: Algorithm for assigning data-points to the clusters

Input:

D = {d_1, d_2,.....,d_n} / / set of *n* data-points.

C = {c_1, c_2,.....,c_k} / / set of *k* centroids

Output:

A set of *k* clusters

Steps:

1. Compute the Euclidean distance of each data-point *di* (1<=i<=n) to all the centroids *cj* (1<=j<=k) as *d(di, cj)*;
2. For each data-point *di*, find the closest centroid *cj* and assign *di* to cluster *j*.
3. Set ClusterId[i]=j; / / j:Id of the closest cluster
4. Set Nearest_Dist[i]= *d(di, cj)*;
5. For each cluster *j* (1<=j<=k), recalculate the centroids;
6. **Repeat**
7. For each data-point *di*,

   7.1 Compute its distance from the centroid of the present nearest cluster;
   7.2 If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster; Else
   7.2.1 For every centroid *cj* (1<=j<=k), Compute the distance *d(di, cj)*;
   Endfor;
   7.2.2 Assign the data-point *di* to the cluster with the nearest centroid *cj;*
   7.2.3 Set ClusterId[i]=j;
   7.2.4 Set Nearest_Dist[i]= *d(di, cj)*;

Endfor;
8. For each cluster *j* (1<=j<=k), recalculate the centroids;
Until convergence (i.e., no more data-points cross the cluster boundaries).