Cross-Domain Recommendation via Cluster-Level Latent Factor Model

Sheng Gao[1], Hao Luo[1], Da Chen[1], and Shantao Li[1]
Patrick Gallinari[2], Jun Guo[1]

[1] PRIS - Beijing University of Posts and Telecommunications, China
gaosheng@bupt.edu.cn, legand1989@gmail.com, chenda104@gmail.com,
buptlishantao@163.com, guojun@bupt.edu.cn
[2] LIP6 - Université Pierre et Marie Curie, France
patrick.gallinari@lip6.fr

Abstract. Recommender systems always aim to provide recommendations for a user based on historical ratings collected from a single domain (e.g., movies or books) only, which may suffer from the data sparsity problem. Recently, several recommendation models have been proposed to transfer knowledge by pooling together the rating data from multiple domains to alleviate the sparsity problem, which typically assume that multiple domains share a latent common rating pattern based on the user-item co-clustering. In practice, however, 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. In this paper, we propose a novel cluster-level based latent factor model 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. To this end, the proposed model is formulated as an optimization problem based on joint nonnegative matrix tri-factorization and an efficient alternating minimization algorithm is developed with convergence guarantee. Extensive experiments on several real world datasets suggest that our proposed model outperforms the state-of-the-art methods for the cross-domain recommendation task.

1 Introduction

Most recommender systems based on collaborative filtering aim to provide recommendations or rating predictions of an active user on a set of items belonging to only a single domain (e.g., movies or books) based on the historical user-item preference records [1]. However, in many cases, users rate only a limited number of items, even the item space is often very large. Then the available rating data can be extremely sparse, which may cause the recommendation models suffer from the overfitting problem and result in low-quality predictions as well.

In fact, there exists a considerable number of publicly available user-item rating datasets from multiple domains, which could have dependencies and correlations among the domains. Taking `Amazon` as an example, since the products in *Book* domain and *Music* domain may have correspondence in genre, and the respective customers can be considered to belong to similar groups sampled from the same population with alike social aspects [2], it would be useful to exploit a user's preferences on best-sellers in *Book* domain to help generate recommendations for that user on MP3 albums from the *Music* domain. Thus, instead of treating items from each single domain independently, users' preferences knowledge acquired in a single domain could be transferred and shared in other related domains, which has been referred to as *Cross-Domain Recommendation* [3]. Recently, several cross-domain recommendation models [2] [4] [5] have been proposed to transfer a common user-item rating pattern from a dense auxiliary rating dataset in other domains to a sparse rating dataset in the target domain of interest, which typically assume that multiple domains share the latent common rating pattern based on the user-item co-clustering. Thus, knowledge transfer and sharing among the related domains can be beneficial to alleviate the data sparsity problem.

However, the limitation of the existing methods is two-fold.

1. The existing models usually construct a latent space to represent the common latent structure shared across domains, which captures the rating pattern of user groups provided on item clusters. But in practice, the rating datasets from multiple domains may rarely contain exactly the same items or users, some domains are more closely related to the target domain of interest than others, simply forcing the subspaces in each domain to be identical is highly unrealistic. For example, books are more closely related to movies than to electronic gadgets, thus the different level of relatedness among multiple domains can not be captured by the identical rating patterns, which implies the existing methods are inflexible under the "all shared" latent factor assumption.

2. In practice, some related domains do not necessarily share such a common rating pattern, which has the intuition of "*Harmony in Diversity*" from the rating patterns in multiple domains. Moreover, the diversity among the related domains might outweigh the advantages of the common rating pattern, which may result in performance degradations. That is, the existing models cannot consider the domain-specific knowledge about the rating patterns to improve the mutual strengths in cross-domain recommendation.

To this end, in this paper, we propose a novel cluster-level based latent factor model to enhance the cross-domain recommendation. By deriving latent user-cluster factor and latent item-cluster factor from the available rating data, our proposed model can construct a latent space to represent the rating patterns of user groups on the item clusters. Based on a subspace learning of the latent space, the model can learn the common cluster-level user-item rating pattern that is shared across domains, especially, with the flexibility in controlling the

optimal level of sharing the relatedness among multiple domains, while the existing models do not provide this function for fine analysis of intrinsic cluster structure in rating data. Meanwhile, our proposed model can simultaneously learn the domain-specific cluster-level rating pattern from each domain, which contains the discriminative information propitious to improve across recommendation accuracy. The proposed model thus can exploit the mutual strengths of related domains by the shared common rating patterns as well as the domain-specific rating patterns immune to the discernable knowledge from each domain.

Moreover, our proposed model can be formulated as an optimization problem based on joint nonnegative matrix tri-factorization, and an efficient alternating method is developed to optimize the objective function with convergence guarantee. Extensive experiments on several real world datasets suggest that our proposed model outperforms the state-of-the-art methods for cross-domain recommendation task.

The paper is structured as follows. In Section 2 we briefly introduce the background and problem definition. In Section 3 the proposed framework based on the cluster-level latent factor model and the model specification are presented, followed by the efficient optimization algorithm. Then we describe experiments on several real world datasets, and provide comparisons with state-of-the-art methods in Section 4. The related work is discussed in Section 5. In Section 6 we present conclusions.

2 Background

2.1 Basic Model

Our proposed latent factor model is based on the orthogonal non-negative matrix tri-factorization (ONMTF) clustering algorithm [6], which is an effective framework for data mining. In this section, we introduce the background paradigm behind ONMTF that motivates our model.

In ONMTF model, a data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ from a rating dataset is factorized into a product of three nonnegative factors $\mathbf{U} \in \mathbb{R}^{M \times K}$, $\mathbf{S} \in \mathbb{R}^{K \times L}$ and $\mathbf{V} \in \mathbb{R}^{N \times L}$, such that $\mathbf{X} \approx \mathbf{U}\mathbf{S}\mathbf{V}^T$. This approximation can be achieved by the following matrix norm optimization:

$$\min_{\mathbf{U},\mathbf{S},\mathbf{V} \geq 0} \mathcal{J}_{\text{ONMTF}} = \|\mathbf{X} - \mathbf{U}\mathbf{S}\mathbf{V}^T\| \tag{1}$$

where $\|\cdot\|$ is the Frobenious norm of matrix. $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$ is an $M \times N$ rating matrix containing M users and N items. From the co-clustering perspective, the three nonnegative factors decomposed from ONMTF can be interpreted in the following way:

- $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_K]$ represent latent user factors, where each $\mathbf{u}_k$ is an $M \times 1$ vector indicating a probability distribution over M users and referred to as a *user-cluster latent factor*. Here $\arg\max_k (\mathbf{U})_{ik} = k^*$ means the ith user belongs to the k^*th user cluster (i.e., user group).

- $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_L]$ represent latent item factors, where each $\mathbf{v}_l$ is an $N \times 1$ vector indicating a probability distribution over N items and referred to as a *item-cluster latent factor*. Here $\arg\max_l (\mathbf{V})_{il} = l^*$ means the ith item belongs to the l^*th item cluster (i.e., item topic).
- $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_L]$ is an $K \times L$ weight matrix representing the rating patterns from K user clusters provided on L item clusters. $\mathbf{S}_{ij}$ can be considered as the probability that the kth user group rates the lth item cluster.

By clustering both sides of the data matrix simultaneously, ONMTF makes use of the interrelatedness between users and items, leading to better performance than other clustering methods. In [4] the authors proposed a latent factor model based on ONMTF clustering algorithm (i.e., Equation (1)) to provide recommendations for a sparse target domain (e.g., $\mathbf{X}_{\mathrm{tgt}}$) by sharing the latent common rating pattern knowledge in a latent space from the related dense domain (e.g., $\mathbf{X}_{\mathrm{src}}$), which was referred to as a *codebook* as $\mathbf{S}$ (i.e., $\mathbf{X}_{\mathrm{src}} \approx \mathbf{U}_{\mathrm{src}}\mathbf{S}\mathbf{V}_{\mathrm{src}}^T$). Thus, the codebook $\mathbf{S}$ was constructed by simultaneously clustering the users (rows) and items (columns) of $\mathbf{X}_{\mathrm{src}}$, indicating the rating that a user belonging to a specific user cluster $\mathbf{u}_{\mathrm{src}}$ provides on an item belonging to a specific item cluster $\mathbf{v}_{\mathrm{src}}$. Then the missing values in the target domain $\mathbf{X}_{\mathrm{tgt}}$ could be learned by duplicating the rows and columns of the codebook using $\mathbf{U}_{\mathrm{tgt}}\mathbf{S}\mathbf{V}_{\mathrm{tgt}}^T$, which was called CodeBook Transfer (CBT) recommendation model. Experimental results have shown that latent common information from a related domain can be derived to improve performance in the target domain. In the next section, we will discuss the way how to derive the latent common rating pattern and domain-specific rating pattern to enhance the cross-domain recommendation.

2.2 Problem Definition

Suppose that we are given multiple rating matrices from related domains for personalized item recommendation. Let τ be the domain index, and $\tau \in [1, t]$. In the τ-th domain rating matrix $\mathbf{D}_\tau$ there are a set of users $X_\tau = \{x_1^\tau, ..., x_{M_\tau}^\tau\}$ to rate a set of items $Y_\tau = \{y_1^\tau, ..., y_{N_\tau}^\tau\}$, where M_τ and N_τ represent the numbers of rows (users) and columns (items) respectively. Here the set of users and items across multiple domains may overlap or be isolated with each other. In this work we consider the more difficult case that neither the users or the items in the multiple rating matrices are overlapping. Moreover, each rating matrix contains a few observed ratings and some missing values to predict. We thus employ a binary weighting matrix $\mathbf{W}_\tau$ of the same size as $\mathbf{D}_\tau$ to mask the missing entries, where $[\mathbf{W}_\tau]_{ij} = 1$ if $[\mathbf{D}_\tau]_{ij}$ is observed and $[\mathbf{W}_\tau]_{ij} = 0$ otherwise. For easy understanding, we call the rating matrix of interest as the *target domain* and other related rating matrices the *source domains*.

In this paper, we consider how to predict the missing ratings in the target domain of interest by transferring correlated knowledge from the source domains as well as to learn the relatedness among multiple domains.

3 Our Proposed Model

Existing cross-domain recommendation models [2] [4] assume that the cluster-level structures hidden across domains can be extracted to learn the rating-pattern of user groups on the item clusters for knowledge transfer and sharing, and to clearly demonstrate the co-clusters of users and items. In this paper, we follow the framework proposed in [4] to extract the co-clustering of users and items as well as the shared common rating pattern. Thus, the initial co-clustering of the data matrix in domain τ can be performed by using ONMTF model as follows:

$$\min_{\mathbf{U}_\tau, \mathbf{S}_\tau^*, \mathbf{V}_\tau \geq 0} \mathcal{J}_\tau = \left\| [\mathbf{D}_\tau - \mathbf{U}_\tau \mathbf{S}_\tau^* \mathbf{V}_\tau^T] \circ \mathbf{W}_\tau \right\|^2 \tag{2}$$

where $\mathbf{U}_\tau \in \mathbb{R}^{M_\tau \times K_\tau}$ denotes the K_τ user clusters in the τth domain, and $\mathbf{V}_\tau \in \mathbb{R}^{N_\tau \times L_\tau}$ denotes the L_τ item clusters in the τth domain. $\mathbf{S}_\tau^* \in \mathbb{R}^{K_\tau \times L_\tau}$ represents the rating pattern of the kth user cluster made on the lth item cluster in the τth domain, where each entry $[\mathbf{S}_\tau^*]_{kl}$ is considered to be the average rating of the corresponding user-item co-cluster. $\mathbf{W}_\tau$ is the binary mask matrix, and $\circ$ denotes the entry-wise product. In the case of multiple related domains involving different sets of users or items, the assumption that users from different domains have a *similar rating pattern* on similar item clusters or topics can be held due to the harmony of users and items across the related domains, that denotes the same clustering topics in items (i.e., $L_\tau = L$) and the same cluster distributions over user groups (i.e., $K_\tau = K$) in each domain [4].

However, as we have introduced, the assumption does not hold in many real-world applications, where the items from multiple domains can not always find their explicit correspondence in the cluster level. Taking movie-rating and book-rating web sites for example, the movies and books can be considered to have similar clusters or topics based on their genre information (e.g., the categories of comedy or tragedy), but various customer groups from different websites may keep some domain-specific knowledge about the items of their respective domains in mind, showing different rating patterns and cognitive styles, such as the rating information about some Oscar-winning movies can not necessarily help to discover the clustering of the books on the topic of Oscar history. Inspired by this observation, we relax the unrealistic assumption in [4], and consider that the users from different domains should have similar explicit cluster-level correspondence while the items in each domain may hold their domain-specific knowledge.

3.1 Model Formulation

We propose a latent factor model based on the ONMTF framework to cluster the users and items in τth domain simultaneously and then learn a latent space to construct the cluster-level rating pattern of user-item co-clusters. Specifically, we partition the latent rating pattern *across domains* into a common part and a domain-specific part by the subspace learning of the latent space, that is $\mathbf{S}_\tau^* = [\mathbf{S}_0, \mathbf{S}_\tau]$, where $\mathbf{S}_0 \in \mathbb{R}^{K_\tau \times T}$ and $\mathbf{S}_\tau \in \mathbb{R}^{K_\tau \times (L_\tau - T)}$, T is the dimensionality of

Fig. 1. Illustration of our proposed CLFM model in the context of two related domains. $\mathbf{U}_1$ and $\mathbf{U}_2$ are the respective user-cluster latent factors, $\mathbf{V}_1$ and $\mathbf{V}_2$ are the respective item-cluster latent factors, $\mathbf{S}_0$ denotes the common latent rating pattern across domains, $\mathbf{S}_1$ and $\mathbf{S}_2$ are the domain-specific rating patterns in each domain.

shared common rating pattern and $(L_\tau - T)$ the dimensionality of τth domain-specific rating pattern.

Here the common part of rating pattern $\mathbf{S}_0$ captures the similar behavior styles of user clusters when they face the shared T clusters of related items from different domains, which can be captured to help alleviate the sparsity problem in each rating matrix. While the domain-specific part of rating pattern $\mathbf{S}_\tau$ can be learned to denote the discriminative aspect of user groups providing ratings on $(L_\tau - T)$ item clusters, which can be used to reveal the relatedness among multiple domains and improve the accuracy of recommendation.

Accordingly, in each domain our proposed Cluster-Level Latent Factor Model (CLFM) can learn the user-cluster latent factor $\mathbf{U}_\tau \in \mathbb{R}^{M_\tau \times K_\tau}$ where $K_\tau = K$, and item-cluster latent factor $\mathbf{V}_\tau = [\mathbf{V}_{\tau 0}^T, \mathbf{V}_{\tau 1}^T] \in \mathbb{R}^{N_\tau \times L_\tau}$ where $\mathbf{V}_{\tau 0} \in \mathbb{R}^{T \times N_\tau}$ corresponds to shared topics of item clusters and $\mathbf{V}_{\tau 1} \in \mathbb{R}^{(L_\tau - T) \times N_\tau}$ corresponds to domain-specific topics of item clusters in τth domain. The illustration of our proposed CLFM model can be found in Figure 1.

Then the learning of our proposed model can be derived in a unified subspace learning paradigm as the following objective function:

$$\min_{\mathbf{U}_\tau, \mathbf{S}_0, \mathbf{S}_\tau, \mathbf{V}_\tau \geq 0} \mathcal{J} = \sum_\tau \left\| [\mathbf{D}_\tau - \mathbf{U}_\tau [\mathbf{S}_0, \mathbf{S}_\tau] \mathbf{V}_\tau^T] \circ \mathbf{W}_\tau \right\|^2 \tag{3}$$

Specifically, to make the latent factors more accurate, we can impose some prior knowledge on the latent factors during the model learning. For example, the ℓ_1 normalization constraint can be imposed on each row of $\mathbf{U}_\tau$ and $\mathbf{V}_\tau$, i.e., $\mathbf{U}_\tau \mathbf{1} = \mathbf{1}$ and $\mathbf{V}_\tau \mathbf{1} = \mathbf{1}$.

Note that, from the construction of our proposed CLFM model, we can easily find that the recently proposed cross-domain recommendation model CBT is a special case of CLFM model with $L_\tau = T$, which means that there is no domain-specific rating pattern for each domain but only the shared common rating pattern across all the domains with the dimensionality of T. Therefore, our proposed CLFM model can not only exploit the optimal level of sharing information across multiple domains, but also reveal the individual differences in each domain.

3.2 Optimization

The optimization of our proposed model can be performed by an alternating minimization algorithm until convergence. For ease of understanding and without loss of generality, we set $\tau = 2$ [3]. The general objective function in Equation (3) can be rewritten as follows:

$$
\min_{\mathbf{U},\mathbf{S}_0,\mathbf{S}_1,\mathbf{S}_2,\mathbf{V}\geq 0} \mathcal{J} = \left\|[\mathbf{D}_1 - \mathbf{U}_1[\mathbf{S}_0,\mathbf{S}_1]\mathbf{V}_1^T]\circ\mathbf{W}_1\right\|^2
$$
$$
+ \left\|[\mathbf{D}_2 - \mathbf{U}_2[\mathbf{S}_0,\mathbf{S}_2]\mathbf{V}_2^T]\circ\mathbf{W}_2\right\|^2 \tag{4}
$$
$$
s.t. \quad \mathbf{U}_1\mathbf{1} = \mathbf{1}, \mathbf{U}_2\mathbf{1} = \mathbf{1}, \mathbf{V}_1\mathbf{1} = \mathbf{1}, \mathbf{V}_2\mathbf{1} = \mathbf{1}
$$

where $\mathbf{U}_1 \in \mathbb{R}^{M_1\times K}$, $\mathbf{U}_2 \in \mathbb{R}^{M_2\times K}$, $\mathbf{V}_1 = [\mathbf{V}_{10}^T, \mathbf{V}_{11}^T] \in \mathbb{R}^{N_1\times L_1}$, $\mathbf{V}_2 = [\mathbf{V}_{20}^T, \mathbf{V}_{21}^T] \in \mathbb{R}^{N_2\times L_2}$, $\mathbf{S}_0 \in \mathbb{R}^{K\times T}$, $\mathbf{S}_1 \in \mathbb{R}^{K\times(L_1-T)}$, $\mathbf{S}_2 \in \mathbb{R}^{K\times(L_2-T)}$.

To optimize the proposed model, we employ the alternating multiplicative updating algorithm [7], which warrants the nonnegativity of latent factors and provides an automatic step parameter selection. Since the objective function $\mathcal{J}$ in Equation (4) is not convex in $\mathbf{U},\mathbf{S}_0,\mathbf{S}_1,\mathbf{S}_2$ and $\mathbf{V}$ together, the alternating updating algorithm optimizes the objective function with respect to one set of parameters while fixing the others, and then exchanges the roles of the parameter sets alternatively. This procedure will be repeated for several iterations until convergence.

Learning $\mathbf{S}_1$ Taking the learning of $\mathbf{S}_1$ as an example, we will show how to optimize $\mathbf{S}_1$ by deriving its updating rule while fixing the other factors. For that we can rewrite the objective function in Equation (4) as follows:

$$
\min_{\mathbf{S}_1} \mathcal{J}(\mathbf{S}_1) = \left\|[\mathbf{D}_1 - \mathbf{U}_1\mathbf{S}_0\mathbf{V}_{10} - \mathbf{U}_1\mathbf{S}_1\mathbf{V}_{11}]\circ\mathbf{W}_1\right\|^2
$$
$$
+ \left\|[\mathbf{D}_2 - \mathbf{U}_2\mathbf{S}_0\mathbf{V}_{20} - \mathbf{U}_2\mathbf{S}_2\mathbf{V}_{21}]\circ\mathbf{W}_2\right\|^2 \tag{5}
$$

Then the derivative of $\mathcal{J}(\mathbf{S}_1)$ with respect to $\mathbf{S}_1$ is as follows:

$$
\frac{\partial\mathcal{J}(\mathbf{S}_1)}{\partial\mathbf{S}_1} = 2(\mathbf{U}_1^T([\mathbf{U}_1\mathbf{S}_0\mathbf{V}_{10}]\circ\mathbf{W}_1)\mathbf{V}_{11}^T - \mathbf{U}_1^T(\mathbf{D}_1\circ\mathbf{W}_1)\mathbf{V}_{11}^T)
$$
$$
+ 2\mathbf{U}_1^T([\mathbf{U}_1\mathbf{S}_1\mathbf{V}_{11}]\circ\mathbf{W}_1)\mathbf{V}_{11}^T
$$

---

[3] Following the definition in [4], domain $\mathbf{D}_1$ can be considered as the source domain, and $\mathbf{D}_2$ the target domain of interest.

Using the Karush-Kuhn-Tucker complementary condition for the nonnegativity of $\mathbf{S}_1$ and let $\frac{\partial \mathcal{J}(\mathbf{S}_1)}{\partial \mathbf{S}_1} = 0$, we can get the following updating rule for $\mathbf{S}_1$:

$$\mathbf{S}_1 \longleftarrow \mathbf{S}_1 \sqrt{\frac{\mathbf{U}_1^T (\mathbf{D}_1 \circ \mathbf{W}_1) \mathbf{V}_{11}^T}{\mathbf{U}_1^T ([\mathbf{U}_1 \mathbf{S}_0 \mathbf{V}_{10}] \circ \mathbf{W}_1) \mathbf{V}_{11}^T + \mathbf{U}_1^T ([\mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_{11}] \circ \mathbf{W}_1) \mathbf{V}_{11}^T}} \tag{6}$$

Learning $\mathbf{S}_2$ The latent factor $\mathbf{S}_2$ can be learned in a similar way. Here we can derive the updating rules for learning $\mathbf{S}_2$ as follows:

$$\mathbf{S}_2 \longleftarrow \mathbf{S}_2 \sqrt{\frac{\mathbf{U}_2^T (\mathbf{D}_2 \circ \mathbf{W}_2) \mathbf{V}_{21}^T}{\mathbf{U}_2^T ([\mathbf{U}_2 \mathbf{S}_0 \mathbf{V}_{20}] \circ \mathbf{W}_2) \mathbf{V}_{21}^T + \mathbf{U}_2^T ([\mathbf{U}_2 \mathbf{S}_2 \mathbf{V}_{21}] \circ \mathbf{W}_2) \mathbf{V}_{21}^T}} \tag{7}$$

Learning $\mathbf{S}_0$ The updating rules for learning latent factor $\mathbf{S}_0$ can be derived as follows:

$$\mathbf{S}_0 \longleftarrow \mathbf{S}_0 \sqrt{\frac{\mathbf{U}_1^T (\mathbf{D}_1 \circ \mathbf{W}_1) \mathbf{V}_{10}^T + \mathbf{U}_2^T (\mathbf{D}_2 \circ \mathbf{W}_2) \mathbf{V}_{20}^T}{\mathbf{A} + \mathbf{B}}}$$
$$\mathbf{A} = \mathbf{U}_1^T ([\mathbf{U}_1 \mathbf{S}_0 \mathbf{V}_{10}] \circ \mathbf{W}_1) \mathbf{V}_{10}^T + \mathbf{U}_1^T ([\mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_{11}] \circ \mathbf{W}_1) \mathbf{V}_{10}^T \tag{8}$$
$$\mathbf{B} = \mathbf{U}_2^T ([\mathbf{U}_2 \mathbf{S}_0 \mathbf{V}_{20}] \circ \mathbf{W}_2) \mathbf{V}_{20}^T + \mathbf{U}_2^T ([\mathbf{U}_2 \mathbf{S}_2 \mathbf{V}_{21}] \circ \mathbf{W}_2) \mathbf{V}_{20}^T$$

Learning $\mathbf{U}_1$ The latent factor $\mathbf{U}_1$ can be learned in the similar way. Here we can derive the updating rules for learning $\mathbf{U}_1$ as follows:

$$\mathbf{U}_1 \longleftarrow \mathbf{U}_1 \sqrt{\frac{(\mathbf{D}_1 \circ \mathbf{W}_1) \mathbf{V}_1 [\mathbf{S}_0, \mathbf{S}_1]^T}{([\mathbf{U}_1 [\mathbf{S}_0, \mathbf{S}_1] \mathbf{V}_1^T] \circ \mathbf{W}_1) \mathbf{V}_1 [\mathbf{S}_0, \mathbf{S}_1]^T}} \tag{9}$$

Note that during the learning of $\mathbf{U}_1$, we formulate the Lagrange function for the optimization with normalization constraint on the latent factor.

Learning $\mathbf{U}_2$ The updating rules for learning latent factor $\mathbf{U}_2$ can be derived as follows:

$$\mathbf{U}_2 \longleftarrow \mathbf{U}_2 \sqrt{\frac{(\mathbf{D}_2 \circ \mathbf{W}_2) \mathbf{V}_2 [\mathbf{S}_0, \mathbf{S}_2]^T}{([\mathbf{U}_2 [\mathbf{S}_0, \mathbf{S}_2] \mathbf{V}_2^T] \circ \mathbf{W}_2) \mathbf{V}_2 [\mathbf{S}_0, \mathbf{S}_2]^T}} \tag{10}$$

Here we also formulate the Lagrange function for the optimization with normalization constraint in learning $\mathbf{U}_2$.

Learning $\mathbf{V}_1$ The latent factor $\mathbf{V}_1$ can be learned in the similar way as for constrained optimization. Here we can derive the updating rules for learning $\mathbf{V}_1$ as follows:

$$\mathbf{V}_1 \longleftarrow \mathbf{V}_1 \sqrt{\frac{[\mathbf{S}_0, \mathbf{S}_1]^T \mathbf{U}_1^T (\mathbf{D}_1 \circ \mathbf{W}_1)}{[\mathbf{S}_0, \mathbf{S}_1]^T \mathbf{U}_1^T ([\mathbf{U}_1 [\mathbf{S}_0, \mathbf{S}_1] \mathbf{V}_1^T] \circ \mathbf{W}_1)}} \tag{11}$$

Note that $\mathbf{V}_{10}^T = \mathbf{V}_1(:, 1 : T)$ and $\mathbf{V}_{11}^T = \mathbf{V}_1(:, (T+1) : L_1)$.

Learning $\mathbf{V}_2$ The updating rules for learning latent factor $\mathbf{V}_2$ as for constrained optimization can be derived as follows:

$$\mathbf{V}_2 \longleftarrow \mathbf{V}_2 \sqrt{\frac{[\mathbf{S}_0, \mathbf{S}_1]^T \mathbf{U}_2^T (\mathbf{D}_2 \circ \mathbf{W}_2)}{[\mathbf{S}_0, \mathbf{S}_2]^T \mathbf{U}_2^T ([\mathbf{U}_2[\mathbf{S}_0, \mathbf{S}_2]\mathbf{V}_2^T] \circ \mathbf{W}_2)}} \tag{12}$$

Note that $\mathbf{V}_{20}^T = \mathbf{V}_2(:, 1 : T)$ and $\mathbf{V}_{21}^T = \mathbf{V}_2(:, (T+1) : L_2)$.

Convergence Analysis Based on the above updating rules for learning different latent factors, we can prove that the learning algorithm is convergent.

Theorem 1. *Using the updating rules for $\mathbf{S}_0$ in Equation (8), $\mathbf{S}_1$ in Equation (6), $\mathbf{S}_2$ in Equation (7), $\mathbf{U}_1$ in Equation (9), $\mathbf{U}_2$ in Equation (10), $\mathbf{V}_1$ in Equation (11) and $\mathbf{V}_2$ in Equation (12), the objective function in Equation (4) will monotonically decrease, thus the learning algorithm converges.*

The proof could be refereed to [8] [9] for more details.

4 Experiments

In the experiments, 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:

- NMF (Nonnegative Matrix Factorization) based model [7]: a single-domain model which employs nonnegative matrix factorization method to learn the latent factors in each domain and provide the prediction performance separately.
- FMM (Flexible Mixture Model) based model [10]: a single-domain model which uses probabilistic mixture model to learn latent cluster structure in each single domain and then provide the single domain performance separately.
- CBT (CodeBook Transfer) based model [4]: a cross-domain model which can only transfer and share the common rating pattern by the codebook information across multiple domains.
- CLFM model: our proposed model.

In terms of the cross-domain recommendation task, we evaluate these methods in terms of two ways: one is the impact of different level of sharing common information across domains, the other is to check the effectiveness of the cross-domain models to alleviate the sparsity problem.

4.1 Datasets

For the experiments we have used three benchmark real-world datasets for performance evaluation:

- *MovieLens* dataset [4]: contains more than 100,000 movie ratings with the scales from 1 to 5 provided by 943 users on 1,682 movies. Following [2] we randomly choose 500 users with more than 16 ratings and 1000 movies for experiments.
- *EachMovie* dataset [5]: contains 2.8 million movie ratings with the scales from 1 to 6 provided by 72,916 users on 1,628 movies. Following [2] we also randomly choose 500 users with more than 20 ratings and 1000 movies for experiments.
- *Book-Crossing* dataset [6]: contains more than 1.1 million ratings with the scales from 0 to 9 provided by 278,858 users on 271,379 books. We still randomly select 500 users and 1000 books with more than 16 ratings for each item in the experiments.

Note that for all the datasets we have normalized the rating scales from 1 to 5 in the average style for fair comparison. Our proposed CLFM model can handle various types of users' rating information, including the explicit rating (e.g., from 1 to 5) as well as the implicit preferences of users (e.g., visit, click or comment), which are based on the flexible function of the inner product of the latent factors learned from the observations in different styles.

4.2 Experimental Setup

Following the work in [4], we examine the compared models for the cross-domain recommendation task. For that, 300 users with their ratings in each dataset are randomly selected as the training data, and the remaining 200 users for testing. For each test user, we consider to keep different sizes of the observed ratings as the initialization of each user in the experiments, i.e., 5 or 10 ratings of each test user are given to avoid cold-start problem (e.g., $ML - Given5$ or $ML - Given10$ in the *MovieLens* dataset as illustrated in Table 1) and the remaining ratings are used for evaluation.

We choose *MovieLens* vs *EachMovie*, *EachMovie* vs *Book-Crossing* and *Movie-Lens* vs *Book-Crossing* as three kinds of related domains (the former is source domain and the latter the target domain for CBT model in the experiments) to discover the relatedness among different domains. To check the performances of different methods, we use MAE (Mean Absolute Error) [7] as the evaluation metric. In the experiments we conduct the methods by repeating the process 10 times and report the average results.

---

[4] http://www.grouplens.org/node/73
[5] http://www.cs.cmu.edu/ ∼ lebanon/IR-lab.htm
[6] http://www.informatik.uni-freiburg.de/ ∼ cziegler/BX/
[7] MAE is computed as $MAE = \sum_{i \in O} |R_i - R_i^*|/|O|$, where $|O|$ denotes the number of test ratings, R_i is the true value and R_i^* is the predicted rating. The smaller the value of MAE is, the better the model performs.

Table 1. MAE performances of the compared models on *MovieLens* vs *EachMovie* related domains under different configurations. ML-Given5 means 5 ratings of each test user in *MovieLens* dataset are given while the remaining ratings are used for evaluation. The combined settings *ML-Given5* vs *EM-Given5* and *ML-Given10* vs *EM-Given10* are conducted. Best results are in bold.

	NMF	FMM	CBT	CLFM
ML-Given5	0.9652	0.9338	0.9242	**0.9121**
ML-Given10	0.9411	0.9203	0.9101	**0.8815**
EM-Given5	0.9803	0.9569	0.9333	**0.9209**
EM-Given10	0.9425	0.9396	0.9185	**0.8907**

4.3 Experimental Results

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. The number of users clusters K and item clusters L in each rating dataset have been chosen in the range of $[10, 100]$. For NMF model, the dimension of factorization is set to be 50. For CBT model, we set $K = 30$ and $L = 80$. For all the matrix factorization based models, the number of iterations has been set at 50.

Table 1 shows the MAE performance of the compared models in the *Movie-Lens* vs *EachMovie* related domains under different configurations, where in the both domains the number of users and item clusters $K = 30$ and $L = 50$ respectively, the parameter $T = 40$ denoting the dimensionality of shared common subspace. In the experiments, we have 5 and 10 ratings of each test user in the *MovieLens* and *EachMovie* datasets that are given for training while the remaining ratings are used for test, and the compared models are evaluated on the different combined settings as *ML-Given5* vs *EM-Given5* and *ML-Given10* vs *EM-Given10*. From the results we can observe that the best performing method among all the models is our proposed CLFM model. The FMM model performs slightly better than the NMF model, which implies that the cluster-level based methods can gain meaningful knowledge from user and item clusters due to the co-clustering property in the FMM model.

Moreover, the cross-domain based models (i.e., CBT and CLFM) clearly outperforms the single domain based models (i.e., NMF and FMM), which shows that the latent cross-domain common rating pattern can indeed aggregate more useful knowledge than the single-domain methods do individually. Specifically, our proposed CLFM model provides even better results than the state-of-the-art cross-domain recommendation model CBT, which proves the benefits of the CLFM model with the ability of extracting the common rating pattern and the domain-specific knowledge to enhance the cross-domain recommendation accuracy.

Again, Table 2 shows the MAE performances of the compared models on *EachMovie* vs *Book-Crossing* related domains under different configurations,

Table 2. MAE performances of the compared models on *EachMovie* vs *Book-Crossing* related domains under different configurations. The combined settings *EM-Given5* vs *BC-Given5* and *EM-Given10* vs *BC-Given10* are conducted. Best results are in bold.

	NMF	FMM	CBT	CLFM
EM-Given5	0.9803	0.9569	0.9541	**0.9334**
EM-Given10	0.9425	0.9366	0.9225	**0.9091**
BC-Given5	0.7326	0.7192	0.6978	**0.6757**
BC-Given10	0.7198	0.6924	0.6805	**0.6514**

where in the both domains the parameters $K = 30$, $L = 80$ and $T = 40$ as the optimal values. The combined settings *EM-Given5* vs *BC-Given5* and *EM-Given10* vs *BC-Given10* are conducted in the experiments. From the results we can draw the similar conclusion, that is, our proposed CLFM model performs better than the other related methods. The results about *MovieLens* vs *Book-Crossing* domains have similar characteristics and are omitted here.

Meanwhile, from Table 1 and Table 2 we can also discover that the performances for the item recommendation in the *EachMovie* dataset are not identical even in terms of the same users and items when combined with different related domains in the experiments. The results show that different domains may have various levels of shared information, which are hidden across domains. The remarkable advantage of our proposed CLFM model is to capture and control the level of sharing the relatedness by the shared subspace dimensionality T. In the CLFM model the value of the parameter T is limited between 0 and $\min(L_1, L_2)$, i.e., ranges from *no sharing* to *full sharing*. Figure 2 provides the performances of the compared models in *EachMovie* domain as a function of the parameter T under the configuration of $K = 30$, $L = 80$ given 10 ratings observed for each test user. From the figure, we can find that T increases with the level of sharing pattern between the two domains until it reaches the optimal value $T = 40$. This observation confirms that our proposed CLFM model has more flexible and efficient ability to capture latent common rating pattern than the other methods.

Figure 3 demonstrates the convergence curve of running the proposed CLFM model in *MovieLens* vs *Book-Crossing* domains under the configuration of $K = 30$, $L = 80$ and $T = 40$ given 10 ratings available for each test user. From the results we can observe that our proposed model can converge quickly after about 20 iterations, which proves the efficiency and scalability of the CLFM model in the cross-domain recommendation task.

5   Related Work

Cross-domain recommendation is an emerging research topic, which considers to incorporate relevant data sources from different domains and combine with the original target data to improve the recommendation [11]. For example, it is possible to merge multiple rating matrices to leverage rating behavior similarity in one domain to predict ratings in the other domain. Kuflik et al. [12] first
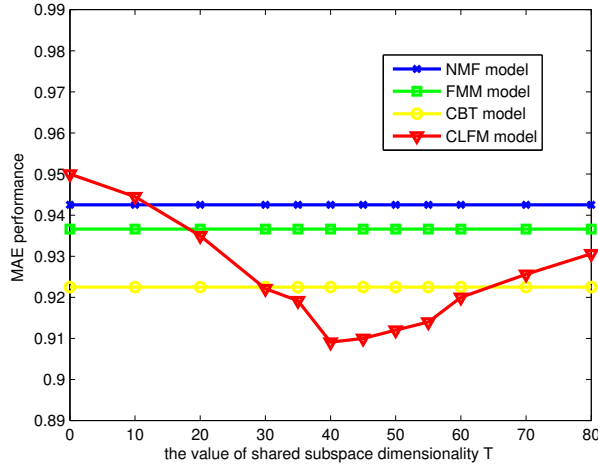
Fig. 2. MAE performance of the compared models with respect to the value of shared subspace dimensionality T in *EachMovie* domain.

proposed cross domain mediation problem and introduced several techniques for importing relevant data. Pan et al. [13] proposed the model to transform knowledge from domains which have heterogeneous forms of user feedback. Cremonesi et al. [14] considered to model the classical similarity relationships as a direct graph and explore all possible paths connecting users or items in order to find new cross-domain relationships. Tang et al. [15] proposed the cross-domain topic learning model to predict the cross-domain collaborations through topic layers instead of at author layers, which alleviated the sparseness issue. Winoto et al. [16] proposed to uncover the association between user preferences on related items across domains to provide the cross-domain recommendation. However, all the above models always consider the same users or items across different domains, which is not a realistic setting. Actually, the most relevant works with ours are [2] [4]. They suggested to leverage useful knowledge from a different domain and extracted latent rating pattern across domains with non-overlap users and items. Moreno et al. [5] addressed the sparsity problem by integrating the appropriate amount of knowledge from each domain in order to enrich the target domain. Moreover, the majority of the existing work assumes that the source and target domains are related but do not suggest methods to calculate the relatedness among domains, which has been addressed in our work.

There are other recent studies which have been made on applying cross domain techniques, and transfer learning in particular into information recommendation task. Transfer learning aims to leverage the knowledge in the source domain to facilitate the learning tasks in the target domain [17]. The general idea of the existing methods is to utilize the common latent structure shared across domains as the bridge for knowledge transfer. For example, Xue et al. [18] have
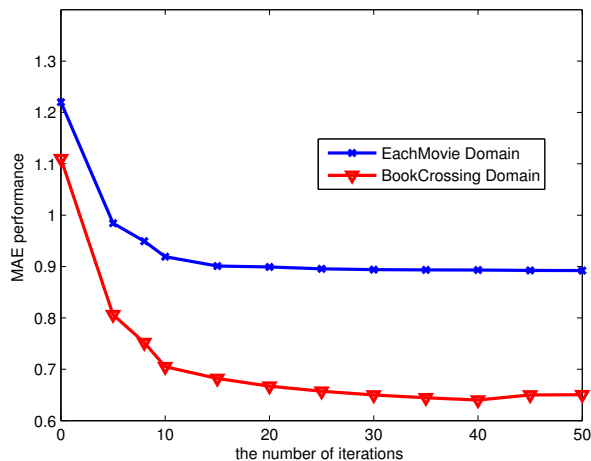
Fig. 3. Convergence curve of running CLFM model in *MovieLens* vs *Book-Crossing* domains with respect to nb. of iterations

addressed the problem of using the auxiliary labeled data to help classify the unlabeled data in the target domain. Shi et al. [19] proposed a generalized tag-induced cross-domain collaborative filtering model to exploit user-contributed tags as common characteristics to link different domains together and transfer the knowledge between different domains. However, transferring knowledge across domains is a challenging task since it cannot be guaranteed that the knowledge of other domains is useful for the target domain. In this paper, we not only consider the common latent rating pattern across domains but also extract the discriminative domain-specific information to improve the mutual strengths in each domain.

6 Conclusion

In this paper, we proposed a novel Cluster-Level based Latent Factor Model (CLFM) based on the framework of joint nonnegative matrix tri-factorization. The CLFM model can construct a latent space to represent the rating patterns of user groups on the item clusters from each domain, then based on a subspace learning of the latent space, CLFM model not only learn shared common rating pattern across multiple rating matrices to alleviate the sparsity problems in individual domain, with the flexibility in controlling the optimal level of sharing the relatedness among domains, but also learn the domain-specific cluster-level rating pattern from each domain that contains the discriminative information propitious to improve across recommendation accuracy. The proposed model thus could exploit the mutual strengths of related domains by the shared common rating patterns as well as the domain-specific rating patterns from each domain.

The experimental results have validated that our proposed CLFM model indeed can benefit from the cluster-level rating patterns and outperforms the state-of-the-art methods for cross-domain recommendation task.

There are still several extensions to improve our work. Firstly, it is necessary to compare our proposed model against the two more recent methods [5] to explore a deeper understanding in the cross-domain recommendation task. Second, our proposed CLFM model should be evaluated on large scale rating dataset to exploit its scalable computational ability. Third, a probabilistic version would be the natural extension of our proposed CLFM model, which may exhibit better interpretable properties.

References

1. Y. 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*, KDD '08, pages 426–434, 2008.
2. B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 617–624, 2009.
3. B. Li. Cross-domain collaborative filtering: a brief survey. *23rd IEEE International Conference on Tools with Artificial Intelligence*, pages 1085–1086, 2011.
4. B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 2052–2057, 2009.
5. O. Moreno, B. Shapira, L. Rokach, and G. Shani. Talmud: transfer learning for multiple domains. In *In 21st ACM International Conference on Information and Knowledge Management*, CIKM'12, pages 425–434, 2012.
6. C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. *SIGKDD*, pages 126–135, 2006.
7. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *NIPS*, pages 556–562, 2000.
8. Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Temporal link prediction by integrating content and structure information. *CIKM '11*, pages 1169–1174, 2011.
9. Chris Ding, Tao Li, and Michael I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2052–2067, 2008.
10. L. Si and R. Jin. Flexible mixture model for collaborative filtering. ICML'03, pages 704–711, 2003.

11. Ignacio Fernadez-Tobis, Iva Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. In *In Proceedings of the 2nd Spanish Conference on Information Retrieval*, 2012.

12. S. Berkovsky, T. Kuflik, and F. Ricci. Cross-domain mediation in collaborative filtering. User Modeling 2007, pages 355–359, 2007.

13. W. Pan, E. Xiang, N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the 24rd AAAI Conference on Artificial Intelligence*, pages 425–434, 2010.

14. Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. Cross-domain recommender systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 496–503, 2011.

15. Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1285–1293, 2012.

16. Pinata Winoto and Tiffany Ya 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 Comput.*, pages 209–225, 2008.

17. S. Pan and Q. Yang. A survey on transfer learning. In *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pages 1345–1359, 2010.

18. G. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. SIGIR, pages 627–634, 2008.

19. Yue Shi, Martha Larson, and Alan Hanjalic. Generalized tag-induced cross-domain collaborative filtering. *CoRR*, abs/1302.4888, 2013.