The Fourth International Conference on Computational Systems Biology (ISB2010)
Suzhou, China, September 9–11, 2010

Constrained Subspace Clustering for Time Series Gene Expression Data

Jibin Qu[1] Michael Ng[2] Luonan Chen[3]

[1]Institute of Applied Mathematics, Academy of Mathematics and Systems Science, CAS, Beijing 100190
[2]Department of Mathematics, Hong Kong Baptist University, Hong Kong
[3]Key Laboratory of Systems Biology, SIBS-Novo Nordisk Translational Research Center for Pre-diabetes, Shanghai Institutes for Biological Sciences, CAS, Shanghai 200031

Abstract For time series gene expression data, it is an important problem to find subgroups of genes with similar expression pattern in a consecutive time window. In this paper, we extend a fuzzy c-means clustering algorithm to construct two models to detect biclusters respectively, i.e., constant value biclusters and similarity-based biclusters whose gene expression profiles are similar within consecutive time points. Finally, we verify our methods on several artificial datasets.

Keywords bicluster; time series; gene expression data; fuzzy c-means; time segmentation

1 Introduction

In analyzing data of DNA microarray experiments, it is important to find groups of genes that share similar expression patterns which characterize a special cellular processes at a specific period [1, 2]. In many situations, a cellular process is active only under a subset of conditions. However, classical clustering techniques such as hierarchical clustering and k-means clustering are generally not designed to detect co-activated gene groups under specific conditions or time periods.

In recent years, biclustering algorithms have been suggested to identify local patterns in gene expression data. The local pattern, called bicluster, is defined as a subset of genes that exhibit compatible expression pattern over a subset of conditions, often means a transcription module or an active pathway. There exist many biclustering methods, including CC algorithm [3], coupled two-way clustering [4], ISA [5], SAMBA [6], Bimax [7] and so on. Many of them are also available from their websites.

In this paper, we focus on detecting biclusters in time series gene expression data. The difference from biclusters in ordinary gene expression data, i.e., sample data, is that the condition of the data is time points, i.e., time series data. In other words, the subset of the conditions in a bicluster must be consecutive, i.e., we face with a bicluster problem with a constraint on time horizon. Specifically, we introduce a framework named fuzzy c-means clustering algorithm [8] to solve such a constrained bicluster problem. By adding a penalty term to guarantee consecutive time points in one bicluster and defining new

weight variables, we aim to detect two types of biclusters based on different distance expressions: one is the constant value bicluster and the other is the similarity-based bicluster whose gene expression profiles are similar within consecutive time points.

Next, we first introduce our methods in detail, and then provide several numerical examples to verify our models.

2 Methods

A microarray dataset can be seen as a $N \times M$ matrix, each row is the profile of a gene in all conditions and each column is an array for all the genes in a condition. There are N genes and M time points, generally $N \gg M$. We want to find K meaningful biclusters in the matrix and each bicluster is formed as a submatrix $\{N_i, M_i\}$ for $i = 1, ..., K$, N_i is the subset of genes and M_i is the subset of conditions for the ith bicluster. In our model, the condition is time point, and therefore the time points in a bicluster must be consecutive, i.e. M_i should consist of consecutive time points. Firstly we simply introduce the fuzzy c-means algorithm, which is a classical clustering algorithm by partitioning the rows only.

The objective function of fuzzy c-means clustering is as follows:

$$\min\ P(U,Z) = \sum_{l=1}^{K}\sum_{i=1}^{N} u_{i,l}^{\alpha} d_{i,l}$$

subject to

$$\begin{cases} d_{i,l} = \|x_i - z_l\|^2, & 1 \le i \le N, 1 \le l \le K \\ \sum\limits_{l=1}^{K} u_{i,l} = 1, & 1 \le i \le N \\ 0 \le u_{i,l} \le 1, & 1 \le i \le N, 1 \le l \le K \end{cases}$$

- $u_{i,l}$ is the rate that object i is allocated to cluster l. z_l is the centroid of cluster l.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership $u_{i,l}$ and the cluster centers z_j:

$$\begin{cases} u_{i,l} = \dfrac{1}{\sum\limits_{k=1}^{K} (\frac{d_{i,l}}{d_{i,k}})^{\frac{2}{\alpha-1}}}, & 1 \le i \le N, 1 \le l \le K \\ z_l = \dfrac{\sum\limits_{i=1}^{N} u_{i,l}^{\alpha} x_i}{\sum\limits_{i=1}^{N} u_{i,l}^{\alpha}}, & 1 \le l \le K \end{cases} \tag{1}$$

Based on the above fuzzy c-means algorithm we propose two models. In our first model, we define every centroid of a bicluster as a value, add a penalty term to guarantee consecutive time points in one bicluster and further define a new weight variable. In our second model, the penalty term and new weight variable are also added and we modify the distance definition. To avoid high false positive results which are hard to be solved by many biclustering algorithms, we introduce an additive bicluster to store outliers in both models.

2.1 Model 1: detection of constant value biclusters in time series gene expression data

This model aims to detect the constant value biclusters in time series gene expression data. We utilize a fuzzy c-means algorithm to divide the data into K biclusters, where we generally use a large value as K. The objective function of the bicluster problem with variables (U,W,Z) is given as follows:

$$\min\ P(U,W,Z) = \sum_{l=1}^{K+1}\sum_{i=1}^{N}\sum_{j=1}^{M} u_{i,l}^{\alpha} w_{l,j} d_{i,j,l} + \beta \sum_{l=1}^{K}(w_{l,1} + w_{l,M} + \sum_{j=1}^{M-1} |w_{l,j} - w_{l,j+1}|) \quad (2)$$

subject to

$$\begin{cases} d_{i,j,l} = (x_{i,j} - z_l)^2, & 1 \le i \le N, 1 \le j \le M, 1 \le l \le K \\ d_{i,j,K+1} = D, & 1 \le i \le N, 1 \le j \le M \\ \sum_{l=1}^{K+1} u_{i,l} = 1, & 1 \le i \le N \\ 0 \le u_{i,l} \le 1, & 1 \le i \le N, 1 \le l \le K+1 \\ \sum_{j=1}^{M} w_{l,j} = 1, & 1 \le l \le K+1 \\ 0 \le w_{l,j} \le 1, & 1 \le j \le M, 1 \le l \le K+1. \end{cases} \quad (3)$$

- U is an $N \times K$ partition matrix and $u_{i,l}$ is between 0 and 1, indicating the rate that object i is allocated to bicluster l. Here α is set to be greater than 1, thereby allowing that a gene can belong to several clusters. In our model we set $\alpha = 2$.

- W is an $(K+1) \times M$ partition matrix, and $w_{l,j}$ is a weight for the jth time point of the lth biclusters. This is the main difference between this model and fuzzy c-means clustering, because fuzzy c-means clustering just has weights for rows, i.e. U. Thus there is a general weight $u_{i,l}^{\alpha} w_{l,j}$ for every data point $x_{i,j}$ of each bicluster l.

- $Z = \{z_1, z_2, \cdots, z_K\}$ is a set of K values representing the centroids of K biclusters.

- $d_{i,j,l} = (x_{i,j} - z_l)^2$ is the distance between the data point $x_{i,j}$ and the centroid z_l. Here we use the Euclidean distance expression just like in ordinary k-means algorithm. For the additive bicluster, d is a constant value: $d_{i,j,K+1} = D$, which means that if the distance between a data point and any centroid is larger than D, then the data point is a outlier and belongs to the additive bicluster. There is no centroid for the additive bicluster.

- The penalty term $w_{l,1} + w_{l,M} + \sum_{j=1}^{M-1} |w_{l,j} - w_{l,j+1}|$ is used to measure the total variation of the time variable weights at the lth bicluster. The idea is to give a piecewise constant function on $w_{l,j}$. This also enables us to determine a consecutive window of time variables in a bicluster. Also β is a parameter for controlling the strength of forming a piecewise constant function on $w_{l,j}$. Edge effect is also considered in this term. There is no need to penalize the additive bicluster, so $w_{K+1,j} = \frac{1}{M}$

The above optimization problem is a large scale nonlinear mathematical programming and is generally difficult to be solved directly. In this paper, we adopt a decomposition scheme to solve this problem, i.e., this problem can be minimized by iteratively solving the following three sub-problems.

2.1.1 Problem 1: Fixing $Z = \hat{Z}, W = \hat{W}$ and solving the reduced problem $min_U\ P(U, \hat{Z}, \hat{W})$

For U:

$$\min\ P(U, \hat{W}, \hat{Z}) = \sum_{l=1}^{K+1} \sum_{i=1}^{N} \sum_{j=1}^{M} u_{i,l}^{\alpha} \hat{w}_{l,j} \hat{d}_{i,j,l}$$

subject to

$$\begin{cases} \hat{d}_{i,j,l} = (x_{i,j} - \hat{z}_l)^2, & 1 \le i \le N, 1 \le j \le M, 1 \le l \le K \\ \hat{d}_{i,j,K+1} = D, & 1 \le i \le N, 1 \le j \le M \\ \sum_{l=1}^{K+1} u_{i,l} = 1, & 1 \le i \le N \\ 0 \le u_{i,l} \le 1, & 1 \le i \le N, 1 \le l \le K+1 \end{cases}$$

The solution is given as follows:

$$u_{i,l} = \begin{cases} 1, & \text{if} \quad X_i = \hat{z}_l I \\ 0, & \text{if} \quad X_i = \hat{z}_h I,\ h \ne l \\ \dfrac{1}{\sum_{h=1}^{k} \left[\dfrac{d(X_i, \hat{z}_l I)}{d(X_i, \hat{z}_h I)} \right]^{\frac{1}{(\alpha-1)}}}, & \text{if} \quad X_i \ne \hat{z}_l I \text{ and } X_i \ne \hat{z}_h I, \quad 1 \le h \le k. \end{cases} \tag{4}$$

where $d(X_i, \hat{z}_l I)$ is the aggregated distance between the ith gene and the lth centroid, and is given by

$$d(X_i, \hat{z}_l I) = \sum_{j=1}^{m} \hat{w}_{l,j} \hat{d}_{i,j,l}.$$

2.1.2 Problem 2: Fixing $Z = \hat{Z}, U = \hat{U}$ and solving the reduced problem $min_W\ P(\hat{U}, \hat{Z}, W)$

For W:

$$\min\ P(\hat{U}, \hat{Z}, W) = \sum_{l=1}^{K+1} \sum_{i=1}^{N} \sum_{j=1}^{M} \hat{u}_{i,l}^{\alpha} w_{l,j} \hat{d}_{i,j,l} + \beta \sum_{l=1}^{K} \left(\sum_{j=1}^{M-1} |w_{l,j} - w_{l,j+1}| + w_{l,1} + w_{l,M} \right)$$

subject to

$$\begin{cases} \hat{d}_{i,j,l} = (x_{i,j} - \hat{z}_l)^2, & 1 \le i \le N, 1 \le j \le M, 1 \le l \le K \\ \hat{d}_{i,j,K+1} = D, & 1 \le i \le N, 1 \le j \le M \\ \sum_{j=1}^{M} w_{l,j} = 1, & 1 \le l \le K+1 \\ 0 \le w_{l,j} \le 1, & 1 \le j \le M, 1 \le l \le K+1. \end{cases}$$

This optimization problem can be solved using linear programming techniques.

2.1.3 Problem 3: Fixing $U = \hat{U}, W = \hat{W}$ and solving the reduced problem $min_Z\ P(\hat{U}, Z, \hat{W})$

After deriving the parameters during the first two steps, we can update the centroids with the new parameters by solving the problem:

$$\min P(\hat{U}, Z, \hat{W}) = \sum_{l=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{M} \hat{u}_{i,l}^{\alpha} \hat{w}_{l,j} (x_{i,j} - z_l)^2$$

The solution is given as follows:

$$z_l = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{u}_{i,l}^{\alpha} \hat{w}_{l,j} x_{i,j}}{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{u}_{i,l}^{\alpha} \hat{w}_{l,j}}, \quad \text{for } 1 \leq l \leq K \tag{5}$$

2.1.4 Algorithm

Hence, the algorithm is as follows. Note that the gene expression matrix should be normalized before the computation.

- Step 1: Given initial Z and W. We can choose initial Z randomly or with experiences, and $\frac{1}{M}$ for initial $w_{l,j}$.
- Step 2: Solve problem 1 with given Z and W to get a new U.
- Step 3: Solve problem 2 with given Z and U to get a new W.
- Step 4: Solve problem 3 with given W and U to get a new Z.
- Step 5: Repeat steps 2-4 until our objective function is minimized. Now we can obtain the optimal Z, W and U.
- Step 6: Decide biclusters. For any data point $x_{i,j}$, compare the weight $u_{i,l}^{\alpha} w_{l,j}$ between the data point and all the biclusters. This data point belongs to the bicluster P:

$$P = \arg\max_{l}\ u_{i,l}^{\alpha} w_{l,j}$$

Because we decompose the nonlinear optimization problem into several sub-problems, many local optimal solutions will emerge depending on different initial values, especially different Z. A straightforward way to alleviate this problem is that we compute repeatedly with different initial values and choose the solution with the minimal object value.

2.2 Model 2: detect biclusters with similar gene expression profiles

However, in many cases, the important biclusters are those in which the gene profiles are similar. These biclusters reflect more general co-regulatory or co-expression relationships among genes. For this purpose, we modify the first model, i.e. model 1, as follows:

- The centroid of every bicluster is a M vector: $Z_l = [z_{l,1}, z_{l,2}, ..., z_{l,M}]$, and $z_{l,j}$ is the gene expression of the centroid of bicluster l at time j. $Z = \{Z_1, Z_2, ...Z_K\}$ is a set of K vectors representing the centroids of the K biclusters.

- The distance is defined to be the angle between the data point vector and the centroid vector:

$$d'_{i,j,l} = \theta(X_{i,j}, Z_{l,j}) = \frac{180 \times \arccos \frac{\langle X_{i,j} \cdot Z_{l,j} \rangle}{|X_{i,j}| \cdot |Z_{l,j}|}}{\pi}$$

  $X_{i,j} = [x_{i,j-1}, x_{i,j}]$ is the data point vector, $Z_{l,j} = [z_{l,j-1}, z_{l,j}]$ is the centroid vector and $\langle X_{i,j} \cdot Z_{l,j} \rangle$ is the inner product. The distance depicts the similarity of the different gene profiles even if they have different expression values.

Except for the distance expression and the defination of the centroids, the objective function is unchanged. We also decompose the optimization problem into three sub-problems, where the first two sub-problems (Problem 1 and Problem 2) are solved similarly to model 1 except for the different distance expression and the centroids. But the third sub-problem should be solved in a different manner as follows.

2.2.1 Problem $3'$: Fix $U = \hat{U}, W = \hat{W}$ and solve the reduced problem $min_Z P'(\hat{U}, Z, \hat{W})$

For Z:

$$\min\ P'(\hat{U}, Z, \hat{W}) = \sum_{l=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{M} \hat{u}_{i,l}^{\alpha} \hat{w}_{l,j} d'_{i,j,l}$$

Intuitively the derived centroid vector $Z_{l,j}$ is in proportion to a vector that is a linear combination of N data point vectors, and the coefficients of the combination are based on the parameters that we got from the first two sub-problems.

Firstly we unitize the data point vector by $X'_{i,j} = \frac{X_{i,j}}{|X_{i,j}|}$, and the solution is given as follows:

$$Z'_{l,j} = \frac{\sum_{i=1}^{N} \hat{u}_{i,l}^{\alpha} \hat{w}_{l,j} X'_{i,j}}{\sum_{i=1}^{N} \hat{u}_{i,l}^{\alpha} \hat{w}_{l,j}}$$

$$z_{l,j} = \frac{z_{l,j-1}}{z'_{l,j-1}} \times z'_{l,j} \quad \text{for } 1 \le l \le K, 1 \le j \le M \tag{6}$$

The procedure of this model is similar to model 1. This model may also generate large amount of local minimal solutions. Therefore, we repeat the computation by different initial values or adopt other methods of selecting initial values.

3 Result

We tested our models using simulated data. Firstly we generated a synthetic dataset, including three constant value biclusters: biclusters 1 and 2 are up-regulatory modules, where bicluster 3 is a down-regulatory module. There are overlaps between biclusters 1 and 2. To test noise resistance of our method, we embedded the biclusters into a noisy background generated by a uniform distribution $U(1,9)$. Gaussian noise with variance of 0.1 was used to degrade the biclusters. The dataset has 50 rows and 10 columns, where biclusters 1 and 2 are 10×4 matrices, and bicluster 3 is a 15×4 matrix. We chose the

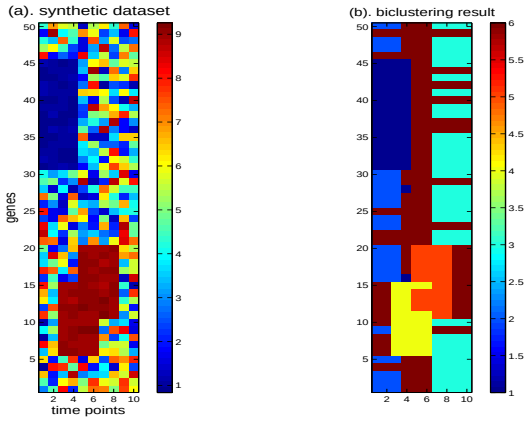

Figure 1: An example for model 1. (a): A synthetic dataset. There are three biclusters in the dataset and we also consider the overlaps and noises. (b): The result of our method. The parameters are: $\beta = 2, D = 4, K = 5$, and the initial centroids are $\{1,3,5,7,9\}$. The elements in the matrix are the labels of biclusters, and the final centroids are $\{1.03, 4.51, 4.93, 8.95, 8.98\}$. We can find bicluster 1 with centroid 1.03 and bicluster 4 and 5 with centroid near 9.

proper parameters ($\beta = 2, D = 4, K = 5$) by trial-and-error method, and used model 1 to detect biclusters. The results are shown in *Fig*.1, which shows that our model can identify all the biclusters.

Then, for testing the computational efficiency of our method, we generated a large gene expression dataset with 10000 genes and 100 time points. The result show that our model can also identify the biclusters correctly in a few minutes.

Finally, we generated a small example to examine the model 2. The dataset is composed of 20 genes and 8 time points. The elements representing the gene expression value which is generated by the uniform distribution $U(1,9)$. Gaussian noise with variance of 0.1 was used to degrade the bicluster. We construct only one bicluster whose 8 genes change similarly within 4 time points. Clearly, our method can identify the bicluster, see *Fig*.2.

4 Discussion

We have tested our models in several simulated datasets. The numerical results show that the models are robust to noises and efficient in large-scale computations.

In our future work, we will focus on applying our model to real microarray data or other high throughput data to identify meaningful biclusters.

Acknowledgements

This work was partially supported by the Chief Scientist Program of Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences with Grant No. 2009CSP002.

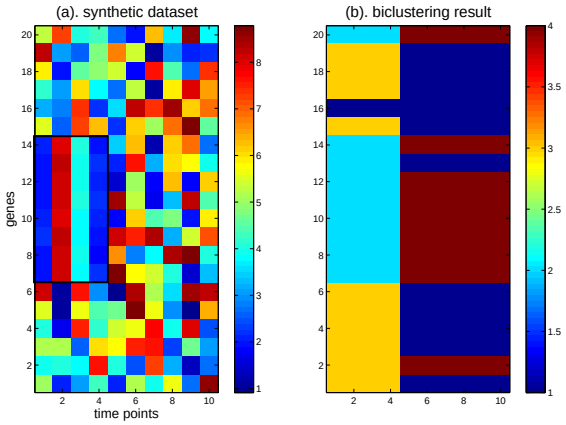

Figure 2: An example for model 2. (a): A synthetic dataset. There is only one bicluster in which gene profile is $[2,8,4,2]$ and we also consider the noises. (b): The result of our method. The parameters are: $\beta = 8, D = 20, K = 3$. We can find bicluster 2 whose centroid is $[2.22, 7.99, 4.02, 2.19]$ within the first four time points.

References

[1] Chen L, Wang R, Zhang X: Biomolecular Network: Methods and Applications in Systems Biology, John Wiley & Sons, 2009.

[2] Chen L, Wang R, Li C, Aihara K: Modelling Biomolecular Networks: Structures and Dynamics. Springer-Verlag, London, 2010.

[3] Cheng Y, Church GM: Biclustering of Expression Data. *In Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology AAAI Press* 2000.

[4] Getz G, Levine E, Domany E: Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences of the United States of America* 2000, 97(22): 12079-12084.

[5] Ihmels J, Bergmann S and Barkai N: Defining transcription modules using large-scale gene expression data. *Bioinformatics* 2004, 20: 1993-2003.

[6] Tanay A, Sharan R, Shamir R: Discovering statistically significant biclusters in gene expression data. *Bioinformatics* 2002, 18(Suppl 1): S136-144.

[7] Amela P, Stefan B, Philip Z: A systematic comparison and evaluation of biclustering methods for gene expression data *Bioinformatics* 2006,22: 1122-1129.

[8] J. C. Bezdek: Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press, New York* 1981.