

ComSoc: Adaptive Transfer of User Behaviors over Composite Social Network*

Erheng Zhong[†], Wei Fan[‡], Junwei Wang[§], Lei Xiao[§], and Yong Li[§]

[†]Hong Kong University of Science and Technology, Hong Kong

[‡]IBM T.J.Watson Research Center, New York, USA

[§]Tencent Inc., Shenzhen, China

ezhong@ust.hk, weifan@us.ibm.com, {jannywang, shawnxiao, nickyyli}@tencent.com

ABSTRACT

Accurate prediction of user behaviors is important for many social media applications, including social marketing, personalization and recommendation, etc. A major challenge lies in that, the available behavior data or interactions between users and items in a given social network are usually very limited and sparse (e.g., $\geq 99.9\%$ empty). Many previous works model user behavior from only historical user logs. We observe that many people are members of several social networks in the same time, such as Facebook, Twitter and Tencent's QQ¹. Importantly, their behaviors and interests in different networks influence one another. This gives us an opportunity to leverage the knowledge of user behaviors in different networks, in order to alleviate the data sparsity problem, and enhance the predictive performance of user modeling. Combining different networks "simply and naively" does not work well. Instead, we formulate the problem to model multiple networks as "composite network knowledge transfer". We first select the most suitable networks inside a composite social network via a hierarchical Bayesian model, parameterized for individual users, and then build topic models for user behavior prediction using both the relationships in the selected networks and related behavior data. To handle big data, we have implemented the algorithm using Map/Reduce. We demonstrate that the proposed composite network-based user behavior model significantly improve the predictive accuracy over a number of existing approaches on several real world applications, such as a very large social-networking dataset from Tencent Inc.

Categories and Subject Descriptors

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

*Part of the work is done during the first author's internship at Tencent Inc.

¹<http://www.tencent.com/en-us/ps/imservice.shtml>

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'12, August 12–16, 2012, Beijing, China.

Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...\$15.00.

Keywords

Social Network Analysis, Transfer Learning

1. INTRODUCTION

An important challenge for user behavior modeling in social media is how to push the right information to the right users at the right time. Accurate user-behavior models enable us to predict users' actions and intents based on data collected from online product purchasing, webpage browsing and ads clicking history, etc. Different approaches of user-behavior modeling, such as collaborative filtering, have been proposed [19]. A major challenge to use these methods lies in the "sparse" nature of historical data, whereby users have interactions with only few other users and items; users may click on only one or two advertisements and listen to a few songs during a time period. For example, the dataset from Tencent has more than 99.9% empty entries in the user-item interaction matrix. The problem with sparse data makes existing methods overfit, and thus fails to model users' behaviors accurately.

With the development of Web 2.0, billions of users are now engaged in multiple online social networks, such as Facebook, Twitter, Tencent's QQ, etc. In many cases, a given user is simultaneously a member of several different networks. These networks form a **composite** social network, where users have various relationships, and can both exhibit different behaviors in each individual network or subnetwork, and share some common latent interests across networks in the same time. Many people use Facebook to communicate with their friends, but post their real-time information on Twitter and browse video uploaded by others on Youtube. Inside a composite social network, links in different sub-networks reflect users' behaviors and relationships from different aspects. For example, friendship relationships on Facebook reflect users' acquaintances in daily life, but contacts on Youtube capture users' common preferences on videos instead. Users tend to build social relations based on interests, and their behaviors are closely related to interests. Thus, one's behaviors can be inferred from that of his/her neighbors. For example, one may want to watch a movie as he/she finds that most friends have watched already. Hence, the knowledge in users' social relationships shed light on solving the sparsity problem.

While interactions in a single social network can be sparse, there is an opportunity to exploit data in multiple networks by considering overlapping users as bridges. Several previous research works have considered combining multiple networks to form a larger network, in which a holistic user

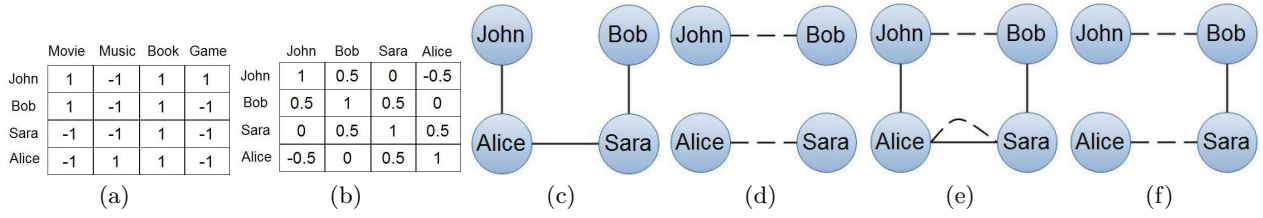


Figure 1: Synthetic example on how to exploit Composite Social Network adaptively. (a) the complete user-item interaction matrix - most items are empty and need to be modeled by recommendation system; (b) the user similarity based on user-item interactions; (c) “Facebook” relationship among them; (d) “Following” relationship on Twitter; (e) the naive combination of (c) and (d); (f) the adaptive combination reflecting the correct user similarity in (b).

model could ideally be built. For example, Pan et al. [16] has proposed to combine multiple user relationships to recommend Apps to users, where the subnetworks are treated uniformly, such that each user-user relationship holds the same importance. However, we observe that in a composite network, the solution by treating each relationship uniformly may likely result in poor prediction performance, due to the following two reasons. First, different networks have their own properties, such as density, degree distributions, clustering coefficient or diameters. If we combine dense and sparse networks together, the information in the sparse network will be overwhelmed inside the dense one. Second, users play different roles and get different levels of influence from their neighbors in individual networks. Some users may prefer Facebook over Twitter for one application, while others may use Twitter more frequently instead. Hence, neighbors in different networks may have different impacts on a given user. To illustrate the idea, a synthetic example is drawn in Figure 1, where there are four users (John, Sara, Bob, Alice) and their preferences on four hobbies: watching movie, listening to music, reading book and playing games. In this figure, “1” represents “like” and “-1” denotes “dislike”. Based on the interaction matrix in Figure 1(a), we compute the users’ similarities with cosine distance [19], as shown in Figure 1(b). Clearly, John is similar to Bob, Bob is similar to John and Sara, and Sara is similar to both Bob and Alice. Figure 1(c) summarizes their relations under a given relationship, such as the friendship on Facebook while Figure 1(d) shows another one, such as the following relationship on Twitter. Clearly, both of the friendship and following relationships can only partially reflect the similarities in interests. If we simply merge them together, as shown in Figure 1(e), the naively combined network does not reflect the correct similarity among them. For example, John and Alice are dissimilar, but there is still a link between them.

In this paper, we present a hierarchical Bayesian model to cope with the above challenges. It transfers knowledge of behavior models across subnetworks in a composite network, specifically adapted and parameterized for different users. The common users across subnetworks form bridges to enable the transfer. To uncover the common knowledge across different subnetworks, we apply transfer learning techniques [15] to exploit both the topological and topical knowledge in social networks. The basic idea is to integrate relationships adaptively, where different users will be influenced by their neighbors in each subnetwork with personalized weights.

We explain the approach briefly as follows. The model is based on latent Dirichlet allocation (LDA) [1]. Essentially, we model users’ interests as latent topics. We maintain two

Table 1: Definition of notations

Group	Notation	Notation Description
Data	$\mathcal{G} = \{G_i\}_{i=1}^{\ell}$	Composite Social Networks
	$\mathbf{U} = \{u_i\}_{i=1}^N$	User set of \mathcal{G}
	$\mathbf{E} = \cup\{E_i\}_{i=1}^{\ell}$	Link set of \mathcal{G}
	$\mathbf{V} = \{v_i\}_{i=1}^T$	Item set
	$\mathcal{A} = \{a_i\}_{i=1}^Q$	Interaction set
	ℓ	Number of subnetworks of \mathcal{G}
	N	Number of users in \mathbf{U}
	T	Number of items in \mathbf{V}
	Q	Number of interactions in \mathcal{A}
	M_i	Number of links in E_i
Model	$\text{Dir}(\cdot)$	Dirichlet distribution
	$\text{Mult}(\cdot)$	Multi-nominal distribution
	$\text{Bern}(\cdot)$	Bernoulli distribution
	θ, ϕ and ψ	Topic proportions
	α, β and ν	Priors
	K	Number of topics

distributions, i.e., users’ distributions over topics and topics’ distributions over items. These distributions are denoted as $\text{Pr}(t|u)$ and $\text{Pr}(v|t)$, where t, v and u denote topic, item and user respectively. To exploit the relational knowledge among users, we add a constraint on $\text{Pr}(t|u)$ to enforce that one user’s distribution over topics should be similar to his/her neighbors’. In addition, we introduce one more measure $\text{Pr}(G_s|u)$ to indicate the importance of a chosen social network G_s for a given user u . This probability is important, since each user is influenced by his/her neighbors in different social networks with different levels of importance. We revisit the synthetic example in Figure 1 to illustrate the idea. By introducing $\text{Pr}(G_s|u)$, each user can adopt relationship from different subnetworks individually according to their similarities to others. For example, Bob adopts the Facebook relationship in Figure 1(c) while Alice adopts the “Following” relationship in Figure 1(d). Hence the adaptively combined network can reflect users’ interests correctly, as shown in Figure 1(f). We derive an efficient Gibbs sampling method with a Map/Reduce implementation to compute parameters in the proposed model, and infer the user-topic distribution $\text{Pr}(t|u)$, the item-topic distribution $\text{Pr}(v|t)$ and the user-network distribution $\text{Pr}(G_s|u)$. Extensive experiments were performed to verify the effectiveness and the efficiency of the proposed model, by comparing with state-of-the-art approaches on several large social-networking datasets - such as the actual datasets from Tencent Inc.

2. PROBLEM FORMULATION

First, we formally define the concept of composite social network. The notations are summarized in Table 1. Let $\mathcal{G} = \{G_i = (U_i, E_i)\}_{i=1}^{\ell}$ denote a composite social network, where G_i is the i -th subnetwork, U_i is the user set of G_i , E_i is the user relationship of U_i and ℓ is the number of subnetworks. In addition, we define a complete user set as

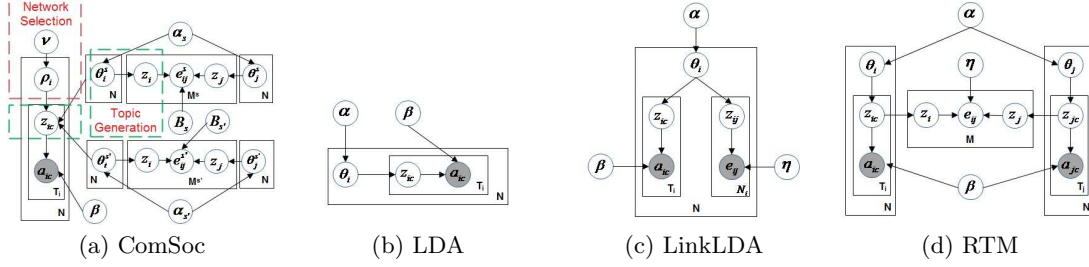


Figure 2: Different Topic Models

\mathbf{U} and a complete link set as \mathbf{E} , where $\mathbf{U} = \cup\{U_i\}_{i=1}^\ell = \{u_j\}_{j=1}^N$ and $\mathbf{E} = \cup\{E_i = \{e_{ij}^j\}_{j=1}^{M_i}\}_{i=1}^\ell$. In a composite network, users and links in different subnetworks overlap:

$$U_i \cap U_j \neq \emptyset, \forall i \neq j, U_i \subseteq \mathbf{U}, U_j \subseteq \mathbf{U} \quad (1)$$

$$E_i \cap E_j \neq \emptyset, \forall i \neq j, E_i \subseteq \mathbf{E}, E_j \subseteq \mathbf{E} \quad (2)$$

These properties are easy to satisfy in practice. For example, users of Facebook and Twitter often overlap, although are not identical. Two users may be friends on Facebook, but may or may not always be followers on Twitter. We notice that in different social networks, a user can have a different set of neighbors, and also have different levels of influence on their neighbors. This depends on how often they interact and the closeness of their relationships. It is a key property to exploit for knowledge transfer in a composite social network, as different subnetworks may capture partial aspects of a given user's behaviors. Consequently, we define the task of user behavior prediction as follows. Let $\mathbf{V} = \{v_j\}_{j=1}^T$ be an item set, where v_j is the j -th item, which can be a movie, music, news, advertisement, webpage, etc. Let the matrix $\mathcal{A} = \{a_i = (u_j, v_k)\}_{i=1}^Q$ denote the historical user behavior data, where each user-item pair (u_j, v_k) represents the interaction between u_j and v_k , such as clicking, rating, etc., and most the entries in this matrix are empty. Formally, the task of user behavior prediction is to predict the non-existing elements in \mathcal{A} with the help of \mathbb{G} .

3. COMPOSITE SOCIAL TOPIC MODEL

We propose a composite social topic model (ComSoc) for selecting subnetworks in a composite social network, in order to adaptively transfer the relationship knowledge. A key point is that, the regularization from different networks is adjusted adaptively depending on the user. This is consistent with the observation that, someone may use Facebook more often than Twitter and hence the neighbors on the Facebook shall have higher level of influence for the given user, while others may be influenced more by their followers on Twitter, as they prefer Twitter over Facebook.

The complete generating process of ComSoc is as follows. Firstly, ComSoc models users' interests as topics and then each user is represented as a distribution over topics, $\Pr(t|u)$. In addition, ComSoc adds one constraint on $\Pr(t|u)$ by enforcing that users' distributions over topics can be used to generate links between users. Intuitively, common interests tend to create relationships among people. For each user in each subnetwork G_d , ComSoc assumes $\Pr(t|u)$, a Multinomial distribution, is generated by a Dirichlet distribution with a prior α_d . Subsequently, a topic indicator is drawn from this distribution for each user. Then every link in each network is generated by a Bernoulli distribution, of which the prior is determined by the topic indicators of the users

and a topic compatibility matrix B^d related to G_d , where B^d represents the topic connections. After generating the user-topic distribution $\Pr(t|u)$ for each user in every social networks, to transfer knowledge from social relationships adaptively, we introduce a term of users' distribution over networks, $\Pr(G_d|u)$, in order to represent the probability of how much a user is influenced by a given network. ComSoc models this user-network distribution in a hierarchical manner. At first, for each user u_i , a network proportion ρ_i is drawn from a Dirichlet distribution with a prior ν . Then, for every interaction of a given user, a social network is drawn from a Multinomial distribution according to the network proportion. The user-topic distribution of the selected network, $\Pr(t|u)$, is picked to generate a topic assignment. Finally, ComSoc generates each interaction according to the item-topic distribution, $\Pr(v|t)$, over the chosen topic. We formulate this process as follows.

- For every user u_i in each network G_d :
 - Draw topic proportion $\theta_{id} \sim \mathbf{Dir}(\alpha_d)$
- For every link of (u_i, u_j) in each network G_d :
 - Draw topic indicator $z_i \sim \mathbf{Mult}(\theta_{id})$, $z_j \sim \mathbf{Mult}(\theta_{jd})$
 - Draw binary link indicator $e_{ij}^d \sim \mathbf{Bern}(B_{z_i, z_j}^d)$
- For each user u_i :
 - Draw network proportion $\rho_i \sim \mathbf{Dir}(\nu)$
 - For each interaction $a_{i,k} \in \mathcal{A}$ with item v_k :
 - * Draw a network indicator $d \sim \mathbf{Mult}(\rho_i)$
 - * Draw assignment $z_{i,k} \sim \mathbf{Mult}(\theta_{id})$
 - * Draw interaction $a_{i,k} \sim \mathbf{Mult}(\beta_{z_{i,k}})$

We observe that, each user u_i has her/his own network proportion ρ_i . This means that we choose different subnetworks to transfer knowledge adaptively for each user.

We draw the graphical model of ComSoc, as well as the basic LDA, LinkLDA [7] and RTM [3] in Figure 2. To make the figures clearer, for ComSoc, we draw only two social networks. In contrast to both LinkLDA and RTM, ComSoc follows a novel heuristic to exploit the knowledge in social networks in two levels or hierarchies: (1) it first introduces a network indicator to judge which network is appropriate to transfer behavior data and help the recommendation for each user adaptively - the portion surrounded by the red box in Figure 2(a); (2) for each user, it generates topics from shared user-topic distribution, in order to produce user-item interactions and user-user relations in each subnetwork at the same time - the portions highlighted by the green box. We notice that, this adaptation is not employed in the global merging approach proposed in [16]. To learn the model, we propose an efficient inference algorithm based on Gibbs sampling. In addition, to cope with the massive data issue, we also propose a large-scale implementation based on Map/Reduce [5].

3.1 Inference Algorithm

The basic idea is to update topic assignments from the posterior distribution. To guarantee sampling efficiency, the posterior distribution is represented in collapsed format [18]. We first define a number of statistics. Let $n_{t,v}$ denote the number of the topic t related to an item, and $n_{u,t}^s$ denote the number of the topic t assigned to a user in the s -th network. Formally, for each topic assignment $z_{i,k}$,

$$n_{t,v} = \sum_{i,k} \{z_{i,k} = t; v_k = v\}; n_{u,t}^s = \sum_k \{z_{i,k}^s = t\}$$

In addition, we denote $n_t = \sum_v n_{t,v}$, where n_t stores the number of times that items are assigned to the topic t , and n_u , which maintains the number of items interacted with the user u . Lastly, we define another statistic $n_{u,d}$, which is the number of times that a user u utilizes subnetwork G_d in order to regularize her/his distribution over topics.

We express the full joint probability of the interaction data in \mathcal{A} with network indicators as:

$$\begin{aligned} \Pr(\mathcal{A}, z, s | \alpha, \beta, \nu) &= \left[\prod_{i=1}^N \frac{\prod_{j=1}^K \Gamma(\alpha_s + n_{u=i,t=j})}{\alpha_s K + n_{u=i}} \frac{\Gamma(\alpha_s K)}{\prod_{j=1}^K \Gamma(\alpha_s)} \right] \\ &\times \left[\prod_{i=1}^K \frac{\prod_{j=1}^T \Gamma(\beta + n_{t=i,v=j})}{\beta T + n_{t=i}} \frac{\Gamma(\beta T)}{\prod_{j=1}^T \Gamma(\beta)} \right] \\ &\times \left[\prod_{i=1}^N \frac{\prod_{j=1}^\ell \Gamma(\nu + n_{u=i,d=j})}{\nu \ell + n_{u=i}} \frac{\Gamma(\nu \ell)}{\prod_{j=1}^\ell \Gamma(\nu)} \right] \end{aligned}$$

We notice that, different from the inference in LDA [18], the above equation has one additional term to model, which network to be selected, in order to regularize users' distributions over topics. Following the inference in [18], let $\mathbf{z}_{-(i,k)}$ denote the topic assignments without the current topic on $a_{i,k}$, and then we rewrite the above equation to obtain the following un-normalized probability, in order to resample the topic $z_{i,k}$ for $a_{i,k}$ between user u_i and item v_k :

$$\begin{aligned} \Pr(z_{i,k} = t, d = s | \mathcal{A}, \mathbf{z}_{-(i,k)}) & \quad (3) \\ \propto (n_{i,t,-(i,k)}^s + \alpha_s) (n_{i,s,-(i,k)} + \nu) \frac{n_{t,k,-(i,k)} + \beta}{n_{t,-(i,k)} + T\beta} \end{aligned}$$

The resulting equation is used to update the topic assignments for user-item interactions. In addition, we need to sample users' pairwise topic assignments for user-user relationships in each subnetwork G_s , which regularizes the user-topic distributions for user-item interactions. In each network, we first define the full joint probability as:

$$\begin{aligned} \Pr(\tilde{z}, E_s | B^s, \alpha_s) &= \left[\prod_{i=1}^N \frac{\prod_{j=1}^K \Gamma(\alpha_s + n_{u=i,t=j})}{\alpha_s K + n_{u=i}} \frac{\Gamma(\alpha_s K)}{\prod_{j=1}^K \Gamma(\alpha_s)} \right] \\ &\times \left[\prod_{i=1}^N \frac{\prod_{j=1}^K \Gamma(\alpha_s + n_{u=i,t=j})}{\alpha_s K + n_{u=i}} \frac{\Gamma(\alpha_s K)}{\prod_{j=1}^K \Gamma(\alpha_s)} \right] \end{aligned}$$

Following the same inference process, the un-normalized probability for resampling the pairwise topic for user relationship between u_i and u_j is

$$\begin{aligned} \Pr(\tilde{z}_{i,j} = t_1, \tilde{z}_{j,i} = t_2 | \alpha_s, B^s, E_s, \tilde{\mathbf{z}}_{-(i,j)}) & \quad (4) \\ \propto \Pi_{t_1,t_2} \Gamma(n_{i,t_1}^s + \alpha_s) \Gamma(n_{j,t_2}^s + \alpha_s) \\ \propto (n_{i,t_1,-(i,j)}^s + \alpha_s) (n_{j,t_2,-(i,j)}^s + \alpha_s) \end{aligned}$$

This equation is used to update the pairwise topic assignments for user-user relations. In summary, in each iteration of Gibbs sampling, we update the topic assignments for

Algorithm 1 Inference of ComSoc

```

1: Input: Composite network:  $\mathcal{G}$ , Interactions:  $\mathcal{A}$ , # of iterations  $I$ , # of burn-in,  $I_b$ , Priors:  $\alpha$ ,  $\beta$  and  $\nu$ 
2: Output: User-topic distribution  $\theta_s$  for each subnetwork  $G_s$ , topic-item distribution  $\phi$  and user-network distribution  $\psi$ 
3: Assign topics for user-item pairs randomly
4: Assign topics for user-user pairs in each subnetwork
5: Initialize the user-network assignments randomly
6: Initialize the  $\theta_d$ ,  $\phi$  and  $\psi$  with zeros
7: for  $i = 1$  to  $I$  do
8:   Update user-item topic assignments using Eq.(3)
9:   Update user-user topic assignments using Eq.(4)
10:  IF  $I > I_b$  Update  $\theta_s$ ,  $\phi$ ,  $\psi$  as
11:     $\theta_{s,i,j} = \frac{1}{I - I_b} \frac{\alpha_s + n_{u=i,t=j}}{\alpha_s K + n_{u=i}}$ 
12:     $\phi_{i,j} = \frac{1}{I - I_b} \frac{\beta + n_{t=j,v=i}}{\beta V + n_{v=i}}$ 
13:     $\psi_{i,j} = \frac{1}{I - I_b} \frac{\nu + n_{u=i,d=j}}{\nu \ell + n_{u=i}}$ 
14:  end for
15: Return  $\theta_s$ ,  $\phi$  and  $\psi$ 

```

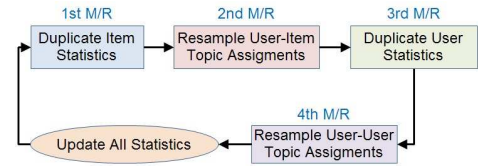


Figure 3: Large Scale Implementation on Map/Reduce

user-item interactions and user-user relations in each subnetwork alternatively using Eq.(3) and Eq.(4) respectively until convergence. We revisit the synthetic example in Figure 1. After learning the model, to predict the preference of each user, it first selects an appropriate social relationship. For example, Bob adopts the Facebook relationship in Figure 1(c) while Alice adopts the “Following” relationship in Figure 1(d). Then, an adaptively combined network is constructed, as shown in Figure 1(f). The neighbors’ knowledge in this network is exploited to help pick a topic. Finally an item is selected to recommend to the given user according to the item proportion of this topic.

Inference Framework The inference framework of ComSoc can be found in Algorithm 1. At the beginning, it randomly picks topics for user-item interactions and user-user relationships. Afterwards, in each iteration, Eq.(3) and Eq.(4) are utilized to update the topic assignments from user-item interactions and user-user relations in each social network. The iteration is repeated until convergence. In addition, a burn-in process is introduced in the first several iterations to remove unreliable sampling results. The parameters θ_s , ϕ and ψ are used to predict user behaviors.

Time Complexity We analyze the time complexity as follows. Suppose the process needs I iterations to reach convergence. In each iteration, it requires to go through all user-item pairs (Q) in the interaction network and user-user pairs (M_i) in each sub social network. In addition, for each pair, it requires $O(K)$ operations to compute the posterior distribution and sample topics, and constant cost to update the statistics. At last, it needs $O(N + T)$ operations to generate topic proportions. Thus, the whole time complexity is $O(I(Q + \sum_{i=1}^\ell M_i)K + N + T) = O(I(Q + \sum_{i=1}^\ell M_i)K)$.

3.2 Large-scale Implementation

In many real world applications, the number of users (N) can be in the order of several billions, the number of items (T) can be up to millions and the number of user-item inter-

Table 2: Summary of Data Characteristics

Collections	Tencent		Douban		Collections	Tencent		Douban		
Networks	QQ	Microblog	Online	Offline	Behaviors	Music	Profile	Movie	Music	Book
#User	$\sim 1M$		$\sim 50K$		#Item	$\sim 360K$	$\sim 5K$	$\sim 57K$	$\sim 180K$	$\sim 170K$
#Edge	$\sim 80M$	$\sim 32M$	$\sim 5M$	$\sim 4M$	#Interaction	$\sim 55M$	$\sim 13M$	$\sim 5.2M$	$\sim 2.5M$	$\sim 3.2M$

actions (Q) and user-user relations (M_i) can be tens of billions. In a way, these are very big datasets. Thus, the ability to scale up the proposed algorithm is crucial for real-world practice. We implement the inference algorithms of ComSoc based on Map/Reduce [5]. The main idea is to compute the posterior distribution and resample the topic assignments on partial data in each single machine and then aggregate pieces of results. Recall that in Eq.(3) and Eq.(4), several statistics will be updated in the sampling process, including the user-topic count $n_{u,\cdot}^s$ of each user in the s -th network, the user-network count $n_{u,\cdot}$, the topic count n_t , the item-topic count $n_{\cdot,v}$ of each item and the topic assignment $z_{i,k}$ for u_i and v_k . We partition the data by users. For example, if there are D machines, then the whole dataset will be divided into D groups and each group will contain parts of users as well as their relationship and interaction data. As stated in [20], $n_{u,\cdot}^s$, $n_{u,\cdot}$ and $z_{i,k}$ are entirely local to each user and thus do not need to be shared. They can be written to the disk after resampling. In addition, n_t and $n_{\cdot,v}$ change slowly, and hence, a delay in obtaining their up-to-date representations will not affect the sampler significantly. Furthermore, the number of topic statistics, $\{n_t\}_{t=1}^K$, is small, where K is the number of topics. Thus we can keep them in the memory permanently. These observations accomplish the local sampling without losing too much precision. In summary, the distributed implementation of Algorithm 1 is also an iterative approach, but in each iteration, it resamples topic assignments in a distributed manner. Let M/R denote a pair of Map/Reduce operations for simplicity.

Before the start of iterations, the inference algorithm performs distributed initialization. Due to limited space, we omit the details here. The sampling process in each iteration needs four M/R. The main flow can be found in Figure 3. The first M/R is used to duplicate the statistics $n_{\cdot,v}$ of every item for each related user. Similarly, the third M/R is used to duplicate the statistics $n_{u,\cdot}^s$, since each user is linked to multiple users in the s -th social network. For each user u_i , the related statistics $n_{u=i,\cdot}^s$ will be copied $|N^s(u_i)|$ times, where $|N^s(u_i)|$ is the number of its neighbors in the s -th network, and each duplicated statistics is used for the sampling of each neighbor $u_j \in N^s(u_i)$. The second M/R is used to resample the topic assignments for user-item interactions and the last one resamples the topic assignments for user-user relations in each subnetwork. All information related to users and items are represented as $\langle key, value \rangle$ pairs, where *key* can be either user or item identities and *value* can be statistics, topic assignments $z_{i,\cdot}$, item interactions $a_{\cdot,k}$ and user relations information $N^s(u_i)$. Let ξ denote the type of *value* and c indicate whether the value is for a given user or her/his neighbors. Formally, the process is as follows:

- Map-I: Map $\langle v_k, n_{\cdot,v=k} \rangle$ and $\langle v_k, a_{\cdot,k} \rangle$, in the form of $\langle v_k, \{\xi, V\} \rangle$ where V is either $n_{\cdot,v=k}$ or $a_{\cdot,k}$.
- Reduce-I: Take $\langle v_k, \{\xi, V\} \rangle$ and emit $\langle u_i, \{v_k, n_{\cdot,v=k}\} \rangle$ for each user $u_i \in a_{\cdot,k}$.
- Map-II: Map $\langle u_i, \{v_k, n_{\cdot,v=k}\} \rangle$, $\langle u_i, n_{u=i,\cdot}^s \rangle$, the user-network count $\langle u_i, n_{u=i,\cdot} \rangle$ and $\langle u_i, z_{i,\cdot} \rangle$ in the form of $\langle u_i, \{\xi, V\} \rangle$.
- Reduce-II: Take $\langle u_i, \{\xi, V\} \rangle$, compute the topic as-

signments according to Eq.(3) and emit the updated $\langle u_i, n_{u=i,\cdot}^s \rangle$, $\langle u_i, n_{u=i,\cdot} \rangle$, $\langle u_i, z_{i,\cdot} \rangle$ and the increments of $\langle v_k, n_{\cdot,v=k} \rangle$ and $\langle t, n_t \rangle$.

- Map-III: Map $\langle u_i, n_{u=i,\cdot}^s \rangle$ and $\langle u_i, N^s(u_i) \rangle$, such that statistics with the same user are shuffled to the same machine in the form of $\langle u_i, \{\xi, V\} \rangle$ where V is either $n_{u=i,\cdot}^s$ or $N^s(u_i)$.
- Reduce-III: Take $\langle u_i, \{\xi, V\} \rangle$ and emit $\langle u_j, \{c, n_{u=i,\cdot}^s\} \rangle$ for each user $u_j \in N^s(u_i) \cup u_i$.
- Map-IV: Map $\langle u_i, \{c, n_{u=j,\cdot}^s\} \rangle$ and $\langle u_i, z_{i,\cdot} \rangle$ in the form of $\langle u_i, \{\xi, V\} \rangle$.
- Reduce-IV: Take $\langle u_i, \{\xi, V\} \rangle$, compute the topic assignments according to Eq.(4) and emit the updated $\langle u_i, n_{u=i,\cdot}^s \rangle$ and $\langle u_i, z_{i,\cdot} \rangle$.

Subsequently, the increments for the counts on topics and item-topic pairs will be used to update $n_{\cdot,v=k}$. Finally, n_t in each single machine and the updated statistics will be shuffled to each single machine for the next iteration.

4. EXPERIMENTS

We evaluate the proposed methods on two data collections, one is from Douban², and the other is from Tencent; both contain composite social networks and rich user behavior information. We compare the proposed approach with three state-of-the-art methods: latent Dirichlet allocation (LDA) [1], LinkLDA [7] and relational topic models (RTM) [3]. For LinkLDA and RTM, we implement two variations: one is to build models with only one single social network, and the other is to build models with a naive combined social network, of which the edge set is the union of all edges in subnetworks. For the model parameters, we set the priors for topic models as $\alpha = 0.5$, $\beta = 50/K$ and $\nu = 0.5$. In addition, we set the number of topics as $K = 50$. We will study the effectiveness of this parameter later. The results are evaluated based on two criteria: perplexity (perp) and mean average precision (MAP) on the hold-out set \mathcal{T} .

$$\text{perplexity} = \exp \left\{ - \frac{\sum_{(u,v) \in \mathcal{T}} \log \Pr(u,v)}{|\mathcal{T}|} \right\} \quad (5)$$

$$\text{MAP} = \frac{1}{N} \sum_{u \in \mathbf{U}} \frac{\sum_{(u,v) \in \mathcal{T}_u} \frac{1}{r} \sum_{r \in [1,r]} pre_r}{|\mathcal{T}_u|} \quad (6)$$

where pre_r represents the precision on top- r predictions. Behavior records in \mathcal{T} are represented as a list of user-item pairs. For example, for the music listening prediction, one user-item pair represents one user will listen to the given music. Then the task is to predict if such behaviors happen or not, in other words, if the corresponding user-item pairs exist in \mathcal{T} or not. Generally, perplexity measures how precisely the algorithm generates the behaviors in the hold-out set, and MAP measures how well the algorithms rank the behaviors in the hold-out set above non-existing behaviors. In addition, to evaluate the significance in improvement of the proposed model, we randomly split the hold-out set into 10 subsets and then calculate the error bar. We notice that, the observed behaviors are implicit, where we only know which items are interacted by a given user.

²<http://www.douban.com>

Table 3: Comparison in MAP and perplexity on Douban Collection (K=50)

Algorithm		LDA	LinkLDA-On	LinkLDA-Off	LinkLDA-C	RTM-On	RTM-Off	RTM-C	ComSoc
Movie	perp	7140.3	7311.1	7342.6	7269.9	7425.1	7433.6	7370.6	6626.6
	MAP	0.8407	0.8407	0.8406	0.8415	0.8434	0.8407	0.8517	0.8581
Music	perp	43676.8	43506.4	45185.9	43730.3	43511.8	44358.1	43667.4	34540.5
	MAP	0.7492	0.7492	0.7489	0.7571	0.7583	0.7493	0.7581	0.7624
Book	perp	47541.0	54646.1	58460.9	54627.8	55388.0	57130.3	55386.9	33624.9
	MAP	0.7084	0.7088	0.7086	0.7085	0.7091	0.7088	0.7089	0.7173

Table 4: Comparison in MAP and perplexity on Tencent Collection (K=50)

Algorithm		LDA	LinkLDA-QQ	LinkLDA-MB	LinkLDA-C	RTM-QQ	RTM-MB	RTM-C	ComSoc
Music	perp	21118.8	21041.2	21032.1	21020.3	21016.8	21042.3	21080.2	20042.3
	MAP	0.8497	0.8580	0.8582	0.8677	0.8559	0.8591	0.8681	0.8795
Profile	perp	1233.1	1217.9	1189.7	1187.1	1231.2	1198.3	1188.1	1150.7
	MAP	0.5745	0.5839	0.5857	0.5861	0.5885	0.5861	0.5866	0.6064

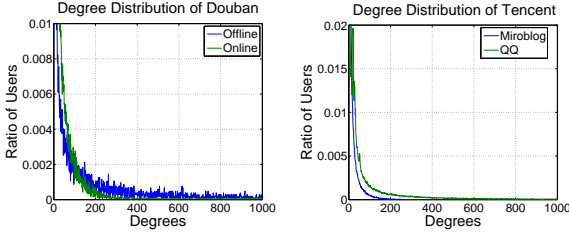


Figure 4: The Long-tail Property

4.1 Dataset Description

The Douban collection contains two subnetworks. One is the online contact network, representing who pay attention to whom, and the other is the offline physical network, indicating who are familiar with whom in the real world. Since the website does not provide APIs to collect the data of offline relations, we construct the offline network using users' co-occurrence in social gatherings, where people physically meet. If two users take part in the same gatherings more than 20 times, we consider that they know each other. Finally, the task is to predict users' behaviors on movie, music and book domains using their historical behavior logs, as well as the information in the online and offline networks. This dataset can be downloaded³.

The Tencent collection contains two subnetworks. One is QQ⁴, which contains about 1 billion users and 80 billions links, making it the biggest instant messaging network in the world, and the other is Tencent's Microblog network⁵, which contains more than 200 millions users, making it one of the biggest Microblog networks in China. We introduce two subtasks: the first one is to predict which songs users will collect in the future, which can be applied to radio applications⁶, and the second one is to predict users' profiles, which is crucial to display online advertisements. To generate users' profiles, we recorded the words from three big Web sites: QQ.com, one of the biggest portals in China, qzone⁷, the biggest social blogging service in China and wenwen⁸, one of the biggest QA systems in China. Subsequently, for a given user, 5,000 words with commercial values, such as insurance, loans, etc., were extracted from webpages that the user has browsed. These words are used to describe users and then each user is represented with a $?/1$ vector, where 1 represents a particular user browsing the corresponding word, and the question mark (?) represents unknown. As

³<http://www.cse.ust.hk/TL/dataset/Douban-50000.zip>

⁴<http://www.tencent.com/en-us/ps/imservice.shtml>

⁵http://t.qq.com/?lang=en_US

⁶<http://fm.qq.com/>

⁷<http://qzone.qq.com>

⁸<http://wenwen.soso.com/>

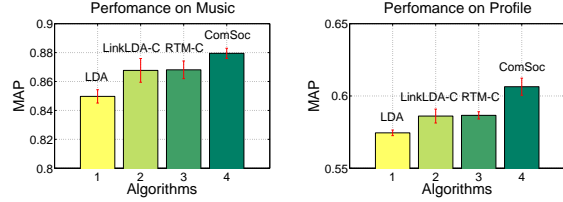


Figure 5: MAP with Error Bars of Tencent Collection (data bars are the means of MAP; the error bars are the upper and lower limits with 95% confidence.)

users' browsing contents can change over time, the task here is to predict which words the user will browse in the future.

Table 2 presents the statistics of the datasets. In summary, the Douban collection contains three types of user behavior: movie watching, music listening, and book reading, while the Tencent collection contains two other types: music collection and user profile modeling. We observe that the user-item interactions and user-user relations in each single network are very sparse. The degree distributions of the social networks in both Douban and Tencent datasets are plotted in Figure 4. We observe that degree distribution of each network follows the power-log distribution, that means each subnetwork is still sparse and most users have limited interactions. Hence we should utilize the knowledge from multiple networks together.

Data Sampling Process For the Douban dataset, we crawled the data through the APIs provided by the website. We randomly retrieved 30 senior users, who registered more than 4 years ago and are still active. Then we performed breadth-first-search based on their online contacts to collect more users iteratively, until the number of users reached a threshold (50,000 in this paper). Subsequently, the contact relations and the social gathering co-occurrence among these users, and their behaviors on music, book and movie domains were recorded. For the Tencent dataset, we firstly picked a user whose number of neighbors is around 100 randomly. Then, this user as well as his/her neighbors on QQ and Microblog networks are retrieved as a seed set. Similar to the Douban data, we performed breadth-first-search on these two networks to extend the user set until the number of users exceeded a threshold. Finally, corresponding social networks of these users, their music collection records and webpages browsing logs within two months were extracted as the experimental data. Behavior data in the final week are hold out as the test set.

4.2 Performance Comparisons

Table 3 and Figure 6 summarize the perplexity and MAP of LDA, LinkLDA, RTM and the proposed ComSoc on the

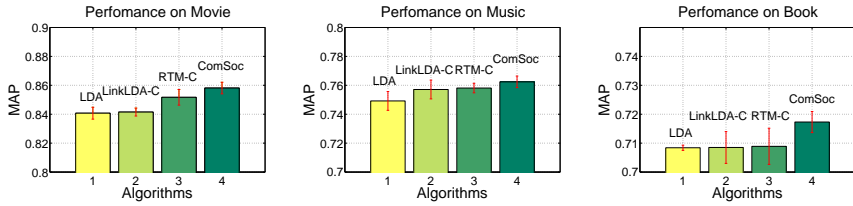


Figure 6: MAP Comparison with Error Bars of Douban Collection

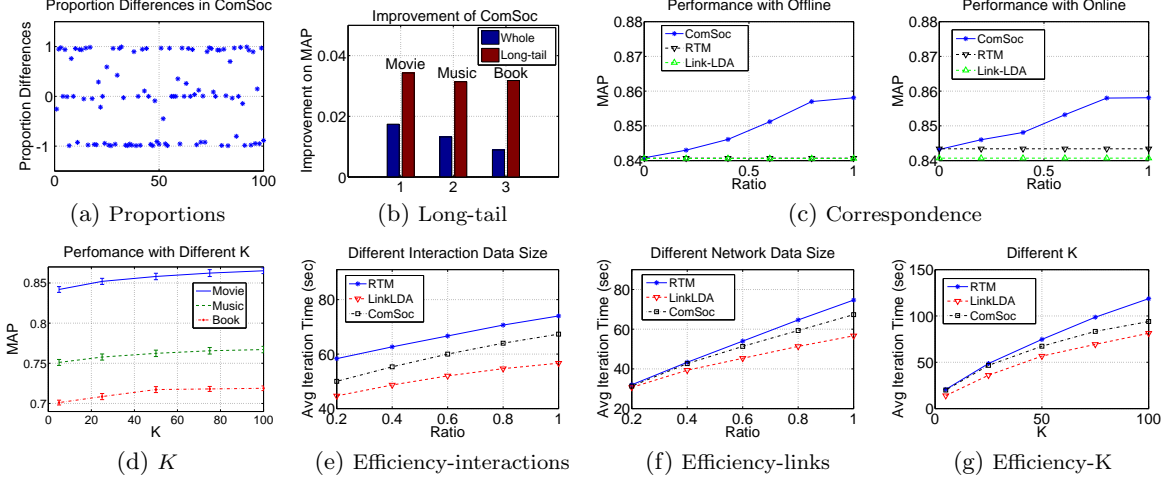


Figure 7: Performance Analysis

three datasets of the Douban collection. For the Movie dataset, as highlighted in bold, ComSoc consistently outperforms LDA, LinkLDA and RTM on MAP and perplexity. The MAP of ComSoc is at least 0.017 larger than LDA. If we overlook the model differences, compared to its competing methods, ComSoc on average achieves 0.017, 0.016 and 0.007 higher MAP as compared to LDA, LinkLDA and RTM respectively. The better performance of ComSoc over RTM and LinkLDA with one subnetwork, e.g. RTM-Online, etc., can be ascribed to the fact that, ComSoc considers more knowledge from auxiliary networks, and hence, captures more aspects of users’ interests. For example, users can share their interests through different subnetworks so some regularization knowledge will be missed if only one network is considered. On the other hand, although RTM-C and LinkLDA-C explore knowledge from multiple networks to improve their accuracies, they assign uniform regularization on users without considering the network differences to each user, i.e., different users actually receive different levels of influence from their neighbors in different social networks. In this sense, RTM-C and LinkLDA-C may introduce unnecessary regularization from unrelated networks. For example, the offline network is less useful to infer users’ interests. As shown in Table 3, LinkLDA-Off and RTM-Off do not improve the performance of LDA. However, ComSoc adaptively selects network and hence solve the above problems. Moreover, in all cases, ComSoc outperforms LDA on all three datasets, implying that when the historical interaction knowledge is sparse, social network information can actually improve the prediction performance.

Table 4 and Figure 5 summarize the results on the two datasets from Tencent. Similar to the performance in the Douban collection, ComSoc consistently outperforms RTM, LinkLDA and LDA on both perplexity and MAP for both datasets. For example, on the Profile dataset, ComSoc gets MAP higher than 0.605 while other baselines do not exceed

0.587. In addition, on the Music dataset, the improvement of ComSoc is as high as 0.03 on MAP comparing to LDA. Importantly, the improvement of ComSoc on Tencent collections is more significant. One reason is that, users share more contents through both QQ and Microblog networks, and hence the links in these two networks can reflect more users’ interests. These justify the effectiveness of the proposed methods empirically.

4.3 Performance Analysis

We conduct five extensive experiments on the Douban collection to answer the following questions: (1). Does ComSoc transfer knowledge from composite social networks to different users adaptively? (2). Does ComSoc enrich the knowledge in the user-item interaction network and therefore reduce the “sparsity”? (3). How does the correspondence ratio between composite social network and user-item interaction network affect ComSoc’s performance? (4) What is the effectiveness of the model parameters?

We answer the first question by examining the network proportions of different users in the Movie dataset captured by ComSoc. We pick 100 users who have the most behavior data, as these users have the highest confidence to derive the weights of different networks. The results are presented in Figure 7(a), where the y-axis represents the proportion difference between online and offline networks of a given user u : $\Pr(\text{network}=\text{online}|u) - \Pr(\text{network}=\text{offline}|u)$. The differences of most users are not equal to zero, meaning different networks have different impacts to different users and ComSoc utilizes the knowledge in each network adaptively.

We evaluate the results on long-tail users, whose number of behaviors are smaller than 10. These users suffer from the harm of “sparsity” the most, since each of them interacts with only $\leq 0.01\%$ items. The results are presented in Figure 7(b). We observe that, the improvement of ComSoc on long-tail users is at least twice more than the average

Table 5: Different Related Works

	Uniform	Adaptively
Single Network	RTM [3], etc.	-
Multiple Networks	Combined KNN [16]	ComSoc

level. For example, on the Book dataset, the improvement of ComSoc is smaller than 0.01 on MAP on the average level, while larger than 0.03 on long-tail users. One reason is that, for those users who have a large number of interactions, LDA has gained enough knowledge to infer their distributions over topics; but for the long-tail users, ComSoc can significantly exploit knowledge from the composite social network to help enhance the prediction. This provides justification to the argument that ComSoc can enrich the knowledge in the user-item interaction network.

In real world, many people may only use parts of on-line services, i.e., they may exist in one subnetwork but not in all networks. We test the effect of the correspondence ratio between the composite social network and the user-item interaction network, e.g., the number of users who exist in both social and interaction networks. Figure 7(c) presents the results with different ratios between the offline/online networks and the Movie interaction network. ComSoc’s performance becomes worse if fewer correspondences are provided across networks. An important observation is that it outperforms LinkLDA consistently if there are corresponding users among networks, implying that ComSoc successfully use the overlapping users as bridges to transfer knowledge.

We analyze the effect to change the number of topics K on the Douban collection. We change K from 5 to 100. The results are illustrated in Figure 7(d). We observe that the performance of ComSoc improves with the increase of K but the increment becomes smaller. The reason is that K represents the model complexity. Thus, when K is too small, the model has limited ability to describe the data. On the other hand, when K exceeds a threshold (e.g. 50, the default value in the experiment), the model is complex enough to handle the data. At this point, it is less helpful to improve the model performance by increasing K . In practice, K can be tuned through cross-validation techniques [25].

4.4 Efficiency Analysis

As analyzed in the end of Section 3.2, the computational time of ComSoc increases linearly with the number interactions between users and items, the number of links in the composite networks, and the number of topics K . We evaluate this empirically as shown in Figure 7(e)~(g). Three sub-figures from left to right illustrate the computational time of LinkLDA-C, RTM-C and ComSoc on the Movie dataset, with different ratio of interactions and links, and different number of topics respectively. We observe that the computational time linearly increases with larger data size and number of topics. For our experiment setting, each iteration takes about 80 seconds in our computer, of which memory is 4G and CPU is 2.0Gz. Although ComSoc has better prediction performance as show in Table 3 and Table 4, it is faster than RTM and has similar time cost with LinkLDA.

5. RELATED WORKS

The proposed methods can be considered as transfer collaborative filtering algorithms based on topic models. We summarize the novelty of the proposed approaches in Table 5. Most previous works either focus on one social network or treat users uniformly, where different users receive

the same influence from different social networks, but the proposed model exploits knowledge adaptively in a composite social network, which contains several subnetworks. Although the problem studied in this paper is how to solve the “sparsity” problem, sparse learning [21] is in fact a different formulation that does not solve this problem; it builds models with very few non-zero coefficients to reduce the computation cost, but does not resolve the data sparsity.

Social Networks Analysis [6] has drawn much research interest, ranging from link prediction [12], predicting how likely an unobserved edge exists between an arbitrary pair of nodes; community detection [11, 10], identifying communities of interest and studying their behaviors over time in social networks; to social influence [23], studying how the perceived relationships with other users influence the behavior of a given user. Most previous works of social influence focus on how to measure the social influence [2] or maximize the user influence [4, 8]. However, the proposed methods in this paper instead exploit social influence to help predict users’ behaviors. **Collaborative Filtering** (CF) is the process to filter information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. The proposed methods in this paper belong to the category of model-based CF which uses the observed user-item interactions to train a compact model that explains the given data. Models in this category include matrix factorization [9] and topic models [22]. Among them, topic models provide an interpretable low-dimensional representation for the interactions between users and items. The simplest topic model is latent Dirichlet allocation (LDA) [1], where a topic is a distribution over items and a user is a distribution over topics. Multiple variants have been proposed to incorporate the relational knowledge [3, 14, 7]. However, these previous works utilize only one single network. **Transfer Learning** solves the lack of class label problem in the target application by “borrowing” knowledge from related domains [15, 24]. It was on collaborative filtering recently. According to the different kinds of auxiliary data, transfer CF can be categorized into three classes. The first one is to exploit the meta data or user/item content [22]. For example, the algorithm in [22] utilizes both articles’ content and users’ interactions on articles together to perform recommendation. The second one is to borrow knowledge from other CF domains, such as transferring knowledge from a movie to a book domain [17]. The third kind is to exploit users’ or items’ relationship, such as a single trust network [13]. More recently, the work in [16] utilizes multiple social networks to recommend Apps to users using a nearest neighbor model. However, it needs to enumerate all user-item pairs and hence cannot scale up to handle large datasets. In addition, these previous works either consider one auxiliary source or regularize users modeling uniformly, which ignore the fact that different networks have different influences to different users.

6. CONCLUSION AND FUTURE WORKS

In this paper, we studied a new problem on how to transfer knowledge from composite social network for user behavior prediction, where the behavior data (or user-item matrix) of the target application can be more than 99.9% empty. We defined a composite social network as a set of nested single networks, where users and links in different subnetworks overlap. Each network reflects partial aspects of users’ interests and has different properties. For different

users, individual networks have a diverse range of influences. Thus, we proposed a hierarchical Bayesian model to transfer knowledge for different users adaptively. The knowledge from the composite social network is utilized to regularize users' distributions over topics. Unlike prior works, the proposed model considers knowledge transfer adaptively. This is formulated into a parametric model that controls the probability of how one user is influenced by a given network. In addition, different from the previous relational topic models, ComSoc exploits user-topic distributions to generate users' social relations. The proposed model is flexible in that, it can be extended to any number of networks. We proposed a large-scale implementation based on Map/Reduce to cope with massive amount of data. We conducted empirical studies on two data collections from Tencent Inc. and Douban, where ComSoc outperforms several state-of-the-art baselines by as high as 0.03 on MAP.

Future Works In this paper, we only build models to predict whether a user will interact with a given item or not. In future work, we plan to extend the proposed model to handle different kinds of behavior, such as rating, multi-class prediction, etc. We will extend the experiments over multiple social networks, such as Facebook + Twitter + MSN. We will consider incorporating heterogeneous auxiliary knowledge together, such as item profiles, user behaviors in auxiliary domains, etc.

Acknowledgement

We thank the support of Hong Kong RGC GRF Projects 621010 and 621211.

7. REFERENCES

- [1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [2] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the 4th International Conference on Weblogs and Social Media, ICWSM '10*, pages 10–17. AAAI, 2010.
- [3] J. Chang and D. M. Blei. Relational topic models for document networks. *J. Mach. Learn. Res.*, 5:81–88, 2009.
- [4] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pages 199–208. ACM, 2009.
- [5] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51:107–113, 2008.
- [6] 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. Springer-Verlag, 2010.
- [7] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5220–5227, 2004.
- [8] D. Kempe, J. Kleinberg, and E. Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32nd International Conference on Automata, Languages and Programming, ICALP'05*, pages 1127–1138. Springer-Verlag, 2005.
- [9] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*, pages 426–434. ACM, 2008.
- [10] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web, WWW '08*, pages 695–704. ACM, 2008.
- [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. ACM, 2010.
- [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. ACM, 2003.
- [13] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on World Search and Data Mining, WSDM '11*, pages 287–296. ACM, 2011.
- [14] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*, pages 542–550. ACM, 2008.
- [15] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [16] W. Pan, N. Aharony, and A. Pentland. Composite social network for predicting mobile apps installation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence, AAAI '11*, pages 821–827. AAAI, 2011.
- [17] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI '10*, pages 230–235. AAAI, 2010.
- [18] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*, pages 569–577. ACM, 2008.
- [19] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web, WWW '01*, pages 285–295. ACM, 2001.
- [20] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Proc. VLDB Endow.*, 3:703–710, 2010.
- [21] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.
- [22] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 448–456. ACM, 2011.
- [23] 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. ACM, 2010.
- [24] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. Turaga, and O. Verscheure. Cross domain distribution adaptation via kernel mapping. In *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1027–1036. ACM, 2009.
- [25] E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Proceedings of the 2010 European Conference on Machine learning and Knowledge Discovery in Databases: Part III, ECML PKDD'10*, pages 547–562. Springer-Verlag, 2010.