

# IMPUTING ITEM AUXILIARY INFORMATION IN NMF-BASED COLLABORATIVE FILTERING

Fatemah Alghamedy<sup>1</sup>, Jun Zhang<sup>2</sup> and Maryam Al-Ghamdi<sup>3</sup>

<sup>1,2</sup>Department of Computer Science,

University of Kentucky, Lexington, Kentucky, USA

<sup>3</sup>Department of Computer Science, University of Jeddah, Jeddah, Saudi Arabia

## ABSTRACT

*The cold-start items, especially the New-Items which did not receive any ratings, have negative impacts on NMF (Nonnegative Matrix Factorization)-based approaches, particularly the ones that utilize other information besides the rating matrix. We propose an NMF based approach in collaborative filtering based recommendation systems to handle the New-Items issue. The proposed approach utilizes the item auxiliary information to impute missing ratings before NMF is applied. We study two factors with the imputation: (1) the total number of the imputed ratings for each New-Item, and (2) the value and the average of the imputed ratings. To study the influence of these factors, we divide items into three groups and calculate their recommendation errors. Experiments on three different datasets are conducted to examine the proposed approach. The results show that our approach can handle the New-Item's negative impact and reduce the recommendation errors for the whole dataset.*

## KEYWORDS

*Collaborative filtering, recommendation system, nonnegative matrix factorization, item auxiliary information, imputation*

## 1. INTRODUCTION

Nowadays, the world steps into new stages that depend mainly on technology. This appears in many different fields, such as everyday life, work, and business. One of the most important results of using technology in business is E-commerce. It has many helpful tools that are used to figure out what the customer wants, such as recommendation systems (RS) [1] which suggest items to users depending on the user's preferences.

Recommendation systems (RS) are classified into three main categories: content-based (CB), collaborative filtering (CF), and hybrid. The content-based (CB) system calculates the similarity between items or users by utilizing external information, like user profiles and item descriptions. The user gets recommendations for items that are similar to what he previously positively rated. Since content-based RS 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. The collaborative filtering (CF) finds users in the community who have same rated items in common. If two users have the same rated items in common, it predicts that they will like the same items in the future. CF doesn't need any external information like the CB method. However, a number of approaches combine these two systems, content-based (CB) and collaborative filtering (CF), into one system to take the advantages of both of them and overcome their limitations.

Collaborative filtering is the most popular approach because its results are more accurate than other approaches and it needs fewer resources. Collaborative filtering algorithms are classified into two main categories, memory-based methods and model-based methods.

Memory-based method, also called neighborhood-based method, relies on the rating of users or items to compute the similarity. It has two types, user-oriented and item-oriented. User-oriented CF computes the similarity between users based on their previous common items ratings, which are known as user neighbors. If there are no common rated items between users, then user-oriented CF will not be able to calculate the similarity, especially with cold-start users. Cold-start users are users who did not rate a lot of items, e.g., less than five items. The system will not be able to recommend items to them because it is hard to find neighbors for them. If we think about the number of items that each user has rated, actually most users rate a small number of items which makes the rating matrix suffer from sparsity and this leads to one of the most significant issues which is called the rating matrix sparseness.

To overcome the memory-based method issues, model-based methods have been proposed. Model-based algorithms model users based on their past items ratings. To predict missing ratings, it employs statistical and machine learning techniques to learn models and use them. However, memory-based RS doesn't need to calculate the similarity and find the users' neighbors. Model-based algorithms also have the problem of data sparsity and still don't solve the issue of cold-start users.

Using only the rating matrix while letting aside all the other information sources in the dataset will decrease the accuracy of the results. Examples of these information are: user information (gender, occupation, location, interests, etc.), item categories, and social information (relationship between users or trust and distrust list). Still, some other data analysis algorithms require complete data.

Imputation is one of the approaches that has been used to complete missing data. The imputation is the process of replacing missing data with substituted values [2]. The imputation method helps recommendation systems to reduce rating matrix sparsity. Even though most recommendation system methods do not require complete data, the imputation has been used. In the recommendation system, if there are more ratings available in the rating matrix, the predicted ratings are more accurate. Due to that fact, the imputation process has been used as a pre-processing step in which missing data are imputed before the rating prediction process, then the system predicts the rating based on original and imputed ratings. Prediction results using the imputation data with an extremely sparse rating matrix often improves [3].

Even though the imputation alleviates the sparsity issue, it must be taken into consideration the error, which may be introduced from the imputed ratings. To get the benefit of the imputation and reduce the imputation error, we need to answer two important questions, (1) which missing data should be imputed and (2) how to impute ratings [4]. For that, the most efficient imputation-based collaborative filtering methods impute a subset of the missing data using strategies that select which missing data should be imputed. There are several methods to impute missing data, such as the ratings mean of either all known ratings or ratings of a particular item or user, and linear regression. In addition, many imputation approaches have been proposed with both collaborative filtering methods: memory-based and model-based collaborative filtering which are sometimes called imputation-based collaborative filtering methods.

We propose a new strategy that handles New-Items issue by incorporating the item auxiliary information with Aux-NMF without hurting other items prediction performance.

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), which is based on the collaborative filtering method, has been applied in the collaborative filtering. Zhang et al. in [5] used NMF to learn the missing ratings in the rating matrix. 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. An unconstrained 3-factor NMF had been proposed by Ding et al. in [6] which has an additional factor matrix to absorb the different scales in the two matrix factors in basic NMF.

It is insufficient to rely only on rating information because most datasets suffer from sparsity. In addition, cold-start items which did not receive many ratings and cold-start users who did not rate many items have the most negative impact. To alleviate this issue, other sources of information have been used, such as user information [7] (gender, location, job title, interests, education level, etc.), item categories [7], and social information (relationship between users or trust and distrust list) [8, 9, 10, 11, 12]. Aux-NMF [7] is one of the studies that incorporates the users' and items' information into NMF method. Their proposed method surpasses the SVD-based data update approach [13].

Moreover, the imputation process has been incorporated into collaborative filtering methods to alleviate rating matrix sparsity. A method called IBCF had been proposed by Su et al. in [14] such that a subset of missing data is imputed after dividing the rating matrix into subset matrices based on the number of ratings each item received. A novel algorithm called (IMULT) had been proposed in [15] based on the classic Multiplicative Update Rules (MULT), which utilizes imputation to fill out the subset of unknown ratings. Furthermore, [16] proposed an imputation method to impute New-Users. The results show that the proposed approach can handle the New-Users issue and reduce the recommendation errors. Enlightened by these papers, we apply the imputation process to Aux-NMF [7] by utilizing item auxiliary information. Our proposed method is different from [16] in many aspects. First, we impute New-Items which focus on the advertising beside the recommendation. In addition, we survey two factors that may affect the imputation: (1) the total number of the imputed ratings for each New-Item, and (2) the value and the average of the imputed ratings.

## 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 rating is missing, that means the user did not rate that item. Nonnegative Matrix Factorization (NMF) [17] is a dimension reduction method. Nonnegative matrix tri-factorization (NMTF) is defined as follows [6],

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

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  $V$ . Due to the fact that we are using Aux-NMF 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 [7],

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

where  $\alpha$ ,  $\beta$  and  $\gamma$  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, NMF cannot recommend items that did not receive any ratings to users. The values in the row that represents this item in matrix  $V$  are zeros. Moreover, unpredictable ratings raise the mean absolute error (MAE) especially when the average value of the ratings in the test set is closer to the maximum rating value than the minimum. In our paper, we call the users that did not rate any items New-Users and the items that did not receive any ratings New-Items.

Aux-NMF can alleviate this issue by adding the users and items cluster constraints such that in each iteration of updating the matrices  $V$ ,  $S$  and  $U$ , the  $\beta$  value is added to the  $U$  matrix and  $\gamma$  to  $V$  matrix. In this paper, we study the impact of the items auxiliary information constraint,  $\gamma$ , in Aux-NMF [7].

Our experiment shows that even though adding the items auxiliary information constraint can alleviate the New-Items issue, other items' MAE may become higher. We divide items into three groups and calculate their MAE. The first group is New-Items which did not receive any ratings at all. The second group is Cold-Start-Items which received at least one rating and at most four ratings. The last group is Heavy-Rated-Items which received more than four ratings. We use the training dataset to count the total number of ratings for each item - not the rating matrix -. In our datasets, we observe that each group of items has different  $\alpha$  and  $\gamma$  values that result in the lowest MAE. With New-Items group, all the datasets prefer to set  $\gamma$  to the maximum value, 0.9, and  $\alpha$  to the minimum, 0.1. This is because adding  $\gamma$  to the rows of New-Items in the  $V$  matrix allows the system to recommend New-Items to users. The best MAE of Cold-Start-Items is when  $\alpha = 1$  and  $\gamma = 0$  with all datasets. However, the best Heavy-Rated-Items MAE results with different  $\alpha$  and  $\gamma$  settings for each dataset. In addition, we observe that the percentage of the New-Items ratings in the test set affects the best settings of  $\alpha$  and  $\gamma$  for the whole dataset. If the percentage of the New-Items in the test set is high, the Aux-NMF will rely more on items auxiliary information constraint even if Cold-Start-Items and Heavy-Rated-Items MAE are getting worse.

We propose a method to impute a subset of New-Items ratings in the training set using the items auxiliary information to alleviate the impact of New-Items on items auxiliary information constraint and handle New-Items issue.

#### 4. PROPOSED METHOD

We propose a new strategy that handles New-Items issue by incorporating the item auxiliary information with Aux-NMF without hurting other items prediction performance. In addition, the proposed method alleviates the impact of the New-Items on the items auxiliary information constraint -  $\gamma$ -. Because imputed ratings introduce error to the system, our proposed method imputes limited ratings for each New-Items whereas each dataset has a parameter of the maximum imputed ratings for each New-Item.

To perform the proposed method, we need to determine the subset of the real ratings that is used to calculate the imputed ratings which are called source ratings, and the users who hold the imputed ratings. For each user, we count the total ratings that the user did to all items that belong to the same New-Item cluster based on the item cluster matrix  $F_I$ . After ordering the users based on the total ratings descendingly, the top-N users are selected to hold the imputed ratings. For each top-N user, only the user's real ratings are utilized to calculate the imputed ratings. Thereby, we ensure that we maintain the user rating pattern without involving other users' ratings which may have different rating pattern.

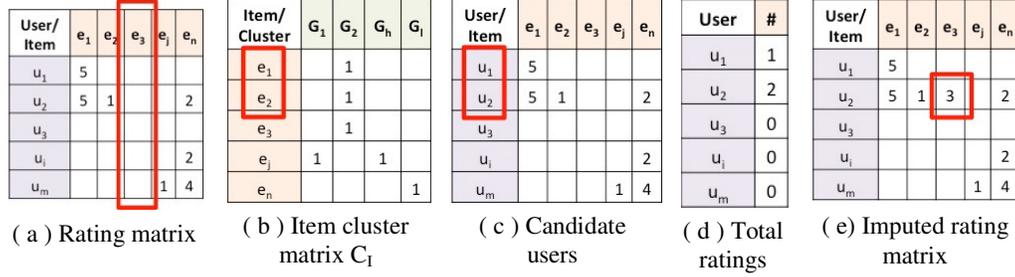


Figure 1. A simple example of the imputation process.

Figure 1 is a simple example to illustrate the basic idea of the imputation. Figure 1 (a) is the rating matrix that presents the users, items, and the users' ratings to the items. As we see, item  $e_3$  is a New-Item because there is no rating for it. To impute  $e_3$ , we need to find all items that belong to the same cluster as  $e_3$ . Figure 1 (b) displays the item cluster matrix  $C_1$ . Item  $e_3$  belongs to cluster  $G_2$  and items  $e_1$  and  $e_2$  belong to the same cluster as  $e_3$  belongs to. The candidate users that may hold the imputed rating are  $u_1$  and  $u_2$  because they did rate at least one of  $e_1$  and  $e_2$  items (Figure 1 (c)). User  $u_1$  rated two items while user  $u_2$  did one rating only that belong to cluster  $G_2$ . If we determine to impute one rating for each New-Item, then  $u_2$  will hold the imputed rating for  $e_3$  because  $u_2$  did the highest number of ratings as we see in Figure 1 (d). The source ratings are the ratings that are used to calculate the imputed rating. In our example, the ratings 5 and 1 of  $u_2$  are the source ratings. The average of the imputed source ratings is 3. The imputed rating of user  $u_2$  to New-Item  $e_3$  is equal to 3 as we see in Figure 1 (e).

In reality, introducing New-Items to the system is actually advertising items to the customers. For that, the prediction error of the users that have a high probability to like the New-Item should be less compared to the users that don't. There are two methods to calculate the imputed ratings. The first one is the average of the subset of the real ratings that are used to impute, source ratings, and the second method is the most frequent rating appears in that subset.

1) *Objective Function*: Aux-NMF developed the objective function for weighted and constrained nonnegative matrix tri-factorization that incorporates the auxiliary information of users and items, as we see in Equation 2.

To handle the New-Item issue, we replace the rating matrix  $R$  with imputed rating matrix  $R'$  such that

$$r'_{ij} = \begin{cases} r_{ij}, & \text{if } r_{ij} \neq 0 \\ \text{Imputed Rating,} & \text{if total ratings of item } j = 0 \text{ and source ratings } \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $r'_{ij} \in R'$ ,  $r_{ij} \in R$ , and Imputed Rating could be either the average of the source ratings or the most frequent ratings.

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

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

By updating Equations (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 AuxNew-Item-NMF.

2) *Update Formula*: The derivation of update formula is the same as Aux-NMF [7] 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 12-14.

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 AuxNew-Item-NMF in each iteration is  $O(mn(k+l))$ .

3) *Detailed Algorithm*: In this section, we present the AuxNew-Item-NMF algorithm. Algorithm 1 depicts the steps of performing AuxNew-Item-NMF on the imputed rating matrix  $R'$ . We perform this algorithm with two cases. The first case is when the imputed ratings are equal to the average of source ratings which is called the Average-Imputation case. The second case is when the imputed ratings are equal to the most frequent ratings in source ratings which is called Most-Imputation case. 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 New-Item Imputation

---

##### Require:

- User-Item rating matrix:  $R \in \mathbb{R}^{m \times n}$ ;
- 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:  $\alpha, \beta$  and  $\gamma$ ;
- Number of maximum iterations: *MaxIter*;
- Number of maximum imputed ratings for each New-Item: *MaxImputedRatings*;

##### 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: **Function** New-Items Imputation( $R, C_{I_{Row}}, j, Imputation$  Case)
- 2:     **for** each group  $g_I$  in  $C_{I_{Row}}$  **do**
- 3:         **if**  $g_I == 1$  **then**
- 4:              $g_IItems = g_IItems +$  all items belong to  $g_I$
- 5:         **end if**
- 6:     **end for**
- 7:     **for** each user  $u_i$  **do**
- 8:          $candidateImputedUsers =$  count the total ratings of  $u_i$  for all items in  $g_IItems$
- 9:     **end for**
- 10:      $OrderedUsers =$  sort  $candidateImputedUsers$  based on the total ratings indescending order

```

11:   for  $u_{imputed} = 1 : MaxImputedRatings$  in  $OrderUsers$  do
12:     if Imputation Case = Average then
13:        $r'_{u_{imputed}j}$  = the average ratings of  $u_{imputed}$  for all items in  $g_I Items$ 
14:     else if Imputation Case = Most then
15:        $r'_{u_{imputed}j}$  = the most frequent ratings value of  $u_{imputed}$  for all items in  $g_I Items$ 
16:     end if
17:   end for
18:   return  $r'_{:j}$ 
19: end function

```

```

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 item  $e_j$  do
5:   if  $e_j$  total ratings == 0 then
6:      $r'_{:j}$  = New-ItemsImputation( $R, C_{I_{e_j}}, j,$  Imputation Case)
7:   end if
8: end for
9: Build weight matrix  $W'$  by Eq.(4);
10: Set  $iteration = 1$  and  $stop = false$ ;
11: while ( $iteration < MaxIter$ ) and ( $stop == false$ ) do
12:    $U_{ij} \leftarrow U_{ij} \cdot \frac{[\alpha(W' \circ R')VS^T + \beta C_U]_{ij}}{\{\alpha[W' \circ (USV^T)]VS^T + \beta U\}_{ij}}$ 
13:    $V_{ij} \leftarrow V_{ij} \cdot \frac{[\alpha(W' \circ R')^T US + \gamma C_I]_{ij}}{\{\alpha[W' \circ (USV^T)]^T US + \gamma V\}_{ij}}$ 
14:    $S_{ij} \leftarrow S_{ij} \cdot \frac{[U^T(W' \circ R')V]_{ij}}{\{U^T[W' \circ (USV^T)]V\}_{ij}}$ 
15:    $L \leftarrow \alpha \cdot \|W' \circ (R' - USV^T)\|_F^2 + \beta \cdot \|U - C_U\|_F^2 + \gamma \cdot \|V - C_I\|_F^2$ 
16:   if  $L$  increases in this iteration then
17:      $stop = true$ ;
18:     Restore  $U, S,$  and  $V$  to their values in last iteration.
19:   end if
20: endwhile
21: 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 1. Statistics of the datasets.

Dataset	# Users	# Items	# Ratings	New-Items ratings % in the test set
CiaoDVD	17,615	16,121	72,345	13.22%
Ciao	7,375	21,978	184,024	0.57%
Epinions	22,166	15,000	180,889	5.34%

In the experiments, we adopt CiaoDVD [18], Ciao [19], and Epinions [19] as the test data. Table 1 shows the statistics information of the datasets.

The CiaoDVD was crawled from [ciao.co.uk](http://ciao.co.uk), the DVD category, in December 2013 [18]. 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. Users are allowed to rate the items using 5-scale integer ratings (from 1 to 5).

The Ciao dataset was crawled from [Ciao.co.uk](http://Ciao.co.uk) in May 2011 by Tang et al. in [19]. 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 5-scale integer ratings are used to rate the items.

The Epinions dataset was collected by Tang et al. in May 2011 [19]. 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.

## 5.2. Evaluation Strategy

We compare the performance between the proposed approach AuxNew-Item-NMF and Aux-NMF [7] 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 data 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 UNIX operating system. The code was written and run in MATLAB.

## 5.3. Results and Discussion

To study the impact of the New-Items imputation process on predicting ratings and parameter settings of Aux-NMF[7], we divide items into three groups and calculate their MAE: New-Items, Cold-Start-Items, and Heavy-Rated-Items.

Some parameters of the proposed algorithms need to be determined in advance. Table 2 gives the parameter setup in AuxNew-Items-NMF (see Algorithm 1).

Table 2. Parameter Setup in AuxNew-Items-NMF.

Dataset	$\beta$	$k$	$l$	MaxIter	MaxImputedRatings
CiaoDVD	0	2	15	10	3
Ciao	0	10	20	10	15
Epinions	0	10	20	10	5

As mentioned before, with the none imputation case -Aux-NMF method-, the percentage of the New-Items ratings in the test set affects the best settings of  $\alpha$  and  $\gamma$  for the whole dataset. If the percentage of the New-Items ratings is high, the system relies on items auxiliary information constraint,  $\gamma$ , more than the rating matrix, because adding  $\gamma$  value to the  $V$  matrix allows the system to predict the New-Items ratings and then recommend them to the users. However, the other items' group, Cold-Start-Items and Heavy-Rated-Items, may have different best settings of  $\alpha$  and  $\gamma$ . In addition, the difference in the MAE between the best setting of  $\alpha$  and  $\gamma$  for the whole dataset and each item group can be large. In this analysis, we demonstrate that imputing New-Items helps to reduce the difference of MAE between the best setting of  $\alpha$  and  $\gamma$  for the whole dataset and for each item group.

Table 3. MAE results of the whole dataset and item groups with all selected combinations of  $\alpha$  and  $\gamma$  of both methods: Aux-NMF and AuxNew-Item-NMF.

$\alpha$	$\beta$	All-Items MAE		New-Items MAE		Cold-Start-Items MAE		Heavy-Rated-Items MAE	
		Aux-NMF	AuxNew-Item-NMF	Aux-NMF	AuxNew-Item-NMF	Aux-NMF	AuxNew-Item-NMF	Aux-NMF	AuxNew-Item-NMF
<b>CiaoDVD</b>									
0.1	0.9	2.0532	1.9011	2.6477	1.5036	1.8222	1.8153	2.0106	2.0118
0.2	0.8	2.0698	1.8918	2.8351	1.4921	1.7997	1.7951	2.0056	2.0066
0.3	0.7	2.0750	1.8853	2.9164	1.4839	1.7832	1.7799	2.0026	2.0034
0.4	0.6	2.0762	1.8801	2.9588	1.4771	1.7695	1.7671	2.0006	2.0012
0.5	0.5	2.0760	1.8758	2.9834	1.4708	1.7576	1.7558	1.9993	1.9998
0.6	0.4	2.0750	1.8721	2.9985	1.4647	1.7467	1.7454	1.9985	1.9988
0.7	0.3	2.0738	1.8689	3.0073	1.4588	1.7364	1.7357	1.9982	1.9983
0.8	0.2	2.0726	1.8664	3.0123	1.4532	1.7271	1.7267	1.9986	1.9986
0.9	0.1	2.0720	1.8649	3.0148	1.4486	1.7189	1.7187	2.0000	2.0001
1	0	2.1810	1.8660	3.8322	1.4474	1.7142	1.7140	2.0030	2.0036
<b>Ciao</b>									
0.1	0.9	0.8158	0.8036	3.0171	0.8332	0.9207	0.9212	0.7486	0.7487
0.2	0.8	0.8083	0.7954	3.1542	0.8339	0.8942	0.8945	0.7489	0.7489
0.3	0.7	0.8029	0.7897	3.1828	0.8340	0.8752	0.8754	0.7495	0.7495
0.4	0.6	0.7986	0.7855	3.1849	0.8343	0.8603	0.8604	0.7501	0.7501
0.5	0.5	0.7952	0.7820	3.1849	0.8346	0.8478	0.8479	0.7508	0.7509
0.6	0.4	0.7924	0.7792	3.1849	0.8351	0.8370	0.8370	0.7518	0.7518
0.7	0.3	0.7901	0.7769	3.1849	0.8357	0.8273	0.8273	0.7529	0.7529
0.8	0.2	0.7882	0.7750	3.1849	0.8367	0.8183	0.8182	0.7544	0.7544
0.9	0.1	0.7867	0.7735	3.1849	0.8095	0.8381	0.8095	0.7562	0.7562
1	0	0.7911	0.7723	4.1654	0.8401	0.8007	0.8006	0.7586	0.7586

Epinions									
0.1	0.9	1.3005	1.2205	2.6663	1.1633	1.8002	1.7721	1.1912	1.193
0.2	0.8	1.2991	1.2077	2.8476	1.1302	1.6772	1.6589	1.1857	1.1871
0.3	0.7	1.2957	1.1997	2.9053	1.1018	1.5988	1.5858	1.1829	1.1839
0.4	0.6	1.2927	1.1938	2.9291	1.0762	1.5426	1.5326	1.1812	1.1819
0.5	0.5	1.2900	1.1892	2.9379	1.0539	1.4986	1.4909	1.1801	1.1805
0.6	0.4	1.2876	1.1857	2.9400	1.0350	1.4628	1.4565	1.1793	1.1795
0.7	0.3	1.2857	1.1827	2.9404	1.0174	1.4323	1.4275	1.1789	1.1788
0.8	0.2	1.2841	1.1802	2.9405	0.9986	1.4056	1.4030	1.1786	1.1786
0.9	0.1	1.2831	1.1781	2.9405	0.9752	1.3831	1.3822	1.1788	1.1788
1	0	1.3349	1.1780	3.9059	0.9653	1.3679	1.3674	1.1799	1.1801

Before the New-Items imputation, the best setting of the New-Items group is when  $\alpha$  is equal to the minimum value, 0.1, and  $\gamma$  is equal the maximum value, 0.9 in all datasets as we see in Table 3. After imputing New-Items with the average of the source ratings, the New-Items prediction improves remarkably for all selected  $\alpha$  and  $\gamma$  combinations in all datasets as we see in Table 3. In addition, the best setting of CiaoDVD and Epinions New-Items group is  $\alpha = 1$  and  $\gamma = 0$ . However, Ciao dataset has the same  $\alpha$  and  $\gamma$  best setting of New-Items group before and after the imputation. The best setting of  $\alpha$  and  $\gamma$  for other items groups, Cold-Start-Items and Heavy-Rated-Items, did not change for all datasets and the MAE is almost the same.

We observe that the best  $\alpha$  and  $\gamma$  setting of New-Items group is the same as the item group that MaxImputedRatings value within its limits. For example, each New-Item in CiaoDVD and Epinions datasets is imputed with 3 and 5 imputed ratings, respectively, and the best  $\alpha$  and  $\gamma$  setting of New-Items of both datasets are equal to Cold-Start-Items group best setting. However, the best  $\alpha$  and  $\gamma$  setting of New-Items in Ciao dataset is the same as Heavy-Rated-Items because each New-Item is imputed with 15 imputing ratings which make them as a Heavy-Rated-Item. This explains the reason that the best  $\alpha$  and  $\gamma$  setting of Ciao New-Items dataset did not change after the imputation.

As we see in Table 3, the imputation process improves the results and the best  $\alpha$  and  $\gamma$  settings are different in all the datasets. After the imputation, Ciao and Epinions datasets rely totally on the rating matrix with  $\alpha = 1$  and  $\gamma = 0$ . In addition, CiaoDVD dataset relies almost on the rating matrix with  $\alpha = 0.9$  and  $\gamma = 0.1$ . The difference between MAE of the item groups with the best  $\alpha$  and  $\gamma$  setting of the whole dataset and of each item group is moot compared to the none imputation case. Before New-Items imputation, the difference in Epinions dataset between the lowest MAE of New-Items and MAE of the same group with the best  $\alpha$  and  $\gamma$  setting of the whole dataset is the highest, which is 0.2742. However, after the imputation, Ciao dataset has the most difference which is between the lowest MAE of the Heavy-Rated-Items group and the MAE of them with the best  $\alpha$  and  $\gamma$  setting of the whole dataset, which is 0.0099.

As a conclusion, using item auxiliary information for imputation, not the NMF process, is a better strategy.

#### 5.4. The Impact of Imputed Rating Value

In this section, we demonstrate how the value of the imputed ratings and the average of all the imputed ratings impact the results. There are two cases to calculate the imputed rating value. The

first one is when the imputed rating value is equal to the average of the source ratings. We call this case, New-Items Average-Imputation case. In the second case, the imputed rating value is equal to the most frequent rating value that appears in the source ratings instead of the average. We call this case New-Items Most-Imputation case. The predicted rating is zero when the system cannot predict the rating which is called unpredictable ratings. This happens because of the impact of the New-Users. After applying NMF, some of the New-Item rows in matrix V are zeros even though all New-Items are imputed. For each rating value of New-Items in the test set, we consider its MAE as a high value when it is larger than the whole dataset MAE. On the other hand, we consider the MAE as a low value when it is equal to or lower than the whole dataset MAE.

Table 4. The average of the imputed ratings with both New-Items imputation cases: Average and Most.

Dataset	Average	Most
CiaoDVD	3.63	4.04
Ciao	4.10	4.46
Epinions	3.89	4.3

By applying Average-Imputation case to Ciao dataset, 96.12% of the rating value 4 of New-Items in the test set get low MAE which is the highest percentage among all other rating values, as we see in Table 6. This is because of the average of the imputed ratings which is 4.10 as shown in Table 4. With the second imputation case, the average of the imputed ratings increases up to 4.46. The low MAE percentage of rating value 5 for New-Items in the test set increases from 55.41% to 85.40%, which is the highest percentage among all other rating values as we see in Table 6. On the other hand, the low MAE percentage of the rating value 4 declines to 80.77%. Because the imputed rating average of both imputation cases is above 4, none of the rating value 1 and 2 MAE of New-Items in the test set are low even though there are few 1 and 2 imputing ratings in the second imputation case as we see in Table 5.

Table 5. The percentage and average for each imputed rating value range with both imputation cases: Average and Most.

Rating value range		CiaoDVD		Ciao		Epinions	
>	<=	%	average	%	average	%	average
<b>New-Item Average-Imputation Case</b>							
0	1	0.00%	N/A	0.00%	N/A	0.00%	N/A
1	2	0.00%	N/A	0.04%	1.52	0.02%	2
2	3	20.06%	2.74	1.82%	2.67	2.71%	2.89
3	4	52.55%	3.55	39.32%	3.72	48.74%	3.56
4	5	27.39%	4.42	58.82%	4.40	48.52%	4.29
<b>New-Item Most-Imputation Case</b>							
0	1	0.53%	1	0.18%	1	0.19%	1
1	2	0.34%	2	0.32%	2	1.36%	2
2	3	24.50%	3	4.29%	3	10.69%	3
3	4	44.13%	4	44.17%	4	44.14%	4
4	5	30.51%	5	51.04%	5	43.63%	5

Table 6. The percentage of the New-Items rating values in the test set and the percentage of their MAE cases (high/low) after the New-Item imputation with both cases: Average and Most.

Rating Value	Rating %	Unpredictable Rating	High MAE		Low MAE	
			Average	Most	Average	Most
<b>CiaoDVD</b>						
1	4.85%	11.22%	71.42%	84.45%	17.36%	4.34%
2	8.88%	8.33%	21.25%	44.83%	70.43%	46.84%
3	18.80%	9.69%	0.80%	6.84%	89.52%	83.48%
4	33.15%	18.40%	0.06%	0.09%	81.53%	81.50%
5	34.33%	26.76%	2.22%	1.46%	71.03%	71.78%
<b>Ciao</b>						
1	3.59%	2.22%	97.78%	97.78%	0.00%	0.00%
2	4.95%	3.75%	96.25%	96.25%	0.00%	0.00%
3	12.14%	1.41%	76.57%	89.71%	22.02%	8.88%
4	31.84%	1.90%	1.97%	17.33%	96.12%	80.77%
5	48.74%	1.77%	42.83%	12.83%	55.41%	85.40%
<b>Epinions</b>						
1	4.68%	5.43%	91.82%	92.98%	2.75%	1.59%
2	7.20%	2.60%	90.70%	92.87%	6.70%	4.53%
3	17.64%	2.45%	17.66%	38.74%	79.89%	58.81%
4	33.82%	2.98%	1.61%	1.15%	95.41%	95.87%
5	36.66%	4.310%	25.55%	13.27%	70.14%	82.43%

The CiaoDVD dataset has the lowest average of the imputed ratings in the first and second imputation cases among other datasets, as shown in Table 4. For the first imputation strategy, the average of the imputed ratings is 3.63. The rating value 3 of New-Items has the highest percentage of the low MAE, then rating value 4 and 5, respectively (Table 6). In addition, some 1 and 2 rating values of New-Items in the test set have low MAE. With the second imputation strategy, the imputed rating average increases up to 4.04 as we see in Table 4. This leads to decrease the low MAE percentage of rating values 1, 2, and 3 (Table 6). However, there is almost no improvement in the rating prediction (low MAE percentage) of 4, and 5 rating values of the New-Items. This is probably because of several reasons. First, the total number of the ratings in the test set in CiaoDVD is much less than other datasets, as we see in Table 1. The second reason is the unpredictable ratings is much more than other datasets especially for the high rating values: 4 and 5, as we see in Table 6. The third one is the sum of the New-Items high rating values (4 and 5) percentage in the test set is the lowest compared to other datasets as we see in Table 6. Due to these facts, the increase in the low MAE percentage of the high rating values (4 and 5) is not notable in this case, even though there is an increase in the average of imputed ratings. Although the percentage of imputed ratings with low values (1,2 and 3) in the second imputation case are more than in the first imputation case, the percentage of the high MAE of the low rating values (1,2 and 3) increase because the average of the imputed ratings increased, too.

The imputed ratings average of Epinions dataset is in between CiaoDVD and Ciao datasets as shown in Table 4. With the first imputation case, the highest percentage of the low MAE is for rating values 4, then 3 and 5, respectively, where the average of the imputed ratings is 3.89. However, the average of the imputed ratings in the second imputation case is 4.30 which raises the percentage of the low MAE of rating value 5 up to 82.43% and reduces the percentage of the

low MAE of rating value 3 to 58.81%. As we observe in other datasets, there are more imputed ratings of low values (1,2 and 3) in the second imputation case than the first one. However, the low MAE percentage of the low rating values (1,2 and 3) decreases.

Table 7. The MAE of both New-Items imputation cases: Average and Most when  $\alpha = 1$ .

Dataset	Imputation Case	All-Items MAE	New-Items MAE	Cold-Start-Items MAE	Heavy-Rated-Items MAE
CiaoDVD	Average	<b>1.8660</b>	1.4474	<b>1.7140</b>	<b>2.0036</b>
	Most	1.8700	<b>1.4752</b>	1.7152	2.0038
Ciao	Average	0.7723	0.8400	<b>0.8006</b>	0.7586
	Most	<b>0.7720</b>	<b>0.7910</b>	<b>0.8006</b>	<b>0.7585</b>
Epinions	Average	<b>1.1780</b>	<b>0.9653</b>	<b>1.3674</b>	<b>1.1800</b>
	Most	1.1796	0.9806	1.3711	1.1807

Table 7 shows the MAE results of both New-Items imputation cases: Average and Most when  $\alpha = 1$  and  $\gamma = 0$ . We set *MaxImputedRatings* of both New-Items imputation cases as is shown in Table 2. The results show MAE for the whole dataset and for each item group. Only MAE of Ciao dataset is slightly lower with the New-Items Most-Imputation case than the Average-Imputation case. This is because Caio dataset has the highest percentage of the rating value 5 in the test set among other datasets (Table 6). In addition, the most improvement in the prediction in the second imputation case is with rating value 5, as we see in Table 6. On the other hand, the best MAE for other datasets is New-Items Average-Imputation case. For the New-Items group, the results of Epinions and Ciao dataset are better with the second imputation case because the strategy improves the prediction for the rating value 5 which reduces the error average. However, the results of New-Items group in CiaoDVD dataset did not improve in the second imputation case because there is no improvement in the prediction with any rating values as shown in Table 6.

As a conclusion, the prediction accuracy of the rating values that are close to the average of imputed ratings is better than other rating values. In addition, the influence of the imputed rating average is more effective than the value of the imputed ratings. Hence, the average of the imputed ratings determines which rating values will have high or low MAE compared to the whole dataset MAE. Because recommending New-Items to users considers as an advertisement, we think that the users that have a high probability to like the New-Item need to have more accurate prediction than the users that don't. Raising the average of the imputed ratings allows the system to predict the high rating values more accurately than the low rating values.

## 5.5. Parameter Study

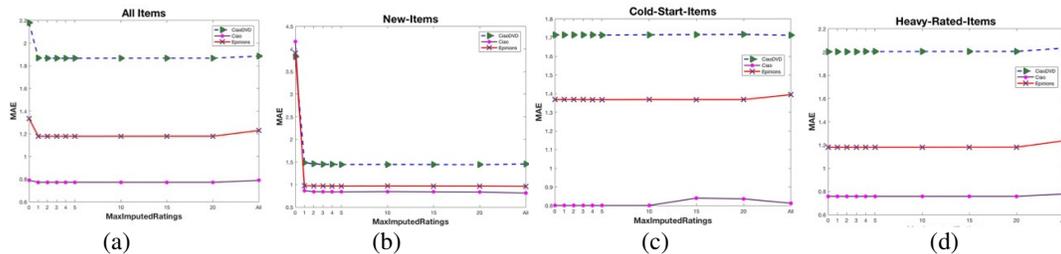


Figure 2. The MAE of New-Item Average-Imputation case with different values of *MaxImputedRatings* when  $\alpha = 1$ .

In AuxNew-Item-NMF, the parameter *MaxImputedRatings* needs to be set. We run the experiment with different total numbers of the imputed ratings for each New-Item. In this experiment, we set  $\alpha = 1$  and  $\gamma = 0$  with New-Item Average-Imputation case. In general, the MAE of all three datasets are lower after New-Items imputation than all MAEs of all selected combinations  $\alpha$  and  $\gamma$  before the imputation regardless of the total number of imputed ratings, *MaxImputedRatings*, as shown in Figure 2(a).

Mostly, adding more imputed ratings, *MaxImputedRatings*, improves the results of the New-Items group prediction results slightly. Nevertheless, adding only one imputed rating to each New-Item allows the system to recommend New-Items to users and reduces the New-Items MAE remarkably compared to none imputation case as we see in Figure 2(b). When all available imputed ratings are imputed for each New-Item, CiaoDVD and Ciao MAE are worse. However, the result of Epinions dataset slightly improves but requires a long time to impute the rating matrix. This demonstrates that adding imputed ratings is not always advantageous because they introduce errors to the system at the same time even for New-Items.

As we see in Figure 2(d), the results of Heavy-Rated-Items show that more imputed ratings lead to increase the MAE of them. However, there is a difference in the increment ratio of MAE between the datasets. Ciao dataset has the lowest New-Items rating percentage in the test set among other datasets, as we see in Table 1. For that, the Heavy-Rated-Items MAE did not increase with the *MaxImputedRatings* increment but did increase when all possible imputed ratings of New-Items are imputed. On the other hand, the highest percentage of New-Items ratings in the test set among other datasets is in CiaoDVD dataset and Heavy-Rated-Items MAE increases with each time the *MaxImputedRatings* is increased as shown in Figure 2(d). The New-Items rating percentage in the test set of Epinions's dataset is in the middle of Ciao and CiaoDVD. As we see in Figure 2(d), there is an increment in the Heavy-Rated-Items MAE but not with each time *MaxImputedRatings* is increased. Overall, the best of Heavy-Rated-Items MAE is without imputation process.

In general, to set *MaxImputedRatings* parameter, we need to balance between the imputation advantage and the imputation error that impacts Heavy-Rated-Items results. Table 2 shows the best setting of *MaxImputedRatings* that improves the rating prediction of New-Items and limits the error that may introduce to the other items. As we see from both Tables 1 and 2, there is an inverse relationship between the best *MaxImputedRatings* parameter setting and the percentage of New-Items ratings in the test set. CiaoDVD dataset has the most New-Items rating percentage in the test set and the lowest *MaxImputedRatings*. On the other hand, Caio has the lowest New-Items ratings percentage in the test set and the highest *MaxImputedRatings*.

## 6. CONCLUSION

In this paper, we proposed a method to incorporate item auxiliary information into the Aux-NMF using the imputation process. Our results show that the proposed method allows the system to recommend New-Items to the users. In addition, using item auxiliary information for imputation, not the NMF process, is a better strategy. In addition, increasing the average of imputed ratings improves the prediction accuracy of the users that have a high probability to like the New-Item.

As a future work, we want to study the behavior of other items group, i.e., Cold-Start-Items and Heavy-Rated-Items with the imputation. In addition, we want to study the factors that affect the best setting for each item group.

**REFERENCES**

- [1] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [2] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. Hoboken, New Jersey: John Wiley & Sons, 2014.
- [3] X. Su, T. M. Khoshgoftaar, and R. Greiner, "Imputed neighborhood 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] X. Wang, J. Zhang, P. Lin, N. Thapa, Y. Wang, and J. Wang, "Incorporating auxiliary information in collaborative filtering data update with privacy preservation," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 4, pp. 224–235, 2014.
- [8] W.-S. Hwang, S. Li, S.-W. Kim, and K. Lee, "Data imputation using a trust network for recommendation," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 299–300, ACM, 2014.
- [9] 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, pp. 9:1–9:23, 2011.
- [10] 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.
- [11] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 203–210, ACM, 2009.
- [12] P. Massa and B. Bhattacharjee, "Using trust in recommender systems: an experimental analysis," in *International Conference on Trust Management*, pp. 221–235, Springer, 2004.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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), ACM, 2018.

- [17] 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.
- [18] 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.
- [19] 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.

## AUTHORS

**Fatemah Algahmedy** 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.



**Maryam Al-Ghamdi** is a graduate student of computer science at University of Jeddah, Saudi Arabia. Her research interests artificial intelligence, machine learning, human computer interaction, and web development.