A Cross Domain Collaborative Filtering Algorithm Based on Latent Factor Alignment and Two-Stage Matrix Adjustment

Xu Yu[1], Junyu Lin[2](✉), Feng Jiang[1], Yan Chu[3], and Jizhong Han[2]

[1] School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China
[2] Institute of Information Engineering, CAS, Beijing 100093, China
linjunyu@iie.ac.cn
[3] College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Abstract. Sparsity is a tough problem in a single domain Collaborative Filtering (CF) recommender system. In this paper, we propose a cross domain collaborative filtering algorithm based on Latent Factor Alignment and Two-Stage Matrix Adjustment (LFATSMA) to alleviate this difficulty. Unlike previous Cross Domain Collaborative Filtering (CDCF) algorithms, we first align the latent factors across different domains by pattern matching technology. Then we smooth the user and item latent vectors in the target domain by transferring the preferences of similar users and the contents of similar items from the auxiliary domain, which can effectively weaken the effect of noise. Finally, we convert the traditional UV decomposition model to a constrained UV decomposition model, which can effectively keep the balance between under-fitting and over-fitting. We conduct extensive experiments to show that the proposed LFATSMA algorithm performs better than many state-of-the-art CF methods.

Keywords: Cross Domain Collaborative Filtering · Knowledge transfer
Latent Factor Alignment · Constrained UV decomposition model

1 Introduction

In recent years, recommender systems are widely used in e-commerce sites and online social media and the majority of them offer recommendations for items belonging to a single domain. Now collaborative Filtering (CF) [1] algorithm is the most widely used method for recommender systems. However, in real-world recommender systems, the rating matrix is very sparse, which leads to a poor recommendation performance. To alleviate this difficulty, recently a number of Cross-Domain Collaborative Filtering (CDCF) methods have been proposed [2]. They can effectively relieve the sparsity problem in the target domain.

Currently CDCF methods can be categorized into two classes. One class [3–5] assumes shared users or items. The other class contains a limited number of CDCF methods [6, 7] that do not require shared users and items. However, methods in the

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 473–480, 2018.
https://doi.org/10.1007/978-3-319-73317-3_54

second class may not perform well, as they are based on matrix factorization. Matrix factorization techniques fail in the cross-domain recommendation task because the learned latent factors are not aligned over different domains.

In this paper, for the second class, we proposed a CDCF algorithm based on Latent Factor Alignment and Two-Stage Matrix Adjustment (LFATSMA). We first align the latent factors across different domains, so the knowledge transfer from the auxiliary domain to the target domain would be more correct and reasonable. Then we propose a two-stage matrix adjustment method to achieve more effective U and V matrices with the help of the data in the auxiliary domain. Consequently, the prediction performance in the target domain can be improved.

The remainder of this paper is organized as follows: Sect. 2 proposes a method to align the latent factors across different domains. In Sect. 3, we propose the two-stage matrix adjustment method to transfer knowledge from the auxiliary domain to the target domain. We conduct extensive experiments to test the performance of the proposed algorithm in Sect. 4 and conclude the whole paper in Sect. 5.

2 Aligning the Latent Factors

We align the latent factors across different domains by pattern matching technology. As shown in Fig. 1, we **first** construct a mixture rating matrix M_0 by combining the data from the two domains together. The main-diagonal blocks are filled with the rating matrix M_1 in the target domain and the rating matrix M_2 in the auxiliary domain. The off-diagonal blocks are filled with zeros.

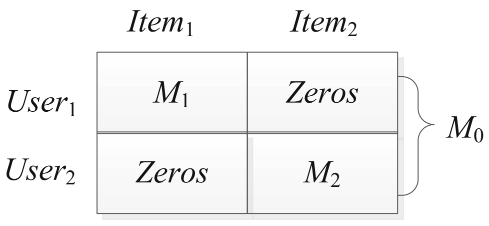

Fig. 1. Constructing a mixture rating matrix

Let n_1 and n_2 denote the size of $User_1$ and $User_2$ respectively. **Then** we decompose M_0, M_1, and M_2 to obtain the latent factors by the UV decomposition model [8]. Let $M_0 = U_0V_0^T$, $M_1 = U_1V_1^T$, $M_2 = U_2V_2^T$, and let f denote the dimensionality of the latent factor space, so the size of U_0 is $(n_1 + n_2, f)$, the size is of U_1 is (n_1, f), and the size of U_2 is (n_2, f). The reason why we construct the mixture matrix M_0 and decompose M_0 is to use it as a reference. Considering that the order of the latent factors in U_0 is unique, we can align the latent factors between the target domain and the auxiliary domain by this order.

Finally, we align the latent factors across the two domains, and return the updated U_1 and U_2.

In Fig. 2, each column vector in U_i (i = 0, 1, 2) represents a latent factor. Obviously, for the same latent factor (e.g., SF) in two U matrices, if the users are identical in order, then we can expect the corresponding columns to be with a large similarity. Therefore we can determine whether two latent factors F_i and F_j from two different U matrices are identical according to the similarity of the corresponding columns c_i and c_j.

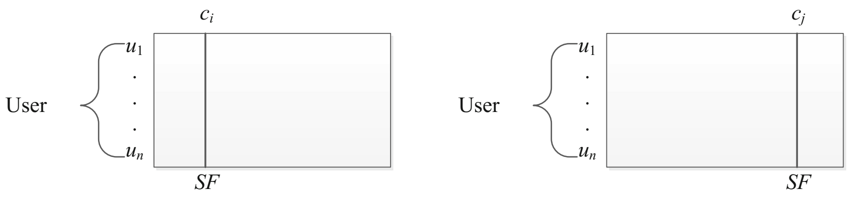

Fig. 2. The same latent factor in two U matrices can be expected to be with a large similarity

Let U_3 denote the upper block of U_0, including the upper n_1 rows, and U_4 denote the lower block of U_0, including the lower n_2 rows. It is clear that each column in both U_3 and U_1 represents the interest values on a latent factor of all the users in the target domain, so we can use U_3 as a reference to align the latent factors of U_1 according to the similarities among columns. In the same way, we can also use U_4 as a reference to align the latent factors of U_2. As U_3 and U_4 correspond to the upper and lower blocks of the same matrix U_0 respectively, so the order of the latent factors in U_3 is the same with that in U_4. As a result, we can align the latent factors of U_1 and U_2.

We align the latent factors by comparing the similarities among columns. Here the similarity can be computed by a cosine measure in the form

$$s(c_i, c_j) = \frac{c_i^T c_j}{\|c_i\| \|c_j\|} \tag{1}$$

For each column c_i in U_1, we compute the similarity between it and the first column C_1 in U_3 by Eq. (1), and denote the column in U_1 with the maximum similarity to C_1 as F_1, and exchange this column with the first column in U_1. Then determine F_2 from the rest columns, and exchange the corresponding column with the second column in U_1. The rest can be done in the same manner. The columns of U_2 can be adjusted in the same way.

In the UV decomposition model, the order of the latent factors in the V matrix is the same with that in the U matrix. Since the orders of latent factors in U_1 and U_2 matrices have been adjusted, if we adjust the orders of latent factors in V_1 and V_2 matrices by the same adjustment process, we can also align the latent factors between V_1 and V_2. As a result, the latent factors between the target domain and the auxiliary domain are aligned, which makes the following knowledge transfer more correct and reasonable.

3 Transferring Knowledge via a Two-Stage Matrix Adjustment

3.1 Weakening the Effect of Noise

For any user u in the target domain, we first choose from the auxiliary domain the l most similar users to user u. Then we compute the mean of the latent vectors over the l most similar users and replace the latent vector p_u with the mean. As the auxiliary domain contains more user rating data, the latent vector $p_{u'}$ in the auxiliary domain is relatively accurate, so the mean of the latent vectors over the l most similar users is a good replacement of p_u. We replace p_u with the mean of the l most similar users from the auxiliary domain, which is a smooth method and can effectively weaken the effect of noise. The detailed process is given by the following.

(1) Choose from the auxiliary domain the l most similar users to user u
Let u_a denote a user in the target domain and u_b denote a user in the auxiliary domain. We choose from the auxiliary domain the l most similar users to u_a. Here the similarity can also be computed by a cosine measure that was given in Eq. (1), and accordingly the similarity between u_a and u_b can be computed in the form

$$s(p_{u_a}, p_{u_b}) = \frac{p_{u_a}^T p_{u_b}}{\|p_{u_a}\| \|p_{u_b}\|} \tag{2}$$

(2) Compute the mean of the latent vectors over the l most similar users
We compute the mean of the latent vectors over the l most similar users. Let p denote the mean, and p_{u_i} denote the latent vector of the *i-th* most similar user, $(i = 1, \cdots, l)$. The mean of the latent vectors is defined as

$$p = \sum_{i=1}^{l} p_{u_i} \Big/ l \tag{3}$$

(3) Replace the latent vector p_u of user u in the target domain with the corresponding p
Thus we can update the U matrix. In the same manner, we can also update the V matrix.

3.2 Solving a Constrained UV Decomposition Model

Although we transfer important information from the auxiliary domain to smooth the original data in the target domain, there arises a new problem that the updated U and V matrices may not fit the rating data of the target domain accurately. For convenience, we use $U^{(1)}$ and $V^{(1)}$ to denote the first updated matrices. In order to avoid this problem, an intuitive idea is to use $U^{(1)}$ and $V^{(1)}$ as an initial point, and to solve the traditional UV model for a better U and V matrices. However, this may cause a large change of $U^{(1)}$ or $V^{(1)}$. As $U^{(1)}$ and $V^{(1)}$ are obtained by transfer important information from the auxiliary domain, and can effectively weaken the effect of noise, we expect that they are changed as small as possible. To achieve this goal, we convert the traditional unconstrained UV decomposition model into a constrained UV decomposition model in the following form

$$\begin{gathered}\min_{q*,p*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 \\ s.t.\ \left\| q_i - q_i^{(1)} \right\|^2 \ and\ \left\| p_u - p_u^{(1)} \right\|^2 \ is\ as\ small\ as\ possible\ for\ any\ i\ and\ u\ belonging\ to\ \kappa\end{gathered} \tag{4}$$

We can convert (4) into the following unconstrained optimization problem

$$\min_{q*,p*} \quad F = \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\left\| q_i - q_i^{(1)} \right\|^2 + \left\| p_u - p_u^{(1)} \right\|^2) \tag{5}$$

where $q_i^{(1)}$ and $p_u^{(1)}$ are the item and user latent vectors respectively corresponding to the $V^{(1)}$ and $U^{(1)}$ matrices, and the constant λ is a penalty factor, which penalizes the change between q_i and $q_i^{(1)}$ and the change between p_u and $p_u^{(1)}$. Clearly, the UV decomposition model may arise over-fitting if λ is set to a very small number. On the contrary, it will cause under-fitting if λ is set to a very large number. A proper λ is usually determined by cross-validation. We use $U^{(2)}$ and $V^{(2)}$ to denote the solution of the optimization problem (5). We can also use stochastic gradient descent to achieve $U^{(2)}$ and $V^{(2)}$.

For each given training case, firstly the gradient can be computed in the following form

$$\begin{aligned}\frac{\partial F}{q_i} &= -2\left[e_{ui}p_u - \lambda(q_i - q_i^{(1)})\right] \\ \frac{\partial F}{p_u} &= -2\left[e_{ui}q_i - \lambda(p_u - p_u^{(1)})\right]\end{aligned} \tag{6}$$

where $e_{ui} \overset{def}{=} r_{ui} - q_i^T p_u$. Then we modify the parameters by a magnitude proportional to γ (i.e., the learning rate) in the opposite direction of the gradient, yielding:

$$\begin{aligned}q_i &\leftarrow q_i + \gamma\left[e_{ui}p_u - \lambda(q_i - q_i^{(1)})\right] \\ p_u &\leftarrow p_u + \gamma\left[e_{ui}q_i - \lambda(p_u - p_u^{(1)})\right]\end{aligned} \tag{7}$$

Since the U and V matrices updated in the first adjustment absorb useful information from the auxiliary domain, we use $U^{(1)}$ and $V^{(1)}$ as an initial point in the optimization problem (5). Finally, we can obtain the rating matrix M by computing $M = U^{(2)}V^{(2)T}$.

4 Experiments

In this section, we compare our algorithm to 3 state-of-the-art algorithms. One is a well-known single domain algorithm Funk-SVD (the UV decomposition model), and the other two methods are cross domain methods, namely CBT and RMGM. By comparison with Funk-SVD, we can investigate the effectiveness of transferring knowledge from the auxiliary domain. By comparison with CBT and RMGM, we can investigate the effectiveness of aligning the latent factors across different domains.

4.1 Data Sets

In this part, we use EachMovie and MovieLens data sets.

(1) EachMovie (the auxiliary domain): 500 users and 500 movies are extracted from EachMovie to compose the auxiliary domain.
(2) MovieLens (the target domain): 500 users and 1000 movies are extracted from MovieLens to compose the target domain.

4.2 The Setting of the Compared Methods

(1) Funk-SVD (the UV decomposition model): Here we simply set $f = 50$.
(2) CBT (Codebook transfer): According to the setting in reference [6], the numbers of user and item clusters, K and L, are set to 50.
(3) RMGM (Rating Matrix Generative Model): In order to compare the methods more reasonable and fairer, like CBT, both K and L in RMGM are also set to 50.
(4) **LFATSMA** (the proposed method): In order to compare the methods more reasonable and fairer, like Funk-SVD, the dimension of the latent space is set to 50, and the number l of similar users or items is set to10.

In the experiments, we set $\gamma = 0.3$ in each algorithm.

4.3 Evaluation Protocol

We use the first 100, 200, and 300 users in the target data set as training data, respectively, and we use the last 200 users as testing data. For each test user, Given5 denotes 5 observed ratings are used for training. Given10 and Given15 are defined in the same way.

We use mean absolute error (MAE) and root mean square error (RMSE) as evaluation metrics in our experiments. MAE is defined as

$$\left(\sum\nolimits_{i\in T} |r_i - \tilde{r}_i|\right)/|T| \tag{8}$$

and RMSE is defined as

$$\sqrt{\sum\nolimits_{i \in T} (r_i - \tilde{r}_i)^2 / |T|} \quad (9)$$

where T denotes the set of test ratings, r_i is the ground truth and $\tilde{r}_i$ is the predicted rating.

4.4 Results

Table 1 lists the MAE and RMSE scores on MovieLens (ML).

Table 1. MAE and RMSE scores

Training set	Method	MAE			RMSE		
		Given5	Given10	Given15	Given5	Given10	Given15
ML100	Funk-SVD	1.249	1.241	1.234	1.500	1.491	1.481
	CBT	0.692	0.677	0.655	0.893	0.881	0.866
	RMGM	0.694	0.668	0.653	0.895	0.879	0.864
	LFATSMA	**0.633**	**0.605**	**0.561**	**0.860**	**0.826**	**0.775**
ML200	Funk-SVD	1.033	1.093	1.057	1.261	1.329	1.286
	CBT	0.675	0.661	0.644	0.889	0.873	0.861
	RMGM	0.666	0.657	0.632	0.880	0.867	0.849
	LFATSMA	**0.624**	**0.592**	**0.556**	**0.856**	**0.822**	**0.770**
ML300	Funk-SVD	0.918	0.899	0.897	1.178	1.165	1.162
	CBT	0.664	0.659	0.639	0.875	0.869	0.852
	RMGM	0.661	0.663	0.644	0.873	0.876	0.858
	LFATSMA	**0.619**	**0.589**	**0.551**	**0.846**	**0.816**	**0.765**

As shown in Table 1, CBT, RMGM, and LFATSMA all perform better than Funk-SVD on all different configurations. The main reason is that Funk-SVD predicts the ratings only according to the sparse data in the target domain.

As expected, our method performs better than the other two CDCF methods (CBT and RMGM), I think the main reasons are as follows. Firstly, the alignment of latent factors between the target domain and the auxiliary domain makes the following knowledge transfer more correct and reasonable. Secondly, the smooth of the user and item latent vectors in the target domain can effectively weaken the effect of noise. Thirdly, the solution of the constructed constrained UV decomposition model can effectively keep the balance between under-fitting and over-fitting. Consequently, the prediction performance in the target domain can be improved.

5 Conclusion

In this paper, we propose a CDCF algorithm based on Latent Factor Alignment and Two-Stage Matrix Adjustment (LFATSMA). By aligning the latent factors across different domains, and transferring the preferences of similar users and the contents of similar items from the auxiliary domain to the target domain, LFATSMA can effectively alleviate the sparsity problem in the target domain and weaken the effect of noise. Moreover, since we construct a constrained UV decomposition model to control the balance between under-fitting and over-fitting, the effectiveness of the knowledge transfer can be guaranteed. The experimental results have validated the effectiveness of the proposed LFATSMA algorithm.

Acknowledgments. This work is sponsored by the National Natural Science Foundation of China (No. 61402246), a Project of Shandong Province Higher Educational Science and Technology Program (No. J15LN38), Qingdao indigenous innovation program (No. 15-9-1-47-jch).

References

1. Goldberg, D., Nichols, D., Oki, B.M., et al.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12), 61–70 (1992)
2. Berkovsky, S., Kuflik, T., Ricci, F.: Cross-domain mediation in collaborative filtering. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 355–359. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73078-1_44
3. Pan, W., Xiang, E.W., Liu, N.N., et al.: Transfer learning in collaborative filtering for sparsity reduction. In: Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 2010
4. Pan, W., Liu, N.N., Xiang, E.W., et al.: Transfer learning to predict missing ratings via heterogeneous user feedbacks. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2011, Barcelona, Catalonia, Spain, July 2011
5. Loni, B., Shi, Y., Larson, M., Hanjalic, A.: Cross-domain collaborative filtering with factorization machines. In: de Rijke, M., Kenter, T., de Vries, A.P., Zhai, C., de Jong, F., Radinsky, K., Hofmann, K. (eds.) ECIR 2014. LNCS, vol. 8416, pp. 656–661. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06028-6_72
6. Li, B., Yang, Q., Xue, X.: Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2009, Pasadena, California, USA, pp. 2052–2057. July 2009
7. Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, pp. 617–624. June 2009
8. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)