ISSN No. 0976-5697

DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.4897

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

CROSS DOMAIN COLLABORATIVE FILTERING RECOMMENDER USING PROBABILISTIC MATRIX FACTORIZATION

Nazima Khanam
Department of CSE
Chaitanya Bharathi Institute of Technology
Hyderabad, India

M. Venu Gopalachari
Department of CSE
Chaitanya Bharathi Institute of Technology
Hyderabad, India

Abstract: Recommender systems (RS) are targeted towards users who lack sufficient experience to evaluate the overwhelming number of alternative examples that a system may offer. Collaborative Filtering RS is one of the approach that provide recommendations without taking into account the contents of the items being recommended however, they face several challenges such as cold start, data sparsity, low confidence etc. Of late there has been considerable interest in Cross Domain RS, where we exploit knowledge from auxiliary domains which contains additional user preference data to improve recommendation on target domains. This paper, focus on Probabilistic Matrix Factorization (PMF) model in Cross Domain Recommender (CDR) that outperforms on other model based Collaborative Filtering recommenders. Experiments are conducted on benchmark datasets shown significant improvement in the quality of the CDR over various test sets.

Keywords: cross domain; matrix factorization; collaborative filtering; transfer learning; recommender system

I. INTRODUCTION

Recommender systems (RS) are designed to find items of interest, narrow down the set of choices, help the user explore the space of options. The suggestion provided by recommender systems provides additional and probably unique personalized service for the customer. RS are used to filter information and suggestions for items to users, where “Item” is a generic term used to refer what the system refers to the users, as a ranked list of items. Recommender systems come in various types based on the methodology and information used in the process: Content Based Filtering (CB) [1], Collaborative Filtering (CF) [2] and Hybrid Recommender (HR) [3]. CB uses the information of the user or item considered for personalization and accordingly the suggestions get processed. CF methods predict the ratings of a particular user to a given item based on their past preferences and interests along with the rating information provided by similar users and items. Consider a matrix X for an example as shown in figure 1, the rows represents the users and columns represent items, non- zero entries are treated as known ratings, remaining entries represent missing ratings. There are different approaches of CF: Memory based algorithm and Model based algorithm. On the other hand, based on the number of domains involved in the process RS classified as Single Domain RS (SDR) and Multi Domain RS (MDR) [4, 5], where SDR takes input from only one domain and MDR considers from more than one domain for ranking of items. The disadvantages of single domain RS mostly occurred with data sparsity, by which, CF method suffers from lack of available data, and prediction is difficult in single domain [6, 7]. This occurs when more number of new users added to the system, or new items added to the system or when users provide very less rating. This issue is called Cold Start problem i.e. when the system is not able to recommend items to a newly added user.

User/Item	**I_1**	**I_2**	**I_3**	**I_4**
U_1	4	1	2	?
U_2	2	5	?	3
U_3	3	4	?	2
U_4	?	2	1	4

Figure 1: Example user-item matrix

If the input information for RS is from one domain and rankings for items are made in other domain then the RS called to be Cross Domain RS (CDR) [8]. Data sparsity problem can be avoided with CDR by using the available rating information in dense domain, where this available information is extracted and then transferred to target domain, by using which, the recommendations are done in target domain [9]. Cold user problem is avoided by CDR when recommending items to new users by finding similar users in other domains [10].

There are two different types of CDR approaches: linking/aggregating knowledge and sharing/transferring knowledge, based on how the knowledge from the source domain is attained and how it is transferred to the target domain as shown in figure 2. The former one links source and target domain by merging user preferences from both domains and prepares the recommendations accordingly where the later one shares the latent features, rating patterns, of the source domain to the target domain by which the recommendation get prepared in the target.

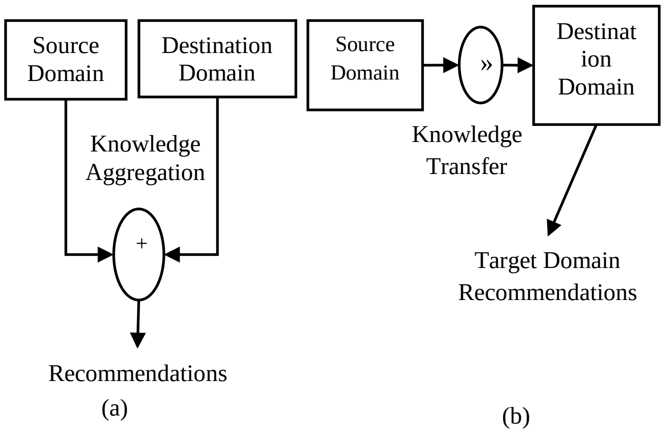


Figure 2: a) Knowledge Linking/Aggregating b) Knowledge Sharing\Transferring

In order to generate recommendations with CDR, the quality of the recommender is highly desirable [11]. However CF approach proved to be the qualitative recommender with model based techniques. In this paper the main focus is on matrix factorization models in CDR, where Probabilistic Matrix Factorization (PMF) model outperforms other model based CF methods. Experiments carried on benchmark data sets shown significant improvement of the recommendations quality for PMF when compared with other models.

The rest of the paper is organized as follows. The related work is presented in the next section and then the problem formulation of PMF is explained. There after results and analysis shows the experimentation part followed by conclusion.

II. RELATED WORK

This section provides different matrix factorization methods in the area of recommender systems and techniques of cross domain recommender systems [12]. The idea behind MF is to predict the rating for any unobserved given user-item pair using latent features identified with user-item matrix [13]. Given a partially observed rating matrix Y ~ $R_{m\times n}$, the aim is to find two matrices U ~ $R_{m\times p}$ and V ~ $R_{p\times n}$ such that Y ~ UV where p is number of latent features. When p is much smaller than either m or n, a factorization allows the matrix to be stored inexpensively, and to be multiplied to vectors or other matrices rapidly. X = UV is called as low-rank approximation of Y . Several general purpose matrix factorization techniques have been proposed in the literature such as Non-negative Matrix Factorization (NMF), Regularized Matrix Factorization (RMF), Positive Matrix Factorization (PoMF) and Maximum Margin Matrix Factorization (MMMF) [14, 15]. NMF attempts to impose a restriction on individual elements of factors U and V as nonnegative elements (instead of introducing a regularization constraint). NMF focused to determine non-negative low-rank matrices U and V which minimizes the following loss function. The objective of RMF is to determine a pair of factors matrices U and V such that the element-wise aggregated squared error for the observed values are minimized. Regularization constraint is added to restrict the domains of U and V to prevent over fitting. In PoMF, the problem of non-optimal scaling has been explicitly addressed. In order to properly scale the data, it is necessary to look explicitly at the problem as a least-squares problem. To begin this analysis, the elements e_{ij} of the "Residual Matrix" E are defined with latent factors U and V as

$$e_{ij} = x_{ij} - \sum u_{ik} \,.\, v_{kj}$$

Initially the problem was solved iteratively using alternating least squares [Paatero and Tapper, 1993]. In alternating least-squares, one of the matrices, U or V, is taken as known and the chi-squared is minimized with respect to the other matrix as a weighted linear-least-squares problem. Then the roles of G and F are reversed so that the matrix that has just been calculated is fixed and the other is calculated by minimizing Q. The process then continues until convergence, however, this process can be slow. MMMF is the method in which, in addition to the latent factor matrices U and V matrices, R - 1 thresholds, θ_{ia} (1 <= a <= R - 1), are also learned for each user i. The prediction for item j by user i is obtained by comparing the real valued predictions U_iV_j against R - 1 thresholds, θ_{ia} (1 <= a <= R - 1), to generate discrete rating values in {1….R}.There are different sets of thresholds for each user and the objective of MMMF is to minimize this objective function with respect to U, V [16].

Prepare **Probabilistic Matrix Factorization**

III. PROBABILISTIC MATRIX FACTORIZATION

For a matrix R which is N×M (where every row represents a user, every column represents a movie), PMF derives a N×D matrix U^T and a D×M matrix V, thus approximates $R=U^T\cdot V$ [17]. The vector V is a factor matrix of items and the vector U^T gives coefficients of these factors for every user.

Given the preference matrix with entries, R_{ij} , we find factorization that minimizes the root mean squared error (RMSE) on the test set. One trial is to use a linear model where we assume that there is Gaussian noise in the data. Define I_{ij} to be 1 if R_{ij} is known (i.e. user i has rated item j) and 0 otherwise.

Also, let $\aleph(x|\mu,\sigma^2) = f_X(x)$ with $X \sim \aleph(\mu,\sigma^2)$.Then, the conditional probability of ratings with σ^2 is as follows.

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\aleph\left(R_{ij}|U_i^T V_j,\sigma^2\right)\right]^{I_{ij}}$$

And priors on U and V with hyper parameters σ_U^2, σ_V^2 are as follows

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\aleph(U_i|0,\sigma_U^2\mathbf{I})$$

$$p(V|\sigma_V^2) = \prod_{i=1}^{M}\aleph(V_i|0,\sigma_V^2\mathbf{I})$$

The objective function E framed to make PMF effective is

$$E = \frac{1}{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij} - U_i^T V_j\right)^2 + \lambda_U \sum_{i=1}^{N}\|U_i\|_F^2 + \lambda_V \sum_{i=1}^{M}\|V_i\|_F^2\right)$$

Where $\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^2$ is the Frobenius norm. Note that if all values are known, $I_{ij} = 1, \forall(i,j)$, then as $\sigma_U^2, \sigma_V^2 \to \infty$, E reduces to the SVD objective function.

Bayesian Probabilistic Matrix Factorization (BPMF)

BPMF is one of the efficient variant of PMF method that considers the prior distributions over the user as well as item feature vectors are assumed to be Gaussian [18].

$$p(U|\mu_U,\Lambda_U) = \prod_{i=1}^{N} \aleph(U_i|\mu_U,\Lambda_U^{-1})$$

$$p(V|\mu_V,\Lambda_V) = \prod_{i=1}^{M} \aleph(V_i|\mu_V,\Lambda_V^{-1})$$

Further place Gaussian-Wishart priors on user and item hyper parameters

$$\Theta_U = \{\mu_U,\Lambda_U\}\ and\ \Theta_V = \{\mu_V,\Lambda_V\}$$

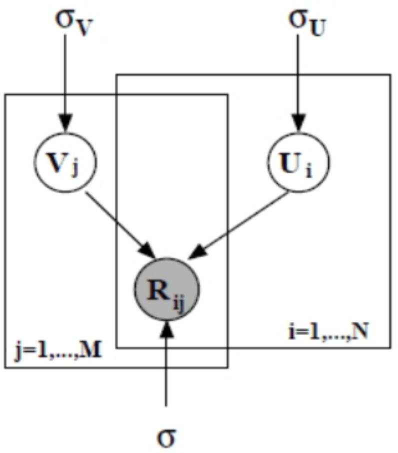
$$p(\Theta_U|\Theta_0) = p(\mu_U|\ \Lambda_U)p(\Lambda_U) = \aleph(\mu_U|\mu_0,(\beta_0\ \Lambda_U)^{-1})W(\Lambda_U\ |W_0,v_0)$$

$$p(\Theta_V|\Theta_0) = p(\mu_V|\ \Lambda_U)p(\Lambda_V) = \aleph(\mu_V|\mu_0,(\beta_0\ \Lambda_V)^{-1})W(\Lambda_V\ |W_0,v_0)$$

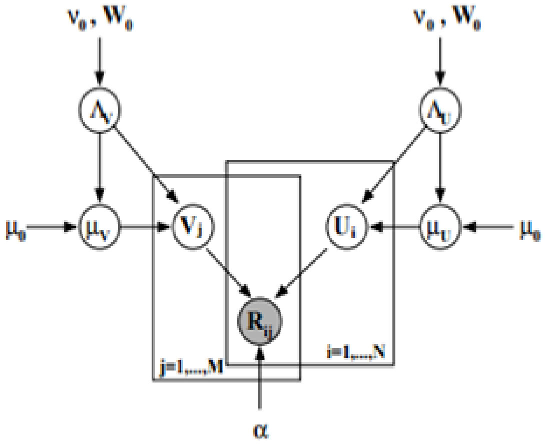
$\Theta_0 = \{\mu_0, \nu_0, W_0\}$

$$W(\Lambda\,|W_0,v_0) = 1/C|\Lambda|^{(v_0-D-1)/2}exp(-1/2\,Tr(W_0^{-1}\ \Lambda))$$

Where W is the Wishart distribution with ν_0 degrees of freedom, W_0 is a D × D scale matrix, C is the normalizing constant.



(a)



(b)

Figure 3:a) Graphical model of PMF b) Graphical model of Bayes PMF

In Cross Domain Recommendation clustering of users or items in one domain relatively works to generate recommendations in target domain [19, 20]. Collaborative Filtering Techniques proved to be efficient in cross domain recommenders [21].

CDR-PMF

In probabilistic matrix factorization matrices U and V are initialized with standard normal distribution along with zero mean and standard deviation as one [22]. Error gives the difference between the actual rating and observed rating. The main goal of PMF is to minimize this error, for which the error batch gradient descent algorithm is used. Batch gradient descent is an optimization algorithm which works by efficiently searching the parameter space, intercept (θ_0) and slope (θ_1) for linear regression, according to the rule, $\Theta = \Theta - \alpha\delta/\delta\Theta\big(J(\Theta)\big)$.

Where α is the learning rate, a free parameter, J(θ) is the least square cost function defined as

$$J(\Theta) = 1/2\sum_{i=1}^{m}\big(h_\Theta(x^i) - y^i\big)^2$$

$$h_\Theta(x^i) = \Theta_0 + \Theta_1 x^i$$

Where m is the total number of training examples, $h_\theta(x^{(i)})$ is the hypothesis function, the super script (i) is used to denote the ith sample.

Finally after minimizing the error by using batch gradient descent algorithm, U and V are updated in each iteration. After convergence of the algorithm, rating is calculated with updated U and V using maximum a posteriori probability (MAP) estimate which gives an estimate of an unknown quantity which equal the mode of posterior distribution. Now with all available ratings the items are recommended to users based on top-k ratings. The steps involved in cross domain matrix factorization are given in algorithm 1.

Algorithm 1: Cross domain Recommender with Matrix Factorization

```
Input : Source and Target data set in the form of rating matrix
Output : Generate recommendations in target domain using the knowledge of Source domain

Step 1: Pass source data set to PMF algorithm and retrieve the
        final updated $U_{source}$ and $V_{source}$
Step 2: Pass movie2 data set to PMF algorithm and retrieve the
        final updated $U_{target}$ and $V_{target}$
Step 3: Apply clustering on source domain for users and items
            Kmeans ($U_{source}$, k)
            Kmeans ($V_{source}$, k)
Step 4: For all users $u_i \in U_{target}$ in target domain do
    — Calculate the distance between the users in target domain with source domain user centroids and assign appropriate cluster id's for each user.
    — For each $u_i$ find a list of similar users in source domain and assign cluster id's for those liked by the users in source domain.
    — For all items $v_j \in V_{target}$ in target domain do
    — Find the distance between the items in target domain with source domain item centroid and assign appropriate cluster id's for each item.
Step 5: Now match these cluster id's of source items with
        target items and finally obtain list of items liked by the
        user in target domain.
    — Calculate the ratings of the corresponding user with list of items and recommend Top-k items.
```

Here CDR is applied with PMF and Bayes PMF to both source and target data in order to find user latent feature vector U and item latent feature vector V. Source domain user latent feature U and item latent feature V are clustered using k-means algorithm. Now cluster id's are assigned for each user and item in the source domain. Then from the target domain, a user is selected and found the distance to all clusters and assigns the nearest cluster-id to that user. This knowledge is used to recommended items to the user in target domain.

IV. RESULTS AND DISCUSSIONS

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

In order to experiment the proposed method the two benchmark data sets about movie ratings were used (Movielens as source domain and Netflix as target domain). These are publicly available datasets which contains information of user ratings on different movies on a scale of 1-5, where 1 means poor and 5 means excellent. The format of each record contains three fields: UserId, MovieId, Rating. In order to experiment the scalability of the model according to number of items three test sets were defined as shown in table 1.

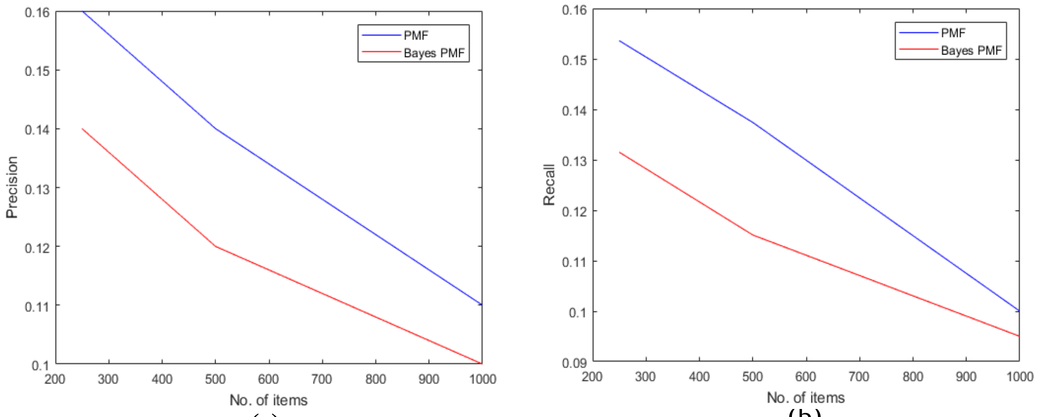
Table 1: Test sets of the source and target domains

Test Set	No of Users	No of Items	No. of Ratings	
			Movielens	Netflix
TS1	600	250	8076	7101
TS2	600	500	14864	13592
TS3	600	1000	23224	22103

Precision and Recall are the measures used to evaluate the performance of the CF models. Precision is used to measure the correctness of recommendation and is defined as the fraction of relevant instances among the retrieved instances. Recall is used to measure the accuracy of recommendation and is defined as the fraction of relevant instances that have been retrieved over total relevant instances. Here table 2 shows the precision and recall values of PMF and Bayes-PMF methods on the three test sets considered. Figure 4 shows the corresponding graphs for the table 2.

Table 2: PMF &Bayes PMF Precision & Recall values

Test set	**PMF**		**Bayes PMF**	
	Precision	**Recall**	**Precision**	**Recall**
TS1	16.0360	15.36	14.120	13.15
TS2	14.00	13.74	12.37	11.51
TS3	11.622	10.00	10.49	9.00



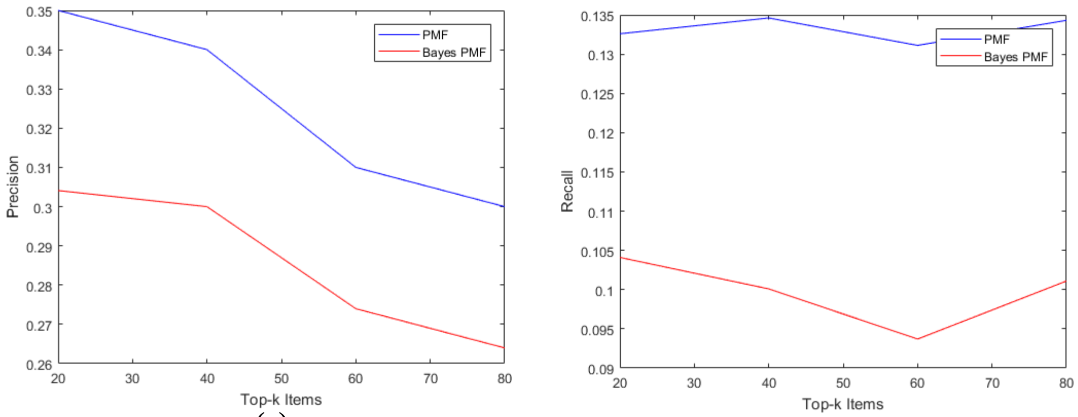
(a) (b)

Figure 4: a)Precision and b)Recall of Bayes PMF & PMF methods on Test set1, Test set2, Test set3

We ran our cross domain experiment on different k values and got best results with k=10 clusters and finally take precision and recall values for Top-k=20, 40, 60 and 80 items. Table 3 shows the precision and recall of the PMF and bayes PMF for top-k recommendations and corresponding graphs are shown in figure 5 and figure 6.

Table 3: Precision and Recall values for top-k items on Test set2 and Test set3

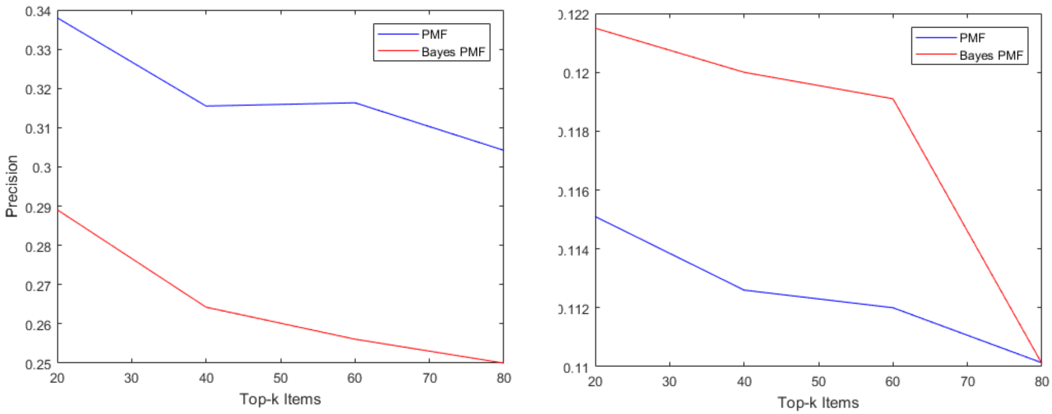
Top-k	**TS2**				**TS3**			
	PMF		**Bayes PMF**		**PMF**		**Bayes PMF**	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
20	35.900	13.260	30.415	10.41	33.800	11.560	28.910	12.1580
40	34.5750	13.461	30.070	10.01	31.550	11.216	26.4250	12.001
60	31.633	13.112	27.409	9.37	31.633	11.200	25.61	11.91
80	30.7875	13.431	26.412	10.11	30.421	11.013	25.00	11.013



(a)

(b)

Figure 5: a) Precision and b) Recall of Bayes PMF &PMF precision for top-k items for TS2



(a)

(b)

Figure 6: a) Precision and b) Recall of Bayes PMF &PMF precision for top-k items for TS3

It is observed that the precision and recall values of test case 2 and test case 3 for PMF method in cross domain CF are better than Bayes PMF method. However recall value for top-k recommendations, when k is huge and numbers of samples are high, converges for PMF and Bayes PMF methods.

V. CONCLUSION

This paper presented a cross domain recommender system which uses model based collaborative filtering system with probabilistic matrix factorization that avoids data sparsity and cold start problem. This method uses data from multiple domains and allows knowledge transfer from source domain to target domain. The cross domain recommender with PMF outperformed bayes PMF with respect to precision and recall. The key findings of the present work are that cross domain recommender provides a provision to resolve the data sparsity issue. As a future work, huge data sets and diverse models have to be experimented in cross domain.

VI. REFERENCES

[1] Raymond J Mooney and Loriene Roy. “Content-based book recommending using learning for text categorization” In Proceedings of the fifth ACM conference on Digital libraries, pages 195 - 204. ACM, 2000.

[2] Rong Pan and Martin Scholz. “Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering” In Proceedings 31 of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 667-676. ACM, 2009.

[3] Michael J Pazzani. “A framework for collaborative, content-based and demographic filtering”, Artificial Intelligence Review, 13(5-6):393-408, 1999.

[4] Wei Chen, Wynne Hsu, and Mong Li Lee, “Making recommendations from multiple domains” In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 892-900. ACM, 2013.

[5] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. “Talmud: transfer learning for multiple domains”, In Proceedings of the 21st ACM international conference on Information and knowledge management, pages 4250-434. ACM, 2012.

[6] Xiaoyuan Su, Taghi M Khoshgoftaar, Xingquan Zhu, and Russell Greiner, “Imputation-boosted collaborative filtering using machine learning classifiers” In Proceedings of the 2008 ACM symposium on Applied computing, pages 949-950. ACM, 2008.

[7] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. “Unifying user-based and item-based collaborative filltering approaches by similarity fusion” In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 501-508. ACM, 2006.

[8] Zhenzhen Xu, Huizhen Jiang, Xiangjie Kong, Jialiang Kang, Wei Wang, Feng Xia, “Cross-Domain Item Recommendation Based on User Similarity” , Journal of Computer Science and Information Systems, vol. 13(2), ,2016, pp.359–373.

[9] Paolo Cremonesi, Antonio Tripodi, Roberto Turrin, “Cross-domain Recommender Systems”, IEEE International Conference on Data Mining Workshops, 2014, pp.496-503.

[10] Bracha Shapira, Lior Rokarch, Shirley Freilikhman, “Facebook Single And Cross Domain Data For Recommendation Systems”, pringer J. for User Model User-Adaptation, vol. 23, 2013, pp. 211-247.

[11] Shaghayegh Sahebi, Trevor Walker, “Content-Based Cross-Domain Recommendations Using Segmented Models”, CEUR Workshop, 2014, pp. 57-63.

[12] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos, “Providing justications in recommender systems”, Humans, 38(6):1262-1272, 2008.

[13] Mingrui Wu, “Collaborative filtering via ensembles of matrix factorizations”, In Proceedings of KDD Cup and Workshop, 2007.

[14] Daniel D Lee and H Sebastian Seung, “Learning the parts of objects by non-negative matrix factorization”, Nature, 401(6755):788-791.

[15] Jasson DM Rennie and Nathan Srebro. “Fast maximum margin matrix factorization for collaborative prediction” In Proceedings of the 22nd international conference on Machine learning, pages 713-719. ACM, 2005.

[16] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola, “Maximum margin matrix factorization. In Advances in neural information processing systems” pages 1329-1336, 2004.

[17] Ruslan Salakhutdinov and Andriy Mnih. “Probabilistic matrix factorization”, In NIPS, 20, pages 1-8, 2011

[18] Ruslan Salakhutdinov, Andriy Mnih, “Bayesian Probabilistic Matrix Factorization using Markov Chain”, Monte Carlo. In NIPS Pages 1-8.

[19] Aleksandr Farseev, Ivan Samborskii, Andrey Filchenkov, and Tat-Seng Chua.,”Cross-Domain Recommendation via Clustering on Multi-Layer Graphs”, In SIGIR, 2017

[20] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang, “Social Recommendation with Cross-Domain Transferable Knowledge”, TKDE 27, vol. 11, 3084–3097, 2015

[21] B. Li, Q. Yang, and X. Xue, “Can Movies And Books Collaborate? – Crossdomain Collaborative Filtering For Sparsity Reduction”, International Joint Conference on Artificial Intelligence, 2009, pp. 2052- 2057.

[22] Yehuda Koren, Robert Bell, Chris Volinsky,”Matrix factorization techniques for recommender systems” Computer, 42(8):30-37, 2009.