Social recommendation based on users’ attention and preference

Jiawei Chen, Can Wang*, Qihao Shi, Yan Feng, Chun Chen

ZJU-LianlianPay Joint Research Center College of Computer Science, Zhejiang University, Hangzhou 310027, China

A R T I C L E   I N F O

Article history:
Received 30 May 2018
Revised 11 February 2019
Accepted 21 February 2019
Available online 4 March 2019
Communicated by Dr. Guan Ziyu

Keywords:
Social recommendation
Preference
Attention
Graphical model

A B S T R A C T

Attention is the behavioral and cognitive process of selectively concentrating on small fraction of information while ignoring other perceivable information [1–3]. Therefore our attention is selective and limited by its nature [4–6]. Users tend to capture information that supports their biased pre-existing views from their prior experience and social network [7,8]. Thus it is not surprising when presented a list of recommendations, we will focus more on the items that draw our attention and skip some other items even if they suit our tastes. This psycho-social effect motivates us to model users’ attention in social recommendation for better recommendation accuracy.

To better understand the characteristics of users’ attention in social recommendation, we conduct empirical analyses based on four large real-world datasets in this paper. Specifically, in recommendation, we explore users’ attention based on their attention behavior, i.e. which items the user have spent mental effort on consuming and rating. An important conclusion can be drawn: Users’ attention behavior is heavily affected by their trust relations, while their rating values stay relatively indifferent to them. Thus, on the one hand, social information can be used more effectively to infer users’ attention than users’ preference. On the other hand, each of us belongs to some content-shared communities and our attention will inevitably be influenced by our social relations [2,9].

* Corresponding author.
E-mail addresses: sleegyhunt@zju.edu.cn (J. Chen), wcann@zju.edu.cn (C. Wang), shiqihao321@zju.edu.cn (Q. Shi), fengyan@zju.edu.cn (Y. Feng), chenc@zju.edu.cn (C. Chen).

https://doi.org/10.1016/j.neucom.2019.02.045
0925-2312/© 2019 Published by Elsevier B.V.
which makes that our attention is not coincident with our preference. Thus, just considering one aspect for recommendation may not produce effective recommendation results. Specifically, when presented a list of recommendations, we usually focus more on the items that draw our attention, and are unwilling to spend effort to learn the information about the rest. This means that the recommended items just suitting our tastes may be ignored because of our limited and selective attention [4,5]. Meanwhile, the items which just draw our attention may not be accepted, since when we make the decisions on the consumption, we usually observe ourselves and judge whether we will like them. Therefore, to improve the recommendation accuracy, in modern recommender systems, how items draw users' attention and how items suit users' tastes should be both taken into consideration.

Based on the above analyses, we propose a new social recommendation model HTPF that explicitly considers both users' attention and preference for better recommendation accuracy. Besides, different from existing methods that model social influence on users' rating values, HTPF differs from these methods in that it uses social information to better infer users' attention, which is more suspectable to social network. Considering Poisson factorization [10] is particularly efficient on sparse data and usually achieves better performance than traditional methods (such as PMF [11], similar-based methods [12], random walk [13]), HTPF employs an integrated latent factor model based on Poisson factorization [10]: HTPF fuses the Poisson factorization of attention behavior information and trust information by sharing common latent vectors of users' attention, since users who have similar social relations tend to pay attention to similar items. At the same time, HTPF fuses the Poisson factorization of attention behavior information and rating information by sharing common latent vectors of items, because of constant item attributes. A scalable variational inference algorithm for our HTPF model has been developed to predict how the specific items draw users' attention and how the items suit users' tastes. Targeting at high recommendation accuracy, HTPF generates recommendations based on the weighted combination of these two aspects. We also conduct extensive experiments and demonstrate that by combining users' attention and preference our method outperforms state-of-the-art social recommendation methods.

It is worthwhile to highlight the following contributions of our work:

- We introduce user's attention in the social recommendation and uncover that the influence of trust relations dwells more on users' attention than on their preference.
- We propose a novel probabilistic model HTPF that explicitly models both user's attention and preference for social recommendation. Besides, We develop an efficient inference method for HTPF.
- Our experiments on real-world datasets clearly demonstrate the superiority of our method over existing social recommendation methods.

Before proceeding further, we now formalize the problem definition. Suppose we have a recommender system with users set \( U \) (including \( n \) users) and items set \( I \) (including \( m \) items). Rating information is represented as \( n \times m \) matrix \( R \) whose entries (\( R_{ui} \)) denote the rating value of item \( i \) given by user \( u \). Attention behavior information indicates observed attention behaviors that users are willing to spend mental effort on consuming and rating some specific items [1,14], represented as \( n \times m \) matrix \( \Omega \). For each entry in \( \Omega \), \( \Omega_{ui} = 1 \) denotes that user \( u \) has consumed and rated item \( i \), and \( \Omega_{ui} = 0 \) means not. Trust information indicates the trust relations between users, represented as \( n \times n \) matrix \( G \). For each entry in \( G \), \( G_{uv} = 1 \) denotes user \( u \) trusts user \( v \), \( G_{uv} = 0 \) denotes not (Note that we can treat undirected friend relations as bi-directed trust relations in this paper). The task of social recommendation is to give each user an item list that will be accepted with pleasure based on these information.

The rest of this paper is organized as follows. We briefly review related works in Section 2. The social empirical analyses are conducted in Section 3. In Section 4, we introduce the details of our proposed model HTPF. The experiment results and discussions are presented in Section 5. Finally, we conclude the paper and present some directions for future work in section 6.

2. Related works

With the exponential growth of information generated on consumer review websites and e-commerce websites, recommender systems are drawing more attention from both academia and industry [15–21]. For the system with explicit feedback (numerical ratings), substantial works have been done about collaborative filtering (CF) model for its accuracy and scalability during the past two decades [22–26]. To address the limitation of the traditional CF methods such as cold-start problem, substantial works has been done about social recommendation model. Here we briefly review the most related works, more details can refer to some excellent surveys and monographs [15,27].

Recent works in social recommendation integrated trust information into rating information by modeling social influence on user's rating values in different ways. Some methods assume that connected users will share similar preference [28–30]. For example, Sorec [31], TrustMF [32], PSLF [33], which jointly factorize rating matrix and trust(social) matrix by sharing a common latent user space; Soreg [28], SocialMF [34] and circle-based methods [29] extends classical PMF (probabilistic matrix factorization[35]) model by constraining user's latent preference close to the average of his trusted friends.

In some other methods [36–39], users' ratings are considered as synthetic results of their preference and social influence. For example, RSTE [36] was proposed by modeling users' rating values based on their own taste and friends' preference, where a specific parameter was used to control the effect of two parts; Xin et al. [40] notice the different roles of strong ties and weak ties in social recommendation, thus they extend RSTE to PTPMF by incorporating the distinction of strong and weak ties for improving recommendation performance; SPF [38] also extends RSTE by integrating social influence into the poisson factorization [10] of users' rating matrix; Bao et al. [39] notice the gap between users' trust and their preference-similarity. Thus, they consider social trust as a multi-faceted phenomenon and attempt to decompose trust into four aspects for better rating prediction; Guo [41] extends SVD+ [23] by further incorporating both the explicit and implicit influence of friends on users' ratings in their TrustSVD model. On the one hand, the preference of trustees(friends) has been considered to generate user's rating values. On the other hand, they constrained that the user-specific vectors decomposed from the rating matrix are the same as those decomposed from the trust(social) matrix.

However, these methods ignore the role of user's attention in social recommendation. Many psycho-social literatures [4,6] suggested the importance of user's attention and our analyses in Section 3 show that the influence of the social relations on user's ratings is smaller than their influence on the user's attention. Thus, different from existing methods, the HTPF proposed in this paper captures user's attention in the social network, which can effectively bridge the trust-preference gap and achieve better performance than state-of-the-art social recommendation methods.
3. Analyses of social influence on datasets

In this section, we conduct empirical analyses on four large real-world datasets: Epinions\(^1\), Ciao\(^2\), Flixster\(^3\) and Douban\(^4\). The four datasets contain several kinds of information. The rating information denotes the rating values that the user gave to some items. The rating values in Epinions, Ciao and Douban are integers from 1 to 5, while those in Flixster are real values from 0.5 to 5.0 with step 0.5; Also, there exist two kinds of social relations between users in these datasets: The trust relations are directed in Epinions and Ciao, while the friend relations are undirected in Flixster and Douban. We treat undirected friend relations as bi-directed trust relations in this paper since friends trust each other; Besides, the four datasets contain the information about user’s attention. Considering the phenomenon that users are willing to spend mental effort on consuming and rating some specific items, we could learn user’s attention from the information about which items the user have rated \(1,14\). In summary, Epinions, Ciao, Flixster and Douban are ideal datasets that have been used widely in recent works for analyses and experiments on social recommendation. The datasets statistics are illustrated in Table 1.

In order to show the influence of social relations on users’ attention and their preference, two empirical analyses were conducted as follows: (1) We calculate the average rating similarity and attention (behavior) similarity for each trustee-trustee pair, where rating similarity denotes the Pearson correlation coefficient \[42\] between their rating values of common items and attention (behavior) similarity denotes the average Jaccard coefficient between the sets of items that have been rated respectively by them (similarity between users’ attention behavior). In comparison, we calculate the average rating similarity and attention (behavior) similarity between arbitrary two users. The results are shown in Fig. 1(a)(b). (2) We divide pairs of users into 8 groups according to the number of their common trustees. We then calculate the average rating similarity and the average attention (behavior) similarity between every user pairs in each group. The results are shown in Fig. 1(c)(d). An important observation is concluded from these results.

Observation. Comparing with user’s rating values, user’s attention is more susceptible to social network.

As showed in Fig. 1(a)(b), socially connected users tend to pay attention to more common items than average user pairs, but their rating values do not exhibit too much more similarity than average user pairs. Meanwhile, as showed in Fig. 1(c)(d), as the number of common trustees increases, users tend to have more commonality in their attention behavior while their rating values remain mostly unaffected by those more similar social relations. In content-sharing social networks, items purchased by trustees are more likely brought to our attention and stored in episodic memory \[43\]. Thus, user’s attention behavior is heavily affected by social relations while his rating values stay relatively indifferent to the contents shared by his trustees. This observation motivates us to learn user’s attention instead of his preference from social network.

Besides, though we may wish to see ourselves as unique individuals, each of us belongs to some content-shared communities and our attention will be influenced by our friends/trustees \[2,9\]. Considering the different sensitivity of user’s attention and his preference to social relations as shown in Fig. 1, we can conclude that our attention is not coincident with our preference. On the one hand, when presented a list of recommendations, we usually focus on the items that draw our attention, and are unwilling to spend effort to learn the information about the rest. This means that the recommended items just suitting our taste may be skipped up because of our limited and selective attention \[4,5\]. On the other hand, the recommended item, which draws our attention, may not be accepted because we dislike it. To improve recommendation accuracy, we need to combine both users’ attention and their preference in social recommendation.

4. Model

In this section, we present our proposed model HTPF and its inference algorithm.

4.1. Hierarchical Trust-based Poisson Factorization (HTPF)

The key to a good social recommendation model is how it exploits trust information. Different from existing methods, we consider trust information as a helpful complementary information to capture users' attention instead of users' preference, since user's attention is important in recommendation as suggested by many psycho-social literatures and is more susceptible to social network as shown in Fig. 1. That is, the trust relations in a large degree indicate what information users pay attention to in the social network. Thus the observed trust relations were modeled as follows:

Considering the advantages of poisson factorization on sparse data \[10\], we model trust relations as poisson distribution, parameterized by the inner product of the K-dimensional vector of latent attention of user \(u\) \((g_\alpha)\) and the K-dimensional vector of latent features of trustee \(v\) \((h_\beta)\), \(G_{uv} = \text{poisson}(g_\alpha^T h_\beta)\). Where \(g_\alpha\) represents "what topics or features user \(u\) pays attention to" and \(h_\beta\) represents "the topics of the information shared by user \(v\)". In the social websites, we tend to trust or follow the users whose shared information draws our attention. Thus, we model trust relations \(G_{uv}\) based on the \(g_\alpha h_\beta\), which captures "how users pay attention to the shared information". Similarly, users’ attention behavior information and rating information can be modeled by poisson factorization too based on “how users pay attention to the items” and “how they like the items”, \(\Omega_{mv} = \text{poisson}(g_{\alpha i}^T \beta_j)\), \(R_{ui} = \text{poisson}(\theta_i^T \beta_j)\). Where \(\beta_j\) denotes the K-dimensional vector of latent attribute of item \(i\), \(\theta_i\) denotes the K-dimensional vector of latent preference of user \(u\).

Besides, we place hierarchical gamma priors on these latent variables \((g_\alpha, h_\beta, \theta_i, \beta_j)\) to control the average size of representation, which can capture the diversity of users and items: (1) users’ activity \(\gamma_u\), where some users who are active tend to pay attention to more items and trustees than others; (2) users’ rating habit \(\alpha_u\), where some users are generous and tend to give relatively high ratings than those users who are critical about their ratings; (3) influence power \(\eta_u\), where some users tend to have more fans than others; (4) items’ quality \(\sigma_i\), where the items with higher quality tend to be more popular and be rated with higher scores than others. HTPF can capture such heterogeneity across users and items.

---

1. http://www.trustlet.org/opinions

---

Table 1: Statistics of the four datasets.

<table>
<thead>
<tr>
<th>Features</th>
<th>Epinions</th>
<th>Ciao</th>
<th>Flixster</th>
<th>Douban</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>49,289</td>
<td>7,375</td>
<td>147,612</td>
<td>129,490</td>
</tr>
<tr>
<td>Number of items</td>
<td>139,738</td>
<td>106,797</td>
<td>48,794</td>
<td>58,541</td>
</tr>
<tr>
<td>Number of ratings</td>
<td>664,824</td>
<td>280,391</td>
<td>8,196,077</td>
<td>16,830,839</td>
</tr>
<tr>
<td>Density of ratings</td>
<td>0.0097%</td>
<td>0.0356%</td>
<td>0.1138%</td>
<td>0.2220%</td>
</tr>
<tr>
<td>Number of trusts</td>
<td>487,183</td>
<td>111,781</td>
<td>2,430,686</td>
<td>1,692,952</td>
</tr>
<tr>
<td>Density of trusts</td>
<td>0.0201%</td>
<td>0.0295%</td>
<td>0.0312%</td>
<td>0.0301%</td>
</tr>
<tr>
<td>Connection type</td>
<td>Directed</td>
<td>Directed</td>
<td>Undirected</td>
<td>Undirected</td>
</tr>
</tbody>
</table>
which was suggested an important factor for the recommendation model [24].

In all, our method has following generative process and the graphical model shows in Fig. 2:

1. For each user $u \in U$:
   - Sample activity: $\gamma_u \sim \text{gamma}(\alpha_a', \sigma_a')$.
   - Sample rating habit: $\alpha_u \sim \text{gamma}(\alpha_u', \sigma_u')$.
   - Sample influence power: $\sigma_u \sim \text{gamma}(\sigma_u', \sigma_u')$.
   - For each component $k(1, 2, \ldots, K)$:
     - Sample latent attention: $g_{uk} \sim \text{gamma}(a_k, \gamma_u)$.
   - Sample latent preference: $\theta_{uk} \sim \text{gamma}(a_u, \alpha_u)$.
   - Sample features of trustee: $h_{uk} \sim \text{gamma}(a_h, \eta_u)$.

2. For each item $i \in I$:
   - Sample quality: $\sigma_i \sim \text{gamma}(\sigma_i', \sigma_i')$.
   - For each component $k(1, 2, \ldots, K)$:
     - Sample attribute: $\beta_k \sim \text{gamma}(a_k, \sigma)$.

3. For each item $i \in I$ and user $u \in U$:
   - Sample attention behavior: $\Omega_{ui} \sim \text{poisson}(g_{ui}^w \beta_i)$.
   - Sample rating: $R_{ui} \sim \text{poisson}(\hat{\Omega}_{ui} \beta_i)$.

4. For each user $u \in U$ and user $v \in U$:
   - Sample trust: $G_{uv} \sim \text{poisson}(\hat{g}_{uv} \beta_v)$.

As we can see from our model, comparing with existing methods that direct fusing trust information into rating information to recommend items, our method uses attention behavior information as medium for combining trust information with rating information. Specifically, our method combines poisson factorization of attention behavior information and rating information by sharing common latent vectors of items, because of constant item attributes. At the same time, we combine poisson factorization of attention behavior information and trust information by sharing common latent vectors of users’ attention, because the users who have similar social relations tend to pay attention to similar items.

Also, different from recent works which recommend items just based on predicted ratings and overlook the importance of users’ attention, we combine users’ preference and their attention for better recommendation accuracy. For each user $u$, we rank items based on the score as follow:

$$
\text{score}_{ui} = (R_{ui})^w \Omega_{ui}^{1-w}
$$

(1)

Where $\hat{R}_{ui}$ means the predicted rating of item $i$ given by user $u$, depicting how user $u$ likes item $i$. $\Omega_{ui}$ means how user $u$ pays attention to item $i$. $w \in [0, 1]$ controls the importance of two aspects. We just employ users’ attention to recommend items when
w = 0, while we just consider users’ preference when w = 1. Once the posterior is fit, \( \hat{R}_{ui} \), \( \Omega_{ui} \) could be calculated by their posterior expected poisson parameters: \( \hat{R}_{ui} = E[\theta_i^u \beta_i] \), \( \Omega_{ui} = E[g_i^u \beta_i] \).

4.2. Approximate inference

Considering that the posterior probability is intractable, we develop an efficient approximate method to compute the posterior based on the variational inference [44]. The mean field theory drives us to partition latent variables into disjoint groups and these variables are governed by their own variational parameters [45]. Also, we add auxiliary latent variables \( Z^R_{uk} \sim \text{poisson}(\theta_i^u \beta_i^R) \), \( Z^G_{uk} \sim \text{poisson}(g_i^u \beta_i^G) \) to facilitate derivation and update [38]. Note that the sum of poisson random variables is itself a poisson with rate equal to the sum of the rates [44]. Thus, these variables can be thought of as the contribution from component \( k \) to the total observations \( R_{ui} \), \( \Omega_{ui} \), \( G_{ui} \) according to the property of poisson distribution.

\[
\sum_k Z^R_{uk} = R_{ui}, \sum_k Z^G_{uk} = \Omega_{ui}, \sum_k Z^G_{uk} = G_{ui}
\] (2)

Besides, according to the theory of variational inference, we specify the form of factored variational distribution of each variable as same as its corresponding conditional distribution [45]. The variational distributions of latent variables \( \beta, \theta, g, h, y, \alpha, \eta, \sigma \) are gamma distributions with their own parameters and \( Z^R, Z^G, Z^G \) are multinomial distributions. So, we define the variational distribution as follows:

\[
\begin{align*}
q(\beta, \theta, g, h, Z^R, Z^G, Z^G, \gamma, \alpha, \eta, \sigma) &= \prod_{i,k} q(\beta_i^u | \lambda_i^u, \lambda_i^R) \\
&\times \prod_{i,k} q(\theta_i^u | \kappa_i^u, \kappa_i^G, \kappa_i^G) q(g_i^u | \psi_i^u, \psi_i^G) \\
&\times \prod_{i,k} q(\alpha_i | \epsilon_i^u, \epsilon_i^G) \prod_{i,k} q(\eta_i | \epsilon_i^u, \epsilon_i^G) \prod_{i,k} q(\sigma_i | \epsilon_i^u, \epsilon_i^G) \\
&\times \prod_{i,k} q(Z^R_{ik} | \phi_i^u, \phi_i^G) q(Z^G_{ik} | \phi_i^G) \prod_{i,k} q(Z^G_{ik} | \phi_i^G)
\end{align*}
\] (3)

where we use superscript ‘s’ and ‘r’ represent the shape parameter and the rate parameter respectively. Note that minimizing the KL divergence between the variational distribution and the true posterior is equivalent to optimizing an evidence lower bound (ELBO) \( L(q) \), a bound on the log likelihood of the observations [45]. The ELBO is

\[
L(q) = E_q[\ln p(R, G, \Omega, \Theta)] - E_q[\ln q(\Theta)]
\]

\[
= \sum_{u \in U} \sum_{i,l} \left( E_q[p(R_{ui} | \theta_i^u, \beta_i)] + E_q[p(G_{ui} | g_i^u, \beta_i)] \right)
\]

\[
+ \sum_{u \in U} \sum_{i,l} E_q[p(G_{ui} | g_i^u, \beta_i)] + \sum_{i,l,k=1} E_q[p(\beta_{ik} | a_{\beta}, \sigma_{\beta})]
\]

\[
+ \sum_{u \in U} \sum_{i,l=1} E_q[p(\theta_{ik} | a_{\theta}, \alpha_{\theta}) + E_q[p(g_{ik} | a_{g}, \gamma_{g})]]
\]

\[
+ \sum_{u \in U} \sum_{i=1} E_q[p(h_{ik} | a_h, \eta_{h}) + \sum_{i=1} E_q[p(\alpha_i | a_{\alpha})]
\]

\[
+ \sum_{u \in U} \sum_{i=1} E_q[p(\alpha_i | a_{\alpha})] + E_q[p(\gamma_i | a_{\gamma})] + E_q[p(\eta_i | a_{\eta})]
\]

\[
- E_q[\ln q(\Theta)]
\] (4)

where \( \Theta = [\beta, \theta, g, h, Z^R, Z^G, Z^R, y, \alpha, \eta, \sigma] \) denotes latent variables. We can use the coordinate ascent method to optimize variational parameters in turns by optimizing the lower bound \( L(q) \). The variational inference of HTPF is illustrated in **Algorithm 1**.

**Algorithm 1** The variational inference of HTPF.

1: initialize \( \lambda, \kappa, \phi, \tau, b, c, d, e \) randomly;
2: while not converge do
3: for each observations \( \Omega_{ui} > 0, R_{ui} > 0, G_{ui} > 0 \) do
4: update variational parameters of \( Z^R_{uk}, Z^G_{uk}, Z^G_{uk} \) (Equation (5)-(7))
5: end for
6: for each user do
7: update variational parameters of \( \theta_i, g_i, h_i, \alpha_i, \eta_i, \sigma_i \) (Equation (8)-(15))
8: end for
9: for each item do
10: update variational parameters of \( \beta_i, \sigma_i \) (Equation (16)-(19))
11: end for
12: end while

As we can see from **Algorithm 1**, in the beginning, we initialize variables to their priors with small offsets, where offsets can be sampled from uniform distribution over the interval [-0.1,0.1]. Then, we calculate the gradient of \( L(q) \) on each parameter and iteratively optimize each parameter while holding the others fixed. For the parameters of observations \( \Omega_{ui}, R_{ui}, G_{ui} \), we just consider those non-zero observations. Since when observations are zero, these variational random variables are not. The posterior distributions of \( Z^R_{uk}, Z^G_{uk}, Z^G_{uk} \) will place all their mass on the zero vector. We calculate the optimal variational parameters of those non-zero observations as follows:

\[
\phi_i^u \propto \exp(\Psi(\kappa_i^u) - \log \kappa_i^u + \Psi(\lambda_i^u) - \log \lambda_i^u)
\] (5)

\[
\phi_i^G \propto \exp(\Psi(\phi_i^u) - \log \phi_i^u + \Psi(\lambda_i^u) - \log \lambda_i^u)
\] (6)

\[
\phi_i^G \propto \exp(\Psi(\phi_i^G) - \log \phi_i^G + \Psi(\tau_i^G) - \log \tau_i^G)
\] (7)

where \( \Psi() \) is the digamma function (the first derivative of the log gamma function).

Then, for each user we calculate the optimal variational parameters of \( \theta_i, g_i, h_i, \alpha_i, \eta_i, \sigma_i \) as follows:

\[
\kappa_i^u = a_0 + \sum_{i=1} \omega_i u R_{uk} \kappa_i^u = b_i^u + \sum_{i=1} \lambda_i^u \tau_i^u
\] (8)

\[
\phi_i^G = c_i^u + \sum_{i=1} \omega_i u \phi_i^G \phi_i^G = d_i^u + \sum_{i=1} \tau_i^u \phi_i^G
\] (9)

\[
\phi_i^G = \frac{d_i^u}{d_i^u + \sum_{i=1} \tau_i^u} + \frac{\lambda_i^u}{\lambda_i^u + \sum_{i=1} \tau_i^u}
\] (10)
\[ \hat{b}_u = \hat{K}_u + \hat{a}_u; \hat{d}_u = \hat{K}_u + \hat{a}_u \]  

\[ \hat{b}_u = \sum_{k=1}^{K} \hat{k}_{uk} \hat{a}_u; \hat{d}_u = \sum_{k=1}^{K} \hat{q}_{uk} \hat{a}_u \]  

\[ \tau_{ik} = a_i + \sum_{u \in U} \phi_{uik}; \tau_{ik} = \frac{e_i}{\tau_{ik}} + \sum_{u \in U} \phi_{uik} \]  

\[ \phi_{uik} = \sum_{k=1}^{K} \frac{\tau_{ik}}{\phi_{uik}} + a_i \]  

Similarly, for each item we calculate the optimal variational parameters of \( \beta_i, \sigma_i \) as follows:

\[ \lambda_{ik} = a_i + \sum_{u \in U} \phi_{uik}; \lambda_{ik} = \frac{\lambda_{ik}}{\phi_{uik}} + \sum_{u \in U} \phi_{uik} \]  

\[ \lambda_{ik} = \frac{\lambda_{ik}}{\phi_{uik}} + \sum_{u \in U} \phi_{uik} \]  

\[ \lambda_{ik} = \sum_{k=1}^{K} \frac{\lambda_{ik}}{\phi_{uik}} + a_i \]  

Note that the optimal shape parameters \( \lambda, \sigma \) are independent on other variational parameters. Thus, we can get those optimal values at once without iteration (Eq. (11), (14), (18)). Finally, we terminate iteration by observing the change in the average predicted log likelihood of the validation set.

4.2.1. Complexity analysis

We can find that our algorithm is very efficient on sparse data. When updating variational parameters \( \phi_{uik}, \phi_{uik}, \phi_{uik} \), we just consider those non-zero observations, since when observations are zero these variational variables are not random. The time to update these variational parameters is \( O(K(|R| + |G|)) \), where \(|R|, |G| \) means the number of non-zero ratings and social relations, \( K \) means the length of latent vector. When updating \( \kappa, \lambda, \sigma \), we just sum over non-zero observations with complexity \( O(K(|R| + |G|)) \). When updating \( \tau, \lambda, \sigma \), we can preprocess the sum of \( \frac{\lambda_{ik}}{\phi_{uik}} + \frac{\lambda_{ik}}{\phi_{uik}} \) over all items or users to speed up our algorithm. Thus, the complexity of our algorithm is \( O(K(|R| + |G|)) \). Generally, the overall computational time is linear with respect to the number of observed non-zero entries and our model has potential to scale up to large datasets.

5. Experiments and results

In this section, we evaluate our algorithm on real-world datasets. We start with the description of four datasets.

5.1. Datasets and evaluation metrics

The four datasets, Epinions, Ciao, Flixster and Douban, presented in Section 3 are used in our experiment. Specially, we double the rating values in Flixster to get integral values. A 5-fold cross-validation for learning and testing is used in our experiment where the datasets are divided into 5 folds: four folds for training and the remaining one for testing in each iteration. Also, we sample 1% data of training set as validation set to measure convergence and tune parameters. The final result is averaged over 5 iterations where all folds are tested. To quantitatively evaluate the experimental results, we used two metrics:

**Normalized precision@M (Pre@M):** We recommend each user M items and calculate the fraction of relevant items in the test set as follow. As recent works [10,38], highly rated items in the test set will be considered as relevant items for each user (larger than 3 in a 5-star system).

\[ \frac{1}{|U|} \sum_{u \in U} \frac{hit_u}{\min(M, t_u)} \]  

where \( hit_u \) denotes the number of relevant items in the user’s top-M recommendations, \( t_u \) denotes the number of relevant items of the user u. We adjust denominator to min(\( M, t_u \)), because traditional measurement of precision will artificially deflate this measurement for the users who have fewer than M relevant items in the test set [10].

**Normalized Discounted Cumulative Gain (NDCG):** This is widely used in information retrieval and it measures the quality of ranking through discounted importance based on positions. In recommender systems, NDCG is computed as follows:

\[ \text{NDCG} = \frac{1}{|U|} \sum_{u \in U} \frac{\text{DCG}_u}{\text{IDCG}_u} \]  

where \( \text{DCG}_u \) is defined as follow and \( \text{IDCG}_u \) is the ideal value of \( \text{DCG}_u \) coming from the best ranking.

\[ \text{DCG}_u = \sum_{i \in \text{Re}(u)} \frac{1}{\log_2(\text{ran}_u + 1)} \]  

where \( \text{ran}_u \) represents the rank of the item \( i \) in the recommended list of the user \( u \) and \( \text{Re}(u) \) denotes the set containing all relevant items of user \( u \).

5.2. Compared algorithms

To demonstrate the effectiveness of our HTPF algorithm, we compared with some methods including:

- The baseline POP: In POP, we rank items by their universal popularity.
- HFP [10]: The state-of-the-art method based on rating information only. HFP conducts poisson factorization on rating matrix to get user’s latent preference for recommendation. Note that HFP is a special case of our model HTPF, where user’s attention is left out in the model.
- SPF [38]: TrustSVD [41]: The state-of-the-art social recommendation methods mentioned in Section 2, which model social influence on the rating values.
- We designed another method HTPF-a as comparison. HTPF-a, a simple version of HTPF, generates recommendations just based on users’ attention by integrating poisson factorization of trust information with attention attentive information, while rating information is left out in HTPF-a. Note that HTPF-a is one of our contributions.

The optimal experimental settings for all methods are determined either by our experiments or suggested by previous works. Specifically, the number of latent components \( K \) are set to 30 across all datasets for our methods (HTPF-a, HTPF), while the best values of \( \omega \) on different datasets are showed in Table 3. For hyperparameters about prior distributions, we set \( \beta \) to 3 and others to 0.3.
Table 2
The recommended performance on four datasets. The bold terms denote the best performance among all methods.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Ciao</th>
<th>Epinions</th>
<th>Flixter</th>
<th>Douban</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP</td>
<td>3.83</td>
<td>0.1551</td>
<td>3.06</td>
<td>0.1335</td>
</tr>
<tr>
<td>TrustSVD</td>
<td>2.37</td>
<td>0.1336</td>
<td>2.23</td>
<td>0.1194</td>
</tr>
<tr>
<td>HPF</td>
<td>4.68</td>
<td>0.1600</td>
<td>4.18</td>
<td>0.1446</td>
</tr>
<tr>
<td>SPF</td>
<td>4.69</td>
<td>0.1629</td>
<td>4.27</td>
<td>0.1449</td>
</tr>
<tr>
<td>HTPF-a</td>
<td>4.86</td>
<td>0.1650</td>
<td>4.31</td>
<td>0.1480</td>
</tr>
<tr>
<td>HTPF</td>
<td>5.29</td>
<td>0.1692</td>
<td>4.86</td>
<td>0.1520</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. The performance in terms of normalized precision@20 with varying values of parameter $w$, where $w$ balances the contribution of users’ preference and attention in recommendation.

5.3. Experimental results and analyses

The experimental results of all methods on four datasets are presented in Table 2. HTPF obviously outperforms all the comparison methods on all the four data sets. The results confirm that by modeling users’ attention HTPF effectively improves recommendation performance. One reason for this improvement is that users’ attention plays an important role in users’ selection on recommendation. This can be seen from the good performance of HTPF-a, which only considers users’ attention in social recommendation. By combining users’ attention and preference in social recommendation, HTPF is able to achieve even better performance than HPF and HTPF-a. Another important reason for this improvement is that comparing with users’ preference their attention is more susceptible to social network. Trust information can act as useful complementary information to deduce user attention but it contains limited information about users’ preference. By modeling users’ attention in social recommendation, HTPF can effectively bridge the trust-preference gap [39] and achieve better performance than state-of-the-art social recommendation methods.

Table 3
Optimal $w$ values on different datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Ciao</th>
<th>Epinions</th>
<th>Flixter</th>
<th>Douban</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal $w$</td>
<td>0.2</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>Density of the network</td>
<td>0.2055%</td>
<td>0.0201%</td>
<td>0.0112%</td>
<td>0.0101%</td>
</tr>
</tbody>
</table>

5.3.1. Impact of parameter $w$

Another experiment is conducted to investigate how parameter $w$ affects the performance of our model, where $w$ balances the contribution of users’ preference and attention in recommendation. The results in terms of precision@20 with varying $w$ are presented in Fig. 3. Also, we present the performance of HPF and HTPF-a in comparison. As we can see, as $w$ becomes larger, the performance becomes better first. This is because when users’ preference plays a more important role in rank, more favorite items will be found. However, when $w$ surpasses a threshold, the performance becomes worse with further increase of $w$. As users’ attention becomes unimportant in rank, the recommended items may be
refused because these items won’t draw users’ attention and may be skipped up. We find HTPF has worst result when w is fixed at 0 or 1, which confirms with the idea that combining users’ attention and preference performs better than considering only one aspect.

The recommendation generated by HTPF is a balanced result between users’ attention and their preference. However, as shown in Fig. 3, even when w = 0 (meaning we recommend items based only on users’ attention), HTPF still achieves better performance than HTPF-a. The reason is that HTPF has already incorporated trust, attention behavior and rating information in its model training and can capture more accurate features of users and items than HTPF-a. Similar result can be observed when w = 1. When HTPF still achieves better performance than HPF.

5.3.2. Optimal w values on different datasets

It will be interesting to explore how the density of network connections will affect the balance parameter w. Table 3 lists the optimal w values in four different networks with different density. As can be seen, more densely connected networks correspond to smaller optimal w values, meaning that user attention plays a more important role. For instance, Ciao has much denser social network than Epinions. The users in Ciao have a higher degree of exposure to social network and are more strongly affected by their trustees/friends. To achieve best recommendation accuracy, HTPF need rely more on users’ attention in Ciao, since users’ attention is susceptible to social network.

6. Conclusions

In this paper, we propose a new probabilistic model HTPF that explicitly considers both users’ attention and preference in social recommendation. Many psycho-social literatures suggest the importance of users’ attention in recommendation and our observations in Section 3 show that the influence of trust relations dwells more on users’ attention than on their preference. Thus, we propose the model HTPF with a generative process where we use social network as complementary information to deduce user’s attention instead of their preference. We also design an efficient stochastic variational inference method for our model that can scale up to large data sets. Our comprehensive experimental results on four real-world datasets clearly demonstrate the effectiveness of our proposed method and its superiority over existing social recommendation methods.

One interesting direction for future work is to further exploit the relations and differences between user’s attention and preference based on more supplementary information, such as content data and browsing data. Also, we can consider both users’ attention and preference to deal with link prediction problem [46], since attention is more susceptible to the social network.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant no: U1866602) and by a Discovery Grant from the National Science and Engineering Research Council of Canada. It is also partially supported by ByteDance.

References

[9] G. Pallà, I. Derénya, I. Farkas, T. Vicsek, in: Uncovering the overlapping commu-

[10] P. Gopalani, J.M. Hofman, D.M. Blei, Scalable recommendation with hierarchi-

cal poisson factorization, in: Proceedings of the Thirty-First Conference on Un-
[12] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering rec-
n
[15] F. Ricci, L. Rokach, B. Shapira, Recommender systems: introduction and chal-

[17] Y. Zhu, J. Zhu, J. Hou, Y. Li, B. Wang, Z. Guan, D. Cai, A brand-level rank-


[21] Y. Zhu, Z. Guan, S. Tan, H. Liu, D. Cai, X. He, Heterogeneous hypergraph em-

[22] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender sys-

[23] V. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Confer-
ence on Knowledge Discovery and Data Mining, ACM, 2008, pp. 426–434.
[24] V. Koren, R. Bell, C. Volinsky, et al., Matrix factorization techniques for recom-

[26] F. Ricci, L. Rokach, B. Shapira, Introduction to Recommender Systems Hand-

[29] X. Yang, H. Steel, Y. Liu, Circle-based recommendation in online social net-

[31] H. Ma, H. Yang, M.R. Lyu, I. King, SOREC: social recommendation using prob-

[32] B. Yang, Y. Lei, D. Liu, Y. Liu, Social collaborative filtering by trust, in: Pro-

ceedings of the Twenty-Third International Joint Conference on Artificial In-

[33] Y. Shen, R. Jin, Learning personal social latent factor model for social recom-


Jiawei Chen received his B.S. at University of electronic science and technology of China in 2014. He is currently a Ph.D. candidate in the college of computer science at Zhejiang University. His main research topics are recommender systems, graphical model and social networks.

Can Wang received the Ph.D. degree and M.S. degree in computer science and B.S. degree in economics from Zhejiang University, in 2009, 2003 and 1995, respectively. His research interests include data mining, machine learning and information retrieval.

Qihao Shi received his B.S. at Nanjing Normal University of China in 2014. He is currently a Ph.D. candidate in the college of computer science at Zhejiang University. His main research topics are social and information networks, algorithmic game theory and Internet economics.

Yan Feng received the Ph.D. degree in computer application from Zhejiang University in 2004. She is currently an associate professor in the College of Computer Science at Zhejiang University, China. Her research interests include database, data mining etc.

Chun Chen is a professor in the College of Computer Science, Zhejiang University. His research interests include data mining, computer vision, computer graphics and embedded technology.