

Analyzing Communities and Their Evolutions in Dynamic Social Networks

YU-RU LIN

Arizona State University

YUN CHI and SHENGHUO ZHU

NEC Laboratories America

HARI SUNDARAM

Arizona State University

and

BELLE L. TSENG

YAHOO! Inc.

8

We discover communities from social network data and analyze the community evolution. These communities are inherent characteristics of human interaction in online social networks, as well as paper citation networks. Also, communities may evolve over time, due to changes to individuals' roles and social status in the network as well as changes to individuals' research interests. We present an innovative algorithm that deviates from the traditional two-step approach to analyze community evolutions. In the traditional approach, communities are first detected for each time slice, and then compared to determine correspondences. We argue that this approach is inappropriate in applications with noisy data. In this paper, we propose *FacetNet* for analyzing communities and their evolutions through a robust *unified* process. This novel framework will discover communities and capture their evolution with temporal smoothness given by historic community structures. Our approach relies on formulating the problem in terms of maximum a posteriori (MAP) estimation, where the community structure is estimated both by the observed networked data and by the prior distribution given by historic community structures. Then we develop an iterative algorithm, with proven low time complexity, which is guaranteed to converge to an optimal solution. We perform extensive experimental studies, on both synthetic datasets and real datasets, to demonstrate that our method discovers meaningful communities and provides additional insights not directly obtainable from traditional methods.

Authors' addresses: Y-R Lin and H. Sundaram, Arts, Media and Engineering Program, Arizona State University, Tempe, AZ 85281; Y. Chi and S. Zhu, NEC Labs America, 10080 N. Wolfe Rd, SW3-350, Cupertino, CA 95014. B. L. Tseng, YAHOO! Inc., 2821 Mission College Blvd, Santa Clara, CA 95054; email: Yu-ru.lin@asu.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2009 ACM 1556-4681/2009/04-ART8 \$5.00
DOI 10.1145/1514888.1514891 <http://doi.acm.org/10.1145/1514888.1514891>

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications—*Data mining*; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information filtering*; J.4 [Computer Applications]: Social and Behavioral Sciences—*Economics*

General Terms: Algorithms, Experimentation, Measurement, Theory

Additional Key Words and Phrases: Community, community net, evolution, evolution net, nonnegative matrix factorization, soft membership

ACM Reference Format:

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., and Tseng, B.L. 2009. Analyzing communities and their evolutions in dynamic social networks. *ACM Trans. Knowl. Discov. Data.* 3, 2, Article 8 (April 2009), 31 pages. DOI = 10.1145/1514888.1514891 <http://doi.acm.org/10.1145/1514888.1514891>

1. INTRODUCTION

This article focuses on the development of a comprehensive framework to analyze community evolution from online social networks. This is motivated by the central belief that semantics of information is an emergent artifact of human activity, and the relationship between semantic concepts and their manifestations (i.e., Web pages, blog posts) evolves over time. Human networks have been the locus for the construction of meaning hence it is crucial to find real human networks that collaborate with respect to some activity.

Data from many social network datasets, including paper coauthorship networks and the blogosphere, is a graph where nodes represent individuals (e.g., club members, authors, and bloggers) and edges represent the relationship and interactions among individuals (e.g., interactions in a club, coauthorship, and hyperlinks in blogs). In such social networks, individuals form communities by building relationships and interactions with each other. The analysis of these communities (membership, structure and temporal dynamics) is an important research issue.

Traditional analysis of social networks treats the network as a *static* graph, where the static graph is either derived from aggregation of data over all time or taken as a snapshot of data at a particular time. These studies range from well-established social network analysis [Wasserman and Faust 1994] to recent successful applications such as HITS and PageRank [Kleinberg 1999; Page et al. 1998]. However, this research omits one important feature of communities in networked data—the temporal evolution of communities. By ignoring community evolution, prior works have overlooked a key aspect of online communities.

More recently, there has been a growing body of work on the analysis of communities and their temporal evolution in *dynamic networks* [Asur et al. 2007; Kumar et al. 2003; Kumar et al. 2006; Leskovec et al. 2005; Lin et al. 2007; Palla et al. 2007; Spiliopoulou et al. 2006; Toyoda and Kitsuregawa 2003]. However, a common weakness in these studies, as we will discuss in detail in related work, is that communities and their evolutions are studied separately—usually, community structures are independently extracted at consecutive timesteps and then in retrospect, evolutionary characteristics are introduced to explain

the difference between these community structures over time. Such a two-stage approach may make sense when the community structure is unambiguous (e.g., when the community affiliation is available). However, more often than not, data from real-world networks are ambiguous and subject to noise. Under such scenarios, if an algorithm extracts community structure for each timestep independently of other timesteps, it often results in community structures with high temporal variation. Consequently, undesirable evolutionary characteristics may have to be introduced in order to explain the high variation in the community structures. Therefore, we argue that a more appropriate approach is to analyze communities and their evolutions in a unified framework where the community structure provides evidence about community evolutions and at the same time, the evolutionary history offers hints on what community structure is more appropriate. For example, a community structure that introduces dramatic evolutions in a very short period of time is less desirable.

Another common problematic issue in current community analysis techniques is that an individual is usually assigned to only one community at a time. On the contrary, an individual may be engaged in multiple communities at the same time. For example, a blogger who is a dance guru may also be an amateur photographer at the same time. Because of this, an individual who usually participates in multiple communities should be assigned to multiple communities at the same time. Therefore, instead of a hard community partition, we argue that a *soft* community membership is more informative, as it provides more details about how an individual participates in each of the communities.

In this paper, we propose a systematic framework for analyzing communities and their evolutions in dynamic networks, and we term our framework *FacetNet*.¹ Our main contributions are threefold:

1. We introduce the *FacetNet* framework to analyze communities and their evolutions in a unified process. In our framework, the community structure at a given timestep t is determined both by the observed networked data at t and by the prior distribution given by historic community structures. Algorithmically, we propose the first probabilistic generative model for analyzing communities and their evolution. We show that the proposed model solves the evolutionary clustering problem from a probabilistic (Bayesian) perspective. Empirically, the discovered communities and their evolutions are more robust to noise and more reasonable (e.g., dramatic change in a short time is unlikely).
2. We adopt a stochastic block model for generating communities and a probabilistic model based on the Dirichlet distribution for capturing the community evolutions. These probabilistic models naturally assign soft community memberships to nodes and these models do not suffer from the problem of non-identifiability of parameters which exists in most existing models. Based on the probability distributions computed by the models, we further

¹*FacetNet* stands for “a Framework for Analyzing Communities and EvoluTions in dynamic NETworks.”

introduce two novel concepts—*Community Net* and *Evolution Net*—to interpret the community-level interaction and transition, respectively.

3. We provide an iterative EM algorithm that is guaranteed to converge to optimal solutions to the proposed formulation. We prove the correctness and convergence of our algorithm and show that this algorithm has low time complexity when the data is sparse. We also provide principled solutions to some practical issues, such as how to determine the number of communities and how to handle adding and removing of individuals in a dynamic network.

We use synthetic and real datasets (including a blog dataset and a paper coauthorship dataset) to demonstrate that compared to traditional methods, our framework provides more accurate and robust results on communities and their evolutions. We also show that our framework is able to discover interesting insights in dynamic networks that are not directly obtainable from existing methods.

This article is a significant extension of our prior work [Lin et al. 2008] and is our first comprehensive discussion on this subject. In this article, we include new theoretical and experimental results as well as detailed algorithms and proofs. In particular there are several major extensions over prior work [Lin et al. 2008].

1. An extension of the theoretical framework: We present a theoretically motivated and more generalized framework in this article. We reformulate the problem in terms of maximum a posteriori (MAP) estimation and show a close connection between the proposed generative probabilistic model and the optimization framework for solving the evolutionary clustering problem (Section 2).
2. Details of the algorithms: We provide an EM algorithm for solving the MAP problem and provide the proof of the convergence of this algorithm (Section 4). In addition, we provide an algorithm with proof of correctness for solving the extended MAP problem where the number of communities is allowed to change over time (Section 5).
3. New and extensive experiment results: In the experiment on synthetic datasets, we compare our algorithm with three baseline algorithms in terms of a graph based error rate measure. We provide a new comparison against the variability of the ground truth community structures and show a significant performance improvement of our algorithm (Section 6) in this aspect. Moreover, we provide more experimental results from a real world paper citation dataset. These concrete findings suggest the applicability of our algorithm to the large scale social networks (Section 6).

The rest of the article is organized as follows. First, we discuss related work. In Section 2, we present the formulation of our model in detail. In Section 3, we describe how to extract communities and their evolutions from the results learned by our probabilistic model. In Section 4, we propose an iterative EM algorithm for solving our model and discuss the time complexity of the proposed algorithm. In Section 5, we introduce extensions of our framework to handle some practical

issues. In Section 6, we provide experimental studies. Finally in Section 7, we give the conclusion and future directions.

1.1 Related Work

Community formation has been extensively studied in various research areas such as social network analysis, Web community analysis, computer vision, etc. In social network analysis, an important research topic is to identify cohesive subgroups of individuals within a network where cohesive subgroups are defined as “subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties” [Wasserman and Faust 1994]. Many approaches, such as clique-based, degree-based, and matrix-perturbation-based, have been proposed to extract cohesive subgroups from social network [Wasserman and Faust 1994]. Communities also play an important role in Web analysis. For example, Flake et al. [2000] defined Web communities as “a set of sites that have more links to members of the community than to non-members,” and proposed algorithms to identify Web communities based on a maximum flow/minimum cut framework. Newman and Girvan [2004] defined a metric called modularity measure to quantify the strength of community structure and they also proposed several community extraction algorithms by using this modularity measure. We will discuss the details about modularity in a later section. In computer vision, community extraction is closely related to image segmentation problem, which is a problem to partition an image into multiple regions so that pixels within the same region are similar and adjacent regions are significantly different. One effective method in this area is the spectral clustering algorithm [Chung 1997; Dhillon et al. 2004; Shi and Malik 2000; Zha et al. 2001] where the eigenvectors of certain normalized similarity matrices are used for the clustering purpose. Later, White and Smyth [2005] pointed out the close relationship between Newman’s modularity and the spectral clustering and proposed several algorithms to combine the two approaches. Yu et al. [2005] proposed a novel clustering framework on graphs where the cluster memberships are assigned in a probabilistic way. In Yu’s framework, cluster memberships can be extracted in different resolutions, representing local or global cluster structures. A common issue in all the above studies is that they only analyzed *static networks* where no temporal analysis is used for evolution study. Another issue is that most of these studies treat community extraction as a graph partition problem and therefore always result in hard community memberships, which disallows an individual to participate multiple communities at the same time.

Recently, there exists a growing body of literature on analyzing communities and their evolutions in *dynamic networks*. Kumar et al. [2003] studied the evolution of the blogosphere as a graph in terms of the change of characteristics, (such as in-degree, out-degree, strongly connected components), the change of communities, as well as the burstiness in blog community. Leskovec et al. [2005] studied the patterns of growth for graphs in various fields and proposed generators that produce graphs exhibiting the discovered patterns. Palla et al. [2007] analyzed a coauthorship network and a mobile phone network, where

both networks are dynamic. They use the clique percolation method (CPM) to extract communities at each timestep and then match communities in consecutive timesteps to analyze community evolution. They studied some interesting characteristics, such as community sizes, ages and their correlation, community autocorrelation (relative overlap between the same community at two timesteps t_1 and t_2 as a function of $\tau = t_2 - t_1$), etc. Toyoda and Kitsuregawa [2003] studied the evolution of Web communities from a series of Web achieves. They first proposed algorithms for extracting communities in each timestep. And then they proposed different types of community changes, such as emerge, dissolve, grow, and shrink, as well as a set of metrics to quantify such changes for community evolution analysis. Spiliopoulou et al. [2006] proposed a framework, MONIC, to model and monitor cluster transitions over time. They defined a set of *external transitions* such as survive, split, disappear, to model transactions among different clusters and a set of *internal transitions*, such as size and location transitions to model changes within a community. Asur et al. [2007] introduced a family of events on both communities and individuals to characterize evolution of communities. Sun et al. [2007] proposed a parameter-free algorithm, GraphScope, to mine time-evolving graphs where the Minimum Description Length (MDL) principle is employed to extract communities and to detect community changes. Mei and Zhai [2005] extracted latent themes from text and used the evolution graph of themes for temporal text mining. All these studies, however, have a common weak point—community extraction and community evolution are analyzed in two separated stages. That is, when communities are extracted at a given timestep, historic community structure, which contains valuable information related to current community structure, is not taken into account.

There are some recent studies on evolutionary embedding and clustering that are closely related to our work. Sarkar and Moore [2005] proposed a dynamic method that embeds nodes into latent spaces where the locations of the nodes at consecutive timesteps are regularized so that dramatic change is unlikely. Chakrabarti et al. [2006] proposed the first evolutionary clustering methods where the cluster membership at time t is influenced by the clusters at time $t-1$. As a result, the cluster membership for a node at time t depends both on its relationship with other nodes at time t and on its cluster membership at time $t-1$. Chi et al. [2007] extended similar ideas and proposed the first evolutionary spectral clustering algorithms. They used graph cut as a metric for measuring community structures and community evolutions.

All these studies differ from our work in that they regularize the current community membership at time t by using historic community membership *indirectly*. In Chakrabarti et al.'s evolutionary hierarchical clustering algorithm, historic community structure affects the tree-node merging step in the current time. In their evolutionary k -means clustering algorithm, historic centroids affect the k -mean process at the current time. In Chi et al.'s algorithms, certain eigenvectors, instead of the community structure, are regularized over time. In the work of Sarkar et al., although the relationship among nodes in latent spaces is preserved over time, the issue of communities are not directly

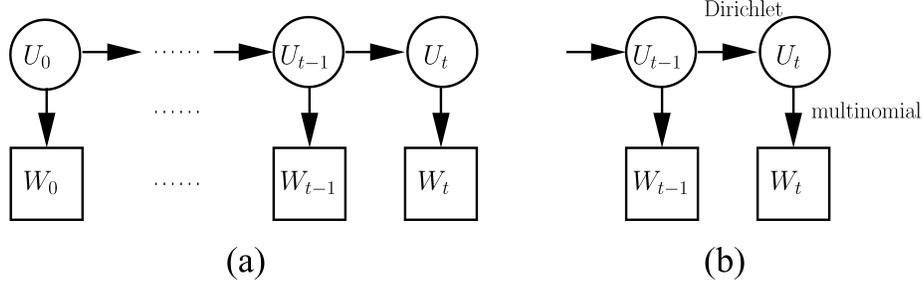


Fig. 1. Schematic illustrations of (a) our high-level abstract probabilistic model and (b) one instance of the concrete probabilistic model.

addressed. In contrast, in our proposed framework, the community membership itself is directly regularized over time.

2. FORMULATION

2.1 Notation

First, a note on notation. In this paper, we use lower-case letters, for example, x , to represent scalars, vector-formed letters, e.g., \vec{v} , to represent vectors, and upper-case letters, for example, W , to represent matrices. Both w_{ij} and $(W)_{ij}$ represent the element at the i -th row and j -th column of W . We use $\text{vec}(W)$ to denote the vectorization of W , that is, stacking the columns of W into a column vector. A subscript t on a variable, for example, W_t or $w_{t,ij}$, denotes the value of that variable at time t . However, to avoid notation clutter, we try not to use the subscript t unless it is needed for clarity.

We assume that edges in the networked data are associated with discrete timesteps. We use a *snapshot* graph $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t, W_t)$ to model interactions at time t where in \mathcal{G}_t , each node $v_i \in \mathcal{V}_t$ represents an individual, each edge $e_{ij} \in \mathcal{E}_t$ represents the presence of interactions between v_i and v_j , and $w_{t,ij} = (W_t)_{ij}$ denotes the edge weight of e_{ij} (i.e., the frequency of interactions between nodes i and j observed at time t). Assuming \mathcal{G}_t has n nodes, $W_t \in \mathcal{R}_+^{n \times n}$ (nonnegative matrix of size $n \times n$) is the adjacency matrix for \mathcal{G}_t . Over time, the interaction history is captured by a sequence of snapshot graphs $\langle \mathcal{G}_1, \dots, \mathcal{G}_t, \dots \rangle$ indexed by time.

2.2 Basic Formulation

We start by introducing our probabilistic generative model that describes communities and their evolutions. The basic principles behind our models are (1) the data observed at time t (i.e., W_t) is generated from the community structure at time t (the structure parameter set is denoted by U_t), following a certain distribution, and (2) the community structure U_t follows a certain distribution that is determined by the community structure U_{t-1} at time $t-1$. Furthermore, we assume that W_t is independent of U_{t-1} given U_t , as illustrated in Figure 1(a).

We estimate the parameters $\{U_t\}$ using maximum a posteriori (MAP) estimation. For simplicity, in this paper we only discuss the forward estimation,

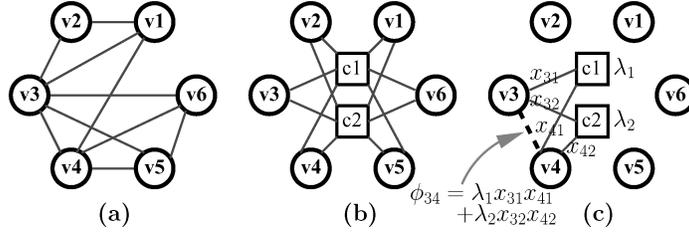


Fig. 2. Schematic illustration of the stochastic block model: (a) the original graph, (b) the bipartite graph with two communities, and (c) how to approximate (ϕ_{34}) .

that is, assuming that U_{t-1} is given when estimating U_t . Mathematically, we write

$$U_t^* = \arg \max_U \log P(W_t|U)P(U|U_{t-1}).$$

Equivalently, we want to maximize the following

$$L(U_t) = \log P(W_t|U_t) + \log P(U_t|U_{t-1}). \quad (1)$$

Therefore, our high-level abstract model contains two components— $P(W_t|U_t)$ and $P(U_t|U_{t-1})$ —and these components can be defined in different ways to give different concrete models. In the following sections, we present some concrete models for $P(W_t|U_t)$ and $P(U_t|U_{t-1})$.

2.3 The Model for $P(W_t|U_t)$

For $P(W_t|U_t)$, we adopt a stochastic block model first proposed in Yu et al. [2005]. Assume there exist m communities at time t and these m communities introduce a probability $\phi_{t;ij}$ for the interaction between nodes i and j by using a mixture model. In the mixture model, $\phi_{t;ij} = \sum_{k=1}^m p_k \cdot p_{k \rightarrow i} \cdot p_{k \rightarrow j}$, where p_k is the prior probability that the interaction between nodes i and j is due to the k -th community, $p_{k \rightarrow i}$ and $p_{k \rightarrow j}$ are the probabilities that an interaction in community k involves node v_i and v_j , respectively. We explicitly write a parameter set U_t , as a nonnegative matrix $X_t \in \mathcal{R}_+^{n \times m}$ and an $m \times m$ nonnegative diagonal matrix Λ_t , where $x_{t;ik} = p_{k \rightarrow i}$, $\lambda_{t;k} = p_k$, $\sum_i x_{t;ik} = 1$ and $\sum_k \lambda_{t;k} = 1$, ($\lambda_{t;k}$ is short for $\lambda_{t;kk}$). Thus, we have $\phi_{t;ij} = (X_t \Lambda_t X_t^T)_{ij}$. An alternative model is that $\{\phi_{t;ij}\}$ is generated by a bipartite graph where nodes on one side of the graph represent the original n nodes in W_t and those on the other side represent the m communities. In Figure 2, we use a toy example with 6 nodes and 2 communities to illustrate this model of community structure. For a general graph (a), we use a special bipartite graph (b) to approximate (a). Note that (b) has two more nodes c_1 and c_2 , corresponding to the two communities. Nevertheless, because (b) is a bipartite graph (i.e., an edge can only occur between a node v and a community c), it has less degree of freedom and so it is a more parsimonious explanation of (a). In (c), we show how the probability ϕ_{34} is generated in the mixture model as the sum of $\lambda_1 x_{31} x_{41}$ and $\lambda_2 x_{32} x_{42}$.

Matrices X_t and Λ_t fully characterize this mixture model and therefore we define $U_t = (X_t, \Lambda_t)$. We then model $P(W_t|U_t)$ by a multinomial distribution

with parameter $\phi_{t;ij}$, which results in

$$\begin{aligned}\log P(W_t|U_t) &= \log \frac{\Gamma(1 + \sum_{ij} w_{t;ij})}{\prod_{ij} \Gamma(1 + w_{t;ij})} \prod_{ij} \phi_{t;ij}^{w_{t;ij}} \\ &= \sum_{ij} w_{t;ij} \log \phi_{t;ij} + \text{const},\end{aligned}$$

where $\Gamma(\cdot)$ is the gamma function.

2.4 The Model for $P(U_t|U_{t-1})$

Because the columns of X_t all sum to ones and Λ_t is a diagonal matrix, $X_t \Lambda_t$ can uniquely determine X_t and Λ_t . Therefore, we have $P(U_t|U_{t-1}) = P(X_t \Lambda_t | X_{t-1} \Lambda_{t-1})$. Since we use the multinomial distribution to model $P(W_t|U_t)$ and since the conjugate prior for the multinomial distribution is the Dirichlet distribution, it is a natural choice to use a Dirichlet distribution as the prior distribution. We let $\text{vec}(X_t \Lambda_t)$ follow a Dirichlet distribution with parameter $\psi_t = \nu \cdot \text{vec}(X_{t-1} \Lambda_{t-1}) + \bar{1}$. Through the parameter ψ_t , the distribution $P(U_t)$ of the community structure at time t is determined by the community structure U_{t-1} at time $t-1$. Under this model, we have

$$\begin{aligned}\log P(U_t|U_{t-1}) &= \log \frac{\Gamma(\sum_j \psi_{t;j})}{\prod_j \Gamma(\psi_{t;j})} \prod_{ik} (X_t \Lambda_t)_{ik}^{\psi_{t;j} \cdot (X_{t-1} \Lambda_{t-1})_{ik}} \\ &= \sum_{ik} \nu \cdot (X_{t-1} \Lambda_{t-1})_{ik} \log (X_t \Lambda_t)_{ik} + \text{const}.\end{aligned}$$

The parameter ν can be set by the user. We will discuss the role of ν in the next section.

2.5 The Overall Model and Its Connection to Evolutionary Clustering

Putting $P(W_t|U_t)$ and $P(U_t|U_{t-1})$ together, we have

$$\begin{aligned}L(U_t) &= \log P(W_t|U_t) + \log P(U_t|U_{t-1}) \\ &= \sum_{ij} (W_t)_{ij} \log (X_t \Lambda_t X_t^T)_{ij} + \sum_{ik} \nu \cdot (X_{t-1} \Lambda_{t-1})_{ik} \log (X_t \Lambda_t)_{ik} + \text{const}. \quad (2)\end{aligned}$$

Figure 1(b) illustrates the overall concrete model that we propose.

The effect of $P(U_t|U_{t-1})$ is illustrated by using a toy example in Figure 3. Figure 3(a)–(c) show the interactions among the six nodes occurring at time $t = 1, 2$ and 3 , where the blue solid lines represent the new observed interactions at each time and the gray dotted lines are the interactions observed previously. Figure 3(d)–(f) show the community membership resulted from each observed snapshot alone (i.e. without considering the historic interactions). The horizontal axis indicates the probabilities of a node v_i belonging to the community $c1$ or $c2$, for example, a node lies at 1 means it belongs to $c2$ with probability 1 and belongs to $c1$ with probability 0. We can simply assign the membership to a node based on the probability value. As can be seen, the community structure at $t=1$ is obvious. At $t = 2$, the interactions between $v3$ and every other nodes make $v3$'s membership ambiguous, and this ambiguity suddenly changes

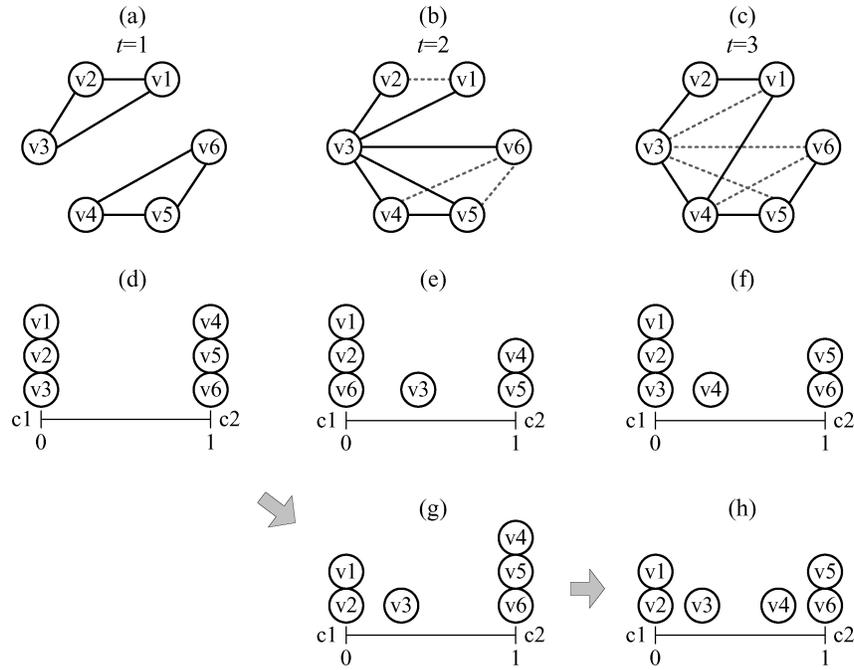


Fig. 3. A toy example for illustrating the evolutionary stochastic block model. (a)–(c): The interactions among the six nodes occurring at time $t = 1, 2$ and 3 . (At each time, blue solid lines represent new observed interactions and gray dotted lines represent previously observed interactions.) (d)–(f): The community membership resulted from each observed snapshot. The resultant abrupt membership fails to capture meaningful community evolution. (g)–(h): The community membership resulted from both the new observation and the historic community structures. The horizontal axis in (d)–(h) indicates the probabilities of a node v_i belonging to the community c_1 or c_2 . Our model gives more desirable results. By considering the historic community structure in (d), the membership of v_3 at $t = 2$ is more discriminate in (g) than in (e), and v_3 's indiscriminate interactions with everyone does not affect others' membership significantly.

v_6 's membership (Figure 3(e)). The abrupt membership assignment can also be noticed at $t = 3$ (v_3, v_4 and v_6). Such results are undesirable because they fail to capture meaningful community evolution. Under our model, both the new observation and the historic community structures (Figure 3(d) and (g)) are considered, which results in different membership assignment as shown in Figure 3(g) and (h). The results are close to our intuition: given we already observe data at $t = 1$, the membership of v_3 at $t = 2$ is more discriminate, and v_3 's indiscriminate interactions with everyone should not affect others' membership significantly.

It turns out that the model described by Equation (2) has a close connection to the evolutionary clustering framework ([Chakrabarti et al. 2006; Chi et al. 2007]). In evolutionary clustering, communities and their evolutions influence each other. That is, at time t , a community structure is preferred if the community evolution from $t-1$ to t is not unreasonably dramatic. To achieve this goal, the community structure at time $t-1$ (already extracted) is used to regularize the community structure at current time t (to be extracted). To incorporate such

a regulation, a cost function is introduced to measure the quality of community structure at time t , where the cost consists of two parts—a *snapshot cost* and a *temporal cost*:

$$\text{cost} = \alpha \cdot \mathcal{CS} + (1 - \alpha) \cdot \mathcal{CT}. \quad (3)$$

In this cost function, the snapshot cost \mathcal{CS} measures how well a community structure fits W_t , the observed interactions at time t . The temporal cost \mathcal{CT} measures how consistent the community structure is with respect to historic community structure (at time $t-1$). The parameter α is set by the user to control the level of emphasis on each part of the total cost.

To see the connection between our probabilistic generative model and that in evolutionary clustering, we define the snapshot cost \mathcal{CS} as the following. By using a bipartite graph as before, we approximate W_t , which has rank n , by a product in the form of $X_t \Lambda_t X_t^T$, which has rank m . Based on this model, we define the snapshot cost \mathcal{CS} as the error introduced by such an approximation, that is,

$$\mathcal{CS} = D(W_t \| X_t \Lambda_t X_t^T),$$

where $D(A \| B) = \sum_{i,j} (a_{ij} \log \frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij})$ is the KL-divergence between A and B . So the snapshot cost is high when the approximate community structure $X_t \Lambda_t X_t^T$ fails to fit the observed data W_t well.

The temporal cost \mathcal{CT} is used to regularize the community structure so that it is less probable for unreasonably dramatic community evolution from time $t-1$ to t . We can achieve this regularization by defining \mathcal{CT} as the difference between the community structure at time t and that at time $t-1$; that is,

$$\mathcal{CT} = D(X_{t-1} \Lambda_{t-1} \| X_t \Lambda_t),$$

where D is the KL-divergence as defined before. So the temporal cost \mathcal{CT} is high when there is a dramatic change of community structure from time $t-1$ to t .

Putting the snapshot cost \mathcal{CS} and the temporal cost \mathcal{CT} together, we have an optimization problem as to find the best community structure at time t , expressed by X_t and Λ_t , that minimizes the following total cost

$$\text{cost} = \alpha \cdot D(W_t \| X_t \Lambda_t X_t^T) + (1 - \alpha) \cdot D(X_{t-1} \Lambda_{t-1} \| X_t \Lambda_t) \quad (4)$$

subject to $X_t \in \mathcal{R}_+^{n \times m}$, $\sum_i x_{t;ik} = 1$, and Λ_t being an m -by- m nonnegative diagonal matrix.

The connection between our probabilistic generative model and the evolutionary clustering problem formulated in Equation (4) is the following: minimizing Equation (4) turns out to be equivalent to maximizing Equation (2) under certain conditions, as described in the following theorem. We provide the proof in the appendix.

THEOREM 2.1. *Under the assumption that $\sum_{ij} (W_t)_{ij}$ remains a constant over time (i.e., the number of observed interactions in all timesteps are the same), minimizing the cost in Eq. (4) is equivalent to minimizing the logarithm in Eq. (2) with $v = (1 - \alpha)/\alpha$.*

It is worth mentioning that although Equation (4) shares a similar form to that of evolutionary spectral clustering [Chi et al. 2007], it has a good property that is not in evolutionary spectral clustering. In evolutionary spectral clustering, the parameters learned are eigenvalues and eigenvectors and as a result, a second step of k -means clustering is needed to determine the final cluster memberships. With such a second step, in addition to being nonintuitive, evolutionary spectral clustering suffers from a problem of nonidentifiability of parameters. That is, the clusters at time $t-1$ are not explicitly mapped to those clusters at time t and therefore some partition-matching algorithms [Lovasz and Plummer 1986] must be used to obtain the optimal cluster mapping between those at time $t-1$ and those at time t , where the partition-matching is an NP-hard problem. Equation (4), in contrast to evolutionary spectral clustering, has the property that X and Λ are nonnegative. The benefit of this nonnegativity is that the resulting $X\Lambda$ at each timestep *directly* indicates the cluster memberships (as a probabilistic distribution). As a result, the problem of nonidentifiability of parameters is usually not an issue: because of the regularization term $D(X_{t-1}\Lambda_{t-1}\|X_t\Lambda_t)$, the algorithm usually converges to a solution with the optimal mapping between the clusters at time $t-1$ and those at time t automatically chosen.

3. EXTRACTING COMMUNITIES AND THEIR EVOLUTIONS

After obtaining the solution U_t for all timesteps, here are the ways we utilize the solution to analyze communities and their evolutions.

3.1 Community Membership

Assume we have computed the result at time $t-1$, that is, (X_{t-1}, Λ_{t-1}) , and the result at time t , that is, (X_t, Λ_t) . In addition, we define a diagonal matrix D_t , whose diagonal elements are the row sums of $X_t\Lambda_t$, that is, $d_{t,ii} = \sum_j (X_t\Lambda_t)_{ij}$. Then we can see that the i -th row of $D_t^{-1}X_t\Lambda_t$ indicates the soft community memberships of v_i at time t . We illustrate this by using an example shown in Figure 4. Recall that in the bipartite graph at time t (the right side of Figure 4(a)), the weights of edges connecting v_i to c_1 , c_2 , and c_3 represent the joint probability $P(v_i, c_1)$, $P(v_i, c_2)$, and $P(v_i, c_3)$. The D_t^{-1} part normalizes this joint probability to get $P(c_1|v_i)$, $P(c_2|v_i)$, and $P(c_3|v_i)$, that is, the conditional probability that v_i belongs to c_1 , c_2 , and c_3 , respectively. And this conditional probability is exactly the soft community membership we are looking for. Furthermore, we can see that the i -th diagonal element of D_t is the marginal probability of v_i which integrates the soft membership over all communities. Hence D_t provides information about the level of activity of v_i at time t .

3.2 Community Net

The community structure itself, on the other hand, is expressed by $\Lambda_t X_t^T D_t^{-1} X_t \Lambda_t$. For this we again look at the bipartite graph at time t (the right side of Figure 4(a)). Induced from this bipartite graph, $X_t \Lambda X_t^T$ gives a marginal distribution on the subgraph with nodes $\{v_1, \dots, v_6\}$ in order to generate W_t . In a dual fashion, also induced from this bipartite graph, $\Lambda_t X_t^T D_t^{-1} X_t \Lambda_t$ gives a

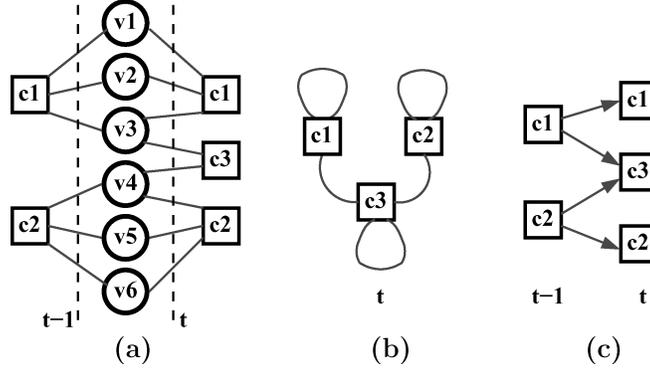


Fig. 4. Schematic illustration of communities and their evolutions: (a) two bipartite graphs at time $t-1$ and time t (merged by v_i 's), (b) the Community Net at time t induced by the bipartite graph at time t , and (c) the Evolution Net from $t-1$ to t induced by the two bipartite graphs.

marginal distribution on the subgraph with nodes $\{c_1, c_2, c_3\}$ (Figure 4(b)) and this is exactly the community structure we are looking for. We call this induced subgraph on the community nodes (i.e., $\{c_1, c_2, c_3\}$) a *Community Net*. Note that to induce the community net, each node v_i participates in all the communities, with different levels. This is more reasonable than traditional methods in which each node can only contribute to a single community.

3.3 Evolution Net

To derive the community evolutions, we align the two bipartite graphs, that at time $t-1$ and that at time t , side by side by merging the corresponding network nodes v_i 's, as illustrated in Figure 4(a). Then a natural definition of community evolution (from $c_{t-1;i}$ at time $t-1$ to $c_{t;j}$ at time t) is the probability of starting from $c_{t-1;i}$, walking through the merged bipartite graphs, and reaching $c_{t;j}$. Such a walking process produces what we call the *Evolution Net* to represent community evolutions, as illustrated in Figure 4(c). A simple derivation shows that $P(c_{t-1;i}, c_{t;j}) = (\Lambda_{t-1} X_{t-1}^T D_t^{-1} X_t \Lambda_t)_{ij}$ and $P(c_{t;j} | c_{t-1;i}) = (X_{t-1}^T D_t^{-1} X_t \Lambda_t)_{ij}$. Again, each node and each edge contribute to the evolution from $c_{t-1;i}$ to $c_{t;j}$. That is, all individuals and all interactions are related to all the community evolutions, with different levels. We believe this is more reasonable than how community evolutions are derived in traditional methods. In tradition methods, usually the intersection and union of *community members* at different time are used alone to compute community evolutions, with a questionable assumption that all members in a community should be treated with identical importance.

4. SOLUTION

In this section, we first give an EM algorithm for solving the MAP problem for Equation (2) and then provide the time analysis for the proposed algorithm.

4.1 An Iterative Algorithm

In our algorithm, we use the following update rules and as proved in Theorem 4.1, the algorithm converges to a solution to the MAP problem in

our model. In the theorem and its proof, for clarity we skip the subscript t and define an auxiliary matrix Y as $Y \doteq X_{t-1}\Lambda_{t-1}$.

THEOREM 4.1. *The following update rules will monotonically increase the logarithm defined in Equation (2) and therefore converge to an (local) optimal solution to the corresponding MAP problem:*

$$x_{ik} \leftarrow x_{ik} \cdot 2 \sum_j \frac{w_{ij} \cdot \lambda_k \cdot x_{jk}}{(X \Lambda X^T)_{ij}} + \nu \cdot y_{ik} \quad (5)$$

then normalize such that $\sum_i x_{ik} = 1, \forall k$

$$\lambda_k \leftarrow \lambda_k \cdot \sum_{ij} \frac{w_{ij} \cdot x_{ik} \cdot x_{jk}}{(X \Lambda X^T)_{ij}} + \nu \cdot \sum_i y_{ik} \quad (6)$$

then normalize such that $\sum_k \lambda_k = 1$.

PROOF. We provide an EM algorithm for solving the MAP problem defined in Equation (2) and show that the EM algorithm results in the update rules in the theorem. In the following discussion, variables with tilde (e.g., $\tilde{\lambda}_k$ and \tilde{x}_{ik}) denote the solutions in the *previous* iteration.

For the E-step, we have have

$$\phi_{ijk} = \tilde{x}_{ik} \tilde{\lambda}_k \tilde{x}_{jk} / (\tilde{X} \tilde{\Lambda} \tilde{X}^T)_{ij}, \quad (7)$$

and for the M-step, we have the expectation as

$$E_\phi L(X, \Lambda | \Psi) = \sum_{ijk} \phi_{ijk} w_{ij} \log(x_{ik} \lambda_k x_{jk}) + \nu \cdot \sum_{ik} y_{ik} \log(x_{ik} \lambda_k). \quad (8)$$

Then by introducing Lagrange multipliers for the constraints $\sum_i x_{ik} = 1$ and $\sum_k \lambda_k = 1$, we can easily show that the update rules in the theorem maximize the expectation in Equation (8). \square

4.2 Time Complexity

We now show the time complexity for each iteration of the updates in Theorem 4.1. The most time-consuming part is to compute $(X \Lambda X^T)_{ij}$ for all $i, j \in \{1, \dots, n\}$. However, it turns out that we do not have to compute $(X \Lambda X^T)_{ij}$ for each pair of (i, j) , thanks to the sparseness of W . In W , the number of nonzero elements is the number of edges in the snapshot graph \mathcal{G}_t , which we denote by ℓ . Then for each nonzero w_{ij} , we compute the corresponding $(X \Lambda X^T)_{ij}$, which takes $O(m)$ time with m being the number of communities. As a result, the total complexity is $O(\ell m)$. If we consider m , the number of communities, to be a constant and if the degree of nodes in the snapshot graph \mathcal{G}_t is bounded by another constant, then the complexity is reduced to $O(n)$, that is, linear in the number of nodes in the snapshot graph.

5. EXTENSIONS

In this section we introduce two extensions to our basic framework in order to handle inserting and removing of individuals and to determine the number of

communities in a dynamic network over time. In addition, we briefly discuss the issue of determining the parameter ν .

5.1 Inserting and Removing Nodes

In real applications, it occurs very often that some new individuals join a dynamic network (e.g., a new author in a paper coauthorship network) and existing ones leave (e.g., a blogger who stops blogging). We provide the following heuristic techniques in our algorithm to handle such inserting and removing of nodes.

Assume that at time t , out of the n nodes in the network, n_1 existing nodes are removed from and n_2 new nodes are inserted into the network. We first handle the n_1 removed nodes by removing the corresponding n_1 rows from Y in Equations (5) and (6) to get Y' . Next, we scale Y' to get Y'' so that Y'' is a valid joint distribution, i.e., $Y'' = Y' / \sum_{ij} y'_{ij}$. The basic idea behind this heuristic is that we assume the n_1 nodes are randomly selected, independent of their community membership. Under such an assumption, Y'' is a conditional distribution, conditioning on the remaining $n - n_1$ nodes in the network. To add the n_2 nodes, we pad n_2 rows of zeros to Y'' to get \hat{Y} . This heuristic is actually equivalent to assuming that these n_2 nodes have already existed at time $t-1$ but as isolated nodes.

5.2 Changing Community Numbers

So far we have assumed that the number of communities, m , is given beforehand by the user. However, such an assumption will limit the scope of application of our framework. In this subsection we try to answer two questions: how to automatically determine the number of communities at a given time t and how to revise our framework to allow the number of communities to change in different timesteps.

5.2.1 Soft Modularity. Newman and Girvan [2004] introduced an elegant concept, the *modularity* Q , to measure the goodness of a community partition \mathcal{P}_m where Q is defined as

$$Q(\mathcal{P}_m) = \sum_{k=1}^m \left[\frac{\mathcal{A}(V_k, V_k)}{\mathcal{A}(V, V)} - \left(\frac{\mathcal{A}(V_k, V)}{\mathcal{A}(V, V)} \right)^2 \right] \quad (9)$$

with $\mathcal{A}(V_p, V_q) = \sum_{i \in V_p, j \in V_q} w_{ij}$. Basically, Q measures the deviation between the chance for edges among communities to be generated due to the community structure and the chance for the edges to be generated randomly. Extensive experimental results have demonstrated that Q is an effective measure for the community quality, where a maximal Q is a good indicator of the best community structure and therefore the best community number m [Newman and Girvan 2004; White and Smyth 2005].

Here we extend the concept of modularity to handle soft membership by defining a *Soft Modularity* Q_s :

$$Q_s = \text{Tr} \left[(D^{-1}X\Lambda)^T W (D^{-1}X\Lambda) \right] - \bar{\mathbf{1}}^T W^T (D^{-1}X\Lambda) (D^{-1}X\Lambda)^T W \bar{\mathbf{1}}, \quad (10)$$

where $\bar{\mathbf{1}}$ is a vector whose elements are all ones. Q_s has the following nice property.

THEOREM 5.1. *The Q_s defined in Equation (10) has the same probabilistic interpretation as the Q defined in Equation (9), but in the context of soft community membership. In addition, Q_s is a generalized modularity in that Q_s is identical to Q when $D^{-1}X\Lambda$ becomes a hard community membership (i.e., each row of $D^{-1}X\Lambda$ has one 1 and $m-1$ zeros).*

PROOF. In the standard Q formula Equation (9), the first term $\frac{A(V_k, V_k)}{A(V, V)}$ is the empirical probability that a randomly selected edge has both ends in community k . For our case, this empirical probability should be $\sum_{i,j} w_{ij} P(k|i)P(k|j)$. For the second term in Equation (9), $\frac{A(V_k, V)}{A(V, V)}$ is the empirical probability that a randomly selected edge is related to (i.e., has at least one end in) community k . For our case, this empirical probability should be $\sum_i P(k|i) \sum_j w_{ij}$. By summing these two terms over all k 's and noticing that $P(k|i) = (D^{-1}X\Lambda)_{ik}$, we get formula Equation (10). In addition, it is straightforward to verify that Q_s is equal to Q when $D^{-1}X\Lambda$ becomes a 0/1 indicator matrix. \square

So to detect the best community number m at time t , we run our algorithm for a range of candidates for m and pick the best one determined by Q_s .

5.2.2 Extended Formulas. If we allow different community numbers at time t and $t-1$, then we have to revise Equation (2) accordingly because in Equation (2), the term $\sum_{ik} v \cdot (X_{t-1}\Lambda_{t-1})_{ik} \log(X_t\Lambda_t)_{ik}$ requires $X_{t-1}\Lambda_{t-1}$ (the community structure at $t-1$) and $X_t\Lambda_t$ (the community structure at t) to have the same number of columns and therefore the same number of communities. To solve this issue, we revise the logarithm to be

$$L(U_t) = \sum_{ij} (W_t)_{ij} \log (X_t\Lambda_t X_t^T)_{ij} + \sum_{ij} v \cdot (X_{t-1}\Lambda_{t-1} X_{t-1}^T)_{ij} \log (X_t\Lambda_t X_t^T)_{ij}. \quad (11)$$

The basic idea is that when the community numbers are different at time t and $t-1$, instead of regularizing the community structure itself, we regularize the marginal distribution induced by the community structure at time t (i.e., $X_t\Lambda_t X_t^T$) so that it is not too far away from that at time $t-1$ (i.e., $X_{t-1}\Lambda_{t-1} X_{t-1}^T$).

For this new problem, we use the following update rules and as proved in Theorem 5.2, the algorithm converges to a solution to the new MAP problem. In the theorem and its proof, for clarity we skip the subscript t and define an auxiliary matrix Z as $Z \doteq X_{t-1}\Lambda_{t-1} X_{t-1}^T$.

THEOREM 5.2. *The following update rules will monotonically increase the logarithm defined in Equation (11) and therefore converge to an optimal solution*

to the corresponding MAP problem:

$$x_{ik} \leftarrow x_{ik} \cdot \sum_j \frac{(w_{ij} + v \cdot z_{ij}) \cdot \lambda_k \cdot x_{jk}}{(X \Lambda X^T)_{ij}} \quad (12)$$

then normalize such that $\sum_i x_{ik} = 1, \forall k$

$$\lambda_k \leftarrow \lambda_k \cdot \sum_{ij} \frac{(w_{ij} + v \cdot z_{ij}) \cdot x_{ik} \cdot x_{jk}}{(X \Lambda X^T)_{ij}} \quad (13)$$

then normalize such that $\sum_k \lambda_k = 1$.

PROOF. We again provide an EM algorithm for solving the MAP problem defined in Equation (11) and show that the EM algorithm results in the update rules in the theorem. Notice that we have

$$\sum_{ij} w_{ij} \log (X_t \Lambda_t X_t^T)_{ij} + \sum_{ij} v \cdot z_{ij} \log (X_t \Lambda_t X_t^T)_{ij} = \sum_{ij} (w_{ij} + v \cdot z_{ij}) \log (X_t \Lambda_t X_t^T)_{ij}.$$

For the E-step, we have have

$$\phi_{ijk} = \tilde{x}_{ik} \tilde{\lambda}_k \tilde{x}_{jk} / (\tilde{X} \tilde{\Lambda} \tilde{X}^T)_{ij}, \quad (14)$$

and for the M-step, we have the expectation as

$$E_{\Phi} L(X, \Lambda | \Psi) = \sum_{ijk} \phi_{ijk} \cdot (w_{ij} + v \cdot z_{ij}) \log(x_{ik} \lambda_k x_{jk}). \quad (15)$$

Then by introducing Lagrange multipliers for the constraints $\sum_i x_{ik} = 1$ and $\sum_k \lambda_k = 1$, we can easily show that the update rules in the theorem maximize the expectation in Equation (15). \square

5.3 Determining v

How to determine the v in Equation (2) (or equivalently, the α in Equation (4)) is a challenging issue. When the ground truth is available, standard validation procedures can be used to select an optimal v . However, in many cases there is no ground truth and the community extraction performance depends on the user's subjective preference (e.g., to what level the user believes the prior distribution). In this respect, through the parameter v , our algorithms provide the user a mechanism to push the community extraction results toward his or her preferred outcomes. The problems of whether a "correct" v exists and how to automatically find the *best* v when there is no ground truth are beyond the scope of this paper.

6. EXPERIMENTAL STUDIES

In this section, we use several synthetic datasets, a blog dataset, and a paper coauthorship dataset to study the performance of our *FacetNet* framework.

6.1 Synthetic Datasets

6.1.1 Synthetic Dataset # 1. We start with the first synthetic dataset, which is a *static* network, to illustrate some good properties of our framework. This

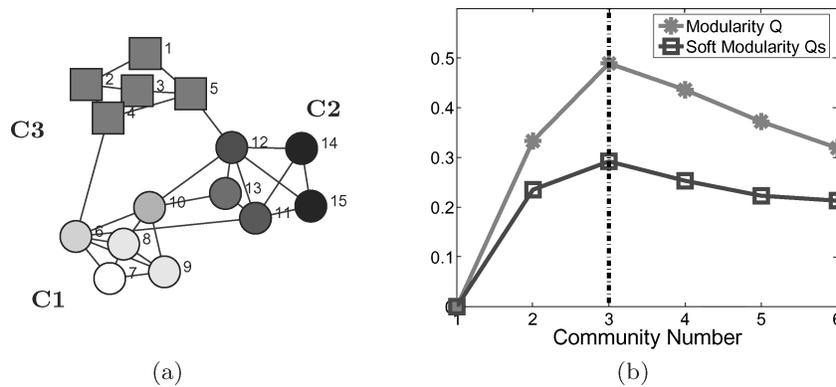


Fig. 5. The first synthetic dataset: (a) the static network, (b) modularity and soft modularity under different community numbers.

dataset was first studied in White and Smyth [2005] and is shown in Figure(5a). The network contains 15 nodes which roughly form 3 communities—C1, C2, and C3—where edges tend to occur between nodes in the same community. We first check our soft modularity measure. We apply our algorithm to the network with various community numbers m and the resulting Q_s values are plotted in Figure(5b). In addition, in Figure(5b) we also show the modularity values Q that are reported in [White and Smyth 2005]. As can be seen from the plot, both Q_s and Q show distinct peaks when $m = 3$, which corresponds to the correct community number.

Next, after our algorithm correctly partitions the 15 nodes into three communities, we illustrate the soft community membership by studying two communities among the three—C1 = {6, 7, 8, 9, 10} and C2 = {11, 12, 13, 14, 15}. In Figure(5a), we use the same circle shape to represent these 10 nodes but use different gray levels to indicate their community membership—we use white color to illustrate the level that a node belongs to C1 and dark color to show the level that a node belongs to C2. As can be seen, while node 7, node 14, and node 15 have very clear community memberships, node 10 and node 13, who are on the boundary between C1 and C2, have rather fuzzy membership. That is, our algorithm is capable of assigning meaningful soft membership to a node to indicate to which level the node belongs to a certain community.

6.1.2 Synthetic Dataset # 2. The second dataset is generated according to the description in Newman and Girvan [2004]. This dataset contains 128 nodes, which are divided into 4 communities of 32 nodes each. We generate data for 50 consecutive timesteps. In each timestep from 2 to 50, dynamics are introduced in the following way: from each community we randomly select certain number of members to leave their original community and to join randomly the other three communities. Edges are added randomly with a higher probability p_{in} for within-community edges and a lower probability p_{out} for between-community edges. However, the average degree for the nodes is set to 20. As a result, a single parameter z , which represents the mean number of edges from a node to nodes in other communities, is enough to describe the data. In this study, we

investigate two values for z : $z = 5$ (which results in $p_{in} = 0.05$ and $p_{out} = 0.16$) where the community structure is clearer, and $z = 6$ (which results in $p_{in} = 0.06$ and $p_{out} = 0.12$) where the community structure is fuzzier. Furthermore, for each fixed z , we investigate two cases: the case where the community structure is relatively stable over time (with 10% members change their communities at each timestep) and the case where the community structures change more dramatically over time (with 30% members change their communities at each timestep).

We compare our *FacetNet* framework with 3 baseline algorithms. The first baseline, which we call *EvolSpec*, is the evolutionary spectral clustering algorithm proposed in Chi et al. [2007]. Because *EvolSpec* is an evolutionary version of the Normalized Cut (*NCut*) algorithm in Shi and Malik [2000], we take *NCut* as our second baseline. Similarly, *FacetNet* is an evolutionary version of the soft clustering method (*SNMF*) in Yu et al. [2005], we take *SNMF* as our third baseline. Notice that *FacetNet* and *EvolSpec* are evolutionary algorithms whereas *NCut* and *SNMF* are not—*NCut* and *SNMF* work on each snapshot graph independently of other snapshot graphs. In addition, to make the results comparable, for *FacetNet* and *SNMF* we convert the soft membership into 0/1 indicators by assigning each node to the community it most likely belongs to. Furthermore, in all the experiments, for *FacetNet* and *EvolSpec* we set α to be 0.8 (i.e., $\nu = 1/5$).

For the performance metric, because we have the ground truth for the community memberships at each timestep, we directly study the error rate of the community structures obtained by different algorithms. The error rate is computed in the following way. The community structure computed by a given algorithm can be represented by a 128-by-4 indicator matrix Z , where the i -th row of Z indicates the community membership of the i -th node (i.e., if the i -th node belongs to the k -th community, then $z_{ik} = 1$ and $z_{ik'} = 0$ for $k' \neq k$). A similar indicator matrix G can be built for the ground truth. Then the error rate is represented by the norm $\|ZZ^T - GG^T\|_F$, which measures the distance between the community structure represented by Z and that represented by G (see Bach and Jordan [2006] for a detailed discussion). All the experiments are repeated 50 times with different random seeds and the average results are reported.

Figure 6 gives the performance on the two datasets with $z = 5$. Figure 6(a) shows the adjacency matrix of the snapshot graph at the first timestep, where the nodes are ordered according to their truth community memberships so that the true community structure can be seen. As can be seen from the figure, the community structure is relatively easy to detect when $z = 5$. Figure 6(b) and Figure 6(c) show the error rates over time for the four algorithms for the cases of 10% and 30% nodes changing their communities at each timestep, respectively. From the figures we have the following observations. First, on these datasets, among the two nonevolutionary algorithms (*NCut* and *SNMF*), *NCut* outperforms *SNMF*. Second, our algorithm *FacetNet* clearly outperforms the evolutionary spectral clustering algorithm *EvolSpec* in both cases. Third, when the community structure changes more dramatically over time, the benefit of the evolutionary spectral clustering disappears (in Figure 6(c), *NCut* and *EvolSpec*

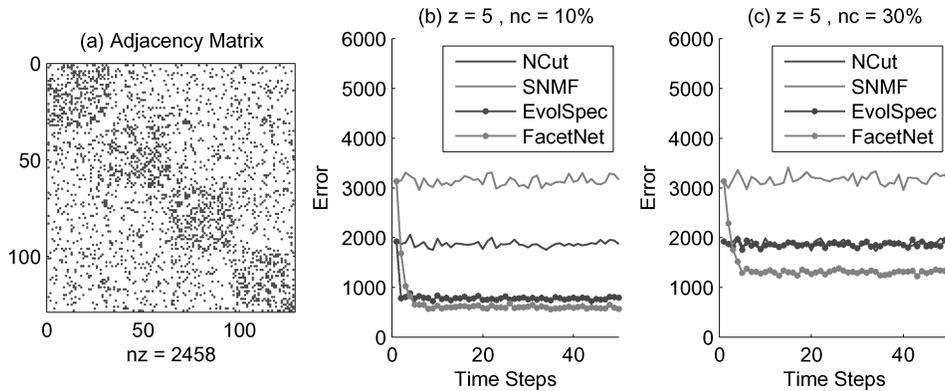


Fig. 6. Error with respect to the ground truth over 50 timesteps on the datasets with $z = 5$: (a) the adjacency matrix of the first timestep, (b) 10% nodes change their community at each timestep, (c) 30% nodes change their community at each timestep.

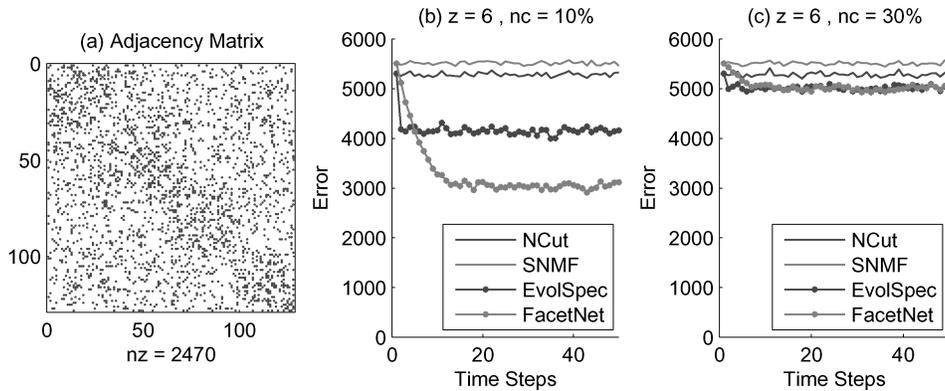


Fig. 7. Error with respect to the ground truth over 50 timesteps on the datasets with $z = 6$: (a) the adjacency matrix of the first timestep, (b) 10% nodes change their community at each timestep, (c) 30% nodes change their community at each timestep.

have almost identical error rates) while the benefit of our *FacetNet* algorithm is still clearly visible.

Figure 7 gives the performance on the two datasets with $z = 6$. Figure 7(a) shows the adjacency matrix of the snapshot graph at the first timestep, from which we can see that the community structure is more difficult to detect. Figure 7(b) and Figure 7(c) show the error rates over time for the four algorithms for the cases of 10% and 30% nodes changing their communities at each timestep, respectively. From the results we can see that on these two datasets, *FacetNet* still outperforms other baseline algorithms although when the community structure changes dramatically (as shown in Figure 7(c)), the improvement of *FacetNet* becomes rather marginal.

We study the performance over different values of α . Figure 8 reports the mean error rates (averaging over timesteps 2-50) of two evolutionary algorithms, *EvolSpec* and *FacetNet*, with error bars represent one standard

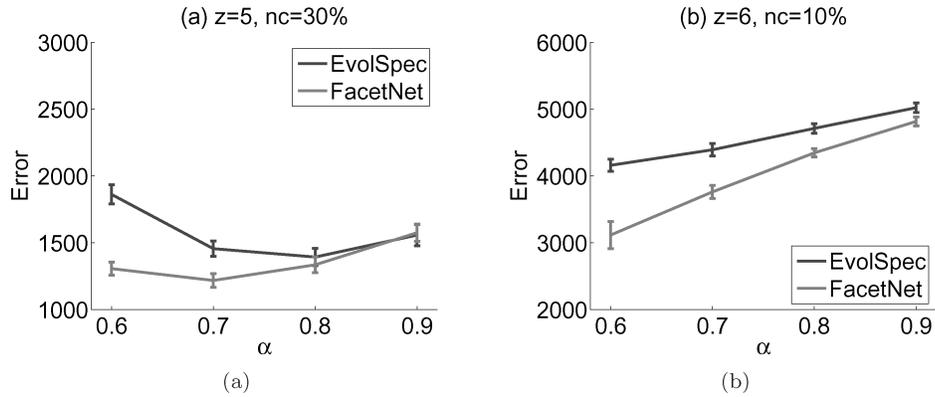


Fig. 8. Mean error rates (averaging over timesteps 2-50) of two evolutionary algorithms, with error bars represent one standard deviation: (a) on datasets with $z = 5$, 30% nodes change their community every timestep, (b) on datasets with $z = 6$, 10% nodes change their community every timestep.

deviation. Figure 8(a) and (b) show results on the same datasets as in Figure 6(c) and Figure 7(b). As can be seen, first, when decreasing α (i.e., emphasize more on temporal smoothness), the improvement of *FacetNet* against *EvolSpec* become more significant. Second, as shown in Figure 8(a), the mean error rates do not necessarily decrease with more temporal smoothness, i.e. smaller value of α . This is expected because the importance of new observations are neglected. The results also suggest *FacetNet* can better tradeoff between new observation and temporal smoothness.

6.1.3 Synthetic Dataset # 3. Next, we study the time performance of *FacetNet*. We repeat the previous experiment over a family of networks of various sizes (node numbers). In Figure 9(a) we show the average running time per iteration of our algorithm on networks with different sizes (the algorithm is implemented in Matlab). In Figure 9(b) we showed the number of iteration needed for convergence when the convergence criterion is that the relative change in log likelihood between two consecutive iterations is below a threshold of $1e-5$. As can be seen, first, the running time per iteration scales linearly with the size of network, which validates our theoretical analysis in Section 4.2; second, the number of iteration needed for convergence is insensitive to the network size, which implies that the overall running time of our algorithm scales linearly to the size of network.

6.2 NEC Blog Dataset

The blog data was collected by an NEC in-house blog crawler. Given seeds of manually picked highly ranked blogs, the crawler discovered blogs that are densely connected with the seeds, resulting in an expanded set of blogs that communicate with each other. The crawler then continued monitoring for new entries over a long time period. This NEC blog data set contains 148,681 entry-to-entry links among 407 blogs crawled during 12 consecutive month (a timestep

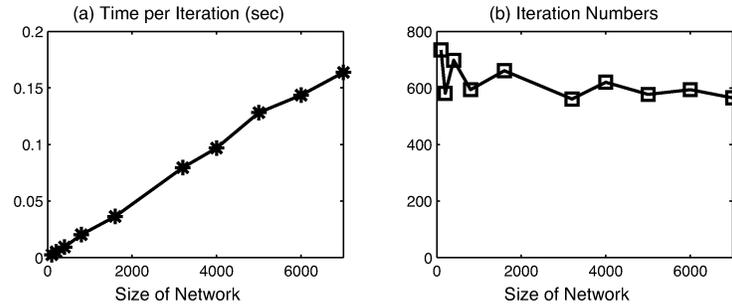


Fig. 9. Running time for networks of different sizes: (a) running time (sec) per each iteration, (b) number of iterations until converge.

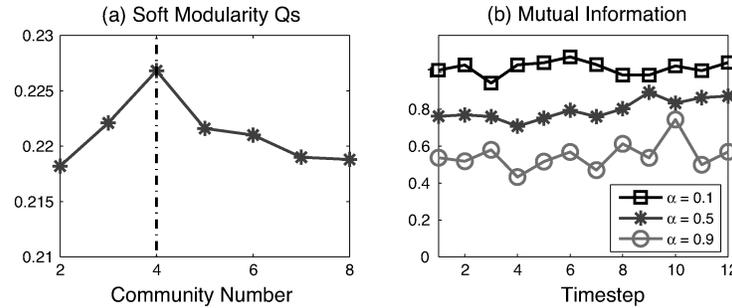


Fig. 10. (a) Soft modularity and (b) mutual information under different α for the NEC Blog dataset.

is one month), between August 2005 and September 2006. Following [Ning et al. 2007], we define W by $w_{ij} = \tilde{w}_{ij} / \sum_{p,q} \tilde{w}_{pq}$ where $\tilde{w}_{ii} = 1$, $\tilde{w}_{ij} = \exp(-1/(\gamma \cdot l_{ij}))$ if $e_{ij} \in \mathcal{E}_t$, and otherwise $\tilde{w}_{ij} = 0$. In the formula, l_{ij} is the edge weight (e.g., # of links) of e_{ij} and γ , which is set to 0.2, is a parameter to control marginal effect when l_{ij} is increased.

We start with analyzing the overall picture of the dataset. We first aggregate all the edges over all timesteps into a single network and apply our algorithm to compute the soft modularity score Q_s under different community numbers. As can be seen in Figure 10(a), a clear peak shows when the community number is 4. We draw the aggregated graph in Figure 11, according to the main community each blog most likely belongs to. In addition, in Table I we list the top keywords, measured by the *tf-idf* score, that occur in the entries of these four communities. It seems that $C1$ focuses on technology, $C2$ on politics, $C3$ on international entertainment, and $C4$ on digital libraries.

Next, we analyze the blog data as a dynamic network. After studying the content of the blogs, we find that the above four communities stay rather stable over all the timesteps. This effect is partially due to the way these blogs are selected by our focused crawler—our crawler chose to crawl a densely connected subgraph of the blogosphere where each node in the subgraph has large number of links and high level of interaction intensities. Therefore, most of the selected blogs belong to some well-known bloggers and they seldom move around between communities. We apply our *FacetNet* algorithm on the data with different

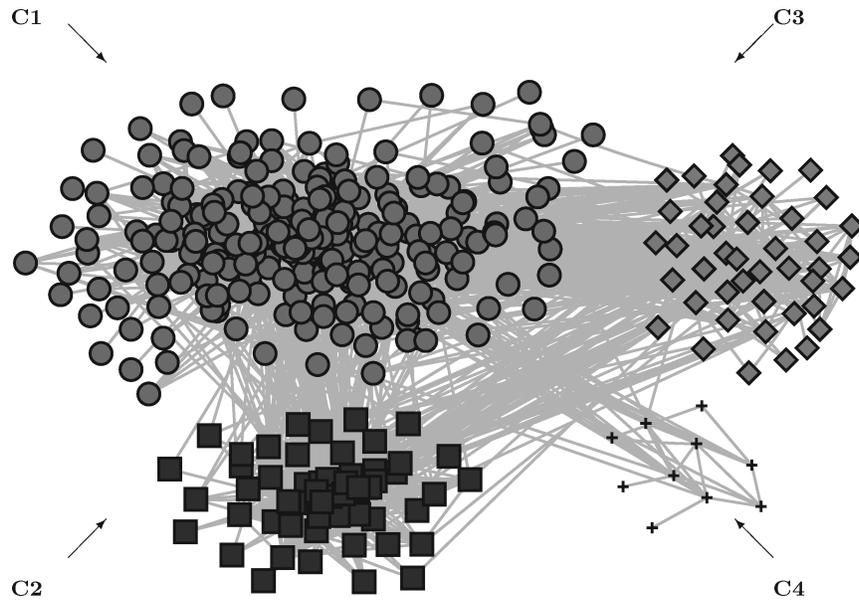


Fig. 11. Four communities in the NEC Blog dataset.

Table I. Top Keywords Among the Four Communities in the NEC Blog Dataset, Sorted by the *tf-idf* Score

C1	adsense, beta, skype, firefox, msn, rss, aol, yahoo, google, ebay, desktop, wordpress, voip, feeds, myspace, podcasting, technorati, search, engine, browser, ads, gmail, windows, os, developer, venture, marketing, apple, podcasts, developers, engines, mac, publishers, ceo, linux
C2	gop, uranium, hezbollah, democrats, rove, cia, republicans, saddam, qaeda, tax, republican, iraqi, roberts, bush, clinton, iraq, senate, troops, terrorists, administration, terrorist, wilson, conservative, taxes, liberal, intelligence, israel, terror, iran, weapons, war, soldiers
C3	shanghai, robots, installation, japan, japanese, architecture, art, chinese, china, saudi, phones, filed, mobile, games, korea, rfid, sex, green, camera, sound, cell, body, africa, phone, entertainment, film, gay, india, fuel, archive, design, elections, flash, device, water, wireless, south
C4	library, learning, digital, resources, collection, conference, staff, communities, students, session, books, database, access, survey, university, science, canada, myspace, articles, education, technologies, knowledge, filed, virtual, tools, research, david, learn, services, flickr, computers

α . Because there is no ground truth in this dataset, we instead use the communities obtained from the aggregated graph (Figure 11) as a reference, and we compute the mutual information between the extracted communities at each timestep and the four communities shown in Figure 11. We refer interested readers to Xu et al. [2003] for detailed definition of the mutual information between two partitions. Basically, high mutual information between two partitions indicates that the two partitions are similar to each other.

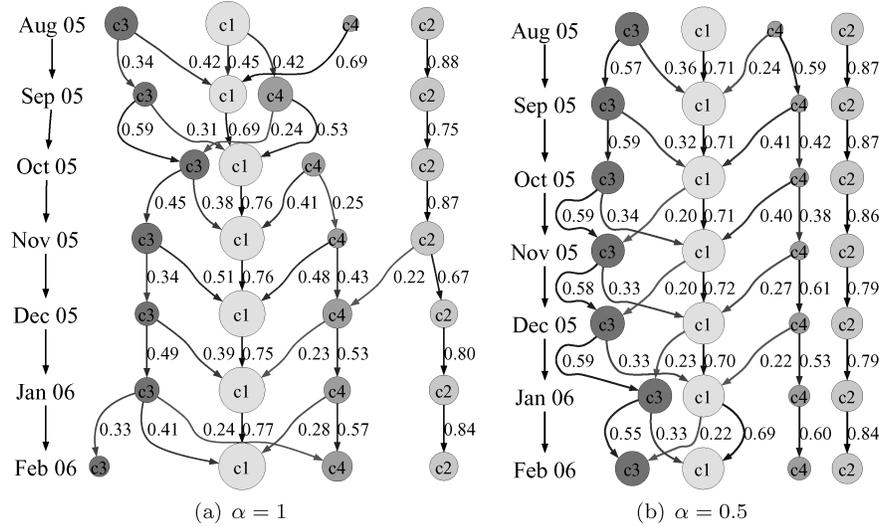


Fig. 12. The Evolution Net of the NEC Blog dataset when (a) $\alpha = 1$ and (b) $\alpha = 0.5$.

Figure 10(b) shows the mutual information results under $\alpha = 0.1, 0.5$, and 0.9 . As can be seen, as α increases, our algorithm emphasizes less on the temporal smoothness and as a result, the community structure has higher variation over time. In addition, as α increases, the communities at each timestep deviate further from the communities obtained from the aggregated data. These results on one hand justify our arguments in the introduction section and on the other hand demonstrate that our *FacetNet* framework is capable of controlling the tradeoff between the snapshot cost and the temporal cost in the cost function Equation (3).

Figure 12 shows the Evolution Net derived from our framework when $\alpha = 1$ and 0.5 ($\alpha = 1$ means no temporal smoothness is considered). In the Evolution Net, the size of a node is proportional to λ_k and it represents the size of the corresponding community. The edge label indicates the probability of a transition from the source community at $t-1$ to the target community at t . (To avoid clutter, we did not show edges with probabilities less than 0.2). From Figure 12(a) we see that when no temporal smoothness is considered, $C4$ disappeared at the second timestep (Sep 2005) and reappeared in the third timestep (Oct 2005). However, by carefully examining the original data, we did not find any significant events so support such changes. Therefore, we conjecture that these changes are due to the data noises at the second timestep, which triggered the algorithm to split $C1$ into two communities and merge $C4$ to one of them. In comparison, as can be seen from Figure 12(b), when there is a temporal smoothness term, the four communities remain relatively stable. That is, although there exist transitions among different types of communities, the majority of transitions are between communities of the same type. These results demonstrate that the *FacetNet* framework is more robust to data noise.

From the Evolution Net, we can also obtain some other observations. For example, the political community $C2$ is rather isolated from the rest communities

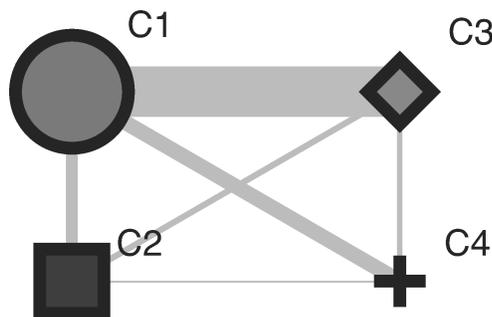


Fig. 13. The Community Net for the NEC Blog dataset in September 2005.

over all the time. In comparison, both $C3$ and $C4$ interacts with $C1$ heavily. In addition, in Figure 13 we show the Community Net at an arbitrary timestep (Sep 2005). In the Community Net, the node sizes are proportional to $\lambda_{t;k}$ and the edge weights are proportional to the corresponding entries in $\Lambda_t X_t^T D_t^{-1} X_t \Lambda_t$ (self-loops are not shown). We can see that this Community Net is a good synopsis of the aggregated network in Figure 11.

6.3 DBLP Coauthorship Dataset

The DBLP coauthorship dataset is described in Asur et al. [2007]. It contains the coauthorship information among the papers in 28 conferences over 10 years (1997–2006). The 28 conferences span three main areas—data mining, database, and artificial intelligence. By removing authors with too few papers or too few coauthors in the dataset, we select 1,071 authors from the original dataset and build snapshot graphs with these 1,071 authors as nodes. The aggregated coauthorship graph among these authors is given in Figure 14. For analyzing community evolutions, we aggregate data in every two years into one timestep and therefore we have 5 timesteps in total (and therefore 5 snapshot graphs) for the dynamic network. In the experiments, we fixed the community number m to be 3, which corresponds to the number of main research areas among the 28 conferences.

In the first experiment, we apply the *FacetNet* algorithm on the aggregated coauthorship graph and in Table II we list the extracted top authors in the 3 communities, where the rank is determined by the value x_{ik} , that is, $p_{k \rightarrow i}$. Recall that $p_{k \rightarrow i}$ indicates to what level the k -th community involves the i -th node. So from our framework, we can directly infer who are the important members in each community. However, notice that by *important*, we are not judging the quality or quantity of papers by an author. Instead, in our framework the importance of a node in a community is determined by its contribution to the community structure.

In the second experiment, we apply our *FacetNet* algorithm on the snapshot graphs over 5 timesteps (with $\alpha = 0.9$ or equivalently $\nu = 1/9$). Using the solution of our algorithm, we examine some individual authors listed in Table II and see how their community membership change over time. It turns out that

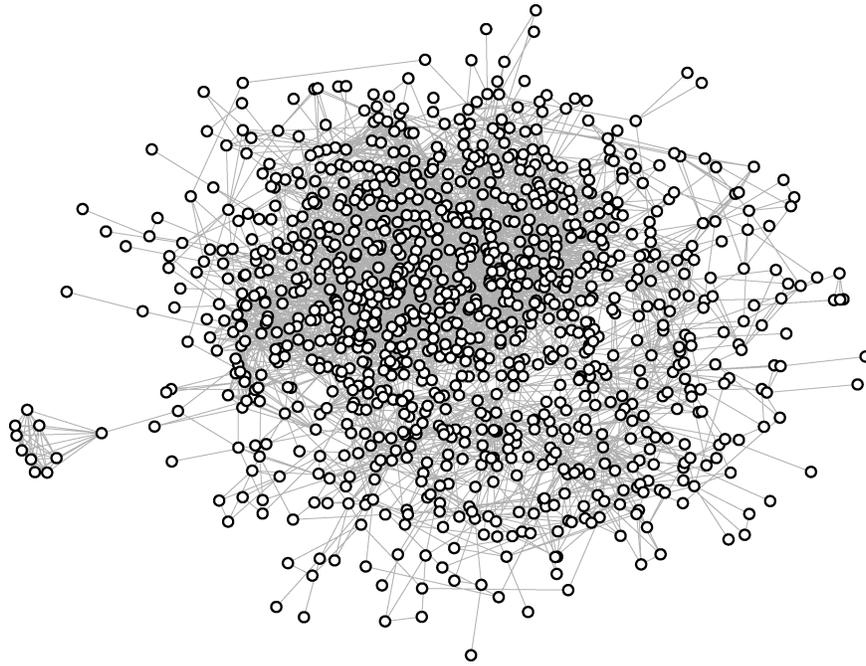


Fig. 14. The aggregated coauthorship graph among the 1,071 authors in the DBLP dataset.

the top authors in the Artificial Intelligence community remain very focused over all timesteps—most of them have $x_{ik} = 0$ when k does not correspond to the AI community. In contrast, a couple of authors in the Data Mining (DM) community and the Database (DB) community switched their community membership during the 10 year period covered by the dataset. For one example, in Figure 15 we demonstrate one top author (Philip S. Yu) in the DM community whose community membership remains stable over all the timesteps and another top author (Laks V. S. Lakshmanan) whose community membership varies very much over the 5 timesteps. In the figure, each compass indicates a pair $(p_{k_1 \rightarrow i}, p_{k_2 \rightarrow i})$ where k_1 and k_2 correspond to the DB and the DM communities, respectively. So in a compass, a vertical arrow (which has a large projection on the y -axis) indicates a large value of the community membership in the DM community and a horizontal arrow (which has a large projection on the x -axis) indicates a large value of the community membership in the DB community. From Figure 15 we can see that the first author consistently played an important role mainly in the DM community, whereas the second author had a varying role in both the two communities. By looking at the publication record of this second author we can see that he had published on the topic of association rules (DM) during the periods of $t = 1997$ – 1998 and $t = 2001$ – 2002 . However, during the periods of $t = 1999$ – 2000 and $t = 2003$ – 2006 , this author co-authored a large number of papers in top DB conferences (e.g., 15 papers in SIGMOD and 12 in VLDB), which is reflected by his high values of membership in the DB community (large projections on the x -axis) during these periods.

Table II. Top Members in the Three Communities, Sorted by x_{ik} , i.e., $p_{k \rightarrow i}$

Data Mining	Database	Artificial Intelligence
Philip S. Yu	Divesh Srivastava	Daphne Koller
Jiawei Han	Nick Koudas	Sebastian Thrun
Jian Pei	Divyakant Agrawal	Maurizio Lenzerini
Wei Wang	Hans-Peter Kriegel	Giuseppe De Giacomo
Haixun Wang	Surajit Chaudhuri	Diego Calvanese
Beng Chin Ooi	Amr El Abbadi	Bernhard Scholkopf
Kian-Lee Tan	H. V. Jagadish	Michael I. Jordan
Charu C. Aggarwal	Rajeev Rastogi	Andrew Y. Ng
Jiong Yang	Minos N. Garofalakis	Craig A. Knoblock
Hongjun Lu	S. Muthukrishnan	Daniela Florescu
Mong-Li Lee	Jennifer Widom	Satinder P. Singh
Jeffrey Xu Yu	Rakesh Agrawal	Wei-Ying Ma
Tok Wang Ling	Elke A. Rundensteiner	Thomas S. Huang
Anthony K. H. Tung	Jeffrey F. Naughton	Didier Dubois
Dimitris Papadias	Rajeev Motwani	Brendan J. Frey
Wynne Hsu	Flip Korn	Steven Minton
Bing Liu	Michael J. Franklin	Henri Prade
Ke Wang	Johannes Gehrke	Heung-Yeung Shum
Yufei Tao	Hector Garcia-Molina	Michael J. Kearns
Xifeng Yan	Vivek R. Narasayya	Nir Friedman
Wei Fan	Raghu Ramakrishnan	Craig Boutilier
Laks V. S. Lakshmanan	Laks V. S. Lakshmanan	Dieter Fox
Sourav S. Bhowmick	Walid G. Aref	Riccardo Rosati
Guozhu Dong	Christos Faloutsos	Alex J. Smola
Jianyong Wang	Sihem Amer-Yahia	Alon Y. Levy

As another example, in Figure 16 we plot the community memberships over the 5 timesteps for two authors in the DB community, namely Divesh Srivastava and Christos Faloutsos. The research of the first author consistently focused on traditional database only, which is reflected by his homogeneous community membership over time. The community membership of the second author, however, evolves distinctively over time. In the early timesteps, this author mainly contributes to the DB community. However, in later timesteps, the contributions of this author are totally switched to the DM community. The biography of author from the author’s homepage verifies that this switch has indeed happened. In addition, from this author’s publication record in this DBLP dataset we can see an evolution from almost exclusively focusing on DB (SIGMOD, VLDB, etc.) to almost exclusively focusing on DM (SIGKDD, ICDM, etc.) over the 10-year period. Because the conference venue information is not used in our algorithm, these anecdotic facts validate the effectiveness of our algorithms on capturing individual community memberships and their evolutions over time.

7. CONCLUSION AND FUTURE DIRECTIONS

The analysis of communities and their evolutions in dynamic temporal networks is a challenging research problem with broad applications. In this article, we proposed a framework, *FacetNet*, that combines the task of community extraction and the task of evolution extraction in a unified process. To the best of our knowledge, our framework is the first probabilistic generative model that

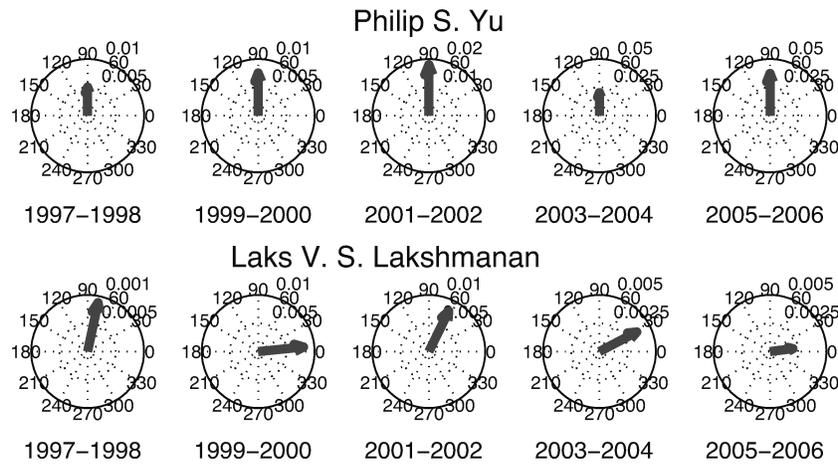


Fig. 15. The evolution of community memberships for two top authors in the Data Mining community.

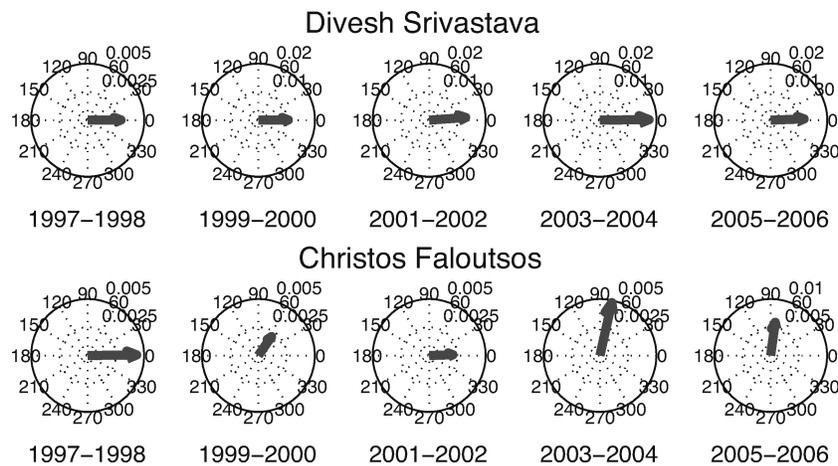


Fig. 16. The evolution of community memberships for two top authors in the Database community.

simultaneously analyzes communities and their evolutions. We also proved the connection between our probabilistic framework and a nonnegative version of the evolutionary spectral clustering algorithm. The results obtained from our model allow us to assign soft community memberships to individual nodes, to analyze the strength of ties among various communities, to study how the affiliations of an individual to different communities change over time, as well as to reveal how communities evolve over time. Extensive experimental studies demonstrated that by using our *FacetNet* framework, we are able to extract more accurate communities, to capture more robust communities, to reveal the community membership evolution for an individual or the evolutions of the communities, as well as to discover many interesting insights in dynamic networks that are not directly obtainable from existing methods.

There are some open issues: (1) Our community factorization depends on the number of communities, m . In this work we propose soft modularity for selecting an appropriate m and show meaningful results in our experiments. However, this approach requires some domain knowledge to determine a range of candidates for m ; otherwise, evaluating the soft modularity over all different m 's can be expensive. (2) The community factorization results relies on human interpretation, which is also expensive. We will continue investigating methods for efficiently validating social mining results.

As part of future work, we plan to extend our framework in two directions. First, our current model only considered the link information. In many applications, the content information (e.g., the contents of blog entries and the abstracts of papers) is also very important. We are investigating how to incorporate content information into our framework. Second, so far we only use our model to *explain* the observed data. To extend our model to *predict* future behaviors of individuals in a dynamic social network is also one of our future directions.

APPENDIX

We prove Theorem 2.1 as follows.

PROOF. When $\sum_{ij}(W_t)_{ij}$ remains a constant over time, without loss of generality we can assume that $\sum_{ij}(W_t)_{ij} = 1$. Then a necessary condition for Λ_t to minimize the cost in Eq. (4) is that $\sum_k \lambda_{t;k} = \sum_{ij}(W_t)_{ij} = 1$. So we have

$$\begin{aligned} cost &= \alpha \cdot D(W_t \| X_t \Lambda_t X_t^T) + (1 - \alpha) \cdot D(X_{t-1} \Lambda_{t-1} \| X_t \Lambda_t) \\ &= \alpha \cdot \left[\sum_{ij} (W_t)_{ij} \log(W_t)_{ij} - \sum_{ij} (W_t)_{ij} \log(X_t \Lambda_t X_t^T)_{ij} \right] \\ &\quad + (1 - \alpha) \cdot \left[\sum_{ik} (X_{t-1} \Lambda_{t-1})_{ik} \log(X_{t-1} \Lambda_{t-1})_{ik} \right. \\ &\quad \left. - \sum_{ik} (X_{t-1} \Lambda_{t-1})_{ik} \log(X_t \Lambda_t)_{ik} \right]. \end{aligned}$$

So minimizing the cost is equivalent to maximizing

$$\alpha \left[\sum_{ij} (W_t)_{ij} \log(X_t \Lambda_t X_t^T)_{ij} + \frac{1 - \alpha}{\alpha} \sum_{ik} (X_{t-1} \Lambda_{t-1})_{ik} \log(X_t \Lambda_t)_{ik} \right].$$

By setting ν to $(1 - \alpha)/\alpha$, we prove the theorem. \square

ACKNOWLEDGMENTS

We thank Kai Yu for providing us with the SNMF source code and for the helpful discussion; Junichi Tatemura and Koji Hino for helping to prepare the blog dataset; Sitaram Asur and Srinivasan Parthasarathy for providing us with the DBLP dataset.

REFERENCES

- ASUR, S., PARTHASARATHY, S., AND UCAR, D. 2007. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- BACH, F. R. AND JORDAN, M. I. 2006. Learning spectral clustering, with application to speech separation. *J. Mach. Learn. Resea.* 7.
- CHAKRABARTI, D., KUMAR, R., AND TOMKINS, A. 2006. Evolutionary clustering. In *Proceedings of the 12th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- CHI, Y., SONG, X., ZHOU, D., HINO, K., AND TSENG, B. L. 2007. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- CHUNG, F. R. K. 1997. *Spectral Graph Theory*. American Mathematical Society.
- DHILLON, I. S., GUAN, Y., AND KULIS, B. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the 10th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- FLAKE, G., LAWRENCE, S., AND GILES, C. 2000. Efficient identification of web communities. In *Proceedings of the 6th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- KLEINBERG, J. M. 1999. Authoritative sources in a hyperlinked environment. *J. ACM* 46, 5.
- KUMAR, R., NOVAK, J., RAGHAVAN, P., AND TOMKINS, A. 2003. On the bursty evolution of blogspace. In *Proceedings of the 12th International Conference on the World Wide Web (WWW)*.
- KUMAR, R., NOVAK, J., AND TOMKINS, A. 2006. Structure and evolution of online social networks. In *Proceedings of the 12th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- LESKOVEC, J., KLEINBERG, J., AND FALOUTSOS, C. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- LIN, Y., CHI, Y., ZHU, S., SUNDARAM, H., AND TSENG, B. L. 2008. FacetNet: A framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th International Conference on the World Wide Web (WWW)*.
- LIN, Y., SUNDARAM, H., CHI, Y., TATEMURA, J., AND TSENG, B. L. 2007. Blog community discovery and evolution based on mutual awareness expansion. In *Proceedings of the International Conference on Web Intelligence*.
- LOVASZ, L. AND PLUMMER, M. 1986. *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest.
- MEI, Q. AND ZHAI, C. 2005. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the 11th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- NEWMAN, M. E. J. AND GIRVAN, M. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E*.
- NING, H., XU, W., CHI, Y., GONG, Y., AND HUANG, T. 2007. Incremental spectral clustering with application to monitoring of evolving blog communities. In *Proceedings of the SIAM International Conference on Data Mining*.
- PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1998. The PageRank citation ranking: Bringing order to the web. In Tech. Rep. Stanford Digital Library Technologies Project, Stanford University, Stanford, CA.
- PALLA, G., BARABASI, A.-L., AND VICSEK, T. 2007. Quantifying social group evolution. *Nature* 446.
- SARKAR, P. AND MOORE, A. W. 2005. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.* 7, 2.
- SHI, J. AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intel.* 22, 8.
- SPILOPOULOU, M., NTOUTSI, I., THEODORIDIS, Y., AND SCHULT, R. 2006. Monic: modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.
- SUN, J., FALOUTSOS, C., PAPADIMITRIOU, S., AND YU, P. S. 2007. GraphScope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM International SIGKDD Conference on Knowledge Discovery and Data Mining*.

- TOYODA, M. AND KITSUREGAWA, M. 2003. Extracting evolution of web communities from a series of web archives. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia (HYPERTEXT '03)*.
- WASSERMAN, S. AND FAUST, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.
- WHITE, S. AND SMYTH, P. 2005. A spectral clustering approach to finding communities in graph. In *Proceedings of the SIAM International Conference on Data Mining*.
- XU, W., LIU, X., AND GONG, Y. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- YU, K., YU, S., AND TRESP, V. 2005. Soft clustering on graphs. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*.
- ZHA, H., HE, X., DING, C. H. Q., GU, M., AND SIMON, H. D. 2001. Spectral relaxation for k-means clustering. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*.

Received September 2008; revised December 2008; accepted January 2009