Improving a Recommender System by Collective Matrix Factorization with Tag Information

Bu Sung Kim[1], Heera Kim[2], Jaedong Lee[3], Jee-Hyong Lee†

Department of Computer Engineering[1]
Department of Software Platform[2]
Department of Information and Communication Engineering[3, †]
Sungkyunkwan University
Suwon, Republic of Korea
{hiyakiss, hrahoy, ultrajaepo, john}@skku.edu

***Abstract*— Collaborative filtering (CF) is the most widely used method of recommender systems. However, it is hard to give users reliable recommendation when there is little information about users. This is the sparsity problem of CF. In this paper, we propose a collective matrix factorization method using tag information to solve the sparsity problem. With tag information, we construct a user-tag matrix that represents users' preferences about tags. Using the user-tag matrix, we convert sparse user-item matrix into dense user-item matrix. In our method, the collective matrix factorization has the role of transferring information between the user-item matrix and user-tag matrix. We experimentally show that our method generates more precise prediction than general CF suffering from the sparsity problem.**

Keywords—recommender system; sparsity problem; collaborative filtering; collective matrix factorization; tag information;

I. INTRODUCTION

Many researchers have studied recommender systems to extract useful information in a huge amount of information. Among various recommender systems, a collaborative filtering (CF) is the most widely used method because it reflects users' preferences on the recommendation [1]. To recommend proper items using CF, we need sufficient information that represents users' preference about items; i.e. ratings in a user-item matrix. However, in most case, there are insufficient ratings in the user-item matrix. This is because there are so many items that most of users only access a limited number of them, and new users and items can be added to the system constantly [2]. The user-item matrix that contains insufficient ratings degrades the performance of the recommender system, and this is one of the most challenging problems of CF [3]. We call this the sparsity problem in CF. In order to address this problem, researchers have suggested various methods. The most common approaches are considering auxiliary information in the process of general CF. Those are content information of items or users' demographic information [4, 5]. However, those have drawbacks because they are based on the ratings of sparse user-item matrix as ever.

More recently, researchers have introduced collective matrix factorization methods to reduce sparsity level of user-item matrix. Assuming they incorporate relevant information on the target domain as additional information, collective matrix factorization methods have been proven to be a highly accurate and scalable approach to recommender systems [2]. On the other hand, there might be some issues that they include less relevant information on the target domain [6].

In this paper, we propose a collective matrix factorization method based on tag information to deal with the sparsity problem in the user-item matrix. The reason why we use tag information is that we regard tags as more detail and clear information to represent users or items than other additional information e.g. users' demographic or content information of items. Users may search items by the tags that they are most interested in, and they can make sense of an item by the tags that other users have applied. In addition, tags can also be used to organize a collection of items, or as a way for users to express their opinions on items to the user community [7]. That is, the tag information is strongly related to the user-item matrix that is the main information of CF. Therefore, we can more reduce the sparsity problem if we use this information in the collected matrix factorization.

To use the tag information in the collective matrix factorization method, we firstly construct a user-tag matrix that represents users' preferences about tags. The constructed user-tag matrix is very dense matrix. This user-tag matrix shares the concept of users with the user-item matrix. Then, it is expected that the sparsity problem may be alleviated by information flow from dense user-tag matrix to sparse user-item matrix. In our method, the collective matrix factorization has the role of transferring information between the user-item matrix and user-tag matrix. In other words, the sparse user-item matrix is converted into dense matrix through learning the user-tag matrix in the process of collective matrix factorization. After

† : Corresponding Author

making dense user-item matrix, we use elements of this matrix as predicted ratings. Lastly, we evaluate the performance of our method by comparing predicted ratings with the actual ratings.

The remainder of this paper is organized as follows. Section 2 gives the related work. Section 3 introduces the proposed method, in which we present an overview of our method, the way to construct auxiliary information and the processes for collective matrix factorization in detail. In Section 4, we report the experimental results and offer some discussions. To evaluate our method, we apply the proposed method in the domain of movie recommendation and measure the accuracy. In Section 5, we draw conclusions and present future works.

II. RELATED WORK

There have been many researches to solve the sparsity problem in CF. The most common approach to solve the sparsity problem is filing blanks of the user-item matrix using additional information, such as demographic information of users and content information about items [4, 8]. Adomavicius et al. assumed that if two users have similar demographic information, they have similar preference. However, this assumption might be so strong in the sense that it is not always guaranteed two similar demographic users have similar preferences. Dai et al. supposed that users mark similar ratings to items that have similar content information, but similar content information could not represent all aspects of items. Therefore, filling blanks directly using such information could not sufficiently improve the performance of recommender system.

Hybrid recommender systems have been introduced to recommend more reliable items dealing with the sparsity problem. There have been various hybrid recommender systems according to their implementations. Lekakos et al. proposed a method that used CF by default and applied content-based approach (CB) whenever there was the sparsity problem in CF [9]. This approach dealt with the sparsity problem indirectly using the property that CB is not affected by the sparsity problem. In this work, they assumed that the CB always gives better results than CF when there is the sparsity problem. However, it may be not always guaranteed that items having similar contents get the similar ratings from a user. Therefore, there are some limits in hybrid approach using combination of CF and CB.

Many researchers have studied collective matrix factorization methods to alleviate the sparsity problem in CF [2, 6, 10, 11]. Vincent W et al. used GPS history and some additional information to recommend proper activities in target locations. They proposed the relevance between activities as additional information. To extract the relevance of two activities, they counted the number of web pages in which two activities appeared at the same time. However, it is hard to say that the relevance of two activities is proportional to the number of web pages that contain both at the same time. In addition, we could not determine the polarity of words using the frequency of words. Thus, this method may have limits and could not guarantee the significant improvement.

III. PROPOSED METHOD

In this section, we propose a method to overcome the sparsity problem in CF. Our method is based on the collective matrix factorization using a user-tag matrix. In the process of the collective matrix factorization, we decompose the user-item matrix and user-tag matrix into two sub matrices respectively. After that, we learn sub matrices to approximate the given user-item matrix and user-tag matrix. We formulate an error function to learn, and use the gradient descent method to get the minimum error. As a result, the sparse user-item matrix is converted into the dense matrix, and elements of the dense matrix are used as predicted ratings of given user-item matrix.

Our method consists of three modules. The first module constructs a user-tag matrix from given data. The user-tag matrix represents users' preferences about tags. The second module, key part of our method, performs the collective matrix factorization using the user-tag matrix constructed in the first module. In the last module, we try to predict unknown ratings using dense user-item matrix that is the result of the prior step to evaluate the accuracy of our method.

3.1 Data modeling

1) Mathematical terms

We use bold uppercase $\boldsymbol{U}$, $\boldsymbol{T}$ and $\boldsymbol{G}$ to denote the user-item matrix, item-tag matrix, and Tag-genome (item-tag matrix). The small case letters m, n, and p denote the number of users, items, and tags respectively. $\boldsymbol{U(u,i)}$ represents the user u's rating about item i. $\boldsymbol{T(u,t)}$ denotes the user u's preference about tag t. In addition, $\boldsymbol{G(i,t)}$ draws the relevance between item i and tag t.

2) User-tag matrix

The user-tag matrix is the information that depicts users' preferences about tags. We use the user-item matrix and Tag-genome to construct the user-tag matrix. Tag-genome is the information about relations between items and tags, which is distributed by GroupLens research group [1]. Tag-genome contains relevance information between 9,734 items and 1,128 tags. It represents the relevance as a real number in range of 0 to 1. The higher value represents the more relevant relation [7]. We combine the user-item matrix and item-tag matrix to draw users' preference about tags using matrix product, and then appropriately normalize the results. This is the formula to represent the above process.

$$\boldsymbol{T}(u,t) = \frac{1}{N}\sum_{k=1}^{n}\boldsymbol{U}(u,k) \times \boldsymbol{G}(k,t) \qquad (1)$$
$$\forall u = 1,2,\dots,m \ and \ \forall t = 1,2,\dots,p$$

Let N be a normalization factor, which is the number of k such that both $\boldsymbol{U}(u,k)$ and $\boldsymbol{G}(k,t)$ are not zero.

3.2 Collective Matrix Factorization

Collective matrix factorization is the method that factorizes given matrix into several sub matrices using latent factors. The result of this process is sub matrices approximating given matrix most closely. To do this, we initially decompose the user-item matrix into two sub matrices. We call the first sub matrix X, second sub matrix Y. X represents relations between

[1] http://www.grouplens.org

users and latent factors, Y is about items and latent factors. Likewise, we also decompose the user-tag matrix into two sub matrices, X and Z. X is same as the previous case, and Z represents relations between tags and latent factors. Using these sub matrices, we can factorize the user-item matrix to XY^T and the user-tag matrix to XZ^T. Then, it is assumed that the information can flow from the user-tag matrix to user-item matrix, because they share same X. Following figure depicts the overview of above process.

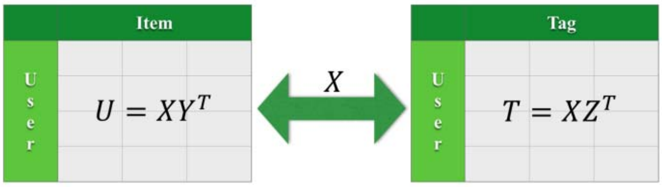

Fig. 1. Overview of proposed method

The goal of this module is finding the X, Y and Z that make XY^T, XZ^T close to the user-item matrix and user-tag matrix. In order to that, we formulate an error function,

$$ER(X,Y,Z) = \frac{1}{2}\| J \circ (\boldsymbol{U} - XY^T)\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{T} - XZ^T\|_F^2 + \frac{\beta}{2}(\|X\|_F^2 + \|Y\|_F^2 + \|Z\|_F^2), \qquad (2)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. $J \in \{0,1\}^{u \times i}$ is the corresponding indicator matrix, with $J(a,b) = 1$ if user a has rated item b, and $J(a,b) = 0$ otherwise. The first term of above function draws the error between given user-item matrix and factorized user-item matrix, and the second term represents the error between given user-tag matrix and factorized user-tag matrix. The last term is a generalization factor, which has a role of preventing overfitting of the error function. In this equation, α and β control the influence of corresponding terms.

The sub matrices minimizing the error function are the best factors of the user-item matrix. To find minimum of (2), we use the Gradient Descant Method (GDM). This is because (2) is not convex, so we cannot find the minimum of the function directly. To use the GDM, we introduce gradients of each variable. The followings are gradients of each variable in GDM.

$$\begin{aligned} \nabla_X ER &= [J \circ (XY^T - \boldsymbol{U})]Y + \alpha(XZ^T - \boldsymbol{T})Z + \beta X \\ \nabla_Y ER &= [J \circ (XY^T - \boldsymbol{U})]^T Y + \beta Y \\ \nabla_Z ER &= \alpha(XZ^T - \boldsymbol{T})^T X + \beta Z \end{aligned} \qquad (3)$$

Using above gradients, we can use GDM to find the minimum of (2). The details of GDM algorithm are given Figure 2.

3.3 Predict Ratings

Using the results of GDM that are sub matrices of given user-item matrix, we can predict unknown ratings. In other words, the ratings in the result matrix become the predicted ratings. The blanks, part of the original user-item matrix, are filled by information flow between user-item matrix and user-tag matrix in the process of collective matrix factorization. In the next section, we evaluate predicted ratings by comparing with ratings in the test set.

Gradient Descent Method

Initialize X,Y,Z with random number in range of (0,1)

while($ER_t - ER_{t+1} > \epsilon$ && iteration < max_iteration)

  Get the gradients using (3)

  $\gamma = 1$

  while($ER(X_t - \gamma\nabla_{Xt}, Y_t - \gamma\nabla_{Yt}, Z_t - \gamma\nabla_{Zt}) > ER(X_t, Y_t, Z_t)$)

    $\gamma = \gamma/2$

  End while

  $X_{t+1} = X_t - \gamma\nabla_{Xt}$

  $Y_{t+1} = Y_t - \gamma\nabla_{Yt}$

  $Z_{t+1} = Z_t - \gamma\nabla_{Zt}$

End while

return XY^T

Fig. 2. Pseudo code of proposed method

IV. Experiments

In this section, we describe the data sets, experimental methodology, evaluation metrics and experimental results of our proposed method.

4.1 Data Set

We demonstrate our approach in the domain of movie recommendation. In experiments, there are two matrices to be required; the user-item matrix and user-tag matrix. MovieLens dataset is used as ratings of the user-item matrix. Since items in this data do not correspond with those of the Tag-genome (item-tag matrix), it is required to extract items that appear in both user-item matrix and item-tag matrix. In addition, we construct the user-tag matrix combining the user-item matrix and item-tag matrix. The data set used in the experiments is summarized in Table 1. In our experiments, sparsity is measured by following equation.

$$\text{Sparsity (\%)} = \frac{\#\ of\ nonzero\ elements}{\#\ of\ total\ elements} \times 100 \qquad (4)$$

TABLE 1. SUMMARY OF DATA SET

# of Users	943
# of Items	1,560
# of Tags	1,128
# of Ratings in U	99,569
Sparsity of U	6.77%
Sparsity of T	100%

4.2 Experiment Design

In order to evaluate the proposed method, we design three experiments. In the first part of our experiments, we use the collective matrix factorization using no additional information, which is the baseline of our experiments. In this experiment, we estimate appropriate β that is the factor of generalization term. After finding β that minimizes the error function, we use this value in other experiments. In the second experiment, we use our proposed method that is the collective matrix factorization with the user-tag matrix. To evaluate proposed method, we apply 5-fold cross validation. We predict the ratings in the test data set only using information of training data set and measure the accuracy of our proposed method. To find the α minimizing the error function, several experiments are performed, changing this value. Finally, we carry out several experiments with fixed α and β to analyze the accuracy of recommender system in varying sparsity condition. During all experiments, the maximum iteration of GDM is limited to 200, and the number of latent factors is assigned to 3.

4.3 Experimental Results

To measure statistical accuracy, we adopt an evaluation metric: Mean Absolute Error (MAE) is defined as the average of absolute difference between predicted ratings and actual ratings.

$$\mathrm{MAE} = \frac{1}{|D|}\sum_{(u,i,r)\in D}|\boldsymbol{U}(u,i)-r| \tag{5}$$

D refers to the test data set, u is the user, i is the item and r is the user u's actual rating about item i. $\boldsymbol{U}(u,i)$ is the predicted rating, which is the result of the collective matrix factorization. Note the smaller value means the better the performance for MAE.

1) Baseline

We employ one baseline: the collective matrix factorization without auxiliary data. To evaluate the baseline, we set α as zero, which means that the user-tag matrix is not considered. Then, we run the CMF with β varying from 0.1 to 100 to examine the influence of β. Following Table 2 is the result of experiment 1.

TABLE 2. PERFORMANCE OF BASELINE

$\boldsymbol{\beta}$	***0.1***	***1***	***10***	***100***
MAE	0.8539	<u>0.8368</u>	0.9395	2.3507

Sparsity = 2%

As can be seen from the Table 2, the performance is improved in the beginning of the experiment, but it is degraded after the critical point. This observation implies that sub matrices are generalized excessively.

2) Proposed method

After finding the suitable β, we evaluate our proposed method about different α. We perform the experiments fixing the β to 1 according to the prior experiment. Following Table 3 is the result of experiment 2.

TABLE 3. PERFORMANCE OF PROPOSED METHOD

α	**MAE**
0.2	0.8313
0.4	0.8034
0.6	0.8003
0.8	0.8004
1	<u>0.7998</u>
10	0.8887
100	2.5011

Sparsity = 2%

It is interesting to be note that, the performance is increased in the early stage of the experiments, but degraded after the critical point in terms of α. This may be because the impact of user-tag matrix is overestimated.

3) Result in varying sparsity

We compare the performance of our proposed method with the baseline in varying sparsity to show whether our method performs better or not. As factors of the error function, we set α and β as 1 according to results of prior experiments. Fixing two factors, we evaluate the baseline and our proposed method, changing the sparsity ratio. Table 4 presents the percentage of improvement of the proposed method over the baseline in terms of MAE.

TABLE 4. PERFORMANCE IN DIFFERENT SPARSITY

Sparsity	**1%**	**2%**	**3%**	**4%**
Baseline ($\boldsymbol{\alpha = 0, \beta = 1}$)	0.9262	0.8368	0.7862	0.7522
Proposed method ($\boldsymbol{\alpha = 1, \beta = 1}$)	0.8566	0.7998	0.7802	0.7483
Improvement (%)	7.5146	4.4216	0.7632	0.5185

4.4 Discussion

As shown in the table 4, our method consistently outperforms the baseline. According to several experiments, we can make the following observations.

- The sparser user-item matrix is given, the more our proposed method improves the accuracy. According to that, we could discover that the user-tag matrix is useful information to alleviate the sparsity problem.

- The performance of our method is affected by α and β, the impact factors of user-tag matrix and generalization. We determine these two factors from several experiments. However, we cannot assure these values of optimizing factors, because these factors may be dependent on given input or target domains.

V. Conclusion

In this paper, we present a collective matrix factorization method with the tag information to solve the sparsity problem in CF. The motivation is that the tag information is highly related to the user-item matrix and it can propagate the user' preferences to sparse user-item matrix. Experimental results show that tag information is considerably useful in recommender systems and the collective matrix factorization with this information performs significantly better than general collective matrix factorization method.

In conclusion, we demonstrate that the collective matrix factorization method with tag information addresses the sparsity problem in CF. In the future, we will study on how to create useful auxiliary information to improve current methods, and how to extend our method in other domains.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2012R1A1A2008062) and IT R&D program of MOTIE/KEIT (10041244, Smart TV 2.0 Software Platform). Also, we thank GroupLens research group for providing data set.

References

[1] Adomavicius, Gediminas, and Alexander Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," Knowledge and Data Engineering, IEEE Transactions, vol. 17, issue 6, pp. 734-749, 2005.

[2] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, Qiang Yang, "Transfer Learning in Collaborative Filtering for Sparsity Reduction," AAAI, vol. 10, pp. 230-235, 2010.

[3] Bobadilla J, Ortega F, Hernando A, and Gutiérrez A, "Recommender systems survey," Knowledge-Based Systems, vol. 46, pp. 109-132. 2013.

[4] Dai, Yae, HongWu Ye, and SongJie Gong, "Personalized recommendation algorithm using user demography information," Knowledge Discovery and Data Mining, pp. 100-103, 2009.

[5] Pazzani, Michael J, "A framework for collaborative, content-based and demographic filtering," Artificial Intelligence Review, vol. 13, issue 5-6, pp. 393-408, 1999.

[6] Vincent W. Zheng, Yu Zheng, Xing Xie, Qiang Yang, "Collaborative location and activity recommendations with GPS history data," WWW '10 Proceedings of the 19th international conference on World wide web, pp. 1029-1038, 2010.

[7] Vig, Jesse, Shilad Sen, and John Riedl, "The tag genome: Encoding community knowledge to support novel interaction," ACM Transactions on Interactive Intelligent Systems, vol. 2, issue 3, no. 13, 2012.

[8] Melville, Prem, Raymond J. Mooney, and Ramadass Nagarajan, "Content-boosted collaborative filtering for improved recommendations," AAAI/IAAI, pp. 187-192, 2002.

[9] Lekakos, George, and Petros Caravelas, "A hybrid approach for movie recommendation," Multimedia tools and applications, vol. 36, issue 1-2, pp. 55-70, 2008.

[10] Koren, Yehuda, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 30-37, 2009.

[11] Singh, Ajit P., and Geoffrey J. Gordon, "Relational learning via collective matrix factorization," Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 650-658, 2008.