

AN EFFECTIVE FRIEND RECOMMENDATION METHOD USING LEARNING TO RANK AND SOCIAL INFLUENCE

Shun-Yuan Shih, College of Management, National Taiwan University, Taipei, Taiwan,
R.O.C., r02725014@ntu.edu.tw

Meng Lee, College of Management, National Taiwan University, Taipei, Taiwan, R.O.C.,
r02725032@ntu.edu.tw

Chien Chin Chen, College of Management, National Taiwan University, Taipei, Taiwan,
patonchen@ntu.edu.tw

Abstract

Social network sites have become an important medium for people to receive information anytime anywhere. Users of social network sites share information by posting updates. The updates shared by friends form social update streams that provide people with up-to-date information. To receive novel information, users of social network sites are encouraged to establish social relations. However, having too many friends can lead to an information overload problem causing users to be overwhelmed by the huge number of updates shared continuously by numerous friends. The information overload problem can result in bad user experiences. It may also affect user intentions to join social network sites and thereby possibly reduce the sites' advertising earnings which are based on the number of users. To resolve this problem, there is an urgent need of effective friend recommendation methods. A user is considered as a valuable friend if people like the updates the user posts. In this paper, we propose a model-based recommendation method which suggests valuable friends to users. Techniques of matrix factorization and learning to rank are designed to model the latent preferences of users and updates. At the same time, social influence is incorporated into the proposed method to enhance the learned preferences. Valuable friends are recommended if the preferences of the updates that they share are highly associated with the preferences of a target user. Our experiment findings that are based on a huge real-world dataset demonstrate the effectiveness of the social influence and learning to rank on a friend recommendation task. The results show that the proposed method is effective and it outperforms many well-known friend recommendation methods in terms of the coverage rate and ranking performance.

Keywords: Recommendation Systems, Learning to Rank, Social Influence, Matrix Factorization.

1 INTRODUCTION

Due to the recent rapid advances in ICT, many internet services have been developed to facilitate information exchange. Among them, social network sites such as Facebook have become increasingly popular, with estimates of over 60% of adults in the U.S.¹ having more than one social network site account. The most popular social network site worldwide is Facebook, which as of 2014 had billions of active users². There is a great deal of evidence showing that social network sites are ubiquitous and have become a part of our daily life. Users on social network sites exchange information by sharing updates using posts, photos, or videos. These updates form *social update streams*, which are sets of chronologically ordered updates shared by users' friends (Hong et al. 2012), that enable users to get the latest information. When users share an update, the update will instantaneously appear in the social update streams of their friends. Due to their varied content and efficient delivery, social update streams have gradually taken the place of traditional media and are becoming an important information dissemination mechanism (Benevenuto et al. 2009; Kwak et al. 2010).

Social network site users need to establish friendships to receive updates. However, when the quantity of friends reaches a fairly high level, users can be overwhelmed by the amount of fresh social updates. The thousands of social updates from hundreds of friends per day may be seen as a kind of spam in the social update streams. The so-called *information overload* problem (Koroleva et al. 2010) that may lead to a bad user experience which in turn may affect user intention to join social network sites. And since a major source of income for social network sites comes from advertising based on the number of site users (Enders et al. 2008), the information overload problem may affect the revenues of the sites. To resolve this problem, we here propose a novel friend recommendation method which suggests valuable friends to users. The function of these recommended friends is a filter that excludes updates irrelevant to the user's interests. The resulting social update streams are thus clearer and more concise.

Existing friend recommendation methods focus on discovering the real-world friends of users on a social site, with the task being comparable to the link prediction of social networks (Hopcroft et al. 2011). These methods analyze structures of social networks (e.g., friends of friends) to predict potential links (i.e., friendships) between users. Here, we design a model-based friend recommendation method which employs *learning to rank* (Hong et al. 2012; Shi et al. 2012) to recommend valuable friends to users. Recent recommendation studies (e.g., Chen et al. 2012; Liu 2009; Shi et al. 2012) advocate learning to rank to incorporate users' implicit feedbacks with the recommendation algorithms. Instead of measuring the preference degree (e.g., rating) of an item, learning to rank utilizes the implicit feedbacks to train a ranking model which discriminates preferences between items. In the proposed method, we consider social updates as items and integrate techniques of matrix factorization with learning to rank in order to learn the latent preferences of users and updates. In addition to this, social influence (the association of friends and their preferences) is incorporated to derive representative latent preferences. Thereafter, valuable friends are recommended if the preferences of the updates they share are highly associated with the preferences of a target user. To examine the proposed method, we adopted a real-world dataset consisting of 277,440 updates shared by 9,981 users. The experiment results based on this large dataset demonstrate the effectiveness of the proposed method in recommending valuable friends; further, the updates shared by the recommended friends were shown to be highly associated with user preferences. To the best of our knowledge, this is the first study on incorporating the information of social ties into the learning-to-rank recommendation methods and the proposed method thus outperforms many well-known friend recommendation methods and learning-to-rank recommendation methods.

The remainder of this paper is organized as follows. The next section contains a review of related works on friend recommendation approaches. We introduce the proposed friend

1 PewResearch (2014). "Social Networking Fact Sheet." From <http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/>.

2 statista (2014). "Global social networks ranked by number of users 2014." From <http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>.

recommendation method in Section 3, and then evaluate it in Section 4. Finally, in Section 5, we provide some concluding remarks and consider future avenues of research.

2 RELATED WORKS

Friend recommendation is often formulated as the link prediction problem which was first introduced by Liben-Nowell and Kleinberg (2007). Basically, the task of link prediction is to predict potential links between nodes in a social network. Aspects such as social psychology theorems and entity matching across different networks were intensively studied to infer social ties. For instance, Liben-Nowell and Kleinberg examined various factors including node neighbourhoods, paths in a network, and unsupervised clustering approaches for link prediction. The prediction task, however, is so difficult that the best prediction accuracy showed in their study was merely 16%. Leskovec et al. (2010) supposed that social networks involve positive (friendly) and negative (opposing) relationships, and employed a logistic regression model to predict positive and negative links in online social networks. The authors incorporated balance theories of social psychology into the prediction model and demonstrated that negative links are useful in predicting positive relationships. Hopcroft et al. (2011) investigated the formation of reciprocal relationships in the dynamic network Twitter. They examined factors of geographic distances, homophily of users, implicit networks, and social balance theories to predict the “follow backs” among Twitter users. Their experiments showed the effects of the aforementioned factors and they concluded that users usually make follow back decisions within 10 days. Zhang et al. (2014) studied the multi-network link prediction problem which focuses on forming social links across different aligned networks, such as friendship networks and location checkin networks. The authors explored the social meta path which is the weighted path that connects two nodes in different networks. Their experiment results demonstrate that heterogeneous features extracted from both intra- and inter-social meta paths enhance the link prediction significantly. Nevertheless, methods of link prediction hardly resolve the information overload of social update streams. This is because the methods focus on identifying new friends but ignore the fact that having too many friends can intensify the information overload problem. In contrast, our method is able to rank friends and thus identifies valuable friends as a means to sort social update streams.

Our research is also related to recommendation systems (Shi et al. 2012; Wan et al. 2013). Methods of recommendation systems examine users’ rating logs to find out the preferences of users and provide personalized item recommendations according to the identified user preferences. Collaborative filtering is a widely used recommendation approach. The approach is based on the assumption that like-minded people prefer similar items and thus utilize the rating logs to identify reference users whose preferences are similar to those of the target user. Items that interest the reference users then are recommended to the target user. It is noteworthy that collaborative filtering has achieved remarkable performance in several recommendation contests, such as the Netflix Prize competition and KDDCUP (Bennett and Lanning 2007; Koren et al. 2009). Recently, a novel collaborative filtering implementation called the latent factor model (Koren et al. 2009) has been developed and has attracted considerable attention from researchers. Techniques of the model are normally based on matrix factorization, the core of which is to model the latent preferences of users and items as Z dimensional preference vectors that approximate the user-item rating matrix. For instance, Sarwar et al. (2000) applied the singular value decomposition into the user-item rating matrix to identify the latent preferences of users. The identified preferences are able to discover reliable reference users for effective item recommendations. Koren et al. (2009) conducted a thorough analysis of matrix factorization techniques and formulated matrix factorization as an optimization problem, thereby introducing a gradient decent-based learning method to rapidly approximate adequate preference vectors. The method achieves remarkable performance on many recommendation datasets and is the state of the art matrix factorization method.

Traditional recommendation methods normally formulated the preference approximation as an optimization problem whose goal is to predict item ratings as correct as possible. For instance, Koren et al. (2009) approximated user/item preferences by minimizing the root mean square error between the actual item ratings and the ratings predicted by the preferences. Pessiot et al. (2007), however, argued that recommendation should be a ranking problem because the essential of recommendation is to rank

items according to user preferences. Moreover, another drawback of the minimization approach is that it is inefficient when there are insufficient rating records. To resolve this rating sparseness problem, techniques of learning to rank (Rendle et al. 2009) utilize the implicit feedbacks of users and have been the subject of active research in many research fields, such as in information retrieval and recommendation systems (Chen et al. 2012; Liu 2009; Shi et al. 2012). Basically, these techniques utilize implicit user feedbacks to optimize the user preferences. Chen et al. (2012), for example, assumed the tweets retweeted by users have a high precedence and developed a novel model to learn the precedence of tweets over users. The experiment results demonstrated that the model was effective in recommending useful tweets for users. Social influence is another useful recommendation technique since decisions of users are generally influenced by friends or trusted people (Ma et al. 2011; Shen and Jin 2012; Ye et al. 2012). To adopt social influence, many recommendation methods (e.g., Ma et al. 2009; Ma et al. 2008; Yang et al. 2012) aggregate the preferences of friends to recommend interesting items to users. In this paper, we also incorporate social influence into the proposed friend recommendation method. Unlike existing works which generally consider preference-similar users as friends, we utilize actual social relationships that have been proven effective in collaborative filtering (Yang et al. 2012).

3 FRIEND RECOMMENDATION USING SOCIAL UPDATE STREAMS

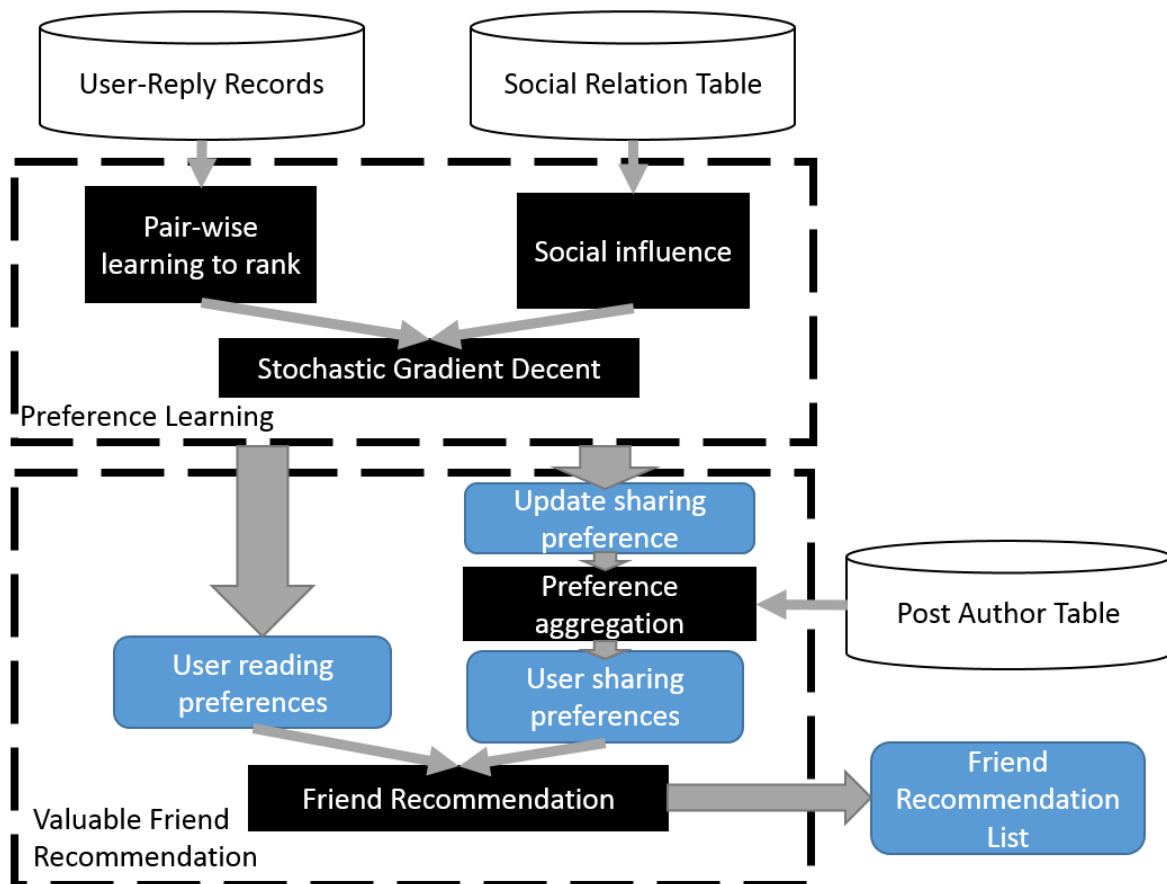


Figure 1. The System Structure.

Figure 1 depicts our model-based friend recommendation method consisting of two major components: *preference learning* and *valuable friendship recommendation*. In the preference learning stage, implicit user feedbacks, such as replies or likes, on social updates are collected. The feedbacks are fed into a

pairwise learning to rank model to learn two types of preferences that affect the precedence of an update in a pair. One relates to users' reading preferences and the other to the sharing preferences from updates. Furthermore, the social influence between users is incorporated into the learning model to enhance the learned preferences. In the valuable friend recommendation stage, a user's sharing preferences are constructed by aggregating the preferences of the updates shared by the user. Next, the similarities between users' reading and sharing preferences are computed and users are classified as valuable friends if their sharing preferences are highly associated (similar) to the reading preferences of a target user. We discuss each component in detail in the following sub-sections.

3.1 Preference Learning

Our preference learning incorporates techniques of learning to rank and social influence into the latent factor model which has been shown to be effective in many recommendation scenarios (Hong et al. 2012). The latent factor model, also known as matrix factorization (Koren et al. 2009), decomposes a user-item matrix to discover the preferences of users and items. In our method, let $U = \{u_1, u_2, \dots, u_M\}$ be a set of users on a social network site and let items $V = \{v_1, v_2, \dots, v_N\}$ be the updates shared by U . The user-item matrix R is an $M \times N$ matrix where the entry r_{ij} is 1 if user u_i has provided a feedback (e.g., like or reply) on update v_j ; otherwise, it is 0. The goal of matrix factorization is to search for matrices P and Q such that

$$r_{ij} \approx \hat{r}_{ij} = \underline{p}_i^T \underline{q}_j, \quad (1)$$

where \hat{r}_{ij} represents the estimation of r_{ij} ; \underline{p}_i and \underline{q}_j are the i th and j th columns of P and Q , respectively. The matrix P is an $Z \times M$ matrix where each column $\underline{p}_i \in \mathbb{R}^Z$ represents u_i 's reading preference vector and the dimension of the preferences is Z . Similarly, Q is a $Z \times N$ matrix where each column $\underline{q}_j \in \mathbb{R}^Z$ is v_j 's sharing preference vector. Since the preferences of users will be affected by those of their friends (Ma et al. 2011; Shen and Jin 2012; Yang et al. 2012; Ye et al. 2012), we therefore modify the definition of \hat{r}_{ij} by incorporating the social influence as follows:

$$\hat{r}_{ij} = (1 - \alpha) \underline{p}_i^T \underline{q}_j + \alpha \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_j, \quad (2)$$

where $F(u_i)$ denotes the set of u_i 's friends and s_{ig} stands for the u_g 's influence degree for users u_i . We adopted (Ma et al. 2008)'s social measure which computes the association of users in a social network as follows:

$$s_{ig} = \sqrt{\frac{d_g^-}{d_g^- + d_i^+}}, \quad (3)$$

where d_i^+ indicates the out-degree of u_i and d_g^- is the in-degree of u_g in the social network. The influence value s_{ig} decreases if u_i makes a lot of friends; however, s_{ig} increases if u_g is followed by a lot of users, that is, u_g is a popular and influential user. So, u_g 's opinions are likely to affect u_i . The second term of Eq. (2) denotes the social influence weighted by α , which ranges from 0 to 1. In the experiment section, we will examine the parameter α and the effect of social influence.

Given the user-item matrix R , methods of the latent factor model are used to search for P and Q that minimize the root mean square between r_{ij} and \hat{r}_{ij} (Koren et al. 2009). More recently, recommendation research has started to advocate learning to rank, which strives to identify the P and Q that characterize the precedence (i.e., relative ordering) of items, instead of minimizing the root mean square error. As the goal of recommendation systems is to rank items according to user preferences, learning to rank closely corresponds with the goal and has been investigated in many recommendation studies (e.g., Hong et al. 2012; Shi et al. 2012).

We adopt the pairwise learning to rank (Liu 2009) that models the precedence of updates in terms of update pairs. Here, we assume that updates with user feedbacks have a higher precedence (preference) than those with no feedback. Based on this, we construct a set of precedence update pairs $D_i = \{\langle v_x, v_y \rangle \mid v_x \in V, v_y \in V, r_{ix} > r_{iy}\}$ for a user u_i and compute the sum of the logistic loss (Chen et al. 2012), which is the core of our preference learning, as follows:

$$\sum_{i=1}^M \sum_{\langle v_x, v_y \rangle \in D_i} \ln(1 + e^{-(\hat{r}_{ix} - \hat{r}_{iy})}). \quad (4)$$

Figure 2 shows the curve of the logistic loss. The trend in the curve shows the logistic loss decreases as the difference between \hat{r}_{ix} and \hat{r}_{iy} increases. In other words, our preference learning aims to derive user/item preferences that preserve the precedence of updates.

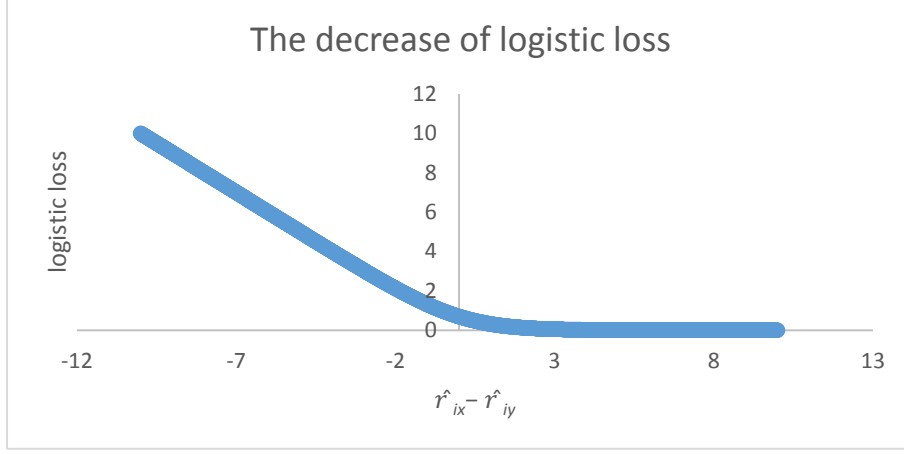


Figure 2. The curve of logistic loss.

By substituting \hat{r}_{ix} and \hat{r}_{iy} with Eq. (4), the goal of our preference learning is to find P and Q that minimize the following loss function F .

$$\begin{aligned} F(P, Q) &= \sum_{i=1}^M \sum_{\langle v_x, v_y \rangle \in D_i} \ln(1 + e^{-(\hat{r}_{ix} - \hat{r}_{iy})}) + \lambda(\|P\|^2 + \|Q\|^2) \\ &= \sum_{i=1}^M \sum_{\langle v_x, v_y \rangle \in D_i} \ln \left(1 + e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y))} \right) \\ &\quad + \lambda(\|P\|^2 + \|Q\|^2), \end{aligned} \quad (5)$$

where the first term is the sum of the logistic loss; $\|P\|^2$ and $\|Q\|^2$ are regularization terms that prevent the overfitting of the learned P and Q , and λ is the corresponding regularization coefficient. We adopt the stochastic gradient descent (Mitchell 1999) to search for adequate preference vectors. Specifically, the stochastic gradient descent first randomly initiates P and Q . Then, \underline{p}_i , \underline{q}_x and \underline{q}_y are iteratively refined by using their derivatives upon F defined below until F reaches a local minimum. Figure 3 illustrates the stochastic gradient descent algorithm.

$$\frac{\partial F}{\partial \underline{p}_i} = \frac{e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y)) \times (1-\alpha) \times (\underline{q}_x - \underline{q}_y)}}{(1 + e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y))})} + \lambda \underline{p}_i. \quad (6)$$

$$\frac{\partial F}{\partial \underline{q}_x} = \frac{\left(e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y)) \times ((1-\alpha) \underline{p}_i + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g))} \right)}{\left(1 + e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y))} \right)} + \lambda \underline{q}_x. \quad (7)$$

$$\frac{\partial F}{\partial \underline{q}_y} = \frac{\left(e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y)) \times (-(1-\alpha) \underline{p}_i - \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g))} \right)}{\left(1 + e^{-((1-\alpha) \times (\underline{p}_i^T \underline{q}_x - \underline{p}_i^T \underline{q}_y) + \alpha \times (\sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_x - \sum_{u_g \in F(u_i)} s_{ig} \underline{p}_g^T \underline{q}_y))} \right)} + \lambda \underline{q}_y. \quad (8)$$

Preference Learning

Input: The user-item matrix R with M users and N items (updates), learning rate γ , dimension of preferences Z .

Output: The learned preference vectors P, Q .

$t = 0$; // t is the iteration count

Initialize P^t and Q^t with random values.

while ($P^t \neq P^{t-1}$ && $Q^t \neq Q^{t-1}$) **do**

for $i = 1, 2, \dots, M$ **do**

 construct a set of precedence update pairs $D_i = \{ \langle v_x, v_y \rangle \mid v_x \in V, v_y \in V, r_{ix} > r_{iy} \}$ for a user u_i

for $\langle v_x, v_y \rangle \in D_i$ **do**

for $z = 1, 2, \dots, Z$ **do**

$$\underline{p}_z^{t+1} = \underline{p}_z^t + \gamma * \frac{\delta F}{\delta \underline{p}_z^t} \text{ based on Eq. (6);}$$

$$\underline{q}_x^{t+1} = \underline{q}_x^t + \gamma * \frac{\delta F}{\delta \underline{q}_x^t} \text{ based on Eq. (7);}$$

$$\underline{q}_y^{t+1} = \underline{q}_y^t + \gamma * \frac{\delta F}{\delta \underline{q}_y^t} \text{ based on Eq. (8);}$$

end

end

end

$t = t + 1$;

end

return $P = P^t, Q = Q^t$

Figure 3. The stochastic gradient descent for preference learning.

3.2 Valuable Friendship Recommendation

Once matrices P and Q are converged, we construct the sharing preference vector \underline{h}_j of user u_j by aggregating all the sharing preference vectors of the updates shared by u_j .

$$\underline{h}_j = \frac{\sum_{v_n \in W(u_j)} \underline{q}_n}{|\sum_{v_n \in W(u_j)} \underline{q}_n|}, \quad (9)$$

where $W(u_j)$ denotes the set of updates shared by u_j , and \underline{q}_n is the sharing preference vector of update v_n . The denominator of Eq. (9) is a normalization factor, which makes the sharing preference vector \underline{h}_j a length-normalized vector. For a target user u_i , a user u_j is deemed a valuable friend if u_j 's sharing preferences are highly similar to the reading preferences of u_i . We adopt the cosine metric to measure the preference similarity between u_i and u_j as follows:

$$\text{sim}(u_i, u_j) = \text{cosine}(\underline{p}_i, \underline{h}_j) = \frac{\underline{p}_i \cdot \underline{h}_j}{|\underline{p}_i| |\underline{h}_j|}. \quad (10)$$

The range of $\text{sim}(u_i, u_j)$ is within $[0,1]$, with the higher the value indicating the more similar the preferences of u_i and u_j . Finally, we rank users according to their cosine values and the top-ranked users are suggested as the valuable friends.

4 EXPERIMENT

4.1 Experiment Setup

4.1.1 Datasets

To evaluate the proposed method, we conducted experiments using the Weibo dataset of the WISE 2012 Challenge³. Weibo⁴ is the most popular social network site in China and its users share their updates by posting weibos, i.e., short messages. There are also follower-followee friendships between users that form a directed social network. Since the functions of Weibo are similar to those provided by Twitter, it has been labelled the Chinese Twitter in East Asian social circles. The Weibo dataset consists of 58,655,849 users and 366,946,149 updates, and is so big that it has been frequently used as a benchmark for big data analytics (Bae et al. 2014; Bao et al. 2013; Pan et al. 2013; Zhu et al. 2013). However, we noticed that the dataset consists of a lot of inactive users who have few social relationships and social interactions. In (Pan et al. 2013), the inactive users are regarded as noisy users, and thus to reduce their influence on system performance, we excluded from the dataset users with fewer than 150 followees. The final experiment dataset consists of 277,440 non-duplicated updates, 9,981 users, and 216,971 friendship links. Table 1 summarizes the experiment dataset.

Number of users	9981
Number of weibos	277440
Number of social links	216971
Number of replied records	104419
Average replies per user	10.46

Table 1. Statistics of the evaluation dataset

4.1.2 Experiment Procedure and Evaluation Metrics

We conducted two types of experiments to evaluate the performance of our valuable friend recommendation method. The first experiment treated followees as valuable friends and examined whether our method was able to recommend all the followees followed by users. We adopted the

³<http://www.wise2012.cs.ucy.ac.cy/challenge.html>

⁴<http://www.weibo.com>

conventional leave-one-out procedure (Chen et al. 2013) to evaluate our method and the evaluation metric was the *coverage rate @ K* (denoted as $C@K$) (Chen et al. 2013). Specifically, for each user, we evaluated the recommendation performance over multiple runs, with each run removing one followee from the user's followee list. Then, the leave-one-out procedure trained user/update preferences in an unbiased manner by deleting all implicit feedbacks made between the followee and the target user. Finally, the top K friends recommended by our method were examined to see if the recommendation contained the removed followee. The results of all the evaluation runs were averaged to obtain the overall recommendation performance. The coverage rate @ K was defined as follows:

$$C@K = \frac{|hit|}{|T|}, \quad (11)$$

where $|hit|$ represents the number of evaluation runs in which the top- K recommendation covers the removed followee, and $|T|$ denotes the total number of evaluation runs.

In Wan et al. (2013), a friend recommendation method was deemed effective if it was able to rank the users (called informational users hereafter) who frequently interacted with a target user higher in the generated friend recommendation list. In our second experiment, we considered the users whose shared updates had ever been replied by a target user as informational users. Note that an informational user here may not be a followee of the target user, and vice versa. Again, the leave-one-out evaluation procedure was followed to evaluate the ranking performance of our method. Specifically, for each target user u_i , we evaluated the ranking performance over multiple runs. Each run selected an informational user u_j of u_i for testing. Then, the implicit feedbacks made between u_i and u_j were deleted for unbiased preference learning. Next, the preference similarity (i.e., Eq. (10)) was used to rank all experiment users for u_i and the ranking position of u_j was saved in $rank_{ij}$. The ranking performance (denoted as RP_i) upon the target user u_i was evaluated by the following equation.

$$RP_i = \sum_{u_j \in \text{the informational users of } u_i} (IF_{ij} / rank_{ij}), \quad (12)$$

where IF_{ij} denotes the frequency of interactions (i.e., replies) made between u_i and u_j . The larger the value of RP_i , the better the ranking performance will be. Also, the ranking has the potential for resolving the information overload of social update streams by showing the updates shared by the top-ranked users. Finally, the RP s of all the target users were averaged to obtain the overall ranking performance.

It should be noted that Apache Mahout⁵, a Java-based machine learning library, was adopted to implement our stochastic gradient descent algorithm. The learning rate γ and regularized coefficient λ of the stochastic gradient descent was set at 1 and 0.01, respectively, as suggested by the library authors.

4.2 Parameter Settings

Before evaluating the recommendation performance, we first investigated the effect of system parameters on preference learning. There were two system parameters in our recommendation method, namely, Z and α . Figure 4 shows the effect of Z which designates the dimension of the latent preferences. Here, α is set at 0.1. Later, we examine its effect. In Figure 4, the x-coordinate is Z and the y-coordinate designates the learning loss (i.e., Eq. (5)) normalized by Z . We normalized the learning loss by Z in order to diminish the influence of the regularization terms when Z is large. The trend in the figure shows that a large Z generally produces a low loss. This is because a large Z is able to differentiate latent preferences. However, the improvement of Z over the loss is not significant when Z is larger than 600. And a large Z will also increase the length of preference learning. For these reasons, we set Z at 600 in the following experiments.

⁵<http://mahout.apache.org/>

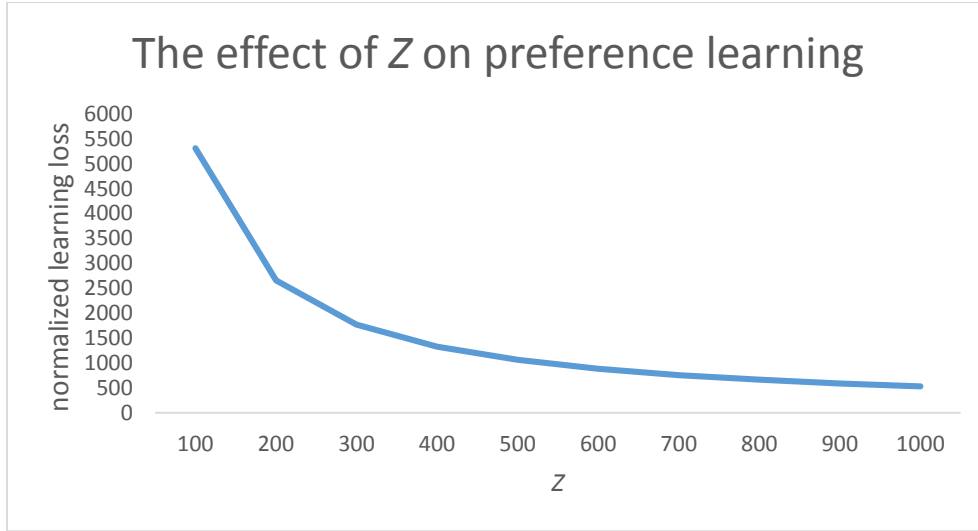


Figure 4. The effect of Z on the normalized learning loss.

As mentioned in Section 3.1, parameter α calibrates the social influence. Figure 5, illustrating α 's effect on preference learning, shows that setting α at 0.1 produces a low learning loss. We therefore used this setting in the following experiments. In the next section, we will examine the performance of the setting and compare the performance with and without using social influence (i.e., $\alpha = 0$).

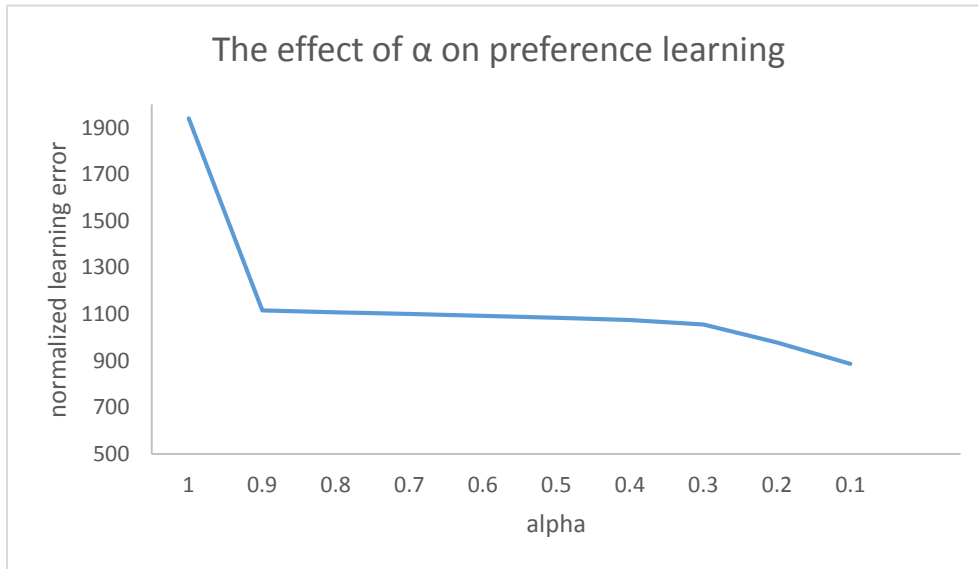


Figure 5. The effect of α on the normalized learning loss.

Lastly, we examined the impact of the above settings on the convergence of our stochastic gradient descent. As shown in Figure 6, the preference learning quickly converges within 10 iterations despite the fact that the evaluation dataset is huge. In other words, the proposed friend recommendation method is scalable and practicable for real-world social network sites.

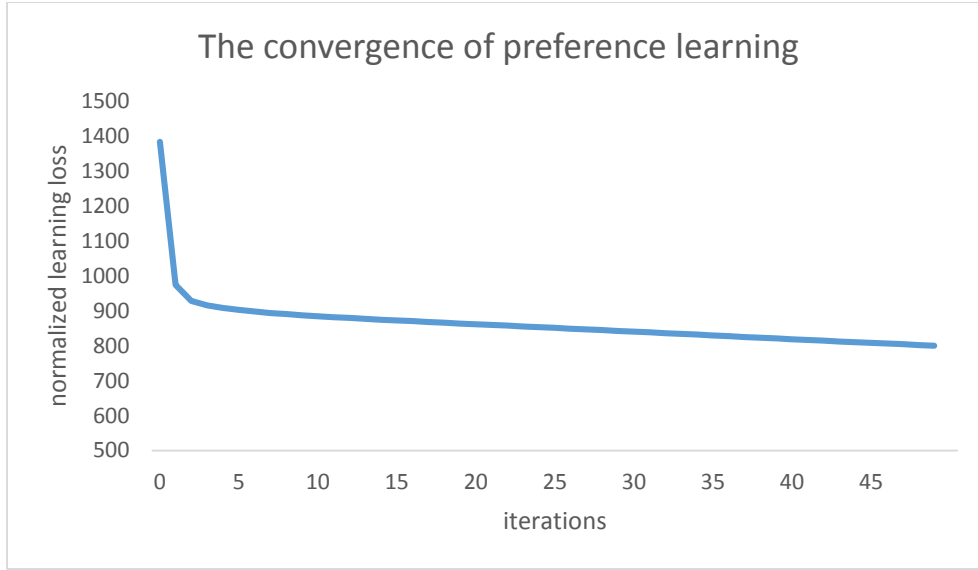


Figure 6. The convergence of preference learning.

4.3 Performance Comparison

In this section, we compare our method with five friend recommendation methods: FdFd (Wan et al. 2013), FIFd (Wan et al. 2013), KNN (Deshpande and Karypis 2004), SVD (Koren et al. 2009), and MF+LTR (Chen et al. 2012). We first show the performance of the methods in experiment 1, which examined the methods' ability in recommending the followees of users. Then, we discuss the ranking performance of the methods. To ensure the comparisons are fair, all the methods were evaluated by means of the leave-one-out evaluation procedure.

- FdFd (Friend-of-Friend): The method and the FIFd method are two baselines studied in (Wan et al. 2013). They analyzed social network structures to recommend friends to users. Here, FdFd recommends the followees of followees, normalized by their in-degrees in the social network to a target user.
- FIFd (Follower-of-Friend): The method analyzes the co-neighbouring degree between users. It recommends the users whose followees are highly overlapped with those of a target user.
- KNN (*K*-Nearest Neighbourhood): the method is a user-based collaborative filtering method. It utilizes the following Jaccard coefficient to measure the similarity between users.

$$\text{JaccardSim}(u_i, u_j) = \frac{|UR_i \cap UR_j|}{|UR_i \cup UR_j|}, \quad (13)$$

where UR_i stands for the set of updates replied by u_i , that is, $UR_i = \{v_x | v_x \in V, r_{ix} = 1\}$. The method treats similar users as friends and recommends the users to a target user.

- SVD (Singular Value Decomposition): Like our method, SVD is based on matrix factorization. However, the method trains user/item preferences by minimizing the root mean square error of the approximated user-item matrix.
- MF+LTR (Matrix Factorization with Learning To Rank): Both this method and our method incorporate pair-wise learning to rank into matrix factorization for preference learning. Nevertheless, MF+LTR does not examine the influence of social ties (i.e., social influence). Comparing the methods helps us comprehend the effect of social influence. It is noteworthy that MF+LTR has been considered the state-of-the-art recommendation method due to its superior recommendation performance (Chen et al. 2012).

Surprisingly, the coverage rates of the methods are all low, as can be seen in Figure 7. This is because the methods need to predict (recommend) the removed followee among 9,980 experiment users in each evaluation run, and the prediction task is not trivial. Nevertheless, our method achieved a superior performance and outperformed the compared methods. FdFd and FIFd had good coverage rates,

and the results show the friend-of-friend and co-neighbouring are representative social patterns. It is interesting to note that the performance of the SVD method was inferior. We observed that the user-item matrix R was very sparse, and for this reason, the root-mean-square-error minimization of SVD was obscured by the sparseness that deteriorated its performance. Similarly, because the KNN method also measured the user similarity in terms of the sparse user replies, its performance was inferior too. The coverage rates of MF+LTR were comparable to those of our method. However, it should be pointed out that both MF+LTR and our method are based on learning to rank. The results indicate that learning to rank handles the data sparseness problem better than the other methods do. The results also correspond with the findings of Rendle et al. (2009) who validated the idea that learning to rank is useful for resolving the sparseness problem of preference learning. Our method achieved around 30% improvement in coverage rate compared to the MF+LTR method. This improvement suggests that social influence is informative for friend recommendations. To further validate the effect of social influence, we compared the $C@10$ scores of the two methods upon the target users who had more than 50 followees. These target users represented the most social users in our dataset. Again, our method significantly outperformed MF+LTR insofar as the coverage rates of MF+LTR and our method were 0.029 and 0.068, respectively (133% improvement). Hence, it is clear that social influence is helpful.

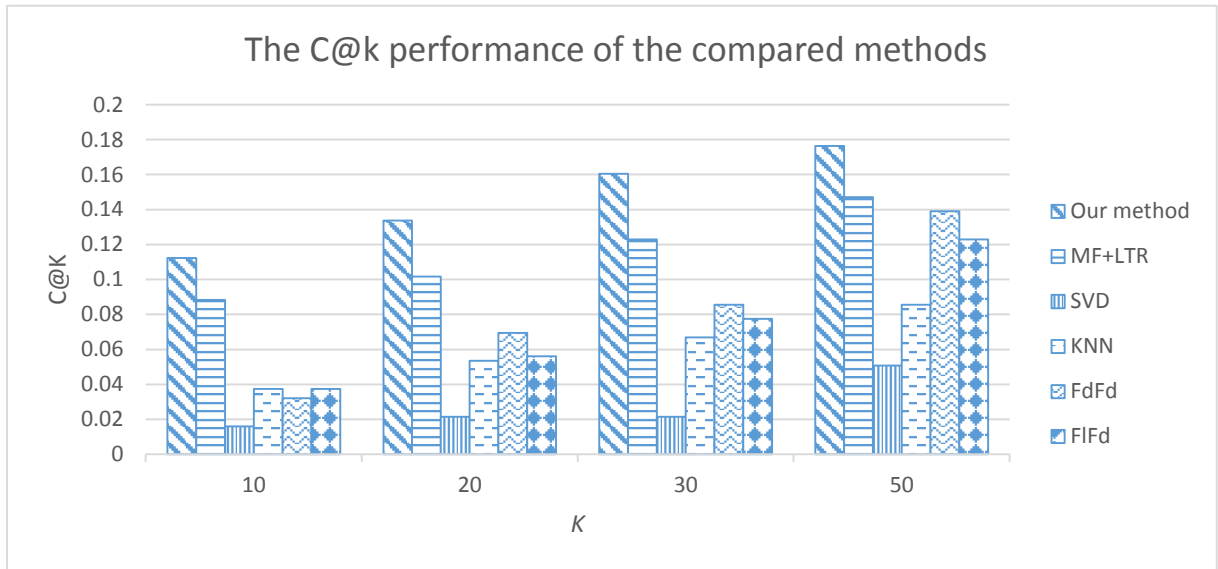


Figure 7. The $C@K$ performance of the compared methods.

Figure 8 illustrates the ranking performance of the compared methods. Similar to the previous experiment, the SVD method appears to be inferior. This may be because SVD focuses on the user-item matrix approximation which ignores the precedence of items. The learned preferences are thus incapable of ranking the informational users in a top position. While KNN's coverage rates were inferior, its ranking performance was surprisingly good. This is because the similarity (i.e., Eq. (13)) used by KNN is based on user interactions. The method therefore is good at predicting informational users. It is interesting to observe that FdFd and FIFd, which produced good coverage rates in the last experiment, were poor at recommending informational users. While the friend-of-friend and co-neighbouring are significant social relationships, they are hardly capable of inferring degrees of social interactions. As a consequence, these methods are inferior at recommending informational users. As in the last experiment, our method outperformed MF+LTR, and these two methods performed better than the other methods. The results again validate the value of learning to rank and social influence. In summary, our method achieved the best performance on the two experiments which focused on two different perspectives (predicting friends and suggesting informational users) of friend recommendations. The method can recommend not only friends but also those whose posts are informational to users. Since users prefer

the updates shared by the informational users, the proposed method is likely to alleviate the information overload of social update streams.

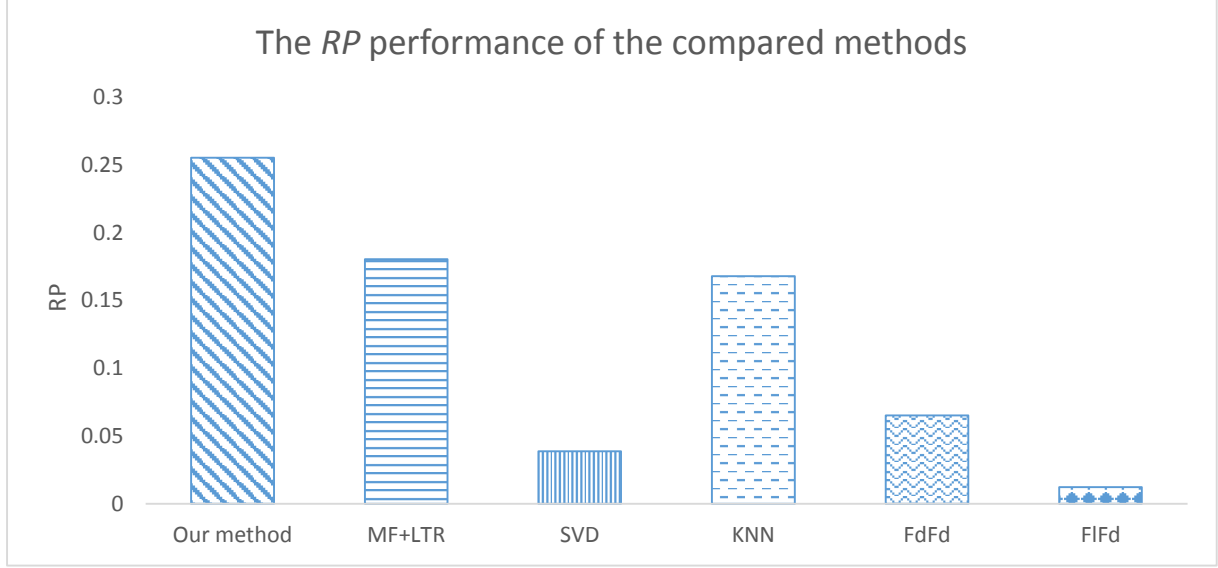


Figure 8. The RP performance of the compared methods.

5 CONCLUSION

Users of social network sites generally suffer from the information overload of social update streams, which can affect users' intentions to visit social network sites, and by extension, the sites' advertising revenues. Basically, given that social update streams are constituted by the updates shared by friends, to resolve the information overload problem and to create a win-win proposition for both users and site owners, recommending valuable friends to users is critical. In this paper, we have proposed an effective friend recommendation method which integrates learning to rank and matrix factorization to learn the preferences of users and updates. In addition, the social influence is incorporated into the proposed method to suggest valuable friends to users. The experiments based on a huge real-world dataset demonstrate that the proposed method is capable of overcoming the data sparseness of preference learning and the social influence is useful in the friend recommendation task. Consequently, the method achieves superior recommendation performance and outperforms many well-known friend recommendation methods.

In this paper, we make the shared updates equally important to construct a user's sharing preference vector. In future work, we plan to design an adaptive preference aggregation process, where representative updates of users will be identified to better characterize the sharing preferences of the users. We will also consider important factors (e.g., the number of friends per user) to customize the weight of the user's social influence, that is, the parameter α . While this paper focused on friend recommendations, the proposed method can be applied to various social recommendation tasks, such as recommending useful groups to social network users by considering social groups as items. We will also extend the method to different social recommendation domains and evaluate their recommendation performances.

ACKNOWLEDGMENT

This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 103-2221-E-002-106-MY2.

REFERENCE

- Bae, Y., Ryu, P. M., and Kim, H. (2014). Predicting the Lifespan and Retweet Times of Tweets Based on Multiple Feature Analysis. *ETRI Journal*, 36 (3), 418-428.
- Bao, P., Shen, H. W., Chen, W., and Cheng, X. Q. (2013). Cumulative effect in information diffusion: empirical study on a microblogging network. *PloS one*, 8 (10), e76027.
- Benevenuto, F., Rodrigues, T., Cha, M., and Almeida, V. (2009). Characterizing User Behavior in Online Social Networks. In *IMC '09 Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 49-62.
- Bennett, J., and Lanning, S. (2007). The Netflix Prize. In *Proceedings of the KDD Cup and Workshop*, 3-6.
- Chen, C. C., Wan, Y. H., Chung, M. C., and Sun, Y. C. (2013). An effective recommendation method for cold start new users using trust and distrust networks. *Information Sciences*, 224, 19-36.
- Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E., and Yu, Y. (2012). Collaborative Personalized Tweet Recommendation. In *SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 661-670.
- Deshpande, M., and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22 (1), 143-177.
- Enders, A., Hungenberg, H., Denker, H. P., and Mauch, S. (2008). The long tail of social networking.: Revenue models of social networking sites. *European Management Journal*, 26 (3), 199-211.
- Hong, L., Bekkerman, R., Adler, J., and Davison, B. D. (2012). Learning to Rank Social Update Streams In *SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 651-660.
- Hopcroft, J., Lou, T., and Tang, J. (2011). Who Will Follow You Back? Reciprocal Relationship Prediction. In *CIKM '11 Proceeding of the 20th ACM conference on Information and knowledge management*, 1137-1146.
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42 (8), 30-37.
- Koroleva, K., Krasnova, H., and Günther, O. (2010). 'STOP SPAMMING ME!'-Exploring information overload on facebook. In *Proceedings of the 16th Americas Conference on Information Systems (AMCIS)*, p. 447.
- Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter, a social network or a news media? In *WWW '10 Proceedings of the 19th international conference on World wide web*, 591-600.
- Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting Positive and Negative Links in Online Social Networks. In *WWW '10 Proceedings of the 19th international conference on World wide web*, 641-650.
- Liben-Nowell, D., and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58 (7), 1019-1031.
- Liu, T. Y. (2009). Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3 (3), 225-331.
- Ma, H., King, I., and Lyu, M. R. (2009). Learning to recommend with social trust ensemble. In *SIGIR '09 Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 203-210.
- Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *CIKM '08 Proceedings of the 17th ACM conference on Information and knowledge management*, 931-940.
- Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. (2011). Recommender systems with social regularization. In *WSDM '11 Proceedings of the fourth ACM international conference on Web search and data mining*, 287-296.
- Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, 42 (11), 30-36.
- Pan, Y., Cong, F., Chen, K., and Yu, Y. (2013). Diffusion-aware Personalized Social Update Recommendation. In *RecSys '13 Proceedings of the 7th ACM conference on Recommender systems*, 69-76.

- Pessiot, J. F., Truong, T. V., Usunier, N., Amini, M. R., and Gallinari, P. (2007). LEARNING TO RANK FOR COLLABORATIVE FILTERING.
- Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452-461.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of Dimensionality Reduction in Recommender System - A Case Study. In *Minnesota Univ Minneapolis Dept of Computer Science*.
- Shen, Y., and Jin, R. (2012). Learning Personal+Social Latent Factor Model for Social Recommendation. In *KDD'12 Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1303-1311.
- Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N., and Hanjalic, A. (2012). CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering. In *RecSys '12 Proceedings of the sixth ACM conference on Recommender systems*, 139-146
- Wan, S., Lan, Y., Guo, J., Fan, C., and Cheng, X. (2013). Informational Friend Recommendation in Social Media. In *SIGIR '13 Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 1045-1048
- Yang, X., Steck, H., Guo, Y., and Liu, Y. (2012). On top k recommendation using social networks. In *RecSys' 12 Proceedings of the sixth ACM conference on Recommender systems*, 67-74.
- Ye, M., Liu, X., and Lee, W. C. (2012). Exploring Social Influence for Recommendation - A Generative Model Approach. In *SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 671-680.
- Zhang, J., Yu, P. S., and Zhou, Z. H. (2014). Meta-path based Multi-Network Collective Link Prediction. In *KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1286-1295.
- Zhu, M., Shen, D., Yu, G., Kou, Y., and Nie, T. (2013). Computing the Split Points for Learning Decision Tree in MapReduce. In *Database Systems for Advanced Applications*, 339-353.