

Cross-domain Recommendation by Combining Feature Tags with Transfer Learning

Yuyu Yin¹, Xin Wang², Jilin zhang^{3*} and Jian Wan⁴

*School of Computer Science and Technology, Hangzhou Dianzi University,
Hangzhou, China*

¹yinyuyu@hdu.edu.cn, ²sallybin@163.com, ³jilin.zhang@hdu.edu.cn,
⁴wanjian@hdu.edu.cn

Abstract

Most recommender systems based on collaborative filtering aim to provide recommendations for a user in one domain. But data sparsity is a major problem for collaborative filtering techniques. Recently, many scholars have proposed recommendation models to alleviate the sparsity problem by transferring rating matrix in other domains. But different domains have different rating scales (e.g., rating scale may be 1-5 or 1-10). Simple process for the rating scale does not reflect the real situation. The diversity of rating scales may cause the opposite effect, making the recommendation results more imprecise. In this paper, we propose a transfer model which learning the common feature tags from other domain. This model ignores the difference of rating scales between two domains, and focus on studying the feature tags. Using its own rating values to fill the missing value. We first get the different types of users (items) based on non-negative matrix tri-factorization from auxiliary domain. The process we call the user (item) clustering. Than we can get a BP neural network which can judge the type of user according to user's feature tags by studying the features of different types of users (items). And we classify the user (items) which from target domain by exploiting the trained neural network and the users' feature tags of target domain. Use the average rating values of the same type of users (items) to fill the missing value of target domain. We perform extensive experiments to show that our proposed model outperforms the state-of-the-art CF methods for the cross-domain recommendation task.

Keywords: *Transfer Learning, Sparsity Reduction, Matrix Factorization*

1. Introduction

Collaborative filtering (CF) in recommendation systems aims at predicting an active user's ratings on a set of items belonging to only a single domain (e.g., movies or books) based on a collection of like-minded user's rating records on the same set of items. A major problem in CF is the data sparsity problem, because in practices, Most users are not possible to rate most items, they can only rate the small part, the new system is especially sharp. Thus, this can easily lead to over-fitting problems by the collaborative filtering technology, and resulting in a bad recommendation result. Although there are some methods based on matrix factorization technique can better processing data sparseness problem, but the problem of sparse cannot be fundamentally improved.

In fact, User will show the same preferences in different domain. For example, the users who like to watch horror movies will like looking horror books. The correlation of the user's preferences in different domain was confirmed by data statistics [1]. So we have reason to believe that different domains share common knowledge. User's preference in one domain can be transferred into other domain. Thus, some recommendation models

* Corresponding Author

which use the thinking of transfer learning have been proposed to alleviate the data sparsity problem. They transfer the user-item rating knowledge from a dense rating matrix (we call it auxiliary matrix) to other related rating matrices (we call it target matrix). So the missing values in target matrix are filled. Singh and Gordon proposed a model that using matrix factorization to obtain the potential user-item rating pattern between different domain, and filling the missing value by these rating patterns [2]. Li, Yang, and Xue establish a bridge between the two domains at a cluster-level of user-item rating patterns in order to transfer more useful knowledge from the auxiliary domain. They first propose an algorithm that compress the ratings in the auxiliary rating matrix into an informative and compact cluster-level rating pattern then they propose another algorithm that reconstructing the target rating matrix by expanding the rating pattern [3]. But these methods do not consider the difference of rating between auxiliary domain and target domain (*e.g.*, the rating scales is different that is mentioned above), and also not take the correlation between the two domains into account. It will cause negative transfer effect, make the recommendation accuracy is lower, and even recommended results irrelevant if transferring knowledge between two unrelated domains. It is important to note that many recommendation systems will add tags to the customer for marking the user's browsing preferences or user characteristics. The same as the items we can use these tags as a bridge to transfer knowledge between two domains. Thus, we can avoid the diversity and irrelevance. Our proposed model is based on the assumption that there are common tags describing user's preferences or item's factors between two different domains for example, in movies domain, there are some tags to describe movie feature, like Sadness, humor, suspense, horror etc. In books domain, Tags like Sadness, humor, suspense, horror also can be used to describe books. It is potentially beneficial to exploit tags shared between those domain for the purpose of recommendation.

In this paper, we propose a cross-domain CF algorithm based on feature tags learning which connect different domain by the shared tags of these domains. The method can alleviate the data sparsity problem, and improve the recommendation accuracy. The algorithm is consisted of three parts. Firstly, we classify the users (items) from the auxiliary matrix which has a dense rating matrix according to the knowledge that the same types of users have a similar rating for the same items. Secondly, we can get a BP neural network which can judging the type of user according to user's feature tags by learning the features of different types of users (items). Lastly, we classify the users (items) from target domain by exploiting the trained neural network and the users' feature tags of target domain. Use the average rating values of the same type of users (items) to fill the missing value of target domain.

The paper is structured as follows. In Section 2 we briefly introduce how to cluster user (item) of the auxiliary domain. In Section 3 we simply show how to learn the features of all types of users (items). In Section 4 we explain how to predict the missing values of the target domain. Then we describe experiments on several real world datasets, and provide comparisons with state-of-the art methods in Section 5. Related work is introduced in Section 6. In Section 7, we present conclusions.

2. User Clustering

Under normal circumstances, the same type of user will have a similar rating for the same type of items. For example, a group of people who like to watch comedy movies will rating a high score to the comedy type of movie, and rating a low score to the sadness type of movie. We can use the feature to classify the users into different types. The same as the items. In this paper, our clustering algorithm is based on the orthogonal non-negative matrix tri-factorization (ONMTF) [4] which has been proved to be equivalent to a two-way clustering. ONMTF can not only for user clustering, but also can for items clustering. The algorithm makes use of the interrelatedness between users and items,

leading to better performance than other clustering methods. we slightly modify the algorithm, to make it better apply to this paper.

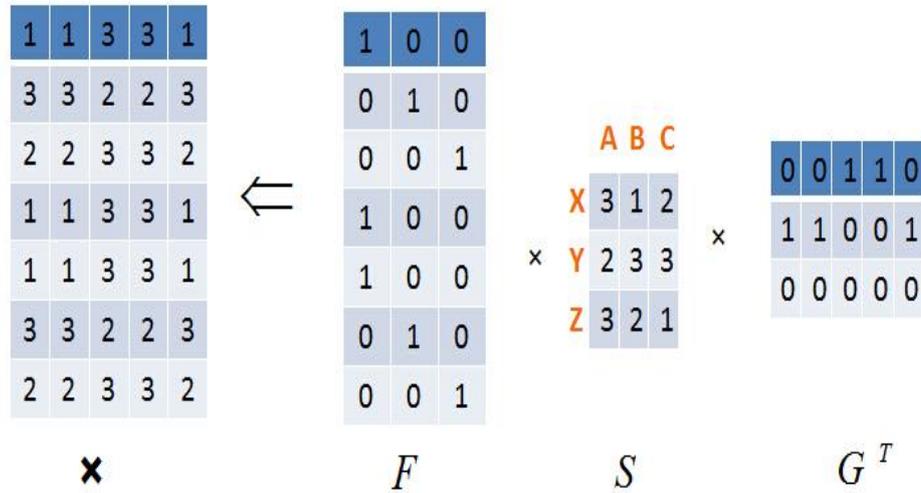


Figure 1. The User Clustering Diagram based on the ONMTF

The rating matrix of auxiliary domain can be decomposed into three non-negative matrix, As show in Figure 1, X is auxiliary task which is represented as a dense $m \times n$ rating matrix, S is a $k \times l$ rating matrix, the size of k and l can't be set too big or too small. if they are set too large, It will take a long time to calculate the matrix tr-factorization. If they are set too small, it will lead to the over-fitting problem. The X, Y, Z respectively represent three types of users (e.g., students, engineers, workers). And A, B, C represent three types of items (e.g., clothing, electronic, books). F is a $m \times k$ non-negative rating matrix. And row vector represents the feature information of each user. The same or similar row vector means they are the same types of users. G is a $n \times k$ non-negative rating matrix and column vector represents the feature information of each item. The same or similar column vector means they are the same types of items. This is an important basis for our classification. This X approximation can be achieved by the following matrix norm optimization:

$$\min_{U \geq 0, V \geq 0, S \geq 0} \|X - FSG^T\|_F^2 \quad (1)$$

$$G_{jk} \leftarrow G_{jk} \sqrt{\frac{(X^T FS)_{jk}}{(GG^T X^T FS)_{jk}}} \quad (2)$$

$$F_{ik} \leftarrow F_{ik} \sqrt{\frac{(XGS^T)_{ik}}{(FF^T XGS^T)_{ik}}} \quad (3)$$

$$S_{ik} \leftarrow S_{ik} \sqrt{\frac{(F^T XG)_{ik}}{(F^T FSG^T G)_{ik}}} \quad (4)$$

After taking formula (2-4) into (1) many times, we can make X reach the minimum value which is the optimal result. At this point, we can get F, S, G. And F represents the feature of users, G represents the feature of items. The ratings values of F and G are decimal which are located between 0-1. If the decimal in F and G is

large. That means it has an important effect on the user (item). Before user clustering, we need to pretreatment the F and G. Each row in the F (each column in the G), the maximum value of the decimal we denoted as M. In the same decimal digit of M in each row (column), we will update 1 if other decimal located between $[M/2, M]$, the rest of the decimal update 0. In this way, we can get the row vectors which represent the feature of users (items).

After the completion of the preprocessing, we calculate the similarity of each row (column) vector in F (G) by the formula (5). And we set a similarity threshold which is denoted as p. We think that the users are the same type if the similarity of the row vectors corresponding to the users is larger than p. Through similarity comparison of every users (goods), we can get the types of every users (items). In the process, some user may belong to the user type of X, and also belong to the user type of Y. We classify the user to the type which their similarity is largest.

$$sim(i, j) = \frac{\sum_{j=0, L_j(j)=1}^n Li(j)}{n} \quad (5)$$

The formula to calculate the similarity of two row(column) vector i, j represents the position of the matrix.

3. Feature Tags Learning

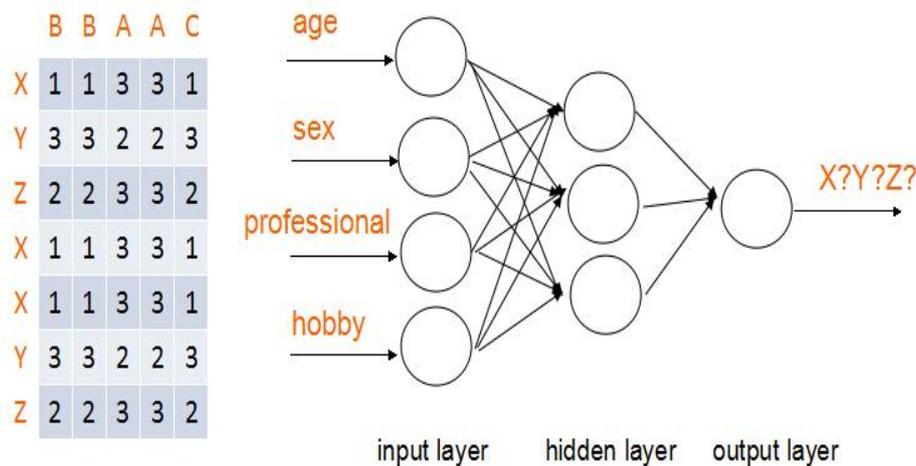


Figure 2. BP Neural Network for Feature Tags Learning

In the recommendation system, there are usually some tags to describe the user's feature or the item's feature. And these tags can show the personality of the users. In common, different types of users have different tags. For example, Girls generally like to keep an eye on clothes, cosmetics and the boy generally focus on electronics, sports items. We can judge the type of user (item) by their tags. After user (item) clustering in Section 2, we can get different types of users. So we have many tag samples to describe a type of user. And we can get a model which can judge the type of user according to their tags by learning these tag samples. In this paper, we adopt BP neural network for feature learning. Because BP neural network can learn and store a large amount of input - output schema mappings (in this paper, the mapping relation is represented as feature - type), To adjust the network weights and threshold by back propagation, and make the neural network convergence. We establish a three layers BP neural network which input layer node number is n (the

size of n is the number of features of the user), the number of hidden layer nodes is $n1$ (the value can be set by formula (6)), the number of output layer nodes is m (the size of m is the type of user). And we quantify the features of user and the type of user. After the training, we can get the BP neural network which can judge the type of user (item) by their tags.

$$n1 = \sqrt{n + m} + a \tag{6}$$

In the formula, n represents the number of input nodes, m represents the number of output nodes. a is a constant and its value in $[1 \sim 10]$.

4. Predicting the Missing Value

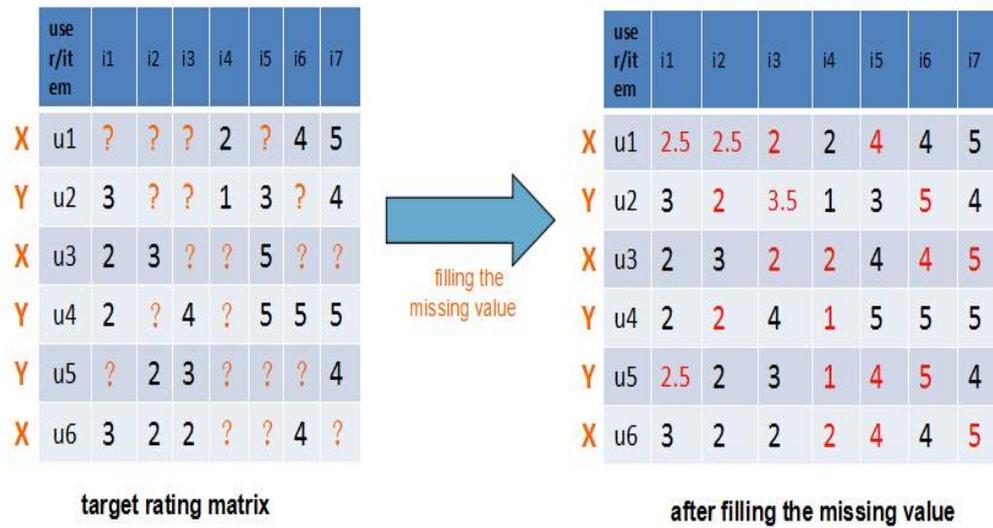


Figure 3. Filling the Missing Value which are Denoted by the ‘?’.

Mentioned above, At present, researchers have introduced transfer rating matrix in other domains for solving the data sparsity problem. But the results are not optimal because different domains have different rating scales and the domains may not have relevance. We can make the user’s features (not the rating matrix) as the connection of two domains and exploiting the rating of target domain to fill the missing value. This is a good solution to solve the problem which has been mentioned. Although the rating information is relatively sparse in target domain, but the users’ tags is very rich. We can judge the type of each user (item) from target domain by the trained BP neural network in Section 3 and the users’ tags which from target domain we can predict the missing value in target domain by the average rating value of the other users who have the same type with the target user rating scores. Because the same type of user will have the similar rating value on the same items. Thus, we can solve the data sparsity problem in target domain. As show in Figure 3. The notation of ‘?’ in target rating matrix represents the missing value. After user clustering in Section 1. we can get the results of classification. $(u1, u3, u6) \in X$, $(u2, u4, u5) \in Y$, The rating score of $u1$ rating on $i1$ is null. The rating score of $u3$ rating on $i1$ is represented as $r(u3, i1)=2$ and the rating score of $u6$ rating on $i1$ is represented as $r(u6, i1)=3$. Because $u1, u3, u6$ are all the same type of X , we can use the average of $r(u3, i1)$ and $r(u6, i1)$ to predict the rating score of $u1$ rating on $i1$. We can fill the rest of missing value by the formula (7).

$$r(\mathbf{u}_i, \mathbf{i}_j) = \frac{\sum_{(\mathbf{u}_x, \mathbf{i}_j) \in G} r(\mathbf{u}_x, \mathbf{i}_j)}{n} \quad (7)$$

Where $r(\mathbf{u}_i, \mathbf{i}_j)$ denotes the prediction score which user i rating on item j . \mathbf{u}_x denotes the user who has the same type of \mathbf{u}_i and has rated on \mathbf{i}_j . n denotes the number of \mathbf{u}_x .

5. Experiments

In this Section, we examine how our proposed model behaves on real-world rating datasets and compare it with several state-of-the-art single-domain recommendation models and cross-domain recommendation models.

5.1. Data Sets

Our experiments have the requirement that the two domains must have the common tags. So we decide to divide the datasets which from one domain into two parts. In order to simulate the experimental environment. A dense rating matrix we call auxiliary matrix. And another sparse rating matrix we call target matrix. For the experiments we have used two benchmark real-world datasets for performance evaluation:

- MovieLens dataset (1): A movie rating dataset contains more than 100,000 movie rating with the scales from 1 to 5 provided by 943 users on 1682 movies. The dataset has a detailed description tags for each user and each film which contains 16529 unique tags with 95580 tag assignments from users to movies. In order to meet our experiment requirement we extract a sub-matrix to simulate the auxiliary task by selecting 250 users and 250 movies with most rating (rating ratio 58.9%), and we also extract another sub-matrix as the target task with 300 users and 300 movies, the rating ratio just 8.9%.
- Book-Crossing dataset (2): A book rating dataset contains more than 1.1 million rating with the scales from 0 to 9 provided by 278,858 users on 271,379 books. As the same. The dataset has a detailed description tags for each user and each book. We extract two sub-matrix with the same standard like MovieLens dataset the rating ratio of auxiliary task is 55.4% and target task is 9.8%.

5.2. Compared Models

The purpose of our model is filling in the missing values by transfer knowledge from other domain to alleviate the sparsity problem so, our model is compared with several state-of-the-art single-domain recommendation models and cross-domain recommendation models:

- NMF (Non-negative Matrix Factorization) based model [6]: a single-domain model which uses non-negative matrix factorization method to learn the latent factors in each domain and predict the missing value.
- CBS (Scalable cluster-based smoothing) based model: a single-domain model which takes the average of the observed ratings in the same user cluster to fill in the missing values.
- CBT (CodeBook Transfer) based model [2]: a cross-domain model which fill in the missing values with corresponding entries in the reconstructed rating matrix by expanding the codebook.
- CLFM (Cluster-Level Latent Factor Model) [7]: a cross-domain model which can not only learn the common rating pattern shared across domains with the flexibility in controlling the optimal level of sharing, but also learn the domain-specific rating patterns of users in each domain.

- FTLM(Feature Tags Learning Model):our proposed model which uses the Feature Tags to fill in the missing value. Wherever Times New Roman is specified, Times Roman, or Times may be used. If neither is available on your word processor, please use the font closest in appearance to Times New Roman that you have access to. Please avoid using bit-mapped fonts if possible. True-Type 1 fonts are preferred.

5.3. Evaluation Protocol

Following the work in [3], we examine the compared models for the cross-domain recommendation task. In each dataset, we randomly selected 250 user ratings and 250 items as the training data, also randomly selected 200 users and 200 items for testing. In the process of user clustering ,the number of users clusters K and item clusters L in each rating dataset have been chosen in the range of[5, 35].In the experiments, we choose 15×15.Because this kind has the lest MAE value. We can get the conclude from Fig.4. And for each test user, we consider to keep different sizes of the observed ratings as the initialization of each user .In the experiments, There are three sizes(Given-5,Given-10,Given-15).That means 5,10,15 ratings of each test user in dataset are given while the remaining ratings are used for evaluation. And the given observed ratings are randomly selected 10 times, so that the result in Table 1 shows the average results over 10 splits. We compare the performances of different models under different configurations. The parameters of different models have been manually tuned and we report here the best results obtained based on the optimal combination of many parameter settings.

5.4. Evaluation Metric

We adopt mean absolute error (MAE) as the evaluation metric, defined as:

$$MAE = \sum_{i \in T} |r_i - r_i^*| / |O| \quad (8)$$

Where O denotes the set of user-item pairs whose ratings need to be predicted in the test set. We denote by |O| the size of the set O. r_i is the true value and r_i^* is the predicted value. A smaller value of MAE means a better performance.

5.5. Experimental Results

Table 1. MAE Performance of the Compared Models on MovieLens and Book-Crossing

datasets			NMF	CBS	CBT	CLFM	CFTM
MovieLens	MAE	ML-Given5	0.9785	0.9862	0.9451	0.9358	0.9218
		ML-Given10	0.9587	0.9875	0.9356	0.9214	0.8954
		ML-Given15	0.9542	0.9756	0.9217	0.9186	0.8912
Book-Crossing	MAE	ML-Given5	0.7586	0.7153	0.6918	0.6854	0.6654
		ML-Given10	0.7262	0.6972	0.6758	0.6873	0.6413
		ML-Given15	0.7174	0.6893	0.6614	0.6752	0.6345

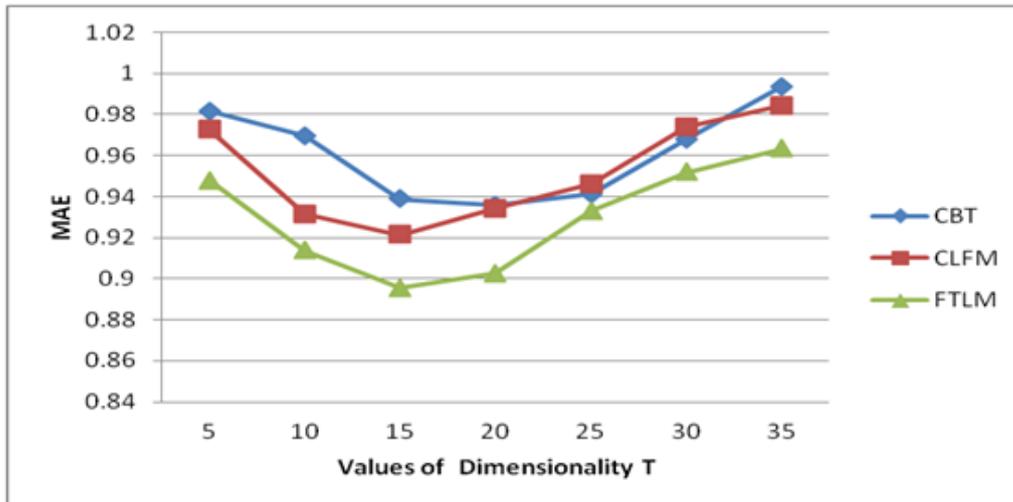


Figure 4. MAE Performance of the Compared Models with the Value of Dimensionality T in MovieLens Dataset

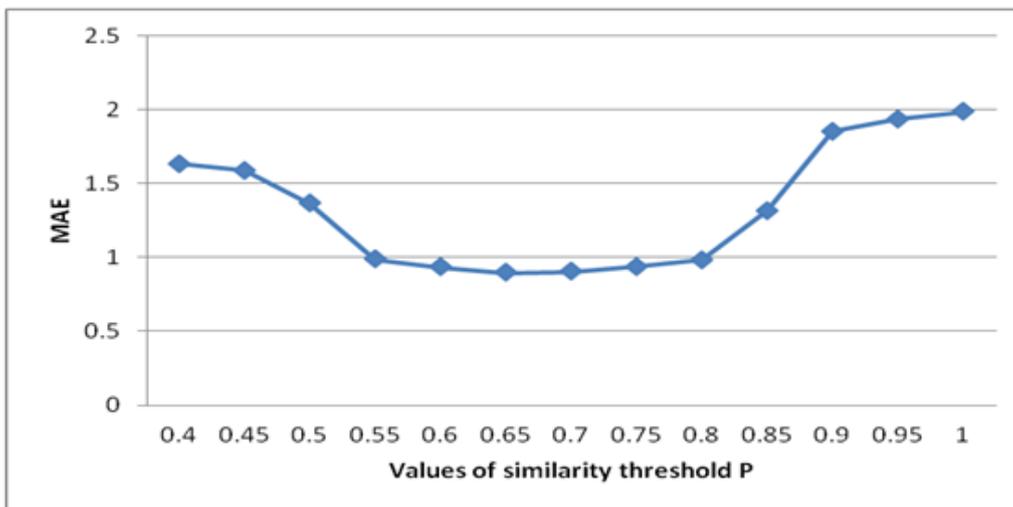


Figure 5. The Impact of the Similarity Threshold p

The comparison results are reported in Table 1. The minimum MAE were represented in bold. We can see that our method clearly outperforms all the other compared methods under all the testing configurations on all the two data sets. Table 1 describes the MAE value of single domain recommendation model and cross domain model on the dataset of MovieLens and Book-Crossing. In particular, We split a dataset into two datasets. One is a dense rating matrix which we call auxiliary matrix. And another is a sparse rating matrix which we call target matrix. We can clearly see two points from the overall trend. First, The value of MAE of single domain recommendation model is generally higher than the recommended value of MAE of cross-domain recommendation. It shows that transfer knowledge between different domains can solve the data sparsity problem and enhance the accuracy of collaborative filtering recommendation system. Second, The test users are given more. The value of MAE is lower, because it will result in over-fitting problem more easily if the test rating information are given less. With the comparison of MAE on different models, We can see clearly that CBS performs better than the NMF, which implies that the cluster-level based methods can gain meaningful knowledge from user and item clusters. The CBT and CLFM perform

better than the CBS and NMF, which implies that there are common rating information between different domains we can fill the missing value in target domain better by the common rating information. The CFTM performs better than the CBT and CLFM, which implies that the methods which transfer the user's preference information between different domains are more accurate than the methods which transfer rating information between different domains.

Figure 4 provides the performances of the compared models with the value of dimensionality in MovieLens dataset under the configuration of given 10 ratings observed for each test user. From the Figure, we can see the overall trend .MAE values decrease quickly at first when the dimensionality T reach a value, MAE values increase. We can find the curve of FTLM when the T is 15, MAE value reach the minimum value. That is why we choose 15×15 compressed matrix in our experiments.

In Section 2, we have a similarity threshold p to judge the type of users in FTLM. Different values of p will get the different types of user group. From the Figure 5, we can see the impact of the similarity threshold p on MAE values. When the similarity threshold p reach 0.65, The MAE values gets the minimum.

6. Related Work

In this Section, We mainly describe the collaborative filtering recommendation in single domain, and cross-domain CF recommendation.

CF in single domain:

In the field of single recommendation system, Collaborative filtering recommendation is a popular technology in information filtering and information system. CF recommendation is different with the traditional content-based filtering recommendation. it analysis the users' interests, and find the similar user of the target user in user groups. Than predict the target user's rating on item by the similar users rating on the same item. Collaborative filtering algorithm is mainly divided into two kinds. One is item-based CF [8] which make recommendations based on the similarity between the item. Another kind is user-based CF [9] which make recommendations based on the similarity between the user. Matrix factorization (MF) techniques [10] have become one of the most popular CF approaches, due to the advantages of accuracy and scalability. Generally, MF techniques learn latent features of users and items from the observed ratings in a user-item matrix. The learned features are then further used to predict unobserved ratings.

A number of Internet applications and online retailers have employed CF. It makes recommendations by only considering user ratings which are independent of the domain content. Meanwhile CF's dependency on user ratings causes two key drawbacks, known as the cold-start and the rating sparsity problem. Also, it can be challenging for the system to form neighbors with users of unusual tastes, since the correlation of ratings between these unusual users and mainstream normal users could be low, resulting in poor recommendations.

Cross-Domain CF:

The data sparsity problem will result in a bad recommendation results. So the cross domain CF models are used to solve the problem. To fill the missing value in target matrix by transfer rating matrix in auxiliary domain. Recently, a lot of cross domain models are proposed. Mehta and Hofmann [12] consider the scenario involving two systems with shared users and use manifold alignment methods to jointly build neighborhood models for the two systems. They focus on making use of an auxiliary recommender system when only part of the users are aligned, which does not distinguish the consistency of users' preferences among the aligned users. Li, *et al.*, present a transfer

learning approach [2] that mitigates the sparsity problem in collaborative filtering. The algorithm is consist of two parts. They first compress the ratings in the auxiliary rating matrix and compact cluster-level rating pattern. they call the rating patterns CodeBook. This process is called Codebook Construction. Then the rating patterns are transferred to a target domain with a high rating sparsity. And Codebook are used for predicting these missing ratings. Assuming that there exists implicit correspondence between the user/item clusters of the auxiliary task and those of the target task. This process is called Codebook Transfer. In future work [5] Li, *et al.*, extend the previous approach by means of a probabilistic model in which a user or item belonging to a particular cluster is not binary, but is described in terms of probability density function. In this case a common model is built from the ratings of all the considered domains, without requiring a dense source domain. The main advantage of both approaches is that they do not require user/item overlap, which makes them applicable in many real situations where profiles from isolated systems have to be merged or linked. Sheng, *et al.*, think the related domains do not necessarily share such a common rating pattern, and diversity among the related domains might outweigh the advantages of such common pattern, which may result in performance degradations. So they propose a cluster-level based latent factor model [6] to enhance the cross-domain recommendation, which can not only learn the common rating pattern shared across domains with the flexibility in controlling the optimal level of sharing, but also learn the domain-specific rating patterns of users in each domain that involve the discriminative information propitious to performance improvement.

7. Conclusion

In this paper, we propose a cross-domain CF model based on feature tags learning. The model transfer knowledge by common tags between two domains. We classify the users in auxiliary domain into different type based on the orthogonal non-negative matrix tri-factorization clustering algorithm, and build a BP neural network to learn the user's tags of different types. Thus, the BP neural network can judge the type of any user who has the same tags of auxiliary domain. Than we classify the users in target domain into different type through the trained BP neural network. We can use the average rating value of the other same type of users to fill the missing value in target matrix because the knowledge that the same type of user will have a similar rating on items. Thus, we can solve the data sparsity problem. The advantages of our proposed model are that we ignore the diversity of different domain, and it is easy to understand the model in our mind.

The experimental results show that our proposed model can transfer knowledge from anther domain by tags successfully and ignore the difference of rating scales between two domain. The results also validated that the method outperform the state-of-the-art methods for cross-domain recommendation task.

There are still several extensions to improve our work. For example, our proposed model need two domains have the common tags which used to describe the features of user. But in practice, this is difficult to achieve. So we need to find the correlation of tags in different domains. We can use the related tags instead of the common tags if we know the correlation. In the next work, we will study the correlation of tags between different domains.

Acknowledgements

This paper is granted by National Natural Science Foundation of China under Gran (No. 61402140), Zhejiang Provincial Natural Science Foundation(No.LY12F02003), China Postdoctoral Science Foundation(No.2013M540492) and The National Key Technology R&D Program under Grant(No.2012BAH24B04).

References

- [1] P. Winoto and T. Tang, "If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations", *New Generation Computing* vol. 26, no. 3, (2008), pp. 209-225.
- [2] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization", *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, (2008), pp. 650-658.
- [3] B. Li, Q. Yang and X. Xue, "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction", *IJCAI*, (2009), pp. 2052-2057.
- [4] W. Pan, E. W. Xiang, N. N. Liu and Q. Yang, "Transfer Learning in Collaborative Filtering for Sparsity Reduction", *AAAI*, (2010), pp. 230-235.
- [5] C. Ding, T. Li, W. Peng and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering", *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, (2006), pp. 126-135.
- [6] B. Li, Q. Yang and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model", *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, (2009), pp. 617-624.
- [7] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari and J. Guo, "Cross-domain recommendation via cluster-level latent factor model", *Machine Learning and Knowledge Discovery in Databases*, Springer Berlin Heidelberg, (2013), pp. 161-176.
- [8] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms", *Proceedings of the 10th international conference on World Wide Web*, ACM, (2001), pp. 285-295.
- [9] S. J. Ko and J. H. Lee, "User preference mining through collaborative filtering and content based filtering in recommender system", *E-Commerce and Web Technologies*. Springer Berlin Heidelberg, (2002), pp. 244-253.
- [10] J. L. Herlocker, J. A. Konstan, A. Borchers and J. Riedl, "An algorithmic framework for performing collaborative filtering", *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, (1999), pp. 230-237.
- [11] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems", *Computer*, (2009), pp. 30-37.
- [12] B. Mehta and T. Hofmann, "Cross system personalization and collaborative filtering by learning manifold alignments", *KI 2006: Advances in Artificial Intelligence*. Springer Berlin Heidelberg, (2007), pp. 244-259.

Authors



Yuyu Yin, he received the Doctors degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently an assistant professor at Hangzhou Dianzi University. His research interests include service computing, cloud computing and middle ware techniques.



Xin Wang, he received the Bachelor Degree of software engineering in Hubei University of Technology, Hubei, China, in 2012. He is now studying the Master of Technology of Computer in Hangzhou Dianzi University, China. His research interest is Cross Domain Recommendation.



Jian Wan, He received the PhD degree in Computing Application Technology from Florida Atlantic University, in 1998. He is currently a professor in software engineering in Hangzhou Dianzi University, China. His research interests include Cloud Computing and Mass Storage.