ARTICLE IN PRESS

Knowledge-Based Systems xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# How friends affect user behaviors? An exploration of social relation analysis for recommendation

Ting Yuan, Jian Cheng *, Xi Zhang, Qingshan Liu, Hanqing Lu

*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

ARTICLE INFO

*Article history:*
Received 30 January 2015
Received in revised form 7 August 2015
Accepted 8 August 2015
Available online xxxx

*Keywords:*
Recommender systems
Social recommendation
Social relation analysis

ABSTRACT

Incorporating the influence of social relationships effectively is fundamental to social recommendation (SR). However, most of the SR algorithms are based on the homophily assumption, where they ignored friends' different influence on users and users' different willingness to be influenced, which may make improper influence information integrated and harm the recommendation results. To address this, we propose a unified framework to properly incorporate the influence of social relationships into recommendation by the guidance of buddy (friends who have strong influence on user) and susceptibility (the willingness to be influenced) mining. Specifically, the Social Influence Propagation (SIP) method is proposed to identify each user's buddies and susceptibility and the Social Influence based Recommendation model is proposed to generate the final recommendation. Experiments on the real-world data demonstrate that the proposed framework can better utilize users' social relationships, resulting in increased recommendation accuracy.



## 1. Introduction

To deal with the information overload on the internet, recommender systems have emerged by suggesting users the potential enjoyed items. Traditional Collaborative Filtering (CF) methods predict users' interests by mining users' rating history. The increasing popular social networks provide additional information to enhance pure rating-based CF. Recently, based on the homophily assumption [1] that users linked with each other in social networks tend to have similar tastes, some social recommendation (SR) methods have been proposed to improve recommendation accuracy by leveraging the social relationships between users. However, among the large volume of information in social relationships, it has much noise which will disturb the recommendation. To get better results, we should take deep analysis on effectively modeling the influence of social relationships into recommendation.

Studies on social networks show that the social relationships have multiple aspects of influence on users. First, they can affect users' decisions directly, which can be regarded as the short-term influence of social relationships. In [2,3], psychology and sociology studies have proved that users' decisions are affected simultaneously by *individual tastes* and *social influence*. This is also intuitive in online social networks. In a product review sites, such as Epinions,[1] when a user rates an item, she/he usually cares about the item's characteristic to see whether it is interesting, depending on the individual tastes. She/He also cares about the opinions from the trusted friends,[2] which is the social influence. Second, users' individual tastes are also affected by friends with time elapsing, which can be regarded as the long-term influence of social relationships. In [4], the authors studied the interaction between taste similarity and social influence in product review sites. They observed that the average difference of rating behaviors between the users and their friends decreased after their relations established, and it continued decreasing as time going. Therefore, it is necessary to incorporate the influence of social relations properly and comprehensively into recommendation. However, the homophily assumption about social relationships used in traditional SR algorithms is debatable.

On the one hand, not all friends are influential to users in social network. As we know, social networks sometimes serve a more general purpose of allowing users socialise among themselves instead of just reflecting their agreement in item ratings. For

* Corresponding author.
*E-mail addresses:* tyuan@nlpr.ia.ac.cn (T. Yuan), jcheng@nlpr.ia.ac.cn (J. Cheng), xi.zhang@nlpr.ia.ac.cn (X. Zhang), qsliu@nuist.edu.cn (Q. Liu), luhq@nlpr.ia.ac.cn (H. Lu).

[1] Epinions: http://www.epinions.com.
[2] In directed networks such as Twitter, the "friends" in our paper represent the persons who link from the user.

http://dx.doi.org/10.1016/j.knosys.2015.08.005
0950-7051/

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

instance, when a user establishes a link to a person she/he knows in the real world, she/he may just want to be informed of the person's activities, and they do not necessarily have similar preferences or influence each others' decisions. This is a common situation in Facebook, Twitter and some other social networks. Therefore, among the hundreds of friends in the social network, there is much noise and only part of them really have effects on users' decisions. It is unnecessary to consider all the friends' influence when predicting the rating behavior. Additionally, in [5], it is indicated that the really influential friends' behaviors play a very important role on diffusion of items. To the best of our knowledge, previous SR works do not pay attention on the effects of these really influential friends, and most of them take account of all the friends' influence uniformly. In our work, we attempt to detect each user's *buddies* who have strong influence on the user, and focus on their rating influence on users. Simultaneously, we treat these buddies differently according to their influence strength.

On the other hand, not all the people are apt to be influenced by others in the social networks. Some users are more susceptible while some are less, which has been revealed in [5] by randomized experimentation on samples of 1.3 million Facebook users. For example, the authors found that younger users are more susceptible to be influenced than older ones, married individuals are the least susceptible to be influenced. Thus, we should also treat the target users differently when considering their friends' influence. For the susceptible users, their rating decisions may depend more on their friends' influence, while for the unsusceptible users, more effects come from their individual tastes. In previous social recommendation works, they confused all the target users together without taking the role of susceptibility into account.

To address all the problems, we propose a novel recommendation framework integrating individual *buddy* and *susceptibility* analysis into *social* influence based *recommendation*, named BSSR for short. To detect each user's buddies and susceptibility, as well as the influence strength of buddies, we develop a *Social Influence Propagation (SIP)* method based on the theory of factor graphs and sum-product algorithm [6]. Specifically, an *Influence Factor Graph* is constructed, which captures users' rating behaviors and social network structure into a unified model to analyze the relationships. Then the accuracy of recommendation can be improved by considering the short-term and long-term social influence guided by the buddy's and susceptibility's effects. The main contributions of this paper are summarized below:

- Proposing a novel social recommendation framework, BSSR, guided by mining users' buddy sets and susceptibility to incorporate social relations more properly than traditional SR methods.
- Developing a Social Influence Propagation (SIP) method based on the unified Influence Factor Graph to mine buddy sets and susceptibility simultaneously.
- Two aspects of social influence, short-term and long-term influence, are considered in the final social influence based recommendation.

The rest of the paper is organized as follows. We first discuss the related work in Section 2. Then, our new recommendation framework, BSSR, is introduced in Section 3. The Social Influence Propagation method for individual buddy and susceptibility mining is described in Section 4, a social influence based recommendation model guided by buddy and susceptibility is presented in Section 5. Then we analyze experimental results on benchmark datasets in Section 6. Finally, we conclude this work in Section 7.

## 2. Related work

In this section, we review some related works, including traditional recommendation approaches based on collaborative filtering (CF), recommendation techniques enriched by social relationship, and works about social relation analysis.

### 2.1. Collaborative filtering

Techniques based on collaborative filtering are widely used in recommender systems [7–11]. In general, it is based on the fundamental assumption that similar users have similar behaviors on similar items [12–15]. CF methods are mainly divided into two categories: memory-based and model-based. Memory-based methods [7,8] usually ask for similar users' or items' advice to produce a prediction. They can be further categorized as user-based methods [16,8,17] or item-based methods [7,18,19], depending on whether the recommendation for a user is aggregated from users with similar preference to her/him or from items similar to those she/he already liked. However, memory-based methods are limited in handling highly sparse data since it is difficult to estimate the similarity accurately.

Different from memory-based methods, the model-based methods learn a model based on patterns recognized in the known ratings of users by machine learning and statistical techniques, and then apply the model to do recommendation. Examples include the latent semantic models [20,21], clustering models [22,23], graphical models [24], and Bayesian models [25,26]. Among different model-based methods, low-rank matrix factorization (MF) techniques have attracted much research attention [9–11,27], due to the advantages of scalability and accuracy. Based on the premise that users' tastes can be represented by a small number of factors, MF techniques learn the low-rank latent factors of users and items from the observed ratings in the user-item rating matrix, and then utilize them to predict user's behavior. However, all the CF methods mentioned above rely only on users' history rating behaviors, it may be insufficient in the context of social networks where the users' interactions influence the decision making dramatically.

### 2.2. Social recommendation

Several social recommendation (SR) algorithms have been proposed to investigate how social relations can be utilized to provide better recommendations [28–35]. In [28], the authors proposed the trust-based model by extending traditional memory-based methods with social network among users. They replace the similarity computation process with the use of a trust metric and make recommendations based on the ratings of users who are trusted. However, the experiments on a large real dataset show that this work can only increase the coverage (number of ratings that are predictable) but fail to improve the prediction accuracy. Jamali and Ester [34] employ the random walk approach [36] to combine the trust-based model and the item-based model. It considers not only ratings of the target item, but also those of similar items. The random walk model helps to measure the confidence of a recommendation. Their experiments show that this method outperform other existing memory based approaches. However, it is not scalable to large datasets, since it needs to calculate pairwise similarities for each prediction.

Recently, some social recommendation methods based on matrix factorization techniques show substantial improvements. One popular way for fusing social relationship into MF model is to factorize the user-item rating matrix and user–user social relation matrix jointly by sharing a common user latent factor matrix.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

Examples can be seen in [30,32,33]. By this way, they treat the social recommendation as a multi-relation learning task, where the user-item rating matrix and user–user social relationship are two kinds of relations. By sharing the same user latent factor matrix, information from social relation can be transferred to improve the rating predictions. However, in [37], the authors have done experiments to show that this kind of jointly factorization model is suitable for the links like membership (user-group), but not very suitable for the links like friendship (user–user).

In [29], the authors fuse social relationship into MF model by social regularization, which targets to constrain the difference between the latent factors of user's friends and herself/himself. The underlying principle for this kind of fusion is homophily, where they think users linked together have similar tastes. Experiments of [37] show that the regularization model is more suitable for fusing friendships into recommendation than the jointly factorization model mentioned above. However, in the regularization model, the real-world recommendation processes, where user makes decisions based on her/his own taste and friends' influence, are not reflected. In [38], the authors consider the real-world decision process into recommendation, where they modeled one user's ratings as the balance between the user's own favors and the tastes of her/his friends. However, this method is also based on the homophily assumption, where they think people tend to behave similarly when they are linked. Under the homophily assumption, above works consider all the friends' influence equally and treat all the users the same to be influenced by friends, which may make improper influence information to be integrated and harm the recommendation results. In our paper, we attempt to incorporate the social relations more properly into recommendation by considering the different influence of buddies and the susceptibility of each user, which are mined from our inner analysis of social influence.

Recently, some extended works have been done on the basis of above social recommendation models. Jiang et al. [39] extended the jointly factorization model for twitter recommendation by considering the contextual information such as the content of tweets and the people who send the tweets. Yang et al. [40] extended the regularization model by considering the category information, they inferred the category-specific circles of friends to influence the recommendation of items belong to each category. However, these works need some additional information besides user-item ratings and user–user relationships, such as content information and category information, which are not the scenarios discussed in this paper. Additionally, nor do they pay attention to the buddies' and susceptibility's role in social recommendation. One related work is [31], the authors extend the regularization model by weighting each social link regularization with the rating similarity between users. By this way, they try to treat friends differently according to the rating similarity. However, it is not sufficient to consider the influence strength simply as the rating similarity, which has been verified in our experiments in Section 6.6. In our paper, we will take inner analysis of social influence to identify the buddies who really influence users' ratings, instead of just considering the influence strength as rating similarity.

### 2.3. Social relation analysis

There exist some works focusing on social relation analysis [41–52]. Additionally, some platforms also try to analyze the social influence, such as the Klout,[3] Followerwonk,[4] InfluenceTracker[5] [53]. Social relation analysis attracts more and more attention. Most of the works aim at the link prediction or link strength analysis. In [45], the authors study how positive and negative relations among users can be predicted using various topological features of a social network. Schifanella et al. [47] shows that users with similar topical interests are more likely to be friends, and similarity measures among users based on their annotation metadata are used as the predictive indication of social links. Xiang et al. [48] proposes a generative model to estimate relationship strength in social networks based on observed user interactions and similarities. Yang et al. [49] studies the problem of labeling the exist edges of social network as positive or negative relations by capturing users' behavior, social interactions and the interplay between them. Since buddies and susceptibility are both social influence related attributes, we detect them on the basis of social influence analysis. Unlike the works mentioned above, we consider the influence as a propagated attribute among users in the network. In [54], the author addresses to infer topic-level influence by incorporating the user's topic distribution and influence propagation into a unified factor graph model. In our work we inherit the notion. But directed towards our task's characteristics, we focus on the effects of both the user-item interaction similarity and topological feature of social network, and incorporate them with influence propagation to detect each user's buddies and susceptibility.

## 3. Our framework – BSSR

### 3.1. Framework overview

Fig. 1 describes an illustration of our framework. The input is a heterogeneous graph containing both user-item ratings and user–user relationships. As discussed before, each user has lots of friends and not all of them have strong influence on the user's decisions. Additionally, not all users are willing to be influenced. Our goal is to find the buddies who really influence users' rating behaviors as well as the susceptibility of each user, and to do recommendation guided by these useful mined information. It consists of the following two stages.

**Stage 1: Buddy and susceptibility mining**. Since buddy can be regarded as the person who has strong influence on user, and susceptibility is the willingness for user to be influenced by others, the essential of this step is to analyze the social influence strength. Considering the information available here (only rating and relationship information), we tackle the problem based on three factors associated with rating behaviors and social network structure, and formulate them into a unified influence factor graph based on the theory of factor graphs [6]. The Social Influence Propagation (SIP) method is proposed to operate in the graph by passing messages under optimization rule between users. The detail of this part is presented in Section 4.

**Stage 2: Social influence based recommendation**. Based on each user's buddies and susceptibility, we can exploit the social relations for recommendation more properly. First, we focus on the buddies' different rating influence on users rather than taking account of all the friends' influence. Second, target users are treated differently by their susceptibility when considering the contribution of their individual tastes and their friends' effects in making the final decisions. Third, we also consider two aspects of the influence of social relationships, the short-term influence and the long-term influence. To address all the issues, we extend our social influence based recommendation model on the basis of Probabilistic Matrix Factorization (PMF) [9] to generate a final prediction. The detail of this part is presented in Section 5.

### 3.2. Problem definition

In social recommender systems, we have a set of users $\mathcal{U} = \{u_1, \ldots, u_N\}$ and a set of items $\mathcal{V} = \{v_1, \ldots, v_M\}$. The users'

[3] https://klout.com.
[4] https://followerwonk.com/.
[5] http://influencetracker.com/.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

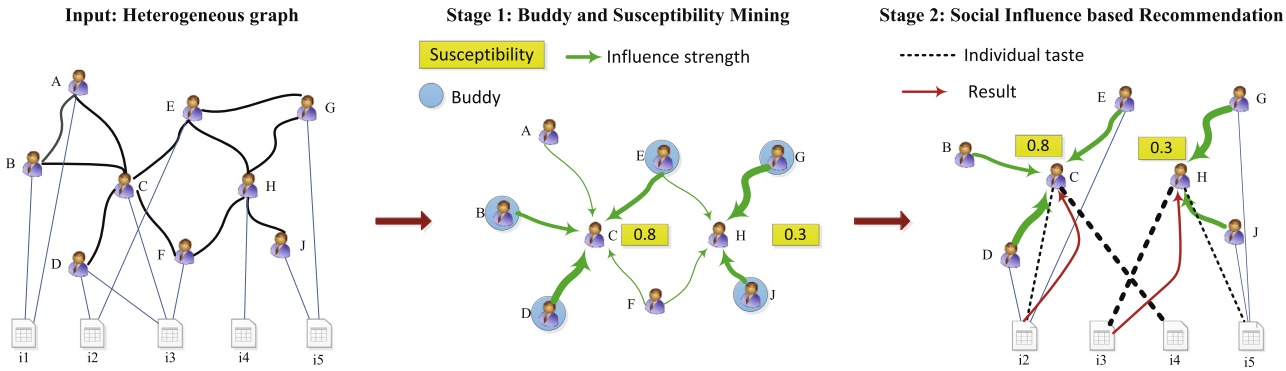


**Fig. 1.** An illustration for BSSR framework. Given the user-item interactions and user–user relationships, to do recommendation for user C and H. Firstly we attempt to mine C's and H's buddies (friends who have strong influence on the user) and susceptibility (the willingness to be influenced) by the social relation analysis. Then recommendation is done by considering both the user's individual taste and buddies' influence. For the susceptible user C (with high susceptibility value), the decision is made mainly by considering buddies' influence, while for the unsusceptible user H (with low susceptibility value), the decision is made mainly based on individual taste.

ratings on items are expressed in a rating matrix $\mathbf{R} = [R_{ij}]_{N\times M}$, where $R_{ij}$ denotes the rating of $u_i$ on item $v_j$. Each user $u_i$ has a set of social linked neighbors $\mathcal{NB}_i^+$ and $\mathcal{NB}_i^-$, where $\mathcal{NB}_i^+$ denotes the neighbors who link from $u_i$ and $\mathcal{NB}_i^-$ denotes the neighbors who link to $u_i$. Note that in undirected networks (e.g. Facebook), each user's $\mathcal{NB}^+$ and $\mathcal{NB}^-$ are the same, both of them are the "friends". While in directed networks (e.g. Twitter), the $\mathcal{NB}^+$ and $\mathcal{NB}^-$ are different. In this paper, "friends" in directed networks stand for the persons who are followed by the users, so $\mathcal{NB}^+$ represent the "friends".

Given the rating records $\mathbf{R}$ and the network relations $\mathcal{NB}^+$ and $\mathcal{NB}^-$, our aim is to find each user $u_i$'s *buddy* set, denoted as $\mathcal{B}_i$, and the susceptibility, denoted as $S_i$, as well as the influence strength $\mathbf{P} = [P_{ik}]_{N\times N}$, where $P_{ik}$ denotes $u_k$'s influence strength on $u_i$, and then predict the missing values in the user-item rating matrix $\mathbf{R}$ effectively by the guidance of the above mined information.

## 4. Buddy and susceptibility mining

In social networks, both buddy and susceptibility can be considered as characteristics associated with the social influence. In this section we focus on analyzing the social influence to detect them. To capture users' rating behavior as well as the social network structure for social influence analysis, we construct an Influence Factor Graph which incorporates multiple crucial factors into a unified model. Then the Social Influence Propagation (SIP) method is proposed based on the factor graph we construct.

### 4.1. Basic ideas

To discover a user's buddies and estimate her/his susceptibility, it is essential to analyze the social influence. In our work, the final goal is to do recommendation with the help of social relationships. It is based on Collaborative Filtering, which has the assumption that similar users have similar tastes on similar items. Thus, the influential friends (buddies) of each user we want to find is the friends whose preference is really similar to the user. Thus the influence we want to infer should reflect the taste similarity. One significant factor is the rating similarity because rating records reflect users' interest and high rating similarity can infer a high taste similarity. As we know, rating similarity is usually measured based on the items that the users rate in common [16,31]. However, in some product review sites like Epinions, the users and their friends usually rate different sets of items. One extreme case is that there is no item they rate in common, the rating similarity may be considered as zero. Since an influential friend does not need to rate the same items as the user, the zero rating similarity score here cannot stand for weak influence definitely. Thus, the rating similarity is not sufficient to represent the social influence.

To analyze the social influence deeply, we also take into account the network structure. In social link analysis [45], the social topological feature *edge embeddedness*, which is defined as the number of the common friends of two nodes, offers important indications to link inference. In this paper, we use edge embeddedness to show how similar the users are with their common friends. It is not only about the number of common friends but also about the rating behavior of their common friends. That is, if a friend has many common friends with similar tastes to the user, she/he should have strong influence on the user.

Both the rating similarity and edge embeddedness can be taken as the local features related to two nodes. Additionally, we consider that influence will propagate between users in the whole network, there should be some global constraints in the view of the whole network structure. In [55,56], a clustering algorithm named affinity propagation (AP) is proposed by passing messages between data points under some global rules. The algorithm tries to identify exemplars among data points and forms clusters around these exemplars. To get the clustering results that not only maximize net similarity but also satisfy global cluster constraints, the AP algorithm utilizes a factor graph to capture both of the factors (net similarity and global cluster constraints) and proposes a message propagation method based on it. The messages propagating in AP algorithm reflect the attitude of one node to another node, which is to some extend similar to the influence between users which we want to analyze. Inspired by the idea of AP, we formulate our problem into a factor graph based on the factor graph theory [6] and try to find the influence-related messages as AP did. Different from AP [55,56], our task is to analyze the social influence for detecting each users' buddies and susceptibility. Thus, we should concentrate on the characteristics of social influence and incorporate all the factors mentioned above into our *Influence Factor Graph* model.

### 4.2. Constructing influence factor graph

In our model, we have a set of observed variables $\mathcal{U} = \{u_i\}_{i=1}^N$ and a set of hidden variables $\mathcal{X} = \{x_i\}_{i=1}^N$ corresponding to the $N$ users. Each $x_i$ respects the friend who has the highest influence strength on the corresponding user $u_i$, and takes the value from the set $\{\mathcal{NB}_i^+ \bigcup u_i\}$. For each user $u_i$, we have two kinds of factor functions associated with the hidden variables, considering all the factors mentioned above:

- **local factor function** $l_{u_i}(x_i)$ captures both the rating similarity and edge embeddedness of user and her/his neighbors.
- **global factor function** $g_{u_i}(x_1, \ldots, x_N)$ constrains the valid configurations of the whole network.

Here, we use Vector Space Similarity (VSS) to define the rating similarity, which is widely used in traditional user-based and item-based recommendation approaches [16,7]:

$$Vss(u_i, u_f) = \frac{\sum_{v_j \in I(i) \bigcap I(f)} R_{ij} \cdot R_{fj}}{\sqrt{\sum_{v_j \in I(i) \bigcap I(f)} R_{ij}^2} \cdot \sqrt{\sum_{v_j \in I(i) \bigcap I(f)} R_{fj}^2}} \tag{1}$$

where $I(i)$ denotes the items rated by $u_i$. As discussed before, considering both the number and the tastes of common friends, we define the edge embeddedness score as:

$$Emb(u_i, u_f) = \frac{\sum_{u_p \in cof(i,f)} Vss(u_i, u_p)}{\sum_{u_q \in \mathcal{NB}_i^+} Vss(u_i, u_q)} \tag{2}$$

where $cof(i,f) = \mathcal{NB}_i^+ \bigcap \mathcal{NB}_f^+$, denotes the common friend sets of $u_i$ and $u_f$. The above definition comes from the following intuition: if user $u_f$ has many common friends with $u_i$, while these common friends' rating tastes are similar to $u_i$, $u_f$ should have strong influence on $u_i$. Note that $Emb(u_i, u_f)$ and $Emb(u_f, u_i)$ are asymmetric.

Local factor function $l_{u_i}$ fuses the two local features (rating similarity and edge embeddedness) as follows:

$$l_{u_i}(x_i) = \begin{cases} \dfrac{\alpha Vss(u_i, x_i) + \beta Emb(u_i, x_i)}{Z}, & \text{if } x_i \neq u_i \\ \dfrac{\sum_{u_f \in \mathcal{NB}_i^-} \alpha Vss(u_f, u_i) + \beta Emb(u_f, u_i)}{Z}, & \text{if } x_i = u_i \end{cases} \tag{3}$$

where Z is the normalization factor,[6] $\alpha$ and $\beta$ are the embedded coefficients, which combines $Vss(u_i, u_f)$ and $Emb(u_i, u_f)$ to represent $u_f$'s local influence feature level on $u_i$. The local factor function is defined mainly considering following intuition: if a friend $u_f$ has a high local feature similarity (rating similarity and edge embeddedness) with $u_i$, $u_f$ should have a strong influence on $u_i$; if other users trust $u_i$ highly, then $u_i$ should trust herself/himself strongly. In [5], extensive experiments show that: highly influential individuals tend to be unsusceptible and trust themselves more, which reveals the latter intuition.

However, not all configurations of $x_i$ are suitable in the view of the whole network. Note that $\mathcal{X} = \{x_i\}_{i=1}^N$ are the most influential friend for the corresponding users. However, $x_i$ can take value from $u_i$'s friends as well as $u_i$ herself/himself. If there is no constrains, everyone will choose herself/himself as the most influential friends, because the local features of herself/himself will be the most similar. To constrain the model to bias towards the true influential users of the whole network, we have the following constrain: user $u_i$ is expected to choose other persons as her/his $x_i$; if it has to be herself/himself ($x_i = u_i$), $u_i$ should be the most influential friend of at least another user ($\exists x_f = u_i, f \neq i$). Thus, for each $u_i$, the global factor function for the satisfaction of the constraint is defined as:

$$g_{u_i}(x_1, \ldots, x_N) = \begin{cases} 0 & \text{if } x_i = u_i, \text{ and } x_f \neq u_i \text{ for all } f \neq i \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

The joint distribution of both the local and global factor function over all the observed data factorizes as follows:

$$\mathcal{L} = \prod_{i=1}^{N} l_{u_i}(x_i) \prod_{k=1}^{N} g_{u_k}(x_1, \ldots, x_N) \tag{5}$$

Our goal is to maximize the log-likelihoods of the joint distribution function.

Based on the factor graph theory [6], we describe the factorization into an *Influence Factor Graph*, as shown in Fig. 2. In [6], a factor graph can represent the product of functions by two kinds of nodes, variable node and factor node, and the edge-connecting factor node to corresponding variable node. Here, each $x_i$ is a variable node, $g_{u_i}$ and $l_{u_i}$ are the factor nodes. Since $l_{u_i}(x_i)$ only responds to the variable $x_i$, it only has an edge to $x_i$; $g_{u_i}(x_1, \ldots, x_N)$ is a global function responding to all the variables, it connects to all the variable nodes. While it is intractable to find the exact solution to Eq. (5), by constructing this influence factor graph, the sum-product algorithm can be used to find the approximate configuration for it [6].

### 4.3. Social Influence Propagation (SIP)

The sum-product algorithm can find the parameter configuration for factor graph by passing two kinds of messages on the edge of the factor graph [6]: messages from variable nodes to factor nodes and messages from factor nodes to variable nodes. The two messages are computed differently according to the sum-product update rule. That is, a message sent from a variable node $y$ on an edge $e$ is the *product* of all the messages received at $y$ on edges other than $e$; a message sent from a factor node $F$ on an edge $e$ is the *product* of the factor function on $F$ with all messages received at $F$ on edges other than $e$, and then *summarized* for the variable associated with $e$. The messages are updated iteratively until convergence.

To learn our Influence Factor Graph equals to maximizing the logarithm of function $\mathcal{L}$ in Eq. (5). Thus, the max-sum algorithm, which is the log-domain version of the max-product algorithm, can be used to search over parameter configurations in our model. This algorithm is identical to the sum-product algorithm, except that it computes maximums instead of sums, and sums instead of products in the update rule.

Denote the message from $x_i$ to $g_{u_k}$ as $\varphi_{i \to k}(x_i)$, the message from $g_{u_k}$ to $x_i$ as $\psi_{i \leftarrow k}(x_i)$. According to the max-sum update rule [6], we have:

$$\begin{aligned} \varphi_{i \to k}(x_i) &= \log l_{u_i}(x_i) + \sum_{f \neq k} \psi_{i \leftarrow f}(x_i) \\ \psi_{i \leftarrow k}(x_i) &= \max_{\sim \{x_i\}} \left[ \log g_{u_k}(x_1, \ldots, x_N) + \sum_{f \neq i} \varphi_{f \to k}(x_f) \right] \end{aligned} \tag{6}$$

where $\sim \{x_i\}$ denotes the variable sets containing all the variables except $x_i$.

In affinity propagation [55], the author introduce two sets of variables *responsibility* $r_{ik}$ and *availability* $a_{ik}$ on the basis of the max-sum update rules, where $r_{ik}$ indicate how strongly data point $i$ favors point $k$ as its cluster center, $a_{ik}$ indicate to what degree point $k$ thinks itself available as a cluster center for point $i$. Thus, for point $i$, the value of $k$ that maximizes $r_{ik} + a_{ik}$ identifies the cluster center for $i$. Inspired by affinity propagation, we also introduce the two sets of variables *responsibility* $r_{ik}$ and *availability* $a_{ik}$ for each social link, which will convert passing messages on the factor graph (see in Eq. (6)) into passing messages directly between users. Through involved derivation from the message update rules in Eq. (6), we can get an equivalent update rules for $r_{ik}$ and $a_{ik}$ (the derivation is given in Appendix A): for $u_k \in \mathcal{NB}_i^+$,

$$r_{ik} = \log l_{u_i}(u_k) - \max_{u_f \in \mathcal{NB}_i^+ \& u_f \neq u_k} \{a_{if} + \log l_{u_i}(u_f)\} \tag{7}$$

$$a_{kk} = \max_{u_f \in \mathcal{NB}_k^-} \min(r_{fk}, 0) \tag{8}$$

$$a_{ik} = \min\{-\min(r_{kk}, 0) - \max_{u_f \in \mathcal{NB}_k^- \& u_f \neq u_i} \min(r_{fk}, 0), \max(r_{kk}, 0)\} \tag{9}$$

[6] $Z = \sum_{u_f \in \mathcal{NB}_i^+} (\alpha Vss(u_i, u_f) + \beta Emb(u_i, u_f)) + \sum_{u_f \in \mathcal{NB}_i^-} (\alpha Vss(u_f, u_i) + \beta Emb(u_f, u_i))$.

According to affinity propagation [55], this two messages *responsibility* and *availability* have a nice explanation in our case: as shown in Fig. 3, for a social relation where $u_i$ is linked to $u_k$ ($u_k \in \mathcal{NB}_i^+$), responsibility message $r_{ik}$, which is sent from $u_i$ to her/his trustee $u_k$, respects how strongly $u_i$ agrees that $u_k$ influences on her/him; availability message $a_{ik}$, which is sent from $u_k$ to her/his truster $u_i$, respects how strongly $u_k$ thinks herself/himself influences on $u_i$. Thus, the two messages $r_{ik}$ and $a_{ik}$ both represent the influence strength from $u_k$ on $u_i$, but from different views of influence receptor ($u_i$) and sender ($u_k$). Note that in the new update rules, the responsibility $r_{ik}$ from $u_i$ to $u_k$ are computed taking into account the availabilities from $u_i$'s other trustee, the availability $a_{ik}$ from $u_k$ to $u_i$ are computed taking into account the responsibilities from $u_k$'s other truster. Thus, by updating $r_{ik}$ and $a_{ik}$ on the social links, the two new kinds of influence related messages propagate between users in the whole social network.

Computing responsibilities and availabilities by the update rules in Eq. (7)–(9) recursively, the social influence propagate in the whole network and converge after numbers of iterations. We name this learning process as *social influence propagation* (SIP).

**Algorithm 1.** Algorithm for Buddy and Susceptibility Mining

**Require:** rating matrices $\mathbf{R} = [R_{ij}]_{N\times M}$, social relationships $\mathcal{NB}^+$ and $\mathcal{NB}^-$
**Ensure:** each users' buddies $\mathcal{B}_i$ and susceptibility $S_i$, the influence strength $\mathbf{P} = [P_{ik}]_{N\times N}$

Calculate the local factor function for each link $l_{u_i}(u_k)$ according to Eq. (3);
Initialize all $r_{ik}$ as 0;
**Repeat**
  **for** each social link ($u_k \in \mathcal{NB}_i^+$) **do**
    Update $r_{ik}$ according to Eq. (7);
  **end for**
  **for** each user $u_k$ **do**
    Update $a_{kk}$ according to Eq. (8);
  **end for**
  **for** each social link ($u_k \in \mathcal{NB}_i^+$) **do**
    Update $a_{ik}$ according to Eq. (9);
  **end for**
**Until** convergence;
**for** each user $u_i$
  **for** each neighboring user $u_k \in \{\mathcal{NB}_i^+ \bigcup u_i\}$
    Calculate influence strength $P_{ik}$;
  **end for**
  Select out the Buddy set $\mathcal{B}_i$;
  Calculate the susceptibility $S_i$ according to Eq. (11);
**end for**

### 4.4. Identify buddy and susceptibility

Based on SIP, we can get the final responsibility and availability between each social relation. To discover each user's buddy sets and estimate her/his susceptibility, at first we define the probability of $u_i$ to be influenced by $u_k$ as follows by considering the two views of influence:

$$P_{ik} = \frac{1}{1 + e^{-(r_{ik}+a_{ik})}} \tag{10}$$

Now, to find the buddies who have strong influence on the target user, we set a natural criterion as: *a user $u_k$ is the buddy of $u_i$, denoted as $u_k \in \mathcal{B}_i$, if and only if $P_{ik} > \varepsilon_i$ and $u_k \in \mathcal{NB}_i^+$*. In our experiments we set $\varepsilon_i$ as the average influence probability of $u_i$'s friends on $u_i$. Note that we will detect different number of buddies for different users.

To get each user's susceptibility, we consider the following two points: first, if a user has strong influence on herself/himself, she/he is confident and will be unsusceptible; second, if other persons' influence on the user is strong, she/he should be susceptible. Thus, we have the following susceptibility $S_i$ for each user $u_i$:

$$S_i = \frac{\frac{1}{nb_i}\sum_{u_k \in \mathcal{NB}_i^+} P_{ik}}{P_{ii} + \frac{1}{nb_i}\sum_{u_k \in \mathcal{NB}_i^+} P_{ik}} \tag{11}$$

where $nb_i$ denotes the number of $u_i$'s friends. We summarize the detail algorithm for buddy and susceptibility mining in Algorithm 1.

## 5. Social influence based recommendation

In this section, we illustrate our social influence based recommendation guided by buddy and susceptibility.

We formulate our problem on the basis of Probabilistic Matrix Factorization (PMF) [9], which learns latent factors of the users and the items to represent their characteristics. Supposing users' tastes can be represented by these latent characteristics, decisions are predicted by users' and items' latent factors. Let $U \in \mathbb{R}^{d\times N}$ and $V \in \mathbb{R}^{d\times M}$ be the user and item latent factor matrices respectively, with column vectors $U_i$ and $V_j$ representing $d$-dimensional user-specific and item-specific latent factors of user $u_i$ and item $v_j$. Our goal is to learn these latent variables and exploit them for recommendation.

First, to predict the rating decision, we consider three points: (1) user makes decisions based on individual tastes and friends' influence. (2) Instead of considering all the friends' influence, we should focus on the buddies' influence on user's ratings. (3) Different users should be treated differently when considering the contributions of their own tastes and others' influence in the final decision. Thus, we have following predicted rating:

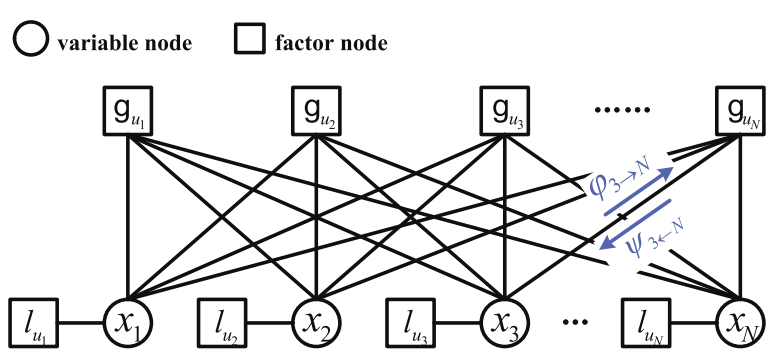


**Fig. 2.** Influence factor graph.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

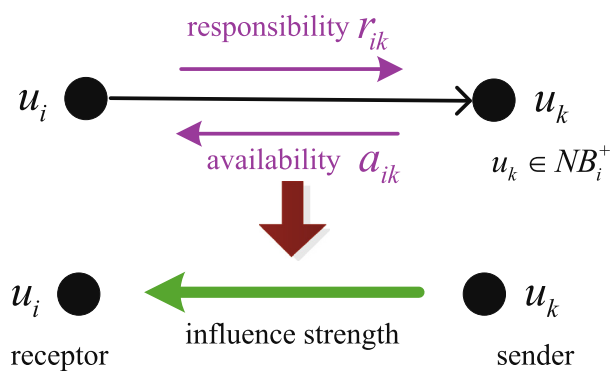


**Fig. 3.** An example of social influence propagation.

$$\hat{R}_{ij} = (1 - S_i)U_i^T V_j + S_i \frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} P_{ik}} \tag{12}$$

In Eq. (12), a user $u_i$'s social rating decision on item $v_j$ is made by two factors: $u_i$'s individual taste on $v_j$ ($U_i^T V_j$); the rating influence of $u_i$'s buddies $\left(\frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} P_{ik}}\right)$, where $\mathcal{B}_i$ is the buddy set of $u_i$. The mined susceptibility $S_i$ controls the contributions of the two factors: if a user is susceptible (with high $S_i$ value), the final decision depends more on friends' influence; if a user is unsusceptible (with low $S_i$ value), the final decision depends more on individual taste. Note that here we also consider the buddies' different influence strength on target user by the inferred influence probability $P_{ik}$.

We adopt a probabilistic model with Gaussian observation noise as in [9]. The conditional distribution over the observed ratings is defined as:

$$\begin{aligned} p(\mathbf{R}|U, V, \mathbf{\Theta}, \sigma_R^2) &= \prod_{i=1}^{N}\prod_{j=1}^{M} [\mathcal{N}(R_{ij}|\hat{R}_{ij}, \sigma_R^2)]^{I_{ij}^R} \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij}|(1 - S_i)U_i^T V_j + S_i \frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} P_{ik}}, \sigma_R^2\right)\right]^{I_{ij}^R} \end{aligned} \tag{13}$$

where $I_{ij}^R$ is the indicator function that is equal to 1 if $u_i$ has rated $v_j$ and equal to 0 otherwise. $\mathbf{\Theta}$ denotes all the information ($\mathcal{B}/\mathbf{S}/\mathbf{P}$) we have mined in the first stage. $\mathcal{N}(R_{ij}|\hat{R}_{ij}, \sigma_R^2)$ indicates the probability density function of the Gaussian distribution with mean $\hat{R}_{ij}$ and variance $\sigma_R^2$.

As discussed before, besides affecting user's rating on items directly (the short-term influence), the social relationships also affect user's individual tastes as time going (the long-term influence). Here, we also take attention to the long-term influence on the basis of the observations in [4]: the average taste difference between users and their friends decreases after their relations have been established. Thus for users' latent factors which represent the user's tastes, we have following prior:

$$\begin{aligned} p(U|\mathcal{NB}, \sigma_U^2, \sigma_N^2) = \prod_{i=1}^{N} \mathcal{N}\left(U_i \middle| \frac{1}{nb_i} \sum_{u_k \in \mathcal{NB}_i^+} U_k, \sigma_N^2 \mathbf{I}\right) \\ \times \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \end{aligned} \tag{14}$$

Here we regularize user's individual taste by the average tastes of the friends, zero-mean Guassian prior is to avoid over-fitting.

Through a Bayesian inference, we have the following posterior distribution of latent factors $U$ and $V$:

$$\begin{aligned} &p(U, V|\mathbf{R}, \mathbf{\Theta}, \sigma_R^2, \sigma_N^2, \sigma_U^2, \sigma_V^2) \\ &\propto p(\mathbf{R}|U, V, \mathbf{\Theta}, \sigma_R^2) p(U|\mathcal{NB}, \sigma_U^2, \sigma_N^2) p(V|\sigma_U^2) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij}|(1 - S_i)U_i^T V_j + S_i \frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} P_{ik}}, \sigma_R^2\right)\right]^{I_{ij}^R} \\ &\times \prod_{i=1}^{N} \mathcal{N}\left(U_i|\frac{1}{nb_i} \sum_{u_k \in \mathcal{NB}_i^+} U_k, \sigma_N^2 \mathbf{I}\right) \times \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \times \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}) \end{aligned} \tag{15}$$

The log of the posterior distribution is given by:

$$\begin{aligned} &\ln p(U, V|\mathbf{R}, \mathbf{\Theta}, \sigma_R^2, \sigma_N^2, \sigma_U^2, \sigma_V^2) \\ &= -\frac{1}{2\sigma_R^2} \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}^R \left(R_{ij} - (1 - S_i)U_i^T V_j - S_i \frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} P_{ik}}\right)^2 \\ &\quad - \frac{1}{2\sigma_N^2} \sum_{i=1}^{N} \left(\left(U_i - \frac{\sum_{u_k \in \mathcal{NB}_i^+} U_k}{nb_i}\right)^T \left(U_i - \frac{\sum_{u_k \in \mathcal{NB}_i^+} U_k}{nb_i}\right)\right) \\ &\quad - \frac{1}{2\sigma_U^2} \sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^{M} V_j^T V_j \end{aligned} \tag{16}$$

Maximizing the log-posterior distribution is equivalent to minimizing the following objective function, which is the-sum-squared errors with regularization terms:

$$\begin{aligned} \mathcal{J}(U, V, \mathbf{R}, \mathbf{\Theta}) = &\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}^R \left(R_{ij} - \left((1 - S_i)U_i^T V_j + S_i \frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} P_{ik}}\right)\right)^2 \\ &+ \lambda_N \sum_{i=1}^{N} \|U_i - \frac{1}{nb_i} \sum_{u_k \in \mathcal{NB}_i^+} U_k\|_F^2 + \lambda_U \sum_{i=1}^{N} \|U_i\|_F^2 + \lambda_V \sum_{j=1}^{M} \|V_j\|_F^2 \end{aligned} \tag{17}$$

where $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, $\lambda_N = \sigma_R^2/\sigma_N^2$, and $\|.\|_F$ is the Frobenius norm.

A local minimum of the objective function in Eq. (17) can be found by performing gradient descent on the latent factors $U_i$ and $V_j$. Let $e_{ij} = \hat{R}_{ij} - R_{ij}$,

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial U_i} = &(1 - S_i) \sum_{j=1}^{M} I_{ij}^R e_{ij} V_j + \sum_{u_f \in \mathcal{A}_i} S_f \frac{\sum_{j=1}^{M} I_{fj}^R e_{fj} P_{fi} V_j}{\sum_{u_k \in \mathcal{B}_f} P_{fk}} \\ &+ \lambda_N \left(U_i - \frac{\sum_{u_k \in \mathcal{NB}_i^+} U_k}{nb_i}\right) - \lambda_N \sum_{u_f \in \mathcal{NB}_i^-} \frac{U_f - \frac{\sum_{u_k \in \mathcal{NB}_f^+} U_k}{nb_f}}{nb_f} \\ &+ \lambda_U U_i \frac{\partial \mathcal{J}}{\partial V_j} = \sum_{i=1}^{N} I_{ij}^R e_{ij} \left((1 - S_i)U_i + S_i \frac{\sum_{u_k \in \mathcal{B}_i} P_{ik} U_k}{\sum_{u_k \in \mathcal{B}_i} P_{ik}}\right) + \lambda_V V_j \end{aligned} \tag{18}$$

where $\mathcal{A}_i$ is the set that includes all the users who have $u_i$ as the buddy. We can achieve the convergent latent variables $U$ and $V$ by updating the values of them repeatedly according to Eq. (18), then we can predict the missing values to make recommendation by Eq. (12). In order to reduce the model complexity, we set $\lambda_U = \lambda_V$ in our experiments as other papers [27,30,29,38].

**Table 1**
Statistics of the datasets.

| | Douban | Epinions |
|---|---|---|
| # of Users | 7711 | 7008 |
| # of Items | 43,269 | 106,693 |
| # of Ratings | 1,688,192 | 349,965 |
| # of Links | 358,692 | 304,971 |
| Ave Ratings per User | 218 | 49 |
| Ave Ratings per Item | 39 | 3 |
| Ave links per User | 46 | 43 |
| Rating Sparsity | 99.49% | 99.95% |
| Network Sparsity | 99.40% | 99.38% |

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

**Table 2**
Performance comparison on Douban datasets.

| Methods | Douban 70% | | Douban 50% | |
|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE |
| ItemCF | 0.8150 ± 0.0013 | 0.6479 ± 0.0011 | 0.8232 ± 0.0014 | 0.6536 ± 0.0011 |
| PMF | 0.7937 ± 0.0009 | 0.6153 ± 0.0007 | 0.8162 ± 0.0005 | 0.6351 ± 0.0005 |
| SoRec | 0.7802 ± 0.0012 | 0.6068 ± 0.0009 | 0.8030 ± 0.0004 | 0.6249 ± 0.0005 |
| RSTE | 0.7738 ± 0.0013 | 0.6026 ± 0.0009 | 0.7898 ± 0.0015 | 0.6165 ± 0.0010 |
| SoReg | 0.7689 ± 0.0017 | 0.5965 ± 0.0013 | 0.7817 ± 0.0017 | 0.6107 ± 0.0015 |
| BSR | 0.7513 ± 0.0016 | 0.5915 ± 0.0011 | 0.7687 ± 0.0015 | 0.6053 ± 0.0012 |
| SSR | 0.7617 ± 0.0013 | 0.5958 ± 0.0011 | 0.7778 ± 0.0009 | 0.6149 ± 0.0010 |
| BSSR-S | 0.7562 ± 0.0016 | 0.5902 ± 0.0014 | 0.7723 ± 0.0013 | 0.6072 ± 0.0010 |
| BSSR | **0.7436 ± 0.0016** | **0.5852 ± 0.0012** | **0.7583 ± 0.0013** | **0.5990 ± 0.0011** |

## 6. Experiments

In this section, we conduct extensive experiments to evaluate our BSSR by comparing to state-of-the-art algorithms on two real-world datasets.

### 6.1. Datasets

The suitable datasets for this research would be the real-world data from online social networks, which contains the users' rating behaviors as well as their social relationships. Thus, we conduct our experiments on the following datasets.

**Douban dataset.** Douban[7] is a Chinese social website which provides user rating, review and recommendation services for movie, books and music. Users can rate items using a five-point scale (1–5) to express their preference on them. In addition, users can establish relations with others to be informed of the friends' activities. To evaluate our models recommendation quality, we conduct our experiments on a subset of Douban for rating prediction, which contains the users' ratings on items as well as the social relations. The dataset contains 7711 users and 43,269 items with 1,688,192 ratings and 358,692 relationships.

**Epinions dataset.** Epinions[8] is a well-known product review website where users can read and write reviews on a various of items (such as electronic products, cars, books, household goods,...) and also rate items using a five-point scale (1–5). Each member of Epinions maintains a "trust" list to indicate explicitly her/his attitude on others, which also forms a social network among users. The dataset consists of 7008 users and 106,693 items with 349,965 ratings and 304,971 trust relationships.

The detailed statistics of the two datasets are showed in Table 1. In both datasets, the high sparsity is rather noticeable in user-item rating matrices as well as user–user relationships. Compared to Douban, Epinions owns more items and less ratings, which leads to a sparser rating matrix. However, the network sparsity of the two datasets are comparable.

### 6.2. Experimental setups

#### 6.2.1. Performance measurement

In the experimental study, we focus on the task of rating prediction to evaluate our model's recommendation quality. The users' rating dataset is randomly divided into two folds, the training set and the testing set. We train the model on the training data, and predict the ratings in the testing set using the model learned from the training set.

Since we want to measure the rating prediction accuracy, we choose the two most popular metrics for this task, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

MAE is defined as:

$$\mathrm{MAE} = \frac{1}{T}\sum_{i,j}|R_{ij} - \hat{R}_{ij}| \tag{19}$$

RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum\nolimits_{i,j}(R_{ij} - \hat{R}_{ij})^2} \tag{20}$$

where $R_{ij}$ denotes the rating score of user $u_i$ on item $v_j$, $\hat{R}_{ij}$ denotes the predicted rating score of user $u_i$ on item $v_j$, and $T$ denotes the number of tested ratings. The smaller MAE or RSME value means a better performance.

To get convincing results, we conducted experiments on different amounts of training data (70% and 50%) to test the models' performance under different sparsity cases. For example, for training data 70%, we randomly select 70% of the ratings for training and the rest for testing. The random selection was carried out 5 times independently, and we report the average results.

#### 6.2.2. Baselines

To validate the effectiveness of our BSSR model, we compare it with the following baselines, which contains the state-of-the-art Collaborative Filtering methods, the social recommendation techniques, and some partial configurations of our model.

- **ItemCF** [7]: The standard item-based collaborative filtering method predicts the rating of the item according to the user historical ratings on the similar items. It only uses the user-item interaction information.
- **PMF** (Probabilistic Matrix Factorization) [9]: The baseline matrix factorization model which learns low-rank latent factors for users and items to predict the preference. It only uses the user-item matrix for recommendation.
- **SoRec** (Social Recommendation) [30]: Assuming the inner product of two user's latent factors can indicate two users' relationship, this method jointly factorizes the user-item rating matrix and user–user relation matrix by sharing the same user latent space.
- **RSTE** (Recommendation with Social Trust Ensemble) [38]: This method incorporate the social relation information into recommendation by modeling one user's ratings as the balance between the user's own favors and the tastes of her/his friends. However, it neglects the different influence strength of friends and the different susceptibilities of users.
- **SoReg** (Recommendation with Social Regularization) [31]: This method designs a matrix factorization objective function with social regularization, which constrains the difference between the latent factors of user's friends and herself/himself. It uses the rating similarities between users to weight the regularization, but it also considers all the friends' influence.

[7] http://www.douban.com.
[8] http://www.epinions.com.

Meanwhile, to investigate the effects of our mined buddies and susceptibility respectively, we also compare the following two partial configurations of our model:

- **BSR** (Buddy based Social Recommendation): This method predicts decisions guided by the different influence of each user's buddies and ignores the susceptibility's effects. The adjusted function is similar to Eq. (17) but with the predicted rating as $\hat{R}_{ij} = (1-S)U_i^T V_j + S\frac{\sum_{u_k \in \mathcal{B}_i} p_{ik} U_k^T V_j}{\sum_{u_k \in \mathcal{B}_i} p_{ik}}$, where $S$ is uniform to all users.
- **SSR** (Susceptibility based Social Recommendation): This method takes into account the individual susceptibility's effects but considers all the friends' influence on decisions. The objective function is similar to Eq. (17) but with the predicted rating as $\hat{R}_{ij} = (1-S_i)U_i^T V_j + S_i \sum_{u_k \in \mathcal{N}\mathcal{B}_i^+} \frac{1}{nb_i} U_k^T V_j$.

To investigate the effects of long-term influence, we also implement following comparison:

- **BSSR-S** (BSSR with only short-term influence): In this method, we follow the framework of BSSR model, but do not consider the long-term influence of social relationships with $\lambda_N = 0$ in Eq. (17).

### 6.3. Performance comparison

The experimental results using 10 dimensions to represent the latent factors are shown in Tables 2 and 3. The parameter settings of our approach are $\lambda_N = 10$ for Douban dataset and $\lambda_N = 20$ for Epinions dataset, for both datasets $\lambda_U = \lambda_V = 0.01,\ \alpha = \beta = 0.5$.

From the results, we can observe that our BSSR model consistently outperforms other approaches on both datasets, no matter using the 70% training set or the 50% training set. For example, when using 70% as training data: on Douban dataset, BSSR can improve the performance by 6.31% and 3.29% in terms of RMSE, and 4.89% and 1.89% in terms of MAE in contrast to PMF (state-of-the-art algorithm in traditional CF models) and SoReg (state-of-the-art algorithm in social recommendation models), respectively; on Epinions, BSSR improves the performance as high as 6.14% and 2.83% in terms of RMSE, and 6.90% and 3.94% in terms of MAE in contrast to PMF and SoReg, respectively. We can observe that social recommendation algorithms are all consistently better than traditional CF models, which provides a strong evidence that the social relations are useful to improve the recommendation accuracy. Note that the improvements of BSSR over the best baseline SoReg are comparable to the improvements of SoReg over PMF on both datasets, which demonstrates that our model gets a significant improvements and can better utilize the social relations into recommendation.

It should be noted that both BSR and SSR achieve better performance than the baseline social recommendation methods (SoRec, RSTE and SoReg), which demonstrates the effectiveness of either buddies or susceptibility. Meanwhile, BSR producing higher improvements than SSR shows that the buddies' guidance plays a larger role than susceptibility. This may because that there are much noise among social relations, it is more imperative to detect each user's real influential friends to filter out the noisy friends' influence. However, the improvement achieved by BSSR over both BSR and SSR demonstrates the importance of considering both the two aspects' information simultaneously.

Note that, both RSTE and BSSR-S consider only the short-term influence, the difference is that RSTE do not consider the buddies' and susceptibility's effect. It is obvious that BSSR-S outperforms RSTE consistently in all cases, which shows the significance of buddy and susceptibility mining for effectively incorporating the short-term influence of relationships into recommendation. Additionally, we can observe that BSSR-S cannot perform as good as BSSR, which demonstrates that it is necessary to consider the long-term influence of social relationships. Furthermore, compared with SoReg which only considers the long-term social regularization, BSSR-S can get improved results. It demonstrates that the first term of Eq. (17) (the short-term influence) plays a more important role for improving results than the social regularization term of Eq. (17) (the long-term influence).

### 6.4. Performance on different users

In social recommendation methods including our model, there are two important user related information: ratings and social relationships. Different users have different numbers of ratings or relationships. How does our model works under these different scenarios? Hence, in order to perform comprehensive comparison with other methods, we evaluate the prediction accuracies of different users.

#### 6.4.1. Users with different number of observed ratings

Here, we group all users based on the number of observed ratings in the training data, and evaluate the performance on different user groups with different rating numbers. The experimental results are shown in Fig. 4 (Douban dataset) and Fig. 5 (Epinions dataset). In both datasets, users are grouped into 6 classes: "0-10", "11-20", "21-40", "41-80", "81-160" and ">160", denoting how many ratings users have rated.

Figs. 4(a) and 5(a) summarize the distribution of the testing data according to the user groups with different ratings in the training data (70% as training data). For example, there are a total 4499 user-item ratings to be predicted in the Epinions testing dataset where the related users in the training data have rating numbers from 1 to 10.

**Table 3**
Performance comparison on Epinions datasets.

| Methods | Epinions 70% | | Epinions 50% | |
|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE |
| ItemCF | 1.1620 ± 0.0016 | 0.9028 ± 0.0012 | 1.1753 ± 0.0014 | 0.9110 ± 0.0010 |
| PMF | 1.1213 ± 0.0017 | 0.8664 ± 0.0014 | 1.1474 ± 0.0007 | 0.8821 ± 0.0008 |
| SoRec | 1.0957 ± 0.0015 | 0.8499 ± 0.0013 | 1.1188 ± 0.0007 | 0.8657 ± 0.0007 |
| RSTE | 1.0899 ± 0.0011 | 0.8434 ± 0.0009 | 1.1053 ± 0.0012 | 0.8601 ± 0.0009 |
| SoReg | 1.0831 ± 0.0016 | 0.8397 ± 0.0013 | 1.0985 ± 0.0011 | 0.8535 ± 0.0006 |
| BSR | 1.0611 ± 0.0017 | 0.8177 ± 0.0013 | 1.0738 ± 0.0014 | 0.8334 ± 0.0009 |
| SSR | 1.0710 ± 0.0011 | 0.8297 ± 0.0010 | 1.0865 ± 0.0009 | 0.8442 ± 0.0007 |
| BSSR-S | 1.0695 ± 0.0013 | 0.8131 ± 0.0010 | 1.0796 ± 0.0011 | 0.8304 ± 0.0006 |
| BSSR | **1.0525 ± 0.0015** | **0.8066 ± 0.0011** | **1.0663 ± 0.0014** | **0.8249 ± 0.0007** |

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

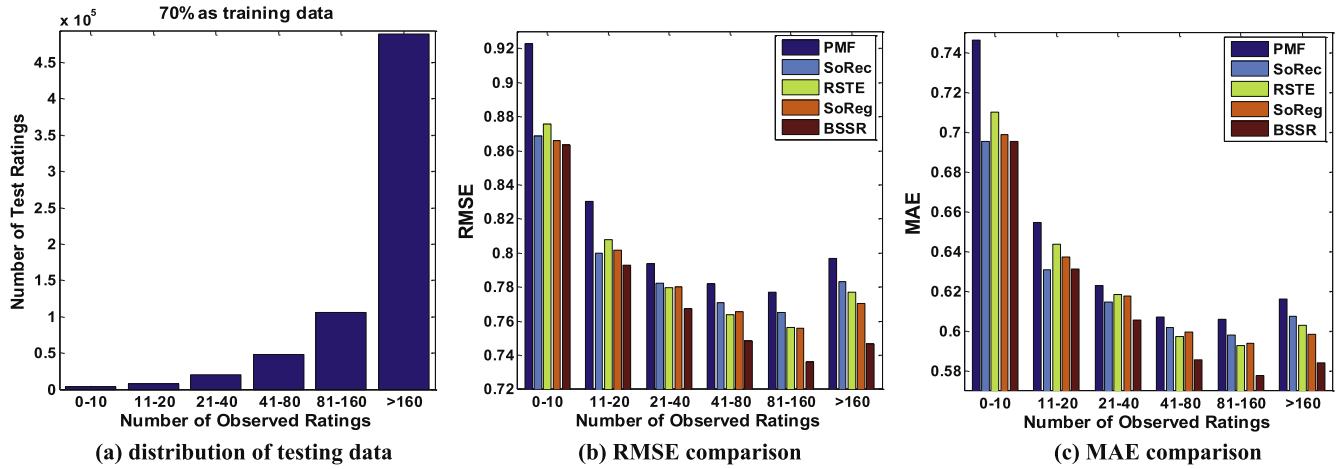


**Fig. 4.** Performance comparison of users with different number of ratings on Douban.

From Figs. 4 and 5, we can get the following observations: (1) In general, almost all the methods get better results on the users who have more ratings, which demonstrates that user's ratings on item are very valuable for modeling his/her taste. (2) For the baseline social recommendation (SR) methods (SoRec, RSTE and SoReg), all of them get consistently better results than PMF (tradition CF method) on users with different number of ratings. Additionally, the improvements over PMF are more obvious when few user ratings are given. When more rating is given, it is more challenging for SR methods to get significant improvement. (3) Our method BSSR gets improvements over other SR methods on all the user cases. However, when the users have few ratings, the improvements are not as obvious as on users with more ratings. This may because that our improvement over other SR methods mainly come from the influence analysis. Since the influence analysis of our model is based on both the user's ratings and the social relationships, it is difficult for us to get good influence analysis results on the users who have few ratings. Nevertheless, when the users have more ratings, BSSR get obvious improvement than other SR methods due to effective influence analysis on users.

### 6.4.2. Users with different number of friends

Here, we group all users based on the number of friends they have, and evaluate the performance on different user groups with different friend numbers. The experimental results are shown in Fig. 6 (Douban dataset) and Fig. 7 (Epinions dataset). In both datasets, users are grouped into 4 classes: "<30", "30-70", "70-150" and ">150", denoting how many friends users have rated.

Figs. 6(a) and 7(a) summarize the distribution of the testing data according to the user groups with different number of friends. For example, there are a total 30,765 user-item ratings to be predicted in the Douban testing dataset where the related users have friend numbers <30.

From Figs. 6 and 7, we can get the following observations: (1) For the baseline social recommendation (SR) methods (SoRec, RSTE and SoReg), all of them get improvements over PMF on users with different number of friends. However, the improvements are obvious only when the users have few friends (<30). When the users have large number of friends, the improvements become less obvious. This may because that there exists much noise among these large number of friends, traditional SR methods do not deeply analyze the social influence to filter out the noise and find the real influential friends, which will make improper social influence information to be incorporated and harm the recommendation results. (2) Our method BSSR perform better than other SR methods consistently, especially get obvious improvement on users with many friends, which demonstrates that social influence analysis is significant for guiding the incorporation of social relations into recommendation.

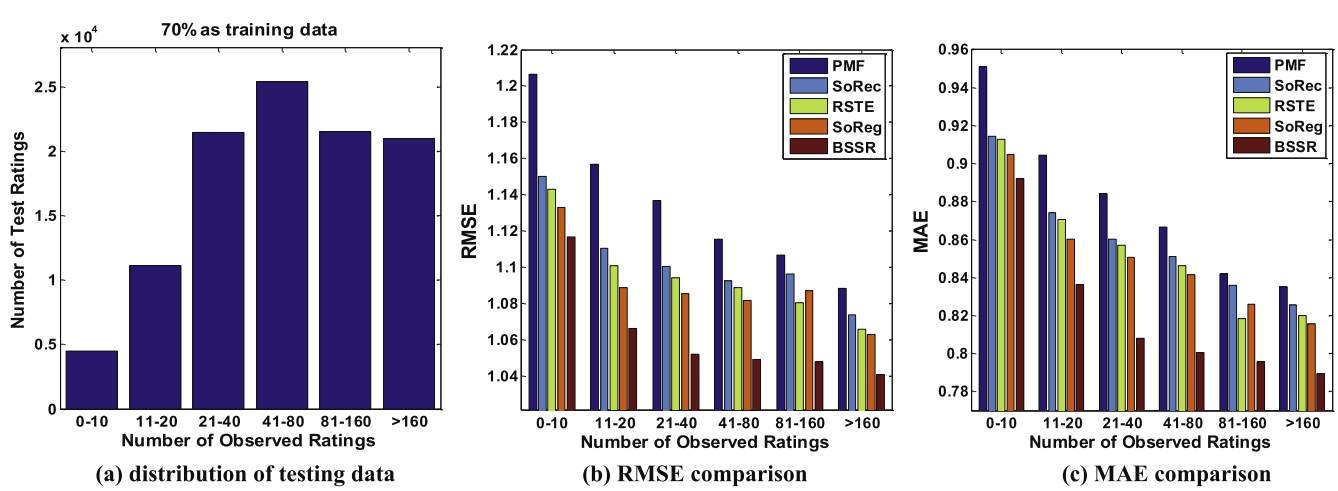


**Fig. 5.** Performance comparison of users with different number of ratings on Epinions.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

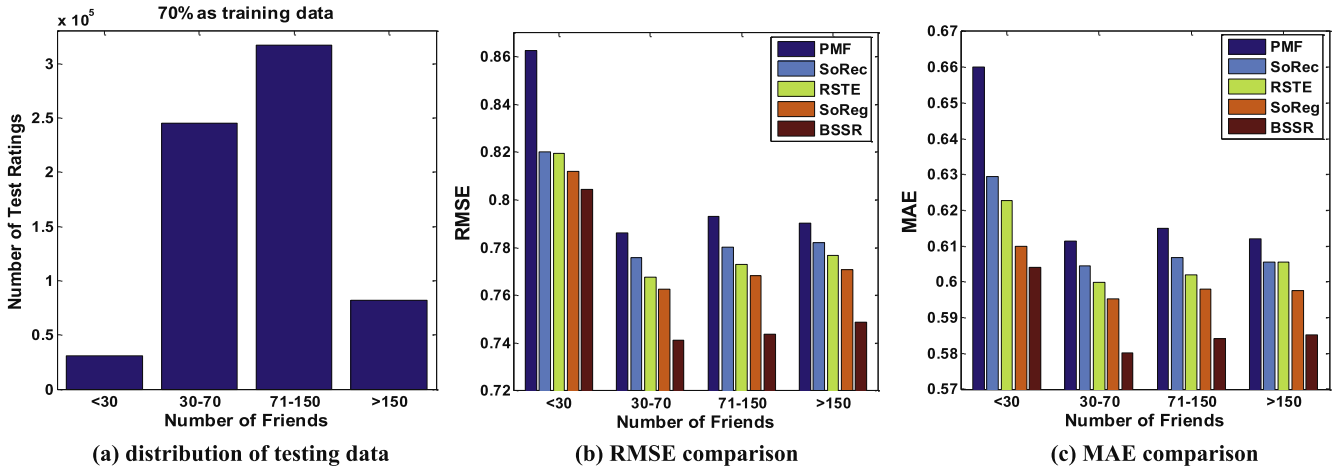


**Fig. 6.** Performance comparison of users with different number of friends on Douban.

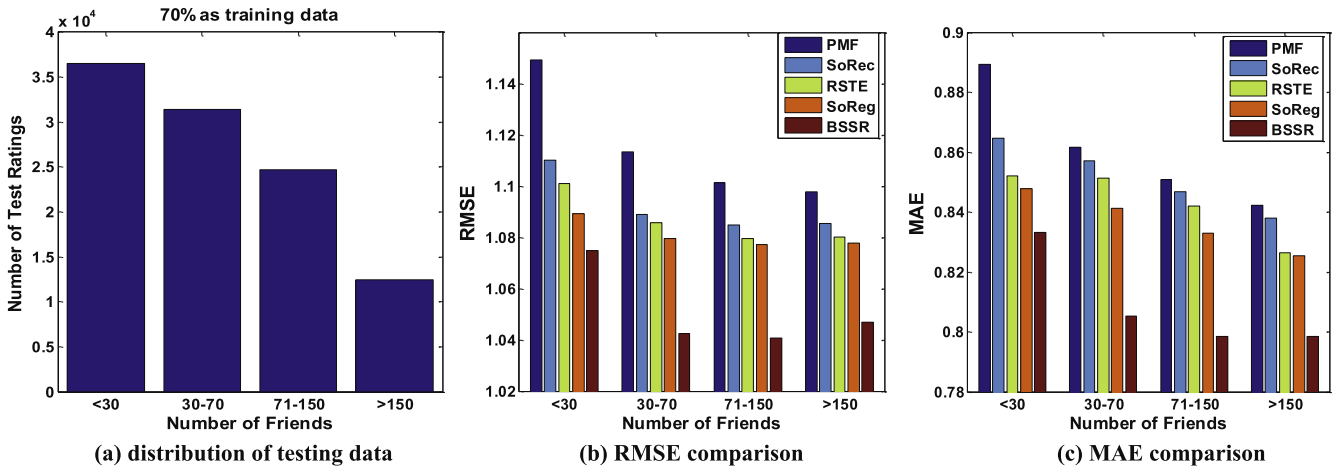


**Fig. 7.** Performance comparison of users with different number of friends on Epinions.

### 6.5. Parameter analysis

Here, we investigate the effects of two important parameters in our model: the number of latent factors $d$ and the $\lambda_N$.

**Number of latent factors $d$:** The number of latent factors $d$ is an important parameter for low-rank matrix factorization techniques. If $d$ is too small, the recommender system cannot make a distinction between any users or items. If $d$ is too large, users and items will be too unique to find their similarities in tastes and the complexity will considerably increase. Note that PMF, SoRec, RSTE, SoReg and BSSR are all based on the matrix factorization models. To investigate the effects of $d$, we conduct experiments on the four different methods by varying $d$ on both datesets with the 70% and 50% training data. The results are shown in Figs. 8 and 9.

We can observe that, in general RMSE and MAE reduce with the latent factor number $d$ increasing. There exist a convergence effect of $d$: when increasing the latent factor number $d$ to be around 10,

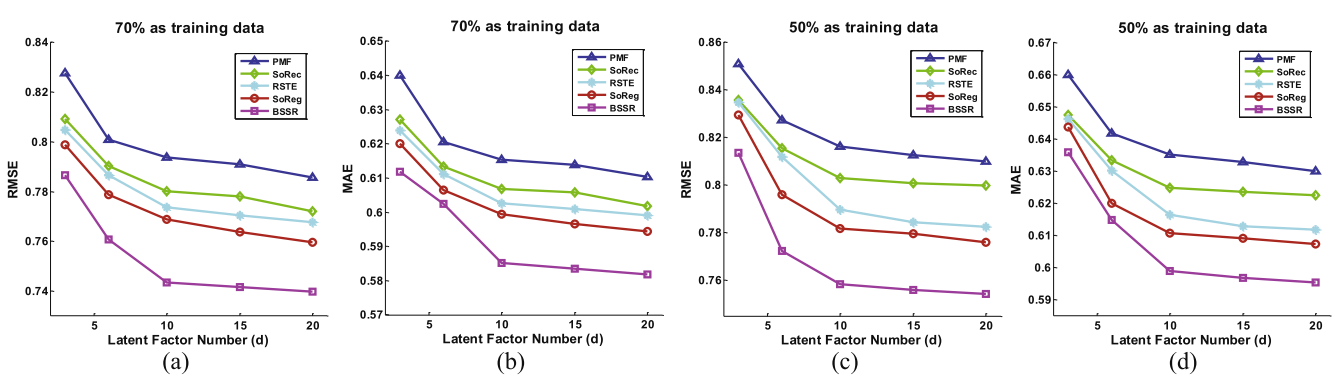


**Fig. 8.** RMSE and MAE performance of different latent factor numbers on Douban.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

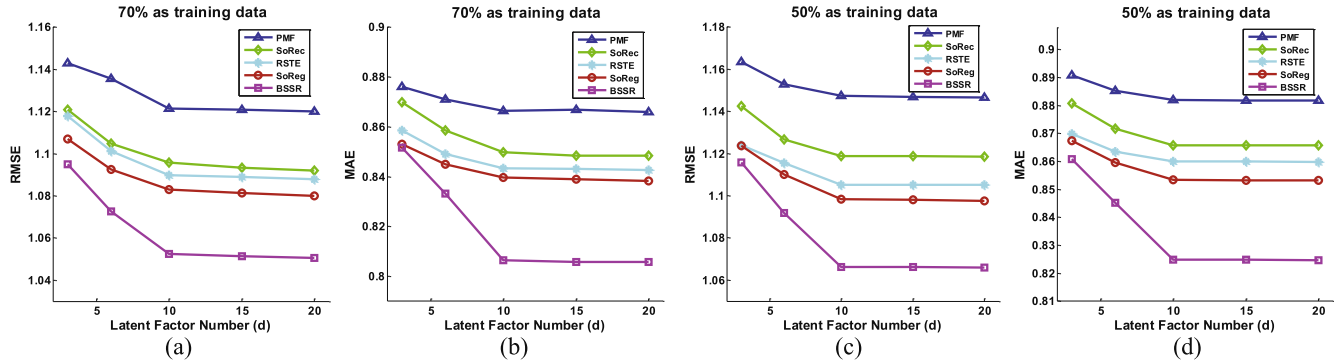


**Fig. 9.** RMSE and MAE performance of different latent factor numbers on Epinions.

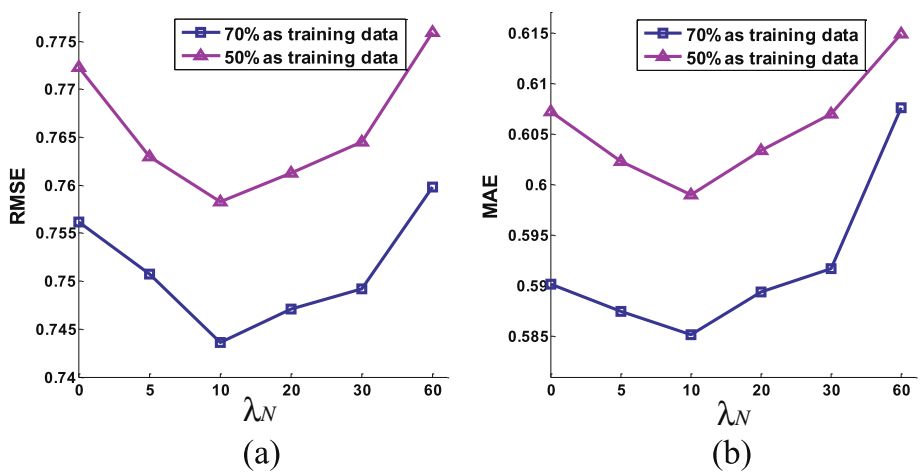


**Fig. 10.** Impact of $\lambda_N$ on the RMSE and MAE performance of Douban.

there seem to be little improvement for larger $d$. This suggests that a small number of latent factors (such as 10) is enough for the four models. Additionally, from the results, it shows obviously that our BSSR methods consistently outperforms other methods varying with different number of latent factors on both datasets.

**Parameter $\lambda_N$:** $\lambda_N$ controls the contribution of social relation's long-term influence in the whole objective function. In the extreme case, if the value of $\lambda_N$ is very small, the user's individual tastes will not be influenced by others. On the other side, if the value of $\lambda_N$ is very large, the social relations' long-term influence will dominate the learning processes. Figs. 10 and 11 show the performance of our BSSR on both datasets with different values of $\lambda_N$. We can see that, no matter whether using 70% training data or 50% training data, in both datasets the RMSE and MAE results decrease at first, but when $\lambda_N$ goes greater than a threshold ($\lambda_N = 10$ for Douban and $\lambda_N = 20$ for Epinions) the results increase. This observation coincides with the intuitions: considering the long-term influence of social relationships is useful for recommendation; yet, if too much

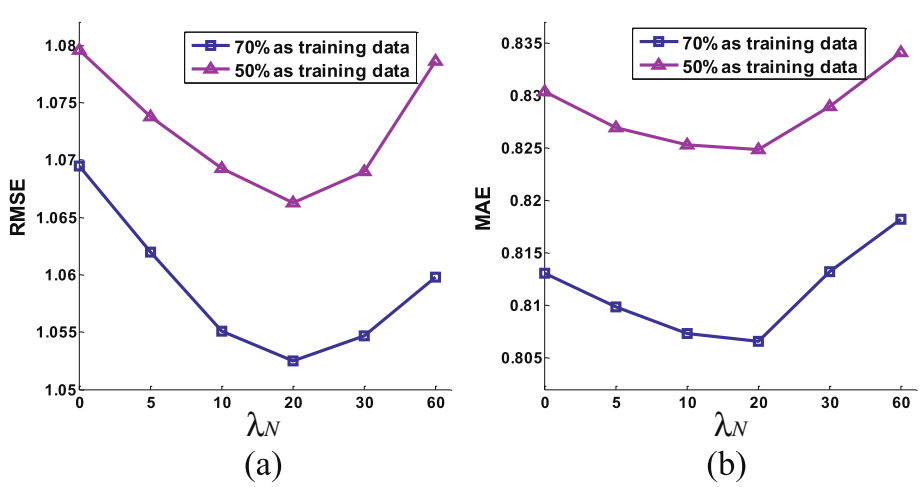


**Fig. 11.** Impact of $\lambda_N$ on the RMSE and MAE performance of Epinions.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

**Table 4**
Results of different buddy mining tactics ($d$ = 10) on Douban.

| Methods | Douban 70% | | Douban 50% | |
|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE |
| SSR | 0.7617 ± 0.0013 | 0.5958 ± 0.0011 | 0.7778 ± 0.0009 | 0.6149 ± 0.0010 |
| BSSR_Ran | 0.7602 ± 0.0027 | 0.5966 ± 0.0019 | 0.7752 ± 0.0036 | 0.6172 ± 0.0025 |
| BSSR_Sim | 0.7554 ± 0.0014 | 0.5934 ± 0.0013 | 0.7695 ± 0.0011 | 0.6115 ± 0.0010 |
| BSSR_SIP | **0.7436 ± 0.0016** | **0.5852 ± 0.0012** | **0.7583 ± 0.0013** | **0.5990 ± 0.0011** |

**Table 5**
Results of different buddy mining tactics ($d$ = 10) on Epinions.

| Methods | Epinions 70% | | Epinions 50% | |
|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE |
| SSR | 1.0710 ± 0.0011 | 0.8297 ± 0.0010 | 1.0865 ± 0.0009 | 0.8442 ± 0.0007 |
| BSSR_Ran | 1.0725 ± 0.0033 | 0.8266 ± 0.0029 | 1.0855 ± 0.0026 | 0.8427 ± 0.0023 |
| BSSR_Sim | 1.0673 ± 0.0014 | 0.8232 ± 0.0011 | 1.0804 ± 0.0010 | 0.8392 ± 0.0006 |
| BSSR_SIP | **1.0525 ± 0.0015** | **0.8066 ± 0.0011** | **1.0663 ± 0.0014** | **0.8249 ± 0.0007** |

weight is given to the long-term influence, it may pollute the other factors which affect user decisions and in turn harm the recommendation performance.

### 6.6. Probe on different buddy mining tactics

As showing in above experiments, a key reason for the performance improvement of BSSR is utilizing the buddies to guide the incorporation of social relations into recommendation. Here one intuitive question is that do we really find the right buddies to get better results? To examine this, we conduct an experimental analysis on different tactics to find each user's buddies. We are especially interested in two cases: (1) What if we randomly select buddies for our BSSR model? (2) What if we select buddies just by the rating similarity between users?

We evaluate different buddy mining tactics on our BSSR model and the comparison results are shown in Tables 4 and 5. In the two tables, BSSR_Ran means to randomly select some friends as the buddies, BSSR_Sim means to select the buddies as the friends who have high rating similarities to users, and BSSR_SIP means to detect the buddies by Social Influence Propagation (SIP) method described in Section 3. To make a fair comparison, we use the same susceptibility $S_i$ detected by SIP in three methods. Comparing with the SSR which considers all the friends' influence, the BSSR_Ran gets comparable performance, which means that not all friends have effects on users' decisions and it is unnecessary to consider all of their influence when recommendation. On the other hand, BSSR_Ran and BSSR_Sim do not get much better performance than SSR, which means these two easy tactics do not find the right buddies. It is obvious that BSSR_SIP outperforms the other two tactics, which demonstrates the effectiveness of our buddy mining method.

## 7. Conclusion and future work

In this paper, we propose a novel social recommendation framework, BSSR, to incorporate the influence of social relations into recommendation more properly by the guidance of mining buddies and susceptibility. First, to detect each user's buddies and susceptibility, the Social Influence Propagation (SIP) method is proposed for social relation analysis by passing two kinds of influence related messages. Then we proposed the social influence based recommendation model to generate the final prediction, which incorporates the influence of social relationship by the guidance of the mined buddies and susceptibility. Experimental results on two datasets from real-world social networks have demonstrated that the proposed method produces better recommendation results than other competitors.

Our future work will involve further exploration of social relation analysis for recommendation. In particular, we could involve more information into our model, such as the domain information and the time context. We would try to take the domain-specific social relation analysis to detect each users' buddy set and susceptibility on each domain, and to do recommendation by considering the domain-specific behavior. Additionally, the time context may also change user's behavior on social network. For example, an popular events may drive the users' behaviors as well as the link strength between users. How to model the time context in social recommendation is an interest topic we can discuss in the future.

## Appendix A

Here, we present the derivation of the updating rule for $r$ and $a$ in Eqs. (7)–(9). As shown in Eq. (6), we have the following update rule for $\varphi_{i\to k}(x_i)$ (message from $x_i$ to $g_{u_k}$) and $\psi_{i\leftarrow k}(x_i)$ (message from $g_{u_k}$ to $x_i$).

$$\varphi_{i\to k}(x_i) = \log l_{u_i}(x_i) + \sum_{f\neq k}\psi_{i\leftarrow f}(x_i)$$

$$\psi_{i\leftarrow k}(x_i) = \max_{\sim\{x_i\}}\left[\log g_{u_k}(x_1,\ldots,x_N) + \sum_{f\neq i}\varphi_{f\to k}(x_f)\right]$$

According to Eq. (4), we have:

$$\log g_{u_i}(x_1,\ldots,x_N) = \begin{cases} -\infty & \text{if } x_i = u_i, \text{ and } x_f \neq u_i \text{ for all } f \neq i \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

We substitute Eq. (21) into Eq. (6) and obtain:

$$\psi_{i\leftarrow k}(x_i) = \max_{(x_k=u_k)\wedge(\exists f\neq k, x_f=u_k)\wedge(x_k\neq u_k)}^{\sim\{x_i\}} \sum_{f\neq i}\varphi_{f\to k}(x_f)$$
$$= \begin{cases} \sum_{f\neq i}\max_{x_f}\varphi_{f\to k}(x_f) + \max_{f\neq i}(\varphi_{f\to k}(u_k) \\ \quad -\max_{x_f}\varphi_{f\to k}(x_f)); & u_i = u_k = x_i \\ \sum_{f\neq i}\max_{x_f}\varphi_{f\to k}(x_f); & u_i = u_k \neq x_i \\ \sum_{f\neq i}\max_{x_f}\varphi_{f\to k}(x_f); & u_i \neq u_k = x_i \\ \sum_{f\neq i}\max_{x_f}\varphi_{f\to k}(x_f) + \max\{\varphi_{k\to k}(u_k) - \max_{x_k}\varphi_{k\to k}(x_k) \\ \quad +\max_{f\neq k,i}(\varphi_{f\to k}(u_k) - \max_{x_f}\varphi_{f\to k}(x_f)), \\ \max_{x_k\neq u_k}\varphi_{k\to k}(x_k) - \max_{x_k}\varphi_{k\to k}(x_k)\}; & u_i \neq u_k \neq x_i \end{cases} \tag{22}$$

Inspired by the derivation of affinity propagation, we can view $\varphi_{i\to k}(x_i)$ and $\psi_{i\leftarrow k}(x_i)$ as the sum of two parts: one is variable relying on $x_i$ and the other is constant. That is:

$$\varphi_{i\to k}(x_i) = \varphi^1_{i\to k}(x_i) + \varphi^2_{i\to k} \tag{23}$$

$$\psi_{i\to k}(x_i) = \psi^1_{i\to k}(x_i) + \psi^2_{i\to k} \tag{24}$$

where we let

$$\varphi^2_{i\to k} = \max_{x_i \neq u_k} \varphi_{i\to k}(x_i) \tag{25}$$

$$\psi^2_{i\to k} = \psi_{i\to k}(x_i \neq u_k) \tag{26}$$

Thus we have:

$$\begin{aligned}\varphi^1_{i\to k}(u_k) &= \varphi_{i\to k}(u_k) - \varphi^2_{i\to k} = \log l_{u_i}(u_k) + \sum_{f\neq k}\psi_{i\leftarrow f}(u_k)\\ &\quad - \max_{u_f\neq u_k}[\log l_{u_i}(u_f) + \sum_{j\neq k}\psi_{i\leftarrow j}(u_f)] = \log l_{u_i}(u_k)\\ &\quad + \sum_{f\neq k}\psi^2_{i\leftarrow f} - \max_{u_f\neq u_k}[\log l_{u_i}(u_f) + \psi^1_{i\leftarrow f}(u_f) + \sum_{j\neq k}\psi^2_{i\leftarrow j}]\\ &= \log l_{u_i}(u_k) - \max_{u_f\neq u_k}[\log l_{u_i}(u_f) + \psi^1_{i\leftarrow f}(u_f)]\end{aligned} \tag{27}$$

Additionally, from Eqs. (25) and (23) we can have:

$$\begin{aligned}\max_{x_i} \varphi_{i\to k}(x_i) &= \max\{\varphi^2_{i\to k}, \varphi_{i\to k}(u_k)\}\\ &= \varphi^2_{i\to k} + \max\{0, \varphi^1_{i\to k}(u_k)\}\end{aligned} \tag{28}$$

With it we can get following:

$$\begin{aligned}\psi^1_{i\to k}(u_k)|_{u_i=u_k} &= \psi_{i\to k}(u_k) - \psi_{i\to k}(x_i\neq u_k)|_{u_i=u_k} \text{ (from Eq.(24) Eq.(26))}\\ &= \max_{f\neq i}(\varphi_{f\to k}(u_k) - \max_{x_f}\varphi_{f\to k}(x_f)) \text{ (from Eq.(22))}\\ &= \max_{f\neq i}\varphi^1_{f\to k}(u_k) - \max\{0, \varphi^1_{f\to k}(u_k)\}\\ &\quad \times \text{(from Eqs. (23) and (28))} = \max_{f\neq i}\min\{\varphi^1_{f\to k}(u_k), 0\}\end{aligned} \tag{29}$$

similarly, we have:

$$\begin{aligned}\psi^1_{i\to k}(u_k)|_{u_i\neq u_k} &= \psi_{i\to k}(u_k) - \psi_{i\to k}(x_i\neq u_k)|_{u_i\neq u_k}\\ &= -\max\{\varphi_{k\to k}(u_k) - \max_{x_k}\varphi_{k\to k}(x_k) + \max_{f\neq k,i}(\varphi_{f\to k}(u_k)\\ &\quad - \max_{x_f}\varphi_{f\to k}(x_f)), \max_{x_k\neq u_k}\varphi_{k\to k}(x_k) - \max_{x_k}\varphi_{k\to k}(x_k)\}\\ &= -\max\{\min\{\varphi^1_{k\to k}(u_k), 0\} + \max_{f\neq k,i}\min\{\varphi^1_{f\to k}(u_k), 0\},\\ &\quad - \max\{\varphi^1_{k\to k}(u_k), 0\}\} = \min\{-\min\{\varphi^1_{k\to k}(u_k), 0\}\\ &\quad - \max_{f\neq k,i}\min\{\varphi^1_{f\to k}(u_k), 0\}, \max\{\varphi^1_{k\to k}(u_k), 0\}\}\end{aligned} \tag{30}$$

let $r_{ik} = \varphi^1_{i\to k}(u_k)$ and $a_{ik} = \psi^1_{i\to k}(u_k)$, we can have Eqs. (28)–(30) as:

$$r_{ik} = \log l_{u_i}(u_k) - \max_{u_f\neq u_k}\{a_{if} + \log l_{u_i}(u_f)\} \tag{31}$$

$$a_{kk} = \max_{u_f\neq u_i}\min(r_{fk}, 0) \tag{32}$$

$$a_{ik} = \min\{-\min(r_{kk}, 0) - \max_{u_f\neq u_k, u_i}\min(r_{fk}, 0), \max(r_{kk}, 0)\} \tag{33}$$

which is equal to Eqs. (7)–(9), considering that link exists only between neighbors.

Note that $a_{ik} + r_{ik} < a_{ij} + r_{ij}$ indicates $\psi_{i\leftarrow k}(u_k) + \varphi_{i\to k}(u_k) < \psi_{i\leftarrow j}(u_j) + \varphi_{i\to j}(u_j)$, thus, the new update rules for $a$ and $r$ in Eqs. (7)–(9) can be used to find the optimal value instead of the messages update rule in Eq. (6).

## References

[1] M. McPherson, L. Smith-Lovin, J.M. Cook, Birds of a feather: homophily in social networks, Annu. Rev. Sociol. (2001) 415–444.
[2] Rod Bond, P.B. Smith, Culture and conformity: a meta-analysis of studies using Asch's (1952b, 1956) line judgment task, Psychol. Bull. 119 (1996) 111–137.
[3] M.J. Salganik, P.S. Dodds, D.J. Watts, Experimental study of inequality and unpredictability in an artificial cultural market, Science 311 (5762) (2006) 854–856.
[4] C.-M. Au Yeung, T. Iwata, Strength of social influence in trust networks in product review sites, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, 2011, pp. 495–504.
[5] S. Aral, D. Walker, Identifying influential and susceptible members of social networks, Science 337 (6092) (2012).
[6] F.R. Kschischang, B.J. Frey, H.-A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Trans. Inf. Theory 47 (1998) 498–519.
[7] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.
[8] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 230–237.
[9] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, 2008.
[10] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–434.
[11] J.D.M. Rennie, N. Srebro, Fast maximum margin matrix factorization for collaborative prediction, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 713–719.
[12] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. (2009) 4:2–4:2.
[13] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. (2005) 734–749.
[14] L. Zhen, G.Q. Huang, Z. Jiang, Recommender system based on workflow, Decis. Support Syst. 48 (1) (2009) 237–245.
[15] L. Zhen, Z. Jiang, H. Song, Distributed recommender for peer-to-peer knowledge sharing, Inf. Sci. 180 (18) (2010) 3546–3561.
[16] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.
[17] R. Jin, J.Y. Chai, L. Si, An automatic weighting scheme for collaborative filtering, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004, pp. 337–344.
[18] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Trans. Inf. Syst. (TOIS) 22 (2004) 143–177.
[19] G. Linden, B. Smith, J. York, Amazon. com recommendations: item-to-item collaborative filtering, Internet Comput. IEEE 7 (2003) 76–80.
[20] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 89–115.
[21] L. Si, R. Jin, Flexible mixture model for collaborative filtering, in: ICML, 2003, pp. 704–711.
[22] A. Kohrs, B. Merialdo, Clustering for collaborative filtering applications, in: Computational Intelligence for Modelling, Control & Automation. IOS, 1999.
[23] M. OConnor, J. Herlocker, Clustering items for collaborative filtering, in: Proceedings of the ACM SIGIR Workshop on Recommender Systems, 1999.
[24] R. Jin, L. Si, C. Zhai, Preference-based graphic models for collaborative filtering, in: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, UAI'03, 2003, pp. 329–336.
[25] M. Harvey, M.J. Carman, I. Ruthven, F. Crestani, Bayesian latent variable models for collaborative item rating prediction, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 699–708.
[26] Y. Zhang, J. Koren, Efficient Bayesian hierarchical user modeling for recommendation system, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 47–54.
[27] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[28] P. Massa, P. Avesani, Trust-aware recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems, 2007, pp. 17–24.
[29] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: Proceedings of the Fourth ACM Conference on Recommender Systems, 2010, pp. 135–142.
[30] H. Ma, H. Yang, M.R. Lyu, 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.
[31] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, 2011, pp. 287–296.
[32] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, H. Zha, Like like alike: joint friendship and interest propagation in social networks, in: Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 537–546.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005

[33] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, L. Schmidt-Thieme, Multi-relational matrix factorization using bayesian personalized ranking for social network data, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, 2012, pp. 173–182.
[34] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 397–406.
[35] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, pp. 195–202.
[36] H. Tong, C. Faloutsos, J.-Y. Pan, Fast random walk with restart and its applications, in: ICDM, 2006.
[37] Q. Yuan, L. Chen, S. Zhao, Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, 2011, pp. 245–252.
[38] H. Ma, I. King, M.R. Lyu, Learning to recommend with explicit and implicit social relations, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 29.
[39] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, S. Yang, Social contextual recommendation, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM, 2012, pp. 45–54.
[40] X. Yang, H. Steck, Y. Liu, Circle-based recommendation in online social networks, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 1267–1275.
[41] E. Cambria, H. Wang, B. White, Guest editorial: big social data analysis, Knowl.-Based Syst. (69) (2014) 1–2.
[42] A. Saulwick, K. Trentelman, Towards a formal semantics of social influence, Knowl.-Based Syst. 71 (2014) 52–60.
[43] J. Wu, F. Chiclana, A social network analysis trust–consensus based approach to group decision-making problems with interval-valued fuzzy reciprocal preference relations, Knowl.-Based Syst. 59 (2014) 97–107.
[44] F. Chen, K. Li, Detecting hierarchical structure of community members in social networks, Knowl.-Based Syst. (2015).
[45] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 641–650.
[46] J. Ye, H. Cheng, Z. Zhu, M. Chen, Predicting positive and negative links in signed social networks by transfer learning, in: Proceedings of the 22Nd International Conference on World Wide Web, 2013, pp. 1477–1488.
[47] R. Schifanella, A. Barrat, C. Cattuto, B. Markines, F. Menczer, Folks in folksonomies: social link prediction from shared metadata, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, 2010, pp. 271–280.
[48] R. Xiang, J. Neville, M. Rogati, Modeling relationship strength in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, 2010, pp. 981–990.
[49] S.-H. Yang, A.J. Smola, B. Long, H. Zha, Y. Chang, Friend or frenemy?: predicting signed ties in social networks, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2012, pp. 555–564.
[50] I. Kahanda, J. Neville, Using transactional information to predict link strength in online social networks, in: ICWSM, 2009.
[51] F. Bravo-Marquez, M. Mendoza, B. Poblete, Meta-level sentiment models for big social data analysis, Knowl.-Based Syst. 69 (2014) 86–99.
[52] J. Bernabé-Moreno, A. Tejeda-Lorente, C. Porcel, H. Fujita, E. Herrera-Viedma, CARESOME: a system to enrich marketing customers acquisition and retention campaigns using social media information, Knowl.-Based Syst. 80 (2015) 163–179.
[53] G. Razis, I. Anagnostopoulos, Influencetracker: rating the impact of a twitter account, in: Artificial Intelligence Applications and Innovations, Springer, 2014, pp. 184–195.
[54] J. Tang, J. Sun, C. Wang, Z. Yang, Social influence analysis in large-scale networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 807–816.
[55] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972–976.
[56] B. Frey, D. Dueck, Mixture modeling by affinity propagation, Adv. Neural Inf. Process. Syst. 18 (2006) 379.

Please cite this article in press as: T. Yuan et al., How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.08.005