UNSUPERVISED CLUSTER ANALYSIS IN BIOINFORMATICS

D.K. Tasoulis, V.P. Plagianakos, and M.N. Vrahatis
Department of Mathematics, University of Patras,
University of Patras Artificial Intelligence Research Center
GR–26110 Patras, Greece
{dtas,vpp,vrahatis}@math.upatras.gr

Abstract: In this paper, we investigate the application of an unsupervised extension of the recently proposed k–windows clustering algorithm on gene expression microarray data. The k–windows algorithm is used both to identify sets of genes according to their expression in a set of samples, and to cluster samples into homogeneous groups. Experimental results and comparisons indicate that this is a promising approach.

1 Introduction

In any living cell that undergoes a biological process, different subsets of its genes are expressed. Gene expression at a given stage crucially affects a cell's function. While traditional genomic research approaches examine and collect data for a single gene locally, DNA microarray technologies enable us to monitor the expression pattern for thousands of genes simultaneously. Thus the discovery of patterns hidden in the gene expression microarray data has become a challenge for functional genomics and proteomics. A key step in this procedure is cluster analysis. Generally, clustering can be defined as the process of partitioning a collection of objects into one or more groups of similar objects. In particular, clustering can be used either to identify sets of genes according to their expression in a set of samples [4], or to cluster samples into homogeneous groups that may correspond to particular macroscopic phenotypes [6]. Recently clustering techniques have been applied to gene expression data [4, 8] and have proved useful for identifying biologically relevant groupings of genes and samples.

In this paper the unsupervised extension of the recently proposed k-windows clustering algorithm is applied on gene expression microarray data. Our approach is compared against the k-means [3] and the Growing Neural Gas (GNG) [5] clustering algorithms. The paper is organized as follows: the next section outlines the clustering algorithms tested. In Section 3 we describe the application of the algorithms in the gene expression microarray data and exhibit experimental results and comparisons. The paper ends with concluding remarks.

2 Unsupervised clustering

In this section we briefly outline the k–means clustering algorithm [3], the Growing Neural Gas (GNG) [5] algorithm, and the unsupervised extension of the k–windows clustering algorithm [10].

2.1 k–means

The well known k–means clustering algorithm [7], works by initializing k prototype vectors p^j over the data set. Next it assigns each of the data vectors x^i to the cluster whose prototype is closest to x^i. After all the data vectors are assigned, the prototypes are recalculated as the centroids of the data vectors assigned to each cluster. The certainty of the assignment of the data sample to the cluster defined by the the prototype is measured by the membership function, defined as:

$$\mu_j(x^i) = \begin{cases} 1 & \text{if} \quad \|x^i - p^j\| \leqslant \|x^i - p^l\| \quad \forall\, l \neq j \\ 0 & \text{otherwise} \end{cases}$$

and the centroid of each cluster is calculated by:

$$p^j = \frac{\sum_{i=1}^{n} \mu_j(x^i) x^i}{\sum_{i=1}^{n} \mu_j(x^i)}$$

To stop the algorithm the following measure of distortion of the actual partition is used:

$$d = \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_j(x^i) \|x^i - p^j\|^2.$$

Thus, the algorithm stops when the change in the distortion is below a prespecified value. To determine the exact number of clusters present in a

dataset the algorithm can be easily extended to an unsupervised version (UKM). This can be achieved by introducing new prototypes each time a new pattern is presented to the algorithm for which the distance from the closest prototype exceeds a threshold v.

2.2 Growing Neural Gas

The Growing Neural Gas (GNG) is an incremental neural network. It can be described as a graph consisting of k nodes, each of which has an associated weight vector, w_j, defining the node's position in the data space and a set of edges between the node and its neighbors. During the clustering procedure, new nodes are introduced into the network until a maximal number of nodes is reached. The GNG starts with two nodes, randomly positioned in the data space, connected by an edge. Adaptation of weights, i.e. the nodes position, is performed iteratively. For each data object the closest node (winner), s_1, and the closest neighbor of a winner, node s_2, are determined. These two nodes are connected by an edge. An age variable is associated with each edge. At each learning step the ages of all edges emanating from the winner are increased by 1. When the edge between s_1 and s_2 is created its age is set to 0. By tracing the changes of the age variable it is possible to detect inactive nodes. Edges exceeding a maximal age, R, and any nodes having no emanating edges are removed. The neighborhood of the winner is limited to its topological neighbors. The winner and its topological neighbors are moved in the data space toward the presented object by a constant fraction of the distance, defined separately for the winner and its topological neighbors. There is no neighborhood function or ranking concept. Thus, all topological neighbors are updated in the same way.

2.3 Unsupervised k-windows

The k-windows clustering algorithm [10] uses a windowing technique to discover the clusters present in a dataset. More specifically, if we suppose that the dataset lies in d dimensions, it initializes a number of d–dimensional windows over the dataset. At a next step it iteratively moves and enlarges those windows to capture all the patterns that belong to one cluster within each window. The movement and enlargement procedures are guided by the points that lie within each window at each iteration. As long as the the movement and enlargement procedure do not change the number of points within each window drastically they terminate. The final set of windows defines the clustering result of the algorithm. The unsupervised k–windows algorithm (UKW) generalizes the original algorithm by endogenously determining the number of clusters. The key idea to achieve this is to apply the k–windows algorithm using a "sufficiently" large number of initial windows. The windowing technique of the k–windows algorithm allows for a large number of initial windows to be examined, without any significant overhead in time complexity. At a final step the windows that contain a high percentage of common points from the dataset are considered to belong to the same cluster. Thus the number of cluster can be determined [1, 2, 9].

3 Experimental results

To investigate the performance of the clustering algorithms in gene expression microarray data we used data from a previous study that examined mRNA expression profiles from 72 leukemia patients to develop an expression–based classification method for acute leukemia [6]. This data set contains a large number of patients and has been well characterized. Each sample is measured over 7129 genes. From those genes 50 genes were selected as proposed in [6]. The first 38 samples have been used for the clustering process (train set), while the remaining 34 were used to evaluate the final clusters (test set). The initial 38 samples contained 27 acute myeloid leukemia (ALL) samples and 11 acute lymphoblastic leukemia (AML) samples. The test set contained 20 ALL samples and 14 AML samples.

As demonstrated in Table 1, the application of the UKM algorithm on the training data set results in 5 clusters. Four of these are classified as ALL clusters since they contain exclusively ALL samples, with the exception of cluster 1 that also contains one AML sample. Cluster 5 contains only AML samples and thus is classified as AML cluster. By assigning the samples of the test set to the clusters that were discovered during training we observe that 5 of the AML samples are assigned to cluster 1 that is classified as an ALL cluster.

Leukemia type	Train Set: ALL Clusters				AML Clusters
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1
ALL	10	9	6	2	0
AML	1	0	0	0	10
	Test Set				
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1
ALL	16	2	0	2	0
AML	5	0	0	0	9

Table 1: Clustering result for the train set for the UKM algorithm

Leukemia type	Train Set: ALL Clusters				AML Clusters	
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
ALL	4	11	4	8	0	0
AML	0	1	0	0	7	3
	Test Set					
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
ALL	0	13	3	4	0	0
AML	0	4	0	0	3	7

Table 2: Clustering result for the train set for the GNG algorithm

Leukemia type	Train Set: ALL Clusters			AML Clusters
	Cluster 1	Cluster 2	Cluster 3	Cluster 1
ALL	9	8	10	0
AML	1	0	0	10
	Test Set			
	Cluster 1	Cluster 2	Cluster 3	Cluster 1
ALL	14	0	6	0
AML	3	0	0	11

Table 3: Clustering result for the train set for the UKW algorithm

On the other hand, the application of the GNG algorithm over the train set results in 6 clusters, as exhibited in Table 2. Similarly, 4 of the clusters are classified as ALL clusters, through the labels of the samples they contain. In this case the ALL cluster 2 contains a minority of 1 AML sample. These clusters result in the assignment of the tests samples to clusters with similar labels with the exception of 4 AML samples that are assigned to the ALL cluster 2. Finally, in Table 3 the results of the UKW algorithm are reported. In this case, 4 clusters are discovered. Three of them are classified as ALL clusters and one as an AML cluster. The assignment of the test samples results in only 3 AML incorrectly assigned to the ALL cluster 1.

4 Conclusions

In this paper we present a modification of the recently proposed k-windows clustering algorithm and compare it against two well–known clustering algorithms. Cluster analysis presented here groups leukemia samples into clusters based on similar gene expression microarray data. This approach makes no distributional assumptions, it is well-founded, and provides a sensitive and robust method to extract relevant information from DNA microarrays. Our results are promising and the proposed clustering methodology exhibits high classification success.

Acknowledgment

The authors acknowledge the support of the "Karatheodoris" research grant awarded by the Research Committee of the University of Patras, and the "Pythagoras" research grant awarded by the Greek Ministry of Education and Religious Affairs and the European Union.

References

[1] P. Alevizos, B. Boutsinas, D.K. Tasoulis, and M.N. Vrahatis. Improving the orthogonal range search k-windows clustering algorithm. In *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, pages 239–245. Washington, D.C., 2002.

[2] P. Alevizos, D.K. Tasoulis, and M.N. Vrahatis. Parallelizing the unsupervised k-windows clustering algorithm. *Lecture Notes in Computer Science (LNCS)*, 3019:225–232, 2004.

[3] R.O Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[4] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, 1998.

[5] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, pages 625–632. MIT Press, Cambridge MA, 1995.

[6] T.R. Golub, D.K. Slomin, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M.L. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[7] J.A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[8] D.K. Tasoulis, V.P. Plagianakos, and M.N. Vrahatis. Unsupervised clustering of bioinformatics data. In *European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, Aachen Germany*, 2004.

[9] D.K. Tasoulis and M.N. Vrahatis. Unsupervised distributed clustering. In *The IASTED International Conference on Parallel and Distributed Computing and Networks*. Innsbruck, Austria, 2004.

[10] M.N. Vrahatis, B. Boutsinas, P. Alevizos, and G. Pavlides. The new k-windows algorithm for improving the k-means clustering algorithm. *Journal of Complexity*, 18:375–391, 2002.