Low-Rank and Sparse Cross-Domain Recommendation Algorithm

Zhi-Lin Zhao[1], Ling Huang[1], Chang-Dong Wang[1(✉)], and Dong Huang[2]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
zhaozhl7@mail2.sysu.edu.cn, huanglinghl@hotmail.com,
changdongwang@hotmail.com

[2] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China
huangdonghere@gmail.com

Abstract. In this paper, we propose a novel Cross-Domain Collaborative Filtering (CDCF) algorithm termed Low-rank and Sparse Cross-Domain (LSCD) recommendation algorithm. Different from most of the CDCF algorithms which tri-factorize the rating matrix of each domain into three low dimensional matrices, LSCD extracts a user and an item latent feature matrix for each domain respectively. Besides, in order to improve the performance of recommendations among correlated domains by transferring knowledge and uncorrelated domains by differentiating features in different domains, the features of users are separated into shared and domain-specific parts adaptively. Specifically, a low-rank matrix is used to capture the shared feature subspace of users and a sparse matrix is used to characterize the discriminative features in each specific domain. Extensive experiments on two real-world datasets have been conducted to confirm that the proposed algorithm transfers knowledge in a better way to improve the quality of recommendation and outperforms the state-of-the-art recommendation algorithms.

Keywords: Low-rank · Sparse · Cross-domain
Recommendation algorithm

1 Introduction

Nowadays, for the sake of the benefit of businesses and the satisfaction of users, many online platforms like Amazon, Netflix and Douban use recommendation algorithms [1–3] to recommend items to users who are the most likely to be interested in them by analyzing huge amounts of data about user behaviour. More often than not, the task of recommendation algorithm is to speculate the value of missing ratings in the sparse user-item rating matrix by analyzing a few known ratings. Then some unrated items with high predicted ratings will be recommended to the target users. Collaborative Filtering (CF) [4,5] is the most widely used recommendation algorithm and one representative technology for collaborative filtering is matrix factorization.

J. Pei et al. (Eds.): DASFAA 2018, LNCS 10827, pp. 150–157, 2018.
https://doi.org/10.1007/978-3-319-91452-7_10

Over the past decade, matrix factorization has attracted an increasing amount of attention and has been applied in many areas such as machine learning [6] and data mining [7]. From the perspective of collaborative filtering [4], matrix factorization firstly extracts latent factors of users and items from the user-item rating matrix and then predicts missing ratings according to those latent factors. In practice, the rating matrix is usually quite sparse so that it is difficult to learn satisfactory latent factors of users and items which will make a great impact on the quality of rating prediction. Besides, it is a challenging task to make reliable predictions for new users due to the lack of the relevant rating data. In order to solve the above data sparsity and cold-start problems as well as improve the quality of recommendation [8], transfer learning [9] has been integrated into matrix factorization to transfer the knowledge from auxiliary data to rating data. Some typical auxiliary data are social media [10], tag information [11], user reviews [12] and product images [13].

As a special case of transfer learning, Cross-Domain Collaborative Filtering (CDCF) [14] transfers knowledge of rating data among multiple domains to make a better recommendation. It predicts ratings for all domains by learning from multiple rating matrices of different domains together. Combining all domains, we have more data to describe the features of users which can mitigate the sparsity problem. If a user does not get any ratings to items in a target domain but there are some ratings in other relative domains, we can solve the cold-start problem by learning the features of the user from other relative domains. Most existing CDCF methods [15–18] tri-factorize the rating matrix of each domain into three low dimensional matrices which represent user latent factors (or labels), codebook (or rating pattern matrix) and item latent factors (or labels) respectively. Usually, the codebook describing the relations between the clusters of users and items is shared among domains. Besides, users may appear in all domains and the user factors can also be shared, so user latent factor matrix or codebook can be viewed as a bridge to transfer knowledge. However, in the scenario where the domains are uncorrelated, the bridge can not transfer meaningful knowledge and even transfer negative knowledge which reduces the effect of recommendation. Besides, even for the same user, the performance will differ from one domain to another. So there are some domain-specific features of each user. Unfortunately, there is still a lack of methods differentiating shared features and domain-specific features adaptively.

To solve the above problems, we propose a novel cross-domain collaborative filtering algorithm termed Low-rank and Sparse Cross-Domain (LSCD) recommendation algorithm. Different from most of the existing CDCF algorithms, we decompose all the rating matrices of different domains into user and item latent feature matrices. More specifically, each user latent feature matrix consists of two parts: user-domain-shared feature matrix and user-domain-specific feature matrix. The former is used to describe the overall preferences of users among multiple domains which is modeled by a low-rank matrix. And the latter is a sparse matrix which is used to characterize the discriminative features in each domain because the expressions of some features vary from one domain to another. For example, if a user is willing to give high ratings to all items on average, we can capture this feature by analyzing the rating records of all domains. But if the user prefers movies to books, the user may give higher ratings on movie domain than book domain. So, a domain-specific feature matrix should be used to adjust shared feature matrix to fit specific domain. Therefore, if the domains are correlated, the performance can be improved because we use more rating data to learn the shared features of users. On the contrary, if the domains are uncorrelated, the performance can still be improved because we can distinguish the domain-specific features of different domains.

Extensive experiments have been conducted on two real world datasets: Amazon and MovieLens. The results show that the proposed LSCD recommendation algorithm can improve the quality of recommendation as the number of domains increases even the domains are uncorrelated and outperforms the state-of-the-art recommendation algorithms.

2 The Proposed Algorithm

We assume there are D domains in a recommendation task. The input user-item rating matrix $R_d \in \mathbb{R}^{m \times n_d}$ represents the rating relation between m users and n_d items in the d^{th} domain. Note that the users are the same in each domain. Each entry denotes the rating of a user to an item within a certain numerical interval $[R_{min}, R_{max}]$ which will vary on different datasets. The rating will be zero if the user has not rated the item. I_d is the indicator matrix of R_d, where the value will be equal to one if the corresponding item in R_d has been rated or zero otherwise. $U \in \mathbb{R}^{m \times l}$ is the user-domain-shared feature matrix and $H_d \in \mathbb{R}^{m \times l}$ is the user-domain-specific feature matrix in the d^{th} domain. $V \in \mathbb{R}^{n_d \times l}$ denotes the item latent feature matrix in the d^{th} domain. Among them, $l \ll \min(m, n_d), \forall d = 1, \ldots, D$ is the number of latent features.

In the d^{th} domain, the user latent feature matrix is the sum of the domain-shared feature matrix and domain-specific feature matrix, i.e., $U + H_d$. In the proposed algorithm, the shared features and domain-specific features can be differentiated adaptively. Based on the traditional matrix factorization algorithm, the predicted ratings of users to unrated items in the domain can be estimated by the product of users' and items' latent feature matrices,

$$P_d = (U + H_d)V_d^T. \tag{1}$$

In matrix factorization, those latent feature matrices will be learnt from rating data by minimizing the sum of squared errors between real ratings and predicted ratings. Besides, we should learn all the parameters among all domains together so that the shared features and domain specific features of users can be distinguished, and the user-domain-shared feature matrix U can be viewed as a bridge to transfer knowledge among domains. So the loss function is,

$$\frac{1}{2}\sum_{d=1}^{D} \|\big(R_d - (U + H_d)V_d^T\big) \odot I_d\|_F^2,$$

where $\|\cdot\|_F$ is Frobenius norm and $\odot$ denotes the Hadamard product.

Multi-Task Learning (MTL) [19] captures the task relationship via a shared low-rank structure, and CDCF is similar to this learning method since we want to explore the latent feature relationship among multiple domains. So the basic idea of MTL inspires us to divide the factors of users into shared part and domain-specific part. The shared part is used to model the overall preferences of users among multiple domains. Although the dimension of U is $\mathbb{R}^{m\times l}$, the number of shared features may be less than l and even equal to 0 if the domains are uncorrelated. When the domains are uncorrelated, it is not necessary to transfer knowledge because transferring negative knowledge may reduce the effect. Besides, if the domains are strongly correlated, the rank of U is at most l. So we need to find some important shared features from the l features, the number of which may be less than l. Besides, we can get a good result even the number of features is not large enough, since the most important features can be selected. Therefore, we assume U is low-rank. So users will share the same low-rank feature subspace and we should minimize rank(U) to get a low-rank structure of user-domain-shared feature matrix.

On the other hand, the preference of a user to different domains will vary slightly as discussed earlier and it can be reflected by a few features. So we use entry-wise sparse regularization term to identify those discriminative features in each domain. Because l_0-norm counts the number of nonzero elements of a solution, we should minimize $\|H_d\|_0$ to get a sparse structure of user-domain-specific feature matrix. And the matrix can be utilized to adjust domain-shared feature matrix to fit specific domain. The objective function is,

$$\begin{aligned}\mathcal{L}_{\mathcal{O}} = &\frac{1}{2}\sum_{d=1}^{D}\|\left(R_d-(U+H_d)V_d^T\right)\odot I_d\|_F^2+\frac{\lambda_V}{2}\sum_{d=1}^{D}\|V_d\|_F^2\\ &+\lambda_U \mathrm{rank}(U)+\lambda_H\sum_{d=1}^{D}\|H_d\|_0,\end{aligned}$$

where λ_V, λ_U and λ_H are regularization coefficients. The ratio between λ_H and λ_U is used to control the composition's ratio of shared and domain-specific parts in user latent features. The Frobenius norm of V_d is added to prevent overfitting. However, for U and H_d, the Frobenius norms are not necessary because the low-rank and sparse regularization terms of the two variables can also do this.

Solving the above nonconvex optimization problem is NP-hard. As pointed out in [20], if the rank of U is not too large and H_d is sparse, the regularization terms $rank(U)$ and $\|H_d\|_0$ can be approximated by the tightest convex relaxation $\|U\|_*$ and $\|H_d\|_1$ respectively where $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the Nuclear norm and the l_1-norm of a matrix, respectively. Therefore, the relaxed objective function is,

$$\mathcal{L} = \frac{1}{2}\sum_{d=1}^{D}\|\left(R_d-(U+H_d)V_d^T\right)\odot I_d\|_F^2+\frac{\lambda_V}{2}\sum_{d=1}^{D}\|V_d\|_F^2+\lambda_U\|U\|_*+\lambda_H\sum_{d=1}^{D}\|H_d\|_1.$$

The approximate projected gradient method is used to solve the above objective function. After obtaining the latent feature matrices U, H_d and V_d, the predicted rating matrix P_d of each domain can be estimated by Eq. (1).

3 Experiments

In this section, some experiments are conducted to evaluate the effectiveness of the proposed method[1].

3.1 Datasets and Evaluation Measures

In our experiments, two real-world datasets with multiple item domains are used, namely Amazon and MovieLens.

– **Amazon**[2]: This dataset is obtained from Julian McAuley [13], which contains 6,643,669 users, 2,441,053 products and 80,737,555 ratings of 24 domains from Amazon spanning Jun 1995–Mar 2013. Each record is a (user, item, rating, timestamp) tuple and the time information is not used. Four item domains are used in our experiments, i.e., Book, CD, Music and Movie.
– **MovieLens**[3]: This dataset is obtained from the Information Retrieval Group at Universidad Autónoma de Madrid, which contains 2113 users, 10,197 movies, and 855,598 ratings from MovieLens spanning 1970–2009. We use the tags of movies to classify the ratings into 18 domains and the four movie domains are used in our experiments, i.e., comedy (COM), dramatic (DRA), action (ACT), thrilling (THR) domains.

Without loss of generality, the two datasets are split randomly with 80% as the training set and 20% as the testing set. In order to evaluate the quality of the recommendation algorithms, two widely used evaluation metrics, namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), will be used to measure the accuracy of the predicted ratings.

3.2 Comparison Experiments

We compare the results of the predicted ratings of the proposed LSCD algorithm with nine state-of-the-art recommendation algorithms, i.e., **N-CDCF-U**, **N-CDCF-I**, **MF-CDCF**, **CMF** [21], **CDTF**, **CLMF** [17], **TALMUD** [15], **CDLD** [22] and **PCLF** [23] where N-CDCF-U, N-CDCF-I, MF-CDCF and CDTF are from the same paper [16]. N-CDCF-U and N-CDCF-I are neighborhood based collaborative filtering methods computing the similarities between users and between items by cosine similarity over all items and users respectively. But the other eight algorithms (including LSCD) are matrix factorization methods. For these matrix factorization methods, we set the dimensionality of latent feature vector $l = 50$. Besides, the step size μ and the item regularization coefficient λ_V are set to 0.001 and 0.1 respectively. To be fair, all the regularization coefficients of the compared algorithms are set to 0.1. We initialize all the latent feature matrices randomly.

[1] Source code and datasets are available at https://github.com/sysulawliet/LSCD.
[2] http://jmcauley.ucsd.edu/data/amazon.
[3] https://grouplens.org/datasets/hetrec-2011.

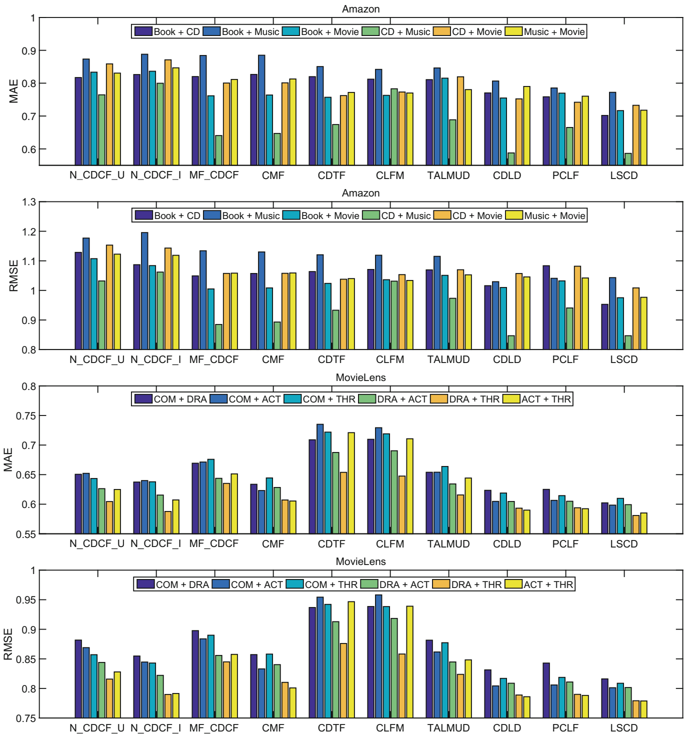

Fig. 1. Performance comparisons of different combination of two domains.

There are 11 combinations of the 4 domains. Plotting all the results in a figure will lead to chaos. Without loss of generality, we plot some typical combinations of domains. The comparison results in terms of MAE and RMSE on the two datasets over two domains are reported in Fig. 1. Six different pairs of domains are selected from each dataset respectively. The results show that the proposed LSCD algorithm outperforms the other state-of-the-art CDCF algorithms in all combinations of two domains. Generally speaking, the other five matrix factorization methods are inferior to the two neighborhood based collaborative filtering methods on the MovieLens dataset but better than the two algorithms on the Amazon dataset. The reason is that, the Amazon dataset is much sparser than the MovieLens dataset, and those matrix factorization methods can work well when data information is sparse because they predict ratings according to latent features rather than the original data. Besides, those neighborhood based collaborative filtering methods have more advantages when the rating matrix is dense since they can get a more precise user or item similarity matrix. But the proposed LSCD algorithm which is also based on matrix factorization can work well on both datasets. One reason may be that the low-rank structure of U makes it able to catch the shared feature subspace of users from few entries.

On the other hand, the results of different combinations of two domains are quite different. For example, the performance of (CD + Music) domains is better than (Book + Music) domains on the Amazon dataset. And the performance of (DRA + ACT) domains is better than (COM + ACT) domains on the MovieLens dataset. Intuitively, Music is more correlated to CD than Book and action movie is more correlated to dramatic movie than comedy movie. So the performance of correlated domains will be better for the eight recommendation algorithms. But the proposed LSCD algorithm can obtain good results even the two domains are uncorrelated since the all sparse matrices H_d can characterize those domain-specific features of users and differentiate all domains better.

4 Conclusion

In this paper, we have proposed a novel cross-domain collaborative filtering recommendation algorithm termed LSCD which can better model the latent features of users to improve the quality of rating prediction. Based on matrix factorization, we assume the user latent features are divided into shared and domain-specific parts. We use low-rank matrix to capture the shared feature subspace of users and use sparse matrix to identify discriminative features in each domain. The objective function is optimized by the approximate projected gradient method and theoretical analysis has shown the complexity and convergence of the proposed algorithm. Extensive experiments on two real-world datasets have confirmed that the proposed algorithm can transfer knowledge among domains in an even better fashion and significantly outperforms state-of-the-art recommendation algorithms.

Acknowledgments. This work was supported by NSFC (61502543 & 61602189), Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014), the Ph.D. Start-up Fund of Natural Science Foundation of Guangdong Province, China (2016A030310457), and Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (2016TQ03X542).

References

1. Zheng, B., Su, H., Zheng, K., Zhou, X.: Landmark-based route recommendation with crowd intelligence. Data Sci. Eng. **1**(2), 86–100 (2016)
2. Zhao, Z.L., Wang, C.D., Wan, Y.Y., Lai, J.H.: Recommendation in feature space sphere. Electron. Commer. Res. Appl. **26**, 109–118 (2017)
3. Hu, Q.Y., Zhao, Z.L., Wang, C.D., Lai, J.H.: An item orientated recommendation algorithm from the multi-view perspective. Neurocomputing **269**, 261–272 (2017)
4. Jannach, D., Zanker, M., Felfering, A., Friedrich, G.: Recommender Systems: An Introduction. Addison Wesley Publishing, Boston (2013)
5. Zhao, Z.L., Wang, C.D., Lai, J.H.: AUI&GIV: recommendation with asymmetric user influence and global importance value. PLoS ONE **11**(2), e0147944 (2016)
6. Allab, K., Labiod, L., Nadif, M.: A Semi-NMF-PCA unified framework for data clustering. IEEE Trans. Knowl. Data Eng. **29**(1), 2–16 (2017)

7. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264 (2007)
8. Guo, G., Zhang, J., Thalmann, D.: Merging trust in collaborative filtering to alleviate data sparsity and cold start. Knowl.-Based Syst. **57**, 57–68 (2014)
9. Tan, B., Song, Y., Zhong, E., Yang, Q.: Transitive transfer learning. In: KDD, pp. 1155–1164 (2015)
10. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296 (2011)
11. Wei, S., Zheng, X., Chen, D., Chen, C.: A hybrid approach for movie recommendation via tags and ratings. Electron. Commer. Res. Appl. **18**, 83–94 (2016)
12. Xin, X., Liu, Z., Lin, C.Y., Huang, H., Wei, X., Guo, P.: Cross-domain collaborative filtering with review text. In: IJCAI, pp. 1827–1834 (2015)
13. McAuley, J.J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: SIGIR, pp. 43–52 (2016)
14. Song, T., Peng, Z., Wang, S., Fu, W., Hong, X., Yu, P.S.: Review-based cross-domain recommendation through joint tensor factorization. In: Candan, S., Chen, L., Pedersen, T.B., Chang, L., Hua, W. (eds.) DASFAA 2017. LNCS, vol. 10177, pp. 525–540. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55753-3_33
15. Moreno, O., Shapira, B., Rokach, L., Shani, G.: TALMUD: transfer learning for multiple domains. In: CIKM, pp. 425–434 (2012)
16. Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Zhu, C.: Personalized recommendation via cross-domain triadic factorization. In: WWW, pp. 595–606 (2013)
17. Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J.: Cross-domain recommendation via cluster-level latent factor model. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8189, pp. 161–176. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40991-2_11
18. Liu, Y.F., Hsu, C.Y., Wu, S.H.: Non-linear cross-domain collaborative filtering via hyper-structure transfer. In: ICML, pp. 1190–1198 (2015)
19. Chen, J., Zhou, J., Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: KDD, pp. 42–50 (2011)
20. Candes, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis. J. ACM **58**(3), 1–39 (2011)
21. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: KDD, pp. 650–658 (2008)
22. Iwata, T., Takeuchi, K.: Cross-domain recommendation without shared users or items by sharing latent vector distributions. In: AISTATS, pp. 379–387 (2015)
23. Ren, S., Gao, S., Liao, J., Guo, J.: Improving cross-domain recommendation through probabilistic cluster-level latent factor model. In: AAAI, pp. 4200–4201 (2015)