On Top-N Recommendation Using Implicit User Preference Propagation over Social Networks

Jun Zou and Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332, USA
Email: {junzou, fekri}@ece.gatech.edu

***Abstract*—Social recommender systems exploit the historic user data as well as user relations in the social networks to make recommendations. However, users are increasingly concerned with their online privacy, and hence, they are not willing to reveal their personal data to the general public. In this paper, we propose a social recommendation algorithm for top-N recommendation using only implicit user preference data. In particular, we model users' consumption behavior in the social network with Bayesian networks, using which we can infer the probabilities for items to be selected by each user. We develop an Expectation Propagation (EP) message-passing algorithm to perform approximate inference efficiently in the constructed Bayesian network. The original proposed algorithm is a central scheme, in which the user data are collected and processed by a central authority. However, it can be easily adapted for a distributed implementation, where users only exchange messages with their directly connected friends in the social network. This helps further protect user privacy, as users do not release any data to the public. We evaluate the proposed algorithm on the Epinions dataset, and compare it with other existing social recommendation algorithms. The results show its superior top-N recommendation performance in terms of recall.**

I. Introduction

To counter the information overloading problem on the Internet, recommender systems have been widely employed by online service websites to suggest items, e.g., movies and books, to users who might like them [1]. In typical practical implementations, a list of N items are rendered to an active user which he may find the most interesting. This is known as the *top-N recommendation* task. There are a variety of recommendation algorithms including content-based recommendations and collaborative recommendations. In content-based recommendations, each item is characterized by a set of attributes, based on which item similarity is estimated, and the active user will be recommended items similar to the ones he liked in the past. However, the content-based recommendation system suffers from limited content-analysis, e.g., it is difficult to explicitly describe multimedia data using features. Instead, in collaborative recommendations, also known as collaborative filtering, the active user will be recommended items favorably rated by other users with similar tastes to the active user. The user similarity can be estimated based on user profiles including detailed personal information, but due to privacy concerns they are very difficult to obtain. Hence, the collaborative recommender systems evaluate user similarity based on users' historic rating data. This causes the cold-start problem for new users or users who do not provide enough ratings, i.e., the recommender systems cannot find similar users for them.

Recently, with the thriving of online social networks (OSN), the social collaborative recommendation has attracted significant attention. In social networks, people are more likely to connect to other people sharing similar interests, and they are influenced more by people they connect to, further fostering similarity to each other [2]. Hence, by exploiting the social structure of social networks, social recommender systems can make satisfactory recommendations even for cold-start users when provided with their social connections. Moreover, social recommendation can also be conveniently incorporated with traditional collaborative filtering algorithms, e.g., matrix factorization [3]–[6] and neighborhood methods [6], [7], improving their recommendation performance, especially for cold-start users.

Today, people are increasingly concerned with their online privacy, adding new challenges to the recommendation problem. Although rating data do not directly tell personal details, it is still possible to infer user demographics, such as age and gender, from their ratings [8], and even uncover user identities and reveal sensitive personal information with access to other databases [9]. Therefore, users are not willing to make their personal data accessible to the general public. Hence, when designing collaborative recommendation systems, we need to take into account user privacy. Several works proposed to obfuscate user ratings with random noise, e.g., perturbation [10] and differential privacy [11], or to disguise genuine user profiles by adding extra fake data [8], [12]. However, such obfuscation techniques do not completely prevent rating data leakage. Alternatively, recommender systems can utilize only implicit data, e.g., whether a user has consumed an item or not, avoiding exposing explicit user rating data.

In this paper, we propose a social recommendation algorithm that exploits the social network to generate recommendations using implicit user data. We develop a probabilistic Bayesian Network (BN) model for item consumption in the social network, and infer the probability for each item to be selected by the active user given the observed implicit user data. Then, the top-N items with the highest probabilities are listed in the recommendation. Specifically, we propose an Expectation Propagation (EP) message-passing algorithm for approximate inference in the BN, based on which we further develop an iterative EP algorithm that performs EP message-passing for all users in a unified bipartite graph representation. We note that the original algorithm is a central scheme, in which the user data are collected and processed by a central authority. However, the algorithm can be easily adapted for distributed implementation, where users only exchange messages with

friends they are directly connected to in social networks. This helps further protect user privacy, since users only share data with their immediate friends. Previously, in [13] we developed probabilistic factor graph models for similarity computation in collaborative filtering recommender systems, and employed Belief Propagation (BP) to perform inference efficiently. In [14], we proposed privacy-preserving item-based collaborative filtering, where users send out probabilistic messages on item similarities without revealing personal rating data.

II. Related Works

Social networks have been exploited to enhance traditional collaborative filtering recommender systems. In [3]–[6], the social network was incorporated into matrix factorization methods, where the user latent factors are learnt with social relations taken into account. All of them require users' explicit rating data. Besides, performing matrix factorization is computationally intensive, which is not suitable for frequent updates of the recommender system with new rating records.

[6], [7] proposed social recommendation algorithms that can be combined with traditional neighborhood methods. In [7] the authors proposed a random walk approach TrustWalker, where the recommendation algorithm carries out random walk over the social network to collect ratings from other users, and computes the average ratings to generate recommendations for the active user. Obviously, user ratings are revealed to all users connected to the social network. [6] proposed a voting-based algorithm PureTrust, using only implicit user data. The algorithm collects votes from users found via Breadth First Search (BFS) in the social network. In contrast to [6], in our work we avoid direct information exchange between indirectly connected friends. Further, if a user has not consumed an item, instead of simply collecting a '0' vote for that item, our work assigns a random binary variable to model the willingness of the user to select that item, whose probability distribution depends on his directly connected friends.

The Bayesian networks were applied to recommender systems in [15], [16]. In [15], a Bayesian network is employed to model probabilistic item-based rating prediction, where each node corresponds to an item and its parent nodes correspond to similar items. In [16], the Bayesian network is also used to predict user ratings, but each node corresponds to an user in the social network. Different from these approaches, our work is interested in inferring the probabilities for items to be selected by users using only implicit data.

III. Top-N Recommendation Using Social Networks

A. Problem Description

We assume a set of M users denoted by $\mathbb{U} = \{1, \ldots, M\}$ and a set of L items denoted by $\mathbb{I} = \{1, \ldots, L\}$ in the social recommender system. Let U_i denote the set of users who have consumed item i, and I_u denote the set of items consumed by user u. Note that user u may also provide explicit feedback on item $i \in I_u$ in the form of rating r_{ui}, but users keep such explicit data private. We arrange the collection of observed implicit data in a $M \times N$ matrix $\mathbb{S}$, where for each user u we place a '1' at the intersection of the u-th row and i-th column if $i \in I_u$ and leave the rest of the entries unfilled.

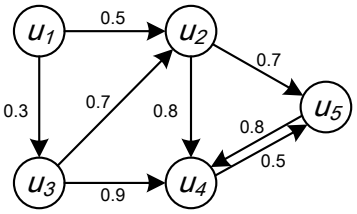


Fig. 1: Graphical representation of the social network.

Further, we assume that there is a social network established among users, where each user u is directly connected to a set of friends denoted by T_u. Let w_{uv} denote the trust value user u assigns to user $v \in T_u$, where larger trust value means higher degree of trust. w_{uv} can be some continuous value, e.g., $w_{uv} \in [0, 1]$, or can be binary value, where '1' indicates the existence of trust relations and '0' indicates unknown or no trust relations. Note that the trust relations are generally directed and asymmetric. We can represent the social network in a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices representing users in $\mathbb{U}$, and $\mathcal{E}$ is the set of edges representing social relations between users. In Fig. 1 we illustrate the graphical representation of a social network, where the weight on the edge is the trust value.

Given an active user z, the top-N recommendation task is to generate a list of N items from $\mathbb{I} \backslash I_z$ for user z. In this work, we focus on exploiting the social relations for collaborative recommendation. We construct probabilistic Bayesian networks to model the item consumption in the social network, and infer the probabilities for items to be selected by the active user given the observed data in $\mathbb{S}$. Then the top-N items with the highest probabilities are recommended to the active user.

B. Social Network-Based Bayesian Network

We define a binary variable s_{ui} to represent whether or not item i is selected by user u, $s_{ui} = 1$ if selected and $s_{ui} = 0$ otherwise. Hence, $s_{ui} = 1$ for item $i \in I_u$. However, for other items $i \in \mathbb{I} \backslash I_u$, instead of simply setting $s_{ui} = 0$, we consider them as unknown, since our goal is to recommend items from $\mathbb{I} \backslash I_u$ to user u. We are interested in inferring the probability distribution of s_{ui}, denoted by $BEL(s_{ui})$, $\forall i \in \mathbb{I} \backslash I_u$, given the observed data in $\mathbb{S}$. We make such inference following the fact that a user is most influenced by the friends he directly connect to in the social network.

We denote by $\mathcal{N}(u, K)$ a subset of K users with the highest trust values from user u's friend set T_u. To determine $\mathcal{N}(u, K)$ in scenarios where users do not explicitly specify trust values, e.g., assuming one for all trusted users, we first compute w_{uv}, $\forall v \in T_u$, as

$$w_{uv} = 1 + \frac{|I_u \cap I_v|}{|I_u \cap I_v| + c}, \tag{1}$$

where $c > 0$ is a constant. If user u has selected many items in common with user v in the past, user u is supposed to have high trust on user v. Given an active user z, to infer $BEL(s_{zi})$ for item i, we construct a Bayesian network $\mathcal{G}_{zi}$ as illustrated in Fig. 2. The root node at Layer 0 represents variable s_{zi}. Its parent nodes at Layer 1 are denoted by $\mathcal{P}_{zi} = \{s_{vi} : v \in \mathcal{N}(z, K)\}$, which correspond to the friends of user z. Similarly,

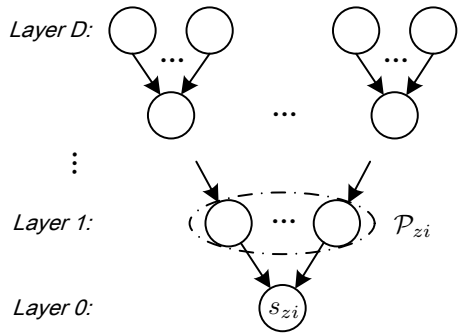


Fig. 2: Bayesian network $\mathcal{G}_{zi}$ for active user z.

for each node s_{vi} at Layer 1, we place its parent nodes at Layer 2. We repeat this process until Layer D. According to the conditional independence in the Bayesian network, each node s_{ui} is conditionally independent of other non-descendant nodes, given the configuration of its parent nodes $\mathcal{P}_{ui}$. We describe the conditional probability $P(s_{ui}|\mathcal{P}_{ui})$ as

$$P(s_{ui}=1|\mathcal{P}_{ui})=\frac{\sum_{v\in\mathcal{N}(u,K)} w_{uv}s_{ui}}{\sum_{v\in\mathcal{N}(u,K)} w_{uv}}, \tag{2}$$

where $P(s_{ui}=0|\mathcal{P}_{ui})=1-P(s_{ui}=1|\mathcal{P}_{ui})$.

C. Inference and Expectation Propagation

Before introducing the inference process, we specify the observed and hidden variables in $\mathcal{G}_{zi}$. Let $\mathcal{V}_i=\{s_{vi}: v\in U_i\}$ be the set of observed variables for users who have consumed item i. Then $s=1$, $\forall s\in\mathcal{V}_i$. Further, we assume the set $\mathcal{L}_{zi}(D)$ of variables at Layer D are also observed. Specifically, we set $s=0$ for $s\in\mathcal{L}_{zi}(D)\cap\bar{\mathcal{V}}_i$, where $\bar{\mathcal{V}}_i=\{s_{vi}: v\in\mathbb{U}\backslash U_i\}$. We treat the rest of the variables in $\bar{\mathcal{V}}_i\backslash\mathcal{L}_{zi}(D)$ as hidden variables whose values are unknown.

We employ the Belief Propagation (BP) algorithm [17] to perform inference. However, exact inference can be intractable due to loops in the Bayesian network $\mathcal{G}_{zi}$. We instead resort to approximate inference by assuming a tree structure for $\mathcal{G}_{zi}$. Then inferring $BEL(s_{zi})$ is completed by passing probabilistic messages along the edges from the variable nodes at Layer d to the ones at Layer $d-1$, starting from Layer D. Specifically, we compute the message $\pi_a(s_{bi})$ sent to node s_{ai} at Layer $d-1$ from node $s_{bi}\in\mathcal{P}_{ai}$ at Layer d as follows

$$\pi_a(s_{bi})=\alpha\sum_{\mathcal{P}_{bi}}P\left(s_{bi}|\mathcal{P}_{bi}\right)\prod_{q\in\mathcal{N}(b,K)}\pi_b(s_{qi}), \tag{3}$$

where α is a normalization factor such that $\sum_{s\in\{0,1\}}\pi_a(s_{bi}=s)=1$. Note that if s_{bi} is observed as $s_{bi}=\hat{s}_{bi}$, $\hat{s}_{bi}\in\{0,1\}$, then $\pi_a(s_{bi}=\hat{s}_{bi})=1$ and $\pi_a(s_{bi})$ will not be updated. $\pi_a(s_{bi})$ expresses the probability distribution of s_{bi}, given the new evidence acquired by s_{bi} from incoming messages. Each node s_{ui} in $\mathcal{G}_{zi}$ generates messages sent to its children nodes once all messages from $\mathcal{P}_{ui}$ have arrived at s_{ui}.

The computational complexity in (3) is $\mathcal{O}\left(K2^K\right)$, which grows exponential with K. However, (3) can be simplified as

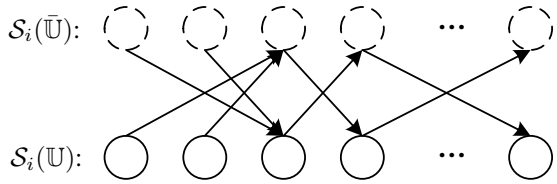


Fig. 3: Iterative EP in a unified bipartite graph $\mathcal{G}_i$.

following. Substituting (2) into (3), we have

$$\begin{aligned}\pi_a(s_{bi}=1)&=\alpha\frac{\sum_{q\in\mathcal{N}(b,K)}\sum_{\mathcal{P}_{bi}} w_{bq}s_{qi}\prod_{q\in\mathcal{N}(b,K)}\pi_b(s_{qi})}{\sum_{q\in\mathcal{N}(b,K)} w_{bq}}\\&=\alpha\sum_{q\in\mathcal{N}(b,K)}\hat{w}_{bq}\mathbb{E}_{\pi_b}(s_{qi}),\end{aligned} \tag{4}$$

where we define $\hat{w}_{bq}\triangleq\frac{w_{bq}}{\sum_{y\in\mathcal{N}(b,K)} w_{by}}$, and denote $\mathbb{E}(\cdot)$ as expectation operation. Note that $\mathbb{E}_{\pi_b}(s_{qi})$ is carried out with respect to the distribution $\pi_b(s_{qi})$. By using (4), the computational complexity of generating a single message is reduced to $\mathcal{O}(K)$. Moreover, since $\mathbb{E}_{\pi_a}(s_{bi})=\pi_a(s_{bi}=1)$, we can rewrite (4) as

$$\mathbb{E}_{\pi_a}(s_{bi})=\alpha\sum_{q\in\mathcal{N}(b,K)}\hat{w}_{bq}\mathbb{E}_{\pi_b}(s_{qi}). \tag{5}$$

We hereafter refer to message passing using (5) as Expectation Propagation (EP). Finally, when all messages from nodes $\mathcal{P}_{zi}$ have arrived at s_{zi}, we compute $BEL(s_{zi})$ as

$$BEL(s_{zi}=1)=\sum_{v\in\mathcal{N}(z,K)}\hat{w}_{zv}\mathbb{E}_{\pi_z}(s_{vi}). \tag{6}$$

We would like to clarify that, the propagation of messages between nodes is performed in a central server that collects and processes user data. Hence, there is no actual exchanges of messages between users in real world. However, the algorithm can be readily adapted for distributed implementation as explained in Sec. III-E.

D. Iterative Expectation Propagation

In Secs. III-B and III-C, we have shown that to infer $BEL(s_{zi})$ for an active user z, we need to construct a Bayesian network $\mathcal{G}_{zi}$ consisting of D layers of variable nodes. However, the intermediate results during the inference process in $\mathcal{G}_{zi}$ are not fully utilized. Indeed, there are significant amounts of repeated computations when performing inference for different users. In the following, we propose an iterative EP algorithm to simultaneously compute $\{BEL(s_{ui}): u\in\mathbb{U}\}$ of all users for each item i in a single unified bipartite graph $\mathcal{G}_i$ that concisely encodes all social connections.

We create a set of virtual users, denoted by $\bar{\mathbb{U}}=\{\bar{1},\ldots,\bar{M}\}$, where each virtual user $\bar{u}\in\bar{\mathbb{U}}$ is an image of a unique user $u\in\mathbb{U}$. We also assign a variable $s_{\bar{u}i}$ for each user $\bar{u}$. We represent the variables in $\mathcal{S}_i(\mathbb{U})=\{s_{ui}: u\in\mathbb{U}\}$ and $\mathcal{S}_i(\bar{\mathbb{U}})=\{s_{\bar{u}i}: \bar{u}\in\bar{\mathbb{U}}\}$ as two rows of nodes as illustrated in Fig. 3. Let $\bar{\mathcal{N}}(u,K)$ denote the set of images of users in $\mathcal{N}(u,K)$. For each variable node $s_{ui}\in\mathcal{S}_i(\mathbb{U})$, we connect to it the image nodes $\bar{\mathcal{P}}_{ui}=\{s_{\bar{v}i}: \bar{v}\in\bar{\mathcal{N}}(u,K)\}$ rather than its

true parent nodes $\mathcal{P}_{ui}$. Similarly, we connect each $s_{\bar{u}i} \in \mathcal{S}_i(\bar{\mathbb{U}})$ to $\mathcal{P}_{ui}$ instead of $\bar{\mathcal{P}}_{ui}$. Then all layers of all concerned Bayesian networks are included in this unified representation $\mathcal{G}_i$. To see this, we can consider $\mathcal{S}_i(\bar{\mathbb{U}})$ and $\mathcal{S}_i(\mathbb{U})$ respectively as Layer $d+1$ and Layer d, when messages are sent from $\mathcal{S}_i(\bar{\mathbb{U}})$ to $\mathcal{S}_i(\mathbb{U})$ along the directed edges, and as Layer $d-1$ and Layer d, when messages are sent to $\mathcal{S}_i(\bar{\mathbb{U}})$ from $\mathcal{S}_i(\mathbb{U})$. Hence, the EP message-passing in any multi-layer Bayesian network can be carried out in $\mathcal{G}_i$ by iteratively passing messages between $\mathcal{S}_i(\bar{\mathbb{U}})$ and $\mathcal{S}_i(\mathbb{U})$. Moreover, we can simultaneously infer all $\{BEL(s_{ui}) : u \in \mathbb{U}\}$ in $\mathcal{G}_i$, reusing intermediate results and thus reducing overall computational complexity.

Following (5), the messages exchanged between variable nodes $\mathcal{S}_i(\bar{\mathbb{U}})$ and $\mathcal{S}_i(\mathbb{U})$ are given as follows

$$\mathbb{E}_{\pi_{\bar{u}}}(s_{vi}) = \sum_{\bar{q} \in \bar{\mathcal{N}}(v,K)} \hat{w}_{v\bar{q}} \mathbb{E}_{\pi_v}(s_{\bar{q}i}), \quad \forall s_{vi} \in \mathcal{S}_i(\mathbb{U}), \tag{7}$$

$$\mathbb{E}_{\pi_u}(s_{\bar{v}i}) = \sum_{q \in \mathcal{N}(\bar{v},K)} \hat{w}_{\bar{v}q} \mathbb{E}_{\pi_{\bar{v}}}(s_{qi}), \quad \forall s_{\bar{v}i} \in \mathcal{S}_i(\bar{\mathbb{U}}). \tag{8}$$

Note that passing messages using (7) and (8) for a total of D times is equivalent to performing inference in D-Layer Bayesian networks shown in Fig. 2. We initialize the messages as in Sec. III-C. Assuming it starts from (7), we set $\mathbb{E}_{\pi_u}(s_{\bar{v}i}) = 1$ if $v \in U_i$, and $\mathbb{E}_{\pi_u}(s_{\bar{v}i}) = 0$ otherwise. In addition, during iterations, we always set both $\mathbb{E}_{\pi_u}(s_{\bar{v}i})$ and $\mathbb{E}_{\pi_{\bar{u}}}(s_{vi})$ to one for $v \in U_i$.

The resulting iterative EP inference algorithm is executed for every item in $\mathbb{I}$ in order to generate top-N recommendations. After obtaining $\{BEL(s_{ui}) : u \in \mathbb{U}\}$, $\forall i \in \mathbb{I}$, for each user u, we rank the items in $\mathbb{I} \backslash I_u$ in descending order of $BEL(s_{ui} = 1)$, and recommend the top-N items to user u. We observe that both steps (7) and (8) have the computational complexity of $\mathcal{O}(K)$. Then, the overall complexity of inferring $\{BEL(s_{ui}) : u \in \mathbb{U}\}$, $\forall i \in \mathbb{I}$, is $\mathcal{O}(DLMK)$, when using the proposed iterative EP algorithm to perform inference for all users simultaneously in the unified bipartite graph with D steps of message passing for each item.

E. Distributed Implementation

The proposed EP message-passing algorithm is very suitable for distributed implementation. To acquire recommendations from the social network, an active user z sends request to his directly connected friends $\mathcal{N}(z,K)$ for their information about items. In addition, user z also sends a decremental counter C with initial value D to its friends. If a friend $v \in \mathcal{N}(z,K)$ has consumed item i, he will send the message $\mathbb{E}_{\pi_z}(s_{vi}) = 1$ back to the requester user z. Otherwise, user v first decreases the counter C by 1, and if C is still greater than 0, he sends a request to his directly connected friends $\mathcal{N}(v,K)$ for information on item i, along with the counter C, otherwise he sends the message $\mathbb{E}_{\pi_z}(s_{vi}) = 0$ back to user z. In cases that user v does send request to his friends, each friend u in $\mathcal{N}(v,K)$ repeats the same procedure as user v undergoes. Suppose user u sends request to his friends. Then he waits for response messages from $\mathcal{N}(u,K)$. Upon receiving all needed messages, user u generates the message $\mathbb{E}_{\pi_v}(s_{ui})$ according to (5), which is then sent back to user v. After the requester user z receives all messages from $\mathcal{N}(z,K)$, he calculates $BEL(s_{ui})$ according to (6).

In the distributed implementation of message-passing, the users only exchange messages with their directly connected friends in the social network. There is no need for users to reveal their data to the public. Also, each user mixes the messages received from his friends to generate a new message. Hence, it is difficult for users to extract the original data of friends of his friends from the received messages. Therefore, user privacy is further protected.

IV. Experimental Results

We evaluate the top-N recommendation performance of the proposed EP algorithm using the Epinions[1] dataset prepared by [18]. The dataset consists of 49,290 users and 139,738 items. A total of 664,824 ratings are given by users on items, and each rating is an integer between 1 and 5. The dataset also includes 487,181 directed trust statements with trust value one. We randomly select 20% users for testing, and the rest for training. As in [6], [7], we define cold start users as users who have less than 5 ratings. Of the testing users, 41.5% are cold start users (52.8% in the overall dataset). For each user u in the testing set $\mathbb{T}$, we withhold an item, denoted by W_u, which has the maximum rating among items rated by this user. The performance of top-N recommendation is measured by recall computed as

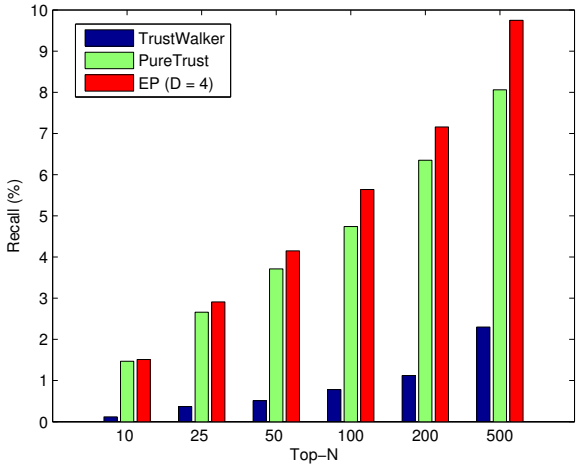
$$\text{Recall}(N) = \frac{\sum_{u \in \mathbb{T}} HIT(W_u, I_u(N))}{|\mathbb{T}|}, \tag{9}$$

where $I_u(N)$ is the set of top-N items recommended to user u, and $HIT(\cdot) = 1$ if $W_u \in I_u(N)$ and $HIT(\cdot) = 0$ otherwise. Given N, higher recall means better performance.

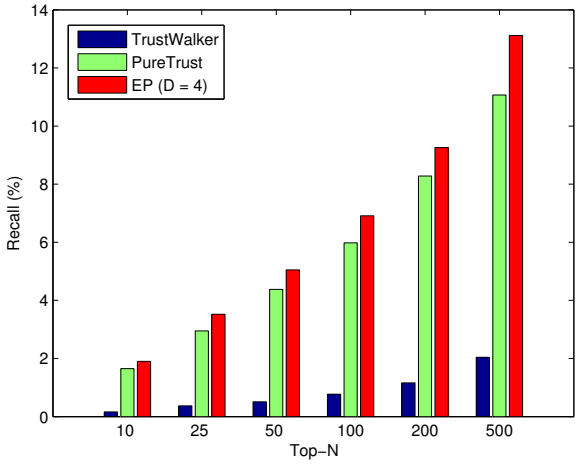
We compare the performance of the proposed EP algorithm, the random walk based TrustWalker algorithm [7], and the voting-based PureTrust algorithm [6]. Note that due to the sparsity of the dataset, the recalls are quite low for all simulated recommendation algorithms when the number of recommended items is too small. In Fig. 4, we present the recall results for these algorithms for top-N recommendations when tested on (a) cold start users, and (b) all users. We set the neighborhood size K as 10 for both EP and PureTrust, and set D of EP as 4. We also set $c = 5$ in (1). The TrustWalker does not have a neighborhood size specification as it performs random walks over all possible users. The results indicate that the proposed EP outperforms the PureTrust algorithm. The achieved improvement for cold start users is 19% and 21% for top-100 and top-500 recommendations, respectively. The TrustWalker has the worst results as similarly observed in [6]. This is because many rating predictions generated by TrustWalker will have equal values, and the top-N items are selected without considering their popularity, which include many rarely rated items that have received one or two 5-ratings. As for computational complexity, to generate recommendations for all users, the proposed iterative EP requires $\mathcal{O}(DLMK)$ as discussed in Sec. III-D, comparable to $\mathcal{O}(LMK)$ of PureTrust.

We further investigate the impacts of parameters K and D on the performance. To better illustrate the effects, we report results for top-500 recommendations, considering the sparsity of the dataset. In Fig. 5, we provide the top-500 recall results

[1] http://www.trustlet.org/wiki/Epinions_datasets.



(a) Cold start users



(b) All users

Fig. 4: Comparison of recall results of top-N recommendations.

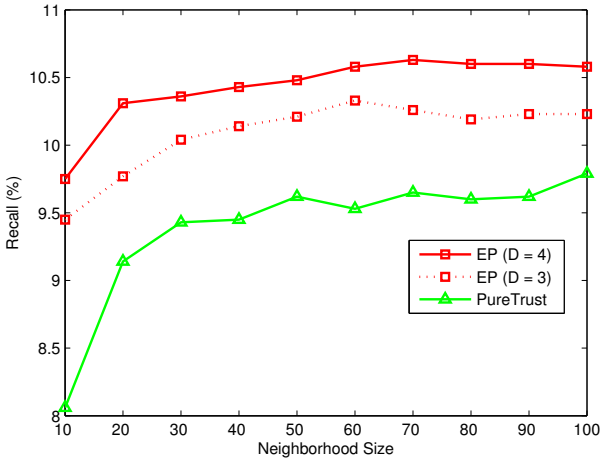


Fig. 5: Top-500 recall versus neighborhood size.

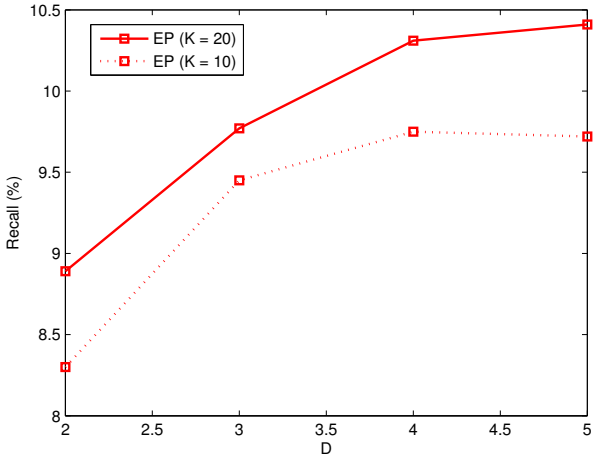


Fig. 6: Top-500 recall versus D.

on cold start users under varying neighborhood size K. The results of EP are shown for both $D = 3$ and 4. It can be seen that EP achieves better results than PureTrust for all K from 10 to 100. In Fig. 6, we investigate the impact of D. We show the top-500 recall results on cold start users for D from 2 to 5, when $K = 10$ and 20. We observe that increasing D from 2 to 4 improves recall, but then as D increases further, the performance starts to saturate or even drop. This is because increasing D too high would allow the propagation of information from users that are further away from the active user in the trust network.

V. Conclusions

In this paper, we propose a message-passing based social recommendation algorithm that exploits the social relations between users in the social network to generate top-N recommendations, using only implicit user preference data. We model the probabilities for items to be selected by the active user in Bayesian networks constructed based on the social network, and develop the EP message-passing algorithm to perform approximate inference efficiently. We also propose an iterative EP algorithm to carry out EP message-passing for all users simultaneously in a unified bipartite graph representation. Moreover, the proposed message-passing algorithm is suitable for distributed implementation, where users only exchange messages with their directly connected friends. The experimental results on the Epinions dataset show that the proposed EP algorithm achieves better top-N recommendation performance in terms of recall than both the random walk based and voting based social recommendation algorithms, while its computational complexity is comparable to theirs.

VI. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1115199.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 734–749, Jun. 2005.

[2] S. Wasserman and K. Faust, *Social Network Analysis*. Cambridge Univ. Press, 1994.

[3] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, 2008.

[4] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of ACM SIGIR Conference on Research and development in information retrieval (SIGIR)*, 2009.

[5] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of ACM Conference on Recommender Systems (RecSys)*, 2010.

[6] X. Yang, H. Steck, Y. Guo, and Y. Liu, "On top-k recommendation using social networks," in *Proceedings of ACM Conference on Recommender Systems (RecSys)*, 2012.

[7] M. Jamali and M. Ester, "Using a trust network to improve top-n recommendation," in *Proceedings of ACM Conference on Recommender Systems (RecSys)*, 2009.

[8] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "BlurMe: Inferring and obfuscating user gender based on ratings," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, Dublin, Ireland, 2012, pp. 195–202.

[9] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, 2008, pp. 111–125.

[10] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Proceedings of the Third IEEE International Conference on Data Mining*, 2003, pp. 625–628.

[11] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the net," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 627–636.

[12] R. Shokri, P. Pedarsani, G. Theodorakopoulos, and J.-P. Hubaux, "Preserving privacy in collaborative filtering through distributed aggregation of offline profiles," in *Proceedings of the Third ACM Conference on Recommender Systems*, 2009, pp. 157–164.

[13] J. Zou, A. Einolghozati, E. Ayday, and F. Fekri, "Iterative similarity inference via message passing in factor graphs for collaborative filtering," in *Proceedings of IEEE Information Theory Workshop (ITW'13)*, Seville, Spain, 2013.

[14] J. Zou, A. Einolghozati, and F. Fekri, "Privacy-preserving item-based collaborative filtering using semi-distributed belief propagation," in *Proceedings of the First IEEE Conference on Communications and Network Security (CNS'13)*, Washington, D.C., USA, 2013.

[15] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, 1998, pp. 43–52.

[16] X. Yang, Y. Guo, and Y. Liu, "Bayesian-inference based recommendation in online social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 642–651, Apr. 2013.

[17] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[18] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of ACM Conference on Recommender Systems (RecSys)*, 2007, pp. 17–24.