

# Identifying Evolving Groups in Dynamic Multimode Networks

Lei Tang, *Member, IEEE*, Huan Liu, *Senior Member, IEEE*, and  
Jianping Zhang

**Abstract**—A multimode network consists of heterogeneous types of actors with various interactions occurring between them. Identifying communities in a multimode network can help understand the structural properties of the network, address the data shortage and unbalanced problems, and assist tasks like targeted marketing and finding influential actors within or between groups. In general, a network and its group structure often evolve unevenly. In a dynamic multimode network, both group membership and interactions can evolve, posing a challenging problem of identifying these evolving communities. In this work, we try to address this problem by employing the temporal information to analyze a multimode network. A temporally regularized framework and its convergence property are carefully studied. We show that the algorithm can be interpreted as an iterative latent semantic analysis process, which allows for extensions to handle networks with actor attributes and within-mode interactions. Experiments on both synthetic data and real-world networks demonstrate the efficacy of our approach and suggest its generality in capturing evolving groups in networks with heterogeneous entities and complex relationships.

**Index Terms**—Data mining, community detection, community evolution, multimode networks, dynamic networks.

## 1 INTRODUCTION

OWING to the widely available network data produced from social networks, technology networks, information networks and genetic regulatory networks [2], network analysis [3] and modeling [4] are attracting increasing attention from many fields. Examples include epidemiology [5], intelligence analysis [6], targeted marketing, recommendation systems [7], relational learning [8], and behavior prediction [9].

A large body of existing work deals with networks of one mode. That is, only one type of actors (nodes) is present in a network, and the connections (interactions) between actors are of the same type. This is common for friendship networks and mobile networks. Recently, burgeoning applications such as web mining, collaborative filtering, and online targeted marketing involve more than one type of entities. Between them are different types of interaction. This kind of network is called *multimode network* [3] (a.k.a. *heterogeneous network*).

### 1.1 Multimode Networks

Take the network in YouTube as an example. A 3-mode network (shown in Fig. 1) can be constructed: users, videos, and tags. Note that in this network, both videos and tags are considered “actors” as well, although the user might be the major mode under consideration. Different interactions

exist between the three types of entities: users can upload videos; users can add tags to a video. Videos and tags are naturally correlated to each other. Meanwhile, a friendship network exists between users, and a video clip can be uploaded to respond to another video. Tags can also connect to each other based on their semantic meanings. In other words, multiple types of entities exist in the same network, and entities relate to others (either the same type or different types) through different links.

Another example of multimode network is the field of academia as shown in Fig. 2. Assorted entities (researchers, conference/journals, papers, words) are intertwined with each other. The scientific literature connects papers by citations; papers are published at different places (conferences, journals, workshops, thesis, etc.); and researchers are connected to papers through authorship. Some might also relate to each other by serving simultaneously as journal editors or on conference program committees. Moreover, each paper can focus on different topics represented by words. Words are associated to each other based on semantics. At the same time, papers are connected to different conferences and journals.

Within a multimode network, different types of entities tend to form groups or communities.<sup>1</sup> In the YouTube example, users sharing similar interests are more likely to form a group; videos are clustered naturally if they relate to similar contents; and tags are clustered if they are associated with similar users and videos. Generally, a user group may interact with multiple groups of another mode, i.e., users might have multiple interests, thus relating to various video or tag groups.

Multimode networks present entities in interrelated forms. One might wonder whether it is necessary to consider this multimode property. For instance, in the

• L. Tang and H. Liu are with the Department of Computer Science and Engineering, Arizona State University, PO Box 878809, Tempe, AZ 85287-8809. E-mail: {L.Tang, huan.liu}@asu.edu.

• J. Zhang is with The MITRE Corporation, McLean, VA 22102. E-mail: jzhang@mitre.org.

Manuscript received 18 Sept. 2009; revised 20 Mar. 2010; accepted 2 May 2010; published online 18 July 2011.

Recommended for acceptance by A. Zhang.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-09-0668. Digital Object Identifier no. 10.1109/TKDE.2011.159.

1. Group and community are interchangeable in this work.

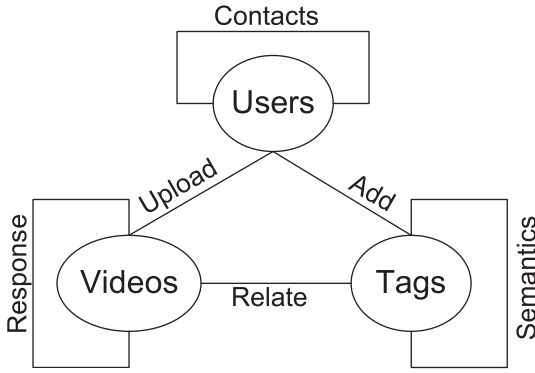


Fig. 1. An example of 3-mode network in YouTube.

YouTube example, can we focus only on friendship network between users? If so, we might miss information from other modes. Online social networks tend to be noisy, some users might have thousands of online friends while a substantial number of users might connect to only one or no friends. On the other hand, users might engage in various other activities like uploading videos and adding tags. Information distilled from other modes can help uncover the latent community structure of users.

Some work has been done to identify communities in a network of heterogeneous entities or relations [10], [11], [12] in terms of *multitype relational clustering*. However, these methods concentrate on static networks. In reality, networks tend to evolve gradually. The communities inside a network could grow or shrink, and the membership for specific actors shift gradually as well [13], [14], [15], [16]. This presents a challenge to identify group evolutions.

## 1.2 Group Evolution

In a multimode network, actors of different modes can evolve differently. For instance, in the previous academia example, interactions between different modes change every year. Within the network, researchers can divert their personal research interests, and the “hot” topics of a field may also change. On the contrary, communities of mode “journals/conferences” tend to be stable. But two venues may also connect due to the topic change. For instance, social network analysis joins researchers from diverse fields recently, thus making those journals on social sciences more related to magazines concentrating on data mining.

Facing heterogeneous entities with dynamic interactions, discovering evolving groups can lead to a clear understanding of interaction between disparate modes as well as long-term evolution patterns. This can benefit visualization of a complex network with heterogeneous entities and interactions, aid decision making in various domains, and signal an event alarm if undesirable evolution patterns are observed in the early stage. For example, to detect user interests shift for more effective targeted marketing, or to detect suspicious financial activities if an abnormal change in transactions is detected.

However, the problem of discovering community evolution becomes challenging in dynamic multimode networks because 1) the evolutions of different modes become correlated; and 2) disparate modes demonstrate distinctive

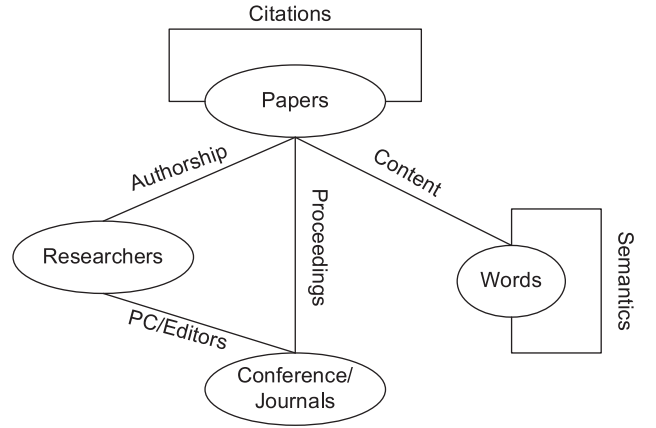


Fig. 2. A multimode network of academic publications.

evolution patterns. In this work, we propose a general model to identify evolving groups in dynamic multimode networks. A spectral clustering framework is adopted. A dynamic multimode network consists of a series of network snapshots. We aim to find out how a community evolves. In this model, a regularization term is added to reflect the effect of temporal change. It is shown that clustering results of interacted modes and neighboring timestamps can be considered as attributes for updating communities of one mode. This is a novel way of connecting dynamic multimode network analysis to conventional attribute-based data mining. An iterative algorithm, which does not suffer from hardware constraints even with tens of thousands actors, is presented to find an optimal solution of the model. The experiments on both synthetic and real-world large-scale network data demonstrate the efficacy of our algorithm and suggest its generality in solving problems with complex relationships.

The paper is organized as follows: we present a general framework to find community evolution in dynamic multimode networks in Section 2, introduce an iterative solver in Section 3 and study its convergence property in Section 4. In Sections 5 and 6, we report extensive experiments on both synthetic and real-world data to study the efficacy and efficiency of our framework. We review some related literature in Section 7 and conclude in Section 8.

## 2 PROBLEM FORMULATION

In order to extract communities in a multimode network, we follow the framework presented in [11]. Given an  $m$ -mode network with  $m$  types of actors  $\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_m$ , we aim to find how the latent community of each mode evolves. In our framework, we only consider discrete timestamps by taking a series of snapshots, which is commonly adopted in network analysis with temporal information [15], [16]. For a snapshot at time  $t$ , a network  $N^t$  is represented as multiple interactions between different modes. Let  $R_{i,j}^t \in \mathbb{R}^{n_i \times n_j}$  denote the interaction between two modes  $\mathbb{X}_i$  and  $\mathbb{X}_j$  at timestamp  $t$ ,  $n_i$  and  $n_j$  the number of actors at mode  $i$  and  $j$ ,  $k_i$  and  $k_j$  the number of latent communities for  $\mathbb{X}_i$  and  $\mathbb{X}_j$ , respectively. We will review the concept of block model approximation for community

TABLE 1  
Symbols and Denotations

Symbol	Denotation
$m$	number of modes
$n_i$	number of actors at mode $i$
$\mathbb{X}_i$	entities at mode $i$
$R_{i,j}^t$	the interaction between two modes $i$ and $j$ at time $t$
$k_i$	number of latent groups of mode $i$
$C^{(i,t)}$	the community indicator matrix of mode $i$ at time $t$
$A_{i,j}^t$	group interaction density between modes $i$ and $j$
$w_a^{(i,j)}$	weight associated with interaction between modes $i$ and $j$
$w_b^{(i)}$	weight associated with temporal regularization

discovery and discuss incorporating temporal regularization for dynamic multimode networks. For simplicity, the discussion below does not consider within-mode interactions. Extensions to handle within-mode interactions are discussed later. For presentation convenience, symbols used in the subsequent derivation are listed in Table 1.

## 2.1 Community Discovery via Block Model Approximation

In order to extract communities at each mode, we assume interactions between modes can be approximated by interactions between groups [17], [18]. This basic idea can be visualized in Fig. 3. In the figure, the network on the left shows the interactions between two modes, each with 400 and 600 actors, respectively. The one on the right demonstrates the effect after we reorder the actors according to their latent community membership. After reordering, it shows a crisp block structure. Thus, we can approximate the original network through interaction blocks. In particular,

$$R_{i,j}^t \approx C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T,$$

where  $C^{(i,t)} \in \{0,1\}^{n_i \times k_i}$  denotes latent cluster (block) membership for  $\mathbb{X}_i$  at timestamp  $t$ , and  $A_{i,j}$  represents the density of group (block) interaction. In other words, the group membership determines how two actors interact. This essentially hinges on a similar assumption as stochastic block models [19]. The difference is that stochastic block models deal with the problem from a probabilistic aspect. Here, we identify the block structure of multimode networks via matrix approximation:

$$\min \|R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T\|_F^2 \quad (1)$$

$$s.t. \quad C^{(i,t)} \in \{0,1\}^{n_i \times k_i}, \quad \sum_{k=1}^{k_i} C_{*k}^{(i,t)} = 1, \quad (2)$$

$$C^{(j,t)} \in \{0,1\}^{n_j \times k_j}, \quad \sum_{k=1}^{k_j} C_{*k}^{(j,t)} = 1. \quad (3)$$

The constraints in (2) and (3) force each row of the indicator matrix to have only one entry being 1. That is, each actor belongs to only one community. Unfortunately, the discreteness of the constraints makes the problem NP-hard. A strategy that has been well studied in spectral clustering

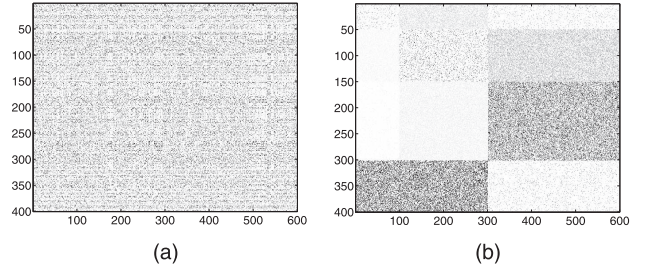


Fig. 3. Networks before and after reordering.

[20] is to allow the cluster indicator matrix to be continuous and relax the hard clustering constraint as follows:

$$(C^{(i,t)})^T C^{(i,t)} = I_{k_i}.$$

In a multimode network, diverse interactions occur between different modes. Hence, the objective in (1) can be changed to

$$\min \sum_{1 \leq i < j \leq m} w_a^{(i,j)} \|R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T\|_F^2, \quad (4)$$

with  $w_a^{(i,j)}$  being the weights associated with different interactions.

## 2.2 Community Discovery with Network Sequence

With a dynamic multimode network, we have multiple snapshots of the network. Naturally, the objective function *without* considering its temporal effect can be formulated as  $\mathbf{F}_1$ :

$$\min \sum_{t=1}^{\mathbb{T}} \sum_{1 \leq i < j \leq m} w_a^{(i,j)} \|R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T\|_F^2 \quad (5)$$

$$s.t. \quad (C^{(i,t)})^T C^{(i,t)} = I_{k_i} \quad i = 1, \dots, m, \quad t = 1, \dots, \mathbb{T}. \quad (6)$$

Then, we have the following theorem.

**Theorem 1.** Let  $C^{(i,t)}, 1 \leq i \leq m, 1 \leq t \leq \mathbb{T}$  be a valid solution of  $\mathbf{F}_1$ , then  $\tilde{C}^{(i,t)}$  defined below is also a valid solution with the same objective value

$$s.t. \quad \begin{aligned} \tilde{C}^{(i,t)} &= C^{(i,t)} Q^{(i,t)}, \\ (Q^{(i,t)})^T Q^{(i,t)} &= Q^{(i,t)} (Q^{(i,t)})^T = I_{k_i}, \\ Q^{(i,t)} &\in \mathbb{R}^{k_i \times k_i}. \end{aligned}$$

**Proof.** It suffices to show that the value of each single term in  $\mathbf{F}_1$  does not change. Given a solution  $C^{(i,t)}$  and  $A_{i,j}^t$ , we can choose  $\tilde{A}_{i,j}^t = (Q^{(i,t)})^T A_{i,j}^t Q^{(j,t)}$ , then

$$\tilde{C}^{(i,t)} \tilde{A}_{i,j}^t (\tilde{C}^{(j,t)})^T = C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T,$$

which completes the proof.  $\square$

## 2.3 Community Discovery with Temporal Regularization

The formulation  $\mathbf{F}_1$  does not consider the relationship between consecutive timestamps. Solving  $\mathbf{F}_1$  boils down to perform clustering at each snapshot independently. In reality, communities tend to evolve gradually. To obtain smooth community evolution, we add a temporal regularization term  $\Omega$  which forces the clustering sequence to be smooth across different timestamps

$$\Omega = \frac{1}{2} \sum_{t=2}^{\mathbb{T}} \|C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T\|_F^2. \quad (7)$$

Here, the coefficient 1/2 is included due to notational conveniences for later derivation. Indeed, we are making a first-order Markov assumption. That is, the current clustering should be similar to the clustering at the previous timestamp. Note that we do not take the regularization as

$$\Omega = \sum_{t=2}^{\mathbb{T}} \|C^{(i,t)} - C^{(i,t-1)}\|_F^2, \quad (8)$$

which seems more natural at first glimpse. As demonstrated in Theorem 1,  $C^{(i,t)}$  is equivalent under an orthogonal transformation. Hence, comparing  $C^{(i,t)}$  and  $C^{(i,t-1)}$  directly as in (8) does not necessarily capture the difference between the cluster indicators at different timestamps. On the contrary, the regularization term of (7) is independent of the orthogonal transformation, thus captures the difference of community structure of neighboring timestamps. With this regularization, the problem of identifying evolving groups can be formulated as

$$\begin{aligned} \mathbf{F}_2 : \min & \sum_{t=1}^{\mathbb{T}} \sum_{i < j} w_a^{(i,j)} \|R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T\|_F^2 \\ & + \frac{1}{2} \sum_{i=1}^m w_b^{(i)} \sum_{t=2}^{\mathbb{T}} \|C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T\|_F^2 \end{aligned} \quad (9)$$

$$s.t. \quad (C^{(i,t)})^T C^{(i,t)} = I_{k_i}, \quad i = 1, \dots, m, \quad t = 1, \dots, \mathbb{T}, \quad (10)$$

with  $w_b^{(i)}$  being the trade-off between the block model approximation of interactions and the temporal regularization. As evolution takes effect gradually, we aim to find a community structure that is consistent with interaction matrix, whereas not drastically different from that of the previous timestamp.

### 3 TEMPORALLY REGULARIZED MULTIMODE CLUSTERING

To capture evolving groups in dynamic multimode networks, we have to solve  $\mathbf{F}_2$ . There is no analytical solution to the problem, but an iterative algorithm can be derived. We show that a closed-form solution exists for  $A_{i,j}^t$  and  $C^{(i,t)}$  if other variables are fixed. Then we present the algorithm in an attribute view for easy comprehension and extension.

#### 3.1 Computation of $A$

**Theorem 2.** Given  $C^{(i,t)}$ , the optimal group interaction matrix  $A_{i,j}^t$  can be calculated as

$$A_{i,j}^t = (C^{(i,t)})^T R_{i,j}^t C^{(j,t)}.$$

**Proof.** Since  $A_{i,j}^t$  is only involved in one term in  $\mathbf{F}_2$ , we can focus on the single term to optimize  $A_{i,j}^t$

$$\begin{aligned} & \|R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T\|_F^2 \\ & = \text{tr}[(R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T)(R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T)^T] \\ & = \text{tr}[R_{i,j}^t (R_{i,j}^t)^T - 2C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T (R_{i,j}^t)^T + A_{i,j}^t (A_{i,j}^t)^T]. \end{aligned}$$

The last equation follows as  $\text{tr}(AB) = \text{tr}(BA)$  and  $C^{(i,t)}$  and  $C^{(j,t)}$  are column orthogonal as in (10). Taking the derivative of the last equation with respect to  $A_{i,j}^t$  to zero, we have

$$A_{i,j}^t = (C^{(i,t)})^T R_{i,j}^t C^{(j,t)}.$$

The proof is completed.  $\square$

#### 3.2 Computation of $C$

Given the optimal  $A_{i,j}^t$ , it can be verified that

$$\begin{aligned} & \|R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T\|_F^2 \\ & = \|R_{i,j}^t\|_F^2 - \|(C^{(i,t)})^T R_{i,j}^t C^{(j,t)}\|_F^2. \end{aligned} \quad (11)$$

Meanwhile,

$$\begin{aligned} & \frac{1}{2} \|C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T\|_F^2 \\ & = \frac{1}{2} \text{tr}[C^{(i,t)}(C^{(i,t)})^T + C^{(i,t-1)}(C^{(i,t-1)})^T \\ & \quad - 2C^{(i,t)}(C^{(i,t)})^T C^{(i,t-1)}(C^{(i,t-1)})^T] \\ & = k_i - \|(C^{(i,t)})^T C^{(i,t-1)}\|_F^2. \end{aligned} \quad (12)$$

Since  $\|R_{i,j}^t\|_F^2$  in (11) and  $k_i$  in (12) are constants, we can transform  $\mathbf{F}_2$  into the following objective:

$$\begin{aligned} \mathbf{F}_3 : \max & \sum_{t=1}^{\mathbb{T}} \sum_{i < j \leq m} w_a^{(i,j)} \|(C^{(i,t)})^T R_{i,j}^t C^{(j,t)}\|_F^2 \\ & + w_b^{(i)} \sum_{t=2}^{\mathbb{T}} \sum_{i=1}^m \|(C^{(i,t)})^T C^{(i,t-1)}\|_F^2. \end{aligned} \quad (13)$$

Note that  $C^{(i,t)}$  is interrelated with both  $C^{(j,t)}$  and  $C^{(i,t-1)}$ . In general, there is no analytical closed-form solution. But the optimal  $C^{(i,t)}$  can be obtained directly if  $C^{(j,t)}$  and  $C^{(i,t-1)}$  are given, which is stated in the following theorem.

**Theorem 3.** Given  $C^{(j,t)}$  and  $C^{(i,t-1)}$ ,  $C^{(i,t)}$  can be computed as the top left singular vectors of the matrix  $P_i^t$  concatenated by the following matrices in column-wise:

$$P_i^t = \left[ \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \sqrt{w_b^{(i)}} C^{(i,t-1)} \right]. \quad (14)$$

**Proof.** We focus only on those terms involving  $C^{(i,t)}$  in the objective function. Without loss of generality, we discuss the cases when  $2 \leq t \leq \mathbb{T} - 1$  first

$$\begin{aligned} L & = \sum_{i < j} w_a^{(i,j)} \|(C^{(i,t)})^T R_{i,j}^t C^{(j,t)}\|_F^2 \\ & \quad + \sum_{k < i} w_a^{(k,i)} \|(C^{(k,t)})^T R_{k,i}^t C^{(i,t)}\|_F^2 \\ & \quad + w_b^{(i)} \|(C^{(i,t)})^T C^{(i,t-1)}\|_F^2 \\ & \quad + w_b^{(i)} \|(C^{(i,t+1)})^T C^{(i,t)}\|_F^2 \\ & = \text{tr}[(C^{(i,t)})^T M_i^t C^{(i,t)}], \end{aligned} \quad (15)$$

where  $M_i^t$  is defined as

$$\begin{aligned} M_i^t = & \sum_{i < j} w_a^{(i,j)} R_{i,j}^t C^{(j,t)} (C^{(j,t)})^T (R_{i,j}^t)^T \\ & + \sum_{k < i} w_a^{(k,i)} (R_{k,i}^t)^T C^{(k,t)} (C^{(k,t)})^T R_{k,i}^t \\ & + w_b^{(i)} C^{(i,t-1)} (C^{(i,t-1)})^T \\ & + w_b^{(i)} C^{(i,t+1)} (C^{(i,t+1)})^T. \end{aligned} \quad (16)$$

So the problem boils down to a max-trace problem with orthogonality constraint as in (6). According to Ky-Fan theorem [21], this max-trace problem has a closed-form solution, which corresponds to the subspace spanned by the top  $k_i$  eigenvectors of  $M_i^t$ .

Note that  $M_i$  is a square symmetric matrix of size  $n_i \times n_i$ . When the number of actors in a mode  $\mathbb{X}_i$  is huge, direct calculating  $M_i^t$  and its eigenvectors can be problematic. Alternatively, we represent  $M_i^t$  in the following matrix form:

$$M_i^t = P_i^t * (P_i^t)^T, \quad (17)$$

where  $P_i^t$  is concatenated by the following matrices in *column-wise* (the order of terms does not matter):

$$\begin{aligned} P_i^t = & \begin{bmatrix} \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{i < j}, \\ \left\{ \sqrt{w_a^{(k,i)}} (R_{k,i}^t)^T C^{(k,t)} \right\}_{k < i}, \\ \sqrt{w_b^{(i)}} C^{(i,t-1)}, \\ \sqrt{w_b^{(i)}} C^{(i,t+1)} \end{bmatrix} \\ = & \begin{bmatrix} \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \sqrt{w_b^{(i)}} C^{(i,t \pm 1)} \end{bmatrix}. \end{aligned}$$

For  $t = 1$  or  $t = \mathbb{T}$ , we only need to keep  $C^{(i,t+1)}$  or  $C^{(i,t-1)}$ , respectively, instead of  $C^{(i,t \pm 1)}$ . Typically, the size of  $P_i^t$  is much smaller compared to  $M_i^t$  if the number of clusters of each mode is small. Let the singular value decomposition (SVD) of  $P_i$  as  $P_i^t = U \Sigma V$ , then  $M_i^t = U \Sigma^2 U^T$ . That is, the top left-singular vectors of  $P_i^t$  correspond to the top eigenvectors of  $M_i^t$ , which completes the proof.  $\square$

### 3.3 Algorithm in Attribute View

To solve the problem  $F_3$  in (13), we resort to alternating optimization. That is, we fix all the other variables while solving  $C^{(i,t)}$ . This process is iterated until convergence.<sup>2</sup> After convergence,  $\{C^{(i,t)}\}$  are the approximate community indicator matrices. One problem to be addressed is to recover the discrete partition of communities. One commonly used postprocessing scheme is to apply k-means clustering to the community indicators [22]. Combined with the results of previous sections, we have the temporally regularized multimode clustering algorithm as shown in Fig. 4.

Interestingly, the algorithm can be interpreted in an attribute view and each step to update  $C^{(i,t)}$  corresponds to a process of latent semantic analysis (LSA). According to

### Temporally-Regularized Multi-Mode Clustering

**Input:**  $R, k_i, w_a^{(i,j)}, w_b^{(i)}$ ;

**Output:**  $idx^{(i,t)}, C^{(i,t)}, A_{i,j}^t$ .

1. Generate initial cluster indicator matrix  $C^{(i,t)}$ .
2. **Repeat**
3.   **For**  $t = 1 : \mathbb{T}, i = 1 : m$
4.     shrink / expand  $C^{(i,t \pm 1)}$  if necessary;
5.     calculate  $P_i^t$  (or  $M_i^t$ ) as in Theorem 3;
6.     calculate SVD of  $P_i^t$  (or eigenvectors of  $M_i^t$ );
7.     update  $C^{(i,t)}$  as top left singular (eigen) vectors;
8. **Until** the relative change of the objective ( $F_3$ )  $\leq \epsilon$ .
9. calculate  $A_{i,j}^t$  as in Theorem 2;
10. calculate the cluster  $idx^{(i,t)}$  with k-means on  $C^{(i,t)}$ .

Fig. 4. Algorithm: temporally regularized multimode clustering.

Theorem 3, the community indicator  $C_i^t$  corresponds to the left singular vectors of  $P_i^t$ , which is defined as in (14). Consider  $P_i^t$  as normal instance-attribute matrix, then finding out the community indicator is essentially equivalent to perform latent semantic analysis [23].

LSA is commonly used in text analysis to capture latent semantic relationship between terms. Given a document-term matrix, LSA relies on singular value decomposition to extract lower-dimensional embeddings of documents and the relationship between different terms. Our problem takes the same procedure as LSA, but the so-called “document-term” matrix is constructed based on the community indicators of interacting modes and neighboring timestamps as in (14). That is, to capture the community structure of one mode at one timestamp, the community indicators of its interacting modes and neighboring timestamps form the features. And the approximate clustering is equivalent to the lower-dimensional embedding obtained via LSA.

This naturally connects the analysis of dynamic multimode networks to attribute-based data analysis as in conventional data mining. The key difference is that the “features” in multimode networks are dynamically updated based on “features” of other modes and timestamps, which are obtained via LSA. This iterative-LSA process is repeated until an equilibrium state is reached. An overview of our algorithm in attribute view is illustrated in Fig. 5.

### 3.4 Extensions to Real-World Networks

With an attribute view to interpret our proposed algorithm, it is simple to extend our framework to handle multimode networks with various properties.

#### 3.4.1 Actor Attributes

The attribute view of the algorithm allows for the integration of entity attributes in a convenient way. When a multimode network has attributes for certain modes of actors, we can simply add these attributes as features in  $P_i^t$  (14) when updating the cluster indicator matrix. That is,

$$\tilde{P}_i^t = \begin{bmatrix} \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \sqrt{w_b^{(i)}} C^{(i,t \pm 1)}, \sqrt{w_c^{(i)}} F_i^t \end{bmatrix},$$

where  $F_i^t$  denotes the attributes for actors in mode  $i$ , and  $w_c^{(i)}$  the weight for actor attributes.

2. The convergence property will be discussed in detail in Section 4.

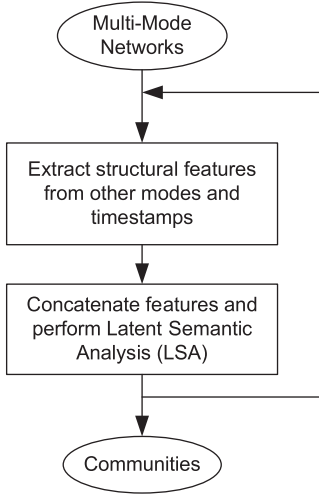


Fig. 5. Algorithm in attribute view: iterative-LSA.

### 3.4.2 Higher Order Temporal Regularization

In our formulation, we make the first-order Markov assumption that a community snapshot should be similar to its previous snapshot. In some domains, it might be necessary to include higher order temporal regularization. An  $\ell$ th order temporal regularization enforces the community structure at timestamp  $t$  to be similar to that of  $t-1, t-2, \dots, t-\ell$ . Based on the attribute view of the algorithm, we just need to modify  $P_i^t$  as follows to reflect the higher order dependency

$$\left[ \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \left\{ \sqrt{w_b^{(i)}} \alpha^{k-1} C^{(i,t \pm k)} \right\}_{k=1, \dots, \ell} \right].$$

Here,  $0 < \alpha \leq 1$  is a decay factor to tune the regularization effect of different snapshots. Those snapshots that are close to current timestamp play a more important role in regularization.

### 3.4.3 Within-Mode Interactions

In certain scenarios, within-mode interactions might occur: e.g., the interactions in the friendship network of the 3-mode network in YouTube (Fig. 1) and those in the paper citations in the example of academic publications (Fig. 2). As we have shown,  $P_i^t$  represents the attributes for clustering. Then,  $M_i^t = P_i^t (P_i^t)^T$  is like a similarity matrix between data instances. So for an undirected within-mode interaction, we can just add the within-mode interaction matrix to  $M_i^t$  (16)

$$\begin{aligned} \widetilde{M}_i^t = & \sum_{i < j} w_a^{(i,j)} R_{i,j}^t C^{(j,t)} (C^{(j,t)})^T (R_{i,j}^t)^T \\ & + \sum_{k < i} w_a^{(k,i)} (R_{k,i}^t)^T C^{(k,t)} (C^{(k,t)})^T R_{k,i}^t \\ & + w_b^{(i)} C^{(i,t-1)} (C^{(i,t-1)})^T \\ & + w_b^{(i)} C^{(i,t+1)} (C^{(i,t+1)})^T \\ & + w_c^{(i)} R_{i,i}^t. \end{aligned}$$

Here,  $R_{i,i}^t$  denotes the within-mode interaction of mode  $i$  at time  $t$ , and  $w_c^{(i)}$  denotes the weight associated with the within-mode interaction.

### 3.4.4 Dormant and Emerging Actors

Some actors might become inactive at a certain time period. Meanwhile, new actors might join the network at a specific timestamp. The framework earlier assumes actors do not change. Here, we analyze more realistic cases where actors become dormant or join a network.

Note that our derived algorithm just uses  $C^{(i,t \pm 1)}$  as attributes for current timestamp. When an actor leaves or hibernates at certain timestamp, we can delete his corresponding entry in  $C^{(i,t \pm 1)}$  when updating  $C^{(i,t)}$ . However, after deletion, the orthogonality constraints might not be satisfied. Since our framework is an approximation to hard clustering, a tiny deviation from orthogonality does not affect the performance. However, when many actors become inactive, the total weights of remaining actors in  $C^{(i,t \pm 1)}$  is relatively small, thus playing a less important role in the computation of the similarity matrix  $M_i^t$  in (16). This is reasonable. A drastic membership change denotes the latent community is experiencing an “overhaul,” thus it is not necessary to overregularize the temporal change. When mode  $\mathbb{X}_i$  at timestamp  $t$  has new actors, we simply set their entries in  $C^{(i,t \pm 1)}$  to 0.

### 3.4.5 Online Clustering

In some real-world scenarios, one might need to track the group evolution in an online fashion, and the algorithm should be modified accordingly. Instead of iteratively updating  $C^{(i,t)}$ , we update the community structure at different timestamps only once. Moreover, the corresponding updating matrix  $P_i^t$  and  $M_i^t$  involve only fixed  $C^{(i,t-1)}$ , but not  $C^{(i,t+1)}$ . That is,

$$\widetilde{P}_i^t = \left[ \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \sqrt{w_b^{(i)}} C^{(i,t-1)} \right].$$

In summary, actor attributes, community structure at different snapshots and within-mode interactions provide additional information for community detection. Based on the attribute view of our proposed algorithm, all these information can be considered as “attributes” or “similarity matrix” for computing community indicator matrix.

## 4 CONVERGENCE ANALYSIS

In the previous section, we provide an iterative algorithm to find communities in dynamic multimode networks. In this section, we present its convergence properties.

**Theorem 4.** *The objective of formulation  $\mathbf{F}_2$  in (9) following the temporally regularized multimode clustering algorithm in Fig. 4 is guaranteed to converge.*

**Proof.** Our algorithm essentially implements an alternating optimization strategy for each  $C^{(i,t)}$ . At each step, we find an optimal solution for one variable while fixing the other variables. Hence, the objective  $\mathbf{F}_2$  is nonincreasing. Since the objective of  $\mathbf{F}_2$  has a lower bound 0, it must converge to a constant value.  $\square$

However, it is impractical to compute  $\mathbf{F}_2$  explicitly. The involved terms  $R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T$  is a full matrix of size  $n_i \times n_j$ . The temporal regularization term



$C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T$  is also huge, of size  $n_i \times n_i$ . It can be problematic to compute them as that might require excessive computational resource. Alternatively, we can compute the equivalent formulation  $\mathbf{F}_3$  in (13). It involves matrices of size  $k_i \times k_j$  or  $k_i \times k_i$ . Since  $k_i \ll n_i$ ,  $\mathbf{F}_3$  can be computed much more efficiently.

Though the objective converges to a constant, it does not necessarily indicate that the algorithm converges to a (local) optimal, whose gradient is nonnegative (nonpositive) along with any direction within the feasible domain for a minimization (maximization) problem. For certain problems, the objective function approaches a constant value while the solution is far from even a local optimal (refer to [24] for concrete examples). Below, we further examine the convergence property in terms of community indicators  $\{C^{(i,t)}\}$ .

It can be shown that  $C_i^t$  is guaranteed to converge to a local stationary point under mild conditions. Let  $Q^{(i,t)} = C^{(i,t)}(C^{(i,t)})^T$ . Then we have some weak convergence result with respect to  $\{Q^{(i,t)}\}$ . The formulation  $\mathbf{F}_3$  in (13) can be reformulated in terms of  $\{Q^{(i,t)}\}$  as follows:

$$\mathbf{F}_4 : \max \sum_{t=1}^{\mathbb{T}} \sum_{1 \leq i < j \leq m} w_a^{(i,j)} \text{tr}[Q^{(j,t)} R_{i,j}^T Q^{(i,t)} R_{i,j}] \quad (18)$$

$$+ w_b^{(i)} \sum_{t=2}^{\mathbb{T}} \sum_{i=1}^m \text{tr}[Q^{(i,t)} Q^{(i,t-1)}]$$

$$\text{s.t.} \quad Q^{(i,t)} \in \mathcal{M}, \quad i = 1, \dots, m, \quad t = 1, \dots, \mathbb{T}, \quad (19)$$

$$\mathcal{M}^{(i,t)} = \{S | S = C^{(i,t)}(C^{(i,t)})^T\}.$$

It can be shown that  $Q^{(i,t)}$  converges to a *coordinate-wise optimal point* under some mild conditions. Before we proceed to the details, we would like to introduce some basic concepts and results of *block coordinate descent* (BCD) method [25] (a.k.a. *alternating optimization* [26]). Without loss of generality, we discuss the case of minimizing a function  $f(x_1, x_2, \dots, x_N)$  with each  $x_i$  denoting one block of variables. In each iteration, BCD optimizes one block of variables while fixing the other blocks. A basic *cyclic rule* is to choose the same block for iterations  $k, k + N, k + 2N, \dots$ , for  $k = 1, \dots, N$ . We restate Theorem 4.1(c) in [25] as a lemma below:

**Lemma 5.** Assume that the level set  $X^0 = \{x : f(x) \leq f(x^0)\}$  is compact and that  $f$  is continuous on  $X^0$ . Then the sequence  $\{x^r = (x_1^r, x_2^r, \dots, x_N^r)\}_{r=0,1,\dots}$  generated by block coordinate descent is defined and bounded. Moreover, if  $f(x_1, \dots, x_N)$  has at most one minimum in  $x_k$  for  $k = 2, \dots, N - 1$ , and the cyclic rule is used, then every cluster point  $z$  of  $\{x^r\}_{r \equiv (N-1) \bmod N}$  is a coordinate-wise minimum point of  $f$ .

Based on the lemma, we have the following theorem:

**Theorem 6.**  $Q^{(i,t)}$  converges to a coordinate-wise optimal point following our algorithm, if the  $k_i$ th and  $(k_i + 1)$ th singular values of  $P_i^t$  defined in (14) are different in each iteration.

**Proof.** Clearly,  $\mathbf{F}_4$  in (18) is continuous with respect to  $Q^{(i,t)}$ . If the  $k_i$  and  $(k_i + 1)$ th singular values of  $P_i^t$  defined in (14) are different, we have a *unique*  $Q^{(i,t)}$  that equals to the outer product of the top  $k_i^t$  left singular vectors of  $P_i^t$ . If this is always the case, i.e.,  $\mathbf{F}_4$  has at most one minimum for

each block optimization,  $Q^{(i,t)}$  converges to a coordinate-wise minimum point of  $\mathbf{F}_4$  according to Lemma 5.  $\square$

**Theorem 7.** Our temporally regularized clustering algorithm approaches a stationary point for  $\{C^{(i,t)}\}$  if the  $k_i$ th and  $(k_i + 1)$ th singular values of  $P_i^t$  defined in (14) are different in each iteration.

**Proof.** Based on Theorem 6,  $\{Q^{(i,t)}\}$  converges to a coordinate-wise minimum point if the  $k_i$ th and  $(k_i + 1)$ th singular values of  $P_i^t$  are always different. Let  $C^{(i,t)}$  be the corresponding decomposition of the limiting point  $Q^{(i,t)}$  following (19). To prove that the limiting point  $C^{(i,t)}$  is a stationary point, we can show that the gradient of the objective function within the feasible domain is 0. Specifically, we show that the gradient of the objective function projected into the tangent space of the feasible domain is 0. For notation convenience, we write  $C^{(i,t)}$  as  $C$ , and  $M_i^t$  as  $M$  in the subsequent proof.

Let  $Q^{(i,t)}$  denote the limiting point, i.e., a coordinate-wise optimal point. Its corresponding  $C$  must satisfy Theorem 3. In other words,

$$MC = C\Lambda, \quad (20)$$

where  $\Lambda$  denotes a diagonal matrix of the top eigenvalues of  $M$ . Following (15),  $\mathbf{F}_3$  in terms of  $C$  can be written as

$$\mathbf{F}_3(C) = \text{tr}[C^T MC] + \text{const},$$

with  $M$  defined in (17). So the gradient of  $\mathbf{F}_3$  with respect to  $C$  is

$$g(C) = 2MC. \quad (21)$$

Recall that  $C \in \mathbb{R}^{n_i \times k_i}$  is column orthogonal, which forms a smooth Stiefel manifold [27] of dimension  $n_i k_i - k_i(k_i + 1)/2$ . The projection of any  $Z \in \mathbb{R}^{n_i \times k_i}$  onto the tangent space of the manifold [28] is

$$\pi_T(Z) := C \frac{C^T Z - Z^T C}{2} + (I - CC^T)Z. \quad (22)$$

Hence, the gradient  $g(C)$  projected into  $C$ 's tangent space is

$$\begin{aligned} \pi_T(g(C)) &= C \frac{2C^T MC - 2C^T M^T C}{2} - 2(I - CC^T)MC \\ &= -2(I - CC^T)MC \end{aligned} \quad (23)$$

$$\begin{aligned} &= -2(I - CC^T)C\Lambda \\ &= 0, \end{aligned} \quad (24)$$

where (23) follows from the symmetry of  $M$ , and (24) follows from the fact that  $C$  corresponds to the top eigenvectors of  $M$  as stated in (20). Hence, the projected gradient of  $\mathbf{F}_3$  with respect to  $C$  within the feasible domain (the Stiefel manifold) is zero. Since the objective function is continuously differentiable and the projected gradient is zero for all the blocks of variables, the final solution of  $\{C\}$  is a stationary point (critical point).  $\square$

In Theorems 6 and 7, we require each block optimization step to satisfy certain properties for singular values. A simple strategy to check whether or not our solution is a stationary point, is to verify whether there is any case that the  $k_i$  and  $(k_i + 1)$ th singular values of  $P_i^t$  are exactly the same. In real-world multimode networks, this rarely happens due to noise. Hence, this convergence result is valid in general.

We want to emphasize that it is improper to use the difference of  $C^{(i,t)}$  between iterations as a guide for convergence verification. As shown in Theorem 3, the optimal  $C_i^t$  corresponds to the top left-singular vectors, which is *not unique*. But this does not refrain the algorithm from reaching a stationary point at the end. A proper convergence criterion is the difference of  $Q^{(i,t)}$  between iterations is sufficiently small. But  $Q^{(i,t)}$  is a full matrix of size  $n_i \times n_i$ , which might be computationally expensive. Since the obtained  $C^{(i,t)}$  is an approximate community indicator matrix, exact convergence may not be necessary. For practical use, it works fine to simply check whether the objective value of  $F_3$  stabilizes.

## 5 EXPERIMENTS ON SYNTHETIC DATA

Three methods are compared: static, online, and our proposed temporally regularized multimode clustering (denoted as *Static*, *Online*, and *Regularized*, respectively). Static clustering is the baseline approach, as it does not consider any temporal regularization. Static clustering performs clustering independently for each network snapshot by solving the formulation  $F_1$  in (5). For completeness, we also include the result of online clustering (as mentioned in Section 3.4).

Typically, real-world data does not contain the ground truth required for validation, i.e., the community membership of actors at different timestamps. Therefore, we first resort to synthetic data to show the efficacy of our proposed algorithm.

### 5.1 Experiment Setup

We construct a synthetic network of three modes with 2, 3, and 4 clusters, and 50, 100, and 200 actors, respectively. Interactions occur between each pair of modes. To generate the interaction, we

- determine the latent community for each actor; and
- generate interactions following a Bernoulli distribution based on the group identity of two interacted entities.

To simulate evolution, we generate interactions at different timestamps as follows:

- At each timestamp, a certain percentage (5-20 percent) of actors switch to another group different from that of the previous timestamp. This is to simulate the community evolution.
- The between-group interaction probability is sampled with a Gaussian distribution centered on the probability of the previous timestamp. This is to simulate the interaction change.

TABLE 2  
Average NMI on Synthetic Data over 100 Runs

Noise Level	Static	Online	Regularized
0.05	0.9251	0.9267	<b>0.9273</b>
0.10	0.8998	0.8997	<b>0.9102</b>
0.15	0.8911	0.8945	<b>0.9029</b>
0.20	0.8569	0.8574	<b>0.8695</b>
0.25	0.7875	0.7876	<b>0.7904</b>
0.30	0.7109	0.7135	<b>0.7242</b>
0.35	0.5888	0.5912	<b>0.6039</b>
0.40	0.4719	0.4760	<b>0.4881</b>
0.45	0.3177	0.3209	<b>0.3219</b>
0.50	0.1991	<b>0.2005</b>	0.1992
0.55	0.1200	<b>0.1215</b>	0.1135

The noise level is gradually increased from the top to the bottom. Bold entries denote the best one among the three methods: static, online, and temporally regularized multimode clustering.

- Noise is inserted at each timestamp. For example, given noise level 0.2, approximately 20 percent of entries in the interaction matrix are randomly selected to set to 0, and then another 20 percent of entries are randomly selected to set to 1. The noise is independent of latent group structure. Hence, the higher the noise level, the less observable a community structure is in the interaction.

To study the property of different clustering approaches, we generate data of various noise levels. For each level, 100 synthetic networks are constructed. For simplicity, we set all the weights  $w_a^{(i,j)}$  and  $w_b^i$  in (9) to 1. An algorithm runs until the relative change of its objective function is less than  $10^{-6}$ . Normalized mutual information (NMI) [29] is adopted to evaluate the clustering performance. Let  $\pi^a, \pi^b$  denote two different partitions of communities. NMI is defined as

$$NMI = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log \left( \frac{n \cdot n_{h,\ell}}{n_h^{(a)} \cdot n_\ell^{(b)}} \right)}{\sqrt{\left( \sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^{(a)}}{n} \right) \left( \sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^{(b)}}{n} \right)}},$$

where  $n$  is the total number of data instances,  $k^{(a)}$  and  $k^{(b)}$  represent the number of communities in partition  $\pi^a$  and  $\pi^b$ , respectively, and  $n_h^{(a)}$ ,  $n_\ell^{(b)}$ , and  $n_{h,\ell}$  are, respectively, the numbers of actors in the  $h$ th community of partition  $\pi^a$ , in the  $\ell$ th community of partition  $\pi^b$ , and in both the  $h$ th community of  $\pi^a$  and  $\ell$ th community of  $\pi^b$ . NMI is a measure between 0 and 1. NMI equals 1 when two partitions are equivalent.

### 5.2 Experiment Results

The average performance over 100 runs is reported in Table 2. As seen in the table, performance deteriorates as noise increases. In most cases, our proposed temporally regularized multimode clustering outperforms the other two. Generally speaking, regularized clustering outperforms online clustering, and online clustering outperforms static clustering. We notice that when the noise is high (e.g., noise level is 0.55), the smoothness of community structure between consecutive timestamps is destroyed. Thus, temporally regularized clustering performs worse than static clustering.



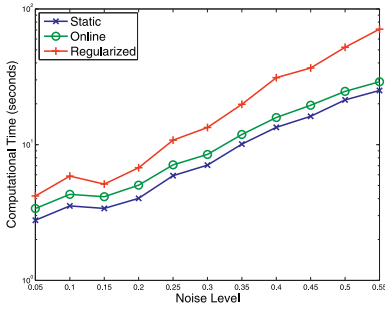


Fig. 6. Computation time of different methods.

We also plot the average computation time in Fig. 6. The computation time tends to increase with increasing noise. Static clustering takes the least time, and online clustering is comparable. Temporally regularized clustering requires more time. This is especially problematic when the noise level is very high. In that case, the temporal smoothness is disturbed. Many more iterations thus are required by our algorithm to find an optimal.

In order to show the effect of parameter tuning, we pick one data set of medium noise level and apply both online clustering and temporally regularized multimode clustering with temporal weights  $w_b$  varying from 0.01 to 1,000 and  $w_a$  fixed to 1. As shown in Fig. 7, it might worsen the performance if the temporal weight is too large, i.e., when the temporal regularization dominates. Most of the time, it helps clustering to take into account the temporal information. This is evident when the temporal weight is within the range of 0.01 and 100.

## 6 EXPERIMENTS ON REAL-WORLD DATA

In this section, we compare different clustering schemes on real-world data sets. Two publicly available network data sets are selected: Enron email corpus<sup>3</sup> and DBLP data.<sup>4</sup> We show the construction of multimode networks from these two data sets, discuss the evaluation methodology without ground truth, and then present empirical comparisons with some case studies.

### 6.1 Data Sets

Enron data [30] provide email documents collected from 150 senior executives in the Enron company. Based on their email communications, a 3-mode network (user, email, words) is constructed in each month. We focus on users (i.e., email addresses) who send and receive at least five emails, resulting in 2,359 users in total. In addition, we remove top frequent words occurring in more than 19,000 documents, and those infrequent words with document frequency less than 10, resulting in a vocabulary of 36,658 words. Because of the data collection problem, email traffic is low in the first few years. Hence, we consider only the time period with reasonable number of email communications (April, 2001 to March 2002). In each month, there are three interactions: sender-email, email-receiver, and email-words. The weights of email-receiver interaction are normalized by the number of receivers. We

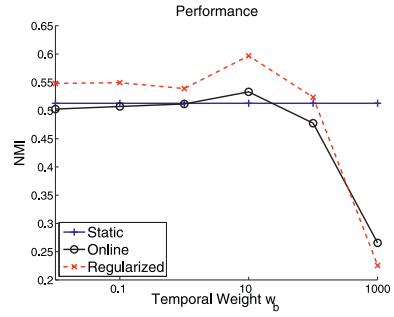


Fig. 7. Performance sensitivity to temporal weight.

notice that some words occur very frequently in one email. To avoid the dominance of these bursty words, we set the entries of email-word interaction to be  $\log(\text{word freq}) + 1$ .

As for DBLP data, four modes are considered: papers, authors, terms (words extracted from the title), and venues (conference or journal names). We extract papers published during 1980-2004, remove those papers without authors or venue information, discard stop words and terms with low document frequency (less than 20). Eventually, the data contains 491,726 papers, 347,013 authors, 2,826 venues, and 9,523 terms. Both processed data sets are publicly available at the first author's homepage.

### 6.2 Evaluation Methodology

Different from synthetic data, no ground truth of community membership is available for real-world data. NMI cannot be employed to evaluate performance. Alternatively, we present an approach to compare the relative performance of different methods based on the approximation error of interactions and a block structure. Below, we use Enron data as an example to show the evaluation methodology. A similar strategy can be derived for the DBLP data.

We follow standard cross validation to split the Enron data of each month into two portions. The first half is used for training, to obtain the clusters of users and words. The remaining half is used to measure how the cluster approximates the interaction. Notice that there is no overlap between training and test documents, so the clustering result of emails cannot be verified. Therefore, we focus only on the block approximation of words and users in the test data.

Given an interaction matrix  $R_{email,i}$  between email and another mode  $\mathbb{X}_i$  (sender, receiver, or words) for testing,  $R_{email,i}$  can be factored into the following form based on block model approximation:

$$R_{email,i} \approx C_{email} A C_i^T.$$

Here,  $C_i$  denotes the cluster membership of mode  $i$  obtained from training data. To measure the quality of  $C_i$ , one natural solution is to use the approximation error  $\|R_{email,i} - C_{email} A C_i^T\|_F$ . Note that in test data, both  $C_{email}$  and  $A$  are not known. We combine these two variables into one single variable  $B$  such that  $B = C_{email} A$ . Then we can approximate  $R_{email,i}$  in terms of  $C_i$  as follows:

$$\min_B \|R_{email,i} - B C_i^T\|_F^2.$$

3. <http://ciir.cs.umass.edu/%7Ecorrada/enron/>.

4. <http://kdl.cs.umass.edu/data/dblp/dblp-info.html>.

TABLE 3  
Performance on Enron Data Based  
on One Year Network Information

Month	Static	Online	Regularized
2001-04	3222.7	3248.6	<b>3313.1</b>
2001-05	2671.9	2660.3	<b>2719.6</b>
2001-06	1012.2	1040.8	<b>1084.9</b>
2001-07	1639.1	1640.1	<b>1656.2</b>
2001-08	1770.6	1738.6	<b>1797.9</b>
2001-09	1620.1	1680.6	<b>1702.2</b>
2001-10	<b>4449.3</b>	4422.3	4370.7
2001-11	<b>3057.2</b>	2926.1	2979.3
2001-12	1884.0	1904.0	<b>1920.4</b>
2002-01	<b>3205.9</b>	3067.4	3156.7
2002-02	969.4	968.9	<b>1041.3</b>
2002-03	801.2	814.5	<b>824.0</b>

Static in the first row represents clustering without any temporal information; Online an online version of temporally regularized clustering; and Regularized our proposed temporally regularized multimode clustering. The bold entry in each row represents the best performance of the three approaches.

It is not difficult to verify that the minimal value is  $\|R_{email,i}\|_F^2 - \|R_{email,i}C_i\|_F^2$ , obtained when  $B = RC_i$ . Since  $\|R_{email,i}\|_F^2$  is a constant, the cluster assignment that maximizes  $\|R_{email,i}C_i\|_F^2$  yields a low approximation error. Therefore, the clustering quality can be measured as follows:

$$\text{Clustering-Quality}(C_i) = \|R_{email,i}C_i\|_F^2.$$

A larger value indicates a better clustering result.

This evaluation strategy approximates a sparse interaction matrix with group interaction blocks. Even with the true latent community structure, the approximation error could still be huge. It is not appropriate to determine the significance of improvement by looking at the absolute value directly. This metric is adopted mainly to compare the *relative* performance of different methods.

### 6.3 Performance Comparison

For simplicity, all the input weights of the algorithm in Fig. 4 are set to 1. The number of communities of each mode is fixed to 50 and 20 for Enron and DBLP, respectively.

Table 3 summarizes the performance of different methods on Enron data. Online clustering, as shown in the table, yields a limited improvement. Most of the time, our proposed regularized clustering works the best. It is observed that in 10/2001, 11/2001, and 01/2002, static

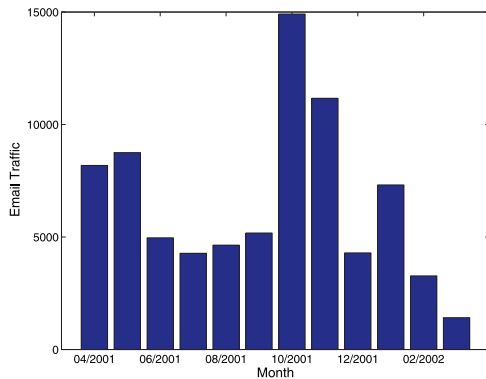


Fig. 8. Emails sent each month in Enron Data. In Oct. 2001, Nov. 2001, and Jan. 2002, heavier email traffic is observed.

TABLE 4  
Performance on DBLP Data ( $\times 10^4$ )

Year	Static	Online	Regularized
1980	0.1265	0.1279	<b>0.1371</b>
1981	0.1695	0.1754	<b>0.1787</b>
1982	0.1762	0.1733	<b>0.1839</b>
1983	0.1789	0.1851	<b>0.1893</b>
1984	0.2127	0.2167	<b>0.2261</b>
1985	0.2403	0.2291	<b>0.2514</b>
1986	0.2576	<b>0.2598</b>	0.2535
1987	0.2855	0.2969	<b>0.3075</b>
1988	0.3046	0.3224	<b>0.3238</b>
1989	0.3534	0.3425	<b>0.3596</b>
1990	0.4012	0.4034	<b>0.4242</b>
1991	0.4389	0.4381	<b>0.4395</b>
1992	0.4992	0.5071	<b>0.5187</b>
1993	0.5610	0.5759	<b>0.5887</b>
1994	0.6418	<b>0.6618</b>	0.6576
1995	0.6885	0.6658	<b>0.7102</b>
1996	0.7861	0.7944	<b>0.8110</b>
1997	0.9470	0.9546	<b>0.9644</b>
1998	1.0054	<b>1.0110</b>	1.0098
1999	1.0939	1.0893	<b>1.0967</b>
2000	1.2308	1.2337	<b>1.2549</b>
2001	1.5123	1.5003	<b>1.5319</b>
2002	1.7874	<b>1.7911</b>	1.7828
2003	2.2491	2.2649	<b>2.2697</b>
2004	2.7684	2.7717	<b>2.7807</b>

multimode clustering is the winner. This can be explained by email traffic as shown in Fig. 8. The email communication is abundant in the three months. Since sufficient data were provided, clustering without considering temporal information works well. Our proposed regularized multimode clustering, in a sense, compensates for the data shortage problem by taking into account the cluster structure at neighboring timestamps. When data are sufficient as in those three months, the temporal regularization becomes unnecessary.

As for DBLP data, a similar trend is observed as shown in Table 4. Temporally regularized clustering often outperforms other two again. Online clustering usually performs better than clustering without temporal regularization. Static clustering, without considering temporal regularization, tends to overfit the training data with a poor latent community structure. In general, taking into account the temporal smoothness helps extract a more accurate community structure.

### 6.4 Case Studies

In previous sections, we showed that temporally regularized clustering tends to outperform those methods without considering the temporal information. But one question remains: dose it discover meaningful communities and evolutions in reality? Here, we show some examples in DBLP data.

TABLE 5  
Venue Community Focusing on Web in 2004

Int. Conf. On Internet Computing, WWW, ICWE, ICWS, APWeb, Web Intelligence, IEEE SCC, J. Web Semantics International Semantic Web Conference,
-----------------------------------------------------------------------------------------------------------------------------------------------------------

TABLE 6  
Venue Community Associated with NIPS in 2004

NIPS, COLT, ECML, ICML, ICANN, IJCAI, IWANN, Machine Learning, Inter. Conference on Algorithmic Learning Theory, Int. J. Neural System, Neural Computation, Int. J. Computational Intelligence and Applications
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

NIPS, Neural Networks

Table 5 shows one example of venue community extracted based on our method. The publication venues in this community are mostly about Web. Table 6 shows another example of venue community associated with NIPS (a conference on Neural Information Processing System) in 2004. As seen in the table, most of those conferences are related to machine learning. However, we found the following group associated with NIPS in 1995:

Here, *Neural Networks* is a journal. NIPS at that moment focused more on neural network and neurobiology related problems. Thus, it is not surprising that NIPS and *Neural Network* are assigned to the same group. This indicates that our method discovered sensible communities and detected the evolution of groups over time.

Next, we check the groups in the author mode. Since (tens of) thousands of authors are involved each year in the multimode network, a community is typically composed of hundreds of authors. It is not convenient to present evolution patterns directly. Alternatively, we pick one target actor and show those actors who are often assigned into the same community with him across years. Specifically, after temporally regularized multimode clustering, each actor is associated with a community membership vector  $(c_a^1, c_a^2, \dots, c_a^T)$ , where  $c_a^t$  denotes community membership of actor  $a$  at timestamp  $t$ . Given a target actor  $a$  and another actor  $b$ , we compute the similarity between the two as follows:

$$\text{sim}(a, b) = \sum_{t=1}^T \delta(c_a^t, c_b^t),$$

with  $\delta$  function defined below:

$$\delta(x, y) = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{otherwise.} \end{cases}$$

The similarity between a target actor and all the remaining ones can be computed.

Three researchers, *Jiawei Han*, *Michael I. Jordan*, and *Subbarao Kambhampati* are selected as the target actor, respectively. Those top-ranking similar actors of each case are shown in Table 7. As seen in the table, most of the researchers in the first column concentrate on data mining or data base, which is consistent with *Jiawei Han*'s research work. For *Michael I. Jordan*, his similar actors are mostly working on machine learning (theory) and computational neurobiology. *Subbarao Kambhampati* is a prestigious researcher on planning. These years, his research also expands to data and information integration. It is not surprising that we find similar researchers from both fields for him. For example, *Qiang Yang*, *Biplav Srivastava*, and *Kathryn B. Laskey* are working on planning related problems, while the remaining ones are mostly working on data integration, database systems, and information retrieval. It is observed that most of those selected actors are actually not coauthors of the target actor, but those who work in similar areas and publish in related conferences and journals. This is because we consider multimode information including publication venues and title terms to help discover groups.

## 7 RELATED WORK

### 7.1 Community Detection

An extensive body of work studies the structural property of interactions between actors. One probabilistic approach is the stochastic block model [19], in which the link between actors is generated conditioned on the latent cluster membership of actors. Two actors within the same cluster are *stochastically equivalent*. That is, the interactions between  $(A_1, B_1)$  and  $(A_2, B_2)$  have the same probability if  $A_1$  and  $A_2$ ,  $B_1$  and  $B_2$  belong to the same cluster, respectively. In classic block models, the number of clusters is fixed. The work in [31] replaces the constraint by assigning a Chinese restaurant process as a prior to generate cluster membership for each actor, thus the number of clusters can be automatically determined by assigning a proper prior. Mixed block model is also developed [32]. Long et al. [12] propose a similar probabilistic framework to handle multimode networks with interactions and actor attributes. Topic models [33] are also extended to model documents within a

TABLE 7  
Researchers Working on Similar Areas across Years

Target Actor	Jiawei Han	Michael I. Jordan	Subbarao Kambhampati
Similar Actors	Christos Faloutsos	Thomas G. Dietterich	Qiang Yang
	Shusaku Tsumoto	Andrew G. Barto	Biplav Srivastava
	Raymond T. Ng	Peter Dayan	Kathryn B. Laskey
	Fosca Giannotti	Terrence J. Sejnowski	Craig A. Knoblock
	Ning Zhong	Shai Ben-David	Alfonso Gerevini
	Howard J. Hailton	H. Altay Guvenir	Gianni Amati
	Elisa Bertino	Andrew Y. Ng	Justin Zobel
	Reda Alhajj	Nader H. Bshouty	Andrei Z. Broder
	Philip S. Yu	Sally A. Goldman	Rajeev Motwani
	Ming-Syan Chen	Vladimir Vapnik	Amit P. Sheth
	Ke Wang	Mitsuo Kawato	Hongjun Lu
	Dominique Laurent	John Case	Kian-Lee Tan

social network [34], [35], [36]. Typically, they are specific for a certain type of documents like Emails or papers.

Another attempt to model the structure is the latent space model. Intuitively, latent space models map actors to a latent low-dimensional space such that the actors whose positions are closer are more likely to interact with each other [37], [38]. Globerson et al. [39] study a two-mode network (authors and words) and map both authors and words into the same euclidean space.

Spectral relational clustering, relevant to multimode networks, tries to discover latent community structures based on multiple relational tables. As the original problem of finding discrete cluster assignment (e.g., the entries of membership vector are either 0 or 1) is NP-hard, spectral clustering relaxes the constraint to allow the membership vector to be continuous. The initial work of coclustering [40], [20], [17], tries to address the problem of clustering both words and documents simultaneously by taking advantage of the structure of a bipartite. Gao et al. [41] extend the problem to a star-typed network with multiple heterogeneous objects, and propose semidefinite programming to solve the problem. Alternatively, reinforcement clustering is proposed [10]. Long et al. [11] present a general spectral clustering framework to handle multi-type relational clustering with different kinds of objects and attributes, and an alternating optimization algorithm is presented to find a solution.

## 7.2 Community Evolution

Temporal change of social networks has been attracting increasing attention [13], [42], [43]. It is empirically observed that some real-world networks are evolving [14]. Practitioners try to investigate how a network evolves and what might be a reasonable generative process to model the dynamics [16]. The critical factors to determine the group evolution are also reported [15].

On the other hand, evolutionary clustering [44] is developed. It assumes clustering result of current situation is similar to that of previous timestamps. Given multiple snapshots of network data, evolutionary clustering finds out a sequence of clustering with temporal smoothness [45], [46]. A Bayesian interpretation is presented in [47]. The latent space model with temporal change is also developed [48]. Tantipathananandh et al. [49] propose a general framework to handle dynamic single-mode network by casting it as a graph coloring problem and some greedy heuristics or approximation algorithms [50] are developed to handle large-scale data. Sun et al. [51] address the group evolution problem from an information-theoretic view. They present a scheme that detects not only the community structure but also the change point. As existing evolutionary approaches often require a specified number of clusters at each time stamp, (hierarchical) Dirichlet process is employed so that this parameter can be learned automatically from data [52], [53]. Most of the aforementioned works focus on data with attributes or single-mode networks. In this work, we explore the community evolution in multimode networks.

## 8 CONCLUSIONS AND FUTURE WORK

In some real-world applications, actors of various types interact with each other, resulting in a multimode network. In such a network, different modes of actors form their own communities and tend to evolve gradually. In this work, we propose a temporally regularized multimode clustering algorithm to identify evolving groups in dynamic multimode networks. We show that this algorithm can be interpreted as an iterative-LSA process, with attributes being community indicators at different modes and neighboring timestamps. Based on this attribute view, it is easy to extend this algorithm to handle networks with attributes, within-mode interactions, and dormant and emerging actors. In addition, we show that this algorithm is guaranteed to converge to a local optimal solution. We applied our framework to both synthetic data sets and real-world multimode networks of large sizes. It is empirically demonstrated that our algorithm tends to find more accurate community structures and evolutions given a sequence of network snapshots.

This work is a solid yet initial attempt to identify evolving groups in dynamic multimode networks. With the current framework, we envision several directions to explore for future work.

- The current framework needs users to provide weights for different interactions and temporal information, as well as the number of communities in each mode. It is challenging to automatically determine the weights and number of communities. Another possible extension is to consider the evolution of group interaction. Currently, our model considers only the group membership change. But interactions between groups may also change gradually. It requires further research to simultaneously capture both the microlevel evolution of actors and macrolevel evolution of groups.
- With large-scale networks, the scalability of one proposed method becomes a critical issue. Some statistical patterns presented in complex networks such as power law distribution for node degrees can be employed to address the scalability. For instance, Java et al. [54] propose to sort nodes based on their degrees and perform matrix factorization to a small subset of nodes with high degrees (as only a small portion of nodes are highly connected). Based on the factorization, the group membership of remaining nodes with few degrees can be identified via Nystrom's method [55] or certain heuristics. We believe such a scheme can be applied to handle large-scale multimode networks as well.
- Most existing clustering algorithms assign an entity to a single group. In reality, actors can be involved in multiple different communities. A viable solution to handle the overlapping nature of communities is edge-view community detection [56]. In the work, a community is defined as a set of edges rather than nodes as in most existing work. One node is associated with one community if any of its connections is inside the community. By partitioning

edges into disjoint sets, one node can be associated with multiple different communities. We are currently working on extending this edge-view community detection from single-mode networks to multimode networks.

- Our proposed temporally regularized multimode clustering algorithm smooths clustering results of neighboring timestamps. This is a reasonable regularization when networks are evolving slowly or gradually. But this smoothing effect might miss or delay the detection of an unprecedented event or major accident which could lead to sharp changes. It is quite interesting to demarcate those timestamps indicative of drastic changes. Moreover, in reality, a network of a history of continuous interactions is the norm. It presents a new challenge to separate these continuous interactions into multiple snapshots. One possible solution is to cluster neighboring network interactions over time such that the group structure within the same time cluster is kept relatively stable or smooth. We are currently investigating methods along this direction and plan to apply them to dynamic multimode networks in social media sites to study user grouping and behavior patterns.

## ACKNOWLEDGMENTS

The work is, in part, supported by a MITRE research grant and AFOSR Research Grant FA95500810132. We acknowledge Dr. Jieping Ye's inspiring comments. A preliminary version of this work is published in [1].

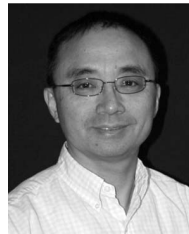
## REFERENCES

- [1] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community Evolution in Dynamic Multi-Mode Networks," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '08)*, pp. 677-685, 2008.
- [2] M. Newman, "The Structure and Function of Complex Networks," *SIAM Rev.*, vol. 45, pp. 167-256, 2003.
- [3] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge Univ. Press, 1994.
- [4] D. Chakrabarti and C. Faloutsos, "Graph Mining: Laws, Generators, and Algorithms," *ACM Computing Surveys*, vol. 38, no. 1, p. 2, 2006.
- [5] M.N. Lauren Ancel Meyers and B. Pourbohloul, "Predicting Epidemics on Directed Contact Networks," *J. Theoretical Biology*, vol. 240, no. 3, pp. 400-418, June 2006.
- [6] J. Baumes, M. Goldberg, M. Magdon-Ismael, and W. Wallace, "Discovering Hidden Groups in Communication Networks," *Proc. Second NSF/NIJ Symp. Intelligence and Security Informatics*, 2004.
- [7] J. Palau, M. Montaner, and B. Lopez, "Collaboration Analysis in Recommender Systems Using Social Networks," *Proc. Eighth Int'l Workshop Cooperative Information Agents (CIA '04)*, 2004.
- [8] L. Tang and H. Liu, "Relational Learning via Latent Social Dimensions," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '09)*, pp. 817-826, 2009.
- [9] L. Tang and H. Liu, "Toward Collective Behavior Prediction via Social Dimension Extraction," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 19-25, July-Aug. 2010.
- [10] J. Wang, H. Zeng, Z. Chen, H. Lu, L. Tao, and W.-Y. Ma, "Recom: Reinforcement Clustering of Multi-Type Interrelated Data Objects," *Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Informaion Retrieval (SIGIR '03)*, pp. 274-281, 2003.
- [11] B. Long, Z.M. Zhang, X. Wu, and P.S. Yu, "Spectral Clustering for Multi-Type Relational Data," *Proc. 23rd Int'l Conf. Machine Learning (ICML '06)*, pp. 585-592, 2006.
- [12] B. Long, Z.M. Zhang, and P.S. Yu, "A Probabilistic Framework for Relational Clustering," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 470-479, 2007.
- [13] *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, R. Breiger, K. Carley, and P. Pattison, eds. Nat'l Academies Press, 2003.
- [14] G. Kossinets and D.J. Watts, "Empirical Analysis of an Evolving Social Network," *Science*, vol. 311, no. 5757, pp. 88-90, 2006.
- [15] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group Formation in Large Social Networks: Membership, Growth, and Evolution," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06)*, pp. 44-54, 2006.
- [16] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining (KDD '05)*, pp. 177-187, 2005.
- [17] B. Long, Z.M. Zhang, and P.S. Yu, "Co-Clustering by Block Value Decomposition," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining (KDD '05)*, pp. 635-640, 2005.
- [18] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal Nonnegative Matrix t-Factorizations for Clustering," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06)*, pp. 126-135, 2006.
- [19] K. Nowicki and T.A.B. Snijders, "Estimation and Prediction for Stochastic Blockstructures," *J. Am. Statistical Assoc.*, vol. 96, no. 455, pp. 1077-1087, 2001.
- [20] H. Zha, X. He, C.H.Q. Ding, M. Gu, and H.D. Simon, "Spectral Relaxation for k-Means Clustering," *Proc. Neural Information Processing Systems (NIPS)*, pp. 1057-1064, 2001.
- [21] R. Bhatia, *Matrix Analysis*. Springer, 1997.
- [22] F.R. Bach and M.I. Jordan, "Learning Spectral Clustering," *Proc. Neural Information Processing Systems (NIPS)*, 2004.
- [23] M.W. Berry, S.T. Dumais, and G.W. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval," *SIAM Rev.*, vol. 37, no. 4, pp. 573-595, 1995.
- [24] M.J.D. Powell, "On Search Directions for Minimization Algorithms," *Math. Programming*, vol. 4, no. 1, pp. 193-201, Dec. 1973.
- [25] P. Tseng, "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization," *J. Optimization Theory and Applications*, vol. 109, no. 3, pp. 475-494, 2001.
- [26] J.C. Bezdek and R.J. Hathaway, "Convergence of Alternating Optimization," *Neural, Parallel and Scientific Computations*, vol. 11, no. 4, pp. 351-368, 2003.
- [27] A. Edelman, T.A. Arias, and S.T. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM J. Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303-353, 1999.
- [28] M.T. Chu and N.T. Trendafilov, "The Orthogonally Constrained Regression Revisited," *J. Computational and Graphical Statistics*, vol. 10, no. 4, pp. 746-771, Dec. 2001.
- [29] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," *J. Machine Learning Research*, vol. 3, pp. 583-617, 2003.
- [30] B. Klimat and Y. Yang, "The Enron Corpus: A New Dataset for Email Classification Research," *Proc. European Conf. Machine Learning*, 2004.
- [31] C. Kemp, T.L. Griffiths, and J.B. Tenenbaum, "Discovering Latent Classes in Relational Data," technical report, Massachusetts Inst. of Technology, 2004.
- [32] E. Airodi, D. Blei, S. Fienberg, and E.P. Xing, "Mixed Membership Stochastic Blockmodels," *J. Machine Learning Research*, vol. 9, pp. 1981-2014, 2008.
- [33] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, pp. 993-1022, 2003.
- [34] D. Zhou, E. Manavoglu, J. Li, C.L. Giles, and H. Zha, "Probabilistic Models for Discovering e-Communities," *Proc. 15th Int'l Conf. World Wide Web (WWW '06)*, pp. 173-182, 2006.
- [35] X. Wang, N. Mohanty, and A. McCallum, "Group and Topic Discovery from Relations and Their Attributes," *Proc. Advances in Neural Information Processing Systems 18*, pp. 1449-1456, 2006.
- [36] A. McCallum, X. Wang, and A. Corrada-Emmanuel, "Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email," *J. Artificial Intelligence Research*, vol. 30, pp. 249-272, 2007.
- [37] P.D. Hoff, M.S.H. Adrian, and E. Raftery, "Latent Space Approaches to Social Network Analysis," *J. Am. Statistical Assoc.*, vol. 97, no. 460, pp. 1090-1098, 2002.

- [38] M.S. Handcock, A.E. Raftery, and J.M. Tantrum, "Model-Based Clustering for Social Networks," *J. Royal Statistical Soc.: Series A*, vol. 170, no. 2, pp. 301-354, 2007.
- [39] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean Embedding of Co-Occurrence Data," *J. Machine Learning Research*, vol. 8, pp. 2265-2295, 2007.
- [40] I.S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," *Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '01)* pp. 269-274, 2001.
- [41] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent Bipartite Graph Co-Partitioning for Star-Structured High-Order Heterogeneous Data Co-Clustering," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining (KDD '05)*, pp. 41-50, 2005.
- [42] G. Palla, A.-L. Barabasi, and T. Vicsek, "Quantifying Social Group Evolution," *Nature*, vol. 446, no. 7136, pp. 664-667, Apr. 2007.
- [43] S. Asur, S. Parthasarathy, and D. Ucar, "An Event-Based Framework for Characterizing the Evolutionary Behavior of Interaction Graphs," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 913-921, 2007.
- [44] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary Clustering," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06)*, pp. 554-560, 2006.
- [45] Y. Chi, X. Song, D. Zhou, K. Hino, and B.L. Tseng, "Evolutionary Spectral Clustering by Incorporating Temporal Smoothness," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 153-162, 2007.
- [46] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B.L. Tseng, "Facetnet: A Framework for Analyzing Communities and Their Evolutions in Dynamic Networks," *Proc. 17th Int'l Conf. World Wide Web (WWW '08)*, pp. 685-694, 2008.
- [47] T. Yang, Y. Chi, S. Zhu, Y. Gao, and R. Jin, "A Bayesian Approach Toward Finding Communities and Their Evolutions in Dynamic Social Networks," *Proc. SIAM Int'l Conf. Data Mining*, 2009.
- [48] P. Sarkar and A.W. Moore, "Dynamic Social Network Analysis Using Latent Space Models," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 31-40, 2005.
- [49] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe, "A Framework for Community Identification in Dynamic Social Networks," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 717-726, 2007.
- [50] C. Tantipathananandh and T. Berger-Wolf, "Constant-Factor Approximation Algorithms for Identifying Dynamic Communities," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '09)*, pp. 827-836, 2009.
- [51] J. Sun, C. Faloutsos, S. Papadimitriou, and P.S. Yu, "Graphscope: Parameter-Free Mining of Large Time-Evolving Graphs," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '07)*, pp. 687-696, 2007.
- [52] T. Xu, Z.M. Zhang, P.S. Yu, and B. Long, "Evolutionary Clustering by Hierarchical Dirichlet Process with Hidden Markov State," *Proc. IEEE Eighth Int'l Conf. Data Mining (ICDM '08)*, pp. 658-667, 2008.
- [53] T. Xu, Z.M. Zhang, P.S. Yu, and B. Long, "Dirichlet Process Based Evolutionary Clustering," *Proc. IEEE Eighth Int'l Conf. Data Mining (ICDM '08)* pp. 648-657, 2008.
- [54] A. Java, A. Joshi, and T. Finin, "Approximating the Community Structure of the Long Tail," *Proc. Second Int'l Conf. Weblogs and Social Media (ICWSM '08)*, Mar. 2008.
- [55] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping Using the Nyström Method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214-225, Feb. 2004.
- [56] L. Tang and H. Liu, "Scalable Learning of Collective Behavior Based on Sparse Social Dimensions," *Proc. 18th ACM Conf. Information and Knowledge Management (CIKM '09)*, pp. 1107-1116, 2009.



**Lei Tang** received the PhD degree in computer science and engineering from Arizona State University (ASU). His research interests include social computing and data mining—in particular, relational learning with heterogeneous networks, group evolution, profiling and influence modeling, and collective behavior modeling and prediction in social media. He was awarded ASU GPSA Research Grant, SDM Doctoral Student Forum Fellowship, Student Travel Awards and Scholarships in various conferences. His tutorial about *Community Detection and Behavior Study for Social Computing* was featured at IEEE International Conference on Social Computing, 2009. His book on *Community Detection and Mining in Social Media* is scheduled to publish in 2010. He is a member of the ACM, IEEE, and AIS.



**Huan Liu** received the PhD degree from University of Southern California and bachelor of engineering from Shanghai Jiao Tong University. He worked at Telecom Research Labs in Australia, and taught at National University of Singapore before he joined Arizona State University (ASU) in Year 2000. He is a professor of computer science and engineering at ASU. He has been recognized for excellence in teaching and research in computer science and engineering at Arizona State University. His research interests include data/web mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world applications with high-dimensional data of disparate forms such as social media, modeling group interaction, text categorization, bioinformatics, and text/web mining. His research has been sponsored by US NSF, NASA, AFOSR, and ONR, among others. His well-cited publications include books, book chapters, encyclopedia entries as well as conference and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a founding organizer of the International Conference Series on Social Computing, Behavioral Modeling, and Prediction (<http://sbp.asu.edu/>) in Phoenix, Arizona (SBP08 and SBP09). His professional memberships include AAAI, ACM, ASEE, and SIAM. He is a senior member of the IEEE. He can be contacted via <http://www.publi.asu.edu/~huanliu>.



**Jianping Zhang** received the PhD degree from University of Illinois at Urbana-Champaign in computer science in 1990. He is a principal AI scientist at the MITRE Corporation. His responsibilities include conducting R&D work in Machine Learning and Data Mining for US Government applications. Prior to joining MITRE, he was the chief architect in AOL from 2004 to 2006, where he led efforts in developing web content categorization service, web filtering services, and web user profiling platform. From 2000 to 2004, he was a lead AI scientist in MITRE, where he worked on various data mining projects for different US Government agencies. From 1990 to 2000, he was a assistant/association professor in Utah State University.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).