Community Evolution in Dynamic Multi-Mode Networks

Lei Tang, Huan Liu
School of Computing Informatics
Arizona State University
Tempe, AZ 85287, USA
{L.Tang, Huan.Liu}@asu.edu

Jianping Zhang, Zohreh Nazeri
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA
{jzhang, nazeri}@mitre.org

ABSTRACT

A multi-mode network typically consists of multiple heterogeneous social actors among which various types of interactions could occur. 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 the membership of groups often evolve gradually. In a dynamic multi-mode network, both actor membership and interactions can evolve, which poses a challenging problem of identifying community evolution. In this work, we try to address this issue by employing the temporal information to analyze a multi-mode network. A spectral framework and its scalability issue are carefully studied. Experiments on both synthetic data and real-world large scale networks demonstrate the efficacy of our algorithm and suggest its generality in solving problems with complex relationships.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Sociology; H.3.3 [**Information Search and Retrieval**]: Clustering; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Experimentation, Performance

Keywords

Evolution, Multi-mode Network, Community Evolution, Dynamic Network Analysis, Dynamic Heterogeneous Network

1. INTRODUCTION

Social network analysis [35] is attracting increasing attention in many fields besides sociology, including epidemiology [20], intelligent analysis [5], targeted marketing and recommendation systems [28], etc. Most existing work concentrates on one-mode network. That is, there is only one type of social actors (nodes) involved in the network and the ties (interactions) between actors are of the same type. This is common in a broad sense such as friendship network, Internet, phone call network, etc. However, some emerging applications such as web mining, collaborative filtering, and online targeted marketing involve more than one type of actors and multiple heterogeneous interactions between different types of actors. Such a network is called *multi-mode network* (also known as heterogeneous network) [35].

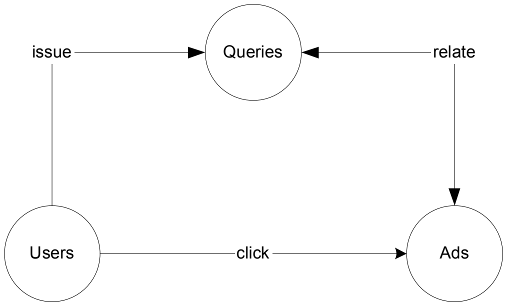


Figure 1: Online Marketing

Take targeted marketing as an example. A three-mode network can be constructed: users, queries, and online ads, as shown in Figure 1. Note that in the figure, both queries and ads are also considered as "social actors", though the user is still the major mode under consideration. There are different interactions among the three types of entities: Users issue queries when searching for his/her desired product; users can click on those ads that seem interesting to him/her; queries and ads are related to each other naturally by its semantic meanings. Intuitively, two users submitting similar queries or browsing related ads might share similar interest. Queries relating to ads of the same/similar product, and submitted by users within a single session are more likely to be related.

One way for one-mode networks to handle the example above is to construct a similarity network on user profiles or queries. Multi-mode network, by employing the relationship between different types of objects, could shed light for accurate online marketing with limited user information. The network might become more complicated if there are interactions between members of the same mode (say, friendship networks between users, similarity networks between queries) or attributes associated with certain objects (like the product information of the ads).

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

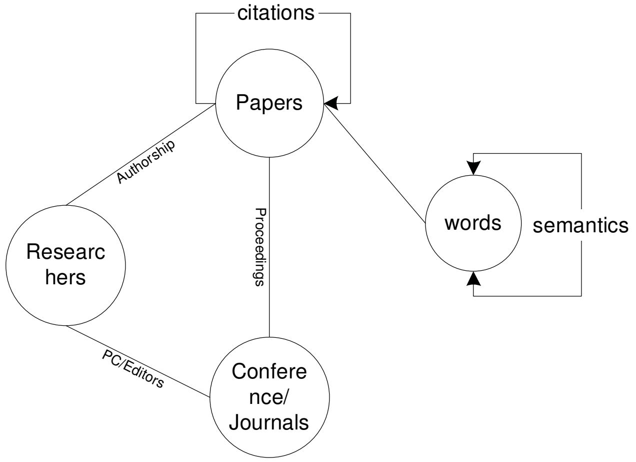


Figure 2: Academic Publications

Another example of multi-mode network is academic publications as shown in Figure 2. Various kinds of related objects (researchers, conference/journals, papers, words) are considered. Scientific literature connects papers by citations; and papers are published at different places (conferences, journals, workshops, thesis, etc.); Meanwhile, researchers are connected to papers through authorship. Some might also relate to each other by serving simultaneously as journal editors or the program committees of a conference. Moreover, each paper can focus on different topics, which are represented by words. Words are associated to each other based on semantics. At the same time, papers connect to different conferences, journals (venues for publication). In other words, multiple types of objects exist in the same network, and objects relate to others (either the same type or different types) through different links. Several objects might have some attributes as well.

Within a multi-mode network, different types of social actors tend to form groups or communities[1]. For instance, users sharing similar interest are more likely to form the same group; queries are clustered naturally if they are relating to similar products; and ads are clustered if they aim at the same/similar products. A special case is when there exists a one-to-one correspondence relationship between groups of different modes (user-query-ads), and the actors inside the the group interact with each other frequently. More generally, a user group could interact with multiple groups of another mode (i.e., users can have multiple interests, thus relating to various query groups). One mode network always assumes that the within-group interaction are dense and between-group interaction are sparse. Different from an one-mode network, a community inside a multi-mode network might have no within-group interaction yet share similar patterns while interacting with groups of other modes.

Some work has been done to identify communities in a network of heterogeneous objects or multiple relations [33, 23, 25] in terms of multi-type relational clustering. However, these methods concentrate on static networks. In reality, some networks tend to evolve gradually; the communities inside a network could grow or shrink, and the membership for specific social actors can shift gradually [8, 19, 4, 21]. In a multi-mode network, actors of different modes evolve differently. For instance, in previous academic example, the interactions between different actors change every year. Within the network, researchers can divert their personal research interests, and the "hot" topics of a field could also change. On the contrary, the community of "journals/conferences" tends to be stable, but two venues could also be connected due to the topic change. For instance, social network analysis joins researchers from diverse fields recently, thus making those journals on social science becoming more related to transactions concentrating on data mining.

Facing heterogeneous actors with dynamic interactions, discovering the community evolution can benefit in various aspects: 1) to detect the user interest shift leading to more effective targeted marketing; 2) to help browse a long history of network or literature by showing the long-term trend; 3) to detect suspicious financial online activities if the subject suddenly signals a drastic change in transaction pattern, etc. However, the problem becomes more challenging in dynamic multi-mode networks as the evolution of each mode becomes correlated.

In this work, we propose a general model to identify community evolution in dynamic multi-mode networks by adopting a spectral clustering framework. Essentially, the dynamic multi-mode network consists of a series of network snapshots. We aim to find out how a community evolves. In this model, the effect of temporal change is adopted as a regularization term while performing the clustering. It can be shown that the interactions between two modes, and clustering result of neighboring time stamps can be considered as attributes in a sense, which connects dynamic multi-mode network analysis to classical 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 for dynamic multi-mode networks in Section 2. An iterative solver is provided in Section 3. After that, we also discuss how to handle an actor dying or a new actor joining the network, and other multi-mode networks of diverse properties. In Section 5, extensive experiments on both synthetic and real-world data are conducted to study the efficacy and efficiency of our framework. We review some related literature in Section 6 and conclude in Section 7.

2. PROBLEM FORMULATION

We follow the framework presented in [23]. Given an m-mode network with m types of actors

$$\mathbb{X}_{(i)} = \{x_1^i, x_2^i, \cdots, x_{n_i}^i\} \quad i = 1, \cdots, m$$

where n_i is the number of actors for $\mathbb{X}_i$, we aim to find how the latent community evolves. In our framework, we only consider discrete time stamps by taking a series of snapshots, which is commonly adopted in network analysis with temporal information [4, 21]. For snapshot at time t, a network N^t is represented as multiple interactions between different modes. Let $R^t_{i,j} \in \mathbb{R}^{n_i \times n_j}$ denotes the interaction between two modes of actors $\mathbb{X}_i$ and $\mathbb{X}_j$ at time stamp t, k_i and k_j denote the number of latent communities for $\mathbb{X}_i$ and $\mathbb{X}_j$ respectively. Ideally, the interaction between actors can be

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

approximated by the interactions between groups in the following form [24, 12]:

$$R^t_{i,j} \approx C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T$$

where $C^{(i,t)} \in \{0,1\}^{n_i \times k_i}$ denotes latent cluster membership for $\mathbb{X}_i$ at time stamp t, and $A_{i,j}$ represents the group interaction. In other words, the group identity determines how two actors interact, which is essentially making similar assumption as block models [27]. The difference is that block models deal with the problem from a probabilistic aspect and concentrate on one-mode or two-mode networks. Here we try to identify the block structure of multi-mode networks via matrix approximation:

$$\min \quad ||R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T||^2_F \tag{1}$$

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

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

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.

Naturally, for dynamic multi-mode networks, the objective function *without* considering its temporal effect can be formulated as

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

where $w_a^{(i,j)}$ are the weights associated with different interactions. Unfortunately, the discreteness of the constraints in (2) and (3) makes the problem NP-hard. A strategy that has been well studied in spectral clustering [37] is to allow the cluster indicator matrix to be continuous and relax the hard clustering constraint as follows:

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

Hence, the problem becomes

$$\begin{aligned} \mathbf{F_1}: \quad & \min \quad \sum_{t=1}^{\mathbb{T}} \sum_{1 \le i < j \le m} w_a^{(i,j)} ||R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T||^2_F \\ & s.t. \quad (C^{(i,t)})^T C^{(i,t)} = I_{k_i} \qquad (4) \\ & with \quad i = 1, \cdots, m, \quad t = 1, \cdots \mathbb{T} \end{aligned}$$

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 $\widetilde{C}^{(i,t)}$ as defined below*

$$\begin{aligned} & \widetilde{C}^{(i,t)} = C^{(i,t)} Q^{(i,t)} \\ where \quad & (Q^{(i,t)})^T Q^{(i,t)} = Q^{(i,t)} (Q^{(i,t)})^T = I_{k_i}, \\ & Q^{(i,t)} \in I\!\!R^{k_i \times k_i}, \end{aligned}$$

is also a valid solution with the same objective value.

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

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

which completes the proof. □

The formulation $\mathbf{F_1}$ does not consider the relationship between consecutive time stamps. But in reality, the clustering tends to evolve gradually. To get a smooth community evolution, we add a regularization term Ω which forces the clustering sequence to be as smooth as possible:

$$\Omega = \frac{1}{2} \sum_{t=2}^{\mathbb{T}} ||C^{(i,t)} (C^{(i,t)})^T - C^{(i,t-1)} (C^{(i,t-1)})^T||^2_F \tag{5}$$

Here, the coefficient $\frac{1}{2}$ is presented due to notational conveniences in later parts. Indeed, we are making a Markov assumption using this regularization. That is, current cluster should be similar to the clustering results in previous time stamp.

Note that we do not take the regularization as

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

which seems more natural at first glimpse. Since $C^{(i,t)}$ is essentially equivalent under an orthogonal transformation as demonstrated in Theorem 1, comparing them directly does not necessarily capture the difference between the cluster indicator at different time stamps. On the contrary, the regularization term (5) captures the difference of community structure of neighboring time stamps, and it is independent of the orthogonal transformation. With this regularization, the problem can be formulated as:

$$\begin{aligned} \mathbf{F_2}: \quad & \min \quad \sum_{t=1}^{\mathbb{T}} \sum_{i<j} w_a^{(i,j)} ||R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T||^2_F \\ & + \frac{1}{2} \sum_{i=1}^{m} w_b^{(i)} \sum_{t=2}^{\mathbb{T}} ||C^{(i,t)} (C^{(i,t)})^T - C^{(i,t-1)} (C^{(i,t-1)})^T||^2_F \qquad (6) \\ & s.t. \quad (C^{(i,t)})^T C^{(i,t)} = I_{k_i} \\ & with \quad i = 1, \cdots, m, \quad t = 1, \cdots \mathbb{T} \end{aligned}$$

with $w_b^{(i)}$ being the trade-off between the loss of interaction approximation and temporal regularization. The evolution is assumed to take effect gradually. Essentially, we aim to find a community structure which is consistent with the interaction matrix, whereas not too different from that of the previous time stamp.

3. EVOLUTIONARY CLUSTERING

Here, we give an effective iterative algorithm to solve the formation $\mathbf{F_2}$ in (6).

3.1 Computation of A

Theorem 2. *If $C^{(i,t)}$ are given, the optimal group interaction matrix $A^t_{i,j}$ can be calculated as*

$$A^t_{i,j} \quad = \quad (C^{(i,t)})^T R^t_{i,j} C^{(j,t)}$$

Proof. Since $A^t_{i,j}$ appears only in a single term, we can focus on the term to optimize $A^t_{i,j}$.

$$\begin{aligned} & ||R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T||^2_F \\ = \quad & tr[(R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T)(R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T)^T] \\ = \quad & tr[R^t_{i,j} (R^t_{i,j})^T - 2C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T (R^t_{i,j})^T + A^t_{i,j} (A^t_{i,j})^T] \end{aligned}$$

The 2nd line is obtained via the property that $tr(AB) = tr(BA)$ and column orthogonality of $C^{(i,t)}$ and $C^{(j,t)}$.

By setting the derivative with respect to $A^t_{i,j}$ to zero, it follows that

$$A^t_{i,j} = (C^{(i,t)})^T R^t_{i,j} C^{(j,t)}$$

The proof is completed. □

3.2 Computation of C

Given the optimal $A^t_{i,j}$, it can be verified that

$$\begin{aligned} & ||R^t_{i,j} - C^{(i,t)} A^t_{i,j} (C^{(j,t)})^T||^2_F \\ = \;& ||R^t_{i,j}||^2_F - ||(C^{(i,t)})^T R^t_{i,j} C^{(j,t)}||^2_F \end{aligned} \tag{7}$$

Meanwhile, we have the temporal regularization term

$$\begin{aligned} & ||C^{(i,t)}(C^{(i,t)})^T - C^{(i,t-1)}(C^{(i,t-1)})^T||^2_F \\ = \;& tr\Big(C^{(i,t)}(C^{(i,t)})^T + C^{(i,t-1)}(C^{(i,t-1)})^T \\ & -2C^{(i,t)}(C^{(i,t)})^T C^{(i,t-1)}(C^{(i,t-1)})^T\Big) \\ = \;& 2k_i - 2||(C^{(i,t)})^T C^{(i,t-1)}||^2_F \end{aligned} \tag{8}$$

Since $||R^t_{i,j}||^2_F$ in (7) and k_i in (8) are constants, we can transform $\mathbf{F_2}$ into the following objective:

$$\begin{aligned} \mathbf{F_3}: \quad \max \; & \sum_{t=1}^{\mathbb{T}} \sum_{l \le i < j \le m}^{m} w_a^{(i,j)} ||(C^{(i,t)})^T R^t_{i,j} C^{(j,t)}||^2_F \\ & + w_b^{(i)} \sum_{t=2}^{\mathbb{T}} \sum_{i=1}^{m} ||(C^{(i,t)})^T C^{(i,t-1)}||^2_F \end{aligned} \tag{9}$$

Note that $C^{(i,t)}$ is interrelated with both $C^{(j,t)}$ and $C^{(i,t-1)}$, and there is no direct solver in general. However, it can be shown that given $C^{(j,t)}$ and $C^{(i,t\pm1)}$, the optimal $C^{(i,t)}$ can be solved as follows:

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^t_i concatenated by the following matrices in column-wise:*

$$\left[\left\{\sqrt{w_a^{(i,j)}} R^t_{i,j} C^{(j,t)}\right\}_{i<j}, \left\{\sqrt{w_a^{(k,i)}} (R^t_{k,i})^T C^{(k,t)}\right\}_{k<i}, \sqrt{w_b^{(i)}} C^{(i,t\pm1)}\right]$$

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

$$\begin{aligned} L \;=\; & \sum_{i<j} w_a^{(i,j)} ||(C^{(i,t)})^T R^t_{i,j} C^{(j,t)}||^2_F \\ & + \sum_{k<i} w_a^{(k,i)} ||(C^{(k,i)})^T R^t_{k,i} C^{(i,t)}||^2_F \\ & + w_b^{(i)} ||(C^{(i,t)})^T C^{(i,t-1)}||^2_F \\ & + w_b^{(i)} ||(C^{(i,t+1)})^T C^{(i,t)}||^2_F \\ =\; & tr\left((C^{(i,t)})^T M^t_i C^{(i,t)}\right) \end{aligned}$$

where M^t_i is defined as

$$\begin{aligned} M^t_i \;=\; & \sum_{i<j} w_a^{(i,j)} R^t_{i,j} C^{(j,t)} (C^{(j,t)})^T (R^t_{i,j})^T \\ & + \sum_{k<i} w_a^{(k,i)} (R^t_{k,i})^T C^{(k,t)} (C^{(k,t)})^T R^t_{k,i} \\ & + w_b^{(i)} C^{(i,t-1)} (C^{(i,t-1)})^T \\ & + w_b^{(i)} C^{(i,t+1)} (C^{(i,t+1)})^T \end{aligned} \tag{10}$$

```
Algorithm: Evolutionary Multi-mode Clustering
Input: $R$, $k_i$, $w_a^{(i,j)}$, $w_b^{(i)}$
Output: $idx^{(i,t)}$, $C^{(i,t)}$, $A^t_{i,j}$.
Generate initial cluster indicator matrix $C^{(i,t)}$.
Repeat
    For $t = 1 : \mathbb{T}$, $i = 1 : m$
        shrink / expand $C^{(i,t\pm1)}$ if necessary;
        calculate $P^t_i$ (or $M^t_i$) as in Theorem 3;
        calculate SVD of $P^t_i$ (or eigen vectors of $M^t_i$);
        update $C^{(i,t)}$ as top left singular (eigen) vectors;
Until the relative change of the objective ($\mathbf{F}3$) $\le \epsilon$.
calculate $A^t_{i,j}$ as in Theorem 2
calculate the cluster $idx^{(i,t)}$ with k-means on $C^{(i,t)}$
```

Figure 3: Algorithm Description

So the problem boils down to a max trace problem with orthogonality constraint as in (4). According to Ky-Fan theorem [6], this max-trace problem has a closed form solution, which corresponds to the subspace spanned by the top k_i eigen vectors of M^t_i.

Note that M_i is a square matrix of size $n_i \times n_i$. When the number of actors in $\mathbb{X}_i$ is huge, direct calculating M^t_i and its eigen vectors could be problematic. Instead, we can represent M^t_i in the following matrix form:

$$M^t_i = P^t_i * (P^t_i)^T \tag{11}$$

where P^t_i is concatenated by the following matrices in column-wise:

$$\left[\left\{\sqrt{w_a^{(i,j)}} R^t_{i,j} C^{(j,t)}\right\}_{i<j}, \left\{\sqrt{w_a^{(k,i)}} (R^t_{k,i})^T C^{(k,t)}\right\}_{k<i}, \sqrt{w_b^{(i)}} C^{(i,t\pm1)}\right] \tag{12}$$

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^t_i is much smaller compared to M^t_i if the number of clusters of each mode is small. Let the SVD of P_i as $P_i = U\Sigma V$, then $M_i = U\Sigma^2 U^T$. That is, the top left singular vectors $[u_1, \cdots, u_{k_i}]$ corresponds to the top eigen vectors of M_i, which completes the proof. Typically, we can use some methods like Arnoldi method [15] to calculate only its top left singular vectors, which could alleviate the computational burden for large-scale sparse networks. □

As can be seen in (12), the clustering result of neighboring time stamps and clustering of interacted objects, essentially form weighted features for clustering of the i-th mode. M^t_i, being the inner product of P^t_i, acts like a similarity matrix for clustering. Based on Theorem 3, we can update the cluster indicator matrix iteratively based on the "attributes" obtained from the clustering results of related objects and neighboring time stamps. Given the cluster indicator matrix, one problem remains to be addressed is how to recover the discrete hard cluster based on cluster indicator matrix. k-means has been shown to be an effective post-processing method [3]. The overall description of the algorithm is presented in Figure 3. In the algorithm, we specify the objective to be calculated via $\mathbf{F}3$, as direct calculation of the original formation $\mathbf{F}2$ usually requires computation of dense matrices, which is not applicable for large scale multi-mode networks.

4. EXTENSIONS

Based on the framework proposed in previous section, it is easy to extend the model to handle other multi-mode networks with various properties.

Actor Attributes. As we have shown in the introduction, a multi-mode network might have attributes for certain modes of actors. In this situation, we can simply add the weighted attributes as features in P_i^t (12) when updating the cluster indicator matrix. That is:

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

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

Within-mode Interactions. It is possible that there is within-mode interaction in a complicated network, e.g, the paper citations in the example of academic publications (Figure 2). For undirected within-mode interaction, we can just add interaction matrix to M_i^t (10). For directed within-mode interaction $R_{i,i}$, we can construct a symmetric similarity matrix by averaging the commutative strength of interactions:

$$R_{symmetric}^t = (R_{i,i}^t + (R_{i,i}^t)^T)/2,$$

or compute a normalized similarity matrix following random walk as shown in [38]. As for one mode, actors within the same community tend to have dense within-group interaction and sparse inter-group interaction. This observation is naturally obeyed when the similarity matrix M_i^t is affected with within-mode interaction for updating clustering.

Inactive and Emerging Nodes. Some actors might become inactive at a certain time period. Meanwhile, new actors might join the network at a specific time stamp. The framework presented in previous section focuses on cases where actors under study do not change. Here, we analyze more realistic cases where actors hibernate or join a network.

Note that our derived algorithm just uses $C^{(i,t\pm1)}$ as attributes for current time stamp. When an actor dies or hibernates at certain time stamp, we can delete the 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 the hard clustering, some tiny deviation from orthogonality does not affect the performance. On the other hand, when lots of actors become inactive, the total weights of remaining actors after removal is relatively small, thus affect less while calculating the similarity matrix M_i^t. This also makes sense since a drastic membership change denotes the latent community is experiencing an "earthquake", thus it is not necessary to over-regularize the temporal change.

When $\mathbb{X}_i$ at time stamp t has new actors, a reasonable way is to set the entries related to the new actors in similarity matrix (computed by $C^{(i,t\pm1)}(C^{(i,t\pm1)})^T$) to be the average similarity as in[10]. We adopt a simple strategy in this case by setting the entries corresponds to the new nodes to 0 in $C^{(i,t\pm1)}$. As we know, the indicator matrix approximates the discrete cluster by $1/\sqrt{n_i}$ [37], where n_i is the number of members in the cluster. Hence, setting the new actor's indicator matrix of neighboring time stamps to 0, has the tendency to attach this new member to the large community.

Online Clustering. In some real-world scenarios, the task is to track the cluster evolution in an online fashion. Then the algorithm could be modified accordingly. Instead of iteratively updating $C^{(i,t)}$, we fix t. The corresponding updating matrix P_i^t and M_i^t only involve fixed $C^{(i,t-1)}$ and does not involve $C^{(i,t+1)}$.

5. EXPERIMENTS

In this section, we compare our proposed evolutionary method in Figure 3 with the method that does not consider any temporal information on both synthetic and real-world data. For completeness, we include the results of both online and offline evolutionary methods.

5.1 Synthetic Data

Since typical real-world data does not have the ground truth (the latent community of each social actor at certain time stamp) available, we resort to synthetic data as a proof of concept to show the efficacy of our algorithm. The synthetic network consists of three modes: A, B and C, with 2, 3 and 4 clusters, and 50, 100, and 200 actors, respectively. The interactions occur between each of the two modes. To generate the interaction, we

- Determine the latent community for each actor;
- Generate interactions following a Bernoulli distribution based on the group identity of interacted two actors.

To simulate the evolution,

- The community membership is determined in the initial stage. Later in each time stamp, certain percentage of $(5\% - 20\%)$ actors change their group identity randomly;
- The probability of interaction between different groups is sampled with a Gaussian distribution centered on the probability of previous time stamp;
- Uniform random noise is added in each time stamp.

Three methods are compared: independent, online, and evolutionary spectral multi-mode clustering. To study the property of the methods, we generate data of various noise level. For each level, 10 synthetic network are constructed. For simplicity, we set all the weights $w_a^{(i,j)}$ and w_b^i in (6) to 1, and deem the algorithm to be converged if the relative objective change is less than 10^{-6}. Normalized mutual information (NMI) [31] is adopted to evaluate the clustering performance. NMI is a measure between 0 and 1. NMI=1 when two clusters are exactly the same. The average performance of all the methods are reported in Table 1.

It is evident that the performance deteriorates as the noise increases. When the noise level is low (the left part of the table), all the methods perform similarly. When the noise is of medium level, our proposed evolutionary multi-mode clustering approach outperforms the other two. Surprisingly, it is difficult to distinguish online clustering from independent clustering. This is probably due to the fact that we do not perform any parameter tuning.

To show the effect of parameter tuning, we choose one data set of medium noise level and apply both online clustering and evolutionary multi-mode clustering with temporal weights w_b varying from 0.01 to 1000 and w_a fixed to 1. As shown in Figure 4, it might worsen the performance if the temporal weight is too large, that is, when the temporal regularization dominates. However, for all the three modes A, B and C, taking into account the temporal information could help clustering. This is implied when the temporal weight is within the range of 0.01 and 100. It is noticed that complicated clustering structure (e.g. mode C has more clusters than the other 2 modes) requires a larger temporal weight. This could be suggestive for practical deployment.

We also plot the average computation time in Figure 5. Independent clustering takes the least time and online clus-

Table 1: NMI on synthetic data with three modes A, B and C over 10 runs. The noise are gradually increasing from left to right when generating the network. Bold entries denote the best one among the methods.

A	Independent	**0.999**	**0.999**	**0.994**	0.9515	0.9502	0.8579	0.6666	**0.432**	0.3086	0.2192
	Online	**0.999**	**0.999**	**0.994**	0.9544	0.9487	0.8584	**0.675**	0.4234	**0.309**	**0.22**
	Evolutionary	**0.999**	**0.999**	**0.994**	**0.966**	**0.96**	**0.875**	0.6638	0.4308	0.3072	0.2026
B	Independent	0.933	0.9164	0.9092	0.8692	0.8078	0.7251	0.5603	0.4286	0.2987	0.1922
	Online	**0.941**	**0.936**	0.9068	0.8771	0.824	0.7195	0.5637	0.4245	0.298	0.1904
	Evolutionary	0.9377	0.9269	**0.911**	**0.885**	**0.824**	**0.736**	**0.588**	**0.446**	**0.308**	**0.196**
C	Independent	0.8694	**0.84**	0.8652	0.7532	0.7042	0.6094	0.4742	0.3469	0.2442	0.1332
	Online	**0.875**	0.8365	0.8644	0.7601	0.7033	0.6062	0.4775	0.3497	0.2453	**0.135**
	Evolutionary	0.8678	0.8296	**0.868**	**0.77**	**0.717**	**0.621**	**0.492**	**0.366**	**0.251**	0.1343

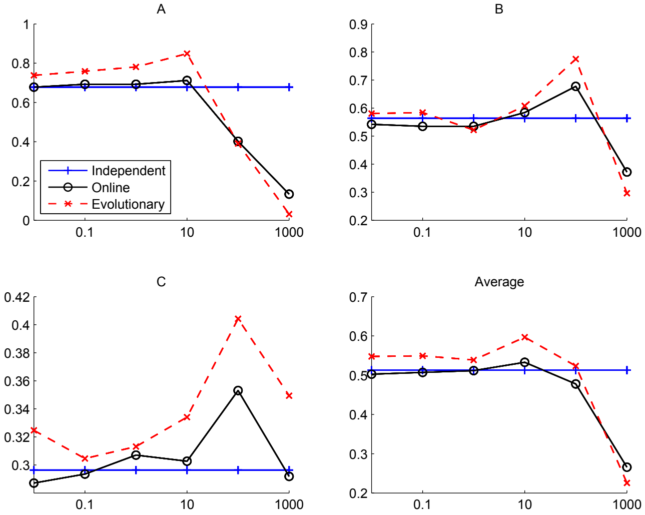


Figure 4: Sensitivity to the temporal weights

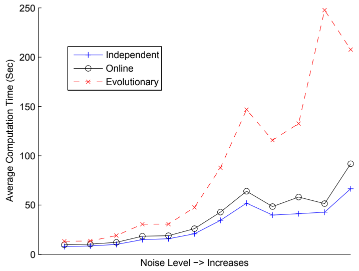


Figure 5: Average Computation Time

tering is comparable. Evolutionary clustering requires more time compared with others. This is especially severe when the noise level is very high. In that case, the temporal smoothness is destroyed, thus our algorithm requires more iterations to find an optimal.

5.2 Real-world Data

Two publicly available network data are selected: one is Enron email corpus[2], and the other is DBLP data[3].

Enron data [18] provides email documents collected from 150 senior executives in the Enron company. Based on the email communication, we construct a three-mode network (user, email, words) for each month. Here, we focus on users (i.e., email addresses) who send and receive at least 5 emails, reducing the total number of users to 2,359. 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. Due to the data collection problem, the initial stage yields low email traffic. Thus, we only consider the time period from April, 2001 to March 2002. There are three interactions in each month: sender-email, email-receiver, and email-words. The weight of email-receiver is normalized by the number of receivers. For word frequency, we notice that some words occur very frequently in one email. To avoid the dominance of these bursty words, we set the weight of interaction to be $\log(\text{word freq}) + 1$ and then scale each email to unit length.

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 between 1980-2004, remove those papers without authors or venue information, and terms with low document frequency (less than 20) and stop words, resulting in 491,726 papers, 347,013 authors, 2,826 venues, and 9,523 terms. Figure 6(b) demonstrates the number of papers, active authors, venues, and terms in each year, respectively.

5.2.1 *Evaluation Methodology*

Different from synthetic data, no ground truth of the community membership is available for real-world data. Thus it is difficult to verify the performance of our proposed method directly using NMI as shown on the synthetic data. Instead, we present an approach to compare the relative performance of different methods. For convenience, we present the evaluation approach based on the Enron data here. Similar strategy can be derived for the DBLP data.

We follow the standard approach to split the Enron data of each month into two parts. The first half is used for training to obtain the clusters of users and words, and 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 only focus on how the cluster of words and users approximate the interaction in the test data.

Suppose we are given an interaction $R_{email,i}$ between email and $\mathbb{X}_i$ (being sender, receiver, or words) for testing. Ideally, $R_{email,i}$ can be factored into the following form

$$R_{email,i} = C_{email} A C_i^T$$

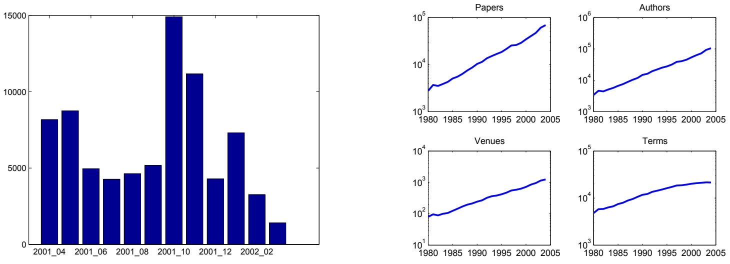
where C denotes the cluster membership of different modes. To measure the quality of C_i, one natural solution is to use the approximation error $R_{email,i} - C_{email} A C_i^T$. A cluster indicator matrix can be constructed based on the cluster membership and then plugged into C_i to minimize the ap-

[2] http://ciir.cs.umass.edu/%7Ecorrada/enron/

[3] http://kdl.cs.umass.edu/data/dblp/dblp-info.html

Table 2: Performance on Enron data based on one year network information. "Inde" in the first row represents clustering without any any temporal information; "Online" represents an online version of evolutionary multi-mode clustering; "Evol" in the last row represents our proposed evolutionary multi-mode clustering. The bold entry in each column represents the best performance of the three approaches.

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



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

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

(a) (b) (c)

Figure 6: (a): Emails sent each month on Enron data; (b) Number of papers, active authors, venues and terms each year on DBLP data; (c) Venue communities extracted from DBLP data in 2004.

proximation error. Note that the calculation of a hard clustering of C_{email} might introduce more noise in the process, so we directly approximate $R_{email,i}$ by C_i^T in the following form:

$$\min_B ||R_{email,i} - BC_i^T||_F^2$$

It is not difficult to verify that the minimal value is $||R||_F^2 - ||RC||_F^2$, obtained when $B = RC$. Since $||R||_F^2$ is a constant, the cluster assignment that maximizes $||RC||_F^2$ yields a low approximation error. Therefore, we adopt $||RC||_F^2$ in measuring the quality of a clustering result. The larger the value, the better a cluster.

Note that this strategy, essentially, is to approximate the sparse interaction with group interaction blocks. Even with the true latent community structure, the approximation error could still be large. It is not appropriate to determine the significance of improvement by comparing the absolute value directly. Here, we adopt this measure mainly to compare the *relative* performance of different methods.

5.2.2 Performance Comparison

For simplicity, we set all the input weights of the algorithm in Figure 3 to 1. The number of communities of each mode is fixed to 50 and 20 for Enron and DBLP, respectively. The performance of different methods on Enron data are reported in Table 2. Note that most of the time, our proposed method works best. Except three months: October, November 2001, and January 2002, in which independent multi-mode clustering method works best. Note that during these three months, the email communication is abundant as shown in Figure 6(a). That is, when the training process has enough data, it is not necessary to consider any temporal information. Online clustering, as shown in the table, yields a limited improvement and sometimes could worsen the performance.

For DBLP data, as shown in Table 3, evolutionary clustering often outperforms other two. Online clustering occasionally performs best, and usually better than clustering without temporal regularization. Independent clustering, which does not consider temporal smoothness, tends to overfit the training data with a poor latent community structure. In general, taking into account the temporal smoothness helps to extract a better community structure.

Table 3: Performance on DBLP data ($\times 10^4$)

Year	Independent	Online	Evolutionary
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**

One question remains: dose our method find out meaningful community and evolutions in reality. In the first block of Figure 6(c), we show one example of venue community concentrating on Web extracted from our method, which is reasonably consistent with the topics associated with those conference and journals. To check the validity of our method for detecting evolution, we focus on NIPS conference as an example. We found that *NIPS* is associated with *Neural Network* in 1995, but is more aligned with machine learning later as shown in the second block of Figure 6(c). This indicates that our method can detect the evolution of community in multi-mode networks and find out the latent community more accurately.

We also recorded the computation time for all the methods in Table 4. Independent clustering converges quickly, while online clustering moderately increases the time, and evolutionary clustering costs most computational time. However, as seen in the table, the computational cost is of the same order on both data sets. We found out that independent cluster usually converges in 3 or 4 iterations, whereas evolutionary clustering requires less than 10 iterations.

Table 4: Computation Time

Method	Independent	Online	Evolutionary
Enron	5.0699×10^3	8.4974×10^3	1.1076×10^4
DBLP	2.1033×10^3	2.6945×10^3	5.0491×10^3

6. RELATED WORK

There have been extensive works studying the structural property of interactions between actors. One probabilistic approach is the stochastic block model [27], in which the link between actors is generated conditioned on the latent cluster membership of actors. Two actors within the same cluster are treated *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 are fixed. The work in [17] 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 proper prior. Mixed block model is also developed [1]. Long et al [25] proposes a probabilistic framework similar to stochastic block model to handle multi-mode networks with interactions and attributes. Typically, some MCMC technique is employed to approximate the inference. With the development of topic models [7], it is also extended to model documents within a social network [39], and the author/document, or sender/receiver/email interactions [34, 26]. The model is typically specific for certain type of documents like Emails or papers.

Another attempt to model the structure is latent space model. Intuitively, latent space models map the social actors to a latent low-dimensional space such that the actors whose positions are closer to each other are more likely to interact with each other [29, 16]. However, existing latent space models mainly study one-mode networks. Some works try to address the problem in multi-mode networks. [14] essentially study a two-mode network (authors and words) and maps both authors and words into the same Euclidean space.

Spectral relational clustering, which is most related to multi-mode network, tries to discover the latent structure 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 [11, 36, 24] tries to address the problem of clustering both words and documents simultaneously by taking the advantage of bipartite. [13] extends the problem to a star-typed network with multiple heterogeneous data object, and proposes semi-definite programming to solve the problem. [33] proposes reinforce clustering for multiple heterogeneous data objects. In [23], a general spectral clustering framework is proposed to handle multi-type relational clustering with different kinds of objects and attributes, and an alternating optimization algorithm is presented to find a local optimal.

Temporal change of social networks has been attracting increasing attentions [8, 2]. It is empirically observed that some real-world networks are evolving [19] and some practitioners try to investigate how the network evolve and what could be a reasonable generative process to model the dynamics [21] or the critical factors to determine the group evolution [4]. On the other hand, clustering to handle evolutionary data is also developed. It is assumed that clustering result of current situation should be similar to previous time stamps. Instead of taking multiple snapshots of the data and independently clustering objects, evolutionary clustering finds out a sequence of clustering with temporal smoothness [9, 10, 22]. Latent space model with temporal change is also developed [30], which aims to find an embedding that is consistent with the trade-off between previous time stamp and current distance information extracted from the social network. [32] proposes a general framework to handle dynamic single-mode network by casting it as a graph coloring problem and some greedy heuristics is developed to handle large-scale data. All the aforementioned works are focusing on data with attributes or single-mode network. In this work, we explore the community evolution in multi-mode networks.

7. CONCLUSIONS

In some real-world applications, social actors of various types interact with each other resulting in a multi-mode network. In such a network, different modes of social actors form their own community and tend to evolve gradually. In this work, we propose an evolutionary multi-mode clustering algorithm to find community evolution in multi-mode networks. Within this framework, it is easy to handle networks with attributes, within-mode interactions, and hibernating and emerging social actors. We applied our framework to both carefully designed synthetic data sets and real-world large scale multi-mode networks. It is empirically demonstrated that our algorithm tends to find more accurate community information given a sequence of network snapshots. This is more likely when the collected interactions are few and noise is within medium level. Our algorithm can easily scale to large complicated multi-mode networks of tens of thousands of nodes.

The shortcoming of the current framework is that it needs users to provide weights for different interactions and temporal information, as well as the number of communities in each mode. How to automatically determine the weights and number of communities is a challenging problem. Another possible extension is to consider the evolution of group interaction. Currently, our model only considers the group membership change. But the interactions between groups could also change gradually. How to capture both the micro-level evolution of actors and macro-level evolution of groups simultaneously requires further research.

8. ACKNOWLEDGMENTS

The work is in part supported by a MITRE research grant.

9. REFERENCES

[1] E. Airodi, D. Blei, S. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 2008.

[2] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *KDD*, pages 913–921, 2007

[3] F. R. Bach and M. I. Jordan. Learning spectral clustering. In *NIPS*, 2004.

[4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD*, 2006.

[5] 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.

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

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[8] R. Breiger, K. Carley, and P. Pattison, editors. *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*. 2003.

[9] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD*, pages 554–560, 2006

[10] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*, pages 153–162, 2007

[11] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001

[12] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*, pages 126–135, 2006

[13] 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*, pages 41–50, 2005

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

[15] G. H. Golub and C. F. V. Loan. *Matrix Computation*. Johns Hopkins University Press, 1996.

[16] M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal Of The Royal Statistical Society Series A*, 127(2), 2007.

[17] C. Kemp, T. L. Griffiths, and J. B. Tenenbaum. Discoverying latent classes in relational data. Technical report, Massachusetts Institute of Technology, 2004.

[18] B. Klimat and Y. Yang. The enron corpus: A new dataset for email classification research. In *ECML*, 2004.

[19] G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757), 2006.

[20] M. N. Lauren Ancel Meyers and B. Pourbohloul. Predicting epidemics on directed contact networks. In *Journal of Theoretical Biology*, volume 240, 2006.

[21] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, pages 177–187, 2005.

[22] 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*, pages 685–694, 2008.

[23] B. Long, Z. M. Zhang, X. Wú, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML*, pages 585–592, 2006.

[24] B. Long, Z. M. Zhang, and P. S. Yu. Co-clustering by block value decomposition. In *KDD*, 2005.

[25] B. Long, Z. M. Zhang, and P. S. Yu. A probabilistic framework for relational clustering. In *KDD*, pages 470–479, 2007.

[26] 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*, (0):249–272, 2007.

[27] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[28] 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.

[29] M. S. H. Peter D. Hoff, Adrian E. Raftery. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 2002.

[30] P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, 2005.

[31] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

[32] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *KDD*, 2007.

[33] 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.

[34] X. Wang, N. Mohanty, and A. McCallum. Group and topic discovery from relations and their attributes. In *NIPS*, pages 1449–1456. 2006.

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

[36] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM*, pages 25–32, New York, NY, USA, 2001. ACM.

[37] H. Zha, X. He, C. H. Q. Ding, M. Gu, and H. D. Simon. Spectral relaxation for k-means clustering. In *NIPS*, pages 1057–1064, 2001.

[38] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML*, pages 1036–1043, 2005.

[39] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *WWW*, pages 173–182, 2006.