A Multifaceted Model for Cross Domain Recommendation Systems

Jianxun Lian[1](✉), Fuzheng Zhang[2], Xing Xie[2], and Guangzhong Sun[1]

[1] University of Science and Technology of China, Hefei, Anhui, China
lianjx@mail.ustc.edu.cn,gzsun@ustc.edu.cn
[2] Microsoft Research, Beijing, China
{fuzzhang,xingx}@microsoft.com

Abstract. Recommendation systems (RS) play an important role in directing customers to their favorite items. Data sparsity, which usually leads to overfitting, is a major bottleneck for making precise recommendations. Several cross-domain RSs have been proposed in the past decade in order to reduce the sparsity issues via transferring knowledge. However, existing works only focus on either nearest neighbor model or latent factor model for cross domain scenario. In this paper, we introduce a Multifaceted Cross-Domain Recommendation System (MCDRS) which incorporates two different types of collaborative filtering for cross domain RSs. The first part is a latent factor model. In order to utilize as much knowledge as possible, we propose a unified factorization framework to combine both CF and content-based filtering for cross domain learning. On the other hand, to overcome the potential inconsistency problem between different domains, we equip the neighbor model with a selective learning mechanism so that domain-independent items gain more weight in the transfer process. We conduct extensive experiments on two real-world datasets. The results demonstrate that our MCDRS model consistently outperforms several state-of-the-art models.

Keywords: knowledge-boosted recommender systems, collaborative filtering, cross domain, knowledge transferring.

1 Introduction

With the boosting of online services, recommendation systems (RS) are playing an increasingly important role in filtering information for customers. Since most users are only connected to a small set of items, data sparsity becomes a major bottleneck for building an accurate RS. This is especially true for newly joined users/items, which is known as the cold start problem. To address this problem, researchers have introduced cross domain RSs which can transfer knowledge from some relatively dense source domains to the target domain. They assume that there are some consistent patterns across domains. Take book domain and movie domain for example, users with similar interests in the movie domain may also have similar interest in the book domain (rule of collaborative filtering); and users who like fantasy movies have a higher probability of liking fantasy books than users who do not (rule of content-based filtering). Therefore, when we do not have enough data on the book domain, leveraging the data from movie domain appropriately can improve the quality of book recommendations.

According to whether the items or users from different domains have overlapping or not, existing literatures for cross domain RSs can be roughly divided into two categories: with overlap and without overlap. CMF [26], CST [21], TCF [20], MV-DNN [8] and [9] assume the users or/and items from multiple domains are fully or partially aligned, and the knowledge is transferred through the known common user/item factors; On the contrary, CBT [12], RMGM [13] and [6] do not require any overlap of users/items between auxiliary domains and the target domain. They transfer useful knowledge through cluster-level rating patterns.

In this paper, we assume the users between multiple domains are overlapped. This can be the case when a company owns multiple products but only a few of them provide enough data, while the others' data are too sparse to build effective models; or when a company wants to promote new services or products to its customers, they can leverage customer data on existing services or products. The aforementioned cross domain methods only focus on transferring either content-based filtering or collaborative filtering knowledge. However, models which combine the two algorithms have been widely discussed in the single domain case. Thus we consider the first challenge: will it produce a large amount of improvement if we transfer both content-based filtering and CF across different domains simultaneously? To address the the question, we propose a unified factorization model to transfer both CF and content-based filtering cross domains. The key idea is to learn content embeddings so that the content-based filtering has a same latent factor formulation as model-based CF. Actually we propagate the CF not only to user-item level but also to user-content level.

Meanwhile, we notice that the multifaceted model can outperform the pure latent factor model [10]. However, there may be some inconsistency between different domains so that the neighborhood built from the source domain may not hold for the target domain. Take news and movie recommendation for instance, suppose Bob and David come from the same city, they may be similar in the news domain because they both love to read news related to the city. However, the similarity between them in movie domain may be low, since their interest in movies are not quite related to where they live. Motivated by this, we propose a novel selective mechanism which can iteratively learn a specific weight for each item in the source domain. The goal is to transfer preference on domain-independent items rather than domain-dependent items.

The key contributions of this paper are summarized as follows:

- We introduce the multifaceted model for cross domain RSs, which includes collective learning (the latent factor model) and selective learning (the neighbor model) modules.
- We propose a novel selective neighbor model to automatically assign weights to items so that domain inconsistency problem can be mitigated.
- We propose a unified factorization model to collectively learn content-based filtering and CF for cross domain RSs. Although models for combining the

CF and content have been widely studied in the single domain situation, we aim to figure out the improvement in the cross domain scenario.

- We conduct extensive experiments on two real-world datasets, and the experimental results reveal that our proposed model consistently outperform several baseline methods.

2 Content-boosted CF for Cross Domain RS

Suppose we have two domains: the target domain and the source domain. For each domain the data are comprised of user-item ratings and item attributes. In the target domain, we denote the user-item ratings as a matrix $\mathbf{R}^{(1)} = [r_{ui}]_{N\times M^{(1)}}$ with $r_{ui} \in \{[r_{min}, r_{max}], ?\}$, where ? denotes a missing value, N and $M^{(1)}$ denote the number of users and items respectively. The item content is denoted as a matrix $\mathbf{A}^{(1)} = [a_{ik}]_{M^{(1)}\times T^{(1)}}$, where each row represents an item, $T^{(1)}$ is the number of attributes, $a_{ik} \in [0, 1]$ is a normalized weight on attribute k, and a_{ik}=0 indicates item i does not have attribute k. Our goal is to predict the missing value in $\mathbf{R}^{(1)}$. For various reasons $\mathbf{R}^{(1)}$ is sparse, and we want to improve the model with the help of the auxiliary domain, whose corresponding data is denoted as $\mathbf{R}^{(2)} = [r_{ui}]_{N\times M^{(2)}}$ and $\mathbf{A}^{(2)} = [a_{ik}]_{M^{(2)}\times T^{(2)}}$. Note that the users are fully or partially overlapped, and we know the mapping of users between domains.

A widely used method in collaborative filtering is latent factor model. It factorizes the rating matrix $\mathbf{R}_{N\times M}$ into two low-rank matrices, $\mathbf{U}_{N\times D}$ and $\mathbf{V}_{M\times D}$, as latent factors for users and items. In probabilistic matrix factorization (**PMF**) model [24], these latent factors are assumed to be generated from zero-mean spherical Gaussian distributions, while each rating is generated from a uni-variate Gaussian. Actually, the the PMF model only learns from the user-item rating pairs and makes recommendations through clustering similar rating patterns. However, auxiliary signals such as user demographics and item attributes can help improve the restaurant recommendation model. Motivated by [19], we try to embed the attributes into a shared latent space, and then augment the item latent vector V_j with the weighted average of the embedded representation of attributes.

Formally, for each attribute k, $k \in 1..T$, we denote its embedding representation by $B_k = \langle b_{k1}, b_{k2}, ..., b_{kL}\rangle$. So we have a attribute embedding matrix $\mathbf{B}$ with each row indicates an attribute. The augmented latent vector for item j is denoted by $\widetilde{V}_j = \{V_j, P_j\}$ where $P_j = A_j\mathbf{B}$. The user latent vector is simply extended as $\widetilde{U}_i = \{U_{ia}, U_{ib}\}$, where U_{ia} denotes the original CF part while U_{ib} denotes the content preference part. Then the mean rating of user i on item j is changed as follows:

$$r_{ij} = \widetilde{U}_i \cdot \widetilde{V}_j = U_{ia} \cdot V_j + U_{ib} \cdot (A_j\mathbf{B}) \tag{1}$$

and the generative procedure is updated as follows:

(i) For a user i, sample two vectors: $U_{ia} \sim \mathcal{N}(\mathbf{0}, \sigma_{ua}\mathbf{I})$, $U_{ib} \sim \mathcal{N}(\mathbf{0}, \sigma_{ub}\mathbf{I})$

(ii) For an item j, sample a vector: $V_j \sim \mathcal{N}(\mathbf{0}, \sigma_v \mathbf{I})$.
(iii) For an attribute k, sample a vector: $B_k \sim \mathcal{N}(\mathbf{0}, \sigma_b \mathbf{I})$
(iv) For each user-item entry, generate the rating $r_{ij} \sim \mathcal{N}(U_{ia} \cdot V_j + U_{ib} \cdot \sum_k a_{jk} B_k, \sigma_r)$

We name the new model Probabilistic Preference Factorization (**PPF**).
Cross Domain Collective Learning. When $\mathbf{R}^{(1)}$ is too sparse, the learning of latent factors may be inadequate and thus lead to overfitting on the training set. Next we study how to make use of the data from the auxiliary domain. We adapt the Collective Matrix Factorization (**CMF**) [26] model to our situation. The key idea of CMF is to factorize multiple matrices jointly through a multi-task learning framework. Since we have one-one mapping on the user side between the target domain and source domain, we simply assume the users share the same latent factors across these domains: $\mathbf{U} = \mathbf{U^{(1)}} = \mathbf{U^{(2)}}$, just like the approach in [17]. We also apply the aforementioned **PPF** generative process to the source domain. This model can be easily extended by introducing priors on the hyperparamenters and applying fully Bayesian methods such as MCMC [23]. However, to avoid high computational cost, here we regard $\sigma_{u*}, \sigma_v, \sigma_r$ as constant hyperparameters and use grid search to find their best values. The parameter set are reduced to $\Theta = \{\mathbf{U_a}, \mathbf{U_b}, \mathbf{V^{(1)}}, \mathbf{V^{(2)}}, \mathbf{B^{(1)}}, \mathbf{B^{(2)}}\}$ and we want to maximize the following Bayesian posterior formulation:

$$p(\Theta|\mathbf{R^1}, \mathbf{R^2}) \propto p(\mathbf{R^1}, \mathbf{R^2}|\Theta) p(\Theta) \tag{2}$$

Maximizing Eq.(2) is equivalent to minimizing the following loss function:

$$\begin{aligned}\mathcal{L} = &\sum_{i,j} (r_{ij}^{(1)} - U_{ia} \cdot V_j^{(1)} - U_{ib} \cdot \sum_k a_{jk}^{(1)} B_k^{(1)})^2 \\ &+ \lambda_1 \sum_{i,j} (r_{ij}^{(2)} - U_{ia} \cdot V_j^{(2)} - U_{ib} \cdot \sum_k a_{jk}^{(2)} B_k^{(2)})^2 \\ &+ \lambda_2 \sum_i \left\| V_i^{(*)} \right\|^2 + \lambda_3 \sum_k \left\| B_k^{(*)} \right\|^2 + \lambda_4 \sum_i \left\| U_{ia}^{(*)} \right\|^2 + \lambda_5 \sum_i \left\| U_{ib}^{(*)} \right\|^2\end{aligned} \tag{3}$$

where we replace σ_* with λ_* after eliminating some constants for notation simplicity.

3 The Multifaceted Model

Yehuda ([10]) introduces a multifaceted model to smoothly merge the factor and neighborhood models. Following this idea, we plan to equip the **PPF** model with a neighborhood module. However, as we have discussed before, in the cross domain situation, the neighborhoods of the same user under different domains may be different. Thus it is questionable to compute the user similarity scores directly through items from the source domain. To address this problem, we propose a selective learning algorithm to assign weight for each item in the source domain. Our goal is to select those domain-free items to build an accurate neighborhood for the target domain. The new prediction rule becomes:

$$\hat{r}_{ij} = b_{ij} + \frac{w(i)}{\sqrt{|N^m(i,j)|}} \sum_{k \in N^k(i,j)} (r_{kj} - b_{ij}) s(i,k) \tag{4}$$

Here, b_{ij} represents the prediction of the aforementioned latent factor model. $w(i) = 0.2e^{-0.5|R_{i*}^{(1)}|}$ is a tradeoff function controlling the weight of the neighborhood model, and it decreases when the number of rating records of user i in the target domain increases. $N^m(i,j)$ represents user i 's top m nearest neighbors who have rated item j. $s(i,k)$ denotes the similarity value between user i and k. Due to the sparsity of target domain, we have to compute $s(i,k)$ using the data from source domain. In the traditional neighbor-based CF model, the similarity is static and calculated from their common rated items. Now we assign each item in the source domain with some parameters, which determine the weight of the item for the similarity calculation:

$$s(i,k) = e^{-\gamma_1 \frac{\sum_{t \in V(i,k)} D(i,k,t) * \beta(t)}{\sum_{t \in V(i,k)} \beta(t)}} \tag{5}$$

$V(i,k)$ indicates the common items rated by user i and k. $D(i,k,t)$ measures the difference of ratings on item t from user i and user k. We assign a small value, δ, to it when r_{it} equals to r_{kt}:

$$D(i,k,t) = max\{(r_{it} - r_{kt})^2, \delta\} \tag{6}$$

$\beta(t) \in (0,1)$ represents the weight of item t. We assume that the weight is determined via item id and item's content information. So for each item t, there is a corresponding parameter η_t; similarly, for each item attribute k, its corresponding parameter is denoted by α_k. We apply a logistic regression process to learn the optimal parameters:

$$\beta(t) = \frac{1}{1 + e^{-(\eta_t + \alpha_0 + \sum_k \alpha_k A_{tk})}} \tag{7}$$

Now, with eq.(4) we introduce a Multifaceted model for Cross Domain Recommendation Systems (**MCDRS**). Its first part is a collective latent factor model, in which we embed both collaborative filtering and content-based filtering in order to exploit as much knowledge as possible. And its second part is a novel neighborhood model whose main purpose is to learn residuals based on the latent factor model.

Learning There are several groups of parameters in our multifaceted model, i.e. latent factors for CF, latent factors for content, and weights in the neighbor model. It is hard for SGD to learn a good solution due to the fact that it does not discriminate between infrequent and frequent parameters. So we use AdaDelta [30] to automatically adapt the learning rate to the parameters, leading to the update rule:

$$\Delta x_t = -\frac{RMS[\Delta x]_{t-1}}{RMS[g]_t} g_t \tag{8}$$

where RMS represents the root mean squared criterion with exponentially decaying:

$$RMS[g]_t = \sqrt{E[g^2]_t + \epsilon} \tag{9}$$

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2 \tag{10}$$

In the experiments we set ρ=0.9 and ϵ=1e-8 .

4 Experiments

We evaluate the proposed model on two rating datasets: Douban[3] and MovieLens 20M[4].

Douban is a leading Chinese online community which allows users to record information related to multiple domains such as movies, books and music. Users can rate movies/books/songs on a scale from 1 to 5, and each movie/book/song has a list of tags (tag_id,count) indicating how many users have rated the tag_id on this item, and the tags can be used as the item's attributes. Thus Douban is an ideal source for our cross domain experiments. We build a distributed crawler to fetch the item information and user-item rating records from Douban. After filtering out users who appear in only one domain or have less than 20 ratings, and movies/books/songs which have fewer than 10 ratings, we randomly sample 100k users, 50k books, 30k movies and 30k music for experiments. Via switching the choice of source domain and target domain, we report the results of three cross domain tasks, i.e., $\langle$*movie* → *music*$\rangle$, $\langle$*movie* → *book*$\rangle$, and $\langle$*book* → *music*$\rangle$. In each task we split the target domain into training/valid/test set by 70%/15%/15%. In order to study the performance lift under different sparsity levels, we keep the validation and test set unchanged and randomly sample a subset from the training set with different sparsity levels of 1%, 2%, 5%, 8%, 10% correspondingly.

Besides the Douban dataset, we also evaluate the proposed algorithm on a widely used benchmark dataset, MovieLens 20M. It is currently the latest stable benchmark dataset from GroupLens Research for new research, and it contains more than 20 million ratings of 27,000 movies by 128,000 users, with rating score from 0.5 to 5.0. Because our proposed model is to study the combination of CF and content-based methods, we filter out the movies which have no tags, and then use tags as the movie attributes.

Since the MovieLens dataset has only one domain, we split the movies into two disjointed parts to simulate two different domains. This approach has also been used in existing works such as [21] and [20]. Specifically, denote the full rating matrix as $\mathbf{R}_{1\sim N,1\sim M}$. We take the first half sub-matrix $\mathbf{R}_{1\sim N,1\sim \frac{M}{2}}$ as the target domain, while the other half $\mathbf{R}_{1\sim N,\frac{M}{2}+1\sim M}$ is the auxiliary domain. Again we split the target domain into training/valid/test set by 70%/15%/15% and extract several subsets from the training set according to different sparsity levels.

[3] http://www.douban.com

[4] http://grouplens.org/datasets/movielens/

4.1 Baselines and Evaluation Metrics

We compare our model with the following methods:
Bias Matrix Factorization (BMF) [11]. This is a standard SVD matrix decomposition with user and item bias: $\widehat{r}_{uv} = \mu + b_u + b_v + U_u \cdot V_v$, where b_u, b_v indicate the deviations of user u and item i respectively, and U_u, V_v are latent factor vectors.
SVDFeature [5]. This is a famous recommendation toolkit for feature-based CF. The authors of the toolkit have used it to win the KDD Cup for two consecutive years. It can include attributes into CF process and is one of the state-of-the-art recommendation methods for single domain.
STLCF [17]. This is a selective knowledge transfer method and can outperform some cross domain RSs such as CMF. It applies the AdaBoosting framework in order to capture the consistency across domain in CF settings, and it does not consider the rich content information of items.
MV-DNN [8]. This is a content-based multiple domain recommendation. It uses deep learning to match rich user features to item features. However, it does not exploit collaborative filtering.
MCDRS-. This is a variant of our proposed method which removes the neighbor model submodule.

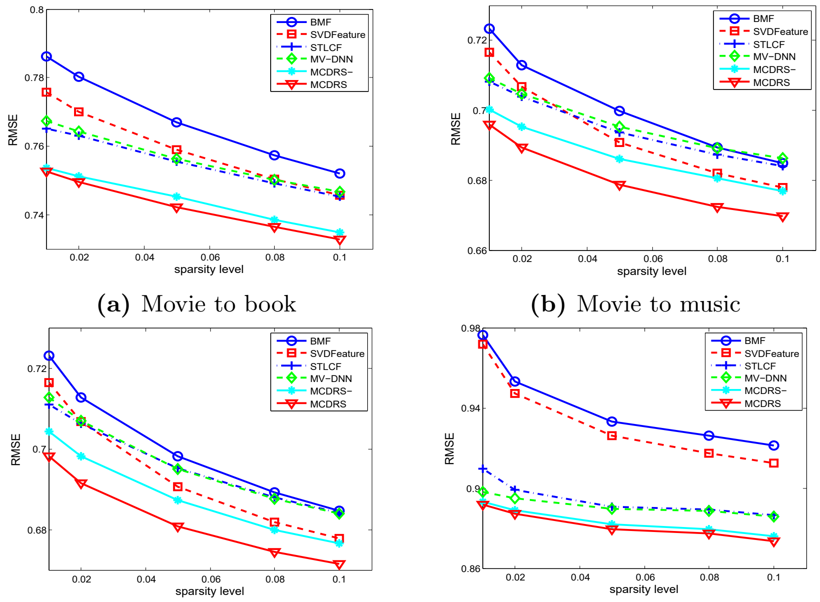
We adopt Root Mean Square Error (RMSE) to evaluate the performance of different methods, where $\mathfrak{R}$ indicates the test set:

$$RMSE = \sqrt{\frac{1}{|\mathfrak{R}|} \sum_{(u,v,r_{uv}) \in \mathfrak{R}} (r_{uv} - \widehat{r}_{uv})^2}$$

We use grid search to find the best parameters for each method. We have tried λ_* from {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0}. To reduce computational cost, for all algorithms we use a same fixed latent feature dimension, i.e. we fix the dimensions of BMF, SVDFeature and STLCF to be 16, and fix the size of the last layer in MV-DNN model to be 16 while changing the sizes of previous hidden layers in {32, 64, 128}. For our proposed model, we set the CF dimension to be 8 and the content-related dimension to be 8, so that the total size of the latent factors is 16. After running these groups of parameters, we pick the best parameter set for each model according to the validation set. We re-run the experiment pipeline five times for each model by fixing the best parameter set, and report the corresponding average $RMSE$ on the test set. For our model, the best setting is $\{\lambda_1 = 0.8, \lambda_3 = \lambda_5 = 0.005, \lambda_4 = \lambda_2 = 0.1\}$.

4.2 Results

Figure 1 shows the RMSE results of different algorithms. Generally speaking, due to consistency problem across domains, a small volume of new data in the target domain matches up to a large volume of data from the source domain. And our MCDRS is designed to consider as much knowledge as possible in the target



(a) Movie to book

(b) Movie to music

(c) Book to music

(d) MovieLens 20M

Fig. 1: RMSE evaluation on Douban dataset (a-c) and MovieLens 20M dataset (d). Here, "movie to book" represents the source domain is movie and target domain is book.

domain, and at the same time transfer knowledge from the source domain. So it is expected to observe the fact that our proposed model consistently outperforms the other models under various tasks.

Figure 1a reports the performance comparison when we use movie domain as the source domain and take book domain as the target domain. The BMF model is the weakest one because it is a single domain CF algorithm, and it does not consider the rich content information. SVDFeature is the single domain model which combines both CF and content information. It's no surprise that SVDFeature significantly outperforms BMF. STLCF and MV-DNN work better than SVDFeature only under the low sparsity levels, and the superiority trend to disappear when the data become denser. MCDRS- significantly outperforms STLCF and MV-DNN due to its utilization of both collaborative filtering and content-based filtering in the cross domain situation. At last, MCDRS shows a further improvement, which demonstrates the necessary of selective learning part.

We observe some difference in RMSE trends when comparing Figure 1b,1c with Figure 1a. Movie and book are two close domains and they share a lot of topics, such as adventure, history fiction, and science fiction. Some movies are even adaptations of famous books. So it will be easier to build cross domain RSs between movie and book domain. However, music domain is relatively not so close with movie or book domain. Thus in Figure 1b and 1c, STLCF and MV-DNN are worse than SVDFeature when the sparsity level is above 5%. MCDRS model works well in the two tasks, which verifies that our designed selective learning module can mitigate the inconsistency problem between domains.

In the MovieLens task, there is no consistency problem since both the two domains are actually derived from the original movie domain. So all the cross domain models can easily outperform the single domain models. Comparing MCDRS- with STLCF and MV-DNN, we again observe that it is quite necessary to incorporate both collaborive filtering and content-based filtering in the cross domain RSs.

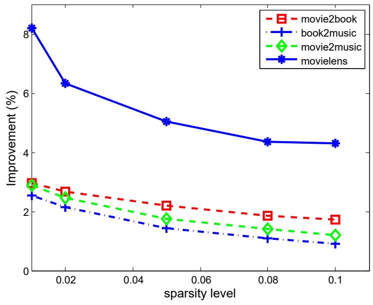


Fig. 2: RMSE improvement of MCDRS against single domain model at various sparsity levels. Here, “movie2book” represents the source domain is movie and the target domain is book.

Figure 2 summarizes the performance gain when comparing MCDRS against the SVDFeature. There is a clear pattern that the cross domain RS works well when data in the target domain is severely sparse, and the degree of improvement decreases when the data of target domain becomes denser. Relatively speaking, the MovieLens task shows the greatest improvement as expected. Results of the task *movie2book* is better than the *movie2music* because that the inconsistency on the movie-book domain is smaller than that on the movie-music domain.

5 Related Works

Single domain recommendation systems. In general, recommendation systems can be divided into content-based model, collaborative filtering (CF) models and hybrid models [4]. Content-based models represent user and item under a same feature space and make recommendation according to their similarity [15]. CF models make predictions about the interests of a user by collecting preferences or taste information form many users. Among CF models the latent factor models are the most widely studied methods [7, 24, 23]. Hybrid models try to integrate content-based method and CF method into one unified intelligent system [2, 18, 16, 28]. In the recent years, the great success of deep learning models on computer vision and natural language processing has motivated researchers to develop deep learning models for recommender systems [25, 31, 29, 27, 1].

Cross domain methods with user/item aligned. [3] studies some earliest CDCF models including the cross domain neighbor model. CMF [26] simultaneously factorizes several matrices while sharing latent factors when an entity participates in multiple relations. CST [21] is an adaptation style tri-factorization algorithm which transfers knowledge from the auxiliary domain to improve performance in the target domain. TCF [20] uses the tri-factorization method to collectively learn target and auxiliary matrices. In their case both users and items are aligned between the two matrices. The major advantage of the TCF approach is that through tri-factorization it is able to capture the data-dependent effect when sharing data independent knowledge. [17] proposes a criterion which can be embeded into a boosting framework to perform selective knowledge transfer. [22] suggests a cross domain CF based on CCA to share information between domains. [8, 14] use multi-view deep learning models to jointly latent features for users and items from multiple domains. Our paper belongs to this category of cross domain RSs. While the existing works focus on transfer knowledge through collaborative filtering or content-based filtering, we focus on incorporating the advantages of both the two models and further explore how to reduce the inconsistency problem through the rich information.

Cross Domain methods with latent clusters. CBT [12] studies the cases in which neither users nor items in the two domains overlap. It assumes that both auxiliary and target domains share the cluster-level rating patterns. So it compresses the auxiliary rating matrix into a compact cluster-level rating pattern representation referred to as a codebook, and reconstructs the target rating matrix by expanding the codebook. RMGM [13] extends CBT to a generative model and they share the same assumption. [6] exploits tensor factorization to model high dimensional data and transfers knowledge from the auxiliary domain through a shared cluster-level tensor. In this paper, we discuss the scenario in which users of the two domains are well aligned, and thus is different from these hidden cluster-linking approaches.

6 Conclusion

In this paper, we introduce a multifaceted model for cross domain RSs. It includes two parts which are delicately designed for cross domain situation. First we propose a latent factor model to incorporate both collaborative filtering and content-based filtering, and different domains are bridged through the aligned latent user features. Second, we propose a selective neighbor model to reduce the inconsistency problem across domains. Experimental results demonstrate that our proposed model consistently outperforms several state-of-the-art methods on both Douban and MovieLens datasets. For future works, we will explore how to exploit deep learning techniques to learn better representation from the rich content information for cross domain RSs.

References

1. Trapit Bansal, David Belanger, and Andrew McCallum. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 107–114. ACM, 2016.
2. Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, page 9. ACM, 2004.
3. Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-domain mediation in collaborative filtering. In *International Conference on User Modeling*, pages 355–359. Springer, 2007.
4. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109 – 132, 2013.
5. Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. Svdfeature: A toolkit for feature-based collaborative filtering. *J. Mach. Learn. Res.*, 13(1):3619–3622, December 2012.
6. 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, New York, 2013. ACM.
7. Michael D Ekstrand, John T Riedl, and Joseph A Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.
8. Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 278–288, Florence, Italy, 2015.
9. Meng Jiang, Peng Cui, Nicholas Jing Yuan, Xing Xie, and Shiqiang Yang. Little is much: Bridging cross-platform behaviors through overlapped crowds. In *AAAI*, pages 13–19. AAAI Press, 2016.
10. Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.
11. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.
12. Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. IJCAI'09, pages 2052–2057, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
13. Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. ICML '09, pages 617–624, New York, NY, USA, 2009. ACM.
14. Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. Cccfnet: A content-boosted collaborative filtering neural network for cross domain recommender systems. WWW '17 Companion, pages 817–818, 2017.
15. Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

16. Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. Content-based collaborative filtering for news topic recommendation. In *AAAI*, pages 217–223. Citeseer, 2015.
17. Zhongqi Lu, Erheng Zhong, Lili Zhao, Evan Wei Xiang, Weike Pan, and Qiang Yang. Selective transfer learning for cross domain recommendation. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 641–649. SIAM, 2013.
18. Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Aaai/iaai*, pages 187–192, 2002.
19. T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
20. Weike Pan, Nathan N. Liu, Evan W. Xiang, and Qiang Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. IJCAI'11, pages 2318–2323. AAAI Press, 2011.
21. Weike Pan, Evan W. Xiang, Nathan N. Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. AAAI'10, pages 230–235. AAAI Press, 2010.
22. Shaghayegh Sahebi and Peter Brusilovsky. It takes two to tango: An exploration of domain pairs for cross-domain collaborative filtering. RecSys '15, pages 131–138, New York, NY, USA, 2015. ACM.
23. Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.
24. Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, volume 20, pages 1–8, 2011.
25. Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.
26. Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. KDD '08, pages 650–658, New York, NY, USA, 2008. ACM.
27. Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, pages 2643–2651, 2013.
28. Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.
29. Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM, 2015.
30. Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.
31. Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 353–362. ACM, 2016.