Evolutionary Soft Co-Clustering

Wenlu Zhang* Shuiwang Ji* Rui Zhang†

Abstract

We consider the mining of hidden block structures from time-varying data using evolutionary co-clustering. Existing methods are based on the spectral learning framework, thus lacking a probabilistic interpretation. To overcome this limitation, we develop a probabilistic model for evolutionary co-clustering in this paper. The proposed model assumes that the observed data are generated via a two-step process that depends on the historic co-clusters, thereby capturing the temporal smoothness in a probabilistically principled manner. We develop an EM algorithm to perform maximum likelihood parameter estimation. An appealing feature of the proposed probabilistic model is that it leads to soft co-clustering assignments naturally. To the best of our knowledge, our work represents the first attempt to perform evolutionary soft co-clustering. We evaluate the proposed method on both synthetic and real data sets. Experimental results show that our method consistently outperforms prior approaches based on spectral method.

1 Introduction

Many real-world processes are dynamically changing over time. As a consequence, the observed data generated by these processes also evolve smoothly. For example, in literature mining, the author-conference co-occurrence matrix evolves dynamically over time, since authors may shift their research interests smoothly. In computational biology, the expression data matrices are evolving, since gene expression controls are deployed sequentially in many biological processes. Temporal data mining aims at discovering knowledge from time-varying data and is now receiving increasing attention in many domains, including graph and network analysis [18, 2, 26], information retrieval [27], text mining [23], clustering analysis [1, 4, 6, 19], and matrix factorization [28]. Since the data are evolving smoothly over time, the patterns embedded into the data are also expected to change smoothly. Therefore, one of the key challenges in temporal data mining is how to incorporate temporal smoothness into the patterns identified from temporally adjacent time points.

Co-clustering, also known as bi-clustering, aims at discovering block structures from data matrices [14, 5, 10]. In traditional clustering analysis, the sample and feature dimensions of the data matrix are treated asymmetrically [15]. In contrast, co-clustering aims at clustering both the samples and the features simultaneously to identify hidden block structures embedded into the data matrix. Currently, co-clustering has been widely applied in many domains, including biological data analysis [22, 16], text mining [10, 8], and social studies [11]. However, most existing studies on co-clustering assume that the data are static; that is, they do not evolve over time, thereby preventing their applicabilities in the dynamic settings.

In this paper, we consider the mining of hidden block structures from data matrices that evolve dynamically over time. A simple approach is to apply co-clustering methods to each data matrix separately. This approach, however, ignores the smoothness between temporally adjacent matrices, yielding inconsistent results. To discover smooth hidden block structures from dynamic data, we systematically study the evolutionary co-clustering of time-varying data matrices. Existing methods are based on the spectral learning framework and do not require the co-clustering indicator matrices to be nonnegative, hindering a probabilistic interpretation of the results [13].

To overcome this limitation, we develop a probabilistic model for evolutionary co-clustering. The proposed probabilistic model assumes that the observed data matrices are generated via a two-step process that depends on the historic co-clusters, thereby capturing the temporal smoothness in a probabilistically principled manner. To enable maximum likelihood parameter estimation, we develop an EM algorithm for the probabilistic model. An appealing feature of the proposed probabilistic model is that it leads to soft co-clustering assignments naturally. To the best of our knowledge, our work represents the first attempt to perform evolutionary soft co-clustering. This formalism is very useful in many applications. For example, in gene expression data analysis, each gene usually functions in multiple pathways. We evaluate the proposed methods on both synthetic and real data sets. Experimental results show that the proposed probabilistic model consistently outperforms prior methods based on spectral learning.

*Old Dominion University, Norfolk, VA 23529.

†City College of New York, New York, NY 10031.

Notations: We use $\mathrm{Tr}(W)$ to represent the trace of matrix W where $\mathrm{Tr}(W) = \sum_{i=1}^{n} w_{ii}$ for any matrix $W \in \mathbb{R}^{n\times n}$. The squared Frobenius norm of a matrix W is defined as $\|W\|_F^2 = \sum_{i,j} w_{i,j}^2 = \mathrm{Tr}(W^T W)$. We use $A \in \mathbb{R}^{m\times n}$ to denote the data matrix. For a problem with k co-clusters, the co-clustering results can be encoded into a co-cluster indicator matrix $R \in \mathbb{R}^{(m+n)\times k}$. Let $R^T = [R_1^T, R_2^T]$, where $R_1 \in \mathbb{R}^{m\times k}$ and $R_2 \in \mathbb{R}^{n\times k}$. The indicator matrix R is defined as follows: $(R_1)_{ij} = 1$ if the ith row belongs to the jth co-cluster, and zero otherwise; $(R_2)_{ij} = 1$ if the ith column belongs to the jth co-cluster, and zero otherwise. We further define $\tilde{R} \in \mathbb{R}^{(m+n)\times k}$, where each column of $\tilde{R}$ is the corresponding column in R divided by the square root of the number of ones in that column.

2 Background

Cluster analysis aims at grouping a set of data points into clusters so that the data points in the same cluster are similar, while those in different clusters are dissimilar. Given a data matrix $A = [a_1, a_2, \cdots, a_n] \in \mathbb{R}^{m\times n}$ consisting of n data points $\{a_i\}_{i=1}^n \in \mathbb{R}^m$. Let $\Pi = \{\pi_j\}_{j=1}^k$ denote a partition of the data into k clusters; that is, $\pi_j = \{v | a_v \text{ in cluster } j\}$ and $\pi_i \bigcap \pi_j = \emptyset$ for $i \neq j$. The partition can also be encoded equivalently into an $n \times k$ cluster indicator matrix $Y = [y_1, y_2, \cdots, y_k]$, where $Y_{pq} = 1$ if the pth data point belongs to the qth cluster, and 0 otherwise. We further define a normalized cluster indicator matrix $\tilde{Y} = [\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_k]$, where $\tilde{y}_i = y_i/\sqrt{|\pi_i|}$ and $|\pi_i|$ denotes the number of data points in the ith cluster. It can be verified that the columns of $\tilde{y}$ are orthonormal, i.e., $\tilde{y}^T\tilde{y} = I_k$.

2.1 Spectral Clustering

In spectral clustering [25, 21, 24, 9], the data set is represented by a weighted graph $G = (V, E)$ in which the vertices in V correspond to data points, and the edges in E characterize the similarities between data points. The weights of the edges are usually encoded into the adjacency matrix W. Several constructions of similarity graph are regularly used, such as the ϵ-neighborhood graph and the k-nearest neighbor graph [21].

Spectral clustering is based on the idea of graph cuts, and different graph cut measures have been defined. Two popular approaches are to maximize the average association and to minimize the normalized cut [25]. For two subsets, π_p, $\pi_q \in \Pi$, the cut between π_p and π_q is defined as $cut(\pi_p, \pi_q) = \sum_{i\in\pi_p, j\in\pi_q} W(i,j)$. Then the k-way average association (AA) and the k-way normalized cut (NC) can be written as

$$\text{AA} = \sum_{l=1}^{k} \frac{cut(\pi_l, \pi_l)}{|\pi_l|}, \quad \text{NC} = \sum_{l=1}^{k} \frac{cut(\pi_l, \Pi \setminus \pi_l)}{cut(\pi_l, \Pi)}, \tag{2.1}$$

where $\setminus$ denotes the set minus operation. In [6], the negated average association is defined as $\text{NA} = \mathrm{Tr}(W) - AA$. Note that the average association characterizes the within cluster association, while the normalized cut captures the between cluster separation. Furthermore, maximizing the average association is equivalent to minimizing the negated average association. Hence, the negated average association will be used throughout this paper.

It has been shown [25] that exact minimization of common graph cut measures, such as the normalized cut and the negated average association, is intractable. Hence, a two-step procedure is commonly employed in spectral clustering. In the first step, the graph cut problems are relaxed to a trace optimization problem, whose solution typically can be obtained by computing the eigen-decomposition of the graph Laplacian matrices [21, 7]. Then in the second step, the final clustering results are generated by clustering the solution of the relaxed problem. Note that we focus on how to incorporate smoothness constraints into the first step in this paper, so the second step will not be discussed further in the rest of this paper.

2.2 Spectral Co-Clustering

In [8, 30], the spectral clustering formalism is extended to solve co-clustering problems. Given a data matrix $A \in \mathbb{R}^{m\times n}$, such as the word-by-document matrix, a bipartite graph is constructed, where the two sets of vertices correspond to the rows and the columns, respectively. Then the co-clustering problem is reduced to perform graph cuts on this bipartite graph. Formally, the similarity matrix of the bipartite graph can be written as

$$W = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}. \tag{2.2}$$

A variety of graph cut criteria can then be applied to partition the bipartite graph. For example, when the normalized cut is used, the Laplacian matrix and the degree matrix for this bipartite graph can be written as

$$L = \begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \tag{2.3}$$

where D_1 and D_2 are diagonal matrices whose diagonal elements are defined as

$$D_1(ii) = \sum_j A_{ij}, \quad D_2(jj) = \sum_i A_{ij}.$$

Then the normalized cut criterion can be relaxed, and the solution for the relaxed problem can be obtained by solving the following eigenvalue problem:

$$\begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{2.4}$$

where $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ are the relaxed row and column cluster indicator matrices, respectively.

2.3 Evolutionary Clustering When the data matrices evolve along the temporal dimension, it is desirable to capture the temporal smoothness in clustering analysis. Recently, several evolutionary clustering methods have been developed to cluster time-varying data by incorporating temporal smoothness constraints directly into the clustering framework [4, 6, 19].

In [6], two main frameworks, known as preserving cluster quality (PCQ) and preserving cluster membership (PCM), are proposed to incorporate temporal smoothness. In these two formulations, the cost functions contain two terms, known as the snapshot cost (CS) and the temporal cost (CT) as $\text{Cost} = \alpha \cdot \text{CS} + (1-\alpha)\text{CT}$, where $0 \le \alpha \le 1$ is a tunable parameter. In this formulation, the snapshot cost captures the clustering quality on the current data matrix, while the temporal cost encourages the temporal smoothness with respect to either historic data or historic clustering results. The main difference between PCQ and PCM lies in the definitions of the temporal costs. Specifically, the temporal cost in PCQ is devised to encode the consistency between current clustering results with historic data, while that in PCM is used to encourage temporal smoothness between current and historic clustering results.

Let Y_t denotes the cluster indicator matrix for time t, then the objective function for PCQ can be expressed as $\text{Cost}_{\text{PCQ}} = \alpha \cdot \text{Cost}_t|_{Y_t} + (1-\alpha) \cdot \text{Cost}_{t-1}|_{Y_t}$, where $\text{Cost}_t|_{Y_t}$ and $\text{Cost}_{t-1}|_{Y_t}$ denote the costs of applying the clustering results in Y_t to the data at time points t and $t-1$, respectively. In contrast, the temporal cost in PCM is expressed as the difference between the current and the historic clustering results, leading to the following overall objective function $\text{Cost}_{\text{PCM}} = \alpha \cdot \text{Cost}_t|_{Y_t} + (1-\alpha) \cdot \text{dist}(Y_t, Y_{t-1})$, where $\text{dist}(\cdot,\cdot)$ denotes certain distance measure.

Following the soft clustering framework proposed in [29], an evolutionary clustering method based on nonnegative matrix factorization (NMF) has been developed in [19]. Let W_t be the similarity matrix for time point t, the objective function for evolutionary clustering in [19] can be expressed as

$$\begin{aligned} \text{Cost}_{\text{NMF}} &= \alpha \cdot D(W_t \| X_t \Lambda_t X_t^T) \\ &\quad + (1-\alpha) \cdot D(X_{t-1}\Lambda_{t-1} \| X_t \Lambda_t), \end{aligned} \tag{2.5}$$

where $D(\cdot\|\cdot)$ is the KL-divergence, X_t is the soft clustering indicator matrix, and Λ_t is a diagonal matrix. An iterative procedure is devised to compute the solution. It is also shown in [19] that the proposed method can be interpreted from the perspective of probabilistic generative models.

3 Evolutionary Soft Co-Clustering

Although both co-clustering and evolutionary clustering have been intensively studied, the field of evolutionary co-clustering remains largely unexplored [13]. In addition, prior method (discussed in Section 4) employs singular value decomposition (SVD) in computing the solutions of relaxed problems. In many applications, such as image and text analysis, the original data matrices are nonnegative. A factorization such as SVD produces factors containing negative entries. This leads to complex cancelations between positive and negative numbers, and the results are usually difficult to interpret [17]. To address this challenge, we propose a probabilistic model for evolutionary co-clustering in this section. This model results in nonnegative factors, thereby overcoming the limitation of spectral methods. In addition, the probabilities can be interpreted to produce soft co-clusters.

3.1 The Proposed Model In the proposed model, we assume that the similarity matrix W_t of the bipartite graph can be factorized as

$$W_t = H_t \tilde{H}_t, \tag{3.6}$$

where

$$W_t = \begin{bmatrix} 0 & A_t \\ {A_t}^T & 0 \end{bmatrix}, \tag{3.7}$$

$A_t \in \mathbb{R}^{m \times n}$ is the data matrix,

$$H_t = \begin{bmatrix} H_{1,t} & 0 \\ 0 & H_{2,t} \end{bmatrix}, \quad \tilde{H}_t = \begin{bmatrix} 0 & H_{2,t}^T \\ H_{1,t}^T & 0 \end{bmatrix}, \tag{3.8}$$

where $H_t \in \mathbb{R}^{(m+n)\times(2k)}$, $\tilde{H}_t \in \mathbb{R}^{(2k)\times(m+n)}$, $H_{1,t} \in \mathbb{R}^{m\times k}$ denotes the row cluster indicator matrix, and $H_{2,t} \in \mathbb{R}^{n \times k}$ denotes the column cluster indicator matrix. It follows that

$$H_t \tilde{H}_t = \begin{bmatrix} 0 & H_{1,t} H_{2,t}^T \\ \left(H_{1,t} H_{2,t}^T\right)^T & 0 \end{bmatrix}, \tag{3.9}$$

which matches the structure of W_t in Eq. (3.7).

In the proposed probabilistic model, the similarity matrix W_t is generated via a two-step process. In the first step, $H_t \tilde{H}_t$ is generated based on the co-clustering results $H_{t-1}\tilde{H}_{t-1}$ at time point $t-1$ using

$P(H_t\tilde{H}_t|H_{t-1}\tilde{H}_{t-1})$. In the second step, the observed similarity matrix W_t is generated based on $H_t\tilde{H}_t$ using $P(W_t|H_t\tilde{H}_t)$. Following [19], we employ the Dirichlet and multinomial distributions in the first and second steps, respectively. This gives rise to the following log likelihood function of observing the current weight matrix W_t:

$$\begin{aligned} L &= \log P(W_t|H_t\tilde{H}_t) + \nu \log P(H_t\tilde{H}_t|H_{t-1}\tilde{H}_{t-1}) \\ &= 2\sum_{ij}(A_t)_{ij}\log(H_{1,t}H_{2,t}^T)_{ij} \\ (3.10) + & \quad 2\nu\sum_{ij}(H_{1,t-1}H_{2,t-1}^T)_{ij}\log(H_{1,t}H_{2,t}^T)_{ij}, \end{aligned}$$

where parameter ν controls the temporal smoothness.

3.2 An EM Algorithm

To maximize the log likelihood in Eq. (3.10), we derive an EM algorithm in the following. To simplify notation, we omit the subscript t when the time information is clear from context. We use variables with hat (e.g., $\hat{h}_{1;ik}$ and $\hat{H}_1$) to denote the values obtained from the previous iteration.

In the E-step, we compute the expectation as

$$\phi_{ijk} = \hat{h}_{1;ik}\hat{h}_{2;jk}/(\hat{H}_1\hat{H}_2^T)_{ij}, \tag{3.11}$$

where $\sum_k \phi_{ijk} = 1$, $\hat{h}_{1;ik}$ and $\hat{h}_{2;jk}$ denote the ikth and the jkth entries, respectively, of H_1 and H_2 computed from the previous iteration.

In the M-step, we maximize the expectation of log likelihood with respect to $\Phi = (\Phi)_{ijk}$

$$\begin{aligned} E_\Phi[L] &= 2\times\sum_{ijk}\phi_{ijk}a_{ij}^t\log(h_{1;ik}^t h_{2;jk}^t) \\ (3.12) \qquad &+ 2\times\nu\sum_{ijk}h_{1;ik}^{t-1}h_{2;jk}^{t-1}\log(h_{1;ik}^t h_{2;jk}^t), \end{aligned}$$

where the superscripts t and $t-1$ are used to denote variables at the corresponding time points. To facilitate a probabilistic interpretation of the co-clustering results, we impose the following normalization constraints:

$$\sum_i h_{1;ik}^t = 1, \quad \sum_j h_{2;jk}^t = 1.$$

By using Lagrange multipliers for these constraints, it can be shown that the following update rules will monotonically increase the expected log likelihood defined in Eq. (3.12), thereby leading to convergence to an locally optimal solution [29]:

$$\begin{aligned} h_{1;ik} &\leftarrow 2\times\sum_j\frac{\hat{h}_{1;ik}\hat{h}_{2;jk}a_{ij}^t}{(\hat{H}_1\hat{H}_2^T)_{ij}} + 2\times\nu\sum_j(h_{1;ik}^{t-1}h_{2;jk}^{t-1}), \\ h_{2;jk} &\leftarrow 2\times\sum_i\frac{\hat{h}_{1;ik}\hat{h}_{2;jk}a_{ij}^t}{(\hat{H}_1\hat{H}_2^T)_{ij}} + 2\times\nu\sum_i(h_{1;ik}^{t-1}h_{2;jk}^{t-1}). \end{aligned}$$

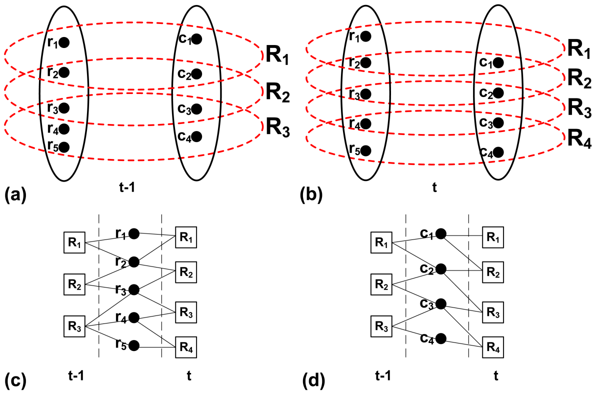

Figure 1: Illustration of co-cluster evolution. Panels (a) and (b) show the co-clustering results at time points $t-1$ and t, respectively. Panels (c) and (d) show the row and column co-cluster evolution, respectively, between time points $t-1$ and t. See text for detailed explanations.

The results are then normalized such that $\sum_i h_{1;ik}^t = 1$ and $\sum_j h_{2;jk}^t = 1$, $\forall k$.

The E-step and and M-step are repeated until a locally optimal solution is obtained. Then the matrices $H_{1,t}$ and $H_{2,t}$ can be used as row and column co-cluster indicator matrices, respectively, to obtain soft co-clustering results. Our experimental results show that this probabilistic model achieves superior performance on both synthetic and real data sets.

3.3 Co-Cluster Evolution

An unique property of the proposed probabilistic model is that the identified co-clusters can be related across time points, giving rise to co-cluster evolution. Figure 1 shows how co-clusters evolve for a 5 × 4 example data matrix, where r_1 to r_5 correspond to the five rows, c_1 to c_4 correspond to the four columns, and R_1 to R_4 denote the co-clusters. In panel (a), the matrix is co-clustered into 3 co-clusters as indicated by the dashed ovals. At time t in panel (b), the data is clustered into 4 co-clusters. The row and column co-clusters across time points can be related naturally by considering the sharing of rows and columns between co-clusters. This is illustrated in panels (c) and (d), which depict how the row and column co-clusters, respectively, evolves from time points $t-1$ to t. Note that the co-cluster evolution is a direct product of the soft co-cluster assignment proposed in this paper. This demonstrates that the soft co-cluster assignment formalism captures additional temporal dynamics, which have been ignored by prior methods. More importantly, we show in Section 5 that our evolutionary soft co-clustering formulation outperforms prior methods consistently.

4 Related Work and Extensions

Following the evolutionary spectral clustering framework in [6], two spectral methods for evolutionary co-clustering have been proposed in [13]. In this section, we systematically extend the spectral methods in [13] using two different graph cut criteria, leading to four different methods for capturing the temporal smoothness. Our experimental results in Section 5 show that the probabilistic model proposed in this paper consistently outperforms the spectral methods.

4.1 Preserving Co-Cluster Quality In preserving co-cluster quality (PCCQ), the temporal cost measures the quality of current co-clustering results when applied to historic data. In the following, we describe the PCCQ formalism using both the negated average association and the normalized cut criteria.

4.1.1 Negated Average Association Given a data matrix $A \in \mathbb{R}^{m\times n}$, the negated average association objective function in co-clustering can be written as

$$NA = \mathrm{Tr}(W) - \mathrm{Tr}(\tilde{R}^T W \tilde{R}), \tag{4.13}$$

where $\tilde{R} \in \mathbb{R}^{(m+n)\times k}$ is the normalized co-cluster indicator matrix, W is defined in Eq. (2.2) and denotes the similarity matrix associated with the bipartite graph. Writing $\tilde{R} = [P^T, Q^T]^T$, where $P \in \mathbb{R}^{m\times k}$ and $Q \in \mathbb{R}^{n\times k}$ are the row and column cluster indicator matrices, respectively, and substituting W into Eq. (4.13), we obtain

$$NA = -\mathrm{Tr}(P^T A^T Q + P^T A Q) = -2\mathrm{Tr}(P^T A Q). \tag{4.14}$$

We propose to employ the following cost function for the PCCQ evolutionary co-clustering formalism based on negated average association:

$$\begin{aligned} \mathrm{NA}_{\mathrm{PCCQ}} &= \alpha \cdot NA_t|_{\tilde{R}_t} + (1-\alpha)\cdot NA_{t-1}|_{\tilde{R}_t} \\ &= -\mathrm{Tr}\left(P_t^T\left(\alpha A_t + (1-\alpha)A_{t-1}\right)Q_t\right), \end{aligned}$$

where A_t, P_t, and Q_t denote the corresponding matrices for time point t. Since solving the above problem exactly is intractable, we propose to relax the constraints on the entries in P_t and Q_t while keeping the orthonormality constraints. It follows from the spectral co-clustering formalism [8] that columns of the optimal P_t^* and Q_t^* that minimize the relaxed problem are given by the k principal left and right, respectively, singular vectors of the matrix $\alpha A_t + (1-\alpha)A_{t-1}$.

4.1.2 Normalized Cut It follows from Proposition 1 in [3] that the normalized cut criterion can be expressed equivalently as

$$NC = k - \mathrm{Tr}\left(S^T(D^{-\frac{1}{2}} W D^{-\frac{1}{2}})S\right), \tag{4.15}$$

where

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \qquad W = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}, \tag{4.16}$$

and $S \in \mathbb{R}^{(m+n)\times k}$ satisfies two conditions: (a) the columns of $D^{-1/2}S$ are piecewise constant with respect to R, and (b) $S^T S = I$. Let $S = [E^T, F^T]^T$, where $E \in \mathbb{R}^{m\times k}$ and $F \in \mathbb{R}^{n\times k}$, then the normalized cut criterion in Eq. (4.15) can be written as $NC = k - 2\mathrm{Tr}\left(E^T(D_1^{-1/2} A D_2^{-1/2})F\right)$.

We propose to employ the following cost function in PCCQ under the normalized cut criterion:

$$\begin{aligned} \mathrm{NC}_{\mathrm{PCCQ}} &= \alpha \cdot NC_t|_{S_t} + (1-\alpha)\cdot NC_{t-1}|_{S_t} \\ &= k - 2\mathrm{Tr}\Big(E_t^T(\alpha D_{1,t}^{-1/2} A_t D_{2,t}^{-1/2} \\ &+ (1-\alpha) D_{1,t-1}^{-1/2} A_{t-1} D_{2,t-1}^{-1/2})F_t\Big), \end{aligned}$$

where $D_{1,t}$ and $D_{2,t}$ are the diagonal matrices at time t. Similar to the case of negated average association, we relax the constraints on the entries of E_t and F_t while keep the orthonormality constraints. It can be verified that columns of the optimal E_t^* and F_t^* that minimize the relaxed problem consist of the principal left and right, respectively, singular vectors of the matrix $\alpha D_{1,t}^{-1/2} A_t D_{2,t}^{-1/2} + (1-\alpha) D_{1,t-1}^{-1/2} A_{t-1} D_{2,t-1}^{-1/2}$. Then the rows of the matrix $\begin{bmatrix} D_{1,t}^{-1/2} E_t^* \\ D_{2,t}^{-1/2} F_t^* \end{bmatrix}$ are clustered to identify co-clusters.

4.2 Preserving Co-Cluster Membership In preserving co-cluster membership (PCCM), the temporal cost measures the consistency between temporally adjacent co-clustering results. Let U_t and V_t denote the solutions of the relaxed problems at time point t as described in Section 4.1. Note that columns of U_t and V_t are the left and right singular vectors, respectively, of certain matrix. Since the singular vectors of a matrix may not be unique [12], we cannot require U_t and U_{t-1} to be similar and V_t and V_{t-1} to be similar. However, it is known that $U_t V_t^T$ is unique in all cases. Hence, we propose to employ the following temporal cost in PCCM:

$$\mathrm{CT}_{\mathrm{PCCM}} = \|U_t V_t^T - U_{t-1} V_{t-1}^T\|_F^2. \tag{4.17}$$

4.2.1 Negated Average Association By using the temporal cost in Eq. (4.17) to quantify the smoothness, we propose the following overall cost function for PCCM

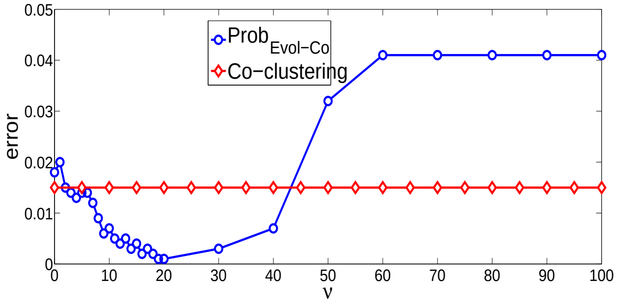

Figure 2: Performance comparison between the proposed probabilistic model ($\text{Prob}_{\text{Evol-Co}}$) with that of the co-clustering method when ν varies from 0 to 100.

under the negated average association criterion:

$$\begin{aligned}\text{NA}_{\text{PCCM}} &= \alpha \cdot \text{CS}_{\text{NA}} + (1-\alpha) \cdot \text{CT}_{\text{PCCM}} \\ &= 2(1-\alpha)k \\ &\quad -2\text{Tr}\left(U_t^T \left(\alpha A_t + (1-\alpha)U_{t-1}V_{t-1}^T\right) V_t\right).\end{aligned}$$

Minimizing NA_{PCCM} is equivalent to maximizing $\text{Tr}\left(U_t^T \left(\alpha A_t + (1-\alpha)U_{t-1}V_{t-1}^T\right) V_t\right)$. Hence, columns of the the optimal U_t^* and V_t^* consist of the principal left and right singular vectors, respectively, of the matrix $\alpha A_t + (1-\alpha)U_{t-1}V_{t-1}^T$.

4.2.2 Normalized Cut When the temporal cost in Eq. (4.17) is used along with the normalized cut criterion, we obtain the following problem:

$$\text{NC}_{\text{PCCM}} = (2-\alpha)k$$
$$-2\text{Tr}\left(U_t^T \left(\alpha D_{1,t}^{-1/2} A_t D_{2,t}^{-1/2} + (1-\alpha)U_{t-1}V_{t-1}^T\right) V_t\right).$$

Minimizing NC_{PCCM} is equivalent to maximizing

$$\text{Tr}\left(U_t^T \left(\alpha D_{1,t}^{-1/2} A_t D_{2,t}^{-1/2} + (1-\alpha)U_{t-1}V_{t-1}^T\right) V_t\right).$$

Hence, columns of the the optimal U_t^* and V_t^* consist of the principal left and right singular vectors, respectively, of the matrix $\alpha D_{1,t}^{-1/2} A_t D_{2,t}^{-1/2} + (1-\alpha)U_{t-1}V_{t-1}^T$. The final co-clusters are obtained by clustering the rows of the matrix $\begin{bmatrix} D_{1,t}^{-1/2}U_t^* \\ D_{2,t}^{-1/2}V_t^* \end{bmatrix}$.

5 Experimental Studies

5.1 Synthetic Data # 1 We generate a synthetic data set with 7 time-steps and 5 co-clusters, each containing 200 instances and 10 features. At $t = 0$, the entries corresponding to rows and columns in the same co-cluster are set to nonzero with a high probability p while other entries are set to nonzero with a low probability q which satisfies $p = 4q$ and $p + 4q = 1$. The data at $t = 1$ are generated by adding a Gaussian noise to each entry of the data at $t = 0$. To simulate the evolving nature of the data, 20% of the instances in co-cluster I are set to be weakly correlated to features in co-cluster III at $t = 2$. The level of correlation by the same set of instances is increased at $t = 3$ so that they are equally correlated to features in co-clusters I and III. At $t = 4$, this set of instances are no longer correlated to features in co-cluster I, and their correlations with features in co-cluster III are further increased. At $t = 5$, a sudden change occurs and the data matrix at $t = 1$ is restored. At $t = 6$, the size of the data matrix is changed by adding some extra instances to co-cluster I.

To demonstrate the effectiveness of the temporal cost, we compare our formulation with co-clustering method without the temporal cost. We use error rate as the performance measure, since the co-cluster memberships are known for synthetic data. The performance of the proposed model along with that of the co-clustering method (equivalent to $\nu = 0$) is reported in Figure 2. It can be observed that when ν is increased from 0 to 20, the error rate drops gradually. When ν is increased beyond 20, the error rate increases gradually. When ν lies in the interval $[5, 40]$, the proposed method outperforms the co-clustering method significantly. This shows that the evolutionary co-clustering formulation yields improved performance for a large range of ν.

5.2 Synthetic Data # 2 The second synthetic data set is generated to evaluate the performance of the proposed model in comparison to prior methods based on spectral learning. This data set contains 50 time-steps, each with 4 co-clusters, and each co-cluster contains 100 instances and 10 features. At $t = 0$, the data set is generated by following the same strategy as the first synthetic data set when $t = 0$. In each of the 0 to 49 time-steps, we add Gaussian noise to the data from previous time-step. We optimize the α and ν values on the synthetic data separately. This set of experiments, including data generation, are repeated 40 times and the average results are reported in Figure 3 for all time-steps.

We can observe from Figure 3 that the proposed probabilistic model ($\text{Prob}_{\text{EVOL-CO}}$) consistently outperforms prior methods (i.e., NA_{PCCQ}, NC_{PCCQ}, NA_{PCCM}, and NC_{PCCM}). This demonstrates that the proposed model is very effective in improving performance by requiring the factors to be nonnegative. Similar to the observation in Section 5.1, all evolutionary co-clustering approaches outperform co-clustering method consistently across most time-steps. This demonstrates that the temporal cost is effective in improving performance.

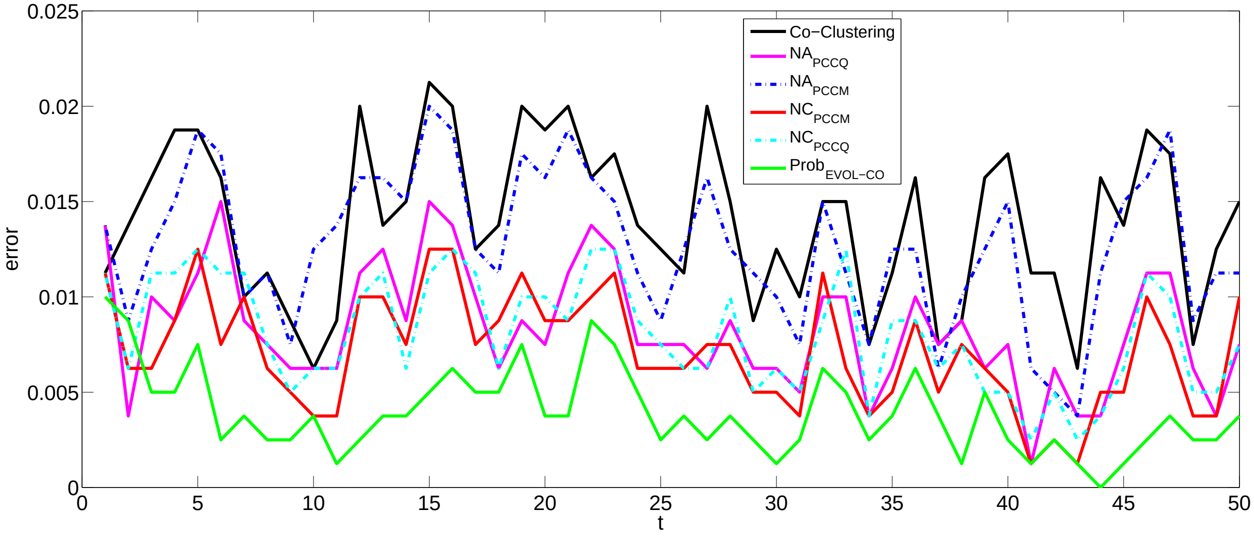

Figure 3: Performance of the probabilistic model with four methods based on spectral learning and the co-clustering method on synthetic data # 2.

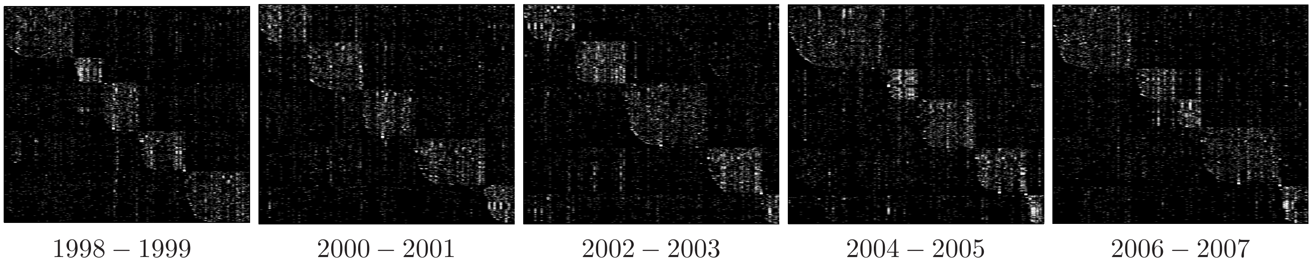
1998 – 1999 2000 – 2001 2002 – 2003 2004 – 2005 2006 – 2007

Figure 4: The block structures identified by the proposed probabilistic model on the DBLP data.

5.3 DBLP Data We conduct experiments on the DBLP data to evaluate the proposed methods. The DBLP data [27, 28] contain the author-conference information for 418,236 authors and 3,571 conferences during 1959-2007. For each year, the author-conference matrix captures how many papers are published by an author in a conference. The author-conference data matrices are very sparse, and we sample 252 conferences spanning 12 main research areas (Internet Computing, Data Mining, Machine Learning, AI, Programming Language, Data Base, Multimedia, Distributed System, Security, Network, Social Network, Operating System) in our experiments. We also remove authors with too few papers, resulting in 4147 authors from the 252 conferences. We choose the data for ten years (1998-2007) and add the data for two consecutive years, leading to data of five time points.

We apply the probabilistic model to the DBLP data in order to discover the author-conference co-occurrence relationship and their temporal evolution. We set the number of co-clusters to be 12 in the experiments, and this results in 5 major co-clusters and 7 minor co-clusters as shown in Figure 4. The 5 major co-clusters can be easily identified from our co-clustering results, and their evolutions are temporally smooth. A close examination of the results shows that related conferences are clustered into the same co-cluster consistently across all time points. For example, the co-cluster for Data Mining always contains KDD, ICDM, SDM etc., and the co-cluster for Data Base always contains SIGMOD, ICDE, VLDB, etc.

We also investigate how the authors' research interests change dynamically over time. In Figure 5, we plot the results for three authors: Jiawei Han, David Wagner, and Elisa Bertino. For each author and each time point, we distribute the 12 conference categories evenly around a circle, and each category occupies a sector. We then use an arrow pointing to a particular sector to indicate the author's participation in the conferences in this category, where the level of participation is indicated by the length of the arrow.

It can be observed from Figure 5 that Jiawei Han

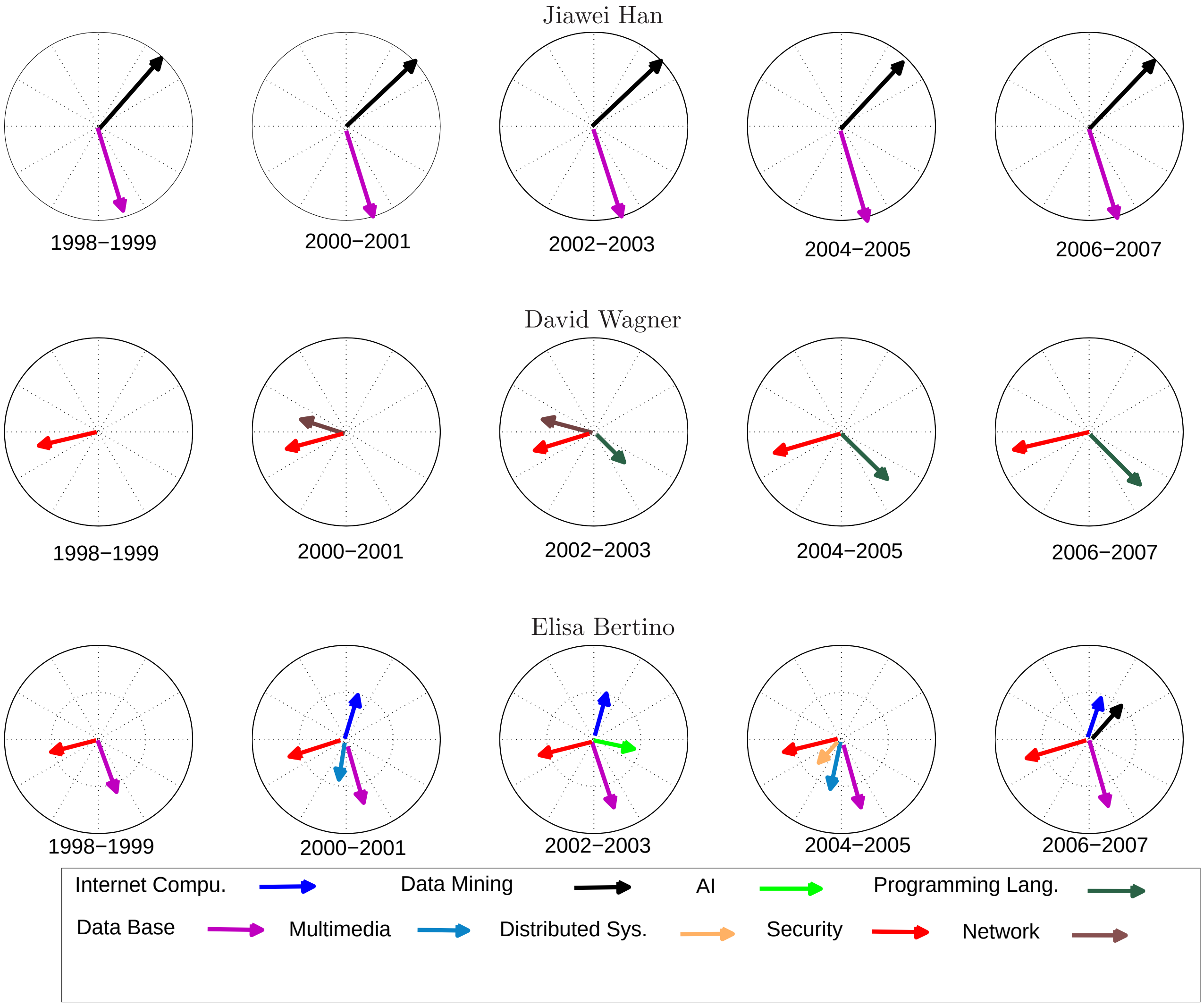

Figure 5: The evolution patterns of three authors identified by the proposed probabilistic model.

was actively participating Data Mining and Data Base conferences across all five time points, and this pattern remains very stable across years. On the other hand, David Wagner showed some change of research interests. He is actively participating Security conferences across all years. During 2000-2001, he developed interests in Network, and this is maintained through 2002-2003 before he smoothly switched to Programming Language. Elisa Bertino showed very dynamic change of research interests during this ten-year period. She is actively participating Data Base and Security conferences across all years. During some period of time, she also participated Internet Computing, Distributed Systems, AI, and Data Mining conferences. These results demonstrate that the proposed methods can identify smooth evolution of author's research interests over years.

6 Conclusions and Discussions

This paper studies the evolutionary co-clustering of time-varying data for the identification of smooth block structures. To overcome the limitation of existing methods and enable a probabilistic interpretation of the results, we propose a probabilistic model for evolutionary co-clustering. We propose an EM algorithm to perform maximum likelihood parameter estimation for the probabilistic model. The proposed methods are evaluated on both synthetic and real data sets. Results show that the proposed method consistently outperforms prior methods.

In this work, we describe a method for unsupervised learning from bipartite graphs. In many applications, the relational data are more conveniently captured by k-partite graphs [20]. We will extend our methods for unsupervised mining of dynamic k-partite graphs. In this paper, we assume that the number of co-clusters across all time points is the same. We will extend our method to this more general setting in the future.

Acknowledgments

We thank Hanghang Tong and Fei Wang for providing the DBLP data, Yun Chi and Yu-Ru Lin for many insightful discussions. This research is supported by NSF DBI-1147134 and by Old Dominion University Office of Research.

References

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, *A framework for clustering evolving data streams*, in Proceedings of the 29th International Conference on Very Large Data Bases, pp. 81–92.

[2] S. Asur, S. Parthasarathy, and D. Ucar, *An event-based framework for characterizing the evolutionary behavior of interaction graphs*, in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 913–921.

[3] F. R. Bach and M. I. Jordan, *Learning spectral clustering, with application to speech separation*, J. Mach. Learn. Res., 7 (2006).

[4] D. Chakrabarti, R. Kumar, and A. Tomkins, *Evolutionary clustering*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 554–560.

[5] Y. Cheng and G. M. Church, *Biclustering of expression data*, in Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, 2000, pp. 93–103.

[6] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, *On evolutionary spectral clustering*, ACM Transactions on Knowledge Discovery from Data, 3 (2009), pp. 17:1–17:30.

[7] F. R. K. Chung, *Spectral Graph Theory*, 1997.

[8] I. S. Dhillon, *Co-clustering documents and words using bipartite spectral graph partitioning*, in Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 269–274.

[9] I. S. Dhillon, Y. Guan, and B. Kulis, *Kernel k-means: spectral clustering and normalized cuts*, in Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 551–556.

[10] I. S. Dhillon, S. Mallela, and D. S. Modha, *Information-theoretic co-clustering*, in Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 89–98.

[11] E. Giannakidou, V. Koutsonikola, A. Vakali, and Y. Kompatsiaris, *Co-clustering tags and social data sources*, in Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management, 2008, pp. 317–324.

[12] G. H. Golub and C. F. van Loan, *Matrix computations (3. ed.)*, Johns Hopkins University Press, 1996.

[13] N. Green, M. Rege, X. Liu, and R. Bailey, *Evolutionary spectral co-clustering*, in The 2011 International Joint Conference on Neural Networks, 2011, pp. 1074–1081.

[14] J. A. Hartigan, *Direct clustering of a data matrix*, Journal of the American Statistical Association, 67 (1972), pp. 123–129.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn, *Data clustering: a review*, ACM Computing Surveys, 31 (1999), pp. 264–323.

[16] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, *Spectral biclustering of microarray data: Coclustering genes and conditions*, Genome Research, 13 (2003), pp. 703–716.

[17] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[18] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters*, ACM Transactions on Knowledge Discovery from Data, 1 (2007).

[19] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, *Analyzing communities and their evolutions in dynamic social networks*, ACM Transactions on Knowledge Discovery from Data, 3 (2009), pp. 8:1–8:31.

[20] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu, *Unsupervised learning on k-partite graphs*, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 317–326.

[21] U. Luxburg, *A tutorial on spectral clustering*, Statistics and Computing, 17 (2007), pp. 395–416.

[22] S. C. Madeira and A. L. Oliveira, *Biclustering algorithms for biological data analysis: A survey*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1 (2004), pp. 24–45.

[23] Q. Mei and C. Zhai, *Discovering evolutionary theme patterns from text: an exploration of temporal text mining*, in Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005, pp. 198–207.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, in Advances in Neural Information Processing Systems 14, 2001, pp. 849–856.

[25] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., (2000), pp. 888–905.

[26] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, *GraphScope: parameter-free mining of large time-evolving graphs*, in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 687–696.

[27] H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos, *Proximity tracking on time-evolving bipartite graphs*, in Proceedings of the SIAM International Conference on Data Mining, 2008, pp. 704–715.

[28] F. Wang, H. Tong, and C.-Y. Lin, *Towards evolutionary nonnegative matrix factorization*, in Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[29] K. Yu, S. Yu, and V. Tresp, *Soft clustering on graphs*, in Advances in Neural Information Processing Systems 18, Y. Weiss, B. Schölkopf, and J. Platt, eds., MIT Press, Cambridge, MA, 2006, pp. 1553–1560.

[30] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, *Bipartite graph partitioning and data clustering*, in Proceedings of the tenth International Conference on Information and Knowledge Management, 2001, pp. 25–32.