Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Biclustering of ARMA time series*

Jeonghwa LEE, Chi-Hyuck JUN‡

(*Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, Gyungbuk, Korea*)

E-mail: bls83@postech.ac.kr; chjun@postech.ac.kr

Received Oct. 28, 2010; Revision accepted Oct. 29, 2010; Crosschecked Oct. 29, 2010

Abstract: Biclustering is a method of grouping objects and attributes simultaneously in order to find multiple hidden patterns. When dealing with a long time series, there is a low possibility of finding meaningful clusters of whole time sequence. However, we may find more significant clusters containing partial time sequence by applying a biclustering method. This paper proposed a new biclustering algorithm for time series data following an autoregressive moving average (ARMA) model. We assumed the plaid model but modified the algorithm to incorporate the sequential nature of time series data. The maximum likelihood estimation (MLE) method was used to estimate coefficients of ARMA in each bicluster. We applied the proposed method to several synthetic data which were generated from different ARMA orders. Results from the experiments showed that the proposed method compares favorably with other biclustering methods for time series data.

Key words: Biclustering, Time series, Autoregressive moving average (ARMA), Maximum likelihood estimation (MLE)
doi:10.1631/jzus.A1001334 **Document code:** A **CLC number:** TP391

1 Introduction

Clustering is an unsupervised learning method of grouping homogeneous data and separating heterogeneous data. Traditional clustering methods find homogeneous object groups or attribute groups as shown in Fig. 1a. However, biclustering is a method which finds groups having partial objects and attributes simultaneously as shown in Fig. 1b (Lee *et al*., 2010). If partial attributes show a similar pattern, then biclustering will group these attributes even if other parts show a different pattern.

Time series data clustering is a method of grouping data together showing similar time series (Lee *et al*., 2010). Distance measure between two time series is important in time series clustering. Popular distance measures, such as the Euclidian distance, Pearson correlation, and Minkowski distance, can be used only in the cases that all objects have equal lengths of time series (Liao, 2005). On the other hand, dynamic time warping (DTW) is defined as a distance measure for time series of unequal length (Berndt and Clifford, 1994). Keogh and Kasetty (2004) proposed a time series clustering method using a Fourier transform. Model-based clustering methods, based on autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA), are also used in time series data clustering (Kalpakis *et al*., 2001; Xiong and Yeung, 2004).

Objects usually do not exhibit high correlation over the whole time interval, but they may often show a highly correlated pattern over partial intervals (Zhang *et al*., 2005). Thus, a biclustering algorithm can find more homogeneous groups than a traditional time series clustering method.

In the past few years, several methods have been proposed for the biclustering of time series data (Lee *et al*., 2010). Zhang *et al*. (2005) proposed a time series biclustering algorithm based on Cheng and Church (2000) which considered inherent sequential relationship between crucial time points. Madeira and Oliveira (2005) investigated a biclustering algorithm

‡ Corresponding author
* Project (No. 2010-0016800) supported by the Basic Science Research Program through the National Research Foundation (NRF) funded by the Ministry of Education, Science and Technology, Korea
© Zhejiang University and Springer-Verlag Berlin Heidelberg 2010

for categorical time series data using a suffix tree. Zhao and Zaki (2005) studied a clustering algorithm considering 3D gene expression datasets. Xu *et al.* (2009) proposed a 3D cluster model considering time lags between correlated gene expression patterns.

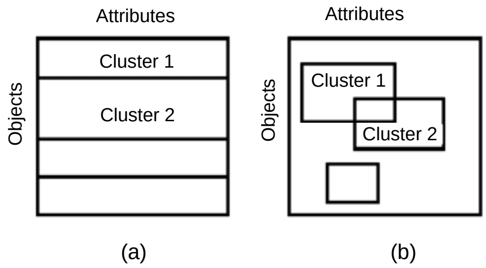

Fig. 1 Traditional clustering solution (a) and biclustering solution (b) (Lee *et al.*, 2010)

In this study, we proposed a new biclustering method for time series data following the ARMA model. The rest of this paper is organized as follows. Some related work on the ARMA model and biclustering are reviewed in Section 2. In Section 3, we proposed a new time series biclustering method. In Section 4, we applied the proposed method to synthetic datasets for verification. The datasets included extreme cases of structure of biclusters. The experimental results and discussion are reported. Finally, we concluded the paper in Section 5.

2 Related work

2.1 ARMA model

The ARMA model with autoregressive order p and moving average order q is commonly denoted as ARMA (p, q). Given a time series $x=\{x_t\}_{t=1}^n$, ARMA (p, q) model takes the form as

$$x_t = \phi_0 + \sum_{j=1}^{p} \phi_j x_{t-j} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t, \tag{1}$$

where n is the length of time series, ϕ_0 is a constant term, $\{\phi_0, \phi_1, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ is the set of ARMA (p, q) coefficients, and $\{\varepsilon_t\}_{t=1}^n$ is a sequence of independent and identically distributed white noise terms (Xiong and Yeung, 2004). To estimate the model parameters, the maximum likelihood estimation (MLE) method is generally used (Wei, 2006).

When $\{\varepsilon_t\}_{t=1}^n$ are distributed as normal distribution with variance σ^2, the log likelihood function of x is given by (Wei, 2006)

$$\ln L(\Phi \mid x) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{n}\varepsilon_t^2, \tag{2}$$

where $\Phi=\{\phi_0, \phi_1, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q, \sigma^2\}$ represents the set of all model parameters. Because ε_t is not given, it is estimated reclusively by (Xiong and Yeung, 2004)

$$\varepsilon_t = x_t - \phi_0 - \sum_{j=1}^{p}\phi_j x_{t-j} - \sum_{j=1}^{q}\theta_j \varepsilon_{t-j}. \tag{3}$$

2.2 Plaid model

Lazzeroni and Owen (2002) proposed the plaid model. Let Z_{ij} be an element value of the ith object and the jth attribute in the given data, then Z_{ij} can be modeled as

$$Z_{ij} = \psi_{ij0} + \sum_{k=1}^{K}\psi_{ijk}\rho_{ik}\kappa_{jk} + \varepsilon_{ij}, \tag{4}$$

where K is the number of biclusters, and ε_{ij} is a noise. The value of ρ_{ik} is one if the ith object belonging to the kth bicluster and is zero otherwise. Similarly, κ_{jk} is one if the jth attribute belonging to the kth bicluster and is equal to zero otherwise. ρ_{ik} and κ_{jk} are called membership parameters. ψ_{ijk} is a bicluster value such that $\psi_{ijk}=\mu_k+\alpha_{jk}+\beta_{jk}$. μ_k denotes the mean effect in the kth bicluster, α_{jk} and β_{jk} denote the adjustments for the ith object and the jth attribute in the kth bicluster, respectively. α_{jk} and β_{jk} become zero when ρ_{ik} and κ_{jk} are zero. ψ_{ij0} is a background value.

To estimate the plaid model, Lazzeroni and Owen (2002) formulated the following optimization problem:

$$\min \sum_i \sum_j \left(Z_{ij} - \sum_{k=0}^{K}\psi_{ijk}\rho_{ik}\kappa_{jk} \right)^2 \quad \text{s.t.} \quad \rho_{ik},\ \kappa_{jk} = 0,\ 1,\ \forall i, j, k. \tag{5}$$

However, it is very difficult to solve Eq. (5) directly, so they took a method to extract each bicluster step by step to solve the optimal problem as follows:

$$\min \sum_i \sum_j \left(Z_{ij}^{(k)} - \psi_{ijk}\rho_{ik}\kappa_{jk}\right)^2 \quad \text{s.t.} \quad \rho_{ik},\ \kappa_{jk} = 0,\ 1,\ \ \forall i, j, k, \tag{6}$$

where $Z_{ij}^{(k)} = Z_{ij} - \sum_{m=0}^{k-1} \psi_{ijm}\rho_{im}\kappa_{jm}$.

To solve the optimization problem in Eq. (6), Lazzeroni and Owen (2002) used a linear programming (LP) relaxation approach. Turner *et al*. (2005) solved Eq. (6) using the binary least square to find ρ_{ik} and κ_{jk} to avoid the problem of relaxation. There is no closed form solution to Eq. (6). Therefore, the above two studies provide a method which estimates (μ_k, α_{jk}, β_{jk}) and (ρ_{ik}, κ_{jk}) iteratively.

3 Proposed method

We propose a new biclustering algorithm for time series data following the ARMA model. The model of the observed data is the same as shown in Eq. (4). However, the bicluster value, ψ_{ijk}, is modeled as

$$\psi_{ijk} = \phi_{i0} + \sum_{t=1}^{p} \phi_{jk} Z_{i,j-t} + \sum_{t=1}^{q} \theta_{jk}\varepsilon_{i,j-t} + \varepsilon_{ij}, \tag{7}$$

where ε_{ij}~*iid* $N(0, \sigma^2)$.

In addition, there are some assumptions for simplifying the model: (1) biclusters are not allowed to be overlapped; (2) each object has a different constant value in bicluster; and (3) all biclusters have the same known ARMA order.

We modify the optimization problem as shown in Eq. (8) to satisfy assumptions. The second assumption restricts that each value include one bicluster at most. The third assumption restricts biclusters to a consecutive time period.

$$\min \sum_i \sum_j \left(Z_{ij} - \sum_{k=1}^{K} \psi_{ijk}\rho_{ik}\kappa_{jk}\right)^2, \tag{8}$$
$$\text{s.t.} \quad \rho_{ik},\ \kappa_{jk} = 0,\ 1,\ \ \forall i, j, k,$$
$$\sum_{k=1}^{K} \rho_{ik}\kappa_{jk} \le 1,\ \ \forall i, j,$$
$$\kappa_{(p+1)k} = \kappa_{(p+2)k} = \ldots = \kappa_{(q-2)k} = \kappa_{(q-1)k} = 1,$$
$$\text{if } \kappa_{pk} = \kappa_{qk} = 1,\ p < q.$$

As shown in Lazzeroni and Owen (2002) and Turner *et al*. (2005), we also extracted each bicluster step by step to solve the problem using Eq. (6) by adding the second and third assumptions in Eq. (8).

Now, we will explain how to find the *m*th bicluster after finding (*m*−1) biclusters. We use an iterative method, which will be mentioned in subsection 3.1. Extracting biclusters is completed as soon as the predefined number of biclusters is found, or when we fail to find a bicluster five consecutive times.

3.1 Estimation of the ARMA model

Let $\rho_{im}^{(h)}$ and $\kappa_{jm}^{(h)}$ denote values of membership parameters after the *h*th iteration in the *m*th bicluster, $\Theta_m^{(h)}$ denotes the ARMA parameter after the *h*th iteration, and $\psi_{ijm}^{(h)}$ represents the estimated time series value using $\Theta_m^{(h)}$ and $Z_{ij}^{(m)}$.

Given $\rho_{im}^{(h-1)}$ and $\kappa_{jm}^{(h-1)}$, the ARMA model parameters can be updated using the reduced dataset, *D*, corresponding to the objects for which $\rho_{im}^{(h-1)}$ equals one and the time points for which $\kappa_{jm}^{(h-1)}$ equals one. Let *D* have *d* time series, $D = \{x_i\}_{i=1}^{d}$, and x_i follows

$$x_{it} = \phi_{i0} + \sum_{j=1}^{p} \phi_j x_{i,t-j} + \sum_{j=1}^{q} \theta_j \varepsilon_{i,t-j} + \varepsilon_{it}. \tag{9}$$

Then, the likelihood function of *D* is given by

$$L(\Theta_m^{(h)} \mid D) = \prod_{i=1}^{d} L(\Theta_m^{(h)} \mid x_i). \tag{10}$$

So, the log likelihood function of *D* can be express as

$$\ell(\Theta_m^{(h)}; D) = -\frac{dn}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{d}\sum_{j=1}^{n}\varepsilon_{ij}^2. \tag{11}$$

To estimate the model parameters, we use the MLE method, where we need to solve the following equation:

$$\frac{\partial}{\partial \varpi}\ell(\Theta_m^{(h)}; D) = 0,\ \text{ for } \varpi \in \Theta_m^{(h)}. \tag{12}$$

3.2 Initialization of membership parameters

Turner *et al*. (2005) proposed the initialization by a two-means clustering. It is set to one if ρ_{ik} or κ_{ik} belongs to the smaller object or attribute cluster, and otherwise it is set to zero. This is not appropriate for our proposed method because consecutiveness of time period is required.

Before initializing ρ_{ik}, we ensure that each mean of time series is zero. In the next step, we use k-means clustering with predefined k. We use a value which is a predefined number of biclusters. We select a cluster having the largest value which is the origin to centroid distance. Then ρ_{ik} belonging to the selected cluster is set to one.

To initialize κ_{ik}, we only use objects, which have ρ_{ik}=1. We use k-means clustering with k=3. We select a cluster having the largest value which is origin to centroid distance and then we set all κ_{ik} that lie between the minimum and maximum time points of the selected cluster to one.

3.3 Updating membership parameters

To update membership parameters, we modify the binary least square method. $\rho_{im}^{(h)}$ can be updated using the reduced dataset corresponding to the objects for which $\kappa_{jm}^{(h-1)}$ equals one. We estimate time series value for each object using $\Theta_m^{(h)}$. If there is no constant term, it is replaced by the mean of the time series. If the ith object satisfies the following condition, $\rho_{im}^{(h)}$ is set to one.

$$\sum_{j\in\{\kappa_{jm}^{(h-1)}=1\}} e_{ijm}^2 < (1-\tau_1) \sum_{j\in\{\kappa_{jm}^{(h-1)}=1\}} \left(Z_{ij}^{(m)}\right)^2, \qquad (13)$$

where $e_{ijm}=Z_{ij}^{(m)}-\psi_{ijm}^{(h)}$, and τ_1 lies on (0,1) and denotes the minimum proportion of explaining data.

To update κ_{jm}, there are two steps: extension and reduction. We estimate time series value for each object using $\Theta_m^{(h)}$ and D. The extension step applies to time points having $\kappa_{jm}^{(h-1)}=0$. Let T_s be the start time point and T_e the end time point in D.

If the (T_s−1)th time point satisfies the following conditions, $\kappa_{T_s-1,k}$ is set to one, and then we look for the next time point until reaching the first time point. Similarly, if the (T_e−1)th time point satisfies Eq. (14), then $\kappa_{T_s+1,k}$ is set to one and we look for the next time point until reaching the final time point. τ_2 lies on (0,1) and denotes the minimum proportion of explaining data.

$$\sum_{i\in\{\rho_{im}^{(h)}=1\}} e_{ijk}^2 < (1-\tau_2) \sum_{i\in\{\rho_{im}^{(h)}=1\}} \left(Z_{ij}^{(m)}\right)^2. \qquad (14)$$

We define a new parameter C controlling the connection in extension step. Let T_t be a time point satisfying Eq. (14). Let T_b be the time point closest to T_t and $\kappa_{T_b k}=1$. If the interval between T_t and T_b is smaller than C, then all κ_{jk} between $\kappa_{T_t k}$ and $\kappa_{T_b k}$ are set to one. Otherwise, if time points do not satisfy Eq. (14) for C consecutive times, we terminate the extension step.

The reduction step applies to time points having $\kappa_{jm}^{(h-1)}=1$. If the T_s time point does not satisfy Eq. (14), then κ_{jm} is set to zero and we look for the next time point until reaching another side of bicluster. If the time point satisfies Eq. (14), we terminate the reduction step.

We update the ARMA model parameters and membership parameters until $\rho_{im}^{(h)}$ and $\kappa_{jm}^{(h)}$ are the same as $\rho_{im}^{(h-1)}$ and $\kappa_{jm}^{(h-1)}$, respectively. If $\sum_i \rho_{im} \le 1$ or $\sum_j \kappa_{jm} \le \min(p,q)$, then the bicluster is eliminated because it is meaningless.

After extracting biclusters, we review the data for overlapped biclusters. If overlapped biclusters exist, we define a set, OL, which includes overlapped objects. Let the kth and mth biclusters be included in OL. If the ith object in OL satisfies Eq. (15), then ρ_{im}=0, otherwise ρ_{ik}=0.

$$\frac{1}{\sum_j \kappa_{jk}} \sum_{i\in\{\rho_{ik}=1\}} \varepsilon_{ijk}^2 < \frac{1}{\sum_j \kappa_{jm}} \sum_{i\in\{\rho_{im}=1\}} \varepsilon_{ijm}^2, \qquad (15)$$

where $\varepsilon_{ijk}=Z_{ij}-\Psi_{ijk}$ and $\varepsilon_{ijm}=Z_{ij}-\Psi_{ijm}$.

After reviewing the data for overlapped biclusters, we re-estimate ARMA parameters.

4 Numerical experiments

To validate the accuracy of the proposed method, we conducted experiments with synthetic data.

4.1 Validation index

M_1 and M_2 are different biclustering results from the same data. M_j consists of K_j biclusters (j=1, 2). Let $B_k^{(j)}$ be the kth bicluster of M_j, which consists of an object set $O_k^{(j)}$ and an attribute set $A_k^{(j)}$. Then, M_j can be defined as (Lee *et al*., 2010)

$$\begin{aligned} M_j &= \left\{B_1^{(j)}, B_2^{(j)}, ..., B_{K_j}^{(j)}\right\} \\ &= \left\{\left(O_1^{(j)}, A_1^{(j)}\right), \left(O_2^{(j)}, A_2^{(j)}\right), ..., \left(O_{K_j}^{(j)}, A_{K_j}^{(j)}\right)\right\}. \end{aligned} \quad (16)$$

The F_1 index is a measure to evaluate the similarity between two biclusters (Turner *et al*., 2005), which is defined as

$$F_1\left(B_k^{(1)}, B_m^{(2)}\right) = \frac{2\left|O_k^{(1)} \cap O_m^{(2)}\right|\left|A_k^{(1)} \cap A_m^{(2)}\right|}{\left|O_k^{(1)}\right|\left|A_k^{(1)}\right| + \left|O_m^{(2)}\right|\left|A_m^{(2)}\right|}. \quad (17)$$

where F_1 index lies between 0 and 1. The F_1 index is set to 1 if two biclusters are exactly the same, while it is set to 0 if two biclusters are exclusive (Lee *et al*., 2010).

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

$$S(M_1, M_2) = \frac{1}{K_1}\sum_{i=1}^{K_1} \max_{j=1}^{K_2} F_1\left(B_i^{(1)}, B_j^{(2)}\right). \quad (18)$$

where S index lies between 0 and 1, and 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 (Lee *et al*., 2010).

4.2 Construction of synthetic data

We generated three different types of synthetic datasets (Data I, Data II, and Data III) to verify the proposed method. The datasets have some common properties. Biclusters in each dataset were generated to have a stationary ARMA model, with eight different ARMA orders: (2, 0), (3, 0), (0, 2), (0, 3), (1, 1), (1, 2), (2, 1), and (2, 2). Constant terms of ARMA models are randomly generated between 2 and 3. The true biclusters consist of 30 objects and 40 time points. Background noises follow a standard normal distribution. We generate 20 replications at each ARMA order. Therefore, there are 160 different datasets in each type of dataset.

The first type of dataset (Data I) contains one bicluster, which consists of 100 objects and 100 time points. Fig. 2 shows one example dataset from Data I. The second type of dataset (Data II) contains two biclusters, each of which consists of 100 common objects but 150 different time points as shown in Fig. 3a. The third type of dataset (Data III) contains two biclusters, each of which consists of 120 different objects but 100 common time points as shown in Fig. 3b.

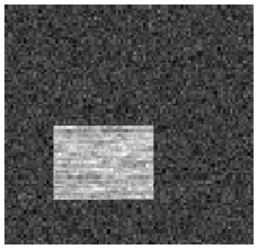

Fig. 2 Synthetic data with one bicluster (Data I)

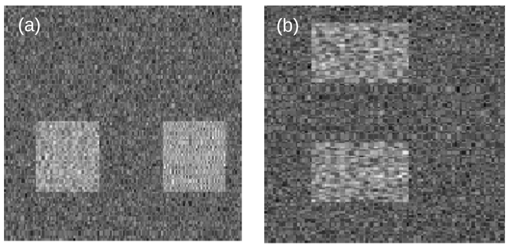

Fig. 3 Synthetic data with two biclusters
Biclusters share the same objects (Data II) (a) and the same time points (Data III) (b)

4.3 Experimental results for synthetic datasets

The proposed method was tested on three types of datasets described in Section 4.2. Also, we obtained a bicluster solution for each data using the algorithm in (Lee *et al*., 2010) for the purpose of comparison. This method is the modified biclustering method for time series based on Turner *et al*. (2005)'s method, which showed a better or similar performance as compared with the Turner's method.

The maximum number of biclusters to be found was set to 2 for Data I, and 4 for other datasets. τ_1 and τ_2 were set to 0.5. We repeated the proposed method 10 times for each value of $C\in\{0,1,2,3\}$. Let *True* be the true bicluster solution and *Exp* be an obtained bicluster solution.

As shown in Figs. 4a–4c, C has little effect on the values of S(*True*, *Exp*). The proposed method shows better results than Lee *et al.* (2010)'s method. As shown in Fig. 4b, the performance is the lowest when biclusters share the common objects.

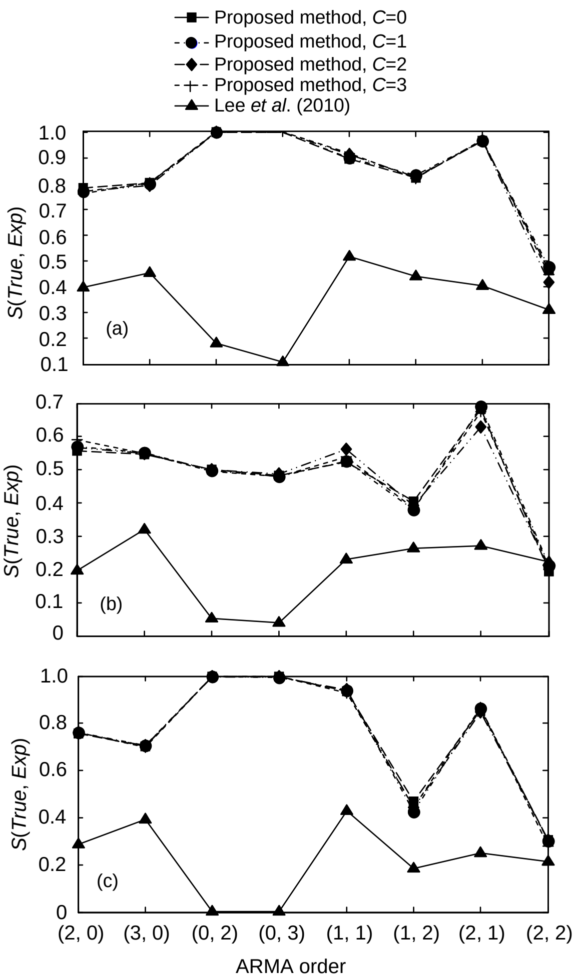

Fig. 4 Means of *S*(*True*, *Exp*) value according to the ARMA order (a) Data I, (b) Data II, and (c) Data III

Table 1 shows the number of failures in finding biclusters at each dataset among 200 repetitions, as shown in Figs. 4a–4c, the ARMA (2, 2) case shows the lowest performance in all datasets. But Table 2 shows a relatively high value of S(*Exp*, *True*) and S(*True*, *Exp*) if only successful results are summarized.

Table 1 Number of failures in finding bicluster in each dataset (the maximum value is 200)

ARMA (*p*, *q*)		(2, 0)	(3, 0)	(0, 2)	(0, 3)
Data I	Proposed method	0	5	0	0
	Lee *et al.* (2010)	82	68	72	74
Data II	Proposed method	32	30	0	0
	Lee *et al.* (2010)	69	43	74	101
Data III	Proposed method	0	0	0	0
	Lee *et al.* (2010)	63	45	171	191
ARMA (*p*, *q*)		(1, 1)	(1, 2)	(2, 1)	(2, 2)
Data I	Proposed method	7	28	0	105
	Lee *et al.* (2010)	52	44	61	91
Data II	Proposed method	33	47	11	125
	Lee *et al.* (2010)	59	47	41	60
Data III	Proposed method	0	82	0	110
	Lee *et al.* (2010)	37	112	68	88

Table 2 Means of *S*(*Exp*, *True*) and *S*(*True*, *Exp*) value in ARMA (2, 2) case when successfully finding biclusters

	S(*Exp*, *True*)			*S*(*True*, *Exp*)		
	Data I	Data II	Data III	Data I	Data II	Data III
Proposed method, *C*=0	0.97	0.92	0.66	0.98	0.52	0.68
Lee *et al.* (2010)	0.36	0.25	0.14	0.57	0.32	0.38

5 Conclusions

We proposed a new biclustering algorithm for time series data following the ARMA model. This method used an MLE method to estimate the ARMA parameters and modified the plaid model structure. The proposed method may apply to long time series data differently from other biclustering methods which focus on microarray data.

We applied the proposed method to some synthetic data, which included extreme cases with sharing whole objects or time points between biclusters. Experimental results showed a good performance for a small ARMA order. However, the ARMA (2, 2) case showed some unstable results.

We are planning to modify the proposed method to obtain more stable biclustering results regardless of the ARMA order and for determining the suitable ARMA order in the future work. Also, we will apply the proposed method to some real data such as economic time series data.

References

Berndt, D.J., 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, Seattle, USA, p.359-370.

Cheng, Y., Church, G., 2000. Biclustering of Expression Data. Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, California, USA, p.93-103.

Kalpakis, K., Gada, D., Puttagunta, V., 2001. Distance Measures for Effective Clustering of ARIMA Time-series. Proceedings of the IEEE International Conference on Data Mining, California, USA, p.273-280. [doi:10.1109/ICDM.2001.989529]

Keogh, E., Kasetty, S., 2004. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery*, **7**(4):349-371. [doi:10.1023/A:1024988512476]

Lazzeroni, L., Owen, A., 2002. Plaid models for gene expression data. *Statistica Sinica*, **12**:61-86.

Lee, J., Lee, Y., Jun, C.H., 2010. A biclustering method for time series analysis. *Industrial Engineering & Management Systems*, **9**:129-138.

Liao, T.W., 2005. Clustering of time series data—a survey. *Pattern Recognition*, **38**(11):1857-1874. [doi:10.1016/j.patcog.2005.01.025]

Madeira, S.C., Oliveira, A.L., 2005. A linear time biclustering algorithm for time series gene expression data. *LNCS*, **3692**:39-52. [doi:10.1007/11557067_4]

Santamaría, R., Quintales, R., Theorón, R., 2007. Method to Bicluster Validation and Comparison in Microarray Data. Intelligent Data Engineering and Automated Learning-ideal 8th International Conference, Birmingham, UK, p.780-789.

Turner, H., Bailey, T., Krzanowski, W., 2005. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, **48**(2):235-254. [doi:10.1016/j.csda.2004.02.003]

Wei, W.W.S., 2006. Time Series Analysis: Univariate and Multivariate Methods. Addison-Wesley, USA.

Xiong, Y., Yeung, D.Y., 2004. Time series clustering with ARMA mixtures. *Pattern Recognition*, **37**(8):1675-1689. [doi:10.1016/j.patcog.2003.12.018]

Xu, X., Lu, Y., Tan, K.L., Tung, A.K.H., 2009. Finding Time-lagged 3D Clusters. Proceedings of the IEEE International Conference on Data Engineering, Shanghai, China, p.445-456. [doi:10.1109/ICDE.2009.80]

Zhang, Y., Zha, H., Chu, C., 2005. A Time-series Biclustering Algorithm for Revealing Co-regulated Genes. Proceedings of the International Conference on Information Technology, Coding and Computing, Las Vegas, USA, **1**:32-37. [doi:10.1109/ITCC.2005.46]

Zhao, L., Zaki, M.J., 2005. Tricluster: an Effective Algorithm for Mining Coherent Clusters in 3D Microarray Data. Proceedings of the ACM SIGMOD International Conference on Management of Data, Maryland, USA, p.694-705. [doi:10.1145/1066157.1066236]