

# Modeling the Dynamics of Composite Social Networks

Erheng Zhong<sup>†</sup>, Wei Fan<sup>‡</sup>, Yin Zhu<sup>†</sup>, Qiang Yang<sup>†‡</sup>

<sup>†</sup>Hong Kong University of Science and Technology, Hong Kong

<sup>‡</sup>Huawei Noah's Ark Lab, Hong Kong

{ezhong,yinz,qyang}@cse.ust.hk, david.fanwei@huawei.com

## ABSTRACT

Modeling the dynamics of online social networks over time not only helps us understand the evolution of network structures and user behaviors, but also improves the performance of other analysis tasks, such as link prediction and community detection. Nowadays, users engage in multiple networks and form a “composite social network” by considering common users as the bridge. State-of-the-art network-dynamics analysis is performed in isolation for individual networks, but users’ interactions in one network can influence their behaviors in other networks, and in an individual network, different types of user interactions also affect each other. Without considering the influences across networks, one may not be able to model the dynamics in a given network correctly due to the lack of information. In this paper, we study the problem of modeling the dynamics of composite networks, where the evolution processes of different networks are jointly considered. However, due to the difference in network properties, simply merging multiple networks into a single one is not ideal because individual evolution patterns may be ignored and network differences may bring negative impacts. The proposed solution is a nonparametric Bayesian model, which models each user’s common latent features to extract the cross-network influences, and use network-specific factors to describe different networks’ evolution patterns. Empirical studies on large-scale dynamic composite social networks demonstrate that the proposed approach improves the performance of link prediction over several state-of-the-art baselines and unfolds the network evolution accurately.

## Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications  
– Data Mining

## Keywords

Social Network Analysis, Transfer Learning

Permission to make digital or hard copies of all or part 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 bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

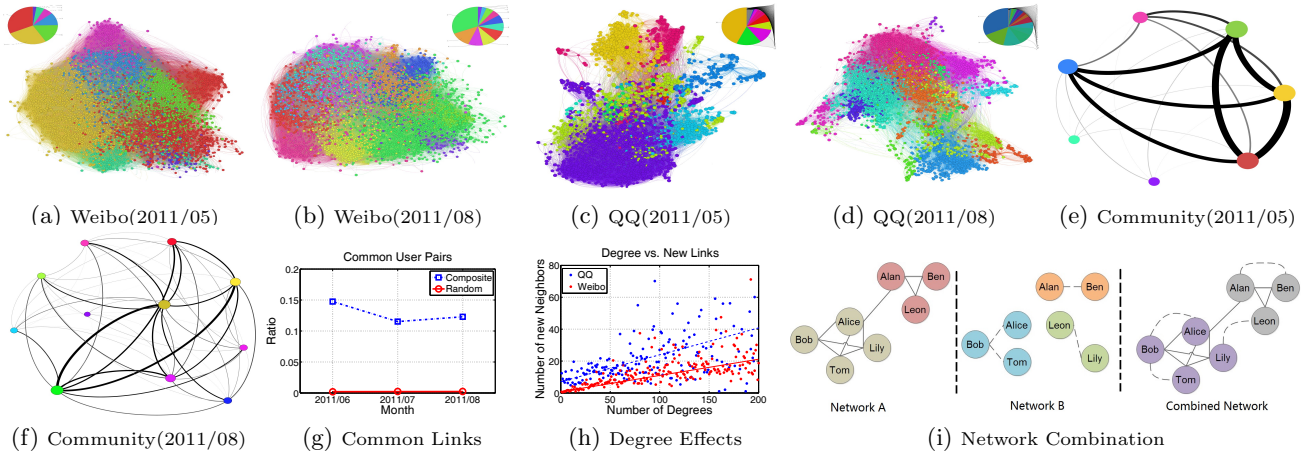
KDD’13, August 11–14, 2013, Chicago, Illinois, USA.

Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...\$15.00.

## 1. INTRODUCTION

With the growth of online social media, social network analysis has attracted much research interest with a broad range of applications. As an important research topic, modeling the dynamics of social networks can help people understand the evolution of network structures, e.g., the community evolution, the change of network statistics, e.g., diameter and clustering coefficient, and the shifting of users’ preferences, etc. In addition, considering the dynamic properties helps improve the performance of other tasks, such as link prediction [16] and community detection [14]. One important observation is that nowadays people join multiple networks for different applications. For example, users may use Facebook to connect with their friends but follow celebrities on Twitter. In addition, even in an individual network, users may have different interactions, such as “Forward (RT)” and “Mention (@)” on Twitter to share instant information. Consequently, by considering shared users, who can be identified by their unified IDs, as the bridges, different relationships or networks of interactions are nested together as composite networks [19].

Most existing research works only study the dynamics of individual networks and do not consider their intra-network correlations, but in practice, different networks can be highly correlated. As shown in Figure 1(g), the ratio of common links between Tencent’s Instant Messaging and Microblog networks (QQ vs. Tencent Weibo) is much higher than the one between two random graphs (0.143 vs. 0.007 respectively). In reality, individual networks reflect only partial aspects of users’ social activities, thus the information captured by individual networks may be incomplete. For example, one user builds more links on LinkedIn and interact less with her friends on Facebook when she just graduates. If we model the dynamics in each network independently, we cannot infer the graduation activity correctly, as people have different reasons to be inactive on Facebook and use LinkedIn when they just want to change jobs. Consequently, we may fail to model this user’s social dynamics, e.g., her social interactions on Facebook in the future, without knowing the fact of graduation. In addition, the reasons that cause the network to change can come from other networks. Two users without any common neighbors may follow each other on Twitter as they are familiar on Facebook, and one user may mention someone as she has forwarded many tweets of the user. Without considering the cross-network knowledge, one cannot correctly model these network dynamics. Furthermore, the data in an individual network may be very sparse, as users may have limited interactions when a net-



**Figure 1: Network Illustration.** (a)~(b) and (c)~(d) indicate the different numbers of communities and community memberships over time, where color indicates community and size presets community membership. (a)~(c) and (b)~(d) indicate the differences of community structures across networks. (e) and (f) indicate the different community relation strength over time. (g) presents the ratio of common links across networks, where the ratio is much higher than that between two random graphs, implying individual networks influence each other. (h) illustrates the correlation between the number of node degrees and the number of new links generated by the users. (i) illustrates the shortage of simply combining networks.

work is just beginning to form. Data sparsity makes models overfit the rare observations and thus have poor generality. Modeling the dynamics of composite networks collectively can lead to more accurate and comprehensive results, and help understand the interactions across different networks.

However, simply considering these nested networks as a single one does not work either, as different networks have different properties. Firstly, different networks reflect users' different social interests. Users in different networks may have overlapped but different social circles, as shown in Figure 1(i), where one user belongs to different communities in different networks. Secondly, different networks have different levels of density. For example, Network-A is denser than Network-B. Simply merging may bring unnecessary and even harmful knowledge to each individual network. For example, if we combine networks A and B, the network structure in Network-B is hidden and we cannot discover users' social interests in Network-B anymore. In addition, different types of networks have their own growth patterns, e.g., different shrinkage ratios of diameters, degree distributions, etc. For example, the average number of users' neighbors in Weibo network is much higher than instant messaging network (QQ). As shown in Figure 1(a)~(d), communities in Weibo are larger than those in QQ and the evolution process in QQ is more stable.

To solve the aforementioned problems, we propose a nonparametric Bayesian model which integrates network evolution over composite social networks directly. On one hand, it utilizes an adaptive and time-dependent prior for each user to denote different users' latent interests that decide users' community memberships globally. Since this prior is related to all nested networks and thus it can be considered as a bridge to capture the cross-network dynamics. On the other hand, it introduces temporal network-specific factors to encode growth patterns in each individual network, which adjusts users' concrete time-evolving community memberships. These two priors are combined together to generate a hybrid prior and build dynamic mixed membership models in each individual network, where users' community memberships as well as the community relations are exploited to generate links between them.

**Table 1: Definition of notations**

Group	Notation	Notation Description
Data	$\mathcal{G} = \{\mathbb{G}^t\}_{t=1}^T$	Sequence of Composite Networks
	$\mathbf{U} = \{u_i\}_{i=1}^N$	User set of $\mathcal{G}$
	$\mathbf{E}^t = \cup\{E_i^t\}_{i=1}^\ell$	Link set of $\mathbb{G}^t$
Model	$\text{Dir}(\cdot), \text{Mult}(\cdot)$	Dirichlet and Multinomial distributions
	$\text{Bern}(\cdot), \text{Beta}(\cdot)$	Bernoulli and Beta distributions
	$\text{Gam}(\cdot), \text{N}(\cdot)$	Gamma and Gaussian distributions
	$\pi, v, \mathbf{x}$ and $\omega$	Latent variables
	$\gamma, \lambda_u, \lambda_d$ and $\lambda$	Hyperparameters
	$K_d^t$	Number of communities at time $t$

To allow community structures vary over time and across networks, we use a nonparametric generative process for mixed memberships with the hybrid prior. A state-space model is applied on the top to model the cross-network dynamics. Specifically, users' latent interests, network-specific factors and community relations are changing over time. As time evolves, the observations of one network at a given time stamp can be encoded in users' latent interests and affect the structures of other networks in the next time stamp, while the network-specific factors and community relations allow every individual networks to have own growth patterns.

**Problem Formulation** The notations are summarized in Table 1. We consider a sequence of composite networks, denoted by  $\{\mathbb{G}^t\}_{t=1}^T$ , where  $\mathbb{G}^t = \{G_i^t = (U_i, E_i^t)\}_{i=1}^\ell$  is the composite network observed at time  $t$ . In each  $\mathbb{G}^t$ ,  $G_i^t$  is the  $i$ -th individual network,  $U_i$  is the user set of  $G_i^t$ ,  $E_i^t$  is the user relationship of  $U_i$  at time  $t$  and  $\ell$  is the number of single networks. We define the complete user set as  $\mathbf{U}$  and the link set at time  $t$  as  $\mathbf{E}^t$ , where  $\mathbf{U} = \cup\{U_i\}_{i=1}^\ell = \{u_j\}_{j=1}^N$  and  $\mathbf{E}^t = \cup\{E_i^t = \{e_{ij}^t\}_{j=1}^{M_i^t}\}_{i=1}^\ell$ . We assume the set of nodes  $\mathbf{U}$  is constant. Although new nodes can join networks and existing nodes can leave networks, we can still include these nodes in  $\mathbf{U}$  but consider them to be inactive. The goal is to construct the composite network at time  $T+1$  given the network sequence  $\{\mathbb{G}^t\}_{t=1}^T$ .

## 2. INFINITE TIME-EVOLVING COMPOSITE NETWORK MODELING

We propose a novel nonparametric Bayesian model, Infinite Time-evolving Composite Network Models (ITCom), to cap-

ture the dynamics of composite social networks. In the following, we firstly describe the foundation of ITCom, Mixed Membership Stochastic Blockmodels (MMSB) [1], and then present the details of three extensions: infinite, dynamic and knowledge transfer. Finally, we propose an efficient Sequential Monte Carlo (SMC) method to infer the latent variables and construct ITCom. To help understand the motivation of the proposed model, we visualize two sub networks from Tencent. These networks contain users who are 2-hops away from the first author of the paper, as well as their relationships in Tencent’s instant messaging network QQ and Weibo network, as shown in Figure 1.

## 2.1 Background of MMSB

As a foundation of ITCom, MMSB assumes that each user  $u_i \in \mathbf{U}$  possesses a latent mixture of  $K$  roles, which determine the membership of  $K$  communities in the network  $\mathcal{G}$ . We denote this role mixture as a normalized  $K \times 1$  vector  $\pi_i$ . In MMSB, these vectors are drawn from some priors  $p(\pi)$ , such as Dirichlet distribution [1] and Logistic-Normal distribution [6]. MMSB generates a  $K \times K$  community relation matrix  $\mathbf{B}$ , which represents the probability of having a connection from a user in a community to another user in another community. Given the vector  $\pi_i$  of each user  $u_i$ , the network edge  $e_{ij}$  is generated stochastically as follows:

- For each pair of users  $(u_i, u_j) \in \mathbf{E}$  in the network  $\mathcal{G}$ :
  - Draw indicator for  $u_i$ ,  $\mathbf{z}_{ij} \sim \mathbf{Mult}(\pi_i)$
  - Draw indicator for  $u_j$ ,  $\mathbf{z}_{ji} \sim \mathbf{Mult}(\pi_j)$
  - Sample the link,  $e_{ij} \sim \mathbf{Bern}(\mathbf{z}_{ij}^T \mathbf{B} \mathbf{z}_{ji})$

where  $\mathbf{z}_{ij}$  and  $\mathbf{z}_{ji}$  are two  $K \times 1$  unit indicator vectors for the sender  $u_i$  and the receiver  $u_j$  respectively. As MMSB only models static and individual networks with fixed number of communities, as follows, we evolve MMSB step-by-step to model the dynamics of composite networks based on evidences and observations from real-world social networks.

## 2.2 Infinite Modeling

In reality, with the evolution of networks, communities can come and go. For example, in social networks, e.g., Facebook, a set of users can form a community when they join the same school and get familiar with each other; in social media, e.g., Twitter, a discussion group may dismiss when a hot topic is out-of-date. An example can be found in Figure 1(a) and (b), where the number of communities as well as the community sizes are different over time in Tencent Weibo. Different from previous research works [6, 4], which keep the community structure unchanged in different time stamps, ITCom allows communities to vary over time and determines the number of communities automatically. Inspired by Chinese Restaurant Process [2] and its successful applications on topic models [3] and MMSB [18], we introduce a stick-breaking prior on each  $\pi_i$  to let  $\sum_{k=1}^{\infty} \pi_{ik} = 1$ . The construction process is as follows:

$$\pi_{ik} = v_{ik} \prod_{j=1}^{k-1} (1 - v_{ij}) \quad (1)$$

where  $v_{ij}$  is the latent factor that needs to be estimated. This process can be understood as follows. Let  $\pi_{i1} = v_{i1}$  and  $1 - \pi_{i1}$  be the remainder of the stick after chopping off the length  $v_{i1}$ . To calculate the length  $\pi_{i2}$ , draw  $v_{i2}$  randomly and chop off this fraction of the remainder of the stick, giving  $\pi_{i2} = v_{i2}(1 - v_{i1})$ . This process is repeated and thus  $v_{ik}$  is the fraction to chop off from the stick’s remainder, and  $\pi_{ik}$

is the length of the stick that was chopped off. Thus, the probability to generate a community  $K + 1$  can be calculated as  $1 - \sum_{k=1}^K \pi_{ik}$ . In addition,  $v_{ik}$  evolves over time, and thus a community  $k$  can die if  $v_{ik}$  becomes too small for each user  $u_i$ . To avoid overfitting and incorporate domain knowledge, we apply a logistic-normal prior on  $v_i$ . Formally, we have

$$\epsilon_i \sim \mathbf{N}(\mu, \lambda^{-1}I) \quad v_i = \frac{1}{1 + \exp(-\epsilon_i)} \quad (2)$$

where  $\mu$  is the mean of all  $\epsilon_i$  and  $\lambda$  is used to control the precision. As stated below, the logistic-normal prior allows us to incorporate global, network-specific and time-dependent knowledge naturally by adjusting  $\mu$  over time and networks.

## 2.3 Knowledge Transfer Across Networks

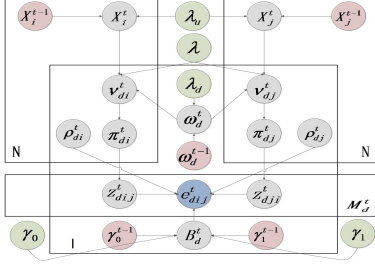
Common knowledge can be transferred by overlapping users across individual networks while each individual network contains specific factors. For example, Facebook reflects users’ friendship in daily life, while Twitter is more about users’ interests on daily news. But at the same time, the same users in different individual networks can also have similar neighbors as they have similar interests over different dimensions like friendship, movies, etc. This phenomenon can be formulated as: different individual networks in a composite network may have different community structures, and each user follows community memberships over different networks while keeps some intrinsic features unchanged. One example can be found in Figure 1(a), (c) and (g), where QQ and Weibo networks have different community structures and users’ community memberships but share many common links. To model these observations, we decompose  $v_i$  into two parts, one representing each user’s latent features and the other reflecting network-specific factors. Let  $\mathbf{x}_i \in R^{1 \times D}$  denote the latent features of user  $u_i$  and  $\omega_d \in R^{D \times K_d}$  denote the network-specific factors of the  $d$ -th individual network, where  $K_d$  is the corresponding number of communities.  $\mathbf{x}_i$  encodes each user’s latent interests while  $\omega_d$  maps users’ latent interests to network-dependent communities. For each user  $u_i$  in  $G_d$ , we set  $\mu_{di} = \mathbf{x}_i \omega_d$ . Thus, cross-network influence can be captured by  $\mathbf{x}_i$  while network-dependent properties can be described by  $\omega_d$ . To avoid overfitting, we assign Gaussian priors to  $\mathbf{x}_i$  and  $\omega_d$ , the generation process of  $\pi_i^d$  is as follows:

$$\begin{aligned} \omega_d &\sim \mathbf{N}(\mathbf{0}, \lambda_d^{-1}I) & \mathbf{x}_i &\sim \mathbf{N}(\mathbf{0}, \lambda_u^{-1}I) \\ \epsilon_{di} &\sim \mathbf{N}(\mathbf{x}_i \omega_d, \lambda^{-1}I) & v_{di} &= \frac{1}{1 + \exp(-\epsilon_{di})} \\ \pi_{dik} &= v_{dik} \prod_{\ell=1}^{k-1} (1 - v_{di\ell}) & k &= 1 \dots K_d \end{aligned}$$

The number of communities is network-dependent and the communities with the same index in different networks are not necessary identical. This is consistent with the real-world data in Figure 1, where the number of communities and community sizes of QQ and Weibo are different.

## 2.4 Dynamic Modeling

One important property of social networks is its dynamics, where communities can come into being and phase out, users’ community membership can change, and users can form and deform links. Examples can be found in Figure 1: (e) and (f) show the different community structures and their relation strengths over time; (a) and (b) present the different community memberships over time; (h) shows that the levels of users’ activities to generate links are closely related



**Figure 2: The Graphical Representation of ITCOM.** Blue nodes indicate observed variables, greens present hyper priors, reds mean variables in last time stamp and grays indicate latent variables need to be inferred.

to their degrees, which evolve over time as well. In mixed membership models, these can be manipulated by changing the community compatibility  $\mathbf{B}$  and users' community memberships  $\pi$ , and varying the number of users' links in different time stamps respectively. Firstly, with the shifting of users' interests, the connection between different communities changes over time. Considering that older interactions have relatively smaller impacts, we use exponential decay to modify the priors of  $\mathbf{B}$ . Let  $n_{ij1}^t$  denote the times that users build links between the  $i$ -th and the  $j$ -th communities at time  $t$  and  $n_{ij0}^t$  denote the number of user pairs that do not interact with each other but select the  $i$ -th and  $j$ -th communities at time  $t$ . With a kernel parameter  $\kappa$ , we have

$$\tilde{\gamma}_{ij0}^t = \sum_{h=1}^{t-1} \exp\left(\frac{h-t}{\kappa}\right) n_{ij0}^h, \quad \tilde{\gamma}_{ij1}^t = \sum_{h=1}^{t-1} \exp\left(\frac{h-t}{\kappa}\right) n_{ij1}^h \quad (3)$$

Then the generation of the community compatibility matrix at time  $t$  is  $\mathbf{B}^t \sim \text{Beta}(\gamma_0 + \tilde{\gamma}_{ij0}^t, \gamma_1 + \tilde{\gamma}_{ij1}^t)$ . Note that  $\gamma_0$  and  $\gamma_1$  here enforce that the connection between new and existing communities can appear as time goes by. Secondly, users' community memberships shift over time as well. This can be understood from two aspects: (1). users' latent interests  $\mathbf{x}_i$  change over time and (2). the mapping from latent interests to communities changes with the evolution of community structures. Motivated by state-space models for series data, we let

$$\omega_d^t \sim \mathbf{N}(\omega_d^{t-1}, \lambda_d^{-1}I) \quad \mathbf{x}_i^t \sim \mathbf{N}(\mathbf{x}_i^{t-1}, \lambda_u^{-1}I) \quad (4)$$

That means, users' latent interests as well as the interest-community mapping at time  $t$  are shifted from those in the last time stamp. Finally, users' activity levels can be very different in different time stamps. According to the study in [10], the probability that one user generates one link with time gap  $\delta$  is  $\delta^{-\alpha_d} \exp(-\beta_d n_i^d \delta)$ , where  $n_i^d$  is the number of degrees of user  $u_i$  in the network  $G_d$ , and  $\alpha_d$  and  $\beta_d$  are two network-dependent parameters. In other words, in unit time, the expected number of interactions for user  $u_i$  is  $\exp(\beta_d n_i^d)$ . Instead of setting the number of interactions explicitly, we introduce a sparsity parameter  $\rho_i^d = \frac{\exp(\beta_d n_i^d)}{N_d}$  to characterize the source of interaction, where  $N_d$  is the number of users in  $G_d$ . Then, we down-weight the probability of successful interaction as

$$e_{ij}^d \sim \text{Bern}(\rho_i^d \rho_j^d \mathbf{z}_{ij}^T \mathbf{B}_d \mathbf{z}_{ji}) \quad (5)$$

Note that the number of node degree varies over time, thus  $\rho_i^d$  changes with time as well.

## 2.5 Model Summary

We now put all the pieces together and give the full generative process of ITCOM as follows. The graphical model in one time stamp can be found in Figure 2.

- For each user  $u_i \in \mathbf{U}$ : Sample  $\mathbf{x}_i^0 \sim \mathbf{N}(\mathbf{0}, \lambda_u^{-1}I)$
- For each network  $G_d^0 \in \mathcal{G}$ : Sample  $\omega_d^0 \sim \mathbf{N}(\mathbf{0}, \lambda_d^{-1}I)$
- Set each  $n_{ij0}^0 = 0$  and  $n_{ij1}^0 = 0$
- From  $t = 1$  to  $T$ 
  - For each network  $G_d^t$ :
    - \* Draw a  $D \times K_d^t$  feature matrix  $\omega_d^t \sim \mathbf{N}(\omega_d^{t-1}, \lambda_d^{-1}I)$
    - \* Draw  $\mathbf{B}_d^t \sim \text{Beta}(\gamma_0 + \tilde{\gamma}_{d0}^t, \gamma_1 + \tilde{\gamma}_{d1}^t)$
  - For each user  $u_i \in \mathbf{U}$ :
    - \* Draw a  $D$  latent feature vector  $\mathbf{x}_i^t \sim \mathbf{N}(\mathbf{x}_i^{t-1}, \sigma_u^2 I)$
    - \* Set sparsity parameter  $\rho_i^d = \frac{\exp(\beta_d n_i^d)}{N_d}$
    - \* For each network  $G_d^t$ :
      - $v_{di}^t = \frac{1}{1 + \exp(-\epsilon_{di}^t)}$ ,  $\epsilon_{di}^t \sim \mathbf{N}(\mathbf{x}_i^t \omega_d^t, \lambda^{-1}I)$
      - Draw  $\pi_{di}^t$ :  $\pi_{dik}^t = v_{dik}^t \prod_{\ell=1}^{k-1} (1 - v_{di\ell}^t)$
  - For each pair  $(u_i, u_j) \in E_d^t$  in  $G_d^t$ :
    - \* Draw indicator for  $u_i$ ,  $\mathbf{z}_{dij}^t \sim \text{Mult}(\pi_{id}^t)$
    - \* Draw indicator for  $u_j$ ,  $\mathbf{z}_{dji}^t \sim \text{Mult}(\pi_{jd}^t)$
    - \* Sample the link,  $e_{dij}^t \sim \text{Bern}(\rho_i^d \rho_j^d \mathbf{z}_{dij}^t \mathbf{B}_d^t \mathbf{z}_{dji}^t)$

## 2.6 Model Building

Given observed links for some or all node pairs, we employ a Sequential Monte Carlo (SMC) method to draw samples from the latent variables' posterior distribution and a Maximum a posteriori (MAP) method to estimate the hyperparameters. SMC incrementally runs a fast batch sampling method over the data at epoch  $t$  given the state at earlier epochs. As the priors and posteriors are not conjugate, we divide the inference process into three phrases to reduce the computational cost. It firstly samples the community assignments of every links by assuming the community membership  $\pi$  is given, then infers users' membership over communities, and finally estimates the hierarchical priors.

**Sampling Community Assignments** Due to the challenge of infinite number of communities, we employ retrospective sampling [15] to approximate the true infinite model. Let  $\gamma_{d0}^t = \gamma_0 + \tilde{\gamma}_{d0}^t$  and  $\gamma_{d1}^t = \gamma_1 + \tilde{\gamma}_{d1}^t$ , and then the pairwise community assignments in the  $d$ -th network at time  $t$  can be written as

$$\begin{aligned} p(\mathbf{z}_d^t | \pi_d^t, \gamma_{d0}^t, \gamma_{d1}^t, E_d^t) &\propto p(E_d^t | \mathbf{z}_d^t, \gamma_{d0}^t, \gamma_{d1}^t) p(\mathbf{z}_d^t | \pi_d^t) \\ &= \int p(E_d^t | \mathbf{z}_d^t, \mathbf{B}_d^t) dp(\mathbf{B}_d^t | \gamma_{d0}^t, \gamma_{d1}^t) p(\mathbf{z}_d^t | \pi_d^t) \\ &\propto \prod_{k, k'} \frac{B(\gamma_{d0}^t + \gamma_{d1}^t + \mathbf{n}_{d,k,k'}^t)}{B(\gamma_{d0}^t + \gamma_{d1}^t)} \prod_{u_i} \pi_{dik}^t \prod_{u_j} \pi_{djk'}^t \end{aligned}$$

where  $B(\omega) = \frac{\prod_k \Gamma(\omega_k)}{\Gamma(\sum_k \omega_k)}$  and  $\mathbf{n}_{d,k,k'}^t$  denotes the total number of pairwise community assignments between community  $k$  and  $k'$  at time  $t$  in the  $d$ -th network. As the conjugate property between Beta and Multinomial distributions,  $\mathbf{B}_d^t$  can be marginalized. The posterior of community assignments can be defined as

$$\begin{aligned} p(\mathbf{z}_{dij}^t = k, \mathbf{z}_{dji}^t = k' | \pi_{di}^t, \gamma_{d0}^t, \gamma_{d1}^t, \mathbf{z}_{d,-(i,j)}^t, E_d) &\quad (6) \\ \propto \frac{\tilde{\pi}_{dik}^t \tilde{\pi}_{djk'}^t}{n_{d,k,k',0}^{t,-(i,j)} + \gamma_{d0}^t} \frac{(n_{d,k,k',0}^{t,-(i,j)} + \gamma_{d0}^t)^{1-y_{dij}^t} (n_{d,k,k',1}^{t,-(i,j)} + \gamma_{d1}^t)^{y_{dij}^t}}{n_{d,k,k',0}^{t,-(i,j)} + \gamma_{d0}^t + n_{d,k,k',1}^{t,-(i,j)} + \gamma_{d1}^t} \end{aligned}$$

where  $\mathbf{z}_{d,-(i,j)}^t$  denotes the set of community assignments without two assignments over the link between  $u_i$  and  $u_j$ , and  $n_{d,k,k',y}^t$  represents the total number of links in type  $y$  with  $(k, k')$  as the participating communities in the  $d$ -th network at time  $t$ . In addition,  $y_{dij}^t$  denotes the sign of the link  $e_{dij}^t$ , where  $y_{dij}^t = 1$  represents that  $u_i$  and  $u_j$  are linked at time  $t$  and  $y_{dij}^t = 0$  represents that  $u_i$  and  $u_j$  will not build a link between each other. Importantly, assuming the number of current communities is  $K$ , if  $1 \leq k \leq K$  then  $\tilde{\pi}_{dik}^t = \pi_{dik}^t$ ; otherwise  $\tilde{\pi}_{dik}^t = 1 - \sum_{k=1}^K \pi_{dik}^t$  and  $n_{k,k',0}^{t,d,-(i,j)} = n_{k,k',1}^{t,d,-(i,j)} = 0$ . Eq.(6) represents the posterior probability of selecting community  $k$  if  $k \leq K$  but represents the aggregate posterior probability of the infinite ‘‘tail’’ of communities with indexes greater than  $K$  if  $k > K$ .

**Sampling the Parameter  $\mathbf{v}$**  Given the community assignments, we employ the Metropolis-Hastings algorithm to sample  $v$  for community memberships independently. Given users’ latent features  $\mathbf{x}$  and network factors  $\omega$ , to sample a new community activation  $v_{di}^*$  for each user  $u_i$  in the  $d$ -th network, we define the accept ratio of  $v_{di}^*$ ,  $A(v_{di}^*, v_{di})$ , as

$$A(v_{di}^*, v_{di}) = \prod_{u_j} \frac{p(\mathbf{z}_{dij}^t | v_{di}^*) p(\mathbf{z}_{dji}^t | v_{di}^*)}{p(\mathbf{z}_{dij}^t | v_{di}) p(\mathbf{z}_{dji}^t | v_{di})} = \prod_{k=1}^K \left( \frac{\pi_{dik}^t}{\pi_{dik}^t} \right)^{D_{dik}} \quad (7)$$

where  $D_{dik}$  denotes the total number of indicators attached to user  $u_i$  assigned to community  $k$  in the network  $G_d$ . Then, at each time, we sample  $v_{di}^*$  from  $\sim \mathbf{N}(v_{di}^* | \mathbf{x}_i \omega_d, \lambda^{-1} I)$  and accept it with probability  $\min(A(v_{di}^*, v_{di}), 1)$ .

**Sampling  $\mathbf{x}$  and  $\omega$**  Given  $v$ , for each time stamp  $t$ , we can infer the hierarchical priors:  $\omega_d^t$  for each network and  $\mathbf{x}_i^t$  for each user  $u_i$ . To find the optimal values, we first define the union distribution over  $v$  and priors as

$$p(v^t, \omega^t, \mathbf{x}^t) = \prod_{d=1}^{\ell} \prod_{i=1}^N \mathbf{N}(v_{di}^t | \mathbf{x}_i^t \omega_d^t, \lambda^{-1} I) \quad (8)$$

$$\prod_{i=1}^N \mathbf{N}(\mathbf{x}_i^t | \mathbf{x}_i^{t-1}, \lambda_u^{-1} I) \prod_{d=1}^{\ell} \mathbf{N}(\omega_d^t | \omega_d^{t-1}, \lambda_d^{-1} I)$$

where  $\mathbf{x}_i^0 = \mathbf{0}$  and  $\omega_d^0 = \mathbf{0}$ . By calculating the derivatives of the log of likelihood with respect to  $\omega_d^t$  and  $\mathbf{x}^t$  and set them as zero, we obtain the sampling equations as

$$\omega_d^t \sim \mathbf{N}((\lambda_d I + \mathbf{x}^{tT} \mathbf{x}^t)^{-1} \mathbf{x}^{tT} v^t + \omega_d^{t-1}, \lambda_d^{-1} I) \quad (9)$$

$$\mathbf{x}^t \sim \mathbf{N}(v^t \omega_d^{tT} (\lambda_u I + \sum_d \omega_d^{tT} \omega_d^t)^{-1} + \mathbf{x}^{t-1}, \lambda_u^{-1} I) \quad (10)$$

where  $\mathbf{x}^t$  is a  $N \times D$  matrix, each row of which is  $\mathbf{x}_i^t$ .

**Parameter Estimation** As follows, we propose a maximum a posteriori (MAP) method to estimate the hyperparameters, including  $\lambda$ ,  $\lambda_u$  and  $\lambda_d$ . To avoid overfitting, we place Gamma priors so that their prior distributions are conjugate with the likelihoods. For  $\lambda$ , we have

$$p(\lambda | v, \mathbf{x}, \omega, a, b) \quad (11)$$

$$\propto \mathbf{Gam}(\lambda | a, b) \prod_{t=1}^T \prod_{d=1}^{\ell} \prod_{i=1}^N \mathbf{N}(v_{di}^t | \mathbf{x}_i^t \omega_d^t, \lambda^{-1} I)$$

$$\propto \mathbf{Gam}\left(\lambda \middle| \frac{1}{2} NT\ell K + a, \frac{1}{2} \sum_{k,t,d,i} (v_{dik}^t - [\mathbf{x}_i^t \omega_{dk}^t])^2 + b\right)$$

Similarly, we obtain the sampling equations for  $\lambda_u$  and  $\lambda_d$ :

$$p(\lambda_u | \mathbf{x}, a_u, b_u) \propto \mathbf{Gam}\left(\lambda_u \middle| \frac{1}{2} NT D + a_u, \frac{1}{2} \sum_{f,t,i} \mathbf{x}_{if}^t\right) \quad (12)$$

$$p(\lambda_d | \omega, a_d, b_d) \propto \mathbf{Gam}\left(\lambda_d \middle| \frac{1}{2} T\ell K + a_d, \frac{1}{2} \sum_{k,t,d} \omega_{dk}^t\right) \quad (13)$$

**Framework** We put all the components above into an SMC framework. Firstly, we set priors as arbitrary values, then initialize the latent variables and finally sample the community assignments. Consequently, it alternatively updates community assignments, community memberships, priors and hyperparameters in each round of inference:

- Update community assignments using Eq.(6)
- Update community activations  $v$  using Eq.(7)
- Update network-specific priors  $\omega$  using Eq.(9)
- Update user priors  $\mathbf{x}$  using Eq.(10)
- Update hyperparameters using Eq.(11), (12) and (13)

This process is repeated until convergence. Besides, the sparsity ratio  $\rho$  can be estimated independently by maximizing likelihood [10]. To avoid overfitting and reduce the computational cost, we update latent variables periodically. Specifically, we update  $v$  every 5 iterations, both  $\lambda$  and  $\mathbf{x}$  every 10 iterations, and hyperparameters every 20 iterations. In addition, as in most real-world datasets, we observe only positive links. For each user, we randomly sample other users who are more than two-hops away and build negative links to the current user. The number of negative links is kept in the same order to that of positive links.

**Time Complexity** At each iteration, SMC needs to look up all  $M$  links in nested networks to pick community assignments and then update  $v$  with time  $O(T\ell NKD)$  where  $N$  is the number of users. Metropolis-Hastings sampler spends  $O(TND)$  on updating  $\mathbf{x}$  and  $O(T\ell K)$  on updating  $\omega$ . Consequently, the updating of hyperparameters takes  $O(T\ell NK)$ . Typically,  $T\ell K$ ,  $TND$  and  $T\ell NK$  are much smaller than  $T\ell NKD$ . Thus, with  $I$  iterations, the whole complexity is  $O(I(MK + T\ell NKD))$  which linearly increases with the number of links  $M$  and users  $N$ . Following the similar idea in [19], that partitions data into multiple machines for parallel computing and then combines pieces of results, the inference process can be implemented on Map/Reduce straight forward to cope with large datasets efficiently.

### 3. EXPERIMENTS

We evaluate ITCom, on eight large-scale datasets from Tencent, Sina Weibo<sup>1</sup>, Xiaonei, Facebook/Twitter<sup>2</sup>, Github<sup>3</sup>, Stackoverflow<sup>4</sup> and Epinion<sup>5</sup>. Most datasets can be downloaded from the URLs in footnote and the pre-processed Stackoverflow dataset is also available<sup>6</sup>. The tasks include (1) link prediction, that predicts who will interact with each other in a given time stamp, and (2) macro-evolution, which predicts changes of networks’ statistics, e.g., clustering coefficients and degree distribution, etc. We adopt Mean Average Precision (MAP) as the evaluation criterion. It measures how well the algorithm ranks new links above non-existing links. We set  $D = 5$ ,  $\kappa = 1$  and other hyperparameters as 1.0. The number of iterations is 1000.

In the link prediction task, we compare ITCom with Mixed Membership Stochastic Blockmodels (MMSB) [1], dynamic MMSB (dMMSB) [6], Nonparametric Metadata Dependent Relational Model (NMDR) [9], dynamic Infinite Relational Model (dIRM) [8] and Tensor Factorization (TF) [7].

<sup>1</sup><http://www.wise2012.cs.ucy.ac.cy/challenge.html>

<sup>2</sup><http://socialnetworks.mpi-sws.org>

<sup>3</sup><http://www.githubarchive.org/>

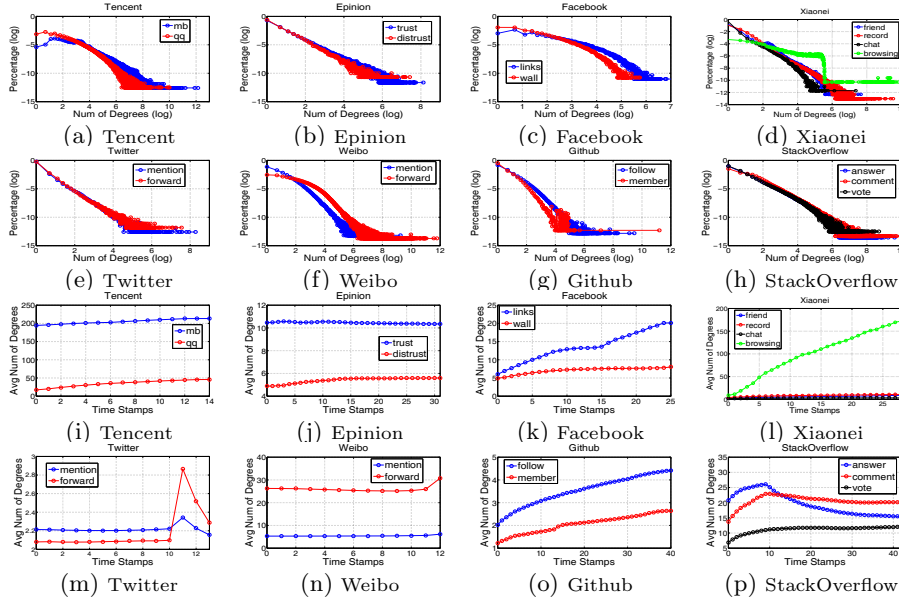
<sup>4</sup><http://meta.stackoverflow.com/>

<sup>5</sup><http://konect.uni-koblenz.de/networks/epinions>

<sup>6</sup><http://www.cse.ust.hk/~ezhong/dataset/so.zip>

**Table 2: Summary of Data Characteristics**

Collections	#User	#Relations/Interactions	Starting time	#T	Types of Relations/Interactions
Tencent	$\sim 1M$	$\sim 110M$	2011/05	12 weeks	Instant Messaging (QQ), Microblog Following (MB)
Epinion	$\sim 0.1M$	$\sim 0.8M$	2001/01	32 months	Trust, Distrust
Facebook	$\sim 0.06M$	$\sim 1.8M$	2006/09	25 months	Link, Wall Posting
Xiaonei	$\sim 0.5M$	$\sim 32M$	2012/04	28 weeks	Recording, Browsing, Chat, Friends Application
Twitter	$\sim 0.3M$	$\sim 0.9M$	2009/09	13 months	Forwarding, Mention
Sina Weibo	$\sim 6M$	$\sim 320M$	2009/12	25 months	Forwarding, Mention
Github	$\sim 0.05M$	$\sim 1M$	2012/03	43 weeks	Following, Collaborating
StackOverflow (SO)	$\sim 0.8M$	$\sim 33M$	2008/11	43 months	Answering, Commenting, Voting


**Figure 3: Degree Distribution ((a)~(h)) and Density Evolution ((i)~(p))**

MMSB is the basic baseline, dMMSB is a dynamic extension of MMSB, NMDR is a nonparametric extension of MMSB, dIRM is a dynamic and nonparametric extension of MMSB, and TF models multiple relations and time factors together but it does not model the network differences and in it the number of communities is fixed. For MMSB, dMMSB and TF, we set the number of communities  $K$  as 50 while NMDR, dIRM and ITCOM can determine  $K$  automatically. We set other parameters as the default values in original papers. For macro-evolution, we introduce the Microscopic Evolution (ME) model [10] as the baseline.

### 3.1 Dataset Description

Each dataset contains a composite network, where each composite network contains multiple individual networks. According to different link types, these datasets can be classified as relational networks, e.g., Tencent and Epinion, and interaction networks (the remaining six datasets), where user pairs in relational networks are distinct but users in interaction networks can interact with each other multiple times. Specifically, Tencent collection contains an instant messaging network (QQ) and a Weibo network (MB). In Epinion, people can build trust or distrust relations with others. Facebook collection captures users' friendships and wall posting actions. In Xiaonei collection, users can leave records, visit friends' homepages, chat with friends, and send friends applications. For Twitter and Sina Weibo, people can forward (RT) and mention (@) others' tweets. In Github, users follow each other as well as being collaborators in different projects. In Stackoverflow, people can answer others' questions, comment on others' answers and

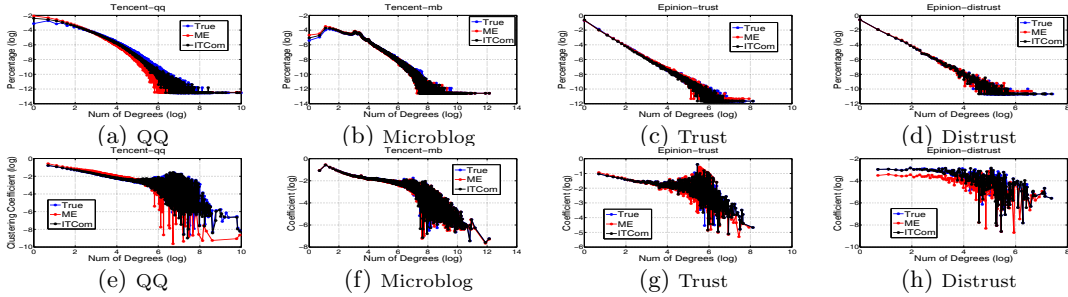
vote others' posts. Their statistics can be found in Table 2. Their degree distributions in the last time stamp and density evolutions are plotted in Figure 3. Although the degree distributions all follow the power-law distribution and networks become denser as time evolves, different individual networks in each collection have similar but different properties. It also implies that simply merging networks does not work, where networks' specific patterns can be hidden. In summary, these datasets come from different applications with different scales and time lengths. The correlations between individual networks in different datasets are very different. For example, forwarding and mention may have highly positive correlation while trust and distrust relations are strongly negative correlated. This variety property makes the experiments convinced. To crawl data, we employ random walk based sampling method to select sub networks in Tencent and Xiaonei networks and extract relational knowledge from whole public data dumps of other datasets. In each dataset, users in different individual networks can be identified by unified user identity, such as the QQ number in the Tencent collection. In the link prediction task, the networks at the previous  $T - 1$  time stamps are considered as the model input and the output is the prediction at time  $T$ . To simulate the evolution process, we consider the network series  $\{G_t\}_{t=1}^{T/2}$  and evolve it from  $t = T/2, \dots, T$ .

### 3.2 Interaction/Friendship Prediction

Table 3 summarizes the MAP of all baselines in the link prediction task. ITCOM consistently outperforms other baselines on MAP in all networks except Tencent's Microblog against TF. The MAP of ITCOM is at least 0.02 higher than

**Table 3: Performance Comparisons on MAP**

Networks		MMSB	MMSB-C	dMMSB	dMMSB-C	NMDR	NMDR-C	dIRM	dIRM-C	TF	ITCom
Tencent	QQ	0.5097	0.5034	0.5135	0.5211	0.5131	0.5055	0.5132	0.5127	0.5241	<b>0.5599</b>
	MB	0.2154	0.2151	0.2255	0.2397	0.2473	0.2293	0.2339	0.2337	<b>0.2551</b>	0.2549
Epinion	Trust	0.7235	0.6937	0.7282	0.7112	0.7251	0.7010	0.7205	0.7005	0.7300	<b>0.7375</b>
	Distrust	0.7020	0.6738	0.7151	0.6864	0.6996	0.6829	0.6945	0.6937	0.7001	<b>0.7116</b>
Facebook	Link	0.7298	0.7372	0.7360	0.7376	0.7330	0.7339	0.7337	0.7345	0.7358	<b>0.7469</b>
	Wall	0.6873	0.6773	0.6981	0.6956	0.6927	0.6954	0.6975	0.6972	0.6963	<b>0.7029</b>
Xiaonei	Record	0.5112	0.5290	0.5219	0.5211	0.5196	0.5211	0.5193	0.5196	0.5191	<b>0.5477</b>
	Browsing	0.2038	0.2070	0.2140	0.2125	0.2098	0.2131	0.2185	0.2186	0.2188	<b>0.2477</b>
	Chat	0.2813	0.2661	0.3010	0.2977	0.3003	0.2969	0.2960	0.2965	0.2960	<b>0.3306</b>
Twitter	Friend	0.3490	0.4092	0.3206	0.3968	0.3874	0.4095	0.3788	0.4171	0.4119	<b>0.4535</b>
	Forwarding	0.6945	0.7003	0.7204	0.7033	0.7082	0.7043	0.7218	0.7131	0.7135	<b>0.7361</b>
Weibo	Mention	0.7538	0.7450	0.7799	0.7743	0.7694	0.7701	0.7716	0.7728	0.7756	<b>0.8059</b>
	Forwarding	0.5562	0.5563	0.5876	0.5819	0.5831	0.5836	0.5913	0.5915	0.5933	<b>0.6233</b>
Github	Mention	0.6652	0.6653	0.6775	0.6817	0.6781	0.6801	0.6834	0.6834	0.6826	<b>0.7222</b>
	Following	0.7447	0.7522	0.7662	0.7661	0.7649	0.7649	0.7685	0.7609	0.7627	<b>0.7991</b>
SO	Collaborating	0.8136	0.8181	0.8228	0.8188	0.8377	0.8237	0.8272	0.8162	0.8270	<b>0.8387</b>
	Answering	0.8416	0.8476	0.8493	0.8589	0.8426	0.8442	0.8539	0.8539	0.8597	<b>0.8715</b>
	Commenting	0.7715	0.7769	0.8298	0.8259	0.8138	0.8149	0.8144	0.8136	0.8280	<b>0.8384</b>
	Voting	0.8000	0.7987	0.8152	0.8141	0.8117	0.8117	0.8129	0.8130	0.8152	<b>0.8220</b>

**Figure 4: Network Evolution on Degree Distribution ((a)~(d)) and Clustering Coefficient ((e)~(h))**

MMSB. If we overlook the model differences, ITCom on average achieves 0.042, 0.028, 0.027, 0.026 and 0.021 higher MAP as compared to MMSB, dMMSB, NMDR, dIRM and TF respectively. The better performance of ITCom over baselines with one individual network can be ascribed to the fact that, ITCom considers more knowledge from auxiliary networks and captures more aspects of users' interests. Due to the data sparseness, not all users' interests can be reflected by one individual network. Thus some knowledge will be missed if only one network is considered. On the other hand, although models with simply combined network, such as MMSB-C, exploit knowledge from multiple networks to improve accuracies, they do not consider network differences, i.e., users actually make different friends in different networks. Thus, they may introduce unnecessary regularization from unrelated networks. However, ITCom adaptively captures the cross-network influence and keeps the network-specific growth patterns, hence solves the above problems.

We have other interesting discoveries. First, comparing the performance of other baselines and MMSB, infinite and dynamic modeling can help improve the prediction precision. Second, due to the network differences, combining knowledge may not always be helpful. For example, the performance on the combined networks degrades as the trust and distrust relations are different across networks, thus, simply combined different networks bring negative impacts. Thirdly, the prediction precisions in Github and Stackoverflow are higher than in other networks. The reason is that, users in Github and Stackoverflow are programmers, who may have more consistent interaction patterns. Last but not the least, network influence is not symmetrical. For example, in Xiaonei collection, the improvement in friends application network is higher than that in other networks, as friends ap-

plications are much fewer than other interactions but others are highly dependent on friends application.

### 3.3 Network Evolution

Figure 4 summarizes the evolution results of ME and ITCom in Tencent and Epinion networks on degree distribution and clustering coefficient. Both networks are relational networks which are suitable for the settings of ME, and thus the comparison is fair. Overall, the performance of ME and ITCom are close on the evolution of degree distribution in the last time stamp. They approximate the true distribution of networks well in both collections. Nonetheless, on the Tencent dataset, ITCom performs slightly better than ME. The reason is that ME models network evolution anonymously, that ignores users' personalities. When a network is large, the estimated distribution from a set of anonymous users is stable. However, when data is insufficient, estimation may overfit the rare observations. On the other hand, ITCom models users' activities in person and hence provides more accurate modeling. In addition, ITCom collectively exploits knowledge from two networks, where knowledge in different networks can regularize the evolution process in each individual network and thus alleviate the overfitting issue. The performances of ME and ITCom on clustering coefficient are similar as well. But for long-tail users, e.g., whose number of neighbors is less than 10, the estimation of ITCom is more accurate than ME. That ascribes to the fact that ITCom considers more knowledge from multiple networks.

### 3.4 Performance Analysis

We perform extended experiments on the Tencent dataset to answer three questions: (1). does ITCom capture the network differences and reflect specific growth patterns? (2). how do the model parameters affect the performance of IT-

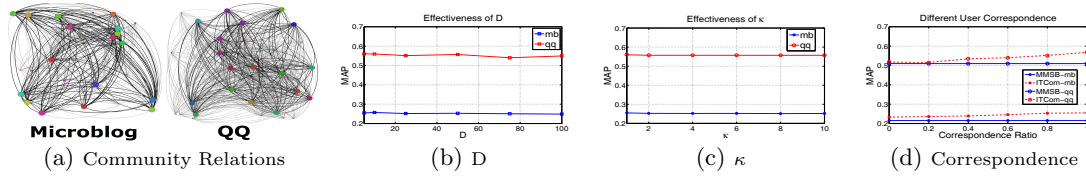


Figure 5: Performance Analysis

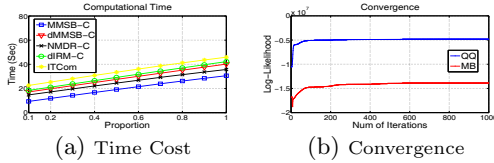


Figure 6: Efficiency Analysis

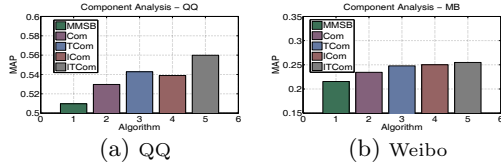


Figure 7: Component Analysis

Com? (3). what is the effectiveness of corresponding ratio among individual networks? (4). what is the effectiveness of each component in ITCom?

We plot the community relation matrices of QQ and Weibo in Figure 5(a), where nodes are communities and the thickness of edges represent the relational strength between communities. Obviously, the number of communities and community relations are different (43 in QQ vs. 33 in Weibo). Communities in Weibo are more centralized while those in QQ are more diverse. This is consistent with real applications, as people on Weibo follow others based on interests while people on QQ may interact with only close friends. This suggests that different networks have different community structures and ITCom can capture this difference.

We analyze the effects to change the number of features  $D$  (5 to 100). The results are illustrated in Figure 5(b). The performance of ITCom is not sensitive to the number of  $D$  but decreases with very large  $D$ . The reason is that  $D$  denotes the representation complexity of each user. Thus, if  $D$  exceeds a threshold (e.g., 50), the model is too complex and results in overfitting. At this point, it is less helpful to improve the model performance by increasing  $D$ . In practice,  $D$  can be tuned through cross-validation techniques. For  $\kappa$ , we vary it from 1 to 10, the results are shown in Figure 5(c). The performance of ITCom drops down slightly with larger  $\kappa$  but it is not sensitive nonetheless.

As many people may only use parts of online services, i.e., they may exist in one individual network but not in all networks, we test the effect of the correspondence ratio between different individual social networks, e.g., the number of users who exist in both networks. Figure 5(d) presents the results with different ratios between the QQ and MB networks. ITCom’s performance becomes worse if fewer correspondences are provided across networks. In addition, it outperforms MMSB consistently if there are corresponding users, implying that ITCom successfully uses the overlapping users as bridges to capture cross-network influences.

To test the effectiveness of each component, i.e., composite, infinite and dynamic in ITCom, we perform experiments on the Tencent dataset by removing specific components in ITCom and produce three baselines. Com considers only composite network knowledge but fixes the number of com-

munities and ignores the temporal information, ICom is the same as ITCom but ignores the temporal information and TCom fixes the number of communities. As shown in Figure 7, the composite modeling contributes most improvements while the infinite and dynamic extensions improve the model performance further.

### 3.5 Efficiency Analysis

As analyzed in the end of Section 2.6, the computational time of ITCom increases linearly with the number of interactions between users. In addition, the computational time is closely related to the convergence property of the inference algorithm. We evaluate these empirically as shown in Figure 6. Figure 6(a) illustrates the computational time of MMSB-C, dMMSB-C, NMDR-C, dIRM-C and ITCom on the Tencent dataset, with different ratio of links. We observe that the computational time increases linearly with larger data size. For our experimental setting, every round of inference takes about 45 seconds in our computer, of which memory is 16G and CPU is 3.2Gz. ITCom not only has better prediction performance as shown in Table 3, but also has similar time cost with dIRM-C and dMMSB-C. Figure 6(b) shows the convergence property of ITCom. Clearly, ITCom becomes convergent after about 400 iterations. Combining with the time cost of each round, the inference algorithm of ITCom takes about five hours to build a model.

## 4. RELATED WORKS

**Social Network Analysis** [5] has drawn much research interest, ranging from link prediction [12], community detection [11], to social influence [17]. Several works have been proposed to handle social networks with multiple relations, such as tensor factorization [7]. However, these works treat each type of relationship as equally important and do not consider network differences. In previous research [19], we introduced the concept of composite social networks. Based on the property that different social networks reflect different aspects of users’ interests, we introduced a hierarchical Bayesian model to predict users’ behaviors by exploiting knowledge in different individual networks adaptively.

**Network Dynamics** As social networks change over time, studying network dynamics and evolution is also an important topic. Network dynamics focuses on the microscopic changes of networks, such as the links generated by specific users, etc. For example, Nguyen et al. proposed a two-step framework to trace the evolution of overlapped network communities in dynamic mobile networks [14]. On the other hand, network evolution aims to describe the change of macroscopic network properties, such as diameter, clustering coefficients, etc. For example, Leskovec et al. proposed a simulation method to construct networks step-by-step depending on parameters estimated by maximal-likelihood methods [10]. However, these existing works studied only one network but did not consider the influences across networks. Besides, previous research works mostly deal with one



aspect, dynamics or evolution, but the proposed model can handle both simultaneously.

**Mixed Membership Models** Recently, mixed membership models have been demonstrated to be effective to model relational data, such as MMSB [1]. The main idea is to represent each entity as a mixed membership vector over communities or topics. Recently, MMSB has been extended from two aspects: dynamic [6, 8, 4] and nonparametric [18], in order to model dynamical data and release the constraints on the number of communities. In addition, auxiliary data are also considered to improve the model performance [9]. Cross-domain collaborative filtering is introduced [13], where hierarchical Bayesian models are proposed to solve multiple domain user personalization. However, it is applied on static user-item interaction networks.

## 5. CONCLUSION

In this paper, we studied a new problem on dynamics analysis of composite social networks. We defined a composite social network as a set of nested individual networks, where users and links in different individual networks overlap. The dynamical process of each individual network can influence each other while keeping specific evolution patterns. To model this co-evolution process, we proposed a nonparametric Bayesian model, ITCOM, by capturing cross-network influences adaptively. The main idea is to model users' latent interests, which carry the common knowledge embedded in multiple networks and use network-specific factors to describe the network dependent growth patterns on community structures. These two kinds of knowledge are then encoded as a hybrid prior for a dynamic mixed membership model to generate the links between users in each time stamp. To allow communities vary over time and networks, we construct nonparametric models. Unlike prior works, the proposed model can capture the microscopic and macroscopic aspects of network dynamics, including the links of each specific user and the global changes of networks properties. The proposed model is flexible in that, it can be extended to any number of networks and the number of communities can be determined automatically. We conducted large-scale empirical studies on eight real-world network collections, where ITCOM outperforms several state-of-the-art baselines on link prediction by as high as 0.11 on MAP efficiently and estimates the evolution of degree distribution and clustering coefficient accurately.

**Future Works** In this paper, the proposed model require the users across networks to be identical. We propose to relax this restriction in the future. Besides modeling the network dynamics and predicting users' behaviors [19], the rich data embedded in composite social networks can benefit many other applications. Firstly, users' characteristics and their interactions can be exploited to build an accurate user distance measure. In addition, by considering multiple networks together, common patterns can be extracted from users' behaviors/interactions and these patterns can be used to infer their characteristics.

## Acknowledgement

We thank the support of Hong Kong CERG Projects 621211 and 620812.

## References

- [1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.
- [2] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2):7:1–7:30, 2010.
- [3] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
- [4] Y.-S. Cho, G. V. Steeg, and A. Galstyan. Co-evolution of selection and influence in social networks. In W. Burgard and D. Roth, editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA*. AAAI Press, 2011.
- [5] N. Du, H. Wang, and C. Faloutsos. Analysis of large multi-modal social networks: patterns and a generator. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I, ECML PKDD'10*, pages 393–408, Berlin, Heidelberg, 2010. Springer-Verlag.
- [6] W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 329–336, New York, NY, USA, 2009. ACM.
- [7] S. Gao, L. Denoyer, and P. Gallinari. Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks. *CoRR*, abs/1204.2588, 2012.
- [8] K. Ishiguro, T. Iwata, N. Ueda, and J. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 919–927. MIT Press, 2010.
- [9] D. I. Kim, M. Hughes, and E. Sudderth. The nonparametric metadata dependent relational model. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 1559–1566, New York, NY, USA, 2012. Omnipress.
- [10] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 462–470, New York, NY, USA, 2008. ACM.
- [11] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 631–640, New York, NY, USA, 2010. ACM.
- [12] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th International Conference on Information and Knowledge Management, CIKM '03*, pages 556–559, New York, NY, USA, 2003. ACM.
- [13] Y. Low, D. Agarwal, and A. J. Smola. Multiple domain user personalization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge discovery and data mining, KDD '11*, pages 123–131, New York, NY, USA, 2011. ACM.
- [14] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In *Proceedings of the 17th annual international conference on Mobile computing and networking, MobiCom '11*, pages 85–96, New York, NY, USA, 2011. ACM.
- [15] O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95:169–186, 2008.
- [16] T. TyLenda, R. Angelova, and S. Bedathur. Towards time-aware link prediction in evolving social networks. In *Proceedings of the 3rd Workshop on Social Network Mining and Analysis, SNA-KDD '09*, pages 9:1–9:10, New York, NY, USA, 2009. ACM.
- [17] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 981–990, New York, NY, USA, 2010. ACM.
- [18] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite hidden relational models. In *Proceedings of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 544–551, Arlington, Virginia, 2006. AUAI Press.
- [19] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li. Comsoc: adaptive transfer of user behaviors over composite social network. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12*, pages 696–704, New York, NY, USA, 2012. ACM.