

A Biclustering Method for Time Series Analysis

Jeonghwa Lee

Department of Industrial and Management Engineering
Pohang University of Science and Technology, Pohang 790-784, KOREA
E-mail: bls83@postech.ac.kr

Youngrok Lee

Department of Industrial and Management Engineering
Pohang University of Science and Technology, Pohang 790-784, KOREA
E-mail: lyr1004@postech.ac.kr

Chi-Hyuck Jun[†]

Department of Industrial and Management Engineering
Pohang University of Science and Technology, Pohang 790-784, KOREA
E-mail: chjun@postech.ac.kr

Received, February 20, 2010; Revised, April 20, 2010; Accepted, May 17, 2010

Abstract. Biclustering is a method of finding meaningful subsets of objects and attributes simultaneously, which may not be detected by traditional clustering methods. It is popularly used for the analysis of microarray data representing the expression levels of genes by conditions. Usually, biclustering algorithms do not consider a sequential relation between attributes. For time series data, however, bicluster solutions should keep the time sequence. This paper proposes a new biclustering algorithm for time series data by modifying the plaid model. The proposed algorithm introduces a parameter controlling an interval between two selected time points. Also, the pruning step preventing an over-fitting problem is modified so as to eliminate only starting or ending points. Results from artificial data sets show that the proposed method is more suitable for the extraction of biclusters from time series data sets. Moreover, by using the proposed method, we find some interesting observations from real-world time-course microarray data sets and apartment price data sets in metropolitan areas.

Keywords: Biclustering, Time-series Data, Plaid Model, Binary Least Square

1. INTRODUCTION

Biclustering is an unsupervised learning method of detecting informative groups of objects and attributes simultaneously. It was proposed by Hartigan (1972) with the term ‘direct clustering’ and the name ‘biclustering’ has been used by Mirkin (1996). Other names such as co-clustering, block clustering, two-mode clustering, two-way clustering, and simultaneous clustering are also used.

Consider a data set consisting of objects and attributes. For example, microarray data have genes as objects and have conditions as attributes. Traditional clustering methods find homogeneous object groups or attribute groups like in Figure 1(a). However, biclustering is a method of finding groups containing partial objects and attributes simultaneously like in Figure 1(b). While traditional clustering methods assume that objects in a same cluster

are similar over the whole attributes, biclustering algorithms consider objects which are similar in only a subset of attributes as members of a bicluster even if they are dissimilar in other attributes. Therefore, by using biclustering methods, we may find useful groups which may be not detected by conventional clustering methods. For example, microarray data analysis needs to find genes corresponding to some conditions. Because biclustering methods meet this purpose, they are widely used in microarray data analysis.

A biclustering method may assume a specific structure and data type. Figure 2 illustrates some biclustering structures defined by Madeira and Oliveira (2004): (a) single bicluster, (b) exclusive rows biclusters, (c) non-overlapping biclusters with tree structure, and (d) arbitrarily positioned overlapping biclusters.

Biclusters can be divided into a continuous value

[†] : Corresponding Author

type and a categorical value type according to the data type. An element value of the i th object and the j th attribute in a continuous value type bicluster can be modeled as an additive model

$$B_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij} \tag{1}$$

or as a multiplicative model

$$B_{ij} = \mu \times \alpha_i \times \beta_j \times \varepsilon_{ij} \tag{2}$$

where B_{ij} denotes the value of the i th object in the j th attribute; μ denotes the overall mean in a bicluster; α_i and β_j denote the effects for the i th object and the j th attribute, respectively, and ε_{ij} represents the random error. In this paper, only additive models are dealt with because a multiplicative model can be transformed into an additive model by logarithmic transformation.

Furthermore, biclusters based on the additive model can be categorized into four types according to the values of α_i and β_j as shown in Figure 3: (a) constant-values bicluster ($\alpha_i = 0, \beta_j = 0$), (b) constant-rows bicluster ($\alpha_i \neq 0, \beta_j = 0$), (c) constant-columns bicluster ($\alpha_i = 0, \beta_j \neq 0$), (d) coherent-values bicluster ($\alpha_i \neq 0, \beta_j \neq 0$).

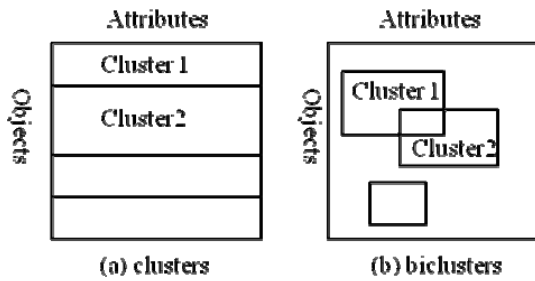


Figure 1. Traditional clustering solution and biclustering solution.

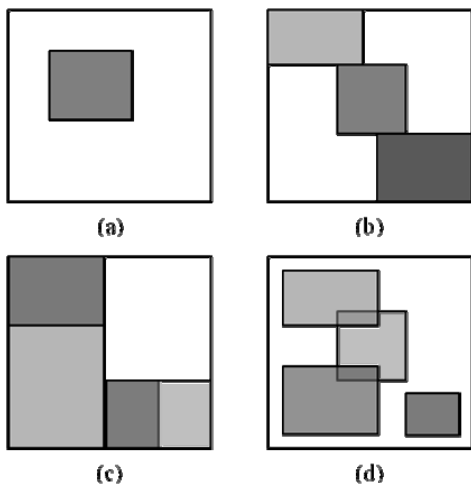


Figure 2. Some biclustering structures.

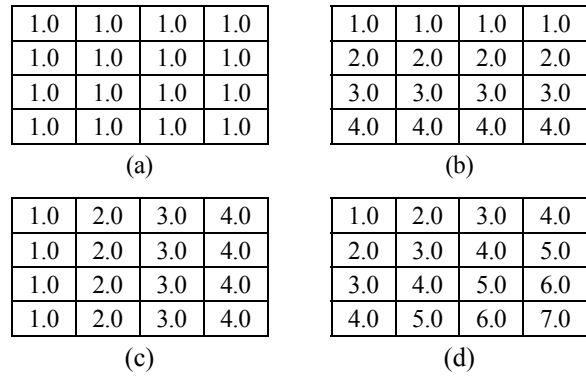


Figure 3. Four types of biclusters.

There are various biclustering algorithms dealing with continuous data based on different assumptions on bicluster structures and types. The biclustering method proposed by Cheng and Church (2000) finds non-overlapped coherent-values biclusters. By allowing biclusters to be overlapped, the plaid model has been proposed by Lazzeroni and Owen (2002). Also, a spectral biclustering algorithm which assumes the constant-values biclusters and a checkerboard structure was introduced by Kluger *et al.* (2003).

1.1 Time Series Data Clustering

Time series data clustering is a method of grouping data showing similar time series patterns together. The three major factors in time series data clustering are data types (continuous or categorical), sampling methods (uniform or not) and equivalence of the number of time points (equal or unequal) (Liao 2005).

Especially, whether the length of each time series is same to each other is an important factor in time series data clustering for the following reasons. First, popular distance metrics such as Euclidian distance, Pearson correlation and Minkowski distance can be used only in the cases that all objects have equal lengths of time series. Moreover, weight vector of SOM (Kohonen, 1990) may not be defined if objects are different in the lengths of time series from each other. Therefore, ordinary clustering algorithms cannot be used in time series of unequal length. Because time series data sets may not always meet the equivalence of time lengths, a number of studies have been made on analyzing those data sets. For example, dynamic time warping (DTW), which uses Markov chain for bending or twisting time so that two objects have the same length, is defined as a distance measure for time series of unequal length (Berndt and Clifford, 1994). DTW is used not only in time series analysis but also in voice recognition.

In addition, model based clustering methods based on ARMA (Auto Regressive Moving Average), MA (Moving Average) and Markov chain are also used in time series data clustering (Liao, 2005).

1.2 Biclustering of Time Series Data

Time series biclustering also has the same purpose as time series clustering, but the similarity may be confined to partial time points instead of the whole time span. In bioinformatics, conventional studies in microarray data have not considered time series pattern even though time-dependent information is included there. Time series microarray data usually contain less than 20 time points because of expensive experimental costs (Ernst *et al.*, 2005). Moreover, the equivalence of time lengths is generally guaranteed in microarray data. Therefore, as discussed in Section 1.1, general clustering methods could be used in analyzing time series microarray data.

In the past few years, however, only several articles have been devoted to the study of biclustering for time series data. Zhang *et al.* (2005) proposed a time series biclustering algorithm by allowing the node deletion and addition procedures in Cheng and Church (2000) only on the edge of time points of a bicluster. Also, Madeira and Oliveira (2005) proposed a biclustering algorithm for categorical time series data. By categorizing gene expression levels into over-expression, non-expression and under-expression, they find subsets showing homogeneous sequences.

Our goal is to develop a method of grouping similar time series having a unique pattern over a certain time period. We propose a modified biclustering algorithm for time series based on the plaid model. In this paper, we assume that a bicluster containing similar time series has coherent values as in Figure 3(d). Unlike other biclustering methods, the proposed method extracts biclusters having consecutive time attributes.

The rest of this paper is organized as follows: basic concepts of the plaid model are reviewed in Section 2; a new time series biclustering method is proposed in Section 3; the experimental results and discussion are reported in Section 4; and we conclude in Section 5.

2. PLAID MODEL

The plaid model is one of the most popularly used biclustering methods (Lazzeroni and Owen 2002), which assumes additive coherent-values biclusters as shown in Figure 3(d). In the plaid model, the term ‘layer’ is used instead of ‘bicluster.’ Let Z_{ij} denote an observed value of the i th object and the j th attribute. Then, the plaid model assumes that Z_{ij} follows as

$$\begin{aligned} Z_{ij} &= \sum_{k=0}^K \Theta_{ijk} \rho_{ik} \kappa_{jk} + \varepsilon_{ij} \\ &= \sum_{k=0}^K (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} \kappa_{jk} + \varepsilon_{ij} \end{aligned} \quad (3)$$

where Θ_{ijk} denotes an element value of i th object and j

th attribute in the k th bicluster; ρ_{ik} is a binary membership variable that is equal to one when the i th object belongs to the k th bicluster; κ_{jk} is another binary variable whose value is one when the j th attribute is in the k th bicluster. When $k = 0$, Θ_{ijk} represents the value of the background. Θ_{ijk} is modeled as summation of a mean value of the k th bicluster (μ_k), the i th object effect (α_{ik}), and the j th attribute effect (β_{jk}). We call (μ_k , α_{ik} , β_{jk}) layer effects.

Lazzeroni and Owen (2002) compares with the plaid model and other methods such as analysis of variance (AN OVA), singular value decomposition (SVD), semi-discrete decomposition (SDD) and other clustering and biclustering methods.

An optimization problem in the plaid model can be formulated by

$$\begin{aligned} \text{Minimize} \quad & \sum_i \sum_j \left(Z_{ij} - \sum_{k=0}^K \Theta_{ijk} \rho_{ik} \kappa_{jk} \right)^2 \\ \text{s.t.} \quad & \rho_{ik}, \kappa_{jk} = 0, 1 \quad \forall i, j, k \end{aligned} \quad (4)$$

Because Eq. (4) is hard to be solved directly, Lazzeroni and Owen (2002) proposed a method of extracting each layer sequentially from the background layer ($k = 0$) to the K th layer by solving the problem below:

$$\begin{aligned} \text{Minimize} \quad & \sum_i \sum_j \left(Z_{ij}^{(k)} - \Theta_{ijk} \rho_{ik} \kappa_{jk} \right)^2 \\ \text{s.t.} \quad & \rho_{ik}, \kappa_{jk} = 0, 1 \quad \forall i, j, k \end{aligned} \quad (5)$$

where $Z_{ij}^{(k)} = Z_{ij} - \sum_{m=0}^{k-1} \Theta_{ijm} \rho_{im} \kappa_{jm}$ for $k = 1, \dots, K$.

Even though the problem in Eq. (5) is simpler than Eq. (4), it is still NP-hard because Θ_{ijk} is determined by ρ_{ik} and κ_{jk} , and vice versa. Therefore, decision variables in Eq. (5) should be estimated through an iterative procedure. Lazzeroni and Owen (2002) adopted LP relaxation to obtain the decision variables. On the other hand, Turner *et al.* (2005) solved the problem without LP relaxation. The rest of this section briefly describes Turner’s algorithm.

2.1 Background Fitting

Because the purpose of the plaid model is detecting unusual patterns from the given data, a common pattern belonging to all data should be eliminated before the search for the first bicluster. Elements of the background layer are estimated as

$$\begin{aligned} \mu_0 &= Z_{..} \\ \alpha_{i0} &= Z_{i.} - \mu_0 \\ \beta_{j0} &= Z_{.j} - \mu_0 \\ \Theta_{ij0} &= \mu_0 + \alpha_{i0} + \beta_{j0} \end{aligned} \quad (6)$$

where Z denotes the mean value of all elements in the given data set Z , Z_i denotes the mean element value of i th object, and Z_j denotes the mean element value of j th attribute. For the background layer, all membership variables, ρ_{j0} and κ_{j0} , are set to one. Then, data is updated to $Z^{(1)}$ by subtracting the background effects as follows:

$$Z_{ij}^{(1)} = Z_{ij} - \Theta_{ij0}\rho_{i0}\kappa_{j0} \quad (7)$$

In general, data is updated as follow after $(k-1)$ st bicluster is obtained

$$Z_{ij}^{(k)} = Z_{ij} - \sum_{m=0}^{k-1} \Theta_{ijm}\rho_{im}\kappa_{jm} \quad (8)$$

2.2 Finding Biclusters

In Turner's algorithm, biclusters are sequentially extracted from data and the k th bicluster may be found from $Z^{(k)}$ in Eq. (8).

First, 2-means object clustering and 2-means attribute clustering are performed to initialize membership variables ρ_{ik} and κ_{jk} . If i th object belongs to the smaller cluster of 2-means object cluster solution, set $\rho_{ik} = 1$. Also, if the j th attribute is in the smaller one of two attribute clusters, set $\kappa_{jk} = 1$. This initialization implies that a smaller cluster represents a more unusual pattern than a larger cluster so that the elements of a smaller cluster are more likely to be a member of a bicluster.

After the initialization of ρ_{ik} and κ_{jk} , a variable set $(\mu_k, \alpha_{ik}, \beta_{jk})$ is estimated as

$$\begin{aligned} \mu_k &= \frac{1}{n_k m_k} \sum_{i:\rho_{ik}=1} \sum_{j:\kappa_{jk}=1} Z_{ij}^{(k)} \\ \alpha_{ik} &= \frac{1}{m_k} \sum_{j:\kappa_{jk}=1} Z_{ij}^{(k)} - \mu_k \\ \beta_{jk} &= \frac{1}{n_k} \sum_{i:\rho_{ik}=1} Z_{ij}^{(k)} - \mu_k \end{aligned} \quad (9)$$

where $n_k = \sum_i \rho_{ik}$ and $m_k = \sum_j \kappa_{jk}$.

The next step is updating ρ_{ik} and κ_{jk} . To solve the minimization problem in Eq. (5) based on previously estimated variables, both zero and one are assigned to ρ_{ik} in Eq. (10) and ε_{ij} is evaluated in both cases. Then, the value of ρ_{ik} which minimizes ε_{ij}^2 is selected. Also, same procedure is applied to κ_{jk} in Eq. (11). This optimization method is called 'binary least squares.'

$$Z_{ij}^{(k)} = \rho_{ik}(\kappa_{jk}(\mu_k + \alpha_{ik} + \beta_{jk})) + \varepsilon_{ij} \quad (10)$$

$$Z_{ij}^{(k)} = \kappa_{jk}(\rho_{ik}(\mu_k + \alpha_{ik} + \beta_{jk})) + \varepsilon_{ij} \quad (11)$$

Estimation of $(\mu_k, \alpha_{ik}, \beta_{jk})$ and (ρ_{ik}, κ_{jk}) is performed iteratively until each of them converges.

2.3 Pruning

In order to prevent inappropriate objects and attributes from belonging to a bicluster obtained in Section 2.2, the plaid model includes a pruning procedure of removing them from a bicluster. If the i th object currently in the k th bicluster does not satisfy the following condition, ρ_{ik} is set to zero.

$$\sum_{j:\kappa_{jk}=1} (Z_{ij}^{(k)} - \hat{\Theta}_{ijk})^2 < (1 - \tau_1) \sum_{j:\kappa_{jk}=1} (Z_{ij}^{(k)})^2 \quad (12)$$

In Eq. (12), τ_1 denotes a pruning parameter for objects which lies on $(0, 1)$. Similarly, for the j th attribute in the k th bicluster which does not satisfy the following condition, κ_{jk} is set to zero.

$$\sum_{i:\rho_{ik}=1} (Z_{ij}^{(k)} - \Theta_{ijk})^2 < (1 - \tau_2) \sum_{i:\rho_{ik}=1} (Z_{ij}^{(k)})^2 \quad (13)$$

where τ_2 denotes a pruning parameter for attributes which lies on $(0, 1)$.

Once objects and attributes are pruned out, $(\mu_k, \alpha_{ik}, \beta_{jk})$ are re-estimated and remaining objects and attributes are pruned out again with the new values of $(\mu_k, \alpha_{ik}, \beta_{jk})$. The pruning is performed iteratively until there is no change in ρ_{ik} and κ_{jk} . After the pruning procedure, a data set $Z^{(k+1)}$ is constructed similarly in Eq. (8).

2.4 Back Fitting

Sometimes, layers previously extracted need to be refined based on the currently extracted layer. For this purpose, the plaid model performs back fitting. From the background layer to k th layer, back fitting is performed sequentially.

For the r th layer $(0 \leq r \leq k)$, a data set $Z^{(r)}$ is re-computed as

$$Z_{ij}^{(r)} = Z_{ij}^{(k+1)} + \Theta_{ijr}\rho_{ir}\kappa_{jr} \quad (14)$$

Then, using the predetermined ρ_{ir} and κ_{jr} , $(\mu_r, \alpha_{ir}, \beta_{jr})$ are re-estimated from $Z^{(r)}$ by using Eq. (9). Finally, $Z^{(k+1)}$ is updated as

$$Z_{ij}^{(k+1)} = Z_{ij}^{(r)} - \Theta_{ijr}\rho_{ir}\kappa_{jr} \quad (15)$$

After back fitting is performed for all layers, $(k+1)$ st bicluster is searched from a data set $Z^{(k+1)}$.

3. PROPOSED METHOD

We propose a modified plaid method which is designed to be appropriate to time series data. The proposed method is based on Turner's algorithm, so the model of the observed data is same as in Eq. (3). The original Turner's algorithm produces biclusters without any constraint on the sequence of attributes.

In time series analysis, however, a pattern over time sequence gives important information to us. Therefore, we propose a biclustering algorithm of finding biclusters whose member objects show a similar pattern over a consecutive time period. To keep the time sequence, we assume that results of biclustering satisfy the following conditions:

$$\begin{aligned} & \text{for } \kappa_{jk}, j=1, 2, \dots, n \\ & \text{if } \kappa_{pk}=1, \kappa_{qk}=1 \text{ and } p < q \\ & \text{then } \kappa_{(p+1)k} = \dots = \kappa_{(q-1)k} = 1 \end{aligned} \quad (16)$$

We add this constraint to Eq. (4) and Eq. (5). Under this constraint, we can regard the attribute effect (B_{ik}) as pattern and the object effect (α_{ik}) as shift scale.

We use an iterative method when we find a bicluster. A detailed method is mentioned in Section 3.1. We extract a bicluster one after another as Turner's algorithm does. Section 3.1, 3.2 and 3.3 explain the difference between the proposed method and Turner's algorithm.

3.1 Finding a Bicluster

Layer effects, $(\mu_k, \alpha_{ik}, \beta_{jk})$, and object membership parameter, ρ_{ik} , are updated according to the iterative procedure described in Section 2.2. However, because κ_{jk} should satisfy the condition in Eq. (16), we update attribute membership parameters in a different way from Eq. (11).

We define a new parameter C controlling the connection of two selected time points. If $\kappa_{pk}=1$, $\kappa_{qk}=1$ and $q-p \leq C$, then $\kappa_{(p+1)k}, \dots, \kappa_{(q-1)k}$ are set to 1. If more than two separate time sequences appear in the same layer, we choose the biggest one. A value of C should be pre-determined. If $C=1$, κ_{jk} is obtained according to Eq. (11). If $C \geq 2$, we may obtain different biclusters from results of Turner's algorithm. As C becomes larger, the size of a bicluster may increase. Therefore, too large C may lead to biclusters with unclear time series pattern. From results of several experiments which will be shown in Section 4.2, we recommend 2 or 3 as a value of C , especially in cases that more distinct time series patterns is needed and the size of data set is smaller.

The parameter of C can also play a role of treating missing values or unexpected data. Experimental results in Section 4.2 show that the proposed method finds bet-

ter biclusters than Turner's algorithm when data has missing values.

3.2 Pruning

In the proposed method, pruning of attributes is performed only at the boundaries of a bicluster through two-step procedure, whereas the pruning procedure of objects is the same as Section 2.3.

First, we find the smallest j that $\kappa_{jk}=1$. Then, we decide whether the j th attribute should be pruned away or not by using Eq. (13). If κ_{jk} becomes 0, j th attribute is pruned away from k th bicluster. Otherwise, the first pruning step is terminated. The pruning is repeated until we find an attribute which is not pruned out, while increasing j by one.

In the second step, we find the largest j that $\kappa_{jk}=1$. Then, we prune away attributes as the first step while decreasing j by one. As a result, a bicluster is produced while keeping a consecutive time sequence.

If all objects or all attributes are pruned out, the algorithm is terminated. Also, if only one object or one attribute remains, the algorithm is terminated because we define that those biclusters are meaningless.

3.3 Starting Values

To update values of μ_k , α_{ik} and β_{jk} , we should set the initial values of ρ_{ik} and κ_{jk} . Turner *et al.* (2005) proposed the initialization by two-means clustering; if ρ_{ik} or κ_{jk} belongs to the smaller object or attribute cluster, it is set to one and it is set to zero, otherwise. However, if the true bicluster contains more than the half of the objects or attributes, the initial values may be poor to obtain a good bicluster. In addition, two-means clustering does not guarantee that κ_{jk} satisfies the conditions shown in Eq. (16).

For these reasons, we propose a new initialization method. At first, we sort differences between Z_i 's and $Z_{..}$ in descending order. In the next step, ρ_{ik} 's belonging to the top $\psi\%$ objects of the sorted list are set to 1 where ψ is a random number generated from uniform distribution between 0 and 100. Similarly, we sort difference between Z_i 's and $Z_{..}$ in descending order. In the case of attributes, after selecting the top 10% attributes from the list, we set all κ_{jk} 's lied between the minimum and maximum time points of the selected attributes to one.

4. NUMERICAL EXPERIMENTS

To validate the accuracy of the proposed method, we conducted experiments with synthetic data and some real data sets. Section 4.1 introduces some validation indices for biclustering that are used in the experiments. Some other validation indices are available in the literature as in Lee *et al.* (2009). Section 4.2 shows properties

of the proposed method with synthetic data. Section 4.3 and Section 4.4 show results obtained by applying the proposed method to real data.

4.1 Validation Index

Let M_1 and M_2 be two different biclustering results, consisting of K_1 and K_2 biclusters, respectively, obtained from a data set. Also, let $B_k^{(j)}$ be the k th bicluster of M_j , which consists of an object set $O_k^{(j)}$ and an attribute set $A_k^{(j)}$ for $j=1, 2$. Then, M_j can be defined as

$$M_j = \left\{ B_1^{(j)}, B_2^{(j)}, \dots, B_{K_j}^{(j)} \right\} \\ = \left\{ \left(O_1^{(j)}, A_1^{(j)} \right), \left(O_2^{(j)}, A_2^{(j)} \right), \dots, \left(O_{K_j}^{(j)}, A_{K_j}^{(j)} \right) \right\} \quad (17)$$

As the validation index we will use the S index by Santamaría *et al.* (2007), which is based on F_1 index. F_1 index is a measure defined to evaluate the similarity between two biclusters (Turner *et al.*, 2005), which is defined as

$$F_1 \left(B_k^{(1)}, B_m^{(2)} \right) = \frac{2 \left| O_k^{(1)} \cap O_m^{(2)} \right| \left| A_k^{(1)} \cap A_m^{(2)} \right|}{\left| O_k^{(1)} \right| \left| A_k^{(1)} \right| + \left| O_m^{(2)} \right| \left| A_m^{(2)} \right|} \quad (18)$$

Santamaría *et al.* (2007) defines the S index to measure the similarity between two biclustering results using F_1 index as follows:

$$S(M_1, M_2) = \frac{1}{K_1} \sum_{i=1}^{K_1} \max_{j=1}^{K_2} F_1 \left(B_i^{(1)}, B_j^{(2)} \right) \quad (19)$$

Values of both F_1 index and S index lie between 0 and 1. The F_1 index is set to 1 if two biclusters are exactly same, while it is set to 0 if two biclusters are mutually exclusive. Therefore, the S index defined in Eq. (19) is set to 1 if all biclusters in M_1 exactly reappear in M_2 .

Because of the 'max' operation, the S index is an asymmetric measure, which means that values of $S(M_1, M_2)$ and $S(M_2, M_1)$ may be unequal. Let T be the true bicluster solution and E be an obtained bicluster solution. Then, $S(T, E)$ represents how well the true biclusters are found by a bicluster algorithm, while $S(E, T)$ represents how reliably an obtained bicluster can be regarded as the true cluster.

In cases that prior knowledge about only objects categorization is available, the purity and the efficiency proposed by Getz *et al.* (2000) can be adopted to evaluate each obtained bicluster. Let t_{ik} be the number of i th category objects in B_k and T_i be the total number of i th category objects. Then, the purity and the efficiency of B_k can be defined as follows:

$$\text{purity}(B_k) = \max_i \frac{t_{ik}}{|B_k|} \quad (20)$$

$$\text{efficiency}(B_k) = \max_i \frac{t_{ik}}{|T_i|} \quad (21)$$

4.2 Experiment with Artificial Data Sets

We generated several synthetic data sets to verify the proposed method. Each data set consists of 100 objects and 50 time points, and it has two true time series biclusters consisting of 40 objects and 20 time points each. Data are generated to follow the model in Eq. (3). β_{jk} 's were generated to have a moving average process as follows:

$$\beta_{jk} = \gamma_{jk} - \phi \times \gamma_{j(k-1)}, \quad \gamma_{jk} \sim N(0, 0.5^2) \quad (22)$$

Also, μ_k 's and α_{ik} 's were generated from uniform distribution between 0 and 1, and a random noise ε_{ij} 's were generated from $N(0, 0.5^2)$.

We generated 11 data sets having different overlapping ratios of two biclusters such as: 0, 1², 2², 3², ..., 9², 10²%. Figure 4 illustrates data sets whose overlapping ratios are (a) 0% and (b) 49%.

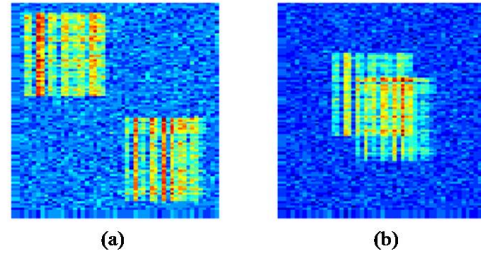


Figure 4. Synthetic data with overlapping ratios (a) 0% and (b) 49%.

The maximum number of biclusters to be found was set to 5, and the pruning parameters τ_1 and τ_2 were set to 0.5. We repeated the proposed method 30 times for each value of $C \in \{2, 3, 4\}$. Also, we obtained bicluster solution for each data set using Turner's algorithm for the comparison purpose.

As a result, the proposed method obtained 2.7, 3.5 and 3.4 biclusters on the average for $C = 2, 3$ and 4, respectively, whereas Turner's algorithm yielded 3.7 biclusters. As shown in Figure 5 and 6, both $S(T, E)$ and $S(E, T)$ were low when the overlapping ratio was around 20%. This can be interpreted that the overlapped area might be considered as an independent bicluster when about half of the two biclusters were overlapped. The proposed method, however, shows better results than Turner's algorithm when the overlapping ratio is around 20%. As shown in Figure 5, C has little effect on the values of $S(T, E)$. On the other hand Figure 6 shows, that larger C ($C = 3$ or 4) has better values of $S(E, T)$.

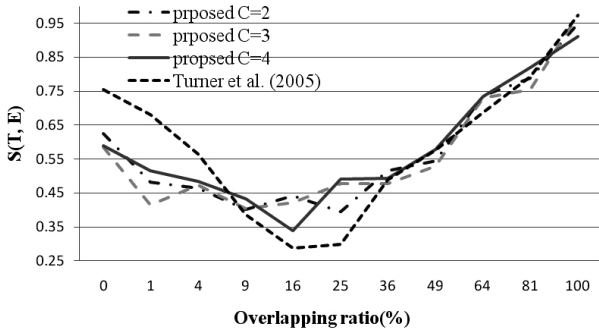


Figure 5. $S(T, E)$ value according to overlapping ratios.

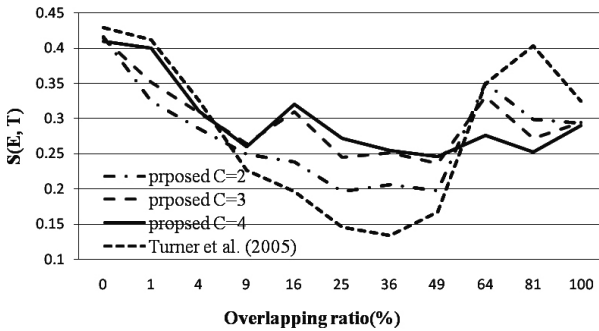


Figure 6. $S(E, T)$ value according to overlapping ratios.

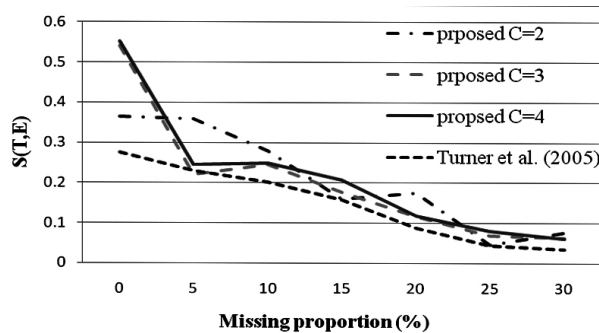


Figure 7. $S(T, E)$ value against missing proportions.

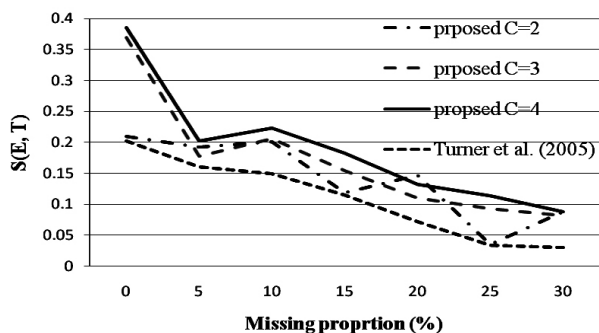


Figure 8. $S(E, T)$ value against missing proportions.

To more deeply investigate the effects of C on bicluster solutions, we discarded some data from each true bicluster. Increasing the missing proportion from 0% to

30% with increments of 5%, we performed bicluster analysis using the proposed method with each value of C . Also, we obtained biclustering results for each data set using Turner’s algorithm. The overlapping ratio was fixed at 16%. We replaced missing values with Θ_{ij0} using Eq. (6).

Figures 7 and 8 illustrate that as the missing proportion increases $S(T, E)$ and $S(E, T)$ decrease. However, the proposed method shows better performance than the Turner’s algorithm when missing values exist. The interval factor C has a little effect on the biclustering results in this experiment.

4.3 Analysis with Yeast Data

The Yeast data we used is a gene expression data which describes genome-wide mRNA transcript levels during the cell cycle of the budding yeast ‘*Saccharomyces cerevisiae*’ (Cho *et al.*, 1998). Categorization of 384 genes of Yeast data into five time series groups have been provided by Yeung *et al.* (2000).

Table 1. Biclustering result for yeast data.

	Proposed $C=2$	Proposed $C=3$	Proposed $C=4$
Mean # of biclusters	4.05	3.45	3.65
Purity	0.466 ± 0.073	0.467 ± 0.078	0.475 ± 0.089
Efficiency	0.762 ± 0.101	0.760 ± 0.106	0.754 ± 0.112

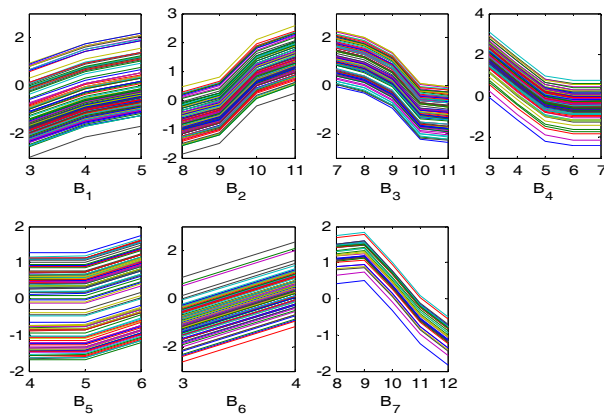


Figure 9. A bicluster solution of yeast data.

The maximum number of biclusters to be found was set to 10, and the pruning parameters τ_1 and τ_2 were set to 0.5. We repeated bicluster analysis 20 times for $C \in \{2, 3, 4\}$. Turner’s algorithm does not find any biclusters. As a result, 4.0, 3.5 and 3.7 biclusters were obtained on the average for $C = 2, 3$ and 4, respectively. Table 1 describes that the interval factor C has a negligible effect on the biclustering result. Figure 9 shows

that each bicluster found by the proposed method has a homogeneous pattern.

4.4 Apartment data in Metropolitan area

We used the monthly data of apartment prices from July 2003 to November 2006 in Korean metropolitan area. The towns considered are Gwacheon (GC), Gangnam (GN), Seocho (SC), Songpa (SP), Bundang (BD) and Gangbuk (GB). The sizes of apartments were transformed into five discrete levels as shown in Table 2. During the period, apartment prices of the selected towns had increased by over 40% except Gangbuk. Because there were only a few apartments whose sizes were level 5 in Gwacheon, the apartment prices were discarded from the data set. Therefore, the final data set has 29 pairs of town and size as objects and 41 time points as attributes. We obtained this data to summarize apartment price information from Kookmin Bank, one of the largest banks in Korea.

Using the proposed method, we tried to find a group of towns which shows a homogeneous pattern of apartment prices in a certain period. The maximum number of biclusters to be found was set to 10, and the pruning parameters τ_1 and τ_2 were set to 0.7. Also, we fixed the value of C at 2.

We repeated bicluster analysis 20 times, and obtained 4.95 biclusters on the average. Table 3 summarizes biclustering solution from the first to the fifth bicluster.

Table 2. Categorization of the apartment size.

Level	size (m ²)
5	>165
4	(132, 165)
3	(99, 132)
2	(66, 99)
1	≤ 66

Table 3. Biclustering result for apartment data.

bicluster	(Town, Level)	Time period
B ₁	(GC, 2), (GC, 3) (GC, 4), (GC, 4)	Jun. 05~Nov. 06
B ₂	All pairs except GB	Jul. 03~Nov. 06
B ₃	(GB, 1), (GC, 2) (GB, 2), (BD, 3) (GB, 3), (BD, 4) (GB, 4), (BD, 5) (GB, 5)	Jul. 03~Nov. 06
B ₄	(GN, 1), (SC, 1) (SP, 1), (BD, 1) (SC, 2), (SP, 2) (BD, 2)	Jan. 06~Nov. 06
B ₅	Same as B ₄	Jun. 03~Mar. 05

The first bicluster includes Gwacheon with all sizes and time period from Jun 2005 to November 2006. As shown in Figure 10, apartment prices of Gwacheon had increased rapidly after June 2005. Hence, we conclude that the first bicluster groups apartments which had been experienced dramatic growths of prices.

On the other hand, the second bicluster includes all towns except Gangbuk, whole time period and all levels. The reason why Gangbuk is excluded from the second bicluster is that apartment prices in Gangbuk were much lower than other areas.

The fourth and fifth biclusters consist of apartments of level 1 and 2 in all towns except Gwacheon and Gangbuk. As illustrated in Figure 11, apartment prices of Gangnam, Seocho, Songpa and Bundang moves together. Also, we can find that the apartment prices begin to increase in the middle of 2005.

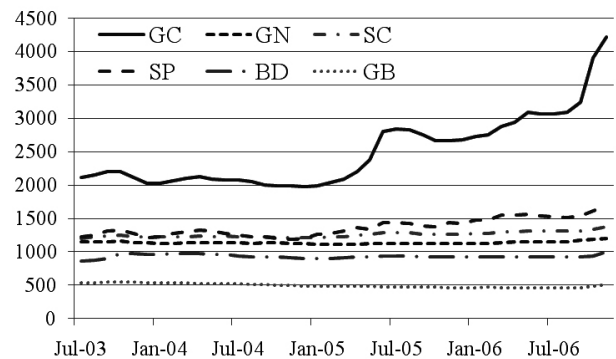


Figure 10. Apartment prices in Level 1.

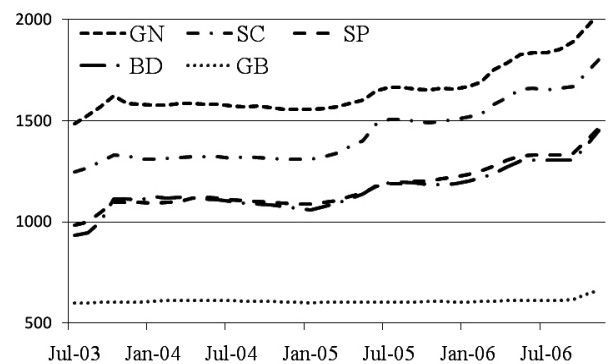


Figure 11. Apartment price in Level 2 except Gwacheon.

Table 4. K-means clustering result for apartment data.

cluster	(Town, Level)
1	(GN, 2), (SC, 3), (BD, 3), (SP, 4), (BD, 4), (BD, 5)
2	(GN, 1), (SC, 1), (SP, 1), (SC, 2), (SP, 2), (BD, 2), (SP, 3)
3	(BD, 1), (GB, 1), (GB, 2), (GB, 3), (GB, 4), (GB, 5)
4	(GC, 3), (GN, 3), (GC, 4), (SC, 4), (SC, 5), (SP, 5)
5	(GC, 1), (GC, 2), (GN, 4), (GN, 5)

Table 4 summarizes 5-means clustering results of the Apartment data. Even though the second cluster in Table 4 is similar to the fourth and fifth bicluster in Table 3, biclusters provide more information about the change in the increasing rate of apartment prices.

4.5 Discussion

Some other observations from this experiment were made as below. The results may be dependent on the magnitude of data values. In apartment price data, Gwachon (GC) whose price is most expensive is extracted as the first layer while Gangbuk (GB) whose price is lowest is extracted as the third layer. In this case, magnitude is important information. However, patterns are usually more important than magnitude in some other data, therefore using normalized data may yield better results. Normalization or scaling of data can be made according to the purpose of analysis.

A method to initialize membership variables, ρ_{ik} and κ_{jk} , may effect results, particularly in Turner's algorithm. Turner's algorithm showed worse performance when the overlapping ratio is between 10 to 50% as shown Figure 5 and 6, which seems to be caused by the initialization. The initialization method of the proposed method, however, performs better when biclusters overlap with each other or have larger values than the background.

5. CONCLUSION

We proposed a new biclustering algorithm which is more suitable to extract time series biclusters than the existing bicluster analysis methods. By introducing an interval parameter to the plaid model and by modifying the pruning procedure of Turner's algorithm, we could obtain biclusters having sequential time points. The proposed method can find time series patterns which may not be detected by the original plaid model.

We applied the proposed method to some synthetic data, microarray data and monthly apartment price data. It is the first research for time series analysis of apartment price data using a biclustering method, as far as we know. We observed several meaningful biclusters using the proposed method. We have found that pattern of bicluster showed significant meaning. We could observe inflection point or peak point in a bicluster solution of yeast data. We also inferred that significant patterns exist in biclustering result of apartment data.

Since apartment prices are related to other economic data such as consumer price index and interest rates, we are planning to apply the proposed method to other economic time series data in future works. Furthermore, we may be able to extract new knowledge by investigating results of the proposed method combining various time series analysis technique.

ACKNOWLEDGMENT

This work was supported by Basic Science Research Program through the NRF funded by the MEST. (Project No. 2009-0072598).

REFERENCES

- Berndt, D. J. and Clifford, J. (1994), Using dynamic time warping to find patterns in time series, *Association for the Advancement of Artificial Intelligence Technical Report*, WS-94-03, 359-370.
- Cheng, Y. and Church, G. (2000), Biclustering of expression data, *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 93-103.
- Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., and Davis, R. (1998), A genome-wide transcriptional analysis of the mitotic cell cycle, *Molecular Cell*, **2**, 65-73.
- Ernst, J., Nau, G. J., and Bar-Joseph, Z. (2005), Clustering short time series gene expression data, *Bioinformatics*, **21**, i159-i16.
- Getz, G., Levine, E., and Domany, E. (2000), Coupled two-way clustering analysis of gene microarray data, *The Proceedings of the National Academy of Sciences of the United States of America*, 12079-12084.
- Hartigan, J. (1972), Direct clustering of a data matrix, *Journal of the American Statistical Association*, **37**, 123-129.
- Kluger, Y., Basri, R., Ghang, J., and Gerstein, M. (2003), Spectral biclustering of microarray data: Coclustering genes and conditions, *Genome Research*, **13**, 703-716.
- Kohonen, T. (1990) The self organizing maps, *Proceeding IEEE*, **78**, 1464-1480.
- Lazzeroni, L. and Owen, A. (2002), Plaid models for gene expression data, *Statistica Sinica*, **12**, 61-86.
- Lee, Y., Lee, J., and Jun, C.-H. (2009), Validation measures of bicluster solutions, *Industrial Engineering and Management Systems*, **8**, 101-108.
- Liao, T. W. (2005), Clustering of time series data—a survey, *Pattern Recognition*, **38**, 1857-1874.
- Liu, J. and Wang, W. (2003), OP-Cluster: clustering by tendency in high dimensional space, *Proceeding, Third IEEE International Conference, Data Mining*, 187-194.
- Madeira, S. and Oliveira, A. (2004), Biclustering Algorithms for Biological Data Analysis: A Survey, *IEEE Transactions on Computational Biology and Bioinformatics*, **1**, 24-45.
- Madeira, S. and Oliveira, A. (2005), A linear time biclustering algorithm for time series gene expression data, *Lecture Notes in Computer Science, Springer Berlin*, 39-52.
- Mirkin, B. (1996), *Mathematical classification and clustering*, Kluwer Academic Publish.

- Santamaría, R., Quintales, R. and Theorón, R. (2007), Method to bicluster validation and comparison in microarray data, *Intelligent Data Engineering and Automated Learning-Ideal 2007: 8th International Conference*, Birmingham, Uk, Proceedings, 780-789,
- Turner, H., Bailey, T., and Krzanowski, W. (2005), Improved biclustering of microarray data demonstrated through systematic performance tests, *Computational Statistics and Data Analysis*, **48**, 235-254.
- Yeung, K. Y., Haynor, D. R., and Ruzzo, W. L. (2000), Validating clustering for gene expression Data, *Technical Report, Department of Computer Science and Engineering, University of Washington*.
- Zhang, Y., Zha, H., and Chu, C. (2005), A Time-series biclustering algorithm for revealing co-regulated genes. Proceedings of the International Conference on Information Technology, Coding and Computing, **1**, 32-37.