

ENHANCE NMF-BASED RECOMMENDATION SYSTEMS WITH SOCIAL INFORMATION IMPUTATION

Fatemah Alghamedy and Jun Zhang

Department of Computer Science,
University of Kentucky, Lexington, Kentucky, USA

ABSTRACT

We propose an NMF (Nonnegative Matrix Factorization)-based approach in collaborative filtering based recommendation systems to improve the Cold-Start-Users predictions since Cold-Start-Users suffer from high error in the results. The proposed method utilizes the trust network information to impute a subset of the missing ratings before NMF is applied. We proposed three strategies to select the subset of missing ratings to impute in order to examine the influence of the imputation with both item groups: Cold-Start-Items and Heavy-Rated-Items; and survey if the trustees' ratings could improve the results more than the other users. We analyze two factors that may affect results of the imputation: (1) the total number of imputed ratings, and (2) the average of imputed rating values. Experiments on four different datasets are conducted to examine the proposed approach. The results show that our approach improves the predicted rating of the cold-start users and alleviates the impact of imputed ratings.

KEYWORDS

Collaborative filtering, recommendation system, nonnegative matrix factorization, trust, matrix, imputation

1. INTRODUCTION

Recommendation systems [1] became an important tool in E-commerce because it can help both sellers and buyers. The way it helps sellers is by increasing the profits and suggesting items to customers. In addition, recommendation systems facilitate customers to find items they are looking for easily.

Recommendation systems (RS) are classified into three categories: content-based (CB), collaborative filtering (CF), and hybrid. The content-based (CB) system recommends items similar to the user's preference of the items in the past by utilizing external information, such as item descriptions and user's profiles to calculate the similarity between items or users. Since content-based does some manual intervention to collect the user profiles and items descriptions, it is susceptible to errors and does not scale to large items basis. On the other hand, collaborative filtering (CF) supposes that users who agree on the items in the past agree in the future, too. CF calculates the similarity measurement between users using their previous ratings of common items. We can predict that two users will like the same items in the future if both have a high similarity between their ratings in the past for the same items. One of the advantages of CF is that there is no need for any external information like the CB method. The third category of RS

combines content-based (CB) and collaborative filtering (CF) to merge the advantages of both systems into one system and avoid each of the system's limitations.

Collaborative filtering is the most popular approach because of the accuracy in prediction results and fewer resources are required. Collaborative filtering algorithms are divided into two main categories: memory-based methods and model-based methods.

Memory-based methods, also known as neighborhood-based methods, utilize the past and common ratings between users for the same item (user-oriented CF) or common ratings between items from the same user (item-oriented CF) to calculate the similarity measure. The issue with this method arises if there are no common rated items between users thus similarity cannot be calculated. The cold-start users who did not rate many items, e.g., less than five items often have this issue and as a result, the system cannot recommend items.

To reduce the issues with the memory-based methods, model-based methods have been proposed whereas users are modeled based on their past ratings by employing statistical and machine learning techniques to learn models and use these learned models to predict the missing ratings. In addition, model-based doesn't need to calculate the similarity and to find the users' neighbors. However, the model-based algorithms still suffer from the data sparsity problem and fail to address the cold-start users issue.

Relying only on the rating matrix and ignore other sources of information in the dataset that we may use to increase the accuracy of the recommendation is irrational. There are several sources that could be used such as user information (gender, job title, address, hobbies, etc.), item categories, and social information (the relationship between users or trust and distrust list). Traditional recommendation systems suppose that users are i.i.d. (independent and identically distributed) and they ignore the connections among users which does not reflect the real world recommendations.

Recommendation is considered as a social activity. For example, people usually ask a friend to recommend movies to see or music to listen. Based on this research [2], friends in real life are more qualified to advise good and useful recommendations than the traditional recommendation system. In [3], Sinha and Swearingen showed that a user chooses recommendations from friends over recommendation systems, in terms of quality and usefulness even if the recommendation systems have a high novelty factor.

The relationship between the users' taste and their friends' taste has been observed by several researchers such as Ziegler and Lausen demonstrated in [4] who concluded an empirical study of a real online community. Their results showed that there is a similarity in the ratings between users and their friends. Singla and Richardson in [5] analyzed over 10 million users on the social network MSN Instant Messenger with their related search records and they concluded that there is higher probable to have similar interests, such as the topics they are searching for, between the users who chat with each other than the users who do not chat. In addition, the analysis of this large dataset in [6] detected that friends have a tendency to give similar ratings to items.

In the beginning, users trust each other because they agree with their ratings and reviews. The user that creates the trust relationship is called a trustor and the user that has been trusted is named a trustee. After a while, the trustee influences the trustor even on some topics that they did not agree on in the past [7]. In addition, [8] showed that most users participate in social networks more than rating items.

Imputation is the process of replacing missing data with substituted values [9]. In addition, it is one of the approaches that has been used to complete missing data, such as recommendation

systems to reduce rating matrix sparsity. Most recommendation system methods do not require complete data, but the imputation has been used because the predicted ratings are more accurate when there are more ratings available in the rating matrix. In addition, the imputation process has been used as a pre-processing step. Prediction results using the imputation data with an extremely sparse rating matrix often improves [10].

It must be taken into a consideration the error that may be introduced from the imputed ratings. In order to reduce the imputation error and benefit from the imputation, two factors must be taken into the account, (1) which missing data should be imputed and (2) how to impute ratings [11]. The most efficient imputation-based collaborative filtering methods do not impute all missing data by applying strategies to select which missing data should be imputed.

There are several imputation approaches that have been proposed with both collaborative filtering methods: memory-based and model-based collaborative filtering. They are called imputation-based collaborative filtering methods.

We propose a new approach to improve the cold-start users prediction results by reducing the sparsity using the trust user network. In the review websites, users trust other users based on their ratings since they don't know that much of information about each other except the ratings. We can expect that if a user did not provide a rating for an item, then his/her rating for that item will be similar to his/her trustees'. We use the imputation process in the rating matrix by imputing a missing rating with the average of the trust ratings for an item if there is at least one rating.

The remainder of this paper is organized as follows. Section 2 shows the related work. Section 3 defines the problems and notations. Section 4 describes the main ideas of the proposed method. Section 5 presents the datasets, experiments and discusses the results. Conclusions and future work are given in Section 6.

2. RELATED WORKS

Nonnegative Matrix Factorization (NMF) has been applied in the collaborative filtering. In [12], Zhang et al. in used NMF to learn the missing values in the rating matrix which is based on the collaborative filtering method. A nonnegativity constraint is enforced in the linear model to guarantee that all users' ratings can be represented as an additive linear combination of canonical coordinates. Ding et al. proposed in [13] unconstrained 3-factor NMF which has an additional factor matrix to absorb the different scales in the two matrix factors in basic NMF.

It is insufficient to rely on rating information only due to the fact that most datasets suffer from sparsity. The most negative impact is shown with cold-start users who have not rated many items. Other sources of information have been used in order to alleviate this issue, such as user information (gender, occupation, location, interests, etc.), item categories, and social information (relationship between users or trust and distrust list) [6, 14, 15, 16, 17]. Aux-NMF [18] is one of the studies that incorporates the users' and items' information based on NMF method. Their proposed method surpasses the SVD-based data update approach [19].

The social network is one of the sources that have been employed to alleviate the most serious problems of the recommendation system: rating matrix sparsity and cold-start users. The social network can be gathered from internal or external resources. There are review websites that allow users to create a list of users whose reviews they suppose are trustworthy which is called a trust list. Social relationship information has been incorporated into both memory-based [6, 17, 20] and model-based collaborative filtering methods [16, 21].

In [20], Massa et al. proposed a new method that incorporates social network into memory-based collaborative filtering which substitutes the similarity measure with the trust metric to predict the missing ratings. Rather than computing the similarity between two users based on their commonly rated items, they compute the trust weight between users based on the trust web network. The results show that their proposed method using only trust metrics is more effective, in terms of accuracy and coverage than the purely collaborative filtering and the system that combines trust and similarity, especially with cold-start users. In [16] they integrated the social network structure and the user-item rating matrix based on probabilistic matrix factorization.

Moreover, the imputation process has been incorporated into collaborative filtering methods to alleviate rating matrix sparsity. For example, IBCF is a method that has been proposed by Su et al. proposed in [22] in which a subset of missing data is imputed after dividing the rating matrix into subset matrices based on the number of ratings each item received. In addition, [23] proposed a novel algorithm called (IMULT) based on the classic Multiplicative Update Rules (MULT), which utilizes imputation to fill out the subset of unknown ratings.

In [21], they proposed a method to impute users in order to improve the ratings prediction. However, the prediction improves only when New-Users are imputed, but not when All-Users are imputed even though the prediction results of cold-start users with some datasets improved. This indicates that imputing cold-start users could improve the prediction with some cases. In addition, with New-Users imputation method, other users groups got worse results. In [14], they used the trust network to impute missing ratings. The proposed method is based on the probabilistic matrix factorization (PMF) model. Enlightened by these papers, we apply the imputation process to Aux-NMF [18] by utilizing the trust network. Our proposed method is different from [14, 21] in the missing data selection which we impute, and the known ratings which are used to impute the missing ratings. In addition, we analyze two factors that may affect results of the imputation: (1) the total number of imputed ratings, and (2) the average of imputed ratings value.

3. PROBLEM DESCRIPTION

In collaborative filtering, there are m users such that $U = \{u_1, \dots, u_m\}$ and n items $E = \{e_1, \dots, e_n\}$. Each user u_i can rate a set of items. Users represent the rating through an explicit numeric rating, such as a scale from one to five. In addition, the rating information is summarized in an $m \times n$ matrix, which is called a rating matrix $R \in \mathbb{R}^{m \times n}$, $1 \leq i \leq m, 1 \leq j \leq n$. The rows in the rating matrix represent the users, and the columns represent items. If a particular user u_i rates a particular item e_j , then the value of the intersection of the user's row and item's column in the rating matrix R_{ij} holds the rating value. If the user did not rate that item, then the rating will be missing. Nonnegative Matrix Factorization (NMF) [24] is a dimension reduction method. Nonnegative matrix tri-factorization (NMTF) is defined as follows [13],

$$R_{m \times n} \approx U_{m \times k} \cdot S_{k \times l} \cdot V_{n \times l}^T \quad (1)$$

In NMTF, the rating matrix R is factored into three matrices, U, V , and S , where U is a matrix that contains the latent factors for users and V contains the latent factors for items. In addition, S matrix absorbs the different scales between U and V . We divide users into three groups. The first group is New-Users who did not rate any items at all. The second group is Cold-Start-Users who rated at least one item and at most four items. The last group is Heavy-Rating-Users who rated more than four items.

The social information is summarized in an $m \times m$ matrix, which is called the trust matrix $T \in \mathbb{R}^{m \times m}$, $1 \leq p \leq m, 1 \leq q \leq m$. The rows correspond to the users who created a trust relationship (trustor), and the columns correspond to the users who have been trusted by others (trustee). If user u_p trusts user u_q , the value of T_{pq} is equal to 1. On the other hand, a zero in the

trust matrix means there is no trust relationship between the users. Due to the fact that we are using Aux-NMF [18] as a basic algorithm, we need more matrices: the user feature matrix $F_U \in \mathbb{R}^{m \times K_U}$ and the item feature matrix $F_I \in \mathbb{R}^{n \times K_I}$, which hold the users' and items' information. Each user and item belongs to one or more features K_U and K_I , respectively. The Aux-NMF is defined as follows [18],

$$\min_{U \geq 0, S \geq 0, V \geq 0} f(R, W, U, S, V, C_U, C_I) = \alpha \cdot \|W \circ (R - USV^T)\|_F^2 + \beta \cdot \|U - C_U\|_F^2 + \gamma \cdot \|V - C_I\|_F^2 \quad (2)$$

where α, β and γ are coefficients that control the weight of each part. C_U and C_I are the user cluster matrix and the item cluster matrix which are obtained by running the K-Means clustering algorithm on the users feature matrix F_U and items feature matrix F_I .

Generally, the Cold-Start-Users group suffers from a high error in the prediction results. In [21], they proposed a method to impute users in order to improve the ratings prediction. When all users are imputed with all available imputed ratings, some dataset's prediction of Cold-Start-Users improves with the imputation, however, the others do not. In this paper, we intend to study the behavior of the non-New-Users groups with the imputation process and analyze the factors that affect prediction when imputation process is used.

Our experiments show that the average of Cold-Start-Users ratings values in the training set is higher than the whole dataset ratings average and Heavy-Rating-Users ratings average. In addition, the average of the training set ratings of all users is higher than the mean of the rating value. This indicates that users tend to rate items that they like more than items that they don't. This could be for several reasons. First, in the e-commerce era, it is easy for users to know all the information that they need about the item before they make a decision to buy it. In addition, users tend to trust their choices. Further, users tend to buy what they know such as a brand instead of taking a risk and buying what they don't know. In this case, users in reality did not try a lot of options to make a fair rating. In general, Cold-Start-Users have higher MAE because of several reasons. The first one is the lack of the ratings in the training set. The second reason is the average of ratings value of the Cold-Start-Users in the training set because the Cold-Start-Users ratings do not have a significant influence on the whole dataset rating average because of the lack of Cold-Start-Users rating in the training set. In our proposed method, we have two goals: (1) improve the Cold-Start-Users predictions (2) limit the impact of the imputed ratings. This could be done by increasing the total number of the Cold-Start-Users rating and increasing the average of the training set rating value through the imputed ratings, simultaneously.

4. PROPOSED METHOD

We propose a new method to improve the Cold-Start-Users predictions by incorporating the trust information into Aux-NMF. In addition, the proposed method alleviates the impact of imputed ratings in AuxTrsut-NMF, especially on Heavy-Rating-Users.

To perform the proposed method, we need to determine the subset of the real ratings that will be used to calculate the imputed ratings which are called source ratings, and the items which will hold the imputed ratings. The value of the imputed ratings equals the average of the ratings value of the imputed user's trustees for that item, i.e., the source ratings for each imputed rating is all trustees' ratings for the users that will be imputed.

For each user group, we impute them with a limited number of imputed ratings to limit the error that is introduced by the imputed rating. For each user, we consider the items that have been rated by the user's trustees as the candidate items that could be impute.

We have to select carefully which candidate items should be imputed because improving the prediction results must be synchronized between the imputed user and the imputed item at the same time. In addition, some items may be vulnerable to the error that is introduced by the imputed rating more than others.

We have two factors that should be balanced between each other in order to select which item should be imputed from candidate items: (1) the total number of the ratings from all users, and (2) the total number of the ratings from only the user's trustees.

The candidate items that received few ratings, which called Cold-Start-Items, share the same issue with Cold-Start-Users, i.e., the lacking of the total number of ratings which results in less accurate predicted ratings. For that, imputing Cold-Start-Items likely improves the prediction results for the whole system as Cold-Start-Users.

On the other hand, the candidate items that received many ratings, called Heavy-Rated-Items, could be considered as Heavy-Rating-Users who are affected by the imputed ratings negatively as we see in [21].

To demonstrate our idea, we propose two strategies in which the candidate items are ordered based on their total number of ratings ascendingly, which is called CSI case, and descendingly, which is named HI case. For candidate items that have a tie total number of ratings, they are ordered based on the total number of the ratings from the user's trustees descendingly for two reasons. First, allowing more source ratings in order to calculate the imputed ratings which means more opinions from different trustees that results in more accurate imputed ratings. In addition, candidate items that have been rated by many of the user's trustees indicate that the user likely agrees with trustees ratings more than candidate items that have been rated by few trustees because many ratings from different trustee corroborate the opinion.

As we mentioned before, the pervious studies showed that there is a similarity in the ratings between users and their friends [4]. In order to study the influence of the total number of the ratings from the user's trustees on the prediction results, we proposed another strategy in which the candidate items are primarily ordered based on the total number of the ratings from the user's trustees descendingly then by the total number of the ratings for the item from all users ascendingly which is called Trustee case. Table 1 show the summary of the three strategies.

Table 1. The summary of the three proposed cases.

| Rating source | All Users | | User's trustee only | |
|---------------|----------------|------------|---------------------|------------|
| | Order priority | Order type | Order priority | Order type |
| Trustee | 2 | acs | 1 | desc |
| CSI | 1 | acs | 2 | desc |
| HI | 1 | desc | 2 | desc |

1) *Objective Function*: In the proposed method, we replace rating matrix R in Equation 2 with the imputed rating matrix R' such that

$$r'_{ij} = \begin{cases} r_{ij}, & \text{if } r_{ij} \neq 0 \\ \text{Imputed Rating}, & \text{if } r_{ij} = 0, \text{ Imputed Rating} \neq 0, \text{ and meet the conditions} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $r'_{ij} \in R'$, $r_{ij} \in R$, and Imputed Rating is the average of the source ratings if the rating r_{ij} is missing on rating matrix R and the imputed rating is not zero. In the proposed method, each users group has a limited number of the imputed ratings and we have to make sure that the total number of the imputed ratings for each user who does not exceed the parameter setting

In addition, we redefined W as a W' such that:

$$w'_{ij} = \begin{cases} 1, & \text{if } r'_{ij} \neq 0 \\ 0, & \text{if } r'_{ij} = 0 \end{cases} \quad (w'_{ij} \in W', r'_{ij} \in R') \quad (4)$$

By updating Equation (2) using Equations (3) and (4), the objective function is:

$$\min_{U \geq 0, S \geq 0, V \geq 0} f(R', W', U, S, V, C_U, C_I) = \alpha \cdot \|W' \circ (R' - USV^T)\|_F^2 + \beta \cdot \|U - C_U\|_F^2 + \gamma \cdot \|V - C_I\|_F^2 \quad (5)$$

We name this matrix factorization AuxTrustCSU-NMF, where CSU stands for Cold-Start-Users.

2) *Update Formula*: The derivation of update formula is the same as Aux-NMF [3] except we replace the rating matrix R with the imputed rating matrix R' and W with W' . The final update formula is in Algorithm 1, Lines 47-49.

We suppose $k, l \ll \min(m, n)$, the time complexities of updating U , V , and S in each iteration are all $O(mn(k+l))$. Thus, the time complexity of AuxTrustCSU-NMF in each iteration is $O(mn(k+l))$.

3) *Detailed Algorithm*: Algorithm 1 depicts the steps of performing AuxTrustCSU-NMF on the imputed rating matrix R' . As we mentioned before, we perform the algorithm with three cases: Trustee, CSI and HI. Because each user group in the proposed method has a limited total number of the imputed ratings, we set three parameters which defined the total number of the imputed ratings for each user group: *NUIR*, *CSUIR*, and *HUIR*. However, it may take hundreds or thousands of iterations to converge to a local minimum. Thus, in the algorithm, we set an additional stop criterion - the maximum iteration counts. In collaborative filtering, this value varies from 10 ~ 100 which can produce good results.

Algorithm 1 AuxTrustCSU-NMF

Require:

- User-Item rating matrix: $R \in \mathbb{R}^{m \times n}$;
- Trust matrix: $T \in \mathbb{R}^{m \times m}$;
- User feature matrix: $F_U \in \mathbb{R}^{m \times k_U}$;
- Item feature matrix: $F_I \in \mathbb{R}^{n \times k_I}$;
- Column dimension of U : k ;
- Column dimension of V : l ;
- Coefficients in objective function: α, β and γ ;
- Total number of the imputed ratings for New-User group: *NUIR*;
- Total number of the imputed ratings for Cold-Start-User group: *CSUIR*;
- Total number of the imputed ratings for Heavy-Rating-User group: *HUIR*;
- Number of maximum iterations: *MaxIter*;
- Imputation Case: *ImpCase*;

Ensure:

- Factor matrices: $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times l}$, and $V \in \mathbb{R}^{n \times l}$;

User cluster membership indicator matrix: $C_U \in \mathbb{R}^{m \times k}$;
 Item cluster membership indicator matrix: $C_I \in \mathbb{R}^{n \times l}$;
 Imputed rating matrix: $R' \in \mathbb{R}^{m \times n}$;

```

1: Cluster users into  $k$  groups based on  $F_U$  by K-Means algorithm  $\rightarrow C_U$ ;
2: Cluster items into  $l$  groups based on  $F_I$  by K-Means algorithm  $\rightarrow C_I$ ;
3: Initialize  $U, S,$  and  $V$  with random values;
4: for each user  $u_i$  do
5:   find the user's  $i$  trustees from the trust matrix  $T \rightarrow L_t$ 
6:   if  $\text{count}(L_t) > 0$  then
7:     find all items that have been rated by  $L_t \rightarrow \text{candidateItems}$ 
8:     if  $\text{count}(\text{candidateItems}) > 0$  then
9:       for each  $\text{candidateItem } c_j$  do
10:        calculate the average of the rating values of  $L_t$  users for item  $c_j \rightarrow$ 
            $\text{ImputedRatingValue}$ 
11:        count the total number of ratings for  $c_j$  from all users
            $\rightarrow \text{totalRatingsAllUsers}$ 
12:        count the total number of ratings for  $c_j$  from
            $L_t \rightarrow \text{totalRatingsTrusteesUsers}$ 
13:       end for
14:       if  $\text{ImpCase} == \text{Trustee}$  then
15:         Order  $\text{candidateItems}$  based on
            $\text{totalRatingsTrusteesUsers}$  descendingly, then for the tie values
           Order  $\text{candidateItems}$  based on  $\text{totalRatingsAllUsers}$  ascendingly
16:       else if  $\text{ImpCase} == \text{CSI}$  then
17:         Order  $\text{candidateItems}$  based on  $\text{totalRatingsAllUsers}$  ascendingly, then
           for the tie values
18:         Order  $\text{candidateItems}$  based on  $\text{totalRatingsTrusteesUsers}$  descendingly
19:       else if  $\text{ImpCase} == \text{HI}$  then
20:         Order  $\text{candidateItems}$  based on  $\text{totalRatingsAllUsers}$  descendingly, then
           for the tie values
21:         Order  $\text{candidateItems}$  based on  $\text{totalRatingsTrusteesUsers}$  descendingly
22:       end if
23:       if total ratings number of  $u_i == 0$  then
24:          $\text{topImpRatings} = \text{NUIR}$ 
25:       else if total ratings number of  $u_i > 0$  AND total ratings number of  $u_i < 5$  then
26:          $\text{topImpRatings} = \text{CSUIR}$ 
27:       else if total ratings number of  $u_i > 4$  then
28:          $\text{topImpRatings} = \text{HUIR}$ 
29:       end if
30:       Set  $\text{ImputedRatingCounter} = 0$ 
31:       Set  $\text{candidateItemsIndex} = 0$ 
32:       while  $\text{ImputedRatingCounter} < \text{topImpRatings}$  do
33:          $j = \text{index of candidateItems}(\text{candidateItemsIndex})$ 
34:         if  $r_{i,j} == 0$  then
35:            $r_{i,j} == \text{ImputedRatingValue}(\text{candidateItemsIndex})$ 
36:            $\text{ImputedRatingCounter} = \text{ImputedRatingCounter} + 1$ 
37:         end if
38:          $\text{candidateItemsIndex} = \text{candidateItemsIndex} + 1$ 
39:       end while
40:     end if
41:   end if

```



```

42:   end if
43: end for
44: Build weight matrix  $W'$  by Eq. (4);
45: Set  $iteration = 1$  and  $stop = false$ ;
46: while( $iteration < MaxIter$ ) and ( $stop == false$ ) do
47:    $U_{ij} \leftarrow U_{ij} \cdot \frac{[\alpha(W' \circ R')VS^T + \beta C_U]_{ij}}{\{\alpha[W' \circ (USV^T)]VS^T + \beta U\}_{ij}}$ 
48:    $V_{ij} \leftarrow V_{ij} \cdot \frac{[\alpha(W' \circ R')^T US + \gamma C_I]_{ij}}{\{\alpha[W' \circ (USV^T)]^T US + \gamma V\}_{ij}}$ 
49:    $S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T(W' \circ R')V]_{ij}}{\{U^T[W' \circ (USV^T)]V\}_{ij}}$ 
50:    $L \leftarrow \alpha \cdot \|W' \circ (R' - USV^T)\|_F^2 + \beta \cdot \|U - C_U\|_F^2 + \gamma \cdot \|V - C_I\|_F^2$ 
51:   if  $L$  increases in this iteration then
52:      $stop == true$ 
53:     Restore  $U, S$  and  $V$  to their values in last iteration.
54:   end if
55: endwhile
56: Return  $R', U, S, V, C_U$  and  $C_I$ .

```

5. EXPERIMENTAL STUDY

In this section, we discuss the datasets' description, evaluation strategy, and experimental results.

5.1. Data Description

Table 2. Statistics of the datasets.

| Dataset | # Users | # Items | # Ratings | # Trust Relationships |
|-----------|---------|---------|-----------|-----------------------|
| Ciao | 7,375 | 21,978 | 184,024 | 111,781 |
| CiaoDVD | 17,615 | 16,121 | 72,345 | 22,484 |
| Epinions | 22,166 | 15,000 | 180,889 | 355,727 |
| FilmTrust | 1,642 | 2,071 | 35,494 | 1,853 |

In the experiments, we adopt four datasets. Ciao [25], CiaoDVD[26], Epinions [25], and FilmTrust [27] as the test data. We adopt these datasets because they have the information that we need to evaluate the proposed approach: the rating matrix R and trust matrix T .

Ciao is one of the popular review website that displays items from different online shopping websites, such as Amazon and compares the prices from different shopping websites for the same item. Users are allowed to rate items using 5-scale integer ratings (from 1 to 5) and trust each other. When a user (trustor) agrees with another user's reviews (trustee), then the trustor can insert the trustee to his/her own trust list.

There are several datasets that have been extracted from the Ciao website. The first one is Ciao dataset which was crawled from Ciao.co.uk in May 2011 by Tang et al. in [25]. There are 7,375 users and 106,797 items. Each item belongs to one or more of 28 different categories. However, there is no information about users. Due to the MATLAB memory limitation, we only chose users who rated at least one item and items that received at least three ratings ending up with 7,375 users, 21,978 items, and 184,024 ratings.

The second one is Ciao DVD which was crawled from ciao.co.uk, the DVD category, in December 2013 [26]. There are 17,615 users, 16,121 items and 72,345 ratings. Each DVD item belongs to one of the 17 genres. However, there is no information about users.

The Epinions dataset was collected by Tang et al. in May 2011 [25]. There are 22,166 users and 296,277 items. Each item belongs to one or more of 27 categories. However, there is no information about users in this dataset. Due to the MATLAB memory limitation, we chose 15,000 out of 296,277 items, which are the first 5,000 items, the middle 5,000 items, and the last 5,000 items. Ending up with 22,166 users, 15,000 items and 180,889 ratings. Users are allowed to rate the items using 5-scale integer ratings.

FilmTrust was crawled from the entire FilmTrust website in June 2011 [27]. FilmTrust is a website that provides predictive recommendations about movies. However, FilmTrust does not recommend a list of movies to the users. Instead, FilmTrust suggests how much the user may like a chosen movie [28]. The FilmTrust dataset has 1,642 users, 2,071 items, 35,494 ratings, and 1,853 trust relationships. The rate is on a scale of a half star from half star to four stars.

5.2. Evaluation Strategy

We compare the performance between the proposed approach, Aux-NMF [18], and AuxTrust-NMF [21] using the Mean Absolute Error (MAE). The MAE is defined as:

$$MAE = \frac{1}{|TestSet|} \sum_{r_{ij} \in TestSet} |r_{ij} - p_{ij}| \quad (6)$$

where r_{ij} is the actual value while p_{ij} is the predicted value.

We use 80% of the ratings as a training set and 20% as a test set. We perform the imputation process after the dataset is split into training and test sets, and we impute missing ratings using the training ratings only. We perform our experiment in a 5-fold cross-validation approach. The machine we used is equipped with a 2.53Ghz quad-core +HT processor, 8GB RAM and is installed with the UNIX operating system. The code was written and run in MATLAB.

5.3. Results and Discussion

In this section, we present and discuss our experimental results. We compare our proposed method with Aux-NMF [18] and with both cases of AuxTrust-NMF [21]: All-Users and New Users imputation, too.

None of our datasets has information about users, so we set the users' feature parameter, β , to zero. However, using value impacts the weight of the rating matrix in the prediction process. To avoid that, we set the item features parameter, γ , to zero to focus on the analysis of the imputation effect.

Table 3. The MAE of Aux-NMF, New-Users Imp, and the proposed methods with the three cases.

| Dataset | Aux-NMF | Aux-NMF All-Users | Aux-NMF New-Users | Trustee | CSI | HI |
|-----------|---------|-------------------|-------------------|---------------|---------------|--------|
| Ciao | 0.8237 | 0.8305 | 0.8224 | 0.8025 | 0.8029 | 0.8127 |
| CiaoDVD | 1.6503 | 1.6721 | 1.6462 | 1.6348 | 1.6368 | 1.6411 |
| Epinions | 1.0816 | 1.0751 | 1.0760 | 1.0382 | 1.0372 | 1.0448 |
| FilmTrust | 0.7288 | 0.7439 | 0.7269 | 0.7206 | 0.7200 | 0.7226 |

Table 4. The percentage of each items group that are imputed in the training set with the three proposed case: Trustee, CSI, and HI. CSI = Cold-Start-Items group; HI = Heavy-Rated-Items group

| Proposed Case | Trustee Case | | CSI Case | | HI Case | |
|---------------|--------------|--------|----------|--------|---------|--------|
| | CSI | HI | CSI | HI | CSI | HI |
| Ciao | 49.96% | 50.04% | 84.63% | 15.37% | 4.16% | 95.84% |
| CiaoDVD | 42.26% | 57.74% | 95.75% | 4.25% | 4.18% | 95.82% |
| Epinions | 16.47% | 83.53% | 57.88% | 42.12% | 0.86% | 97.38% |
| FilmTrust | 32.90% | 67.10% | 51.16% | 48.84% | 2.15% | 97.85% |

In general, the results of the three cases of our proposed method are better than Aux-NMF and AuxTrust-NMF: All-Users and New-Users with all datasets as we see in Table 3. Further, the results of the CSI case are better than HI with all datasets. However, Ciao and CiaoDVD have better results with Trustee case; and Epinions and FilmTrust are better with the CSI case. We notice that the percentage of the Heavy-Rated-Items that are imputed in Epinions and FilmTrust with Trustee case is very high compared to the other datasets as we see in Table 4. This indicates that imputing Heavy-Rated-Items limits the advantages of the imputations.

On the other hand, when all users are imputed with all available imputed ratings in Aux-NMF All-Users method, the results are the worst among all other methods except Epinions dataset. This is because Epinions has the highest difference between the New-Users before and after the imputation which means the most imputed New-Users compared with the other datasets, as we see in Table 9, which lead to the most improvement in the New-Users results.

Table 5. The MAE for whole dataset and each user group of Aux-NMF, AuxTrust-NMF: All Users and New-Users, and the best case of the proposed method AuxTrustCSU-NM.

| Methods | All-Users | New-Users | Cold-Start-Users | Heavy-Rating-Users |
|-------------------------|---------------|---------------|------------------|--------------------|
| Ciao | | | | |
| Aux-NMF | 0.8237 | 4.4118 | 0.8345 | 0.7452 |
| AuxTrust-NMF All-User | 0.8305 | 1.4235 | 0.8399 | 0.7715 |
| AuxTrust-NMF New-User | 0.8224 | 1.3615 | 0.8345 | 0.7453 |
| AuxTrustCSU-NMF Trustee | 0.8025 | 1.3999 | 0.8118 | 0.7438 |
| CiaoDVD | | | | |
| Aux-NMF | 1.6503 | 4.3433 | 1.2397 | 1.0612 |
| AuxTrust-NMF All-User | 1.6721 | 4.2832 | 1.2722 | 1.1122 |
| AuxTrust-NMF New-User | 1.6462 | 4.2830 | 1.2442 | 1.0689 |
| AuxTrustCSU-NMF Trustee | 1.6348 | 4.2824 | 1.2302 | 1.0606 |
| Epinions | | | | |
| Aux-NMF | 1.0816 | 3.9203 | 1.0770 | 0.9316 |
| AuxTrust-NMF All-User | 1.0751 | 1.9541 | 1.0888 | 0.9769 |
| AuxTrust-NMF New-User | 1.0760 | 1.9495 | 1.0964 | 0.9543 |
| AuxTrustCSU-NMF CSI | 1.0372 | 1.9297 | 1.0529 | 0.9311 |
| FilmTrust | | | | |
| Aux-NMF | 0.7288 | 3.3677 | 0.7326 | 0.6455 |
| AuxTrust-NMF All-User | 0.7439 | 2.7780 | 0.7487 | 0.6679 |
| AuxTrust-NMF New-User | 0.7269 | 2.7735 | 0.7324 | 0.6463 |
| AuxTrustCSU-NMF CSI | 0.7200 | 2.7639 | 0.7242 | 0.6478 |

The New-Users group gets slightly better results than AuxTrust-NMF New-Users method but it is worse in Ciao dataset as we see in Table 5. This could be because of the percentage of New-Users ratings in the test set that belong to New-Users that have been imputed in Ciao dataset is the lowest among other datasets which allow imputing more missing ratings of New-Users without concern about the error from the imputed ratings.

When the Heavy-Rating-Users results in the proposed method are compared to Aux-NMF, Table 5, we see that the results are slightly better with all datasets but not with FilmTrust. This is because the average of the ratings in the training set is the closest to the ratings mean among other datasets as we see in Table 7. Increasing the average of the training ratings value after the imputation leads to more error of the low ratings value. However, when we compare the results of the proposed method with AuxTrust-NMF New-Users imputation method, Epinions dataset gets the highest improvement because the Heavy-Rating-Users group gets the worse results with New-Users imputation among other datasets. CiaoDVD dataset gets worse results with New-Users imputation but it improves with the proposed method, too. Other datasets, Ciao and FilmTrust, did not get much worse results with New-Users imputation thus the change in the results with the proposed method is not notable. We conclude that the proposed method capable to handle the impact of the AuxTrust-NMF imputation on the Heavy-Ratings-Users.

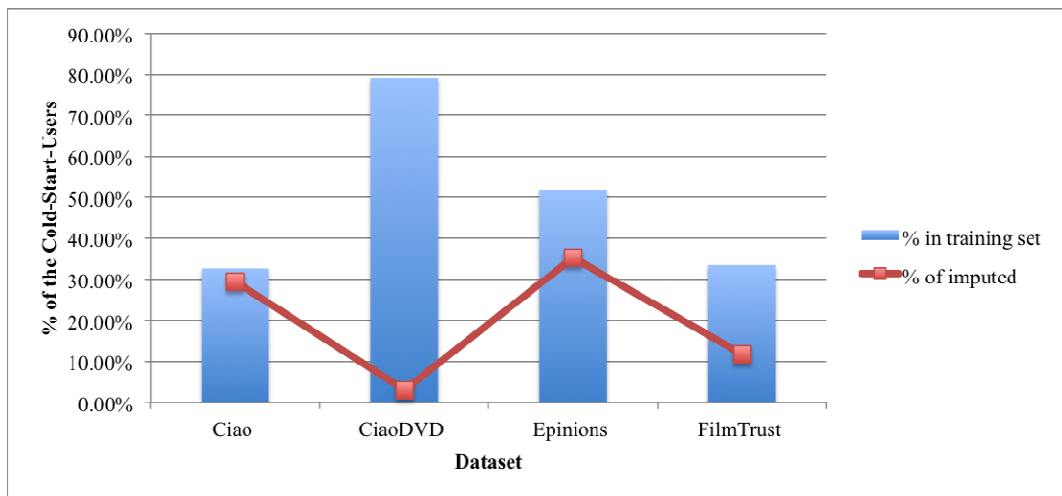


Figure 1: Cold-Start-User information in the training set.

The Cold-Start-Users results improve compared to the Aux-NMF and AuxTrust-NMF with both cases. We notice a proportional relationship between the percentage of the imputed Cold-Start-Users in the training set and percentage of improvement in the results when the proposed method is compared with the Aux-NMF method as we see in Table 5 and Figure 1. Ciao dataset has the highest percentage of improvement in the results and the highest percentage of the imputed Cold-Start-Users in the training set. On the other hand, CiaoDVD has the lowest percentage of the improvement in the results and the lowest percentage of the imputed Cold-Start-Users in the training set. However, when we compare the proposed method to AuxTrust-NMF New-Users, the datasets that have worse results with AuxTrust-NMF New-Users and a high percentage of the imputed Cold-Start-Users in the training set get a better result than other datasets. For example, even though Ciao has the highest percentage of the imputed Cold-Start-Users in the training set than Epinions, Epinions get a better percentage of improvement in the results than Ciao because Epinions get worse MAE with New-Users AuxTrust-NMF method than Ciao. This is the same with CiaoDVD and FilmTrust datasets. We conclude that imputing Cold-Start-Users reduces MAE and capable to handle the impact of the AuxTrust-NMF imputation on the Cold-Start-Users simultaneously.

Table 6. The average of the ratings value in the training set of the original rating matrix R for whole dataset and each user group.

| Dataset | Whole Dataset | Cold-Start-Users | Heavy-Rating-Users |
|-----------|---------------|------------------|--------------------|
| Ciao | 4.1483 | 4.2164 | 4.1442 |
| CiaoDVD | 4.0711 | 4.2860 | 3.9369 |
| Epinions | 3.8742 | 3.9126 | 3.8640 |
| FilmTrust | 3.0028 | 3.1219 | 2.9954 |

Table 7. The average of the ratings value in the training set for whole dataset with Aux-NMF, AuxTrust-NMF: All Users and New-Users, and the best case of proposed method AuxTrustCSU-NMF.

| Dataset | Aux-NMF | AuxTrust-NMF All-Users | AuxTrust-NMF New-Users | AuxTrustCSU-NMF |
|-----------|---------|---------------------------|---------------------------|-----------------|
| Ciao | 4.1483 | 4.1870 | 4.1496 | 4.1569 |
| CiaoDVD | 4.0711 | 3.7887 | 4.0050 | 4.0720 |
| Epinions | 3.8742 | 3.8314 | 3.8382 | 3.9129 |
| FilmTrust | 3.0028 | 2.9376 | 2.9957 | 3.0032 |

As we see in Table 6, the average of Cold-Start-Users ratings value in the training set is higher than the whole dataset ratings value average and Heavy-Rating-Users in all datasets. In addition, the average of the training set ratings of all users is higher than the mean of the rating value. With AuxTrust-NMF All-Users imputation case, the ratings average of the training set after the imputation becomes lower in all dataset except Ciao. In addition, we notice that Cold-Start-Users MAE in Ciao dataset has the lowest increase after the All-Users imputation among other datasets, as we see in Tables 3 and 7.

With AuxTrust-NMF New-Users imputation case, the ratings value average of the training set is higher than AuxTrust-NMF All-Users imputation case. However, Epinions dataset gets the lowest increase in the rating average among other datasets, as we see in Table 7. The Cold-Start-Users result is worse with AuxTrust-NMF New-Users imputation than AuxTrust-NMF All-Users imputation case only in Epinions dataset compared to other datasets. This could be because of the impact of the average of imputed ratings value.

The highest average of ratings values is with the proposed method AuxTrustCSU-NMF with all datasets except Ciao, as we see in Table 7. In addition, the best prediction ratings are with proposed. That indicates that the average of the ratings value in the training set has an important influence on the accuracy of the rating prediction. There is a huge gap between the average of original rating values and the highest average of rating values in Ciao which may result in introducing error. That denotes the need to limit the increase in the average of rating values of the training set.

5.3.1. Parameter Settings

As we mentioned before, we impute each user with a limited number of imputed ratings based on the group that the user belongs to. In our experiment, we set the maximum imputed ratings for each New-Users to 20, Cold-Start-Users to 5, and Heavy-Rating-Users to 3 imputed ratings. Table 8 shows the total number of the imputed ratings for each users group that results in the lowest MAE for the whole dataset.

Table 8. The best parameters setting of the proposed method with the best case of each dataset.

| Dataset | Best Case | New-Users | Cold-Start | Heavy-Users |
|-----------|-----------|-----------|------------|-------------|
| Ciao | Trustee | 12 | 5 | 1 |
| CiaoDVD | Trustee | 8 | 2 | 3 |
| Epinions | CSI | 3 | 4 | 2 |
| FilmTrust | CSI | 10 | 2 | 2 |

Table 9. The percentage of each users groupin the test set before and after the imputation.

| User Group | New-Users | | Cold-Start-Users | | Heavy-Ratings-Users | |
|------------|-----------|--------|------------------|--------|---------------------|--------|
| | Before | After | Before | After | Before | After |
| Ciao | 0.05% | 0.01% | 86.43% | 2.54% | 13.52% | 97.45% |
| CiaoDVD | 13.92% | 13.63% | 73.95% | 50.40% | 12.12% | 35.98% |
| Epinions | 1.29% | 0.43% | 76.93% | 23.77% | 21.78% | 75.80% |
| FilmTrust | 0.30% | 0.23% | 86.19% | 49.01% | 13.50% | 50.76% |

For the New-Users group, there is an inverse relationship between the percentage of the New-Users ratings in the test set that belong to New-Users that have been imputed and the best total of imputed ratings of New-Users. Ciao dataset has the lowest percentage of the New-Users ratings in the test set that belong to New-Users that have been imputed, 0.04%, and the highest total number of imputed ratings among other datasets then FilmTrust comes after Ciao, as we see in Tables 8 and 9. On the other hand, Epinions has the highest percentage of the New-Users ratings in the test set that belong to New-Users that have been imputed, 0.89%, and the lowest total number of imputed ratings among other datasets then CiaoDVD as we see in Tables 8 and 9. This indicates that if we need to predict a lot of ratings for New-Users, we should take into the account the percentage of the imputed New-Users to balance between the advantage of the imputed ratings and the error that is introduced by the imputed ratings.

With the Cold-Start-Users group, there is a proportional relationship between the percentage of imputed Cold-Start-Users in the training set and the total imputed ratings for each Cold-Start-Users as we see in Figure 1 and Table 8. For example, there are more than 60% of Cold-Start-Users in the training set of Ciao and Epinions datasets are imputed. In addition, the total number of imputed ratings for each Cold-Start-Users of Ciao and Epinions datasets are higher than other datasets: CiaoDVD and FilmTrust. The percentage of imputed Cold-Start-Users in the training set are less than 30.5% with CiaoDVD and FilmTrust. This could be because the rating prediction of the un-imputed Cold-Start-Users may hurt via imputed ratings. For that, we need to reduce the imputed ratings for each Cold-Start-User if there is a high percentage of them could not be imputed. In addition, the relationship between the percentage of imputed Cold-Start-Users in the training set and the percentage of ratings in the test set that belong to imputed Cold-Start-Users is proportional. Therefore, there is a proportional relationship between the percentage of the Cold-Start-User ratings in the test set that belong to Cold-Start-Users that have been imputed and the best total of imputed ratings of Cold-Start-User.

Even though Cold-Start-Users group results with the proposed method improve but not Heavy-Rating-Users, both Cold-Start-Users and Heavy-Rating-Users groups are imputed. This could be for several reasons. First, as we mentioned before, imputing Cold-Start-Items improves the results more than imputing Heavy-Rated-Items. Because the candidate items are ordered based on the total ratings from all users ascendingly, imputing Heavy-Rating-Users allows us to impute more Cold-Start-Items. In addition, as we see in Table 10, the average of the ratings in the

training set increases when we impute Heavy-Rating-Users which is one of the factors that results in a lower MAE. However, it decreases in FilmTrust dataset when Heavy-Rating-Users are imputed even though it results in a lower MAE. This is because the average of the ratings for whole dataset and Cold-Start-Users in the training set are the closest to the ratings mean among other datasets as we see in Table 6.

Table10. The average of ratings value in the training set with/without imputing Heavy-Rating-Users.

| <i>NUIR</i> | <i>CSUIR</i> | <i>HUIR</i> | Rating value |
|-------------|--------------|-------------|---------------------|
| Ciao | | | |
| 12 | 5 | 1 | 4.1569 |
| 12 | 5 | 0 | 4.1548 |
| CiaoDVD | | | |
| 8 | 2 | 3 | 4.072 |
| 8 | 2 | 0 | 4.0717 |
| Epinions | | | |
| 3 | 4 | 2 | 3.9129 |
| 3 | 4 | 0 | 3.9035 |
| FilmTrust | | | |
| 10 | 2 | 2 | 3.0032 |
| 10 | 2 | 0 | 3.0042 |

There is an inverse relationship between the percentage of imputed Cold-Start-Users in the training set and the best setting of the imputed ratings of Heavy-Rating-User. In addition, there is an inverse relationship between the best setting of the imputed ratings of Cold-Start-User and the imputed ratings of Heavy-Rating-User. Ciao dataset has the highest percentage of imputed Cold-Start-Users in the training set, highest imputed ratings for each Cold-Start-User, and the lowest imputed ratings for each Heavy-Rating-User. On the other hand, CiaoDVD dataset has the lowest percentage of imputed Cold-Start-Users in the training set, lowest imputed ratings for each Cold-Start-User, and the high estimated ratings for each Heavy-Rating-User. FilmTrust and Epinions datasets are in between. In general, the total of the best setting of the imputed ratings of Cold-Start-User and Heavy-Rating-Users together in our experiment is in the same range which is between four and six imputed ratings in total.

5.3.2. Results Summary

As a conclusion, handling the lack of the Cold-Start-Users and Cold-Start-Items ratings by imputation could improve the rating prediction of them. It must be taken into consideration that each imputed rating affects the average of the training ratings which subsequently affects the prediction performance. In our experiment, the Cold-Start-Users ratings percentage in the test set is really high which we believe that this kind of dataset represents the reality. On the other hand, Cold-Start-Users ratings average in the training set does not have much influence on the whole training set ratings average. This is due to the fact Cold-Start-Users suffer from a lack of the ratings. We suggest using the proposed method with the systems that predict ratings of Cold-Start-Users more than Heavy-Rating-Users.

6. CONCLUSIONS

In this paper, we proposed a method to incorporate social network information into the Aux-NMF using the imputation process to improve the non-New-Users prediction results. We proposed three strategies to select the subset of missing ratings to impute in order to examine the influence

of the imputation with both item groups: Cold-Start-Items and Heavy-Rated-Items; and survey if the trustees' ratings could improve the results more than the other users.

Our results show that imputing Cold-Start-Items improves the results of Cold-Start-Users with AuxTrustCSU-NMF method, especially when the dataset suffers from Cold-Start-Users, but worse at some others. However, two factors must be taken into account, the total number of the imputed ratings and the average of the ratings in the training set after the imputation in order to limit the imputed ratings error. However, our next step is to set the coefficients in AuxTrust CSU-NMF and analyze the impact of user feature matrix and the item feature matrix on the imputed rating matrix.

As a future work, we want to take the advantage of increasing the average of ratings values to improve the prediction results without the need to set the maximum of the total imputed rating for each user group.

REFERENCES

- [1] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [2] P. Bedi, H. Kaur, and S. Marwaha, "Trust based recommender system for semantic web,," in *IJCAI'07 Proceedings of the 20th International Joint Conference on Artificial Intelligence*, vol. 7, pp. 2677–2682, 2007.
- [3] R. R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends,," in *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, vol. 106, 2001.
- [4] C.-N. Ziegler and G. Lausen, "Analyzing correlation between trust and user similarity in online communities," in *International Conference on Trust Management*, pp. 251–265, Springer, 2004.
- [5] P. Singla and M. Richardson, "Yes, there is a correlation - from social networks to personal behavior on the web,," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 655–664, ACM, 2008.
- [6] J. He and W. W. Chu, "A social network-based recommender system (SNRS),," in *Data Mining for Social Network Data*, pp. 47–74, Springer, 2010.
- [7] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback effects between similarity and social influence in online communities,," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 160–168, ACM, 2008.
- [8] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat, "Matrix factorization with explicit trust and distrust side information for improved social recommendation,," *ACM Transactions on Information Systems*, vol. 32, no. 4, p. 17, 2014.
- [9] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. Hoboken, New Jersey: John Wiley & Sons, 2014.
- [10] X.Su,T.M.Khoshgoftaar, and R.Greiner,"Imputed neighbourhood based collaborative filtering,," in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pp. 633–639, IEEE Computer Society, 2008.
- [11] Y. Ren, G. Li, J. Zhang, and W. Zhou, "The efficient imputation method for neighborhood-based collaborative filtering,," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 684–693, ACM, 2012.

- [12] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in Proceedings of the 2006 SIAM International Conference on Data Mining, pp. 549–553, SIAM, 2006.
- [13] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in Proceedings of the 12th ACM SIGKDD, pp. 126–135, ACM, 2006.
- [14] W.-S.Hwang,S.Li,S.-W.Kim,andK.Lee,"Dataimputationusingatrustnetworkforrecommen- dation," in Proceedings of the 23rd International Conference on World Wide Web, pp. 299–300, ACM, 2014.
- [15] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in Pro- ceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 203–210, ACM, 2009.
- [16] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving recommender systems by incorporating social contextual information," ACM Transactions on Information Systems, vol. 29, no. 2, 2011.
- [17] P. Massa and B. Bhattacharjee, "Using trust in recommender systems: an experimental analysis," in International Conference on Trust Management, pp. 221–235, Springer, 2004.
- [18] X. Wang, J. Zhang, P. Lin, N. Thapa, Y. Wang, and J. Wang, "Incorporating auxiliary informa- tion in collaborative filtering data update with privacy preservation," International Journal of Advanced Computer Science and Applications, vol. 5, no. 4, pp. 224–235, 2014.
- [19] X. Wang and J. Zhang, "SVD-based privacy preserving data updating in collaborative filtering," in Proceedings of the World Congress on Engineering, vol. 1, pp. 377–384, 2012.
- [20] P. Massa and P. Avesani, "Trust-aware recommender systems," in Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 17–24, ACM, 2007.
- [21] F. Alghamedy, X. Wang, and J. Zhang, "Imputing trust network information in NMF-based collaborative filtering," in Proceedings of the ACMSE 2018 Conference, ACMSE '18, (New York, NY, USA), pp. 2:1–2:8, ACM, 2018.
- [22] X. Su, T. M. Khoshgoftaar, X. Zhu, and R. Greiner, "Imputation-boosted collaborative filtering using machine learning classifiers," in Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 949–950, ACM, 2008.
- [23] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," Engineering Applications of Artificial Intelligence, vol. 46, pp. 58–66, 2015.
- [24] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in Advances in Neural Information Processing Systems, pp. 556–562, MIT Press, 2001.
- [25] J. Tang, H. Gao, and H. Liu, "mTrust: discerning multi-faceted trust in a connected world," in Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, pp. 93–102, ACM, 2012.
- [26] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "ETAF: An extended trust antecedents framework for trust prediction," in Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, pp. 540–547, IEEE, 2014.
- [27] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," in IJCAI Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2619–2625, 2013.

- [28] J.Golbeck,J.Hendler,etal.,“FilmTrust:Movierecommendationsusingtrustinweb-basedsocial networks,” in Proceedings of the IEEE Consumer Communications and Networking Conference, vol. 96, pp. 282–286, 2006.

AUTHORS

Fatemah Alghamedy is Ph.D. candidate of computer science at University of Kentucky, USA and a faculty at Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. She received her master degree in computer science from Arkansas State University in USA. Her research interests machine learning, data mining, recommendation systems, and biomedical informatics.

Dr. Jun Zhang received his Ph.D. from the George Washington University. He is a professor in the Department of Computer Science at the University of Kentucky. His research interests include, but are not limited to, data mining and privacy, recommendation systems, large scale scientific computing and applications

