

The original publication is available at www.springerlink.com

Cross-domain Recommendation with Consistent Knowledge Transfer by Subspace Alignment

Qian Zhang¹, Jie Lu¹, Dianshuang Wu¹, and Guangquan Zhang¹

Decision Systems and e-Service Intelligence Laboratory, Center for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

{qian.zhang-1, jie.lu, dianshuang.wu, guangquan.zhang}@uts.edu.au

Abstract. Recommender systems have drawn great attention from both academic area and practical websites. One challenging and common problem in many recommendation methods is data sparsity, due to the limited number of observed user interaction with the products/services. Cross-domain recommender systems are developed to tackle this problem through transferring knowledge from a source domain with relatively abundant data to the target domain with scarce data. Existing cross-domain recommendation methods assume that similar user groups have similar tastes on similar item groups but ignore the divergence between the source and target domains, resulting in decrease in accuracy. In this paper, we propose a cross-domain recommendation method transferring consistent group-level knowledge through aligning the source subspace with the target one. Through subspace alignment, the discrepancy caused by the domain-shift is reduced and the knowledge shared local top-n recommendation via refined item-user bi-clustering two domains is ensured to be consistent. Experiments are conducted on five real-world datasets in three categories: movies, books and music. The results for nine cross-domain recommendation tasks show that our proposed method has improved the accuracy compared with five benchmarks.

Keywords: Recommender systems · Cross-domain recommender systems · Knowledge transfer · Collaborative filtering.

1 Introduction

Recommender systems have been in existence for more than twenty years with wide application [15]. They are now an indispensable part of Internet websites such as Amazon.com, YouTube, Netflix, Yahoo, Facebook, Last.fm and Meetup. With great success and promising future, recommender systems are developed to provide users with more accurate and various options. User demands of diverse recommendation have prompted recommender systems to expand from single-domain to multi-domain [1]. The mining of correlation between several domains can benefit every single domain, meanwhile possibly discovering user preferences that cannot be found with single domain data. Further, exploiting several domains together provides a way to solve the data sparsity problem,

which is a common and challenging issue in lots of existing recommendation methods. For example, one user may have few records in a book category in an online review and rating system, but a lot of movie ratings available. The abundance of data in another domain can assist the recommendation in a specific target domain. All of these have brought about the increasing research on cross-domain recommender systems.

Cross-domain recommender systems are developed to solve the data sparsity problem taking advantage of data in multiple domains. These systems aim to extract knowledge from domains that contain relatively rich data and adapt it to the target domain where data is insufficient. Two different types of cross-domain recommender systems have been developed. Some methods connect multiple domains through auxiliary information rather than preference data [17]. It is assumed that some side information on users/items is available, either user generated information [7], social information [9] or item attributes [19]. On the other hand, some methods focus on preference data which are the most commonly collected data on e-commerce or online rating websites. This type of systems is designed in various ways according to the overlap of users and items [16] or the form of the data [8]. Due to the privacy issue, the user IDs are usually de-identified and items from two domains are not always the same. Some previous research tried to find the linkage through user display name [13] or user spatial behavior [2]. In this paper, we focus on cross-domain recommender systems where users and items have no intersections. In this situation, the basic assumption of cross-domain recommendation similar to the basic assumption of collaborative filtering [19]: A group of users tend to rate a group items similarly as another group of users implies similar preference to another group of items implies similar attributes. The knowledge shared by these two domains is group-level rating pattern.

Though cross-domain recommender systems have attracted lots of attention and efforts from academia, they still suffer the “negative transfer” problem [14]. The main reason is that data collected from two correlated domains are probably from two related but different distributions. Thus, domain shift is an issue that must be taken into account and carefully handled in cross-domain recommender systems. Without adaptation to domain shift, cross-domain recommender systems are likely to fail to provide useful and accurate recommendation in the target domain where data are not sufficient [3]. Most existing methods on cross-domain recommendation ignore the domain shifts and fail to extract consistent knowledge shared by two or multiple domains. For example, codebook transfer (CBT) clusters users and items into groups and extracts group-level knowledge as a “codebook” [11]. Later, a probabilistic model named rating matrix generative model (RMGM) is extended from CBT, relaxing the hard group membership to soft membership [12]. These two methods cannot ensure that knowledge extracted from two different domains is consistent, and the effectiveness of knowledge transfer is not guaranteed.

In this paper, we investigate how to eliminate the domain shift and improve the accuracy of cross-domain recommender systems. Domain adaptation is a

transfer learning technique that can deal with the shift between distributions of data from the source domain and the target domain. Subspace alignment is one of the promising domain adaptation approaches to reduce the discrepancy between two domains by moving source subspace basis closer to the target subspace, especially in unsupervised setting as we conduct group-level knowledge transfer where no label is available. Before shared group-level knowledge is extracted from two domains, the source subspace and the target subspace are aligned so that the consistency of knowledge extracted is ensured. Thus, we propose a cross-domain recommendation method with consistent knowledge transfer by subspace alignment (CKTSA). The main contributions of this paper are as follows:

1. A cross-domain recommendation method CKTSA is developed where subspace alignment is used to match the source subspaces and the target subspaces to ensure the consistency of knowledge extracted.
2. The proposed method CKTSA is evaluated on five real-world datasets with nine cross-domain recommendation tasks compared with five other non-transfer or cross-domain recommendation methods. The results show that our proposed method has superior advantages in providing accurate recommendation in sparse data especially when source domain data has divergence with target domain data.

The rest of the paper is organized as follows. Section 2 gives some preliminary and a formal definition of the problem. Section 3 describes our method using subspace alignment to ensure the consistent knowledge transfer in cross-domain recommendation. In Section 4, we present our experiments on five real-world datasets spanning three categories of data. Finally, in Section 5, conclusion is provided with some future directions of this research.

2 Preliminary and Problem Formation

In this section, before presenting our proposed method, recommendation by tri-factorization is briefly introduced. The problem targeted in this paper is also formally formulated.

2.1 Recommendation by tri-factorization

Matrix factorization projects both users and items onto the same latent space so that they are comparable, and through their inner products reconstructs the rating matrix [10]. Similarly, the rating matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ (bold letters represent matrixes) can be factorized into three matrixes (suppose there are M users and N items). Users and items are mapped to different latent spaces and the two spaces are mapped together through a mediate interaction matrix, as in [5]: $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{M \times K}$ is user feature matrix, representing users clustered into K groups, $\mathbf{V} \in \mathbb{R}^{N \times L}$ is item feature matrix, representing items clustered into L groups and $\mathbf{S} \in \mathbb{R}^{K \times L}$ is the group preference matrix, i.e. the

group-level knowledge. Thus, $\Theta = \{\mathbf{U}, \mathbf{S}, \mathbf{V}\}$ are the parameters used to predict the ratings and provide a recommendation.

To calculate the missing values, the user-item rating matrix is reconstructed through $\hat{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. Tri-factorization of \mathbf{X} minimizes the loss function $\mathcal{L}(\mathbf{X}, \mathbf{U}\mathbf{S}\mathbf{V}^T)$, which measures the error of prediction. Since \mathbf{X} is usually sparse, the loss function is in a weighted form as follows:

$$\mathcal{L}(\mathbf{X}, \mathbf{U}\mathbf{S}\mathbf{V}^T) = \|\mathbf{I} \circ (\mathbf{X} - \mathbf{U}\mathbf{S}\mathbf{V}^T)\|_F \quad (1)$$

where \mathbf{I} is the rating indicator matrix the same size of \mathbf{X} representing whether the rating in \mathbf{X} is observed or not, $I_{ij} \in \{0, 1\}$. $I_{ij} = 1$ indicates that the rating is observed and $I_{ij} = 0$ otherwise and \circ denotes the Hadamard product of matrixes. The tri-factorization is:

$$\begin{aligned} & \min_{\mathbf{U}, \mathbf{S}, \mathbf{V}} \mathcal{L}(\mathbf{X}, \mathbf{U}\mathbf{S}\mathbf{V}^T) \\ & \text{s. t. } \mathbf{U} > 0, \mathbf{S} > 0, \mathbf{V} > 0 \end{aligned}$$

2.2 Problem formulation

In this problem setting, the users/items have no correspondence across the domains and are treated as completely different users/items. We assume that explicit rating data are available for both the source and target domains. Formally, the problem is defined as:

Given a source rating matrix $\mathbf{X}_s \in \mathbb{R}^{M_s \times N_s}$ and a target rating matrix $\mathbf{X}_t \in \mathbb{R}^{M_t \times N_t}$, our goal is to develop a recommendation method aiming to help recommendation tasks in the target domain predict the rating $\hat{\mathbf{X}}_t = \mathbf{U}_t \mathbf{S}_t \mathbf{V}_t^T$ using knowledge in the source rating matrix \mathbf{X}_s and $\Theta_s = \{\mathbf{U}_s, \mathbf{S}_s, \mathbf{V}_s\}$, where $\mathcal{U}_s \cap \mathcal{U}_t = \emptyset$ and $\mathcal{I}_s \cap \mathcal{I}_t = \emptyset$. \mathcal{U}_s and \mathcal{I}_s represent the user set and item set in the source domain, while \mathcal{U}_t and \mathcal{I}_t represent the user set and item set in the target domain.

3 Cross-domain Recommendation with Consistent Knowledge Transfer

In this section, our proposed CKTSA method is presented beginning with an overview of the method procedure containing five steps. Each of the five steps is then explained in detail.

3.1 The Method Overview

The proposed method CKTSA uses subspace alignment to ensure the knowledge extracted from the source domain is consistent with that in the target domain. The procedure consists five steps, as show in Fig. 1. 1) Users and items are clustered separately and user feature matrixes and item feature matrixes are

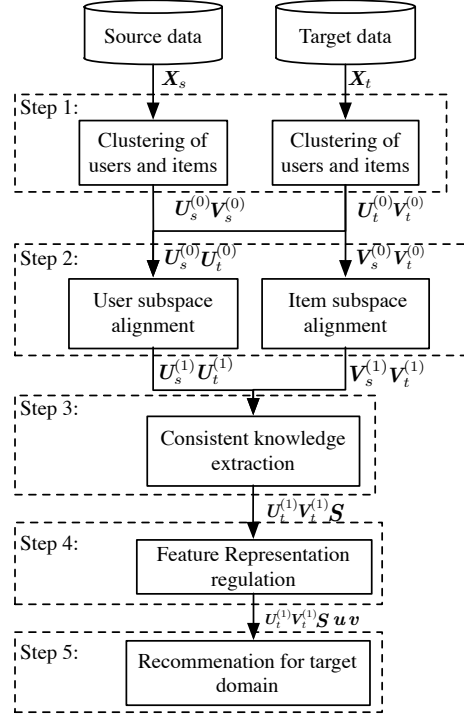


Fig. 1. The CKTSA method procedure.

obtained; 2) Subspace alignment is conducted to move the source subspace closer to the target domain; 3) Consistent knowledge is extracted since source subspaces is aligned to the target subspace and domain discrepancy is eliminated; 4) Feature representation is regulated in the target domain to retain domain specific characteristics; 5) Recommendation in the target domain.

3.2 The CKTSA method

Our proposed method consists five steps.

Step 1: Clustering Users and Items in Both Domains We choose the Flexible Mixture Model (FMM) to cluster the users and items separately [18], since this method allows both users and items to fall into multiple groups with different memberships. This fits to the situation that users may have various preferences and items may have diverse content. The same clustering procedure is used for both the source domain and the target domain; however, for simplicity, we have only provided the description for one domain.

Suppose users are clustered into K user groups $\{Z_u^{(1)}, \dots, Z_u^{(K)}\}$, while items are clustered into L item groups $\{Z_v^{(1)}, \dots, Z_v^{(L)}\}$. Z_u and Z_v are two latent variables that denote the user and item groups respectively. $P(Z_u|u)$ is the conditional probability of a user belonging to a user group, denoting the

group membership of the user; $P(Z_v|v)$ is the conditional probability of an item belonging to an item group, denoting its group membership. Each user group has a rating preference for each item group. r is the variable representing the preference of user groups to item groups. $P(r|Z_u, Z_v)$ is the conditional probability of r given user group Z_u and item group Z_v . The rating for a coupled user-item pair is:

$$R(u, v) = \sum_r r \sum_{Z_u, Z_v} P(r|Z_u, Z_v)P(Z_u|u)P(Z_v|v) \quad (2)$$

Equation (2) can be rewritten into matrix form:

$$\mathbf{X} = \mathbf{USV}^T \quad (3)$$

where $\mathbf{U} \in \mathbb{R}^{M \times L}$ and $\mathbf{V} \in \mathbb{R}^{N \times L}$ are the user and item feature matrix. U_{ij} represents the membership of user u_i for user group $Z_u^{(j)}$. U_{i*} is the i th row of matrix \mathbf{U} representing membership of user u_i to each group. U_{*j} is the j th column of matrix \mathbf{U} representing the membership of each user to user group $Z_u^{(j)}$. The same goes for items. $\mathbf{S} \in \mathbb{R}^{K \times L}$ is the group-level knowledge matrix. S_{ij} represents the preference of user group $Z_u^{(i)}$ for item group $Z_v^{(j)}$.

After clustering, the user group and item group membership matrixes $\mathbf{U}_s^{(0)}$, $\mathbf{V}_s^{(0)}$ are acquired for the source domain and $\mathbf{U}_t^{(0)}$, $\mathbf{V}_t^{(0)}$ for the target domain.

$$\mathbf{U}_s^{(0)} = P(Z_{u_s}|u_s), \mathbf{V}_s^{(0)} = P(Z_{v_s}|v_s) \quad (4)$$

$$\mathbf{U}_t^{(0)} = P(Z_{u_t}|u_t), \mathbf{V}_t^{(0)} = P(Z_{v_t}|v_t) \quad (5)$$

where $P(Z_u|u) = \frac{P(u|Z_u)P(Z_u)}{\sum_{Z_u} P(u|Z_u)P(Z_u)}$ and $P(Z_v|v) = \frac{P(v|Z_v)P(Z_v)}{\sum_{Z_v} P(v|Z_v)P(Z_v)}$. Five parameters $P(u|Z_u)$, $P(v|Z_v)$, $P(r|Z_u, Z_v)$, $P(Z_u)$ and $P(Z_v)$ are learned from the FMM. An expectation maximization (EM) algorithm is used to learn FMM. Suppose there are H ratings in user-item rating matrix represented as $\{(u_1, v_1, r_1), (u_2, v_2, r_2), \dots, (u_H, v_H, r_H)\}$. In the E-step, the parameters are fixed to optimize joint posterior probability. For any user-item-rating triplet (u_h, v_h, r_h) :

$$P(Z_u, Z_v|u_h, v_h, r_h) = \frac{P(r_h|Z_u, Z_v)P(u_h|Z_u)P(v_h|Z_v)P(Z_u)P(Z_v)}{\sum_{Z_u, Z_v} P(r_h|Z_u, Z_v)P(u_h|Z_u)P(v_h|Z_v)P(Z_u)P(Z_v)} \quad (6)$$

Then, in the M-step, the parameters are updated as follows:

$$P(Z_u) = \frac{\sum_h \sum_{Z_v} P(Z_u, Z_v|u_h, v_h, r_h)}{H} \quad (7)$$

$$P(Z_v) = \frac{\sum_h \sum_{Z_u} P(Z_u, Z_v|u_h, v_h, r_h)}{H} \quad (8)$$

$$P(u|Z_u) = \frac{\sum_{u_h=u} \sum_{Z_v} P(Z_u, Z_v|u_h, v_h, r_h)}{H \cdot P(Z_u)} \quad (9)$$

$$P(v|Z_v) = \frac{\sum_{v_h=v} \sum_{Z_u} P(Z_u, Z_v|u_h, v_h, r_h)}{H \cdot P(Z_v)} \quad (10)$$

$$P(r|Z_u, Z_v) = \frac{\sum_{r=r_h} P(Z_u, Z_v|u_h, v_h, r_h)}{\sum_h P(Z_u, Z_v|u_h, v_h, r_h)} \quad (11)$$

By alternatively computing the E-step and the M-step, users and items are clustered into latent groups according to the user-item rating matrix. (For details, see [18]).

Step 2: Subspace Alignment of User and Items After users and items are clustered in both source and target domains, user feature matrixes $\mathbf{U}_s^{(0)}$ and $\mathbf{U}_t^{(0)}$ and item feature matrixes $\mathbf{V}_s^{(0)}$ and $\mathbf{V}_t^{(0)}$ are obtained. It is possible to cluster users in source and target domains both in K groups to make users both lie in K -dimensional space, but in fact the user feature matrixes are from different marginal distributions. The divergence in data distributions need to be eliminated before conducting knowledge extraction. We use subspace alignment in this paper to learn new representations of users and items so that they are in the same subspace coordinate system.

To better handle the characteristics of data distributions, we firstly use the function of Z-score to normalize the original representations of users in both domains. Then, we use principal component analysis (PCA) to extract d eigenvectors corresponding to the largest d eigenvalues which are treated as the basis of source and target spaces, denoted as \mathbf{D}_s^u and \mathbf{D}_t^u for user subspaces and \mathbf{D}_s^v and \mathbf{D}_t^v for item subspaces. For subspace alignment, we align the subspace basis according to [6]. User and item transition matrixes \mathbf{T}^u and \mathbf{T}^v are learned through optimizing:

$$\mathcal{L}(\mathbf{T}^u) = \|\mathbf{D}_s^u \mathbf{T}^u - \mathbf{D}_t^u\|_F \quad (12)$$

$$\min_{\mathbf{T}^u} \mathcal{L}(\mathbf{T}^u)$$

$$\mathcal{L}(\mathbf{T}^v) = \|\mathbf{D}_s^v \mathbf{T}^v - \mathbf{D}_t^v\|_F \quad (13)$$

$$\min_{\mathbf{T}^v} \mathcal{L}(\mathbf{T}^v)$$

Closed forms of optimal \mathbf{T}^u and \mathbf{T}^v are as follows (details refer to [6]):

$$\mathbf{T}^u = (\mathbf{D}_s^u)^T \mathbf{D}_t^u \quad (14)$$

$$\mathbf{T}^v = (\mathbf{D}_s^v)^T \mathbf{D}_t^v \quad (15)$$

Once the transition matrixes are obtained, the user and item subspaces can be aligned to the same one. How the subspace alignment is done is summarized in Algorithm 1. There are two advantages using the subspace alignment: 1) Compared with other domain adaptation methods that directly project source and target domain data to a shared common subspace, subspace alignment is not only limited to the shared features but able to exploit more correlations between two domains; 2) Compared with other domain adaptation methods that model domain shift and learn new representations through a large number of subspaces, subspace alignment uses a linear transition function that is simple and powerful. These two advantages fit well in our cross-domain recommendation problem setting. The first advantage contributes to the potential diverse demand of users while the second advantage meets users' requirement on recommender systems to respond quickly and provide timely support.

Algorithm 1 User and Item Subspace Alignment**Input:** $\mathbf{U}_s^{(0)}, \mathbf{U}_t^{(0)}$, the source and target user feature matrix $\mathbf{V}_s^{(0)}, \mathbf{V}_t^{(0)}$, the source and target item feature matrix**Output:** $\mathbf{U}_s^{(1)}, \mathbf{U}_t^{(1)}$, the aligned source and target user feature matrix $\mathbf{V}_s^{(1)}, \mathbf{V}_t^{(1)}$, the aligned source and target item feature matrix1: $\mathbf{D}_s^u \leftarrow PCA(f_{zs}(\mathbf{U}_s^{(0)})), \mathbf{D}_t^u \leftarrow PCA(f_{zs}(\mathbf{U}_t^{(0)}))$ $\mathbf{D}_s^v \leftarrow PCA(f_{zs}(\mathbf{V}_s^{(0)})), \mathbf{D}_t^v \leftarrow PCA(f_{zs}(\mathbf{V}_t^{(0)}))$ 2: $\mathbf{T}^u \leftarrow (\mathbf{D}_s^u)^T \mathbf{D}_t^u, \mathbf{T}^v \leftarrow (\mathbf{D}_s^v)^T \mathbf{D}_t^v$ 3: $\mathbf{U}_s^{(1)} \leftarrow \mathbf{U}_s^{(0)} \mathbf{D}_s^u \mathbf{T}^u, \mathbf{U}_t^{(1)} \leftarrow \mathbf{U}_t^{(0)} \mathbf{D}_t^u$ $\mathbf{V}_s^{(1)} \leftarrow \mathbf{V}_s^{(0)} \mathbf{D}_s^v \mathbf{T}^v, \mathbf{V}_t^{(1)} \leftarrow \mathbf{V}_t^{(0)} \mathbf{D}_t^v$ 4: Normalize $\mathbf{U}_s^{(1)}, \mathbf{U}_t^{(1)}, \mathbf{V}_s^{(1)}$ and $\mathbf{V}_t^{(1)}$ to $[0, 1]$ 5: **return** $\mathbf{U}_s^{(1)}, \mathbf{U}_t^{(1)}, \mathbf{V}_s^{(1)}, \mathbf{V}_t^{(1)}$

Step 3: Consistent Knowledge Extraction After the domain adaptation, $\mathbf{U}_s^{(1)}, \mathbf{U}_t^{(1)}$ and $\mathbf{V}_s^{(1)}, \mathbf{V}_t^{(1)}$ aligned to the same subspaces. Once the source subspace is aligned to the target subspace, the recommender systems learned from the source and target domains will share the same group-level knowledge matrix \mathbf{S} .

Consistent knowledge \mathbf{S} is obtained by maximizing the approximation of the available data in both the source rating matrix and the target rating matrix. To qualify the approximation, Frobenius norm is used as a measure between the original rating matrix and the approximation. We list the cost function here and more details can be found in [20]:

$$J_s(\mathbf{S}) = \frac{1}{M_s N_s} \|\mathbf{I}_s \circ (\mathbf{X}_s - \mathbf{U}_s^{(1)} \mathbf{S} (\mathbf{V}_s^{(1)})^T)\|_F + \frac{1}{M_t N_t} \|\mathbf{I}_t \circ (\mathbf{X}_t - \mathbf{U}_t^{(1)} \mathbf{S} (\mathbf{V}_t^{(1)})^T)\|_F + \frac{\lambda}{2KL} \|\mathbf{S}\|_F \quad (16)$$

where \mathbf{I}_s is an indicator matrix for \mathbf{X}_s , if $(\mathbf{I}_s)_{ij} = 1$, then $(\mathbf{X}_s)_{ij} \neq 0$ and $(\mathbf{I}_s)_{ij} = 0$, otherwise. The same applies to \mathbf{I}_t for \mathbf{X}_t . \circ is an entry-wise product, λ is the parameter for regularization. Finally, consistent knowledge is learned through the following optimization problem:

$$\begin{aligned} \min_{\mathbf{S}} J_s(\mathbf{S}) \\ \text{s.t. } \mathbf{S} > 0 \end{aligned}$$

Step 4: Feature Representation Regulation In our problem setting, some domain-specific characteristics are embedded in the small amount of available data in the target rating matrix. To reveal these idiosyncrasies of the target domain, we amend feature representations of the target rating matrix to make the model fit better to the task in target rating matrix. The representation regulation is achieved through an optimization problem. The cost function is:

$$J_r(u, v) = \|\mathbf{I}_t \circ (\mathbf{X}_t - \mathbf{U}_t^{(1)} \mathbf{u} \mathbf{S} (\mathbf{V}_t^{(1)} \mathbf{v})^T)\|_F \quad (17)$$

The tuning factors can be learned through optimizing

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}} J_r(\mathbf{u}, \mathbf{v}) \\ \text{s.t. } u \geq 0, v \geq 0 \end{aligned}$$

The optimization problem is solved by alternatively estimating tuning factors \mathbf{u} and \mathbf{v} . For more details, see [20].

Step 5: Recommendation in Target Domain The recommendation in target domain is given by Equation (18).

$$\hat{\mathbf{X}}_t = (\mathbf{U}_t^{(1)} \mathbf{u}) \mathbf{S} (\mathbf{V}_t^{(1)} \mathbf{v})^T \quad (18)$$

where $\hat{\mathbf{X}}_t$ is the reconstructed user-item rating matrix for prediction, \mathbf{u} , \mathbf{v} are user and item tuning factors for target domain, \mathbf{S} is the consistent knowledge, $\mathbf{U}_s^{(1)}$, $\mathbf{U}_t^{(1)}$ are user and item feature matrixes for the target domain after subspace alignment obtained from Algorithm 1.

4 Experiments

In this section, the proposed method CKTSA is evaluated. First, the datasets and evaluation metrics used are introduced in Section 4.1, followed by experimental settings and the baseline methods 4.2. The results of the experiments are presented in Section 4.3.

4.1 Datasets and Evaluation Metrics

To test our proposed method, we need to choose data from similar data so that transfer learning is meaningful, but divergence still exists between the source and target domains. Similar to [20], we choose movie, book and music as three categories for experiments. Our experiments comprise nine cross-domain recommendation tasks with all the combinations of the three categories. Five real-world datasets were used: EachMovie¹, Movielens1M², LibraryThing³, Amazon Book⁴ and YahooMusic⁵. Each is publicly available and has been used to test recommender systems in a variety of scenarios for recommender systems in single domain. But tests on these dataset in this novel cross-domain setting are lacking. For AmazonBooks, we removed all users who had given exactly the same rating for every book, as these data are not effective for constructing a recommender system [20]. EachMovie and LibraryThing was normalized to the range of $\{1, 2, 3, 4, 5\}$ before conducting experiments. The statistical information for original datasets is provided in Table 1. Across all the datasets, 1000 items

¹ <http://www.cs.cmu.edu/~lebanon/IR-lab/data.html#intro>

² <http://grouplens.org/datasets/movielens/1m/>

³ <https://www.librarything.com>

⁴ <http://jmcauley.ucsd.edu/data/amazon/>

⁵ <https://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

Table 1. Statistical information on the original datasets

	EachMovie	Movielens 1M	library Thing	Amazon Book	Yahoo music_1	Yahoo music_2
#user	72916	6040	7279	8026324	200000	200000
#item	1628	3900	37232	2330066	136736	136736
#rating	2811983	1000209	749401	22507155	78344627	78742463
sparsity	97.63%	95.75%	99.72%	99.99%	99.71%	99.71%
range	0-1	1-5	0.5-5	1-5	1-5	1-5

Table 2. Description of data subsets in three categories

Data type	Data source	Domain	Sparsity	Average
Movie	EachMovie	source	96.00%	4.32
	Movielens1M	target	98.50%	2.91
book	LibraryThing	source	87.43%	3.97
	AmazonBook	target	97.87%	3.13
music	YahooMusic_1	source	95.70%	4.14
	YahooMusic_2	target	97.27%	2.66

that had been rated more than 10 times were randomly chosen. We then filtered out the users who had given less than a total of 20 ratings. For the source domain data, we randomly selected 500 users to be regular customers of the site. The source domain data were controlled to be more dense than the target domain data. For the target domain data, we randomly selected 200 users to be regular customers of the site, and another 300 users to be new customers. For new users, five observed ratings were given, and the rest of the ratings were used for evaluation. In the end, the rating matrixes for both the source and target domains were all 500×1000 matrixes. The details of the final datasets are summarized in Table 2. Mean absolute error (MAE) and root mean square error (RMSE) were used as the evaluation metrics:

$$MAE = \sum_{u,v,X_{uv} \in Y} \frac{|\hat{X}_{uv} - X_{uv}|}{|Y|}$$

$$RMSE = \sqrt{\sum_{u,v,X_{uv} \in Y} \frac{(\hat{X}_{uv} - X_{uv})^2}{|Y|}}$$

where Y is the test set, and $|Y|$ is the number of test ratings.

4.2 Experimental Settings and Baselines

The rating average is a very important statistics of the data which we used in our experiments to represent whether data in two domains are of high similarity or not. According to Table 2, we can see a big divergence in the rating average between the source domain data and the target domain data. This fits to our problem setting in Section 2 that data in the source domain and the target domain are similar but divergence still exists.

Three non-transfer learning methods and two cross-domain methods were chosen as comparisons for the proposed method. The non-transfer learning methods were: Pearson’s correlation coefficient (PCC) [4], FMM [18] and single value decomposition (SVD) [10]. The cross-domain methods were: CBT [11] and RMGM [12]. These two cross-domain recommendation methods are all developed without fully considering the domain-shift widely existed in data of two domains. PCC uses user-based CF, and the number of neighborhoods was set at 50. For SVD, the latent feature number was fixed at 10, the regularization factor was set to 0.015, and the learning rate was set to 0.003. For FMM, CBT and RMGM, the user group number and item group number were both set to 10. For the proposed method, CKTSA, the user feature number and the item feature number were both set to 10, and the regularization factor was set to 0.5. Further analysis of the parameters is provided in Section 4.3. All the methods (except for PCC) need to initialize the factorized matrix randomly, we ran 20 random initializations and report the averaged results and standard deviations.

4.3 Results

The experiment results of our proposed CKTSA compared with the other five baselines on two accuracy metrics are presented in Table 3, 4 and 5. Overall, CKTSA has the best performance in all the nine tasks relates to three categories: movie, book and music. In sparse data settings, CKTSA can significantly increase the recommendation accuracy compared with non-transfer learning recommendation methods like PCC, FMM and SVD. These results suggest that CKTSA is an effective method to transfer knowledge that can assist recommendation in the target domain.

Also, CKTSA has better performance than other two cross-domain recommendation methods CBT and RMGM. Compared with other non-transfer learning methods, CBT fails to transfer knowledge to the target domain in most cases. The transfer learning in CBT is even negative in many of the experiments. This is because the core algorithm in CBT is very simple and contains direct group-level knowledge transfer without adjustment or adaptation to the divergence between two domains. On the other hand, RMGM achieves at least positive transfer in most of the tasks compared with CBT. But RMGM still cannot meet the requirement of cross-domain recommendation to extract consistent knowledge from a source domain which is similar but slightly different to the target domain. The comparison between CKTSA and these two cross-domain recommendation methods implies that CKTSA can transfer consistent

Table 3. Prediction performance on a movie target domain

Methods		Source data	MAE	RMSE
Non-trans	PCC	-	1.2123	1.5722
	FMM	-	1.1104±0.0118	1.3540±0.0143
	SVD	-	1.0949±0.0049	1.3540±0.0074
Cross-domain	CBT	movie	1.4587±0.0176	1.7883±0.0161
		book	1.2275±0.0077	1.5551±0.0152
		music	1.2892±0.0239	1.6383±0.0217
	RMGM	movie	1.1896±0.0170	1.4810±0.0225
		book	1.1357±0.0231	1.4102±0.0335
		music	1.1337±0.0191	1.4042±0.0283
	CKTSA	movie	1.0273 ±0.0066	1.2379 ±0.0071
		book	1.0293 ±0.0062	1.2392 ±0.0062
		music	1.0275 ±0.0060	1.2385 ±0.0078

Table 4. Prediction performance on a book target domain

Methods		Source data	MAE	RMSE
Non-trans	PCC	-	1.1802	1.4907
	FMM	-	1.0260±0.0118	1.2670±0.0146
	SVD	-	1.2264±0.0133	1.5283±0.0166
Cross-domain	CBT	movie	1.2762±0.0091	1.5918±0.0085
		book	1.0951±0.0040	1.3997±0.0107
		music	1.1337±0.0115	1.4588±0.0113
	RMGM	movie	1.0072±0.0104	1.2418±0.0155
		book	1.0152±0.0150	1.2492±0.0209
		music	1.0073±0.0117	1.2429±0.0159
	CKTSA	movie	0.9772 ±0.0047	1.1814 ±0.0064
		book	0.9762 ±0.0037	1.1804 ±0.0043
		music	0.9810 ±0.0066	1.1852 ±0.0077

knowledge from the source domain to the target domain. The advantage is more obvious when it comes to the situation that divergence exists between two domains.

Table 5. Prediction performance on a music target domain

Methods		Source data	MAE	RMSE
Non-trans	PCC	-	1.4843	1.8539
	FMM	-	1.2959±0.0211	1.5492±0.0172
	SVD	-	1.4797±0.0208	1.7229±0.0242
Cross-domain	CBT	movie	1.8215±0.0111	2.1673±0.0119
		book	1.6315±0.0092	1.9342±0.0153
		music	1.6932±0.0125	2.0236±0.0147
	RMGM	movie	1.3329±0.0162	1.6057±0.0180
		book	1.3306±0.0190	1.6014±0.0229
		music	1.3306±0.0190	1.6014±0.0229
	CKTSA	movie	1.2915 ±0.0101	1.5022 ±0.0131
		book	1.2922 ±0.0149	1.5046 ±0.0149
		music	1.2911 ±0.0107	1.5004 ±0.0119

4.4 Parameter Analysis

We analyzed how the parameter K , L and λ affect the performance of CKTSA. K is the number of user groups and L is the number of item groups, while λ is the trade-off parameter for the consistent knowledge extraction. Due to the space limitation, only the result of movie-to-movie cross-domain recommendation task is presented. Performance on both MAE and RSME is presented.

To analyze λ , K and L are fixed to 10. In Fig. 2, both MAE and RSME are not significantly affected by λ . To analyze K and L , λ is set to be 0.5. As for the K and L , we use grid search to analyze how these two parameters can interact with each other and how they can affect the performance of CKTSA in Fig. 3. In the range of 5 to 70, the higher the number, the better the performance. Since the higher K and L , the complexity of this method will significantly increase. Thus, in our experiments, we choose K and L for 10 for convenience. And we choose the same value for K and L in every tri-factorization based methods.

5 Conclusion

In this paper, we investigate the cross-domain recommendation problem, specially for two domains where domain-shifts happen. A cross-domain recommendation method called CKTSA is developed that can ensure consistent knowledge transferred from the source domain to the target domain. In the proposed method, we firstly obtained the subspace eigenvectors of both users and items in the source domain and the target domain. Through subspace alignment, the two coordinate systems are aligned to the same one. Since

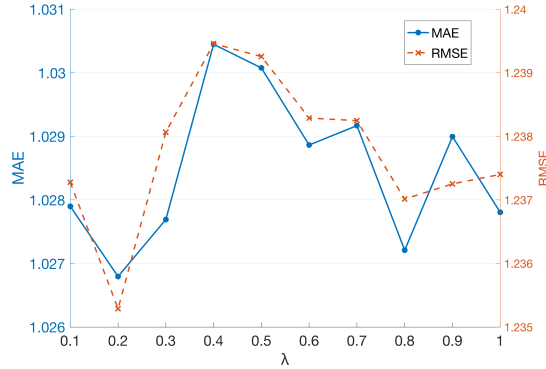


Fig. 2. Results with different settings on parameter λ .

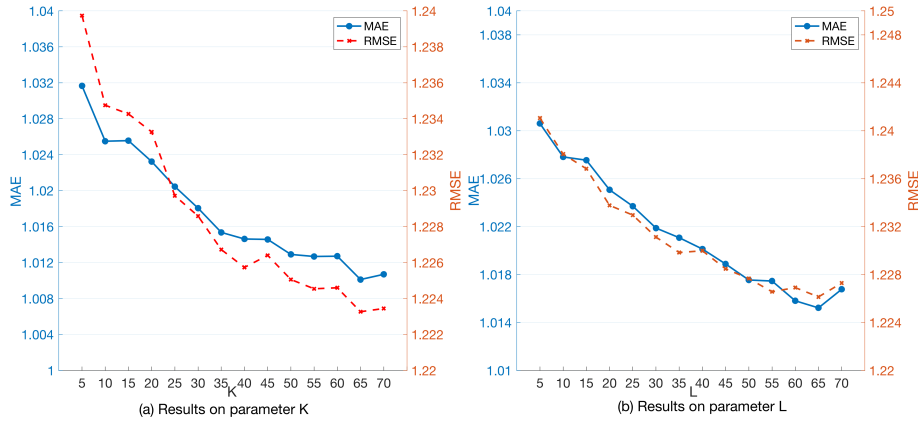


Fig. 3. Results with different settings on parameters K and L .

the subspaces are aligned, consistent knowledge can be extracted from two domains which can help to improve the accuracy of recommendation in the target domain. Extensive experiments on five real-world datasets with nine cross-domain recommendation tasks show that the proposed CKTSA achieves the best performance compared with five baselines including both non-transfer learning and cross-domain recommendation methods. In the future, we will try to develop methods that can deal with heterogeneous data in this problem setting. In this way, implicit feedback and other user behaviors can all be involved to profile the user in multiple domains.

References

1. Cantador, I., Fernández-Tobías, I., Berkovsky, S., Cremonesi, P.: Cross-domain recommender systems. In: Recommender Systems Handbook, pp. 919–959. Springer (2015)

2. Chen, W., Yin, H., Wang, W., Zhao, L., Hua, W., Zhou, X.: Exploiting spatio-temporal user behaviors for user linkage. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 517–526. ACM (2017)
3. Cremonesi, P., Quadrona, M.: Cross-domain recommendations without overlapping data: myth or reality? In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 297–300. ACM (2014)
4. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems* **22**(1), 143–177 (2004)
5. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 126–135. ACM (2006)
6. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: 2013 IEEE International Conference on Computer Vision (ICCV). pp. 2960–2967. IEEE (2013)
7. Hao, P., Zhang, G., Martinez, L., Lu, J.: Regularizing knowledge transfer in recommendation with tag-inferred correlation. *IEEE Transactions on Cybernetics* (2017)
8. Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Zhu, C.: Personalized recommendation via cross-domain triadic factorization. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 595–606. ACM (2013)
9. Jiang, M., Cui, P., Chen, X., Wang, F., Zhu, W., Yang, S.: Social recommendation with cross-domain transferable knowledge. *IEEE Transactions on Knowledge and Data Engineering* **27**(11), 3084–3097 (2015)
10. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8) (2009)
11. Li, B., Yang, Q., Xue, X.: Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In: *IJCAI*. vol. 9, pp. 2052–2057 (2009)
12. Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 617–624. ACM (2009)
13. Li, Y., Peng, Y., Zhang, Z., Wu, M., Xu, Q., Yin, H.: A deep dive into user display names across social networks. *Information Sciences* **447**, 186–204 (2018)
14. Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., Zhang, G.: Transfer learning using computational intelligence: a survey. *Knowledge-Based Systems* **80**, 14–23 (2015)
15. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. *Decision Support Systems* **74**, 12–32 (2015)
16. Mirbakhsh, N., Ling, C.X.: Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data* **9**(4), 33 (2015)
17. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys* **47**(1), 3 (2014)
18. Si, L., Jin, R.: Flexible mixture model for collaborative filtering. In: Proceedings of the 20th International Conference on Machine Learning. pp. 704–711 (2003)
19. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 650–658. ACM (2008)
20. Zhang, Q., Wu, D., Lu, J., Liu, F., Zhang, G.: A cross-domain recommender system with consistent information transfer. *Decision Support Systems* **104**, 49–63 (2017)