

A Method for Tight Clustering: with Application to Microarray

George C. Tseng¹, and Wing H. Wong²

¹*Department of Biostatistics, University of Pittsburgh*

²*Department of Statistics, Harvard University*

¹*ctseng@hsph.harvard.edu*, ²*wwong@hsph.harvard.edu*

Abstract

*In this paper we propose a method for clustering that produces tight and stable clusters without forcing all points into clusters. Many existing clustering algorithms have been applied in microarray data to search for gene clusters with similar expression patterns. However, none has provided a way to deal with an essential feature of array data: many genes are expressed sporadically and do not belong to any of the significant biological functions (clusters) of interest. In fact, most current algorithms aim to assign all genes into clusters. For many biological studies, however, we are mainly interested in the most informative, tight and stable clusters with sizes of, say, 20-60 genes for further investigation. Tight Clustering has been developed specifically to address this problem. The tightest and most stable clusters are identified in a sequential manner through an analysis of the tendency of genes to be grouped together under repeated resampling. We validated this method in the expression profiles of the *Drosophila* life cycle. The result is shown to better serve biological needs in microarray analysis.*

1. Introduction

Cluster analysis, an unsupervised learning method, is widely used to study the structure of the data when the response variable is unknown. Our task is to learn the structure of a d -dimensional distribution based on a training data of n observations from this distribution. The training data are represented by an $n \times d$ matrix.

In cluster analyses of microarray experiments, we start with a data matrix $\{\theta_{ij}\}_{n \times d}$, an $n \times d$ matrix representing the expression levels of n genes in d samples. If the goal is to obtain groups of genes with similar expression patterns which are likely to belong to similar functional pathways, we cluster genes with a given distance (dissimilarity) measure and have n points in d dimensions to be assigned into clusters. To find groups of samples with similar expression patterns, we cluster samples instead of genes, resulting in d points in n dimensions being clustered. This is useful, for example, in the discovery of subtypes of a disease. Microarray experiments normally have 500 to

3000 genes after filtering out genes with low information content and 10 to 500 samples, depending on the study.

Most clustering algorithms assign all points into clusters. However, in microarray experiments, we expect many genes to show uncorrelated variations and should be unrelated to the biological process that we are investigating. These genes should not be assigned to specific clusters and they are called sporadic points. When analyzing data with sporadic points, if the algorithm is forced to divide all points into clusters, both the estimation of the number of clusters will be problematic and the resulting clusters will be distorted and difficult to interpret.

In this paper, we propose a method to address this issue. The average co-membership matrix is introduced to identify candidates of tight clusters with the number of clusters k pre-specified. A large cluster, stably selected for consecutive k , will then be identified as a tight and stable cluster. The algorithm removes this cluster from the whole data and we continue to search for the next such cluster in the remaining points. Finally we validate these methods in a simulated example. We also illustrate the method on a set of expression profiles of *Drosophila melanogaster* during its life cycle.

2. Methods

The procedures of “Tight Clustering” are described below. Algorithm A is developed to identify candidates of tight clusters when the number of clusters k is pre-specified. The cluster that is stably selected by Algorithm A for consecutive values of k is then selected and removed from the whole data. This procedure is iteratively performed to produce a sequence of tight clusters in decreasing degrees of tightness.

2.1. Algorithm A

The following algorithm is used to select candidates of tight clusters when the number of clusters k in the K -means algorithm is pre-specified. The subsampling procedure is used to create variabilities so that a pair of points stably clustered together can be distinguished from those clustered by chance.

(a) Take a random subsample X' from the original data X , say with 70% of the original sample size. Apply K -means with the pre-specified k on X' to obtain the cluster centers $C(X', k) = (C_1, C_2, \dots, C_k)$.

(b) Use the clustering result $C(X', k)$ as a classifier to cluster the original data X according to the distances from each point to the cluster centers. Following the convention of Tibshirani et al. [1], the resulting clustering is represented by a co-membership matrix $D[C(X', k), X]$ where $D[C(X', k), X]_{ij}$, the element of the matrix in row i and column j , takes value 1 if point i and j are in the same cluster and 0 otherwise.

(c) Repeat independent random subsampling B times to obtain subsamples $X^{(1)}, X^{(2)}, \dots, X^{(B)}$. The average co-membership matrix is defined as $\bar{D} = \text{mean}(D[C(X^{(1)}, k), X], \dots, D[C(X^{(B)}, k), X])$.

(d) Search for a set of points $V = \{v_1, \dots, v_m\} \in \{1, \dots, n\}$ such that $\bar{D}_{v_i, v_j} > 1 - \alpha, \forall i, j$ where α is a constant close to 0.

Order sets with this property by size to obtain V_{k1}, V_{k2}, \dots . These V sets are candidates of tight clusters.

2.2. Sequential identification of tight clusters

The following algorithm is used to identify a tight cluster that is stably chosen by consecutive k . We first define a similarity measure of two sets V_i and V_j to be $s(V_i, V_j) = |V_i \cap V_j| / |V_i \cup V_j|$ where $|V|$ is the size of set V .

(a) Start with a suitable k_0 . Apply Algorithm A on consecutive k starting from k_0 . Choose the top 3 tightest clusters for each k , namely $\{V_{k0,1}, V_{k0,2}, V_{k0,3}\}, \{V_{k0+1,1}, V_{k0+1,2}, V_{k0+1,3}\}, \dots$

(b) Stop when $s(V_{k',1}, V_{(k'+1),m}) > \beta$. Here β is a constant close to 1, $k' \geq k_0$ and $1, m \in \{1, 2, 3\}$. Identify $V_{(k'+1)m}$ as the tightest and most stable cluster. Remove it from the whole data.

(c) Decrease k_0 by 1 and repeat step (a) and (b) to identify the next tightest cluster.

3. Result

We applied our algorithm to a cDNA microarray data [2]. The experiment contained 66 time points in four life cycle periods. In Figure 1., the heat map [3] of 15 tight clusters when $\alpha = 0.1, \beta = 0.6, B = 10$ and $k_0 = 25$ is presented. The four life cycle periods are separated by black marks above the heat map. Figure 2. gives a side-by-side comparison of Tight Clustering and K -means algorithm. The left cluster is the first cluster identified by Tight Clustering in Figure 1. The right cluster is the corresponding cluster in K -means clustering when $k = 15$. The two clusters have 61 common genes that were ordered and shown in the upper region. K -means, however, includes additional 67 genes with more variable patterns in the cluster and is likely to

introduce many more false-positives. This figure shows the ability of Tight Clustering to produce tight and informative clusters for biologists to follow up, mainly because it does not require assigning all genes into clusters.

Figure 1. 15 tight clusters in Drosophila data

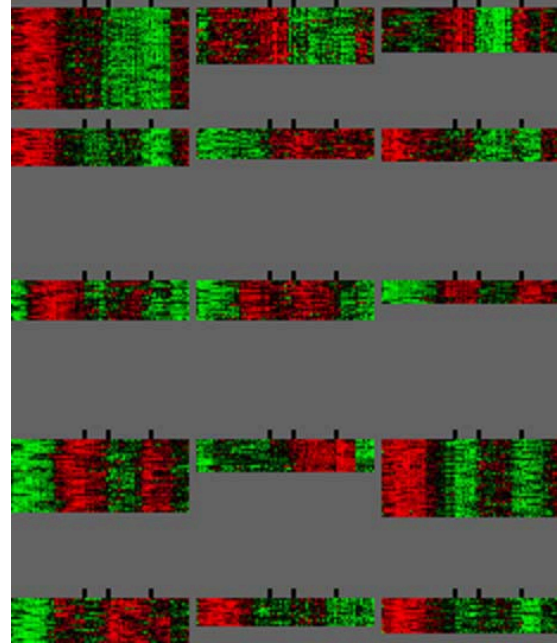
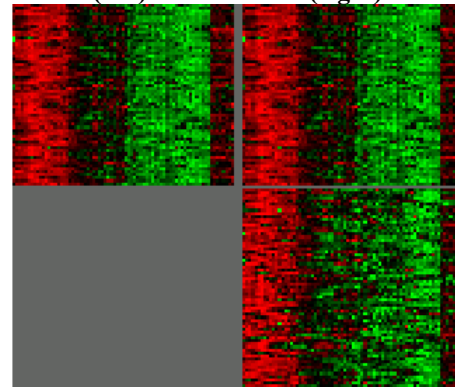


Figure 2. Side-by-side comparison of tight clustering (left) and K -means (right)



4. References

- [1] R. Tibshirani, G. Walther, D. Bostein, and P. O. Brown (2001). "Cluster validation by prediction strength.", Technical report, Department of Statistics, Stanford University.
- [2] M. Arbeitman, E. Furlong, F. Imam, E. Johnson, B. Baker, R. Davis. and K. White (2002). "Gene expression during the life cycle of *Drosophila melanogaster*." *Science* **297**, 2270-2275.
- [3] M. B. Eisen (2000). "TreeView (version 1.5)." Software download: <http://rana.lbl.gov//>