This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A probabilistic model for user interest propagation in recommender systems

SAMUEL MENSAH[1], CHUNMING HU[1], XUE LI[1],XUDONG LIU[1], AND RICHONG ZHANG.[1], (Member, IEEE)

[1]BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China.

Corresponding author: Chunming Hu (e-mail: hucm@act.buaa.edu.cn).

This work was supported in part by the China 973 Program under Grant 2015CB358700, in part by the National Natural Science Foundation of China under Grant 61772059 and Grant 61421003, in part by the Beijing Advanced Innovation Center for Big Data and Brain Computing, and in part by the State Key Laboratory of Software Development Environment.

ABSTRACT User interests modeling has been exploited as a critical component to improve the predictive performance of recommender systems. However, with the absence of explicit information to model user interests, most approaches to recommender systems exploit users activities (user generated contents or user ratings) to inference the interest of users. In reality, the relationship among users also serves as a rich source of information of shared interest. To this end, we propose a framework which avoids the sole dependence of user activities to infer user interests and allows the exploitation of the direct relationship between users to propagate user interests to improve system's performance. In this paper, we advocate a novel modeling framework. We construct a probabilistic user interests model and propose a user interests propagation algorithm (UIP), which applies a factor graph based approach to estimate the distribution of the interests of users. Moreover, we incorporate our UIP algorithm with conventional matrix factorization (MF) for recommender systems. Experimental results demonstrate that our proposed approach outperforms previous methods used for recommender systems.

INDEX TERMS Propagation, Recommender System, Sum-Product Algorithm, User Interest Modelling

I. INTRODUCTION

SOCIAL networks is now the most popular service platform consumed by users and has become a crucial part of our life. Consumers on these platforms are spoiled with choices. For instance, Douban[1] offers a huge selection of movies, books and music for customer satisfaction. Sifting through all the available products to isolate and recommend what is relevant to a customer is the main highlight to ensure customer satisfaction and loyalty in these networks. Thus, a user's interest is key to determine relevant products for users.

The vast amount of content on social networks such as rating scores on items on Douban, arguably, has influence when modeling the interest of a user [26]. As a simple example, it can be argued that the interest of a user can be determined by how frequent she rates movies of a particular genre. This freely available content, provides a great opportunity for the research community to model user interests, which are the foundation of online advertising [24], [37], personalized recommendation [7], [8], [25], [30], including travel recommendations [28], [39]. Based on the intuition that contents generated from users can be employed when modeling user interests, some earlier works have been proposed [22], [32]. These works simply extract keywords from user contents to represent user interests. Besides that, some studies have also modeled user interests at a semantic level, among of which some algorithms are based on matrix factorization [20], [21], [34], others employ Latent Dirichlet Allocation (LDA) topic model [16], [18], [33], while others employ both LDA and matrix factorization [38].

While many of the applications mentioned above essentially employ user contents to enhance the modeling of user interests, they neglect the social relationship of users which occur naturally on online social network platforms. It can be argued that the topological structure of a social network contain essential information which can be harnessed for user interest inference. Consider the social network Twitter for an example, a user may have never posted pop music related tweets, but considering the fact that she follows several pop artists such as Madonna on Twitter, we can make the inference that she is actually interested in pop music. This

[1]https://www.douban.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

intuition suggest that relationship between users in a network has an influence on user interest, and we can infer user interests from these relationships.

A common approach to incorporate social networks for user interest modelling in recommender systems (RS) is via the propagation of interest to a user from similar neighbors of the user [1], [12], [21], [27], [36]. One key problem of these methods is the prediction of user interest when there are few neighbors of the user, or when there is sparsity of the interests of the user. Thus, there is an element of uncertainty as to how the available interests of a user best describe his interest. Hence for the user interest modelling in RS, it is of importance to explore statistical methods to make inference on users interest.

In the present work, we interpret the user interest prediction problem as a propagation problem and formulate a factorized probability model expressed readily in the framework of a factor graph [15]. Our proposed model has two key variables which are used: observed interests and true interests. The observed interests refers to the interests that are expressed in social online networks by users via user generated contents, and the true interests refers to the interests that can be inferred from both the observed interests and the relationship between users. Thus, we wish to predict the true interests of a user by leveraging both user generated content and relationships of the network. Accordingly, we propose a probabilistic user interests model and develop a user interests propagation algorithm (UIP), which utilizes a sum-product algorithm (a belief propagation algorithm) that operates directly on the induced factor graph of the social network [15], with the aim to estimate the distribution of true interests for each user. The algorithm is naturally designed to propagate the interests of users along the relationships.

To demonstrate the effectiveness of UIP, we integrate UIP with a basic matrix factorization (MF) algorithm to predict unknown ratings of user interests as an RS task. Experimental results on public datasets demonstrates that this modified MF-based recommendation algorithm achieves better performance than existing MF-based methods including, the basic matrix factorization (MF), probabilistic matrix factorization (PMF) [23], matrix factorization with social regularization (MF-SR) [21], social matrix factorization (SocialMF) [12], and NeuralMF [11].

II. RELATED WORKS

Accurately predicting the interest of users is crucial for recommender systems (RS). Early works focus on exploiting natural language processing (NLP) tools to process the user generated contents and then identify user interests, among which are the first kind of algorithms which operate at the term level [22], [32]. Specifically, [32] use TF-IDF to extract keywords which are assumed to be informative to express a user's interest, while [22] extend the works of [32] by employing an entity extraction algorithm and knowledge base to link keywords to topics. On one hand, traditional NLP tools perform badly on short and noisy texts, such as tweets on Twitter. On the other hand, term level interests can be very unique to users and it is not easy to judge the similarity between users by several keywords. Moreover, if one user does not express their interests in the text content, their interest will not be discovered at all. Several of these studies are based on matrix factorization [34], where the rating matrix is factorized into factors for users and items, allowing unknown ratings to be predicted. Although the semantic level algorithms are superior to term level algorithms, they only take account of user generated contents, neglecting relationships between users to improve system performance.

In recommender systems, some existing works focus on using side information, such as the relationships in social networks, as an add-on to the existing user-generated contents to enhance user interest (represented as latent features) predictions. Among these works, SoRec [20] proposed a social regularization method which considers the constraints on social relationships to model users features. RSTE [9] proposed social trust ensembles in a factor analysis framework, which linearly combines a basic MF model and a trust-based neighborhood model to learn both user and item features jointly. SocialMF [12] employs a trust propagation mechanism over the neighbors of a user in a social network to model the user's interest. [35] explore both category-specific social trust circles and the social network to improve RS. [21] adds a social relationship regularization term to a loss function of MF, based on the assumption that there is a similarity between users interests and social relationships. TrustMF [2] and RoRec [36] both consider the influence of trusters and trustees in social networks for RS. [27] takes advantage of several factors in social circles such as the interest of the user, relationships between users of similar interest and the influence between users to model users features. Other works which consider social networks for RS include [3], [8].

Majority of these works are based on MF, and have shown that MF-based models regularized by social relationships can enhance user features for RS. However, it is worth noting that these MF-based models learn both user and item features jointly, which may result to the risk of overgeneralizing on user features. Moreover, these methods [2], [12], [21], [35] model two-way linear interactions between user and item features for RS which may not capture the complex interactions between users and items. To mitigate the latter problem, recent methods have considered to use neural networks to learn higher-order interactions between user and item features [10], [11], [19], [31]. For example, NeuralMF [11], leverages a multi-layer perceptron to generalize MF, allowing it to learn higher-order two-way interactions between user and item features. Based on this work, other models have been proposed for cross-domain recommendations [19], [31], while [10] explore the viability of convolutional neural networks to improve the performance of NeuralMF. Some recent works have also taken advantage of social relationships by imposing a graph neural network on the social graph for RS [4], [29]. However, it is worth noting that applying non-linear models to large and sparse datasets may face scalability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

issues. Moreover, the sparsity of user-item interactions may be as a result of several factors, e.g. users may either rate items they like or dislike, which may result in weak parameter estimation of these models.

In this paper, we follow majority of recent works which incorporate the social network to improve user interest modeling [2], [12], [27]. Here, we cast the problem as an inference problem, and formulate user interest in a probabilistic framework, where we use a declarative approach to solve the problem. Here, we propagate interest from neighbors to users by means of a sum-product algorithm imposed on an induced factor graph of the social network, thereby modeling the true interests of users. Thus, we propose the user interest propagation (UIP) algorithm for this purpose. We apply this algorithm for recommender system, showing that our approach can independently enhance the user features modelled by an MF-based method, solving the generalization issue of MF-based methods. Instead of leveraging parameter prone models such as NeuralMF [11], we consider to integrate a basic MF with the UIP algorithm (MF-UIP) for rating predictions in RS. Interestingly, we show that MF-UIP outperforms NeuralMF. To the best of our knowledge, this is the first study to approach the problem in this way.

This paper is an extension of our paper titled "User interest propagation and its application in recommender system" which appeared in IEEE ICTAI'17. In this extension, we provide a detailed description of our model, showing how messages are computed and propagated on the factor graph, and also provide extensive experiments including interest recovery experiments, sparsity experiments, and the impact of parameters. We compare our model to recent proposed methods for RS, showing the effectiveness of our approach.

III. USER INTERESTS MODEL AND INTERESTS PROPAGATION ALGORITHM

We begin by introducing the probabilistic framework for the user interests model, which integrates both user generated contents and relationship between users in the social network. Next, we present the interests propagation algorithm which is based on the sum-product algorithm of an induced factor graph.

A. PROBLEM DEFINITION

Given a social network $G = (N, E)$, where N and E are the set of users and following relationships respectively. We denote $u_i \in N$ as a user, and $e_{ij} \in E$ as a following relationship that connects u_i to u_j, which means that user u_i is a follower/truster of u_j in G.

One can employ existing algorithms such as LDA topic model, or simply count the number of items of interest of a user to extract an interest vector for the user. Here, we define $\overrightarrow{\mu_X}_i$ = [$\overrightarrow{\mu_X}_i(1)$, $\overrightarrow{\mu_X}_i(2)$, $\cdots$, $\overrightarrow{\mu_X}_i(k)$] as the k-dimensional interest vector of u_i. We model the observed interests of u_i as a random variable $\overrightarrow{X_i}$ following a k-dimensional normal distribution, i.e. $\overrightarrow{X_i} \sim \mathcal{N}(\overrightarrow{\mu_X}_i, {\sigma_X}_i^2)$.

As mentioned in the introduction, the observed interests of user u_i is solely based on the content generated by u_i, and neglects information between the connectivity of users in the network. Intrinsically, a user might not express all his interests through user generated contents, we therefore assume that each user u_i has true interests expressed implicitly in the observed interests of all users and the connectivity between users in the social network. In this paper, the true interests of u_i is modeled as a random variable $\overrightarrow{Y_i}$ following a k-dimensional normal distribution, i.e. $\overrightarrow{Y_i} \sim \mathcal{N}(\overrightarrow{\mu_Y}_i, {\sigma_Y}_i^2)$.

In fact, there is a mutual relationship between an observed interest and the true interest of a user u_i. To an extent, the true interests is the intrinsic factor of the observed interests, and the observed interests reflect the true interests of user u_i. We therefore formulate the observed interests of u_i as the true interests plus $\overrightarrow{W_i}$, where $\overrightarrow{W_i}$ denotes a disturbance variable following a k-dimensional normal distribution with a zero mean . Thus, $\overrightarrow{X_i} = \overrightarrow{Y_i} + \overrightarrow{W_i}$. Based on this assumption, we model the conditional distribution of $\overrightarrow{Y_i}$ when the observed interests is known. This can be expressed as $f(\overrightarrow{Y_i}|\overrightarrow{X_i}) \sim \mathcal{N}(\overrightarrow{\mu_X}_i, {\sigma_W}_i^2)$. Here, we set the expectation of the disturbance to zero because we assume that the observed interests and true interests are similar, and there is no empirical relationship between them.

We now focus on how the relationship between users come into play in our framework. Taking account of the homogeneity of social networks [17], when a typical following relationship which indicates that a user u_i follows a user u_j, we expect the interests of u_i to be close to the interests of u_j. For that reason, we can assume the disturbance of the true interests of u_j with a disturbance variable $\overrightarrow{Z_{i\to j}}$ will result in the true interests of u_i. By respectively denoting $\overrightarrow{Y_i}, \overrightarrow{Y_j}$ as the true interests of u_i, u_j, we can express this assumption as $\overrightarrow{Y_i} = \overrightarrow{Y_j} + \overrightarrow{Z_{i\to j}}$, where $\overrightarrow{Z_{i\to j}}$ follows a k-dimensional normal distribution with zero mean, i.e. $\overrightarrow{Z_{i\to j}} \sim \mathcal{N}(\overrightarrow{0}, \sigma^2_{Z_{i\to j}})$. Based on this assumption, the conditional distribution of $\overrightarrow{Y_i}$ given $\overrightarrow{Y_j}$ is given by $f(\overrightarrow{Y_i}|\overrightarrow{Y_j}) \sim \mathcal{N}(\overrightarrow{\mu_Y}_j, \sigma^2_{Z_{i\to j}})$.

In brief, we have defined the distribution of observed interests and the true interests, how these variables are related, and the influence between users. The model is summarized as follows:

$$\begin{cases} \overrightarrow{X_i} \sim \mathcal{N}(\overrightarrow{\mu_X}_i, {\sigma_X}_i^2) \\ \overrightarrow{Y_i} \sim \mathcal{N}(\overrightarrow{\mu_Y}_i, {\sigma_Y}_i^2) \\ \overrightarrow{W_i} \sim \mathcal{N}(\overrightarrow{0}, {\sigma_W}_i^2) \\ \overrightarrow{Z_{i\to j}} \sim \mathcal{N}(\overrightarrow{0}, \sigma^2_{Z_{i\to j}}) \\ \overrightarrow{X_i} = \overrightarrow{Y_i} + \overrightarrow{W_i} \\ \overrightarrow{Y_i} = \overrightarrow{Y_j} + \overrightarrow{Z_{i\to j}} \end{cases} \tag{1}$$

Besides, we have two conditional distributions to be estimated:

$$\begin{cases} f(\overrightarrow{Y_i}|\overrightarrow{X_i}) \sim \mathcal{N}(\overrightarrow{\mu_X}_i, {\sigma_W}_i^2) \\ f(\overrightarrow{Y_i}|\overrightarrow{Y_j}) \sim \mathcal{N}(\overrightarrow{\mu_Y}_j, \sigma^2_{Z_{i\to j}}) \end{cases} \tag{2}$$

In this framework, we derive the observed interests of a user from the generated content, and capture the relationship

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

information between users with the equation $\overrightarrow{Y_i} = \overrightarrow{Y_j} + \overrightarrow{Z_{i\to j}}$. In summary, this modeling perspective takes account of the user generated content and relationships between users, leading to a better performance in user interest prediction.

B. USER INTERESTS PROPAGATION (UIP) ALGORITHM

1) Social network factor graph

We present a probabilistic formulation of the user interest prediction problem as a conditional distribution of the true interests of all users given the observed interests of all users, along with the relationship between users.

$$f(\overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n} | \overrightarrow{X_1}, \ldots, \overrightarrow{X_n}, E) \tag{3}$$

For clarity of exposition, we can simplify the expression $f(\overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n} | \overrightarrow{X_1}, \ldots, \overrightarrow{X_n}, E)$ as $f(Y|X,E)$. We aim to estimate the distribution of each $\overrightarrow{Y_i}$, which is the marginal distribution $f_i(\overrightarrow{Y_i}|X,E)$ of the global distribution $f(Y|X,E)$. In reality, social networks are complex and large in scale, this makes it difficult to directly calculate the marginal distribution from the global distribution. We therefore employ a sum-product algorithm on an induced factor graph to tackle this problem in an efficient way.

A factor graph is a bipartite graph representing the factorization of a global probability function, which efficiently enables message passing algorithms such as the sum-product algorithm to estimate the marginal functions. Consider the factorized function $g(x_1, \ldots, x_n) = \prod f_j(X_j)$ as an example, where X_j is the independent variable set of f_j, we can construct a factor graph as follows:

- For each independent variable x_i, create a variable node x_i.
- For each local function f_j, create a factor node f_j.
- For each pair of x_i and f_j, create an edge between them if x_i is in the independent variable set of f_j.

Since the factor graph is a bipartite graph, edges exist only between variable nodes and factor nodes. The sum-product algorithm is imposed on the factor graph to pass messages along edges, thus, messages are only passed from variable nodes to factor nodes and vice versa. The messages are computed as follows.

- Message from a variable node to a factor node:

$$\mu_{x\to f}(x) = \prod_{h\in n(x)\backslash\{f\}} \mu_{h\to x}(x) \tag{4}$$

- Message from a factor node to a variable node:

$$\mu_{f\to x}(x) = \sum_{\sim x}(f(X) \prod_{y\in n(f)\backslash\{x\}} \mu_{y\to f}(y)) \tag{5}$$

where $n(v)$ is the set of neighbors of a given node v in the factor graph, and $X = n(f)$ is the set of arguments of the function f. Then the marginal function of x_i is

$$g_i(x_i) = \prod_{h\in n(x_i)} \mu_{h\to x_i}(x_i) \tag{6}$$

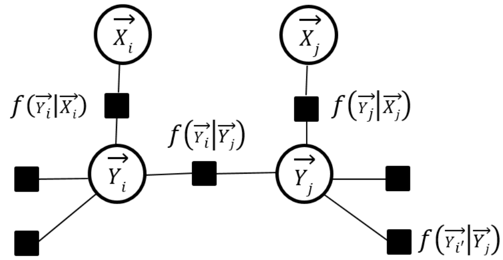

FIGURE 1: A local structure of the factor graph for a social network.

We refer the reader to the standard source [15] for further background on factor graphs and the sum-product algorithm.

For our social network, we assume that the true interests of users are conditionally independent of each other given their observed interests and the edges. Therefore we factorize the global distribution $f(Y|X,E)$ as

$$\prod_{1\le i\le n} f(\overrightarrow{Y_i}|\overrightarrow{X_i}) \prod_{e_{ij}\in E} f(\overrightarrow{Y_i}|\overrightarrow{Y_j}), \tag{7}$$

allowing the construction of a factor graph and enabling a sum-product algorithm to estimate the marginal functions.

To this end, we can construct the factor graph of the social network using the following steps:

- For each u_i, create a variable node $\overrightarrow{X_i}$ to represent the observed interests of u_i.
- For each u_i, create a variable node $\overrightarrow{Y_i}$ to represent the true interests of u_i.
- For each u_i, create a factor node $f(\overrightarrow{Y_i}|\overrightarrow{X_i})$ to represent the conditional distribution of $\overrightarrow{Y_i}$ given $\overrightarrow{X_i}$. The corresponding function is $f(\overrightarrow{Y_i}|\overrightarrow{X_i}) \sim \mathcal{N}(\overrightarrow{\mu_{X_i}}, {\sigma_W}_i^2)$.
- For each following relationship from u_i to u_j, create a factor node $f(\overrightarrow{Y_i}|\overrightarrow{Y_j})$ to represent the conditional distribution of $\overrightarrow{Y_i}$ given $\overrightarrow{Y_j}$. The corresponding function is $f(\overrightarrow{Y_i}|\overrightarrow{Y_j}) \sim \mathcal{N}(\overrightarrow{\mu_{Y_j}}, \sigma^2_{Z_{i\to j}})$.
- The factor node $f(\overrightarrow{Y_i}|\overrightarrow{X_i})$ is connected to the variable nodes $\overrightarrow{X_i}$ and $\overrightarrow{Y_i}$.
- The factor node $f(\overrightarrow{Y_i}|\overrightarrow{Y_j})$ is connected to the variable nodes $\overrightarrow{Y_i}$ and $\overrightarrow{Y_j}$.

We present in Fig. 1 a local structure of the factor graph for a social network consisting of two users u_i and u_j. Recall that the user u_i is a follower/truster of u_j.

In our model, the parameters ${\sigma_W}_i^2$ and $\sigma^2_{Z_{i\to j}}$ are optional, we set these parameters to $\overrightarrow{I}$ for simplification.

2) message passing on the social network factor graph

The factor graph of the social network has two types of factor nodes, namely $f(\overrightarrow{Y_i}|\overrightarrow{X_i})$ and $f(\overrightarrow{Y_i}|\overrightarrow{Y_j})$. The node $f(\overrightarrow{Y_i}|\overrightarrow{X_i})$ connects the observed interests vector $\overrightarrow{X_i}$ and the true interests vector $\overrightarrow{Y_i}$ for user u_i. The node $f(\overrightarrow{Y_i}|\overrightarrow{Y_j})$ connects the true interests $\overrightarrow{Y_i}$ and $\overrightarrow{Y_j}$ of u_i and u_j. Each edge in the factor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

Algorithm 1 User Interest Propagation (UIP) Algorithm

Require: Users observed interests $X \in \mathbb{R}^{n\times k}$, following relationships $E \in \mathbb{R}^{m\times 2}$, threshold for stopping criteria ϵ, max number of iterations K.
Ensure: True interest matrix $Y \in \mathbb{R}^{n\times k}$

```
Initialize true interest by (2), i.e. Y^old ~ N(X, I)
Initialize the message matrix M^old ∈ R^{m×8×k}, where
  M_ij^old ∈ R^{8×k} are messages corresponding to e_ij
for 1:K do
  for each e_ij ∈ E do
    Compute messages corresponding to f(Y_i^old | X_i) using (8) and (9).
    Compute messages corresponding to f(Y_i^old | Y_j^old) using (10) - (13).
    Update messages w.r.t e_ij and obtain M_ij^new ∈ R^{8×k}
  end for
  for each u_i ∈ N do
    Compute and update true interest Y_i^new using M^new by (11).
  end for
  if ||Y^new − Y^old||^2 < ε then
    break
  end if
  M^old = M^new
  Y^old = Y^new
end for
Y = Y^new
return Y
```

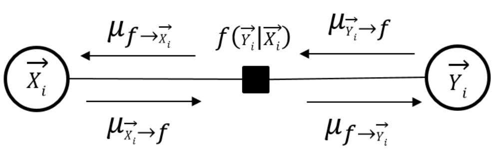

FIGURE 2: Messages corresponding to $f(\vec{Y}_i|\vec{X}_i)$.

graph corresponds to two messages. Next, we compute the messages propagated along the edges.

Messages corresponding to $f(\vec{Y}_i|\vec{X}_i)$. As shown in Fig. 2, the function node $f(\vec{Y}_i|\vec{X}_i)$ is related to four messages, namely $\mu_{\vec{X}_i\to f}$, $\mu_{f\to\vec{X}_i}$, $\mu_{\vec{Y}_i\to f}$ and $\mu_{f\to\vec{Y}_i}$. According to the message passing rule of sum-product algorithm,

$$\mu_{\vec{X}_i\to f} = 1 \tag{8}$$

$$\mu_{f\to Y_i} = \sum_{\vec{X}_i} f(\vec{Y}_i|\vec{X}_i) \propto \mathcal{N}(\vec{X}_i, \vec{I}) \tag{9}$$

The messages $\mu_{f\to\vec{X}_i}$ and $\mu_{\vec{Y}_i\to f}$ is irrelevant to the calculation of marginal functions of Y, so we just ignore them.

Messages corresponding to $f(\vec{Y}_i|\vec{Y}_j)$. As shown in Fig. 3, the function node $f(\vec{Y}_i|\vec{Y}_j)$ is related to four messages, namely $\mu_{\vec{Y}_i\to f}$, $\mu_{f\to\vec{Y}_i}$, $\mu_{\vec{Y}_j\to f}$ and $\mu_{f\to\vec{Y}_j}$, where

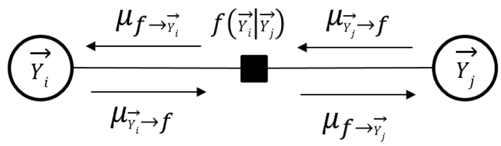

FIGURE 3: Messages corresponding to $f(\vec{Y}_i|\vec{Y}_j)$.

$$\mu_{\vec{Y}_i\to f} = \prod_{g\in n(\vec{Y}_i)\setminus\{f\}} \mu_{g\to\vec{Y}_i(x)} \tag{10}$$

$$\mu_{f\to\vec{Y}_i} = \int f(\vec{Y}_i|\vec{Y}_j)\mu_{\vec{Y}_j\to f}d\vec{Y}_j \tag{11}$$

Symmetrically, we have

$$\mu_{\vec{Y}_j\to f} = \prod_{g\in n(\vec{Y}_j)\setminus\{f\}} \mu_{g\to\vec{Y}_j(x)} \tag{12}$$

$$\mu_{f\to\vec{Y}_j} = \int f(\vec{Y}_i|\vec{Y}_j)\mu_{\vec{Y}_i\to f}d\vec{Y}_i \tag{13}$$

We introduce two equations that can be deduced using advanced mathematics.

$$\mathcal{N}(x, m_1, \sigma_1^2)\mathcal{N}(x, m_2, \sigma_2^2) \propto \mathcal{N}(x, m_3, \sigma_3^2) \tag{14}$$

in which

$$m_3 = \frac{\sigma_2^2 m_1 + \sigma_1^2 m_2}{\sigma_1^2 + \sigma_2^2}, \frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \tag{15}$$

and

$$\int \mathcal{N}(x, m_1, \sigma_1^2)\mathcal{N}(y, \alpha x, \sigma_2^2)dx \propto \mathcal{N}(y, \alpha m_1, \alpha\sigma_1^2 + \sigma_2^2) \tag{16}$$

According to (10), we know that $\mu_{\vec{Y}_i\to f}$ follows a normal distribution. For the message $\mu_{f\to\vec{Y}_i}$, we denote $\mu_{\vec{Y}_j\to f}$ as $\mathcal{N}(\vec{Y}_i, m, \sigma^2)$. Hence the message $\mu_{f\to\vec{Y}_i}$ can be expressed as

$$\mu_{f\to\vec{Y}_i} = \int \mathcal{N}(\vec{Y}_i, \vec{Y}_j, \vec{I})\mathcal{N}(\vec{Y}_i, m, \sigma^2)d\vec{Y}_i \propto \mathcal{N}(\vec{Y}_j, m, \sigma^2 + \vec{I}), \tag{17}$$

which also follows a normal distribution. This suggests that, messages can be represented by the mean value and variance of the normal distribution, making the message passing process efficient.

Flooding algorithm for message passing. The existence of circles in the social network makes the structure of factor graphs complex. Nevertheless, we expect the message passing algorithm to traverse over such structures. To this end, we need a strategy to pass messages efficiently in the factor graph.

As introduced above, the two messages corresponding to $f(\vec{Y}_i|\vec{X}_i)$ are fixed. For the factor graph node $f(\vec{Y}_i|\vec{Y}_j)$, we first initialize its messages with random values, and update the messages until convergence or when the algorithm reaches its pre-defined number of iterations. Basically, for each iteration, we compute messages based on the previous update and use this information to update all messages at the same time. We can then use the updated messages to compute the marginal functions of Y using (11). An iterative procedure is proposed for UIP in Algorithm 1.

IV. EXPERIMENTS AND RESULT ANALYSIS

We perform experiments to evaluate the performance of our user interest propagation (UIP) algorithm. To apply UIP in recommender systems, we perform additional experiments by combining our proposed UIP algorithm with the basic matrix factorization (MF), and make predictions on ratings. In particular, we show that our proposed method is robust in sparsity, outperforming baseline methods. Experiments are performed on benchmark datasets CiaoDVD [5], FilmTrust [6] and Douban [21].

A. DATASETS

The benchamak datasets CiaoDVD, FilmTrust and Douban are ideal for evaluation because it offers generated contents of users and provides a network of users. We preprocess to make the datasets applicable to our proposed algorithm. We perform interest prediction experiments on the CiaoDVD dataset since it is relatively larger than FilmTrust, and computationally less expensive than Douban. Each item in the CiaoDVD dataset belongs to one of 17 categories. For this dataset, we count the number of movies of each category rated by a user, and we take the 17-dimensional normalized vector as the users observed interests. In all three datasets, we limit ourselves to users with trust relationship with other users, and have ratings on not less than 5 items. The statistics of all datasets after preprocessing is shown in Table 1. The datasets are available for download at https://www.librec.net/datasets.html.

TABLE 1: Statistics of CiaoDVD, FilmTrust and Douban dataset

dataset	#user	#movie	$\#rating_{avg}$	$\#trust_{avg}$
CiaoDVD	1108	12450	28.20	16.11
FilmTrust	566	1912	29.68	4.71
Douban	2733	39692	333.70	23.58

B. INTERESTS PREDICTION EXPERIMENT

We seek to discover users true interests given their observed interests and relationships within networks. In situations where users have no records such as ratings in a recommender system or texts in social networks, their interest cannot be directly inferred. This is similar to the cold start problem in recommender systems. Hence in this experiment, we randomly select 10% of users in CiaoDVD and mask their observed interests and set their observed interests to $(\frac{1}{17}, \frac{1}{17}, \cdots, \frac{1}{17}) \in \mathbb{R}^{17}$ based on the 17 item categories of CiaoDVD. We then run our UIP algorithm on the dataset and try to infer the masked interests of the chosen users.

For a comparative analysis, we evaluate Neighbor Average (NA) on the same dataset. NA models the interests of u_i by taking the average value of the interests of the users he follows. NA is widely used in user profiling tasks such as inferring a user's age or sex in social networks. One obvious drawback of NA is its inability to infer the interests of u_i, when u_i does not follow any user. For NA, we also record the proportion of users whose interests cannot be inferred because they do not follow any one.

TABLE 2: Results of Interests Prediction Experiment on CiaoDVD

No.	1	2	3	4	5	avg
UIP r_s	0.54	0.60	0.54	0.60	0.54	0.57
NA r_s	0.49	0.58	0.49	0.58	0.52	0.53
NA fail	57%	48%	58%	49%	53%	53%

The evaluation metric we use is the Spearman's correlation coefficient between the masked observed interests and the inferred interests. Spearman's correlation coefficient is a measure of rank correlation which indicates the monotonic relationship between two variables. The correlation is between -1 and 1. A high correlation value indicates that two variables have a similar rank. Spearman's correlation r_s of vectors X and Y is calculated as

$$r_s = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{18}$$

where rg_X and rg_Y are ranks of X and Y respectively. $cov(rg_X, rg_Y)$ is the covariance of the rank variables and σ_{rg_X} and σ_{rg_Y} are the standard deviations of the rank variables. We sample data and conduct the experiment five times using different random seeds. We compute the performance as the average of the results obtained from the five experiments for both UIP and NA. The results are shown in Table 2.

Fron the results shown in the table, we observe that the correlation between the masked interests and the interests predicted by UIP is higher than those predicted by NA. We also find that, NA fails in predicting the interests of about 50% of the users. Given these results, it is reasonable to say that UIP outperforms NA. This particular experiment demonstrates that UIP is capable of leveraging only relationships between users to reveal the interest of users.

C. TOP3 INTERESTS PREDICTION EXPERIMENT

In some cases, we do not care much about the whole rank of user interests for the different categories, but have more interest in the several categories that users are most interested in. We define **Top k** interests of a user as the k categories that a user is most interested in. Following similar protocols as the first experiment, we again mask 10% of users observed interests for the CiaoDVD, and set their observed interests to $(\frac{1}{17}, \frac{1}{17}, \cdots, \frac{1}{17}) \in \mathbb{R}^{17}$. Then, we run UIP on this dataset and record the **Top3** dimensions, representing the categories that a user is most interested in.

For comparison, we choose Majority Count (MC) which is a voting process [13]. MC is widely used in user profiling tasks. In our experiment, we count the interests of the users

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

TABLE 3: Results of **Top3** Interest Prediction Experiment

No.	1	2	3	4	5	avg
UIP $Rank_{avg}$	3.37	3.12	3.25	3.19	3.28	3.24
MC $Rank_{avg}$	3.60	3.41	3.32	3.28	3.43	3.41
MC fail	61%	60%	57%	48%	49%	55%

TABLE 4: Results of **Top1** Interest Recovery Experiment

No.	1	2	3	4	5	avg
Top3	51%	56%	53%	51%	50%	52%
Top5	71%	77%	69%	73%	70%	72%
Top10	91%	100%	97%	96%	98%	96%

that u_i trusts, and take the 3 most frequent ones as the **Top3** interests of u_i. For evaluation purposes, we compute the average rank $Rank_{avg}$ of the predicted interests in the masked interest ranked list. The smaller the rank, the better the predicting performance of the model. If the **Top3** interests is perfectly predicted, the rank will be $(1 + 2 + 3)/3 = 2$. We perform this experiment five times and take the average score. The results of this experiment is shown in Table 3. We find that MC performs poorly compared to UIP in all 5 runs, suggesting that MC cannot effectively predict the interest of users who have no trusts. More specifically, we observe that MC fails to make predictions on about 50% of users because these users do not trust any other user, which is a great disadvantage.

D. TOP1 INTEREST RECOVERY EXPERIMENT

Based on the number of item categories of CiaoDVD, in this task we set the **Top1** interest of 10% users in CiaoDVD to $\frac{1}{17}$, and normalize the interest vector. We then run the UIP algorithm and record the rank of the original **Top1** interest in the new ranked list. We record the proportion of cases that the original **Top1** interest falls in **Top3**, **Top5**, and **Top10** in the inferred interests. The results are shown in Table 4.

On average, about 50% original **Top1** interest can be recovered in **Top3**, about 70% can be recovered in **Top5**, and almost all **Top1** interest can be recovered in **Top10** for the 17 categories. The results show that UIP is effective at unveiling the interest that users do not show explicitly.

E. APPLYING UIP TO RECOMMENDER SYSTEM

Matrix Factorization (MF) have been explored as one of the most effective tools for collaborative filtering in recommender systems. Let $R = [R_{u,i}]_{n\times m}$ be a rating matrix, where the entry $R_{u,i}$ denotes the rating of user u on item i. $R_{u,i}$ usually takes an integer value from $[0, 5]$. Following the works of [12], we normalize the ratings to the interval $[0, 1]$ to bound the range of predictions. The task is to predict the unknown rating $R_{u,i}$ using R. MF solves this by learning the latent user features U_u and latent item features V_i for u and i, such that the predicted rating $R^*_{u,i}$ approximates $U_u^T V_i$. Formally, let $U \in \mathbb{R}^{k\times n}$ and $V \in \mathbb{R}^{k\times m}$ be the respective latent feature matrices for users and items, with k-dimensional column vectors U_u and V_i for user u and item i. MF models the posterior probability over the user and item latent feature variables as

$$\begin{aligned} &p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) \propto p(R|U, V, \sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ &= \prod_{u=1}^{n}\prod_{i=1}^{m}\left[\mathcal{N}\left(R_{u,i}|g(U_u^T V_i), \sigma_R^2\right)\right]^{I_{u,i}} \times \prod_{u=1}^{n}\mathcal{N}\left(U_u|0, \sigma_U^2\mathbf{I}\right) \\ &\quad\times \prod_{i=1}^{m}\mathcal{N}\left(V_i|0, \sigma_V^2\mathbf{I}\right) \end{aligned} \tag{19}$$

where $I_{u,i}$ is an indicator which takes the value 1 if user u rated the item i, or 0 if otherwise. The function $g(x)$ is the logistic function $g(x) = 1/1 + e^{-x}$ to bound the rating predictions between $[0, 1]$. Given (19), we can learn the users' latent features U and item latent features V purely based on R through minimizing a sum-of-squares error objective function [23]. Thus, U_u can be interpreted as the learned latent features representing the observed interest of user u which does not consider social influence.

It is well noted in the literature that the interest of a user is also influenced by the interest of his neighbors in a social network [12]. To incorporate social influence, we integrate the UIP with MF to develop an MF-UIP algorithm which is composed of an MF step and a UIP step. The MF step estimates the user latent features U and V based on R by means of (19). The UIP step takes advantage of U and the following relationships E in the social network to estimate the user latent features $\hat{U}$ representing the true interests of users. Formally, for each user node $u \in N$, let N_u be its set of neighbors. Each node u is characterized by U_u. At each round $t = 1, 2, \ldots, K$ of the UIP step, every neighbor $v \in N_u$ computes a message M_{uv}^{t+1} and propagates this message to u. Upon receiving all the messages from the neighbors N_u, the UIP step updates the features of u to construct the latent feature vector $\hat{U}_u$ of the true interest of u according to (11). Suppose $\hat{U}_u$ is the latent feature vector upon convergence or at the last layer K of the UIP step. Now, MF-UIP estimates the rating $R_{u,i}$ as the inner product of $\hat{U}_u$ and V_i as:

$$\hat{R}_{u,i} = \hat{U}_u^T V_i \tag{20}$$

1) Experimental Setup

We perform experiments to demonstrate the effectiveness of MF-UIP on rating predictions. To evaluate the performance of MF-UIP, we consider the baseline methods MF, PMF [23], MF-SR [21], SocialMF [12], and NeuralMF [11]. The evaluation metric we use in the rating prediction experiment is the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). We perform each experiment five times using different random seeds and take the average RMSE or MAE.

For evaluation purposes, we randomly select 80% of the ratings of the dataset as training data, leaving out 20% for testing. The dimensionality k of the latent features is set to 10 by default. In all experiments, we set the paramaters $\lambda_U = \lambda_V = 0.001$. For SocialMF, the parameter λ_T

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3001210, IEEE Access

introduced in [12] controls the influence of the social network. We set $\lambda_T = 0.01$ for FilmTrust, and $\lambda_T = 0.001$ for CiaoDVD and Douban. For MF-SR, the parameter β introduced in [21] is set to $\beta = 0.001$ for all experiments. For NeuralMF, we choose the Adam optimizer [14], and one neural collaborative filtering layer. The remaining parameters for NeuralMF on the baseline datasets are shown in Table 5. Parameters are set empirically, and manually tuned without fine-tuning.

Hyper-parameters	CiaoDVD	Douban	FilmTrust
Embedding dimension	10	40	40
Hidden dimension	20	80	100
Batch size	50	1024	24
Learning rate	e^{-4}	$2e^{-4}$	e^{-4}

TABLE 5: Experimental settings of NeuralMF on the baseline datasets

2) Experimental Results for Rating Prediction

The overall performance of the rating prediction experiment on the datasets are shown in Table 6. A preferred model should have a low score for either RMSE or MAE. We find that all models that incorporate social influence (i.e. MF-SR, SocialMF, MF-UIP) outperform the MF-based methods (i.e. MF, PMF), suggesting that users do not always express their interests explicitly, but their interest can also be inferred from their neighbors in the social network. Although MF-SR and SocialMF outperforms the MF-based methods, they are deficient in generalization since their loss function combines both the trust and rating information, and hence the poor performance when compared to MF-UIP.

We also find that the deep learning based method NeuralMF outperforms the methods MF, PMF, MF-SR, but only outperforms SocialMF on the CiaoDVD dataset. We infer from the results that NeuralMF effectively learns the relationship between users and items when compared to MF, however as it ignores the relationships among users it fails to outperform SocialMF. It can be noted that NeuralMF takes advantage of learning higher order interactions between users and item features to show competitive performance with SocialMF. Although MF underperforms when compared to NeuralMF, the performance of our proposed model MF-UIP shows much better performance when compared to NeuralMF on all datasets. The results suggest that UIP effectively extracts relevant information from the relationship among users to enhance the user latent features modeled by MF. Since MF-UIP integrates both MF and UIP, we believe integrating NeuralMF with UIP may boost model performance, but will require the learning of additional trainable parameters. Thus, we can regard UIP as an enhancer of user features for MF-based methods exploiting social networks.

TABLE 6: Performance comparisons on rating prediction

Dataset	**method**	**RMSE**	**MAE**
FilmTrust	MF	0.86632	0.67522
	PMF	0.8661	0.67847
	MF-SR	0.86196	0.66703
	SocialMF	0.84267	0.66054
	NeuralMF	0.8574	0.6699
	MF-UIP	**0.84243**	**0.66049**
CiaoDVD	MF	1.1024	0.8603
	PMF	1.1136	0.88352
	MF-SR	1.0949	0.85877
	SocialMF	1.0888	0.88431
	NeuralMF	1.0796	0.8653
	MF-UIP	**1.0714**	**0.84593**
Douban	MF	0.7880	0.6025
	PMF	0.7877	0.6191
	MF-SR	0.7718	0.6079
	SocialMF	0.76927	0.60919
	NeuralMF	0.7825	0.6180
	MF-UIP	**0.7531**	**0.6017**

3) Impact of the dimensionality k

We analyze the impact of the dimensionality k of users and items for the models under comparison. We perform this study on the FilmTrust dataset. Here, we consider the dimension $k = 15$. We exclude the NeuralMF in this experiment since a model based on neural networks will usually need a relatively larger dimension to capture high-order interactions between the features of users and items. As shown in Table 5, the dimensionality k is set to 40 for NeuralMF to achieve its best result on the FilmTrust dataset. Table 7 shows the results on the impact of k. A quick glance at the performance shows that MF-UIP outperforms the baseline methods on the RMSE. Although MF-SR and SocialMF both aim to exploit social networks to boost performance, they fail to outperform PMF for this setting. This behaviour suggest that it is difficult to jointly learn user and item features while taking knowledge from the social network. MF-UIP simplifies this problem by incorporating social network knowledge into the learned user features independently, and assumes that the learned features of items by MF is sufficient for the recommendation task. Our result suggest that this approach is much effective.

TABLE 7: Performance comparisons on FilmTrust dataset. The dimensionality k of latent features is set to 15

method	**RMSE**	**MAE**
MF	0.8653	0.6761
PMF	0.8380	0.6815
MF-SR	0.8600	0.6695
SocialMF	0.8487	**0.6585**
MF-UIP	**0.8258**	0.6748

4) Validation in sparsity of observed ratings

We perform sparsity experiments on the rating matrix R to demonstrate the robustness of our model. Note that the increase in sparsity of R does not affect the relationship among users. Thus, we implicitly demonstrate the model's ability to leverage information from social relationships. Table 8 shows the prediction results on the training set proportions 60% and 40%. As expected, as sparsity of R increases the model performance deteriorates. Unfortunately, MF-SR fails to perform as we increase sparsity, showing its heavy dependency on R. Interestingly, we find that NeuralMF performs

TABLE 8: Performance comparisons in sparsity of observed ratings on FilmTrust

training data	**method**	**RMSE**	**MAE**
60%	MF	0.87587	0.67935
	PMF	0.87289	0.67918
	MF-SR	0.87610	0.68159
	SocialMF	0.85336	0.67383
	NeuralMF	0.8947	0.6841
	MF-UIP	**0.85085**	**0.65533**
40%	MF	0.8971	0.69971
	PMF	0.89347	0.70693
	MF-SR	0.89061	0.69675
	SocialMF	0.87436	0.69763
	NeuralMF	0.9049	0.6954
	MF-UIP	**0.82601**	**0.69198**

poorly when compared to the baseline models. It turns out NeuralMF overfits the small proportion of training data, and does not have access to information of social relationships among users to improve performance, explaining its poor performance in sparse settings. Meanwhile, we find that SocialMF and MF-UIP remains robust. But MF-UIP outperforms SocialMF in a very sparse R as seen on the 40% training data, suggesting that MF-UIP can effectively leverage information from social relationships for model performance.

5) Computational Complexity of UIP

Accurate rating predictions need large training sets which could be a problem for computational expensive models. Given that propagation based models are usually computational expensive, we investigate the computational complexity of UIP. In the model section, it can be noted that all computations take place at each factor node in the factor graph. Specifically, we compute eight types of messages (two messages are fixed; see (8) and (9)) on each local structure of the factor graph for the following relationship consisting of two users u_i and u_j. Computing these messages takes account of the sets of factor nodes associated with u_i and u_j. Thus, the computational complexity on a following relationship is proportional to the total number of factor nodes associated with u_i and u_j in the factor graph. On the largest dataset, Douban, which has 2733 users and a trust average of 23.58, the time taken for UIP to reach convergence on a CPU with a 2.6GHz processor is 4.76 min.

V. CONCLUSION AND FUTURE WORK

We introduced a User Interest Propagation (UIP) algorithm for user interest prediction which is based on the factor graph and sum-product algorithm. Unlike most previous methods which rely on only user generated content, our method moves a step further by taking account of relationships between users on an online social network to propagate relevant interests to users. Our method is naturally designed to integrate a basic matrix factorization (MF) to form a new rating prediction method, MF-UIP, for recommender systems. Generally as sparsity affects most rating prediction methods, MF-UIP is robust and outperforms several methods, including the deep learning method NeuralMF. Aside applying UIP for recommender systems, the method we develop here can be applied strategically in other network-related applications such as social search, link prediction and community detection. We intend to explore its application in these fields.

ACKNOWLEDGMENT

This work was supported in part by the China 973 Program under Grant 2015CB358700, in part by the National Natural Science Foundation of China under Grant 61772059 and Grant 61421003, in part by the Beijing Advanced Innovation Center for Big Data and Brain Computing, and in part by the State Key Laboratory of Software Development Environment.

REFERENCES

[1] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi, “Inferring user interests in the twitter social network,” in Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014, pp. 357–360.

[2] Y. Bo, L. Yu, J. Liu, and W. Li, “Social collaborative filtering by trust,” in International Joint Conference on Artificial Intelligence, 2013.

[3] Y. Cao, W. Li, and D. Zheng, “An improved neighborhood-aware unified probabilistic matrix factorization recommendation,” Wireless Personal Communications, vol. 102, no. 4, pp. 3121–3140, 2018.

[4] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in WWW, 2019.

[5] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, “Etaf: An extended trust antecedents framework for trust prediction,” in Proceedings of the 2014 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2014, pp. 540–547.

[6] G. Guo, J. Zhang, and N. Yorke-Smith, “A novel bayesian similarity measure for recommender systems,” in Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), 2013, pp. 2619–2625.

[7] ——, “Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems,” Knowledge-Based Systems, vol. 74, pp. 14–27, 2015.

[8] ——, “Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings,” in Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[9] M. Hao, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” 2009.

[10] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, “Outer product-based neural collaborative filtering,” in IJCAI 2018: 27th International Joint Conference on Artificial Intelligence, 2018, pp. 2227–2233.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” in Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.

[12] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010, pp. 135–142.

[13] C. A. D. Jr, G. L. Pappa, D. R. R. D. Oliveira, and F. D. L. Arcanjo, “Inferring the location of twitter messages based on user relationships,” Transactions in Gis, vol. 15, no. 6, p. 735–751, 2011.

[14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[15] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Transactions on information theory, vol. 47, no. 2, pp. 498–519, 2001.

[16] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, “Travel route recommendation using geotags in photo sharing sites,” in Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010, pp. 579–588.

[17] K. Lewis, M. Gonzalez, and J. Kaufman, “Social selection and peer influence in an online social network,” Proceedings of the National Academy of Sciences, vol. 109, no. 1, pp. 68–72, 2012.

[18] H. Li, J. Yan, H. Weihong, and D. Zhaoyun, "Mining user interest in microblogs with a user-topic model," China Communications, vol. 11, no. 8, pp. 131–144, 2014.
[19] J. Lian, F. Zhang, X. Xie, and G. Sun, "Cccfnet: A content-boosted collaborative filtering neural network for cross domain recommender systems," in WWW '17 Companion Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 817–818.
[20] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in Proceedings of the 17th ACM conference on Information and knowledge management, 2008, pp. 931–940.
[21] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011, pp. 287–296.
[22] M. Michelson and S. A. Macskassy, "Discovering users' topics of interest on twitter: a first look," in Proceedings of the fourth workshop on Analytics for noisy unstructured text data. ACM, 2010, pp. 73–80.
[23] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in Advances in neural information processing systems, 2008, pp. 1257–1264.
[24] N. F. Moneypenny and S. D. Flinn, "Influence-based social network advertising," Jul. 13 2008, uS Patent App. 12/172,236.
[25] M. C. Pham, Y. Cao, R. Klamma, and M. Jarke, "A clustering approach for collaborative filtering recommendation using social network analysis." J. UCS, vol. 17, no. 4, pp. 583–604, 2011.
[26] G. Piao and J. G. Breslin, "Inferring user interests in microblogging social networks: a survey," User Modeling and User-Adapted Interaction, vol. 28, no. 3, pp. 277–329, 2018.
[27] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 7, pp. 1763–1777, 2014.
[28] J. Sang, T. Mei, J.-T. Sun, C. Xu, and S. Li, "Probabilistic sequential pois recommendation via check-in data," in Proceedings of the 20th international conference on advances in geographic information systems. ACM, 2012, pp. 402–405.
[29] W. Song, Z. Xiao, Y. Wang, L. Charlin, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," 2019.
[30] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," Autonomous Agents and Multi-Agent Systems, vol. 16, no. 1, pp. 57–74, 2008.
[31] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 185–194.
[32] W. Wu, B. Zhang, and M. Ostendorf, "Automatic generation of personalized annotation tags for twitter users," in Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, 2010, pp. 689–692.
[33] Z. Xu, L. Ru, L. Xiang, and Q. Yang, "Discovering user interest on twitter with a modified author-topic model," in Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01. IEEE Computer Society, 2011, pp. 422–429.
[34] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," Computer Communications, vol. 41, pp. 1–10, 2014.
[35] X. Yang, H. Steck, and L. Yong, "Circle-based recommendation in online social networks," in ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, 2012.
[36] W. Yao, J. He, G. Huang, and Y. Zhang, "Modeling dual role preferences for trust-aware recommendation," in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, 2014, pp. 975–978.
[37] F. Zeng, L. Huang, and W. Dou, "Social factors in user perceptions and responses to advertising in online social networking communities," Journal of interactive advertising, vol. 10, no. 1, pp. 1–13, 2009.
[38] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "Idoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," Future Generation Computer Systems, vol. 66, pp. 30–35, 2017.
[39] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with gps history data," in Proceedings of the 19th international conference on World wide web. ACM, 2010, pp. 1029–1038.

SAMUEL MENSAH received the B.S. degree in Mathematics and Computer Science from the University of Ghana, Legon in 2012 and the M.S. degree in Mathematical Science from Kwame Nkrumah University of Science and Technology, Ghana, in 2015. He is currently a PhD student at the School of Computer Science and Engineering, Beihang University, China. His main area of research is natural language processing, recommender systems and knowledge graphs.

CHUNMING HU received the Ph.D. degree from Beihang University, in 2006. He was a Postdoctoral Research Fellow in the distributed system group with The Hong Kong University of Science and Technology, China, from 2006 to 2007. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University, China. His current research interests include the distributed systems, system virtualization, large scale data management and processing systems, and data mining applications.

XUE LI graduated from Beihang University with Master's degree. Her research topic is social media data analysis during Master's studies. She now works as a software development engineer at Amazon(China) on large-scale data processing and monitoring systems.

XUDONG LIU is a professor and dean of the School of Computer Science and Engineering, Beihang University, Beijing, China. He has led several China 863 key projects and government projects. He has published over 30 papers, and has more than 10 patents. His research interests include software middleware technology, software development methods and tools, large-scale information technology projects and application of research and teaching.

RICHONG ZHANG received his B.Sc. Degree and M.A.Sc. degree from Jilin University, Changchun, China, in 2001 and 2004. In 2006, he received his M.Sc. degree from Dalhousie University. In 2011, he received his Ph.D from the School of Information Technology and Engineering, University of Ottawa. He is currently an associate professor in the School of Computer Science and Engineering, Beihang University. His research interests include recommender systems, knowledge graph and crowdsourcing.