2017 25th Iranian Conference on Electrical Engineering (ICEE)

A clustering-based matrix factorization method to enhance the diversity of recommendation systems

Zahra Shajarian
Department of Computer Engineering
University of Kurdistan
Sanandaj, Iran
z.shajarian@eng.uok.ac.ir

Seyed Amjad Seyedi
Department of Computer Engineering
University of Kurdistan
Sanandaj, Iran
amjadseyedi@eng.uok.ac.ir

Parham Moradi
Department of Computer Engineering
University of Kurdistan
Sanandaj, Iran
p.moradi@uok.ac.ir

***Abstract*— Matrix approximation is a common model-based approach to collaborative filtering in recommender systems. However, due to data sparsity, it is difficult for current approaches to accurately approximate unknown rating values, which may cause low-quality recommendations. In this paper, we proposed a modified latent factor model to provide diverse and accurate recommendations. The proposed method incorporates two conflicting concepts, accuracy and diversity, in a unified optimization framework to predict the missing ratings. The proposed method con be able to overcome data sparsity and also improving matrix approximation by introducing clustering and transfer learning techniques. The performance of the proposed method was evaluated on two real-world benchmarks and results show its superiority compare to the state-of-the-art methods.**

Keywords; Recommender systems; Matrix factorization; Collaborative filtering; matrix approximation; Diversity

I. Introduction

Rapid development of e-commerce websites, leads to overloading of huge quantities of products. Therefore, people needs to find out their interested products in an easy way. Recommender systems (RS) are the most successful tool to support personalized recommendations[1, 2]. RSs are widely used in different scenarios such as e-commerce[3], e-learning[4], e-government[5], tourism[6] and web pages[7]

Recommender system methods can be classified into content-based and the collaborative filtering methods. Content-based (CB) methods recommend items that have similar features with those of previously purchased items to a user[8, 9]. While, collaborative filtering (CF) methods recommend suggested items that other similar users liked in the past[10, 11].

Thus, in CF-based methods, the opinions of the other users with similar tastes are used to provide recommendations for a target user. In other words, the basic idea of the CF-based methods is that two users with similar ratings are likely to prefer similar items[10]. Among the others, matrix factorization methods have been successfully employed in collaborative filtering approach. CB-based methods are also classified into memory–based and model-based methods. Memory-based recommender systems explicitly employed rating matrix in their computations, while model-based methods are based on the idea of using a model to predict user ratings. Since such recommender systems involve scalable algorithms for recommending, they are very advisable when using large recommender systems with a great number of users or items.

Matrix factorization is well-known model-based method. In a basic form, matrix factorization methods represent both items and users by vectors of factors inferred from item rating patterns. These methods offer much scalability and flexibility for modeling various real-world scenarios, thus they become popular in recent years. Up to now, several MF-based methods have been proposed and successfully applied to CF recommenders. For example, the authors of [12] proposed probabilistic latent semantic analysis to build a CF model. The CF recommender based on regularized MF[13] has shown high performance and won the Netflix-prize-competition. Since then, many MF variants have been developed and gained industrial applications. These works include maximum-margin MF-based model[14], regularized MF[15], the expectation maximization (EM)-based MF model[16], biased SVD model[17], biased regularized incremental simultaneous MF model[18], SVD++ model[19] , probabilistic MF model[20], imputation based MF[21] and nonparametric MF model[22].

Recently, Gogna and Majumdar[23] proposed a matrix factorization method for designing an effective recommender system to provide rich diversity recommendations while maintaining necessary relevance to user's choice. Their model is able to improve diversity from user's and also improve the visibility of less popular items. Besides, the authors of [24] proposed a reconstructive method to overcome data sparsity in RS and also improving matrix approximation by introducing clustering and transfer learning techniques. Taking advantages of these methods, in this work, we proposed a modified latent factor model to provide diverse and accurate recommendations. The proposed method incorporates two conflicting concepts, accuracy and diversity, in a unified optimization framework to predict the missing ratings.

II. Matrix Factorization

The aim of matrix factorization is to map both users and items to a joint latent factor space. Using this model, users and items are respectively represented by $V \in \mathbb{R}^f$ and $U \in \mathbb{R}^f$ vectors. The overall user's interest in the item's characteristics is obtained from dot product, $V^T U$. This product, approximates user u's rating of item *i*, which is denoted as:

$$\hat{R}_{ui} = U_u V_i^T \tag{1}$$

The major challenge is computing the mapping of each users and items their corresponding factor vectors. After completing this mapping, using Eq.(1), it can easily estimate the rating a user will give to any item.

Earlier systems relied on imputation to fill in missing ratings and make the rating matrix dense. However, imputation can be very expensive in the presence of huge amount of data. To this end, recent works suggested regularized models to avoid overfitting as follows:

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{i,j} - U_i^T V_j)^2 + \lambda_u \|\mathrm{U}\|_F^2 + \lambda_v \|\mathrm{V}\|_F^2 \tag{2}$$

where I is identity matrix, λ_u and λ_v are regularization coefficients.

III. PROPOSED METHOD

This section presents the proposed clustering-based matrix factorization method for improving diversity and scalability of recommendations. The proposed method consists of three steps including; (1) Factorization, (2) Clustering and (3) Approximation. The pseudo-code of the proposed method is provided in Algorithm 1. The details of each step are provided in its corresponding section.

Algorithm 1. Pseudo-code proposed method

Input: *R*: ratings matrix $n \times m$, *n* number of users, *m* number of items, *k* number of user classes, *l* number of item classes, F number of factors

Output: $\hat{R}$: recommendation matrix $n \times m$

Begin algorithm

1: Calculate *U* and *V* using Eqs (4) and (5).
2: C_U= ***kmeans*** *(U,k)* , C_V=***kmeans*** *(V,l)*
3: $B = C_U \times C_V^T$
4: Calculate M_U and M_V using *Algorithm* 1
5: Calculate $\hat{R}$ using Eq.(6)

End algorithm

A. Factorization

Traditional ranking strategy proposed by previous matrix factorization methods, generates a recommendation list characterized by high accuracy but very limited diversity. In this paper, in order to provide high diversity while maintaining necessary relevance to users' preferences, the basic MF model presented in Eq.(2) is modified by incorporating additional penalty to enhance the diversity of recommendations. To this end, an additional term which minimizes the variance amongst the elements of a user's latent factor vector is added to the base formulation as follows:

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{i,j} - U_i^T V_j)^2 \\ &+ \lambda_u \|\mathrm{U}\|_f^2 + \lambda_v \|\mathrm{V}\|_f^2 + \lambda_d \sum_{u \in Users} \mathrm{var}(\mathrm{U}_u)\end{aligned} \tag{3}$$

Where, *var(Uu)* is the variance of the latent factor vector representing user u and is given by $\mathrm{var}(\mathrm{U_u}) = \sum_{l=1}^{f}(\mathrm{U_u}(l) - \mathrm{m_u})^2$ where, f is the number of latent factors, lth element of user u's latent factor vector; $\mathrm{m_u} = \frac{1}{\mathrm{F}}\sum_{l=1}^{\mathrm{F}} \mathrm{U_u}(l)$ is the mean of latent factor vector and λ_d is the regularization coefficient.

The use of variance minimization term in the objective function, prevents the predicted ratings for each user from being biased towards a particular item group, leads to produce diverse recommendations. This means that the predicted ratings of a user, show uniform distribution across all features resulting in a recommendation list populated with items characterized by varied features.

In this paper, gradient descent method is performed to find proper values for U and V matrixes which minimize the above-mentioned objective function (i.e., Eq.(3)), . The following equations (i. e. Eqs. (4) and (5)) gained the update of U_i and V_j. Additional details are provided in Appendix A.

$$U_i \leftarrow U_i - \gamma_u \sum_{j=1}^{m} I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_u U_i + \lambda_d D^2 U_i \tag{4}$$

$$V_j \leftarrow V_j - \gamma_v \sum_{i=1}^{n} I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_v V_j \tag{5}$$

B. Clustering

The aim of Factorization step was to obtain both U and V matrixes. Using Eq.(1), by producing U and V, unknown rates will be estimated. Note that, due to availability of limited number of ratings for each user in real-world systems, the rating matrix is very sparse and thus the direct approximation may not accurate. In other words, constructing an accurate approximation requires the suitable rating-pattern. Thus, it would be very difficult to approximate rating-patterns directly from the sparse rating. Based on the assumption that users or items with similar latent factor vector behave very similarly, the rating-patterns can be constructed indirectly using *U* and *V*. To this end, k-means clustering algorithm is applied on the row vectors of *U* and *V*, the results are respectively $C_u \in \mathbb{R}^{k \times d}$ and $C_v \in \mathbb{R}^{l \times d}$, where *k* is the number of user clusters and *l* is the number of item clusters. The product of user and item clusters (i.e. $B = C_U \times C_V^T$) gives the cluster-level rating-pattern. The result, denoted by *B* indicates a user who belongs to a specific user cluster will give to an item that belongs to a specific item cluster.

C. Approximation

The aim of this section is to approximate the rating matrix by using cluster-level rating patterns. In this step, we followed the same methodology proposed in[24]. To approximate. To this end, the following equation is used to construct an approximation rating matrix as follows:

$$\tilde{R} = W \odot X + [1 - W] \odot \left[M_u B M_v^T \right] \tag{6}$$

where B is rating patterns, $M_u \in R^{m \times k}$ is user membership values, $M_v \in R^{n \times l}$ is item membership values and $\odot$ denotes the entry-wise product. M_u and M_v are binary matrices, where “1” indicates a membership of user or item to a cluster. Besides, W is the weighting and has the same size as R, where $W_{ij} = 1$ if R_{ij} is rated and $W_{ij} = 0$ otherwise. In the object function, W ensures that the error is calculated based only on the observed entries. We have used the same strategy proposed in[25] to solve to calculate membership values of Eq.(6), the details are provide in Algorithm 2.

Algorithm 2. Calculate membership values

Input: $R_{n \times m}$ rating matrix, $B_{k \times l}$ pattern rating, V matrix, Cv
Output: M_U and M_V
Begin algorithm

V'=**calculateMembership** (C_v, V)
$[M_V^{(0)}]_{i\hat{\jmath}} = V'$
for $t = 1$ **to** T **do**
  for $i = 1$ **to** n **do**
    $\hat{\jmath} = argmin_j \left\| [R]_{i*} - \left[B \left[M_v^{(t-1)} \right]^T \right]_{j*} \right\|_{W_{i*}}^2$
    $\left[M_u^{(t)} \right]_{i\hat{\jmath}} = 0;\ [M_u^{(t)}]_{ij} = 0,$ for $j \in \{1, \ldots, k\} \backslash \hat{\jmath}$
  end for
  for $i = 1$ **to** m **do**
    $\hat{\jmath} = argmin_j \left\| [R]_{*i} - \left[M_u^{(t)} B \right]_{*j} \right\|_{W_{*i}}^2$
    $\left[M_v^{(t)} \right]_{i\hat{\jmath}} = 1;\ [M_v^{(t)}]_{ij} = 0$ for $j \in \{1, \ldots, l\} \backslash \hat{\jmath}$
  end for
end for
End algorithm

D. An illustrated example

In this section, an example is provided to describe different steps of the proposed method. Fig. 1(a) shows a sparse rating matrix. The miss values are denoted by ‘*”. The users and items latent factors are obtained by using Eq.(4) and (5) respectively. The in the next step, k-means clustering algorithm is applied on both U and V matrixes, the results are represented by C_v and C_u in Fig. 1(b). The rating patterns is then obtained by production of user and item clusters ($B = C_u C_v^T$). Then in the third step the membership values are obtained by applying Algorithm 2 and the final rating matrix is approximated using Eq.(6). Fig. 1(3) shows the results of step 3.

$$R = \begin{bmatrix} 1 & 3 & \bullet & \bullet \\ 1 & 2 & \bullet & 3 \\ \bullet & \bullet & 3 & 1 \\ 1 & 3 & \bullet & 1 \\ 1 & \bullet & \bullet & 3 \\ \bullet & 2 & \bullet & 3 \end{bmatrix}_{6\times4} \quad U = \begin{bmatrix} 0.12 & 0.29 & 0.77 \\ 0.25 & 0.11 & 0.58 \\ 0.15 & 0.14 & 0.64 \\ 0.33 & 0.22 & 0.10 \\ 0.73 & 0.00 & 0.70 \\ 0.73 & 0.27 & 0.38 \end{bmatrix}_{6\times3} \quad V = \begin{bmatrix} 0.28 & 0.75 & 0.30 \\ 0.45 & 0.88 & 0.71 \\ 0.92 & 0.28 & 0.47 \\ 0.55 & 0.78 & 0.17 \end{bmatrix}_{4\times3}$$

(a)

$$C_u = \begin{bmatrix} 0.12 & 0.29 & 0.77 \\ 0.33 & 0.22 & 0.10 \\ 0.73 & 0.01 & 0.70 \end{bmatrix}_{3\times3} \quad C_v = \begin{bmatrix} 0.28 & 0.75 & 0.30 \\ 0.92 & 0.28 & 0.47 \end{bmatrix}_{2\times3} \quad B = \begin{bmatrix} 0.49 & 0.57 \\ 0.30 & 0.44 \\ 0.42 & 1.02 \end{bmatrix}_{3\times2}$$

(b)

$$M_u = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}_{6\times3} \quad M_v = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}_{2\times4} \quad \hat{R} = \begin{bmatrix} 1 & 3 & 3.05 & 1.27 \\ 1 & 2 & 1.73 & 3 \\ 1.27 & 3.05 & 3 & 1 \\ 1 & 3 & 3.05 & 1 \\ 1 & 1.73 & 1.73 & 3 \\ 1.49 & 2 & 1.73 & 3 \end{bmatrix}_{6\times4}$$

(c)

Figure 1. An example to describe different steps of the propsoed method

IV. Experiment results

In this section, several experiments are performed in order to evaluate the performance of the proposed method. Our aim is to assess the performance of the proposed method in terms of prediction. All the algorithms were implemented in Matlab R2016b on a personal computer equipped with a Core i5 processor and 8 GB RAM.

A. Dataset

To perform an evaluation, the methods are evaluated using three well-known datasets in the field of CFRS: MovieLens (ml-100k)[26] and Epinions[27] datasets. Each dataset has specific features (see Table I) that affect the performance of different techniques, such as sparsity or the relation between the number of users and items.

TABLE I. SOCIAL NEWTWORK DATASETS

Dataset	Users	Items	Ratings	Sparsity
MovieLens-100k	943	1,682	100,000	0.9369
Epinions	488	43,971	65,536	0.9969

B. *Metrics*

The main purpose of RSs is to predict rates of new items for a test user that is called 'active user'. We chose two metrics, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), to measure the closeness of predicted ratings to the actual ones. These metrics are defined as follows:

$$MAE = \frac{1}{|R_{test}|} \sum_{R_{ij} \in R_{test}} \left| R_{ij} - \hat{R}_{ij} \right| \tag{7}$$

$$,RMSE = \sqrt{\frac{1}{|R_{test}|} \sum_{R_{ij} \in R_{test}} \left(R_{ij} - \hat{R}_{ij} \right)^2} \tag{8}$$

Where R_{test} denotes the test set; R_{ij} and $\hat{R}_{ij}$ denote the true and predicted rating values given from user i to item j, respectively. For given recommender, lower values of MAE and RMSE correspond to higher prediction accuracy.

TABLE II. MAE AND RMSE VALUES FOR SGD, JI ET AL METHOD AND OUR METHOD IN MOVIELENS-100K DATASET FOR DIFFERENT NUMBER OF LATENT FEATURES (K)

Parameter	Error metrics	Methods		
		SGD[28]	Ji et al.[24]	proposed method
F=2	MAE	8.6817	0.7544	**0.7360**
	RMSE	4.7078	0.9584	**0.9392**
F=5	MAE	6.8383	0.7878	**0.7816**
	RMSE	3.8514	1.0068	**0.9953**
F=7	MAE	3.4568	0.7820	**0.7793**
	RMSE	2.0519	**0.9945**	0.9974
F=10	MAE	4.5330	0.7936	**0.7905**
	RMSE	2.7110	1.0099	**1.0096**

C. *Comparisons*

As to evaluate the effectiveness of our method, we compare our method with the popular method, SGD[28] and state-of-the-art matrix factorization model, Ji et al[24]. The experiments were performed for different number of latent features (F=2, 5,7 and 10). Tables II and III report MAE and RMSE for MovieLens an Epinions datasets. For all datasets 80% of rating data is used as the training set and the rest 20% is the test set. Regarding the overall results our method except in one case achieved the less error for all cases.

TABLE III. MAE AND RMSE VALUES FOR SGD, JI ET AL METHOD AND OUR METHOD IN EPINIONS DATASET FOR DIFFERENT NUMBER OF LATENT FEATURES (F)

Parameter	Error metrics	Methods		
		SGD[28]	Ji et al.[24]	proposed method
F=2	MAE	8.7763	0.9990	**0.9356**
	RMSE	4.7820	1.3043	**1.2223**
F=5	MAE	5.1332	1.0515	**0.9511**
	RMSE	2.9325	1.3517	**1.2218**
F=7	MAE	5.1057	1.0887	**0.9834**
	RMSE	2.9262	1.3951	**1.2402**
F=10	MAE	5.5341	1.1790	**0.9482**
	RMSE	3.2807	1.5247	**1.1867**

D. *Parameter settings*

In the experiments for Ji et al.[24]method, the regularization parameters λ_u and λ_v were set at 0.01 and 0.01 on Epinion, 0.001 and 0.01 on MovieLens. The regularization parameter for SGD method was set at 0.001 on MovieLens and 0.01 on Epinion. In the experiments for proposed method, the regularization parameters λ_u, λ_v and λ_d were set at 0.01, 0.01 and 100 on Epinion, 0.001, 0.01 and 100 on MovieLens.

Figure 2 shows how rate of diversity affects the recommendation. A very small or very large leads to inappropriate recommendations. Several experiments were performed on two datasets. From the results, it can be seen that proposed method obtained its highest accuracy when λ_d is set to 10^2.

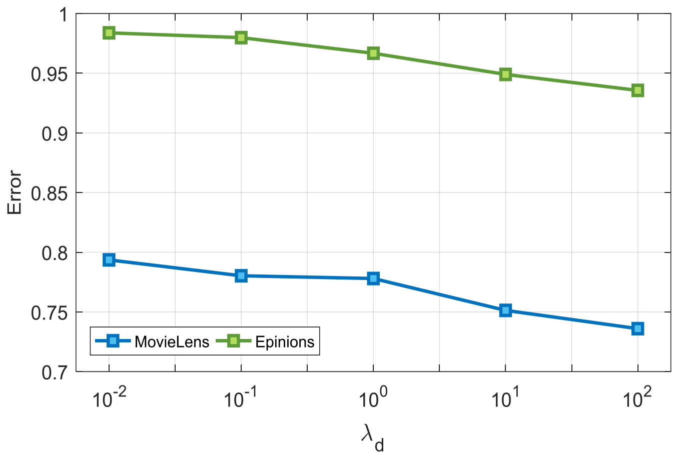

Figure 2. Sensitivity of λ_d parameter

V. CONCLUSION

This paper studies the crucial problem on how to overcome the data sparsity for matrix approximation. We proposed a reconstructive method for improving matrix approximation by innovatively introducing clustering and transfer learning techniques. Our idea is to implicitly find cluster-level rating-pattern based on the latent factor vectors of users and items, and construct an improved approximation by mapping all users/items to the corresponding user/item clusters.

Experiments on real datasets show that our method obviously improves the accuracy of matrix approximation learnt by the state-of-the-art matrix factorization and social recommendation models. This fact verifies that banding together with a bunch of latent factor vectors of similar users/items to generate the prototype for each user/item cluster is an effective way to alleviate data sparsity problem in latent factor models. In this work, we propose a novel formulation for designing an effective recommender system which provides rich diversity while maintaining necessary relevance to user's choice. We are able to achieve improved diversity from user's as well as system's perspective and also improve the visibility of less popular items.

APPENDIX A

For performing gradient descent method:

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{i,j} - U_i^T V_j)^2 + \lambda_u \|\mathrm{U}\|_F^2 + \lambda_v \|\mathrm{V}\|_F^2 + \lambda_d \sum_{u\in Users} \mathrm{var}(\mathrm{U}_u) \tag{A.1}$$

Where has the following expression:

$$\sum_{u\in Users} \mathrm{var}(U_u) = \sum_{l=1}^{F} (U_u(l) - m_u) \tag{A.2}$$

$$, m_u = \frac{1}{F}\sum_{l=1}^{F} U_u(l) \tag{A.3}$$

Coming back to our formulation (3), the variance minimization regularization term can be recast in matrix form as follows:

$$\left\| \begin{bmatrix} U_1(1) & U_1(2) & . & U_1(F) \\ U_2(1) & U_2(2) & . & . \\ . & . & . & . \\ U_m(1) & . & . & U_m(F) \end{bmatrix} - \frac{1}{F} \begin{bmatrix} U_1(1) & U_1(2) & . & U_1(F) \\ U_2(1) & U_2(2) & . & . \\ . & . & . & . \\ U_m(1) & . & . & U_m(F) \end{bmatrix} \bar{1}_{F\times F} \right\|_F^2 \tag{A.4}$$

where m is the number of users; $\bar{1}_{F\times F}$ is a F×F matrix of all $1'$s.

Using (A.4) in Eq.(3), we can represent our proposed formulation as in Eq.3 where, $D = I_{F\times F} - \frac{1}{F}\bar{1}_{F\times F}$ and $I_{F\times F}$ is an identity matrix of dimension F×F

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(R_{i,j} - U_i^T V_j)^2 + \lambda_u \|\mathrm{U}\|_F^2 + \lambda_v \|\mathrm{V}\|_F^2 + \lambda_d \|DU\|_F^2 \tag{A.5}$$

Given the rating matrix R, a local minimum of the objective function given by Eq.3 can be found by performing gradient descent in feature vectors U_i and V_j:

$$\frac{\partial\mathcal{L}}{\partial U_i} = \sum_{j=1}^{m} I_{ij}(U_i^T V_j - R_{ij})V_j + \lambda_u U_i + \lambda_d D^2 U_i \tag{A.6}$$

$$, \frac{\partial\mathcal{L}}{\partial V_j} = \sum_{i=1}^{n} I_{ij}(U_i^T V_j - R_{ij})U_i + \lambda_v V_j \tag{A.7}$$

REFERENCES

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

[2] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Modeling and User-Adapted Interaction,* vol. 22, pp. 101-123, 2012.

[3] D. Wu, G. Zhang, and J. Lu, "A fuzzy preference tree-based recommender system for personalized business-to-business e-services," *IEEE Transactions on Fuzzy Systems,* vol. 23, pp. 29-43, 2015.

[4] D. Wu, J. Lu, and G. Zhang, "A fuzzy tree matching-based personalized e-learning recommender system," *IEEE Transactions on Fuzzy Systems,* vol. 23, pp. 2412-2426, 2015.

[5] X. Guo and J. Lu, "Intelligent e-government services with personalized recommendation techniques," *International Journal of Intelligent Systems,* vol. 22, pp. 401-417, 2007.

[6] J. M. Noguera, M. J. Barranco, R. J. Segura, and L. Martínez, "A mobile 3D-GIS hybrid recommender system for tourism," *Information Sciences,* vol. 215, pp. 37-52, 12/15/ 2012.

[7] G. Castellano, A. M. Fanelli, and M. A. Torsello, "NEWER: A system for NEuro-fuzzy WEb Recommendation," *Applied Soft Computing,* vol. 11, pp. 793-806, 1// 2011.

[8] Y. A. Kim and R. Phalak, "A trust prediction framework in rating-based experience sharing social networks without a Web of Trust," *Information Sciences,* vol. 191, pp. 128-145, 5/15/ 2012.

[9] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 325-341.

[10] P. Moradi and S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems," *Expert Systems with Applications*, vol. 42, pp. 7386-7398, 11/30/ 2015.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer,* vol. 42, pp. 30-37, 2009.

[12] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems (TOIS),* vol. 22, pp. 89-115, 2004.

[13] G. Gorrell, "Generalized Hebbian Algorithm for Incremental Singular Value Decomposition in Natural Language Processing," in *EACL*, 2006, pp. 97-104.

[14] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Advances in neural information processing systems*, 2004, pp. 1329-1336.

[15] E. Bojnordi and P. Moradi, "A novel collaborative filtering model based on combination of correlation method with matrix completion technique," in *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, 2012, pp. 191-194.

[16] M. Kurucz, A. A. Benczúr, and K. Csalogány, "Methods for large scale SVD with missing values," in *Proceedings of KDD cup and workshop*, 2007, pp. 31-38.

[17] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, 2007, pp. 5-8.

[18] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *Journal of machine learning research,* vol. 10, pp. 623-656, 2009.

[19] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender systems handbook,* ed: Springer, 2011, pp. 145-186.

[20] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization."

[21] 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.

[22] K. Yu, S. Zhu, J. Lafferty, and Y. Gong, "Fast nonparametric matrix factorization for large-scale collaborative filtering," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 211-218.

[23] A. Gogna and A. Majumdar, "DiABlO: Optimization based design for improving diversity in recommender system," *Information Sciences,* vol. 378, pp. 59-74, 2/1/ 2017.

[24] K. Ji, R. Sun, X. Li, and W. Shu, "Improving matrix approximation for recommendation via a clustering-based reconstructive method," *Neurocomputing*, vol. 173, Part 3, pp. 912-920, 1/15/ 2016.

[25] B. Li, Q. Yang, and X. Xue, "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction," in *IJCAI*, 2009, pp. 2052-2057.

[26] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.,* vol. 5, pp. 1-19, 2015.

[27] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*, 2003, pp. 351-368.

[28] C.-J. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization," *Neural Computation,* vol. 19, pp. 2756-2779, 2007/10/01 2007.