Open access • Proceedings Article • DOI:10.1109/PASSAT/SOCIALCOM.2011.37

Dynamic Community Detection with Temporal Dirichlet Process — Source link

Xuning Tang, Christopher C. Yang

Institutions: Drexel University

Published on: 01 Oct 2011 - Privacy, Security, Risk and Trust

Topics: Stochastic block model, Chinese restaurant process and Dirichlet process

Dynamic Community Detection with Temporal Dirichlet Process

Xuning Tang
College of Information Science and Technology
Drexel University
xt24@drexel.edu

Christopher C. Yang
College of Information Science and Technology
Drexel University
chris.yang@drexel.edu

Abstract—**Research of detecting dynamic communities from network stream has attracted increasingly attention recently due to its broad applications on social media, e-commence, intelligent security, healthcare and more. Some of the previous techniques employed a two-stage approach to detect communities within each time epoch and then identify the evolutionary relationships between communities from adjacent time epochs. However, since the two-stage approaches detect communities within each epoch independently, the identified communities usually have high temporal variation. Another restriction of the previous techniques is the requirement of predefining the number of hidden communities by a fixed value or within a very narrow range. To overcome these limitations, we propose the Dynamic Stochastic Blockmodel with Temporal Dirichlet Process, which is able to detect communities and track their evolution simultaneously from a network stream. The number of communities is automatically decided by a Recurrent Chinese Restaurant Process without human intervention. In addition, the identified communities exhibit a rich-gets-richer effect and other appealing properties. The experiment results on both simulated dataset and Flickr dataset showed that our proposed algorithm outperformed the baseline algorithm and achieved promising results.**

Keywords-temporal dirichlet process; recurrent chinese restaurant process; community detection; stochastic blockmodel

I. INTRODUCTION

In the recent years, online social networking sites have drawn significant attention and achieved unanticipated success. For instance, MySpace, Facebook and Twitter attract millions of daily visits and their number of registered users is growing incessantly. In addition, attribute to the advance of Web 2.0 technologies, the number of websites with embedded social networking functions, including Flickr, Delicious and YouTube, is also growing substantially. As a result, network data has become the richest and the most valuable information on the Web. Among several research topics of social network analysis, community detection has drawn substantial attention due to its broad range of applications. For example, identifying groups of users with similar interest can foster sharing of resources (in Flickr or Delicious), detecting potential gang or terrorist group in suspicious online forums is helpful in security informatics, identifying active and influential users in communities of online consumer reviews can be helpful to target potential customers and promoting business (especially in online shopping site). In addition, we can further broaden these applications by extending the static social network analysis techniques to detect evolving communities from dynamic social networks. For example, we can analyze the dynamic online social media to predict developing common interest groups for marketing and business intelligence purpose. We can monitor the evolving social movements to ensure the public safety.

In light of the benefit of community detection, a large body of research works has been focusing on this topic. Many of these works detected communities from a static network based on modularity, clique, betweenness, graph spectrum, or generative model [1-5]. However, none of them have taken timestamp into account, and therefore, they are not applicable to dynamic network. A few other works represented dynamic network by a stream of static networks [6-9]. They employed a two-stage approach which identifies communities from the static network of each time epoch, and then examines communities in adjacent epochs to identify their evolutionary relationships. However, since two-stage approaches detect communities of each epoch independently, they often result in communities with high temporal variation [10]. Lately, several research groups proposed novel approaches to detect community and identify evolutionary relationship simultaneously [10-13]. However, these works have another limitation: the number of communities is either fixed or restricted within a narrow range.

In light of these limitations, we anticipate that a good community detection algorithm should include the following characteristics. Firstly, it must be able to detect robust communities dynamically. Community detection and evolutionary relationship identification should be conducted in a unified framework . Secondly, the number of communities in the dynamic social network should be estimated automatically without human intervention. Thirdly, nodes in dynamic graphs appear and disappear over time, so that a good community detection algorithm should be able to handle node addition and node attrition naturally. Last but not least, rich-gets-richer effect is a typical phenomenon in social network. Detected communities should exhibit this phenomenon.

To achieve all of these goals, we extended the stochastic blockmodel by incorporating the Recurrent Chinese Restaurant Process in this paper. Recurrent Chinese Restaurant Process not only enables our model to handle dynamic network but also determines the number of

communities at each time epoch automatically. Our proposed technique possesses all the desired characteristics mentioned above. The major contributions of this work include: (1) synthesizing Recurrent Chinese Restaurant Process and stochastic blockmodel to detect evolving communities; (2) deriving the Dynamic Stochastic Blockmodel with Temporal Dirichlet Process and proposing an efficient Gibbs Sampling algorithm to infer community assignments and model parameters; (3) conducting a rigorous experiment to test our model thoroughly on both large-scale Flickr dataset and simulated dataset. The experiment results indicated the effectiveness and validity of our model.

II. Related Works

Community detection has been widely studied in various research domains for many years. For instance, Scott [14] employed hierarchical clustering approach to discover communities in a social network. Girvan and Newman [3] developed heuristic algorithm to detect community structures in a biological network with strong modularity. Dhillon, Guan and Kulis [2] proposed spectral clustering algorithms to divide a graph into multiple sub-graphs with specific applications of image segmentation. Recently, another group of research works [1, 4, 5] proposed probabilistic model-based techniques to estimate communities based on user interaction. However, all of them neglected the time feature of network.

Recently, there has been a growing body of work taking the time feature into account to analyze evolutionary networks [6-9]. Asur et al. detected communities within each time epoch, and then employed similarity metrics to identify evolutionary relationship between communities in adjacent time epochs[7, 8]. Sun developed a parameter-free algorithm to mine time-evolving networks using information theoretic principles[9]. Berger-Wolf proposed an optimization-based approach which formulated the community detection problem as a coloring problem in network stream[6]. These works adopted a two-stage approach where community detection and community evolution are conducted separately. However, this two-stage approach did not make use of the community information of the previous epoch to detect communities of the current epoch. As a result, there was a high variation of the detected communities in different epochs.

To address this problem, some recent studies worked on dynamic social network analysis by detecting community and identifying evolutionary relationship under a unified framework. Yang et al. extended the stochastic blockmodel and proposed a dynamic stochastic blockmodel which can detect communities and their evolution concurrently [12]. Miller and Eliassi-Rad extended the cDTM model to cDTM-G model which allows the communities to evolve as Brownain motion[13]. Fu, Song and Xing extended Airoldi's work [1] to model the evolution of mixed membership blockmodel [11]. However, the number of communities is fixed over time in all these works. It is difficult to predefine the number of communities in real-world application. Lin et al. extended the graph-factorization clustering technique and proposed the FacetNet algorithm for analyzing dynamic communities, which relaxed the restriction of the number of communities within each time epoch by setting a range of candidates and searching the best number of communities by multiple trials [10], and was computationally expensive. In order to allow countable infinite community theoretically and to find the optimal number of communities naturally, we propose to incorporate Recurrent Chinese Restaurant Process into a dynamic stochastic blockmodel to detect communities and their evolution in a unified framework. In addition, our model enables the number of communities of each epoch to be determined automatically.

III. Problem Statements And Notations

In this section, we formulate the research problem and present the notations that are used throughout this paper. Let $E^{(t)}$ denotes a social network at time t where $E^{(t)} \in \mathbb{R}^{N^{(t)} \times N^{(t)}}$ and $N^{(t)}$ equals to the number of nodes in $E^{(t)}$ at time t. It is important to note that, in real-world application, $E^{(t)}$ can be a network with one component or multiple components. Our proposed technique handles a network with any number of components in the same way. To make it simple, we only consider undirected graphs with binary edges in this study but it can be easily extended to weighted directed graph with minor modification. Given an undirected graph with binary edges only, if there is an edge between node i and node j in $E^{(t)}$, then $E^{(t)}_{i,j} = E^{(t)}_{j,i} = 1$. Otherwise, $E^{(t)}_{i,j} = E^{(t)}_{j,i} = 0$. We assume that $E^{(t)}$ contains some hidden communities, which are treated as latent variables in our model. Each node in $E^{(t)}$ belongs to one hidden community. Any two nodes of the same hidden community have a larger probability to interact with one another. Let $K^{(t)}$ denote the total number of hidden communities at time t, and $P^{(t)}_{a,b}$ denote the parameter of a Bernoulli distribution upon which an edge between community a and b is drawn, where $P^{(t)}_{a,b} \in \mathbb{R}^{K^{(t)} \times K^{(t)}}$ and $1 \leq a, b \leq K^{(t)}$. Moreover, we represent the community assignment of node i at time t by $c^{(t)}_i$. We let $c^{(t)}_{1:i-1}$ denote the community assignments from node 1 to node $i-1$ at time t, $c^{(t)}_{-i}$ denote the community assignments of all nodes except node i at time t, and $C^{(t)}$ denote the community assignments of all nodes of $E^{(t)}$. We further denote $n^{(t)}_{i,k}$ as the number of $c^{(t)}_j$ for $j < i$ that are assigned to community k, and $n^{(t-1)}_k$ equal to the total number of nodes that are assigned to community k at time -1.

In this paper, given a stream of social networks over T epochs, $E^T = \{E^{(1)}, E^{(2)}, \dots, E^{(T)}\}$, the objective of our proposed technique is to determine the optimal number of hidden communities at each time epoch automatically ($\{K^{(1)}, K^{(2)}, \dots, K^{(T)}\}$), uncover the community assignment for nodes of each network snapshot ($\{C^{(1)}, C^{(2)}, \dots, C^{(T)}\}$), and track the evolution of these communities over time.

IV. Preliminaries

In this section, we briefly review the Dirichlet Process Mixture Model and the Stochastic Blockmodel which serves as the foundation of our proposed model.

A. Dirichlet Process Mixture Model

Finite Mixture Model is a frequently used model in clustering which assumes that each observation o_i is generated by one of K fixed unknown distributions parameterized by K different parameters, $\theta_1, \theta_2, \dots, \theta_K$. However, the number of K in Finite Mixture Model is fixed and difficult to be specified appropriately without prior knowledge. To determine K flexibly, Dirichlet Process Mixture Model (DPM) is proposed assuming that the parameter θ is drawn from a distribution G, denoted as $\theta_i|G \sim G$. In DPM, the distribution G is considered to be generated by a Dirichlet Process (DP) with a base measure G_O and a concentration parameter α, denoted as $G \sim DP(\alpha, G_O)$. Formally speaking, the DPM model can be expressed in an equivalent way which is easier for understanding and sampling[15]. Given a Finite Mixture Model with K components with the following form:

$$\begin{aligned} \pi_1, \pi_2, \dots, \pi_K &\sim Dirichlet(\alpha/K, \alpha/K, \dots, \alpha/K) \\ \theta_1, \theta_2, \dots, \theta_K &\sim G_O \\ c_i|\pi &\sim Discrete(\pi_1, \pi_2, \dots, \pi_K) \\ o_i|c_i, \{\theta_k\}_{k=1}^K &\sim f(\theta_{c_i}) \end{aligned} \tag{1}$$

where c_i indicates the latent component that is associated with observation o_i, the DPM model can be obtained by considering the limit of the model as K $\to \infty$. By integrating π and considering K $\to \infty$, the conditional probabilities of c_i in Eq.(1) become:

$$P(c_i = k|c_1, .., c_{i-1}) = \begin{cases} \dfrac{\alpha}{i-1+\alpha} & if\ c_j \neq k\ for\ all\ j < i \\ \dfrac{n_{i,k}}{i-1+\alpha} & otherwise \end{cases} \tag{2}$$

with $n_{i,k}$ is the number of c_j for $j < i$ that are assigned to component k. Taking one step further, Eq.(2) can be well explained by a Chinese Restaurant Process (CRP) metaphor. In the CRP, we assume that there is a Chinese restaurant with an unbounded number of tables. When a customer c_i enters this restaurant, he can either choose a new table with probability $\frac{\alpha}{i-1+\alpha}$ and order a new dish, or pick a table k which already has $n_{i,k}$ customers with probability $\frac{n_{i,k}}{i-1+\alpha}$ and share their dishes. From the CRP metaphor, DPM not only can determine the final number of mixture models flexibly but also reflects a rich-gets-richer effect.

B. Stochastic Blockmodel

Stochastic blockmodel is a generative model which is widely studied to analyze a static social network. A major assumption of the stochastic blockmodel is that each node i of the network belongs to one of K hidden communities with probabilities $\{\varphi_{i,1}, \varphi_{i,2}, \dots, \varphi_{i,K}\}$, where K is predefined. Then, assuming that node i belongs to community a and node j belongs to community b, the probability of observing an edge between these two nodes is generated by a Bernoulli distribution with parameter $P_{a,b}$, where $P_{a,b} \in \mathbb{R}^{K\times K}$ and $1 \leq a, b \leq K$. By following this problem definition, Nowicki and Snijders [4] proposed a technique which discovered the community assignment for each node of a social network with the maximum posterior probability given the observed edges.

V. Dyanmic Community Detection

Given the research problem defined in section III, the focus of this work is to develop an algorithm to detect communities and their evolutions from a network stream. Another research focus is to provide flexibility to the model in determining the number of communities automatically. Since stochastic blockmodel can detect hidden communities from a static network and DPM is good at deciding the number of hidden components, there is an advantage to integrate DPM with stochastic blockmodel to detect unbounded number of hidden communities from a network. However, DPM is incapable to handle network evolution. To address this problem, we extend CRP to **Recurrent Chinese Restaurant Process (RCPR)** which can be considered as a **Temporal Dirichlet Process**, and then synthesis RCRP and stochastic blockmodel to model community evolution.

A. Recurrent Chinese Restaurant Process

The Chinese Restaurant Process (CRP) introduced in section IV can be generalized to the Recurrent Chinese Restaurant Process (RCRP), which operates in discrete time epochs, i.e. days. The major difference is that both the popular dishes and the seating plan of the current time epoch will influence the customers' selections in the next epoch, leading to a rich-gets-richer phenomenon not only within an epoch but also in adjacent epochs. Formally speaking, let's consider the generative process for a **finite** dynamic mixture model with K components. Within a given epoch t, the generation process for each observation i is defined as follows:

$$\begin{aligned} \pi_1^{(t)}, \dots, \pi_K^{(t)} &\sim Dirichlet(n_1^{(t-1)} + \alpha/K, \dots, n_K^{(t-1)} + \alpha/K) \\ \forall k:\ \theta_k^{(t)} &\sim P(.|\theta_k^{(t-1)}) \\ c_i^{(t)}|\pi^{(t)} &\sim Discrete(\pi_1^{(t)}, \dots, \pi_K^{(t)}) \\ o_i|c_i^{(t)}, \{\theta_k^{(t)}\}_{k=1}^K &\sim f(\theta_{c_i^{(t)}}^{(t)}) \end{aligned} \tag{3}$$

By integrating $\pi^{(t)}$ and considering the limit of the model as K $\to \infty$, the conditional probabilities of $c_i^{(t)}$ in Eq.(3) can be derived as:

$$P\left(c_i^{(t)} = k \middle| c_{1:N^{(t-1)}}^{(t-1)}, c_{1:i-1}^{(t)}\right) \tag{4}$$
$$= \begin{cases} \dfrac{\alpha}{N^{(t-1)} + i - 1 + \alpha} & if\ c_j^{(t)} \neq k\ for\ all\ j < i\ and\ \forall i\ c_i^{(t-1)} \neq k \\ \dfrac{n_k^{(t-1)} + n_{i,k}^{(t)}}{N^{(t-1)} + i - 1 + \alpha} & otherwise \end{cases}$$

Similarly, Eq.(4) can be well explained by the Recurrent Chinese Restaurant Process (RCRP) metaphor. In the RCRP, customers enter the restaurant in a given day are not allowed to stay beyond this day. At the end of day $t-1$, the owner of the restaurant records on each tale the dish served in this table and the number of customers who shared it, since he believes that popular dishes will remain popular in the next day [16]. Given these information, When a customer $c_i^{(t)}$ enters this restaurant at day t, he can pick a **non-empty** table k that already has $n_{i,k}^{(t)}$ on day t, and share their dish with probability $\frac{n_k^{(t-1)} + n_{i,k}^{(t)}}{N^{(t-1)} + i - 1 + \alpha}$. If this table does not exist on day $t-1$, $n_k^{(t-1)}$ equals to 0.

Otherwise, $n_k^{(t-1)}$ equals to the number of customers who sit at this table on day $t-1$. Alternatively, he can pick an **empty** table that nobody is sitting at on day t but $n_k^{(t-1)}$ customers sit on at day $t-1$ with probability $\frac{n_k^{(t-1)}+n_{i,k}^{(t)}}{N^{(t-1)}+i-1+\alpha}$ where $n_{i,k}^{(t)}$ equals to 0. Finally, he can pick an **empty new** table with probability $\frac{\alpha}{N^{(t-1)}+i-1+\alpha}$ and then order a new dish. By putting these alternatives together, we arrive at Eq. (4). The table in this metaphor corresponds to community.

B. Dynamic Stochastic Blackmodel with Temporal Dirichlet Process

To model dynamic social networks, we propose the **Dynamic Stochastic Blockmodel with Temporal Dirichlet Process (DBTDP)** which incorporates the Recurrent Chinese Restaurant Process into the Stochastic Blockmodel to detect community evolution in network stream. The DBTDP model is defined in a recursive way. In the initial network $E^{(1)}$ when $t=1$, assuming $i-1$ nodes have been assigned to k communities by following the CRP, we either assign the i^{th} node to one of the k existing communities with probability $\frac{n_{i,k}}{i-1+\alpha}$, or add a new community and then assign the i^{th} node to this new community with probability $\frac{\alpha}{i-1+\alpha}$. Since CRP is exchangeable, the order of assigning nodes to communities can be permuted without affecting the probability of $C^{(1)}$. Given the community assignments $C^{(1)}$ is available, the edges between nodes in $E^{(1)}$ are generated stochastically by probabilities $P^{(1)}$. Similarly, assuming the community assignments $C^{(t-1)}$ is available, in network $E^{(t)}$ of time t, assuming $i-1$ nodes have been assigned to k communities by following the RCRP, for the i^{th} node, we either assign it to one of k existing communities with probability $\frac{n_k^{(t-1)}+n_{i,k}^{(t)}}{N^{(t-1)}+i-1+\alpha}$, or add a new community and then assign it to this new community with probability $\frac{\alpha}{N^{(t-1)}+i-1+\alpha}$. Given the community assignments C^t is available, the edges between nodes in $E^{(t)}$ are decided stochastically by probabilities $P^{(t)}$. The generative process of the DBTDP is shown as follows.

For time $t=1$:
  Draw each $c_i^{(1)}$ based on $c_{1:i-1}^{(1)}$ by CRP
  For each edge (i,j) of $E^{(1)}$:
    Draw each $e_{i,j}^{(1)} \sim Bernoulli(p_{c_i^{(1)}, c_j^{(1)}})$

For $t>1$:
  Draw each $c_i^{(t)}$ based on $c_{1:i-1}^{(t)}$ and $C^{(t-1)}$ by RCRP
  For each edge (i,j) of $E^{(t)}$:
    Draw each $e_{i,j}^{(t)} \sim Bernoulli(p_{c_i^{(t)}, c_j^{(t)}})$

To illustrate the DBTDP model, we present a graphical model representation of DBTDP in figure 1. In figure 1, at the first time epoch, parameters $\pi^{(1)}$ are generated based on the concentration parameter α. For the following time epochs t when $t>1$, parameters $\pi^{(t)}$ are generated based on both the concentration parameter α and the community assignments of latest time epoch $C^{(t-1)}$. The community assignments of current time epoch are decided by $\pi^{(t)}$. Similarly, parameters $\theta^{(t)}$ are decided by both the base measure G_O and the $\theta^{(t-1)}$ of previous time epoch. Within each time epoch, once the community assignments for all nodes are determined, each edge $e_{i,j}^{(t)}$ is generated based on $c_i^{(t)}$, $c_j^{(t)}$ and $P_{i,j}$. The shaded nodes in Fig. 1 represent the observed edges while the white nodes represent hidden variables.

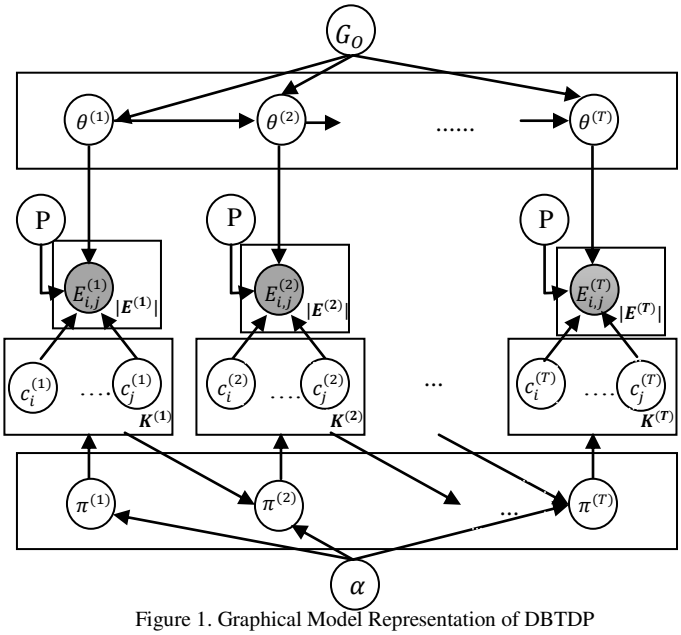


Figure 1. Graphical Model Representation of DBTDP

C. Likelihood Of The Complete Data

Based on the intuitions and observations, we can then introduce three reasonable independent assumptions, which can be useful to derive the likelihood of the data:

A1. For the initial network where t=1, community assignment of a node, say i, in $E^{(1)}$ is determined by the community assignments of all other nodes except i in $E^{(1)}$, denoted as:

$$\Pr\left(C^{(1)}\middle|\alpha, G_o\right) = \prod_{i=1}^{n} \Pr\left(c_i^{(1)}|c_{-i}^{(1)}, \alpha, G_o\right) \quad (5)$$

A2. Similarly, for the networks where time t>1, community assignment of a node, say i, in $E^{(t)}$ is determined by the community assignments of all other nodes except i in $E^{(t)}$ and the community assignments of all nodes in $E^{(t-1)}$, denoted as:

$$\Pr(C^{(t)}|C^{(t-1)}, \alpha, G_o) = \prod_{i=1}^{n} \Pr\left(c_i^{(t)}|c_{-i}^{(t)}, C^{(t-1)}, \alpha, G_o\right) \quad (6)$$

A3. Any edge (i,j) in $E^{(t)}$ is generated independently from the other nodes/edges given the community assignments of node i and j. The generation process follows a Bernoulli distribution:

$$\Pr\left(E^{(t)}\middle|C^{(t)}, P^{(t)}\right) = \prod_{1\le a<b\le K^{(t)}} p_{a,b}^{(t)\,m(a,b)} \left(1-p_{a,b}^{(t)}\right)^{\overline{m(a,b)}} \times \prod_{a=1}^{K^{(t)}} p_{a,a}^{(t)\,m(a,a)} \left(1-p_{a,a}^{(t)}\right)^{\overline{m(a,a)}} \quad (7)$$

where

$$m(a,b) = (1+I\{a=b\})^{-1} \times \sum_{i,j} I\left\{E_{i,j}^{(t)}=1\right\} I\{c_i^{(t)}=a\} I\{c_j^{(t)}=b\} \quad (8)$$

$$\overline{m(a,b)} = (1+I\{a=b\})^{-1} \times \sum_{i,j} I\left\{E_{i,j}^{(t)}=0\right\} I\{c_i^{(t)}=a\} I\{c_j^{(t)}=b\} \quad (9)$$

$I\{A\}$ equals to 1 if condition A is satisfied, and 0 otherwise. With the above assumptions, the Dynamic Stochastic Blockmodel with Dirichlet Process is then given by the joint distribution of (E_T, C_T)

$$\Pr(E_T, C_T|\alpha, G_o, P_T) = \prod_{t=1}^{T} \Pr\left(E^{(t)}\middle|C^{(t)}, P^{(t)}\right) \times \prod_{t=2}^{T} \Pr\left(C^{(t)}\middle|C^{(t-1)}, \alpha, G_o\right) \Pr\left(C^{(1)}|\alpha, G_o\right) \quad (10)$$

VI. Gibbs Sampling Algorithm

A. Model Inference

Given Eq.(10), the objective of this section is to identify the most likely variables that can maximize the likelihood of the data. One intuitive approach is to estimate the most likely values of $\{\alpha, G_o, P_T\}$ which can maximize Eq. (10). However, in DBTDP, the community assignments C_T are unknown. A typical Bayesian treatment for this case is to estimate the posterior probability instead of maximizing the likelihood of data. Since P_T is generated by a Bernoulli distribution, we define a Beta distribution and incorporate it into the model as the conjugate prior of $\Pr\left(E^{(t)}\middle|C^{(t)}, P^{(t)}\right)$:

$$\Pr(P|\gamma, \beta) = \prod_{1\le a<b\le K^{(t)}} \frac{\Gamma(\gamma_{a,b}+\beta_{a,b})}{\Gamma(\gamma_{a,b})\Gamma(\beta_{a,b})} p_{a,b}^{(t)\,\gamma_{a,b}-1}(1 - p_{a,b}^{(t)})^{\beta_{a,b}-1} \times \prod_{a=1}^{K^{(t)}} \frac{\Gamma(\gamma_{a,a}+\beta_{a,a})}{\Gamma(\gamma_{a,a})\Gamma(\beta_{a,a})} p_{a,a}^{(t)\,\gamma_{a,a}-1}(1 - p_{a,a}^{(t)})^{\beta_{a,a}-1} \quad (11)$$

Given this beta prior, the probability of generating $E^{(t)}$ given the community assignments $C^{(t)}$ is represented as:

$$\begin{aligned}\Pr\left(E^{(t)}\middle|C^{(t)}, \gamma, \beta\right) &\propto \int \Pr\left(E^{(t)}\middle|C^{(t)}, P^{(t)}\right) \Pr\left(P^{(t)}|\gamma, \beta\right) dP^{(t)} \\ &= \prod_{1\le a<b\le K^{(t)}} B(m(a,b) + \gamma_{a,b}, \overline{m(a,b)} + \beta_{a,b}) \\ &\times \prod_{a=1}^{K^{(t)}} B(m(a,a) + \gamma_{a,a}, \overline{m(a,a)} + \beta_{a,a}) \quad (12)\end{aligned}$$

Secondly, we employ Maximum A Posteriori Probability estimation (MAP) to uncover the community assignments C_T:

$$C_T{}^{*} = argmax_{C_T} \Pr\left(C_T|E_T, \alpha, G_o, \gamma, \beta\right) \quad (13)$$

Finally, we determine the greedy optimization community assignments $C^{(t)}$ based on previous community assignments $C^{(t-1)}$. Since this is a stream of social networks, it is logical to consider the networks one after one instead of analyzing all networks simultaneously and estimating the global configuration of C_T. The objective posterior probabilities is derived as:

$$\begin{aligned} C^{(t)^{*}} &= argmax_{C^{(t)}} \Pr\left(C^{(t)}\middle|E^{(t)}, C^{(t-1)}, \alpha, G_o, \gamma, \beta\right) \quad t > 1 \\ C^{(1)^{*}} &= argmax_{C^{(1)}} \Pr\left(C^{(1)}\middle|E^{(1)}, \alpha, G_o, \gamma, \beta\right) \quad t = 1 \end{aligned} \quad (14)$$

B. Gibbs Samping Algorithm

Given the construction for the DBTDP model, to maximize the posterior probabilities, we derive the collapsed Gibbs sampling algorithm:

Step 1 Initialization: For the input network $E^{(1)}$, we initialize the model by creating $K^{(init)}$ empty communities, and then assigning each node into one of $K^{(init)}$ communities randomly. With the network $E^{(t)}$ where t>1 and the nodes appeared in $E^{(t-1)}$, we initialize the model by maintaining their community assignment according to the sampling result of $C^{(t-1)}$. For the nodes that do not appear in $E^{(t-1)}$, we assign them to one of $K^{(t-1)}$ communities randomly.

Step 2 Community Assignment Estimation: For each node, we repeat the following two steps until reach iteration number.

Step 2.1: Compute/Update $m(a,b), \overline{m(a,b)}$ for $1 \le a \le b \le K^{(t)}$, $n_{i,k}^{(t)}$ for $node\ i \in E^{(t)}\ and\ 1 \le k \le K^{(t)}$, and $n_k^{(t-1)}$ for $1 \le k \le K^{(t-1)}$ with t>1.

Step 2.2: Sample each of the objects into existing communities or new community, following the posterior probabilities:

$$\begin{aligned}&\Pr\left(c_i^{(t)} = k\middle|E^{(t)}, C^{(t-1)}, c_{-i}^{(t)}, \alpha, G_o, \gamma, \beta\right) \quad (15.1)\\ &\propto \frac{\alpha}{N^{(t-1)}+N^{(t)}-1+\alpha} \prod_{1\le a<b\le K^{(t)}} B(m(a,b) + \gamma_{a,b}, \overline{m(a,b)} + \beta_{a,b}) \times \prod_{a=1}^{K^{(t)}} B(m(a,a) + \gamma_{a,a}, \overline{m(a,a)} + \beta_{a,a})\end{aligned}$$

when t>1 and $c_j^{(t)} \ne k\ for\ all\ j \ne i\ and\ \forall i, c_i^{(t-1)} \ne k$

$$\begin{aligned}&\Pr\left(c_i^{(t)} = k\middle|E^{(t)}, C^{(t-1)}, c_{-i}^{(t)}, \alpha, G_o, \gamma, \beta\right) \quad (15.2)\\ &\propto \frac{n_k^{(t-1)}+n_{i,k}^{(t)}}{N^{(t-1)}+N^{(t)}-1+\alpha} \prod_{1\le a<b\le K^{(t)}} B(m(a,b) + \gamma_{a,b}, \overline{m(a,b)} + \beta_{a,b}) \times \prod_{a=1}^{K^{(t)}} B(m(a,a) + \gamma_{a,a}, \overline{m(a,a)} + \beta_{a,a})\end{aligned}$$

when t>1 and $\exists\ j,\ c_j^{(t-1)} = k\ or\ \exists\ j \ne i,\ c_j^{(t)} = k$

OR

$$\begin{aligned}&\Pr\left(c_i^{(1)} = k\middle|E^{(1)}, c_{-i}^{(1)}, \alpha, G_o, \gamma, \beta\right) \quad (16.1)\\ &\propto \frac{\alpha}{N^{(1)}-1+\alpha} \prod_{1\le a<b\le K^{(t)}} B(m(a,b) + \gamma_{a,b}, \overline{m(a,b)} + \beta_{a,b}) \times \prod_{a=1}^{K^{(1)}} B(m(a,a) + \gamma_{a,a}, \overline{m(a,a)} + \beta_{a,a})\end{aligned}$$

when t=1 and $c_j^{(1)} \ne k\ for\ all\ j \ne i$

$$\begin{aligned}&\Pr\left(c_i^{(1)} = k\middle|E^{(1)}, c_{-i}^{(1)}, \alpha, G_o, \gamma, \beta\right) \quad (16.2)\\ &\propto \frac{n_{i,k}}{N^{(1)}-1+\alpha} \prod_{1\le a<b\le K^{(t)}} B(m(a,b) + \gamma_{a,b}, \overline{m(a,b)} + \beta_{a,b}) \times \prod_{a=1}^{K^{(1)}} B(m(a,a) + \gamma_{a,a}, \overline{m(a,a)} + \beta_{a,a})\end{aligned}$$

when t=1 and $\exists\ j \ne i, c_j^{(1)} = k$

The Gibbs Sampling algorithm is summarized as follow:

Input: Network stream $\{E^{(1)}, E^{(2)}, \dots, E^{(T)}, \dots\}$, $\alpha, G_o, \gamma, \beta$

Output: Community Assignments $\{C^{(1)}, C^{(2)}, \dots, C^{(T)}, \dots\}$

Initialize the community assignment by following step 1;

Repeat

1. Update statistics in step 2.1 incrementally
2. Compute the posterior probability for each node by following step 2.2
3. Assign this node to one of existing communities or create an empty community and then assign this node to it according to the posterior probabilities.

Until reaches iteration number

VII. EXPERIMENT

We conducted two experiments to evaluate our proposed model extensively. We first tested our model in two simulated datasets with different noise levels. We also tested our model on a dynamic Flickr user network with 2.5M nodes and 33M edges. The experiments demonstrated the scalability and effectiveness of our model.

A. Evaluation Metrics

In this study, we adopted the metrics of previous works [12, 17] to measure the performance of our model. These metrics are normalized mutual information (NMI) and robustness (Robust). Normalized mutual information is used when there is a ground truth, e.g. in simulated dataset. Formally speaking, given a true community partition, denoted as $C = \{C_1, C_2, \dots, C_K\}$ where C_i is a group of users of community i, also given the estimation of community partition $C' = \{C_1', C_2', \dots, C_K'\}$, the normalized mutual information is defined as:

$$NMI(C, C') = \frac{\sum_{C,C'} P(C,C') \log \frac{p(C,C')}{p(C')p(C)}}{\max(H(C), H(C'))} \quad (17)$$

where $H(C)$ and $H(C')$ are the entropies of C and C'. According to Eq. (17), NMI is a value between 0 and 1. The higher the NMI score, the more similar the ground truth and the estimation is.

When the ground truth is not available, e.g. in the real-world dataset, robustness will be employed for measuring community partitions[18]. We didn't use Modularity as measurement given its limitation as depicted in [18]. Formally speaking, we first employ a method to divide a network into K communities, denoted as $C = \{C_1, C_2, \dots, C_K\}$. Secondly, we perturb the network by randomly reassigning a number of its links according to a rewiring parameter $\boldsymbol{\alpha}$, which determines the fraction of links rewired. Then, the same method is applied on this perturbed network to gain another partition $C' = \{C_1', C_2', \dots, C_{K'}'\}$, and the variation of information is computed between the two community assignments as:

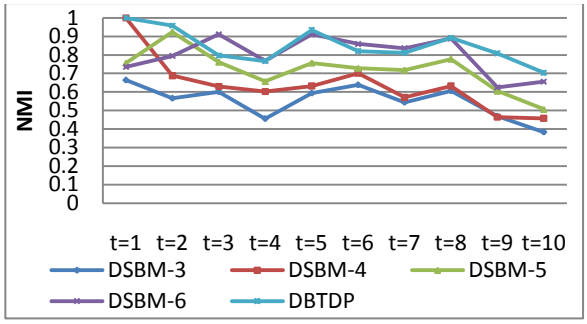
$$R(C, C') = H(C'|C) + H(C|C') \quad (18)$$

where $H(C'|C)$ is the conditional entropy. We normalized this score by 1/Log(N). According to Eq. (18), if two community assignments are still similar to each other after network perturbation, the robustness score will be **low**, which indicates that the detected community structures are able to withstand small perturbations so that they are believable.

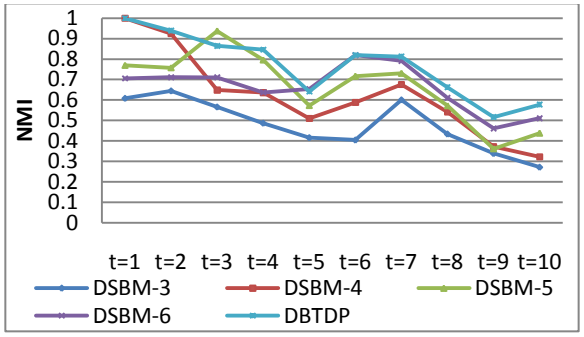
B. Experiments on Simulated Datasets

1) Data Generator: Yang et al. [12] proposed a procedure to automatically generate evolving social networks each of which contains 128 nodes and 4 communities. In their procedure, when time t=1, each node was randomly assigned to one community. Each community contained 32 nodes. Yang predefined two probability value p_{in} and p_{out}, and the average degree of nodes in the network as 16. To generate edges for the network, for each pair of nodes of the same community, an edge was created by the probability p_{in}. Similarly, for each pair of nodes of different communities, an edge was created by the probability p_{out}. At each time epoch after time epoch 1, they randomly chose 10% of the nodes to leave their original community and joined the other three communities at random. Edges of the network were determined by probability p_{in} and p_{out} as before after reassigning the nodes. They generated the networks with evolution in this way for 10 time steps.

However, Yang's procedure only generated evolving networks with constant number of community. To study the network evolution with different number of communities, we further modified Yang's procedure by randomly adding one community, deleting one community, or keeping the same number of communities with the probability of p_a, p_d and p_r respectively at each time epoch ($p_a + p_d + p_r = 1$). If a new community was added, we randomly chose P% of the nodes from each old community to join this new community (P is computed to make sure that each community has the roughly same number of nodes), and then chose 10% of the nodes from each existing communities to leave their original community and join the other communities randomly. If an old community was deleted, we randomly assigned the nodes of that community to the other communities, and then chose 10% of the nodes to leave their original community and join the other communities randomly. If the number of community did not change, we randomly chose 10% of the nodes to leave their original community and join the other communities randomly. Edges of the network were determined by the probabilities p_{in} and p_{out} after reassigning the nodes. We generated the evolving networks in this way for 10 iterations. We tested our algorithm under two different noise levels by setting the ratio of p_{in}/p_{out} approximately equals to 4 and 3, respectively.



(a) Noise Level 1 (p_{in}/p_{out}=4)



(b) Noise Level 2 (p_{in}/p_{out}=3)

Figure 2: Algorithms' performance with respect to the ground truth on datasets with different noise levels over 10 epochs.

2) Comparison with baseline algorithm: In this paper, we compare the performance of our proposed DBTDP algorithm with the online version of DSBM algorithm proposed by Yang et al. [12], since DSBM outperformed all other state-of-the-art algorithms, such as FacetNet [10] and EvolSpect [19]. It is important to note that DSBM algorithm needs a predefined number of communities as input and this number remains unchanged. To compare DBTDP with DSBM fairly, we fed DSBM with different number of communities, from 3 to 6, and then compared DBTDP with all of them. For the DBTDP model, we empirically set $\gamma_{a,a}$=200, $\gamma_{a,b}$= 40, $\beta_{a,a}$ =1, $\beta_{a,b}$ =30, and $\alpha = 1$. DBSM-3 represents DBSM model with the predefined number of communities equaled to 3. DBSM-4 represents DBSM model with the predefined number of communities equaled to 4 and so forth. Since we have the ground truth of the community assignments for all nodes over these 10 epochs, we employed Normalized Mutual Information (NMI) to evaluate them.

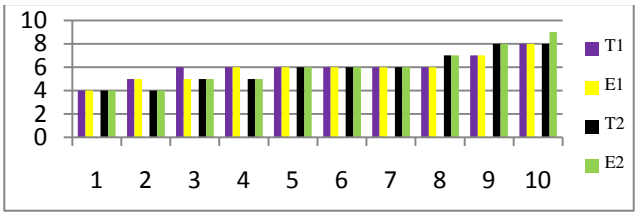


Figure 3: Number of communities of the ground truth and the number of estimated communities by DBTDP

Figure 2 presents the performance of DBTDP, DBSM-3, DBSM-4, DBSM-5 and DBSM-6 with respect to the ground truth on the two datasets with different noise levels over 10 epochs. Figure 2 (a) corresponds to a lower noise level while Figure 2 (b) corresponds to a higher noise level. Both of them demonstrated that DBTDP algorithm achieves the highest NMI score most of the time. In addition, DSBM algorithm performed well only when the true number of communities is close to the predefined number of communities, while DBTDP can automatically adjusted the number of communities and maintained a relatively high NMI score. In figure 3 we plot the number of communities in the ground truth over 10 epochs along with the number of communities estimated by DBTDP of both datasets. T1 and E1 represent the true number of communities and the estimated number of communities of the dataset with the noise level 1. T2 and E2 represented the true number of communities and the estimated number of communities of the dataset with the noise level 2.

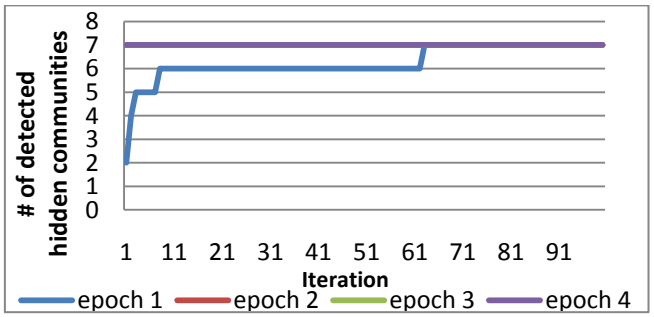
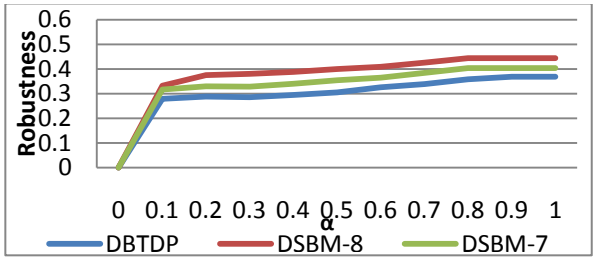
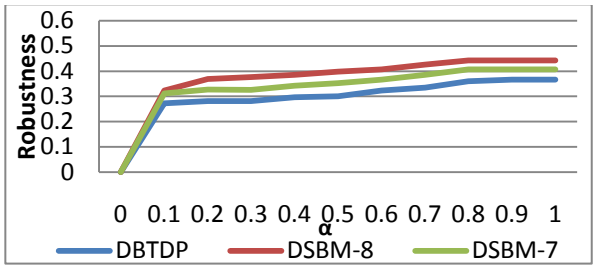


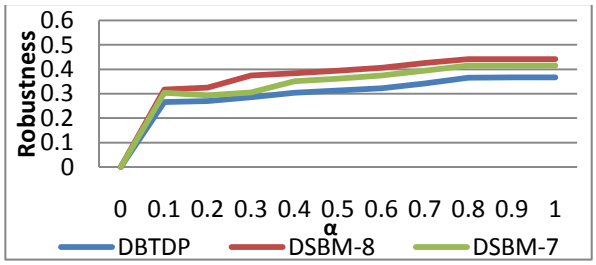
Figure 4: The number of detected communities by DBTDP model from Flickr dataset on different time epoch



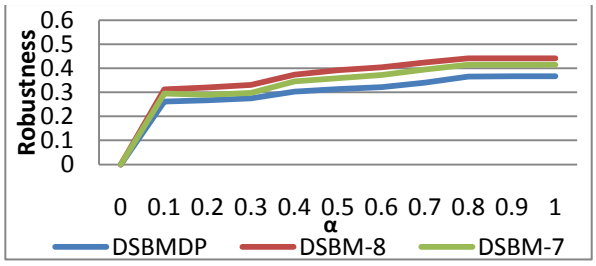
(a) Epoch 1



(b) Epoch 2



(c) Epoch 3



(d) Epoch 4

Figure 5: The performance of the DBTDP model on Flickr dataset evaluated by robustness.

C. *Experiments on Flickr Datasets*

1) Data Description: In the second experiment, we use the Flickr dataset mentioned in [20] to test the scalability of our proposed model. Cha [20] crawled the Flickr social network graph once per day for the period of 104 consecutive days from November 2nd –December 3rd , 2006 and February 3rd – May 18th, 2007. 2.5 million Flickr users and 33 million edges were observed during these two periods. We aggregated the historical data from November 2nd –December 3rd to form the social network of time epoch 1. Similarly, we aggregated the data from February 3rd – March 3rd, March 4th – April 4th and April 5th – May 18th to form the social networks of epoch 2, 3 and 4 respectively. Our proposed model was applied on these networks to identify hidden communities. It is worth to

mention that, due to the power law, most of the Flickr users have very low degree centrality. In this study, we removed the nodes that had low degree centrality and kept the core nodes (11K nodes) and their relationships (6.3M edges).

2) Performance of DBTDP on Flickr Dataset: Due to the limited space, we cannot plot the result of our proposed algorithm with different parameters settings. According to our experience on large scale dataset, comparing to the small scale experiment on simulated dataset, DBTDP is not so sensitive to the parameters when the number of nodes of the networks is large, we empirically set $\gamma_{a,b} = 1000$ and $\beta_{a,a} = 10$ in this experiment. Figure 4 demonstrates the number of detected communities by DBTDP algorithm, which started from 2 and quickly converged to 7 at the first epoch and then maintained stable (epoch 2 & 3 are thus covered by epoch 4 in this figure).. To compare DBTDP with DBSM, we predefined the number of communities to 7 and 8 (7 was the number of detected communities by DBTDP) and compared DBTDP with DBSM-7 and DBSM-8, as shown in Figure 5 (a-d). Figure 5 demonstrates that in all 4 time epoch, DBTDP always achieved the lowest robustness score which showed that the communities detected by DBTDP were more robust and believable.

VIII. CONCLUSION

In this paper, we propose the Dynamic Stochastic Blockmodel with Temporal Dirichlet Process to detect communities and their evolution from dynamic networks. The DBTDP model considers networks arriving as a stream. We incorporate the Recurrent Chinese Restaurant Process with the stochastic blockmodel to adapt our model for network evolution. In particular, no prior knowledge is required to predefine the number of communities in our model. The communities in our proposed model can split, merge, retain, disappear or grow depending on the evolution. Furthermore, the Gibbs Sampling algorithm is proposed to optimize the posterior probability and determine the most probable value of the community assignments of nodes. The experiment results on simulated dataset shows that our proposed DBTDP model outperforms the state-of-art benchmark algorithm, especially when the number of communities changes with the data. The experiment results on the Flickr dataset also demonstrated the effectiveness and validity of our model.

REFERENCES

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg *et al.*, "Mixed Membership Stochastic Blockmodels," *J. Mach. Learn. Res.,* vol. 9, pp. 1981-2014, 2008.

[2] I. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts." pp. 551-556.

[3] M. Girvan, and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences,* vol. 99, no. 12, pp. 7821-7826, 2002.

[4] T. A. B. Snijders, and K. Nowicki, "Estimation and prediction for stochastic blockmodels for graphs with latent block structure," *Journal of Classification,* vol. 14, no. 1, pp. 75-100, 1997.

[5] K. Henderson, T. Eliassi-Rad, S. Papadimitriou *et al.*, "HCDF: A hybrid community discovery framework," *SDM, SIAM*, pp. 754-765.

[6] T. Berger-Wolf, and J. Saia, "A framework for analysis of dynamic social networks." pp. 523-528.

[7] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs." pp. 913-921.

[8] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, "Mining and Visualizing the Evolution of Subgroups in Social Networks." pp. 52-58.

[9] J. Sun, C. Faloutsos, S. Papadimitriou *et al.*, "Graphscope: parameter-free mining of large time-evolving graphs." pp. 687-696.

[10] Y.-R. Lin, Y. Chi, S. Zhu *et al.*, "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks," in Proceeding of the 17th international conference on World Wide Web, Beijing, China, 2008, pp. 685-694.

[11] W. Fu, L. Song, and E. Xing, "Dynamic mixed membership blockmodel for evolving networks." pp. 329-336.

[12] T. Yang, Y. Chi, S. Zhu *et al.*, "A bayesian approach toward finding communities and their evolutions in dynamic social networks." pp. 990-1001.

[13] K. T. Miller, and T. Eliassi-Rad, "Continuous time group discovery in dynamic graphs."

[14] J. Scott, *Social network analysis: A handbook*: Sage Publications Ltd, 2000.

[15] R. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of computational and graphical statistics,* vol. 9, no. 2, pp. 249-265, 2000.

[16] A. Ahmed, and E. Xing, "Dynamic non-parametric mixture models and the recurrent chinese restaurant process," *Proceedings of SDM 2008*, 2008.

[17] M. E. J. Newman, and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E,* vol. 69, no. 2, pp. 026113, 2004.

[18] B. Karrer, E. Levina, and M. Newman, "Robustness of community structure in networks," *Physical Review E,* vol. 77, no. 4, pp. 046119, 2008.

[19] Y. Chi, X. Song, D. Zhou *et al.*, "Evolutionary spectral clustering by incorporating temporal smoothness," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, San Jose, California, USA, 2007, pp. 153-162.

[20] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network." pp. 721-730.