Computer Science and Information Systems 15(2):347–368 https://doi.org/10.2298/CSIS170820003H

Data Imputation Using a Trust Network for Recommendation via Matrix Factorization

Won-Seok Hwang[1], Shaoyu Li[1], Sang-Wook Kim[1], and Kichun Lee[2]

[1] Dept. of Electronics and Computer Engineering
Hanyang University, Korea
{hws23,chris,wook}@hanyang.ac.kr
[2] Dept. of Industrial Engineering
Hanyang University, Korea
skylee@hanyang.ac.kr

Abstract. Existing recommendation methods suffer from the *data sparsity* problem which means that most of users have rated only a very small number of items, often resulting in low recommendation accuracy. In addition, for *cold start users* evaluating only few items, rating predictions with the methods also produce low accuracy. To address these problems, we propose a novel data imputation method that effectively substitutes missing ratings with probable values (*i.e.*, imputed values). Our method successfully improves accuracy of recommendation methods from the following three aspects: (1) exploiting a trust network, (2) imputing *only a part of* missing ratings, and (3) applying them to any recommendation methods. Our method employs a bidirectional connection structure within a distance level for finding reliable users in exploiting a trust network as useful information. In addition, our method imputes only some missing ratings, called *fillable ratings*, whose imputed values are expected to be accurate with a sufficient level of confidence. Moreover, our imputation method is independent of, thus applicable to, any recommendation methods that may include application-specific ones and the most accurate one in each domain. We conduct experiments on three real-life datasets which arise from Epinions and Ciao. Our experimental results demonstrate that our method has recommendation accuracy better than existing recommendation methods equipped with imputation methods or trust networks, especially for cold start users.

Keywords: Recommendation systems, trust networks, data sparsity, imputation.

1. Introduction

Due to the rapid growth of the amount of items such as products, web pages, and multimedia contents in the Internet, users need a tool to find information relevant to their tasks. In order to satisfy users' needs, there have been a substantial number of researches about recommendation methods [1], and several recommendation services are in operation for movies[3], books[4], music[5], etc.

[3] http://www.netflix.com
[4] http://www.amazon.com
[5] http://www.last.fm

In the literature, a large number of recommendation methods have exploited a *rating matrix* including users' ratings for items in order to predict ratings [25][22][14][35]. Figure 1 shows an example of a rating matrix $R = (r_{u,i})_{M \times N}$. In the matrix, a cell specified on the scale of 1 to 5 indicates a rating given to an item by a user. A cell $r_{u,i}$ whose value is empty ($null$) is called a *missing rating* and represents a case in which user u has not evaluated item i. The subscripts M and N mean the numbers of users and items, respectively. For example, in Figure 1 user u_1 has evaluated item i_3 whereas the user has not evaluated item i_1.

	i_1	i_2	i_3	i_4	i_5
u_1			1		
u_2		4			2
u_3				4	
u_4					5
u_5			5		
u_6					

Fig. 1. A sample rating matrix

Recommendation methods predict ratings that an *active user* would assign to his/her unrated items, and then recommend those items whose predicted ratings are top-*K* highest. Some existing methods suffer from low accuracy when most of users rated only a few items, i.e., the rating matrix is very sparse. The situation, called the *data sparsity problem* [1][13][8], is a universal obstacle to most of recommendation methods. In a real-life problem, it has been an important issue how to accurately predict ratings for *cold start users* who rate only a small number of items [1][28] because 50% of users evaluate less than 5 items [10].

To solve the aforementioned problems, researchers [2][15][21] have adopted an approach of imputing missing ratings. The imputation approach infers imputed values for missing ratings by analyzing the users-given ratings. After that, it substitutes the missing ratings with the imputed values. Naturally, the more closely each imputed value on missing rating $r_{u,i}$ reflects user u's preference on item i, the more accurate the results of the recommendation methods are.

In order to construct an effective imputation method that brings forth high recommendation accuracy, we propose three ideas as follows. First, we employ a trust network in our imputation approach. The existing approaches rely on only user-given ratings, so they fail to accurately infer values for the missing ratings when the data sparsity problem occurs. Unlike them, our approach utilizes additional data, *i.e., trust network*, as well as user-given ratings. Trust networks are a kind of a social network representing trust relationships (*i.e.*, represented by directional edges) between users. Indeed, the existing work [6][19][9][18][16][10] already has identified that trust networks provide good information for inferring values for missing ratings. Using trust networks, moreover, we can properly incorporate the preferences of cold start users by referring to those of their connected users, which improves the recommendation accuracy.

Second, we selectively apply the imputation approach to *only a part of* missing ratings rather than all of them. Most existing imputation approaches universally try to fill all missing ratings although they cannot accurately infer imputed values for some missing ratings due to the data sparsity problem. For this reason, we deal with only those missing ratings whose values are inferred with a sufficient level of confidence. We expect that our idea improves the accuracy because we eliminate ratings with inaccurate imputed ratings. Moreover, compared to dealing all missing ratings, our idea needs less time since the recommendation methods analyze less ratings (users-given ratings as well as filling ratings).

Third, we design our imputation approach to be applicable to any recommendation methods. The existing imputation approaches are targeted to specific recommendation methods such as user-based and item-based methods [23][27]. However, in the literature there exists an ample number of recommendation methods that outperform them. In addition, businesses and domain experts can employ any appropriate recommendation methods suiting their particular settings. For this reason, we build the *imputed rating matrix* by substituting the missing ratings with the imputed values in a generic setting. This matrix has the same format as that of the original rating matrix, it can be applicable to any recommendation methods.

Based on the three ideas, we propose a novel *trust-based imputation method*. Our method infers the imputed values for missing ratings by the following sequence. It estimates a value for each missing rating $r_{u,i}$ of item i for user u by aggregating the ratings given by *reliable neighbors* who are reachable from u in a trust network (Idea 1). After that, we estimate only the missing rating of an item evaluated by a sufficient number of reliable neighbors, effectively regulating the amount of imputed ratings in a way that the resulting rating matrix conveys accurate information (Idea 2). Finally, we make an imputed rating matrix by assigning imputed values to the corresponding missing ratings in the original rating matrix, and then apply this matrix to the probabilistic matrix factorization (PMF) [11] model to obtain final recommendation results (Idea 3). PMF performs well on large and imbalanced datasets, requiring less learning time than other matrix factorization approaches.

To demonstrate the effectiveness of the proposed recommendation method, we perform comprehensive experiments on two data sets [19][31] from Epinions[6] and another data set [31] from Ciao[7]. Epinions and Ciao are websites where users assign ratings to items, write reviews, and specify trust relationships to other users. We suggest several ways to find sets of reliable neighbors for a user according to a distance level from the user in consideration of the edges' direction. Examining the accuracy according to different settings in thorough experiments, we find an optimal setting. Also, we analyze the effect of excluding imprecise imputed ratings. We compare the accuracy of the proposed method with a recommendation method equipped with the imputation based on the default voting approach [2] and existing recommendation methods. We demonstrate the superiority of the proposed method to those methods for whole users and cold start users, separately.

To summarize, our main contributions are as follows:

[6] http://www.epinions.com
[7] http://www.ciao.co.uk

- We propose a novel imputation method exploiting a trust network, which can be applicable to any recommendation systems for more accurate results.
- We show that it is better to impute only a few missing ratings rather than all of them unlike the claim of the existing researches.
- We observe that users who are connected to a certain user in any direction (*i.e.*, forward and backward) tend to have similar tastes to his/her.
- We conduct extensive experiments using three real-world datasets and show that recommendation equipped with our method outperforms existing recommendation methods exploiting existing imputation methods or trust networks.

The rest of this paper is organized as follows: We discuss some related work in Section 2. We present our proposed method in Section 3. We report and analyze experimental results in Section 4. Finally, we conclude the paper in Section 5.

2. Related Work

Collaborative Filtering: In this section, we review collaborative filtering methods that are widely studied approaches in the recommendation fields. In general, collaborative filtering methods are categorized into two types: *memory-based* and *model-based* ones [1]. First, memory-based methods [2][23][7] predict the ratings of a user using the similarity of her neighborhoods, and recommend the items with high ratings.

Second, model-based methods [26][22][24] learn a model capturing a users ratings on items, and then predict his/her unknown ratings based on the learned model. Some of them [25][11][13] exploit the matrix factorization for learning a model. These methods focus on fitting the rating matrix using low-rank approximations to make further predictions. The premise behind a low-rank factor model is that a user's preference vector is determined by how each factor applies to that user. However, both of the memory-based and the model-based methods mostly suffer from low accuracy due to the *data sparsity* and *cold-start user* problems [1][13][8].

Imputation Approach: To overcome the problems, some methods that employ an *imputation approach* have been proposed. Default voting [2] is a straightforward imputation-based method that substitutes the missing ratings with default values, such as average ratings by a small group of users for other items. Ma et al. [15] set a threshold to decide whether a user is confident or not and fill the missing data only to users that have highly confident reliable neighbors. Ren et al. [21] proposed an auto-adaptive imputation method which identifies a set of key ratings for a pair of an active user and a target item, and adaptively impute them according to ratings of each user. Zhang et al. [34] propose a recursive recommendation method, where a rating predicted in each recommendation step is used in its next imputation step and the imputed value is also used in the next recommendation step. Moreover, Su et al. [30][29] experimentally examine which recommendation method is the most suitable for the recursive approach.

Those imputation-based recommendation methods show accuracy better than previous collaborative filtering methods. However, most of them ignore additional information such as trust networks and take no advantage of matrix factorization which has been proven to be quite effective in many research results [11][25].

Trust-based Methods: Trust-based recommendation methods, as another solution to the data sparsity and the cold-start user problems, have drawn considerable attention [6][19][9]. These methods exploit a trust network crawled from Epinions, Ciao, and Flixter[8] as additional information. Yang et al. [33] classify the trust-based methods into two categories: neighborhood based and matrix factorization based methods.

In the neighborhood based methods, TidalTrust [6] performs a modified breadth-first search in a trust network to predict a user rating for recommendation. Massa and Avesani [19] introduce MoleTrust, which has a similar idea with TidalTrust. However, when finding users who rated a target item, MoleTrust considers only those users in the trust network within a maximum-depth which is independent of any specific user and item. TrustWalker [9] supposes a random walker [3] who starts from an active user node and moves to the item nodes through the edges on the trust network. The probability of an active user's stay in an item indicates a preference on the item. RelevantTrustWalker [4] is an extended version of TrustWalker. It performs a random walk on a trust relevancy graph built by considering the trust relationships and similarities among users. Dong et al. [5] identify an opinion leader from an active users trustees, and then integrate the comments of those opinion leaders and user preferences to predict a rating on a target item.

Some recent recommendation methods via the matrix factorization use trust networks as well. Ma et al. propose a method, called SoRec [18], which combines probabilistic matrix factorization and trust networks. The method factorizes both of a rating matrix and a trust matrix at the same time with considering latent user features of these matrices are identical. STE [16] considers that each users latent factor is also related to his/her trustors ratings when factorizing a rating matrix. In SocialMF [10], a users latent factor incorporates the users who are reachable from him/her (with decaying weights for more distant users) to improve the quality of recommendation.

Ma et al. [17] propose social regularization as to incorporate trust network information into the training procedure. The social regularization constrains the user's latent factor to be similar his/her rating average or his/her trustees rating average. Wang et al. [32] assume that the topics or categories of a user determines whether she/he trusts someone else or not. Based on this assumption, their method incorporates the multi-faceted trust relationships between users into rating prediction. RTCF method [20] reconstructs a trust network to adjust incorrect trust relationships, and then uses it for more accurate rating prediction.

Noticeably, the trust-based recommendation methods only use ratings given by those users who are reachable from the active user through the edges' *forward direction* in the trust network. The premise behind this is that users have similar preferences with their trustees (*i.e.*, users whom a user trusts). However, because the similarity relationship is symmetrical, users will be similar to not only their *trustees* but also their *trustors* (*i.e.*, users who trust them), which has not been considered in recommender systems. In addition, they merely examine reachable users from a certain user without considering the distance between users.

[8] http://www.flixster.com

3. Proposed Method

The recommendation method equipped with the imputation approach predicts a value for each missing rating based on user given ratings and *imputed ratings*. In this situation, it is a critical aspect for good recommendation accuracy how much each imputed value $v_{u,i}$ well reflects the preference of user u on item i. For this reason, our goal is to infer a proper imputed value for each missing rating.

In our method, first we compute an imputed value $v_{u,i}$ by aggregating the ratings that reliable neighbors for user u, denoted by Rel_u, gave to item i. We define Rel_u as the users who are reachable from user u in a trust network. The rationale behind this is that the existing researches [6][19][9] demonstrated that Rel_u has similar tastes to user u. Specifically, in defining Rel_u we consider two facets: (1) *distance threshold* and (2) *bi-directional connection* of edges. The distance threshold prevents the users whose tastes are dissimilar from being selected as reliable neighbors. The bi-directional connection of edges helps to include more similar users.

We also select only a *part* of missing ratings whose imputed values are determined to be accurate enough rather than all missing ratings. If few users in Rel_u evaluated item i, it would be suspicious to compute imputed value $v_{u,i}$ due to the lack of information. For this reason, we infer an imputed value for only those unrated items evaluated by a sufficient number of reliable neighbors. After that, we build an imputed rating matrix R' by substituting a part of missing rating in the original rating matrix R with the imputed values.

Finally, we apply R' to the well-known recommendation methods such as the probabilistic matrix factorization (PMF) [11] model. The imputed rating matrix R' has the same format to that of the original rating matrix R, so it is applicable to any recommendation methods. In the following subsections, we explain each step in detail.

3.1. Estimating values for imputation through Trust Network

We examine a trust network to define the reliable neighbors Rel_u for each user u. The trust network can be seen as a graph where a node corresponds to each user and an edge corresponds to a trust relationship. Figure 2 demonstrates a sample trust network. For instance, the edge ($U_1 \rightarrow U_4$) represents "U_1 trusts U_4". In this case, we call U_1 and U_4 *trustor* and *trustee*, respectively. Note that a user can be both a trustor and a trustee at the same time: e.g., U_4 in Figure 2 is both a trustee of U_1 and a trustor of U_5.

The imputed values become more accurate as more users are included in Rel_u and those users in Rel_u have tastes more similar to that of user u. In order to gather more neighbors, our imputation method considers the reachable users through *backward* directions of edges as well as *forward* directions in a trust network. The existing researches [6][19][9] observed that the users who are reachable from a user u through the forward direction of edges have similar tastes to u. The rationale behind this is that u's trustees are likely to have similar tastes to that of u. Based on this observation, we derive that u's trustors also would have interests in those items which u is interested in because *similarity relationship* is symmetric by nature. For this reason, unlike existing researches, we consider both forward and backward directions of edges.

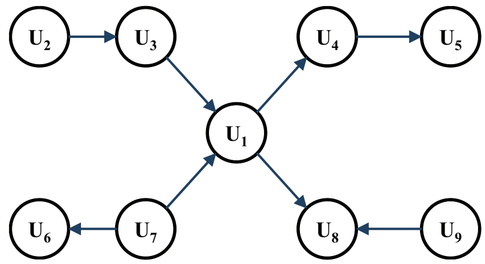


Fig. 2. A sample trust network

In order to prevent users dissimilar to u in tastes from being selected as Rel_u, we include only those users who are reachable within a certain distance δ from u. The rational behind this is that the greater distance between two users is in a trust network, the more different their tastes are.

When we consider bi-directions of edges and distance level δ, we denote the reliable neighbors as $Rel_u^{BID,\delta}$. Similarly, when considering forward directions, we denote it as $Rel_u^{FWD,\delta}$. For example, in Figure 2, if we set $\delta = 1$, $Rel_{U_1}^{BID,1} = \{ U_3, U_4, U_7, U_8 \}$ and $Rel_{U_1}^{FWD,1} = \{U_4, U_8\}$. If we set $\delta = 2$, $Rel_{U_1}^{BID,2} = \{U_2, U_3, U_4, U_5, U_6, U_7, U_8,$ $U_9\}$ and $Rel_{U_1}^{FWD,2} = \{U_4, U_5, U_8\}$.

To impute a missing rating $r_{u,i}$, our method finds $Rel_u^{BID,\delta}$ and estimates the imputed value $v_{u,i}$ as follows:

$$v_{u,i} = \bar{u} + \frac{\Sigma_{x \in X} w_{u,x}(r_{x,i} - \bar{x})}{\Sigma_{x \in X} w_{u,x}} \tag{1}$$

where $r_{x,i}$ denotes the rating of user x on item i, and $\bar{u}$ represents the average rating of user u. As in Equation (1), the proposed imputation method considers not all of the users in $Rel_u^{BID,\delta}$ but those users who rated an item i. The index set X represents users including in $\{x \mid r_{x,i} \neq 0, x \in Rel_u^{BID,\delta}\}$. In addition, $w_{u,x}$ represents the weight which describes the influence of a distance between two users u and x in a trust network. It is necessary and in most cases improves the quality of imputation because it is likely that two users have different preferences if their distance is long. One could simply set it to a constant regardless of the distance if removing the distance influence.

3.2. Imputation Ratio

For estimating the imputed value $v_{u,i}$, the proposed method aggregates the ratings given by users who evaluated item i among the reliable neighbors $Rel_u^{BID,\delta}$. If there are only a few users who evaluated item i, $v_{u,i}$ would be inaccurate due to the lack of information. Figure 3 depicts this problem. Circles represent users, and edges between circles represent trust relationships. The table below each user denotes the ratings given by the user to items. Here, the items without ratings indicate that those items which have not been rated yet by the user. If we want to estimate the imputed values for the unevaluated items of U_1, the ratings given to those items by $Rel_{U_1}^{BID,2} = \{U_2, U_3, U_4, U_5, U_6, U_7, U_8, U_9\}$ will be considered. In the case of item I_2, all the 8 reliable neighbors evaluated it. Thus, we can estimate impute value $v_{1,2}$ by aggregating the opinions of those 8 reliable neighbors. On the other hand, in the case of item I_5 which has been rated by only user U_3, we can use only the opinion of U_3 in the estimation process, which is less reliable relatively. In reality, not so many items have been rated by more than 8 users; on the other hand, many items just like I_5 have been rated by only a small number of users. For instance, in the Epinions dataset crawled by [19], 56% of items are only rated by one user.

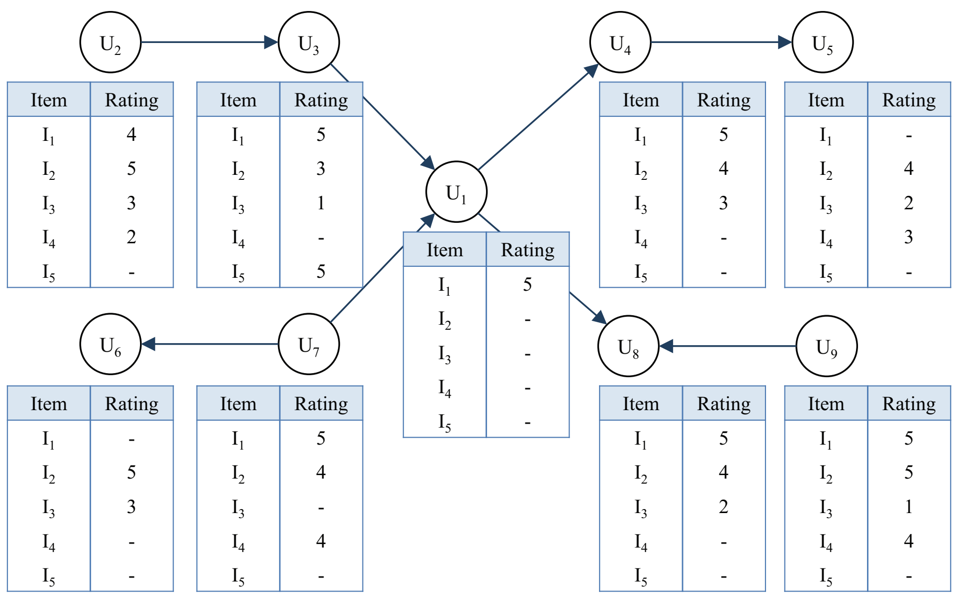


Fig. 3. An example of imputing U_1's missing ratings with the ratings of U_1's reliable neighbors is shown. Item I_5 will be not imputed due to the lack of ratings from the neighbors.

Although Equation (1) can calculate the imputed values for all missing ratings, we claim that it provides more accurate recommendation if we impute *only a part* of missing ratings. For a user u, we call the missing ratings that are substituted with the imputed values *fillable ratings* F_u^θ, where given parameter θ, called *imputation ratio*, controls the number of fillable ratings. Specifically, we sort the missing ratings for a user according to the number of his/her reliable neighbors who evaluate his/her unrated item in descending order; and then, we include top θ percent of missing ratings in the fillable ratings. In Figure 3, if $\theta = 50\%$, only ratings $r_{1,2}$ and $r_{1,3}$ will be included in $F_{U_1}^{\theta=50\%}$.

The imputation ratio has a strong impact on the execution time of the whole recommendation process. Thus, *the control of imputation ratio* is very important in practical sense. For that reason, we apply the fillable ratings to our final imputation method in this paper, which can be defined as:

$$v_{u,i} = \begin{cases} \bar{u} + \dfrac{\Sigma_{x \in X} w_{u,x}(r_{x,i} - \bar{x})}{\Sigma_{x \in X} w_{u,x}}, & \text{if } i \in F_u^{\theta} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

where X indicates users including in $\{x \mid r_{x,i} \neq 0, x \in Rel_u^{BID,\delta}\}$.

After computing the imputed values, we densify the original rating matrix R by assigning the impute values to their corresponding missing ratings and consequently build the imputed rating matrix $R' = (r'_{u,i})_{M \times N}$. This matrix has less missing ratings helping recommendation methods to be free from the data sparsity problem. Moreover, our method eliminates $\theta\%$ missing ratings for every user, so it alleviates the cold start user problem as well. Thus, the recommendation methods would provide more accurate results with the "dense" imputed rating matrix.

3.3. Rating Prediction

Note that our proposed method helps regardless of the choice of underlying recommendation methods as we can simply replace the original rating matrix R by the imputed rating matrix R'. In other words, our method is orthogonal to existing recommendation methods, which is one of strengths of our method. Businesses and applications can choose their own appropriate recommendation methods suiting their particular settings, yet still use our proposed idea for improving accuracy and running time.

In this paper, we apply our imputed rating matrix R' to the probabilistic matrix factorization (PMF) [25], *i.e.*, one of well-known and accurate recommendation methods. For applying R' to PMF, we first map each rating $r'_{u,i}$ to the interval $[0, 1]$ by linearly scaling function $f(x) = (x-1)/(r^{max}-1)$ where r^{max} indicates the maximum value for ratings in R'. Let $X \in R^{K \times M}$ and $Y \in R^{K \times N}$ represent latent feature matrices for users and items, respectively; X_u and Y_i represent K-dimensional latent feature vectors of user u and item i, respectively. The conditional probability of the observed rating is defined as:

$$p(R' \mid X, Y, \sigma_{R'}^2) = \Pi_{u=1}^{M} \Pi_{i=1}^{N} [\mathcal{N}(r'_{u,i} \mid g(X_u^T Y_i), \sigma_{R'}^2)]^{I_{u,i}^{R'}} \quad (3)$$

where $\mathcal{N}(x \mid \mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and $I_{u,i}^{R'}$ is the indicator function that is equal to 1 if user u rated item i and equal to 0 otherwise. The function $g(x)$ is the logistic function $g(x) = 1/(1 + e^{-x})$, which bounds the range of $X_u^T Y_i$ within the range [0, 1]. The zero-mean spherical Gaussian priors are also assumed for user and item feature vectors:

$$p(X \mid \sigma_X^2) = \Pi_{u=1}^{M} \mathcal{N}(X_u \mid 0, \sigma_X^2 \mathbf{I}), \quad p(Y \mid \sigma_Y^2) = \Pi_{i=1}^{N} \mathcal{N}(Y_i \mid 0, \sigma_Y^2 \mathbf{I}) \quad (4)$$

Hence, through a Bayesian inference, we can obtain the posterior probability of the latent variables X and Y as follows:

$$\begin{aligned} & p(X, Y \mid R', \sigma_{R'}^2, \sigma_X^2, \sigma_Y^2) \propto p(R' \mid X, Y, \sigma_{R'}^2) p(X \mid \sigma_X^2) p(Y \mid \sigma_Y^2) \\ & = \Pi_{u=1}^{M} \Pi_{i=1}^{N} [\mathcal{N}(r'_{u,i} \mid g(X_u^T Y_i), \sigma_{R'}^2)]^{I_{u,i}^{R'}} \\ & \quad \times \Pi_{u=1}^{M} \mathcal{N}(X_u \mid 0, \sigma_X^2 \mathbf{I}) \times \Pi_{i=1}^{N} \mathcal{N}(Y_i \mid 0, \sigma_Y^2 \mathbf{I}) \end{aligned} \quad (5)$$

The log of the posterior probability over the user and item features is given by

$$
\begin{aligned}
&\ln p(X, Y \mid R', \sigma^2_{R'}, \sigma^2_X, \sigma^2_Y) = -\frac{1}{2\sigma^2_{R'}} \Sigma^M_{u=1} \Sigma^N_{i=1} I^{R'}_{u,i} (r'_{u,i} - g(X^T_u Y_i))^2 \\
&- \frac{1}{2\sigma^2_X} \Sigma^M_{u=1} X^T_u X_u - \frac{1}{2\sigma^2_Y} \Sigma^N_{i=1} Y^T_i Y_i \\
&- \frac{1}{2}((\Sigma^M_{u=1} \Sigma^N_{i=1} I^{R'}_{u,i}) \ln \sigma^2_{R'} + (M \times K) \ln \sigma^2_X + (N \times K) \ln \sigma^2_Y) + \mathcal{C}
\end{aligned}
\tag{6}
$$

where $\mathcal{C}$ is a constant that does not depend on the parameters. Keeping the parameters (observation noise variance and prior variances) fixed, maximizing the log-posterior over two latent features is equivalent to minimizing the following sum-of-squared-errors objective function with quadratic regularization terms:

$$
\begin{aligned}
\mathcal{L}(R', X, Y) =& \frac{1}{2} \Sigma^M_{u=1} \Sigma^N_{i=1} I^{R'}_{u,i} (r'_{u,i} - g(U^T_u V_i)^2 \\
&+ \frac{\lambda_X}{2} \Sigma^M_{u=1} \|X_u\|^2_F + \frac{\lambda_Y}{2} \Sigma^N_{i=1} \|Y_i\|^2_F
\end{aligned}
\tag{7}
$$

where $_X = \sigma^2_{R'}/\sigma^2_X$, $_Y = \sigma^2_{R'}/\sigma^2_Y$ and $\|\|^2_F$ denotes the Frobenius norm. A local minimum of the objective function can be found by performing a gradient descent search on X_u and Y_i for all user u and all item i.

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial X_u} &= \Sigma^N_{i=1} I^{R'}_{u,i} g'(X^T_u Y_i)(g(X^T_u Y_i) - r'_{u,i}) Y_i + \lambda_X X_u, \\
\frac{\partial \mathcal{L}}{\partial Y_i} &= \Sigma^M_{u=1} I^{R'}_{u,i} g'(X^T_u Y_i)(g(X^T_u Y_i) - r'_{u,i}) X_u + \lambda_Y Y_i
\end{aligned}
\tag{8}
$$

where $g'(x)$ is the derivative of the logistic function, *i.e.*, $g'(x) = e^{-x}/(1 + e^{-x})^2$. In order to reduce the model complexity, in all our experiments, we set $\lambda_X = \lambda_Y$. When estimates of U and V are found, we can predict the rating of user u on item i as follows:

$$
\hat{r}_{u,i} = g(X^T_u Y_i) \tag{9}
$$

4. Evaluation

In this section, we evaluate the effectiveness and efficiency of our imputation method with real-life datasets. Specifically, we first validate that our reliable neighbors from the bi-direction of edges help improve the imputation's accuracy compared with those from the forward direction of edges. In addition, we perform a sensitivity analysis on the distance threshold δ and the imputation ratio θ that are crucial to determine the reliable neighbors and missing ratings to be imputed, respectively. We also evaluate the effectiveness of using a trust network in imputation by comparing it with a simple imputation method without referencing the trust network. Finally, we show the accuracy of the PMF equipped with our imputation method compared with that of the existing work in both cases of the whole users and the cold start users.

4.1. Experimental Set-Up

We use three data sets that are used by previous researches [19][31] for our experiments. Two of those data sets are collected from Epinions, and the last one is collected from Ciao. For convenience, we name those data sets as *Epinons1*, *Epinions2*, and *Ciao* respectively. Table 1 shows the statistics of each data set. Epinions1 has more users than other data sets, and Epinons2 has more items and ratings than the other data sets. The number of trust relationships in Epinions1 is the highest among those of data sets. The size of Ciao is smaller than those of the other two data sets. In Table 1, we define the *cold-start users* as those users who have evaluated less than five items as in the existing researches [19][9][10]. Although it is widely known that most users are generally the cold-start users in many domains, there are only few cold start users in Epinions2 and Ciao data sets. This is because that the previous researches that collected those data sets may not have included the cold start users. The sparsity means the ratio of the number of nonzero elements to the total number of elements in the rating matrix. Table 1 shows that all of the data sets are very sparse.

Table 1. Statistics of Data Sets

Statistics	Data set		
	Epinions1	Epinions2	Ciao
Users	49,289	22,166	7,375
Items	139,738	296,277	106,797
Ratings	664,824	916,085	282,619
Trusts	487,002	355,813	111,781
Cold Users	16,910	21	6
Sparsity	0.0001	0.0004	0.0001

In the evaluation, we employ *mean absolute error (MAE)* and *root mean square error (RMSE)* as accuracy metrics. MAE is a measure that calculates the average of the differences between the predicted ratings and the actual ratings and can be computed by $\frac{\sum_{(u,i)|R_{u,i}}(r_{u,i}-\hat{r}_{u,i})}{\{(u,i)|R_{u,i}\}}$ where $R_{u,i}$ is a boolean variable indicating whether user u has a rating on item i in the test data set. $r_{u,i}$ and $\hat{r}_{u,i}$ denote the user-given and predicted ratings, respectively. RMSE is a measure that puts more emphasis on big errors and is computed by $\frac{\sqrt{\sum_{(u,i)|R_{u,i}}(r_{u,i}-\hat{r}_{u,i})^2}}{\{(u,i)|R_{u,i}\}}$. We perform 5-fold cross validation in our experiments. In each fold, we use 80% of rating data as the training set and the remaining 20% as the test set. We use the same settings for all other experiments.

4.2. Edges Direction in a Trust Network

In order to impute ratings for a user u, the proposed method defines the reliable neighbors who are reachable from u through both of the forward and backward directions (*i.e.*, bi-directions) of edges. Unlike our method, the existing trust-based recommendation methods [6][15][9][10] consider the *forward direction only* in their rating predictions. In this evaluation, we verify our reliable neighbors $Rel_u^{BID,\delta}$ of using bi-direction of edges is more effective than the other neighbors $Rel_u^{FWD,\delta}$ defined by existing researches. This is because $Rel_u^{BID,\delta}$ includes *more* users that have *similar tastes* to those of the users in $Rel_u^{FWD,\delta}$.

First, we analyze Epinions1 to confirm that the users of $Rel_u^{BID,\delta}$ have similar tastes with those users of $Rel_u^{FWD,\delta}$. We set the distance threshold δ from 1 to 6. We compute the average of similarity between a user u and the users in each of $Rel_u^{BID,\delta}$ or $Rel_u^{FWD,\delta}$. As a similarity metric, we employ the Pearson's correlation coefficient to compare two users' ratings on the common items. In the analysis, we ignore the similarity values that are equal or smaller than 0. Table 2 shows that the average similarities for $Rel_u^{BID,\delta}$ are slightly higher than those for $Rel_u^{FWD,\delta}$ in all settings of δ. Thus, as expected, users have similar preferences with their trustors as well as trustees.

Table 2. Similarity between each user and his/her $Rel_u^{BID,\delta}$/$Rel_u^{FWD,\delta}$

Distance Threshold (δ)	Edge's Direction	
	BID	FWD
1	**0.673**	0.667
2	**0.678**	0.671
3	**0.686**	0.679
4	**0.691**	0.685
5	**0.694**	0.689
6	**0.694**	0.691

In addition, we apply two user sets, $Rel_u^{BID,\delta}$ and $Rel_u^{FWD,\delta}$ to our imputation method to compare them in terms of recommendation accuracy. The similarities of a user and its $Rel_u^{BID,\delta}$ are similar to that of $Rel_u^{FWD,\delta}$, so more users in a user set deliver more information that improves the accuracy. We vary the distance threshold δ from 1 to 6 and set the imputation ratio θ as 10% or 20% that are shown as the best values in our experiments (We show details in Section 4.3). We produce the results for whole users and only the cold start users. For cold start users, the rating prediction would be more accurate using $Rel_u^{BID,\delta}$ because $Rel_u^{BID,\delta}$ has much more users than $Rel_u^{FWD,\delta}$ for each cold start user u.

Table 3 compares RMSE of PMF equipped with our imputation when it uses $Rel_u^{BID,\delta}$ and $Rel_u^{FWD,\delta}$ as reliable neighbors. The colored cells indicate the lowest RMSE values of $Rel_u^{BID,\delta}$, and RMSE values of using $Rel_u^{FWD,\delta}$ in the same settings of δ or θ. In the result, $Rel_u^{BID,\delta}$ produces higher accuracy in all data sets and all distance threshold settings. In particular, in the Epinions2, the use of $Rel_u^{BID,\delta}$ shows much better accuracy because the number of users in $Rel_u^{BID,\delta}$ is much bigger than that in $Rel_u^{FWD,\delta}$.

Table 3. Effect of Edges' Direction and Distance Threshold δ for Whole Users (RMSE)

Data and Imputation Ratio (θ)	Edges' Direction	Distance Threshold (δ)					
		1	2	3	4	5	6
Epinions1 (θ=20%)	BID	1.1073	1.0886	**1.0876**	1.0877	1.0878	1.0879
	FWD	1.1100	1.0968	**1.0895**	1.0877	1.0880	1.0880
Epinions2 (θ=10%)	BID	1.1041	1.0900	**1.0843**	1.0851	1.0855	1.0860
	FWD	1.1059	1.1025	**1.0929**	1.0905	1.0912	1.0915
Ciao (θ=20%)	BID	1.0052	0.9864	0.9813	**0.9811**	0.9812	0.9816
	FWD	1.0770	0.9919	0.9843	**0.9822**	0.9822	0.9825

Table 4 shows the result for the cold start users. As shown in Table 1, Epinions2 and Ciao barely include the cold start users, so we perform the experiment on only Epinions1. Compared to Table 3, RMSE of all cases is worse because the cold start users have fewer ratings, which cause *not sufficient for* accurate prediction. For all cases, the accuracy employing $Rel_u^{BID,\delta}$ is higher than that employing $Rel_u^{FWD,\delta}$. The difference between cases using $Rel_u^{BID,\delta}$ and $Rel_u^{FWD,\delta}$ is larger than that in Table 3 because considering bi-directions of edges increases the number of reliable neighbors greatly compared to that considering the forward directions only. We note that the accuracy would increase as our imputation method uses a more number of reliable neighbors.

Table 4. Effect of Edges' Direction and Distacne Threshold δ for Cold Start Users (RMSE)

Data and Imputation Ratio (θ)	Edges' Direction	Distance Threshold (δ)					
		1	2	3	4	5	6
Epinions1 (θ=20%)	BID	1.1995	1.1698	1.1492	1.1427	**1.1415**	1.1418
	FWD	1.2027	1.1911	1.1737	1.1581	**1.1539**	1.1541

Based on the results, we summarize that it is better to consider the bi-direction of edges compared to the forward direction of edges (*i.e.*, considered in the existing work) in our imputation method. Specifically, if our method considers the bi-directions of edges, it can find *more* reliable neighbors for each user u that have also *similar tastes* to u, which leads to higher accuracy in prediction.

4.3. Effect of Parameter δ and θ

In our imputation method, there are two parameters, the distance threshold δ and the imputation ratio θ. δ and θ determine the reliable neighbors and missing ratings to be imputed, respectively. As a result, they have an effect on the accuracy of recommendation with our method. In this section, we examine the accuracy carefully with different settings of parameters δ and θ.

Tables 5, 6, and 7 show the experimental results on Epinions1, Epinions2, and Ciao, respectively. In those tables, the colored cells indicate the MAE and RMSE values of more accurate parameter settings.

Tables 5 and 7 show that our method has the lowest RMSE when the imputation ratio θ is 20%, and MAE is the lowest when θ is 30%. Unlike this result, setting θ as 10% achieves the lowest RMSE while setting θ as 20% does the lowest MAE in Table 6. This is because that Epinions2 has more ratings than Epinions1 and Ciao. In most cases, our method produces the lowest RMSE or MAE when the distance threshold δ is 3 or 4. In Table 5, MAE shows somewhat different tendency to the other results, and it has lowest value when δ is 6.

The results show that setting δ as 3 or 4 and θ as 20% seems to produce the best accuracy in most of data sets regardless of accuracy metric. This result coincides with the claim in Mole Trust [19] that the users who are reachable from the active user within the distance 4 are appropriate to get accurate prediction. Besides, the best imputation ratio θ is not high (*i.e.*, 20%) because imputing more missing ratings cannot guarantee the better accuracy.

Table 5. Effect of Distance Threshold δ and Imputation Ratio θ in Epinions1 Data Set

Metric	Imputation Ratio (θ)	Distance Threshold (δ)					
		1	2	3	4	5	6
RMSE	10%	1.1089	1.0941	**1.0959**	1.0966	1.0968	1.0968
	20%	**1.1073**	**1.0886**	**1.0876**	**1.0877**	**1.0878**	**1.0879**
	30%	1.1072	1.0934	**1.0899**	1.0891	1.0892	1.0893
	40%	1.1062	1.0962	**1.0935**	1.0953	1.0950	1.0950
	50%	1.1051	1.1003	**1.1001**	1.1013	1.1015	1.1023
MAE	10%	0.8666	0.8466	0.8401	0.8392	0.8395	**0.8395**
	20%	0.8626	0.8370	0.8310	0.8295	0.8292	**0.8291**
	30%	**0.8616**	**0.8433**	**0.8271**	**0.8232**	**0.8227**	**0.8226**
	40%	0.8626	0.8437	0.8296	0.8279	0.8281	**0.8280**
	50%	0.8637	0.8450	0.8310	0.8294	0.8284	**0.8284**

Table 6. Effect of Distance Threshold δ and Imputation Ratio θ in Epinions2 Data Set

Metric	Imputation Ratio (θ)	Distance Threshold (δ)					
		1	2	3	4	5	6
RMSE	10%	**1.1041**	**1.0900**	**1.0843**	**1.0851**	**1.0855**	**1.0860**
	20%	1.1030	1.0923	**1.0865**	1.0907	1.0913	1.0916
	30%	1.1013	1.0944	**1.0942**	1.1018	1.1025	1.1029
	40%	1.0994	1.0969	**1.1102**	1.1204	1.1210	1.1249
	50%	1.0977	1.1023	**1.1519**	1.1886	1.1984	1.1995
MAE	10%	0.8631	0.8404	0.8293	**0.8281**	0.8282	0.8294
	20%	**0.8618**	**0.8460**	**0.8285**	**0.8281**	**0.8287**	**0.8300**
	30%	0.8605	0.8465	0.8321	**0.8329**	0.8334	0.8353
	40%	0.8590	0.8469	0.8395	**0.8421**	0.8420	0.8456
	50%	0.8576	0.8501	0.8602	**0.8798**	0.8794	0.8752

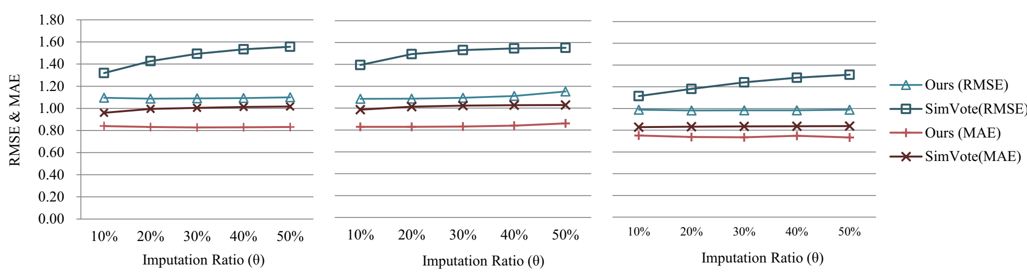
Table 7. Effect of Distance Threshold δ and Imputation Ratio θ in Ciao Data Set

Metric	Imputation Ratio (θ)	Distance Threshold (δ)					
		1	2	3	4	5	6
RMSE	10%	1.0088	0.9899	0.9863	**0.9881**	0.9893	0.9899
	20%	**1.0052**	**0.9864**	**0.9813**	**0.9811**	**0.9812**	**0.9816**
	30%	1.0045	0.9867	0.9816	**0.9826**	0.9827	0.9831
	40%	1.0039	0.9895	0.983	**0.9834**	0.988	0.9934
	50%	1.0036	0.9907	0.9884	**0.9884**	0.998	0.9982
MAE	10%	0.7843	0.7571	**0.7505**	0.7521	0.7556	0.7563
	20%	0.7781	0.7485	**0.7379**	0.7377	0.738	0.7407
	30%	**0.7759**	**0.7476**	**0.7339**	**0.7348**	**0.7348**	**0.7355**
	40%	0.7811	0.7599	**0.7516**	0.7493	0.7484	0.7497
	50%	0.7743	0.7465	**0.7325**	0.7338	0.7521	0.7525

4.4. Effect of a Trust Network in Imputation

After finding the best combination of two parameters, we conduct the experiment to show the use of the trust network is helpful for accurate imputation. In this experiment, we observe the accuracy of prediction using the imputed rating matrix. As a baseline, we introduce a simple imputation method, called *SimVote*, which does not consider the trust network but imputes a part of missing ratings same as our method, and compare it with our imputation method. SimVote fills the missing ratings for each user with average ratings of him/her by following the default voting [2] that is a simple one among the existing imputation methods. To impute similar number of the unrated ratings to our imputation method, we randomly select θ percent of the missing ratings and assign the average ratings to them. Finally, we apply two imputed rating matrix produced by our method and SimVote to PMF and observe their accuracy.

Figure 4 shows the accuracy of prediction using two imputation methods. In this figure, the x-axis and the y-axis represent the imputation ratio θ and the values of MAE or RMSE, respectively. Each sub-figure corresponds to each data set. The proposed imputation method is more accurate than SimVote regardless of the value of θ and the metric. The reason is that the proposed imputation method only imputes those unrated ratings that would be estimated precisely whereas SimVote does not consider this. Therefore, the accuracy of prediction using SimVote is certainly lower. For SimVote, the interesting point is that the more imputed ratings are, the worse accuracy is. This is because the more ratings whose values are inaccurate cannot improve the accuracy of recommendation even if the sparsity problem is addressed. On the other hand, our imputation method accurately fills the missing ratings up to 20%, so the MAE and RMSE is decreasing from 0 to 20% (see Tables 5, 6, and 7).



(a) Epinions1 (b) Epinions2 (c) Ciao

Fig. 4. Comparison Proposed Imputation Method with SimVote

4.5. Accuracy of CF Methods with Our Imputation Method

We compare a recommendation method with our imputation method with four existing recommendation methods: the user-based recommendation method (UBM) [23], AutAI [21], PMF [25], and SocialMF [10]. *UBM*a well-known and simple memory-based method, first selects the neighbors whose ratings are similar to those of an active user and predicts a rating by aggregating the ratings given by the neighbors. We consider UBM as a baseline.

*AutAI*is an imputation-based recommendation method, which is similar to UBM. One important difference is that AutAI defines the neighbors as all the users who rated the target item. Also, AutAI imputes the missing ratings, referred to as a key set, before its prediction. The key set consists of missing ratings on the items on which his/her neighbors did not give ratings but an active user gave ratings. We choose AutAI due to both of the two reasons: (1) to show the effect of the imputation by comparing it with UBM and (2) to show the effect of a trust network by comparing the imputation with a trust network to the imputation without it.

*PMF*PMF is a well-known model-based method, which factorizes a rating matrix into users and items latent features. Next, it calculates the inner product of the two latent features of an active user and a target item for the rating prediction. We choose PMF to show the effect of combining a trust network and imputation by comparing PMF equipped with our imputation method to original PMF. *SocialMF* also employs the matrix factorization and overcomes the data sparsity problem by looking up a trust network. When this method factorizes a rating matrix, it adjusts a user us latent feature to be similar to the (weighted) average of latent features of those users who are reachable from u in the trust network. We select SocialMF to show the strength of our method because it does not employ the imputation and also overlooks the backward directions of the trust network.

In summary, Table 8 shows the difference between our proposed method and four existing methods. Our method is based on the three ideas, i.e., the matrix factorization, the trust network, and the imputation. Unlike our method, the existing methods consider a part or none of them.

Table 8. Comparison of the proposed method with the existing methods

Method	Ideas		
	Matrix Factorization	Trust Network	Imputation
UBM	X	X	X
AutAI	X	X	O
PMF	O	X	X
SocialMF	O	O	X
Ours	O	O	O

In the proposed imputation method, we set $w_{u,v}$ that describes the weight of distance between user u and v to same value 1, which shows best accuracy among the various values. We perform 5-dimensional matrix factorization in PMF, SocialMF and our proposed method, and we set the parameter λU and λV to 0.01 for those methods. Besides performing experiments on all users, we also perform the same experiment for only the cold start users because the rating prediction for them may be more difficult than that for other users.

Table 9 shows the accuracy of the recommendation methods for all users. PMF using our method outperforms other methods especially on Ciao. In addition, two methods exploiting a turst network (*i.e.*, our method and SocialMF overcome PMF that does not consider a trust network. It means that a trust network help rating prediction accuracy to improve. Besides, our method is more accurate than SocialMF due to two reasons. The one reason is that the proposed method considers those users who are reachable through the backward direction, and the other is that it only considers those users who within the distance threshold which can be seen as more reliable. Overall, our method and AutAI that employ the imputation approach overcome other existing recommendation methods because imputation approaches successfully address the data sparsity problem.

Table 9. Accuracy of Recommendation Methods for Whole Users

Dataset	Method	Metric	
		RMSE	MAE
Epinions1	UBM	1.185	0.894
	AutAI	1.097	0.838
	PMF	1.118	0.880
	SocialMF	1.115	0.879
	Ours	**1.088**	**0.823**
Epinions2	UBM	1.136	0.837
	AutAI	1.090	0.835
	PMF	1.109	0.868
	SocialMF	1.106	0.870
	Ours	**1.084**	**0.828**
Ciao	UBM	1.179	0.849
	AutAI	1.006	0.736
	PMF	1.033	0.810
	SocialMF	1.030	0.808
	Ours	**0.981**	**0.734**

Table 10 shows the accuracy of the recommendation methods for the cold start users. The result shows similar tendency to Table 9. All methods show worse accuracy than those in Table 9 because the cold start users have less information. Similar to Table 9, our method has the highest accuracy than other existing recommendation methods. In addition, the differences between the proposed method and those existing methods in terms of RMSE are much bigger compared to those in Table 9. It means that our method deals with the cold start user problem more successfully. AutAI shows worse accuracy than PMF and SocialMF different to Table 9. This is because the imputation for the cold start users cannot impute reliable ratings due to the lack of information for the cold start users. Unlike AutAI, the proposed method can predict the accurate ratings for cold start users because it refers to additional information, a trust network.

Table 10. Accuracy of Recommendation Methods for Cold Start Users

Dataset	Method	Metric	
		RMSE	MAE
Epinions1	UBM	1.350	0.993
	AutAI	1.288	0.934
	PMF	1.210	0.962
	SocialMF	1.209	0.968
	Ours	**1.142**	**0.888**

In summary, our experiments show that the bi-direction of edges is more useful than the forward direction of edges for rating prediction through a trust network. We find that the most appropriate parameters δ and θ are 3 and 20%, respectively in most cases. In addition, we show that a trust network improves imputation approach in terms of accuracy. In addition, our imputation method with the trust network improves the accuracy of recommendation more than SimVote without considering the trust network. Finally, we showed that the proposed method outperforms the existing recommendation methods in terms of RMSE and MAE, and especially it is much better for the cold start users.

5. Conclusions and Future Work

The existing recommendation methods suffer low accuracy from the data sparsity and the cold start user problems. To address these problems, we propose an imputation method that substitutes missing ratings with impute values in a rating matrix. Our method improves the recommendation accuracy in terms of three aspects. First, we employ a bidirectional connection structure in using a trust network as additional information to compute the impute values more accurately. Second, we select missing ratings, that are to be actually filled, whose related information is enough to accurately predict the imputed values. Last, we build the imputed rating matrix that can be applicable to any recommendation method so that more accurate recommendation methods may be utilize it. Through comprehensive experiments, we successfully demonstrated that our proposed method is effective in improving the accuracy of the existing PMF method and outperforms existing recommendation methods. Especially, our method is more effective in addressing the cold start user problem.

One way to improve the accuracy of our method is to utilize additional information other than the trust network. There are a number of information sources such as clicks, bookmarks, and user profiles. Unfortunately, gathering this type of information together with a trust network is difficult. Nevertheless, as the additional information, it is possible to consider the *pre-use preference* [8][12], which indicates a users impression on items before purchasing and using them. Though the users do not leave the pre-use preference explicitly, it is possible to infer them by analyzing only a rating matrix. In our future work, we plan to develop an imputation method that exploits the pre-use preference as well as the trust network.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2017R1A2B3004581), the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069440), and the Ministry of Science, ICT and Future Planning (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2017-2013-0-00881). Professor Sang-Wook Kim is the corresponding author.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence. pp. 43–52 (1998)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (1998)
4. Deng, S., Huang., L., Xu, G.: Social network-based service recommendation with trust enhancement. Expert Systems with Applications 41(18), 8075–8084 (2017)
5. Dong, W., Yi, C., Kai, Y.: Hybrid recommendation algorithm based on trust relationship and user preference. In: Proc. of Int'l Conf. on Electronics Information and Emergency Communication, ICEIEC. pp. 429–433 (2017)
6. Golbeck, J.: Computing and Applying Trust in Web-based Social Networks. Ph.D. thesis, University of Maryland College Park (2005)
7. Hwang, W.S., Lee, H.J., Kim, S.W., Won, Y., Lee, M.: Efficient recommendation methods using category experts for a large dataset. Information Fusion 28, 75–82 (2016)
8. Hwang, W.S., Parc, J., Kim, S.W., Lee, J., Lee, D.: "told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering. In: Proc. of the 32nd IEEE Int'l Conf. on Data Engineering, IEEE ICDE. pp. 349–360 (2016)
9. Jamali, M., Ester, M.: Trustwalker: A random walk model for combining trust-based and item-based recommendation. In: Proc. ACM Int'l. Conf. on Knowledge Discovery and Data Mining, ACM SIGKDD. pp. 397–406 (2009)
10. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proc. of the 4th ACM Conf. on Recommender Systems, Recsys. pp. 135–142 (2010)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
12. Lee, J., Hwang, W.S., Parc, J., Kim, S.W., Lee, D., Lee, Y.: l-injection: Toward effective collaborative filtering using uninteresting items. IEEE Transactions on Knowledge and Data Engineering, TKDE (2017)
13. Lee, J., Lee, D., Lee, Y.C., Hwang, W.S., Kim, S.W.: Improving the accuracy of top-n recommendation using a preference model. Information Sciences 348(20), 290–304 (2016)
14. Lu, Z., Shen, H.: An accuracy-assured privacy-preserving recommender system for internet commerce. Computer Science and Information Systems 12(4), 1307–1326 (2015)

15. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: Proc. of the 30th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR. pp. 39–46 (2007)
16. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: Proc. of the 32nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR. pp. 203–210 (2009)
17. Ma, H., Zhou, D., Liu, C., Lyu, M., King, I.: Recommender systems with social regularization. In: Proc. of ACM Int'l Conf. on Web Search and Data Mining, WSDM. pp. 287–296 (2011)
18. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: Social recommendation using probabilistic matrix factorization. In: Proc. of the 17st ACM Int'l Conf. on Information and Knowledge Management, CIKM. pp. 931–940 (2008)
19. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proc. of the 2007 ACM Conf. on Recommender Systems, RecSys. pp. 17–24 (2007)
20. Moradi, P., Ahmadian, S.: A reliability-based recommendation method to improve trust-aware recommender systems. Expert Systems with Applications 42(21), 7386–7398 (2015)
21. Ren, Y., Li, G., Zhang, J., Zhou, W.: The efficient imputation method for neighborhood-based collaborative filtering. In: Proc. of the 21st ACM Int'l Conf. on Information and knowledge management, CIKM. pp. 684–693 (2012)
22. Rennie, J., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proc. of the 22nd Int'l Conf. on Machine Learning, ICML. pp. 713–719 (2005)
23. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proc. of the ACM Conf. on Computer Supported Cooperative Work. pp. 175–186 (1994)
24. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Proc. of the 25th Int'l Conf. on Machine learning, ICML. pp. 880–887 (2008)
25. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. Advances in Neural Information Processing Systems, NIPS 20, 1257–1264 (2008)
26. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Recommender systems for large-scale e-commerce scalable neighborhood formation using clustering. In: Proc. of the 5th Int'l Conf. on Computer and Information Technology (2002)
27. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaboration filtering recommendation algorithms. In: Proc. of the 19th Intl Conf. on World Wide Web, WWW. pp. 285–295 (2010)
28. Schafer, J., Konstan, J., Riedl, J.: E-commerce recommendation applications. Data Mining and Knowledge Discovery 5(1), 115–153 (2001)
29. Su, X., Khoshgoftaar, T.M., Greiner, R.: A mixture imputation-boosted collaborative filter. In: Proc. of the 21th Int'l Florida Artificial Intelligence Research Society Conference, FLAIRS. pp. 312–317 (2008)
30. Su, X., Khoshgoftaar, T.M., Zhu, X., Greiner, R.: Imputation-boosted collaborative filtering using machine learning classifier. In: Proc. of the 2008 ACM Symp. on Applied Computing, ACM SAC. pp. 949–950 (2008)
31. Tang, J., Liu, H., Gao, H., Das Sarmas, A.: etrust: Understanding trust evolution in an online world. In: Proc. of the 18th Int'l Conf. on Knowledge Discovery and Data Mining, SIGKDD. pp. 253–261 (2012)
32. Wang, N., Chen, Z., Li, X.: Heterogeneous trust-aware recommender systems in social network. In: Proc. of IEEE 2nd Int'l Conf. on Big Data Analysis, ICBDA. pp. 767–771 (2017)
33. Yang, X., Guo, Y., Liu, Y., Steck, H.: A survey of collaborative filtering based social recommender systems. Computer Communications 41, 1–10 (2014)
34. Zhang, J., Pu, P.: A recursive prediction algorithm for collaborative filtering recommender systems. In: Proc. of the 2007 ACM Conf. on Recommender Systesms, RecSys. pp. 57–64 (2007)

35. Zhenzhen, X., Jiang, H., Kong, X., Kang, J., Wang, W., Xia, F.: Cross-domain item recommendation based on user similarity. Computer Science and Information Systems 13(2), 359–373 (2016)

Won-Seok Hwang received M.S. and Ph.D. degrees in Electronics and Computer Engineering from Hanyang University at 2009 and 2016. He currently is a researcher at National Security Research Institute. His research interests include recommendation, data mining, deep learning, and security.

Shaoyu Li received his M.S. degree in Electronics and Computer Engineering from Hanyang University at 2013. He is currently doing data analysis and project manage work at Samsung Electronics. His research interests include recommendation, data mining and deep learning.

Sang-Wook Kim received the BS degree in Computer Engineering from Seoul National University, Seoul, Korea in 1989 and earned the MS and PhD degrees from Korea Advanced Science and Technology (KAIST) at 1991 and 1994, respectively. He is Professor at Department of Computer Science and Engineering, Hanyang University, Seoul, Korea. Professor Kim worked with Carnegie Mellon University and IBM Watson Research Center as a visiting scholar. He is an associate editor of Information Sciences. His research interests include data mining and databases.

Kichun Lee worked for ETRI, Tmax Soft, and Samsung SDS from 1999 to 2006. In 2010, he received his Ph.D. in statistics in Industrial Systems Engineering from Georgia Institute of Technology. After postdoctoral associate positions at Georgia Institute of Technology and Emory University, he is now working at the Department of Industrial Engineering, Hanyang University in Seoul, South Korea. He researches data mining, knowledge extraction, and enterprise computing on the domains of information systems and knowledge services.

Received: August 20, 2017; Accepted: February 2, 2018.