Identifying Evolving Groups in Dynamic Multi-Mode Networks

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

Abstract—A multi-mode network consists of heterogeneous types of actors with various interactions occurring between them. Identifying communities in a multi-mode 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 multi-mode 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 multi-mode 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, Multi-Mode Networks, Dynamic Networks

I. 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] is 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) are 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 *multi-mode network* [3] (a.k.a. *heterogeneous network*).
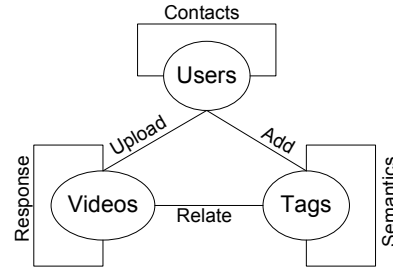
Fig. 1. An example of 3-Mode Network in YouTube

A. Multi-Mode Networks

Take the network in YouTube as an example. A 3-mode network (shown in Figure 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 multi-mode network is the field of academia as shown in Figure 2. Assorted entities (researchers, conference/journals, papers, words) are intertwined with each other. 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 multi-mode network, different types of entities tend to form groups or communities[1]. 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

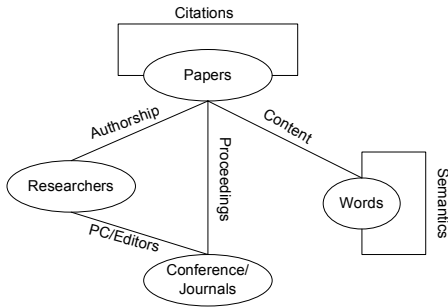[1]Group and community are interchangeable in this work.

Fig. 2.　A Multi-Mode Network of Academic Publications

interact with multiple groups of another mode, i.e., users might have multiple interests, thus relating to various video or tag groups.

Multi-mode networks present entities in interrelated forms. One might wonder whether it is necessary to consider this multi-mode property. For instance, in the 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 *multi-type 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.

B. Group Evolution

In a multi-mode 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 multi-mode networks because 1) the evolutions of different modes become correlated; and 2) disparate modes demonstrate distinctive evolution patterns. In this work, we propose a general model to identify evolving groups in dynamic multi-mode networks. A spectral clustering framework is adopted. A dynamic multi-mode 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 multi-mode 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 multi-mode networks in Section II, introduce an iterative solver in Section III and study its convergence property in Section IV. In Sections V and VI , 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 VII and conclude in Section VIII.

II. Problem Formulation

In order to extract communities in a multi-mode network, we follow the framework presented in [11]. Given an m-mode network with m types of actors $\mathbb{X}_1, \mathbb{X}_2, \cdots, \mathbb{X}_m$, 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 discovery and discuss incorporating temporal regularization for dynamic multi-mode networks. For simplicity, the discussion below does not consider within-mode interactions. Extensions to handle within-mode interactions are discussed later.

A. Community Discovery via Block Model Approximation

In order to extract communities at each mode, we assume interactions between modes can be approximated by
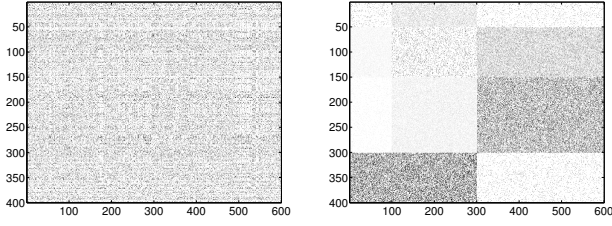
Fig. 3. Networks Before and After Reordering

TABLE I
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

interactions between groups [17], [18]. This basic idea can be visualized in Figure 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 multi-mode networks via matrix approximation:

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

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

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

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 [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 multi-mode network, diverse interactions occur between different modes. Hence, the objective in Eq. (1) can be changed to

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

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

B. Community Discovery with Network Sequence

With a dynamic multi-mode network, we have multiple snapshots of the network. Naturally, the objective function *without* considering its temporal effect can be formulated as $\mathbf{F_1}$:

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

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

Then we have the following theorem:

Theorem 1: Let $C^{(i,t)}, 1 \le i \le m, 1 \le t \le \mathbb{T}$ be a valid solution of $\mathbf{F_1}$, then $\widetilde{C}^{(i,t)}$ defined below is also a valid solution with the same objective value.

$$\widetilde{C}^{(i,t)} = C^{(i,t)} Q^{(i,t)}$$
$$s.t. \quad (Q^{(i,t)})^T Q^{(i,t)} = Q^{(i,t)} (Q^{(i,t)})^T = I_{k_i}$$
$$Q^{(i,t)} \in \mathbb{R}^{k_i \times k_i}$$

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 $\widetilde{A}_{i,j}^t = (Q^{(i,t)})^T A_{i,j}^t Q^{(j,t)}$, then

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

which completes the proof. ∎

C. 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 Ω 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 Eq. (8) does not necessarily capture the difference between the cluster indicators at different timestamps. On the contrary, the regularization term of Eq.(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:

$$\mathbf{F_2}: \min \sum_{t=1}^{\mathbb{T}} \sum_{i<j} w_a^{(i,j)} ||R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T||_F^2 +$$

$$\frac{1}{2} \sum_{i=1}^{m} w_b^{(i)} \sum_{t=2}^{\mathbb{T}} ||C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T||_F^2 \quad (9)$$

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

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.

III. TEMPORALLY-REGULARIZED MULTI-MODE CLUSTERING

To capture evolving groups in dynamic multi-mode 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.

A. 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 $A_{i,j}^t$.

$$||R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T||_F^2$$
$$= tr[(R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T)(R_{i,j}^t - C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T)^T]$$
$$= tr[R_{i,j}^t (R_{i,j}^t)^T - 2C^{(i,t)} A_{i,j}^t (C^{(j,t)})^T (R_{i,j}^t)^T + A_{i,j}^t (A_{i,j}^t)^T]$$

The last equation follows as $tr(AB) = tr(BA)$ and $C^{(i,t)}$ and $C^{(j,t)}$ are column orthogonal as in Eq. (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. ∎

B. Computation of C

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

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

Meanwhile,

$$\frac{1}{2}||C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T||_F^2$$
$$= \frac{1}{2}tr\left[C^{(i,t)}(C^{(i,t)})^T + C^{(i,t-1)}(C^{(i,t-1)})^T - 2C^{(i,t)}(C^{(i,t)})^T C^{(i,t-1)}(C^{(i,t-1)})^T\right]$$
$$= k_i - ||(C^{(i,t)})^T C^{(i,t-1)}||_F^2. \quad (12)$$

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:

$$\mathbf{F_3}: \max \sum_{t=1}^{\mathbb{T}} \sum_{l \le i < j \le m}^{m} w_a^{(i,j)} ||(C^{(i,t)})^T R_{i,j}^t C^{(j,t)}||_F^2$$
$$+ w_b^{(i)} \sum_{t=2}^{\mathbb{T}} \sum_{i=1}^{m} ||(C^{(i,t)})^T C^{(i,t-1)}||_F^2 \quad (13)$$

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 $C^{(i,t\pm1)}$ are given, which is stated in the following theorem.

Theorem 3: Given $C^{(j,t)}$ and $C^{(i,t\pm1)}$, $C^{(i,t)}$ can be computed as the top left singular vectors of the matrix P_i^t concatenated by the following matrices in column-wise:

$$P_i^t = \left[\left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \ne i}, \sqrt{w_b^{(i)}} C^{(i,t\pm1)} \right] \quad (14)$$

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

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

where M_i^t is defined as

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

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 $C^{(i,t\pm1)}$ 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 Multi-Mode Clustering

Fig. 5.   Algorithm in Attribute View: Iterative-LSA

So the problem boils down to a max-trace problem with orthogonality constraint as in Eq. (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:

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

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

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

$$
= \begin{bmatrix} \left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \sqrt{w_b^{(i)}} C^{(i,t\pm1)} \end{bmatrix}
$$

For $t = 1$ or $t = \mathbb{T}$, we only need to keep $C^{(i,t+1)}$ or $C^{(i,t-1)}$ respectively, instead of $C^{(i,t\pm1)}$. 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 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. ∎

C. Algorithm in Attribute View

To solve the problem **F**$_3$ in Eq. (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[2]. After convergence, $\{C^{(i,t)}\}$ are the approximate community indicator matrices. One problem

---

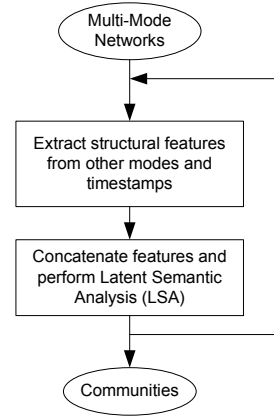[2]The convergence property will be discussed in detail in Section IV.

to be addressed is to recover the discrete partition of communities. One commonly used post-processing scheme is to apply k-means clustering to the community indicators [22]. Combined with the results of previous sections, we have the temporally-regularized multi-mode clustering algorithm as shown in Figure 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. According to Theorem 3, the community indicator C_i^t corresponds to the left singular vectors of P_i^t, which is defined as in Eq. (14). Consider P_i^t as normal instance-attribute matrix, then finding out the community indicator is essentially equivalent to perform latent semantic analysis (LSA) [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 (SVD) 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 Eq. (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 multi-mode networks to attribute-based data analysis as in conventional data mining. The key difference is that the "features" in multi-mode 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 Figure 5.

D. Extensions to Real-World Networks

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

Actor Attributes. The attribute view of the algorithm allows for the integration of entity attributes in a convenient way. When a multi-mode 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:

$$\widetilde{P_i^t} = \left[\left\{ \sqrt{w_a^{(i,j)}} R_{i,j}^t C^{(j,t)} \right\}_{j \neq i}, \sqrt{w_b^{(i)}} C^{(i,t\pm1)}, \sqrt{w_c^{(i)}} F_i^t \right]$$

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

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 ℓ-th order temporal regularization enforces the community structure at timestamp t to be similar to that of $t-1$, $t-2$, $\cdots$, $t-\ell$. Based on the attribute view of the algorithm, we just need to modify P_i^t as follows to reflect the higher-order dependency:

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

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.

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 (Figure 1) and those in the paper citations in the example of academic publications (Figure 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.

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 $C^{(i,t\pm1)}$ as attributes for current timestamp. When an actor leaves or

hibernates at certain timestamp, we can delete his corresponding entry in $C^{(i,t\pm1)}$ 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 $C^{(i,t\pm1)}$ is relatively small, thus playing a less important role in the computation of the similarity matrix M_i^t in Eq. (16). This is reasonable. A drastic membership change denotes the latent community is experiencing an "overhaul", thus it is not necessary to over-regularize the temporal change. When mode $\mathbb{X}_i$ at timestamp t has new actors, we simply set their entries in $C^{(i,t\pm1)}$ to 0.

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.

IV. CONVERGENCE ANALYSIS

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

Theorem 4: The objective of formulation $\mathbf{F_2}$ in (9) following the temporally-regularized multi-mode clustering algorithm in Figure 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 non-increasing. Since the objective of $\mathbf{F_2}$ has a lower bound 0, it must converge to a constant value. ∎

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 $k_i << n_i$, $\mathbf{F_3}$ 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 non-negative (non-positive) 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:

$$\mathbf{F_4}: \quad \max \quad \sum_{t=1}^{\mathbb{T}} \sum_{1 \leq i < j \leq m} w_a^{(i,j)} tr\left[Q^{(j,t)} R_{i,j}^T Q^{(i,t)} R_{i,j}\right]$$

$$+ w_b^{(i)} \sum_{t=2}^{\mathbb{T}} \sum_{i=1}^{m} tr\left[Q^{(i,t)} Q^{(i,t-1)}\right] \quad (18)$$

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

$$\mathcal{M}^{(i,t)} = \{S | S = C^{(i,t)}(C^{(i,t)})^T\} \quad (19)$$

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 $f(x_1, x_2, \cdots, x_N)$ 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, $k+N, k+2N, \cdots$, for $k = 1, \cdots, N$. 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 $\{x^r = (x_1^r, x_2^r, \cdots, x_N^r)\}_{r=0,1,\cdots}$ generated by block coordinate descent is defined and bounded. Moreover, if $f(x_1, \cdots, x_N)$ has at most one minimum in x_k for $k = 2, \cdots, N-1$, and the cyclic rule is used, then every cluster point z of $\{x^r\}_{r \equiv (N-1) \mod N}$ 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 Eq. (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 Eq. (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. ∎

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 Eq. (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 Eq. (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 Λ denotes a diagonal matrix of the top eigenvalues of M. Following Eq. (15), $\mathbf{F_3}$ in terms of C can be written as

$$\mathbf{F_3}(C) = tr\left[C^T M C\right] + const$$

with M defined in Eq. (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:

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

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

$$\pi_{\mathcal{T}}(g(C))$$

$$= C\frac{2C^T MC - 2C^T M^T C}{2} - 2(I - CC^T)MC$$

$$= -2(I - CC^T)MC \quad (23)$$

$$= -2(I - CC^T)C\Lambda \quad (24)$$

$$= 0$$

where Eq. (23) follows from the symmetry of M, and Eq. (24) follows from the fact that C corresponds to the top eigenvectors of M as stated in Eq. (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). ∎

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 multi-mode 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 $\mathbf{F}_3$ stabilizes.

V. Experiments on Synthetic Data

Three methods are compared: static, online, and our proposed temporally-regularized multi-mode 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 $\mathbf{F_1}$ in Eq. (5). For completeness, we also include the result of online clustering (as mentioned in Section III-D).

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.

A. Experiment Setup

We construct a synthetic network of 3 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 $(5\% - 20\%)$ 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.
- Noise is inserted at each timestamp. For example, given noise level 0.2, approximately 20% of entries in the interaction matrix are randomly selected to set to 0, and then another 20% 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.

TABLE II
AVERAGE NMI ON SYNTHETIC DATA OVER 100 RUNS. 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 MULTI-MODE CLUSTERING.

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

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 Eq. (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 π^a, π^b denote two different partitions of communities. NMI is defined as

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

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

B. Experiment Results

The average performance over 100 runs is reported in Table II. As seen in the table, performance deteriorates as noise increases. In most cases, our proposed temporally-regularized multi-mode 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.

We also plot the average computation time in Figure 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
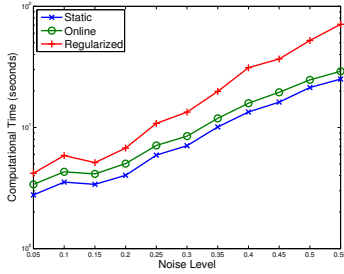
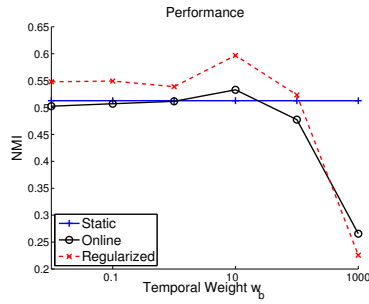Fig. 6.   Computation time of different methods

Fig. 7.   Performance sensitivity to temporal weight

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 multi-mode clustering with temporal weights w_b varying from 0.01 to 1000 and w_a fixed to 1. As shown in Figure 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.

VI. 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[3] and DBLP data[4]. We show the construction of multi-mode networks from these two data sets, discuss the evaluation methodology without ground truth, and then present empirical comparisons with some case studies.

A. Data Sets

Enron data [30] provides 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 5 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 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.

B. 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:

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

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 $B = RC_i$.

---

[3]http://ciir.cs.umass.edu/%7Ecorrada/enron/
[4]http://kdl.cs.umass.edu/data/dblp/dblp-info.html

TABLE III
PERFORMANCE ON ENRON DATA BASED ON ONE YEAR NETWORK
INFORMATION. *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 MULTI-MODE CLUSTERING.
THE BOLD ENTRY IN EACH ROW REPRESENTS THE BEST
PERFORMANCE OF THE THREE APPROACHES.

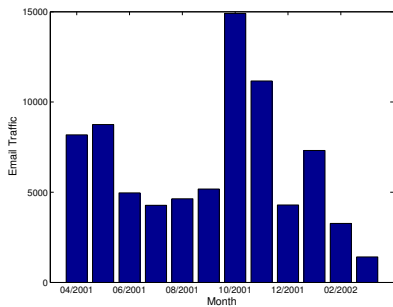| Month	Static	Online	Regularized
2001-04	3222.7	3248.6	**3313.1**
2001-05	2671.9	2660.3	**2719.6**
2001-06	1012.2	1040.8	**1084.9**
2001-07	1639.1	1640.1	**1656.2**
2001-08	1770.6	1738.6	**1797.9**
2001-09	1620.1	1680.6	**1702.2**
2001-10	**4449.3**	4422.3	4370.7
2001-11	**3057.2**	2926.1	2979.3
2001-12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

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

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.

C. Performance Comparison

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

Table III 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 multi-mode clustering is the winner. This can be explained by email

TABLE IV
PERFORMANCE ON DBLP DATA ($\times 10^4$)

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

traffic as shown in Figure 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 multi-mode 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 IV. 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.

D. 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 V shows one example of venue community extracted based on our method. The publication venues in this community are mostly about Web. Table VI 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:

TABLE V
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 VI
VENUE COMMUNITY ASSOCIATED WITH NIPS IN 2004

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

NIPS, Neural Networks

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 multi-mode 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 multi-mode clustering, each actor is associated with a community membership vector $(c_a^1, c_a^2, \cdots, c_a^{\mathbb{T}})$, 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:

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

with δ 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 VII. 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 co-authors of the target actor, but those who work in similar areas and publish in related conferences and journals. This is because we consider multi-mode information including publication venues and title terms to help discover groups.

VII. RELATED WORK

A. 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 multi-mode networks with interactions and actor attributes. Topic models [33] are also extended to model documents within a 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 multi-mode 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 co-clustering [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 semi-definite 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.

B. Community Evolution

Temporal change of social networks has been attracting increasing attention [13], [42], [43]. It is empirically ob-

TABLE VII
RESEARCHERS WORKING ON SIMILAR AREAS ACROSS YEARS

Target Actor	Jiawei Han	Michael I. Jordan	Subbarao Kambhampati
	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
Similar Actors	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

served 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 multi-mode networks.

VIII. CONCLUSIONS AND FUTURE WORK

In some real-world applications, actors of various types interact with each other, resulting in a multi-mode 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 multi-mode clustering algorithm to identify evolving groups in dynamic multi-mode 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 multi-mode 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 multi-mode 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 micro-level evolution of actors and macro-level 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 multi-mode 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 multi-mode networks.

- Our proposed temporally-regularized multi-mode 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 multi-mode networks in social media sites to study user grouping and behavior patterns.

ACKNOWLEDGMENTS

REFERENCES

[1] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," in *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2008, pp. 677–685.

[2] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.

[3] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[4] D. Chakrabarti and C. Faloutsos, "Graph mining: Laws, generators, and algorithms," *ACM Comput. Surv.*, vol. 38, no. 1, p. 2, 2006.

[5] M. N. Lauren Ancel Meyers and B. Pourbohloul, "Predicting epidemics on directed contact networks," in *Journal of Theoretical Biology*, vol. 240, no. 3, June 2006, pp. 400–418.

[6] J. Baumes, M. Goldberg, M. Magdon-Ismail, and W. Wallace, "Discovering hidden groups in communication networks," in *2nd NSF/NIJ Symposium on intelligence and Security Informatics*, 2004.

[7] J. Palau, M. Montaner, and B. Lopez, "Collaboration analysis in recommender systems using social networks," in *In Eighth Intl. Workshop on Cooperative info. Agents (CIA'04)*, 2004.

[8] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2009, pp. 817–826.

[9] ——, "Toward collective behavior prediction via social dimension extraction," *IEEE Intelligent Systems*, 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," in *SIGIR*, 2003, pp. 274–281.

[11] B. Long, Z. M. Zhang, X. Wú, and P. S. Yu, "Spectral clustering for multi-type relational data," in *ICML '06: Proceedings of the 23rd international conference on Machine learning*. New York, NY, USA: ACM, 2006, pp. 585–592.

[12] B. Long, Z. M. Zhang, and P. S. Yu, "A probabilistic framework for relational clustering," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2007, pp. 470–479.

[13] R. Breiger, K. Carley, and P. Pattison, Eds., *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*. The National 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," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006, pp. 44–54.

[16] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD*, 2005, pp. 177–187.

[17] B. Long, Z. M. Zhang, and P. S. Yu, "Co-clustering by block value decomposition," in *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM, 2005, pp. 635–640.

[18] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *KDD*. New York, NY, USA: ACM, 2006, pp. 126–135.

[19] K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," *Journal of the American Statistical Association*, 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," in *NIPS*, 2001, pp. 1057–1064.

[21] R. Bhatia, *Matrix Analysis*. Springer, 1997.

[22] F. R. Bach and M. I. Jordan., "Learning spectral clustering," in *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," *Mathematical Programming*, vol. 4, no. 1, pp. 193–201, December 1973.

[25] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, 2001.

[26] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel Sci. Comput.*, 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 Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1999.

[28] M. T. Chu and N. T. Trendafilov, "The orthogonally constrained regression revisited," *Journal of 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. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.

[30] B. Klimat and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Proceedings of European Conference on Machine Learning*, 2004.

[31] C. Kemp, T. L. Griffiths, and J. B. Tenenbaum, "Discovering latent classes in relational data," Massachusetts Institute of Technology, Tech. Rep., 2004.

[32] E. Airodi, D. Blei, S. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, 2008.

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, 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," in *WWW*, 2006, pp. 173–182.

[35] X. Wang, N. Mohanty, and A. McCallum, "Group and topic discovery from relations and their attributes," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 1449–1456.

[36] A. McCallum, X. Wang, and A. Corrada-Emmanuel, "Topic and role discovery in social networks with experiments on enron and academic email," *Journal of Artificial Intelligence Research*, no. 0, pp. 249–272, 2007.

[37] P. D. Hoff and M. S. H. Adrian E. Raftery, "Latent space approaches to social network analysis," *Journal of the American Statistical Association*, 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," *Journal Of The Royal Statistical Society Series A*, vol. 127, no. 2, pp. 301–354, 2007.

[39] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *J. Mach. Learn. Res.*, vol. 8, pp. 2265–2295, 2007.

[40] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2001, pp. 269–274.

[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," in *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM, 2005, pp. 41–50.

[42] G. Palla, A.-L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, April 2007.

[43] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2007, pp. 913–921.

[44] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006, pp. 554–560.

[45] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, "Evolutionary spectral clustering by incorporating temporal smoothness," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2007, pp. 153–162.

[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," in *WWW*, 2008, pp. 685–694.

[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," in *SIAM International Conference on Data Mining*, 2009.

[48] P. Sarkar and A. W. Moore, "Dynamic social network analysis using latent space models," *SIGKDD Explor. Newsl.*, 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," in *KDD*. New York, NY, USA: ACM, 2007, pp. 717–726.

[50] C. Tantipathananandh and T. Berger-Wolf, "Constant-factor approximation algorithms for identifying dynamic communities," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2009, pp. 827–836.

[51] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "Graphscope: parameter-free mining of large time-evolving graphs," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2007, pp. 687–696.

[52] T. Xu, Z. M. Zhang, P. S. Yu, and B. Long, "Evolutionary clustering by hierarchical dirichlet process with hidden markov state," in *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 658–667.

[53] ——, "Dirichlet process based evolutionary clustering," in *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 648–657.

[54] A. Java, A. Joshi, and T. Finin, "Approximating the Community Structure of the Long Tail," in *Proceedings of the Second International Conference on Weblogs and Social Media (ICWSM 2008)*, AAAI. AAAI Press, March 2008, poster Paper.

[55] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, 2004.

[56] L. Tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions," in *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2009, pp. 1107–1116.

Lei Tang is a PhD candidate in computer science and engineering at Arizona State University. His research interests are in 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 ACM, IEEE and AIS.
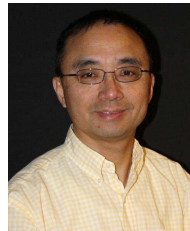
Huan Liu is a professor of Computer Science and Engineering at Arizona State University. He received his Ph.D. from University of Southern California and Bachelor of Engineering from Shanghai Jiao Tong University, worked at Telecom Research Labs in Australia, and taught at National University of Singapore before he joined ASU in Year 2000. Huan Liu has been recognized for excellence in teaching and research in Computer Science and Engineering at Arizona State University. His research interests are in 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 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, AZ (SBP08 and SBP09). His professional memberships include AAAI, ACM, ASEE, SIAM, and IEEE. He can be contacted via http://www.publi.asu.edu/~huanliu.

Jianping Zhang is 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 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, Dr. Zhang was a Lead AI Scientist in MITRE, where he worked on various data mining projects for different US Government agencies. From 1990 to 2000, Dr. Zhang was Assistant/Association Professor in Utah State University. Dr. Zhang obtained his Ph.D from University of Illinois at Urbana-Champaign in Computer Science in 1990.