Deep Transfer Collaborative Filtering with Geometric Structure Preservation for Cross-Domain Recommendation

Yachen Kang*, Sibo Gai*, Feng Zhao and Donglin Wang†
School of Engineering, Westlake University, Hangzhou, China
Westlake Institute for Advanced Study, Hangzhou, China
{kangyachen, gaisibo, zhaofeng, wangdonglin}@westlake.edu.cn

Ao Tang
WeCar (Shenzhen) Technology Co., Ltd.
Shenzhen, China
ao.tang@weicheche.cn

***Abstract*—Collaborative filtering (CF) is one of the most effective approaches for recommender systems by exploiting user-item behavior interactions. However, in real applications, the rating matrix is usually sparse, causing a poor performance. Numerous CF methods tend to incorporate the side information for the enrichment of priors. Cross-domain recommendation is an alternative to alleviate these two problems by referring to the knowledge of relevant domains. Due to the sparsity of the ratings and side information, the resulting latent factors might not be effective as expected. In this paper, we incorporate both the side information and deep knowledge transfer in CF models. A general architecture of deep transfer collaborative filtering with geometry preservation (DTCFGP) is proposed by integrating cross-domain collective matrix factorization, deep feature learning and the graph modeling. We exhibit a instantiations of our architecture by employing non-negative matrix tri-factorization and stacked denoising autoencoder (SDAE) in both source and target domains, where the common latent factors are taken as a bridge between domains due to its across-domain stability and data geometric structure is evaluated by using a nearest neighbor graph modeling. Extensive experiments on various real-world datasets demonstrate the effectiveness of our proposed approach in comparison to state-of-the-art approaches.**

***Index Terms*—recommendation, collaborative filtering, deep learning, transfer learning, non-negative matrix tri-factorization**

I. Introduction

Recommendation becomes more important and draws much more attention in current information-explosion era. A great number of classical recommendation methods have been proposed during the last decade, which could be largely classified into two categories: content-based methods and collaborative filtering (CF) based methods [32]. Content-based methods consider user profile or item content information for recommendation while CF-based methods ignore the content information and utilize the user's past activities or preferences for recommendation. Thus, CF-based methods are more likely to be selected for real recommendation applications owing to a better performance.

Matrix factorization techniques are the main cornerstone, which can learn effective latent factors for users and items from the rating matrix ([6], [18], [29]). Moreover, neural CF methods are recently designed by using neural networks to learn the interaction function from the original data ([7], [13]). However, both matrix factorization and neural CF methods suffer from two main issues: data sparsity and cold start. When the historical data is sparse, it is hard to achieve a satisfactory performance by using these methods. On the other hand, before acquiring the ratings assigned by a large number of users, we are unable to implement the recommendation.

In order to overcome these problems, one solution is to integrate CF with the side information to exploit piror features ([33], [34]), where the side information can be either utilized as a regularization [6] or tightly coupled with CF by using deep learning ([20], [39]). These hybrid methods seek to combine the side information and CF-based methods for a better performance. Nevertheless, due to the sparsity of the ratings as well as side information, the resulting latent factors might not be effective as expected [3].

Another solution is to address these problems by transferring or learning the knowledge from relevant domains to the current domain (called target domain) and then utilizing cross-domain recommendation techniques [14]. In real applications, we can track the same user's participation in a couple of recommendation systems to acquire various information in different domains. Due to the factorization on a sparse rating matrix, it is better to use deep transfer structures to learn effective latent factors. It does improve the recommendation performance in the target domain by deep cross-domain learning ([15], [21]). However, no prior work has tightly integrated cross-domain collective matrix factorization and deep structure for recommendations.

Furthermore, from the angle of geometric structure, the original data samples may be randomly drawn from a latent distribution expanded by a low-dimensional manifold embedded in a high-dimensional space [2]. This geometric structure indicates that close samples are more likely to be assigned identical labels while far samples are assigned different labels. This kind of geometry should be preserved when the common latent factors are taken as the bridge for knowledge transfer [24]. Otherwise, the transfer learner is incapable of estimating

*Equal contribution, joint first authors.
†Corresponding author

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

labels smoothly.

In this paper, we attempt to integrate cross-domain collective matrix factorization and deep structure for recommendation, where we minimize the variation of the geometric structure during the knowledge transfer. A general architecture of deep transfer collaborative filtering with geometry preservation (DTCFGP) is proposed by integrating cross-domain matrix tri-factorization and deep learning, where the geometric structure is evaluated and preserved via graph modeling.

Deep structure in DTCFGP deals with both the ratings and the side information to achieve effective latent representations. Non-negative matrix tri-factorization is considered in this paper because it decomposes an association matrix that provides additional degree of freedom and represents a common component of both domains [5]. On one hand, non-negative matrix tri-factorization on the ratings generates private latent factors for both users and items in each domain, which are respectively connected with an individual deep structure and the graph model. On the other hand, this tri-factorization generates common latent factors, implying the association between users' latent factors and items' latent factors, which are treated as a bridge between source and target domains due to the across-domain stability [44]. DTCFGP jointly optimizes deep representation learning, cross-domain CF and data geometric structure in the process of knowledge transfer.

II. Related Work

We aim to propose deep transfer collaborative filtering with geometry preservation. In general, our work is related to the following topics: matrix factorization based CF, deep learning-based CF, and matrix factorization based transfer learning.

A. Matrix factorization based CF

Matrix factorization is the most popular technique to derive latent factor models. By adopting different loss functions, a variety of matrix factorization models have been investigated, such as non-negative matrix factorization [19], probabilistic matrix factorization [29], Bayesian probabilistic matrix factorization [28], and max-margin matrix factor [27]. One of important matrix factorization methods is collective matrix factorization and its tri-factorization variants ([4], [10], [19], [33]). Non-negative matrix tri-factorization provides additional degree of freedom by decomposing the rating matrix as the product of three matrices [5].

Incorporating the side information in matrix factorization approaches has shown a promising performance in handling with the sparsity issue ([25], [26]). Bayesian matrix factorization approach with side information and Dirichlet process mixtures are proposed in [26]. A variational Bayesian matrix factorization method is proposed in [17] while a hierarchical Bayesian matrix factorization method is proposed in [25], where the side information is utilized in both works. However, the resulting latent factors are still not so good as expected when the rating matrix and side information are sparse [6]. Different from these methods, we further take the advantage of deep structure and knowledge transfer.

B. Deep learning based CF

Deep learning is a very powerful tool for learning representations these years. Deep learning based CF is comparatively new, which could alleviate the sparse problem to certain extent. Restricted Boltzmann machine is modified in [30] for CF tasks while ordinal Boltzmann machines are proposed in [36] for CF tasks. Recently, some deep learning models learn latent factors from the side information by combining deep structure and CF ([6], [20], [21], [39]) . In [20], a deep collaborative filtering model is presented by integrating matrix factorization and Bayesian stacked denoising autoencoders, where only item's features are extracted by deep learning. In [39], deep collaborative filtering is proposed by combining marginalized denoising auto-encoder and matrix factorization, which learns deep features for both users and items. In [6], a hybrid deep collaborative filtering model is proposed by involving the side information with SDAEs. A deep heterogeneous autoencoder is proposed in [21] for collaborative filtering on multiple data sources, where the time-series sequence is learned by using long short-term memory (LSTM) structure while non-time-series sources are encoded by using the conventional autoencoders. In ([8], [16]), the authors use transfer learning but miss to preserve the geometric structure. Different from these methods, we further consider the transfer of knowledge from related domains and meanwhile preserve the geometric structure.

C. Matrix factorization based transfer learning

Knowledge can be transferred across domains by using multiple decomposed matrices via collective matrix factorization [33]. In transfer learning, some common latent factors are taken as a bridge to link source and target domains [24]. Previous matrix factorization based transfer learners usually uncover these latent factors by optimizing predefined objective functions, including maximizing the empirical likelihood ([40], [44], [45]), or preserving the intrinsic geometric structure ([22], [35], [37], [38]). Its tri-factorization variants have been extensively studied for transfer learning recently [11]. Collective matrix factorization jointly factorizes multiple matrices with correspondences between rows and columns while sharing a set of common latent factors across different matrices [23]. Collective matrix factorization based methods maximize the empirical likelihood among multiple domains and the common latent factors are then used as a bridge for knowledge transfer [44]. Recently, the geometric structure instead of the empirical likelihood has been explored for transfer learning, including cross-domain spectral classification [22], manifold alignment ([37], [38]) and transfer component analysis [35]. Most of the prior works consider these two objectives separately without exploring the benefit of integrating them in a unified manner ([31], [42], [41]). One exception is that the authors in [24] attempt to propose matrix factorization with graph co-regularization. Different from all these methods above, aside from optimization on two objective functions, we further take the advantage of deep structure ([1], [14], [15]) and a hybrid modes to integrate the ratings and side information.

III. Notations and Overview

Define $\mathcal{D}_s$ as the source domain and $\mathcal{D}_t$ as the target domain in this paper. *For convenience, the domain indices are denoted as* $d \in \{s, t\}$. Table III summarizes the primary symbols used in our approach.

In a recommendation setting, the user-item matrix in domain $\mathcal{D}_d$ can be decomposed as the production of three non-negative matrices $\mathbf{R}_d = \mathbf{U}_d \mathbf{H} \mathbf{V}_d^{\mathrm{T}}$, where $\mathbf{U}_d$ indicates the user latent factors, $\mathbf{V}_d$ indicates the item latent factors, and $\mathbf{H}$ indicates the association of $\mathbf{U}_d$ and $\mathbf{V}_d$. Non-negative matrix tri-factorization is integrated with deep structure by private latent factors $\mathbf{U}_d$, $\mathbf{V}_d$ and each deep representation, where an individual deep structure deals with both the side information and the ratings. The source domain $\mathcal{D}_s$ is connected with the target domain $\mathcal{D}_t$ via the common latent factors $\mathbf{H}$.

Geometric structure implies a strategy that close samples are more likely to be assigned the same label while far samples tend to assign a different label, which should be preserved when the common latent factors $\mathbf{H}$ are taken as a bridge for cross-domain recommendation [24]. Based on the ratings on items, we could conduct an item graph $G_d^{(v)}$ with vertices each representing an item in domain $\mathcal{D}_d$. According to the duality property between users and items, the users are also sampled from a distribution supported by another low-dimensional manifold [9]. Thus we could also construct a user graph $G_d^{(u)}$ with vertices each representing a user in domain $\mathcal{D}_d$. Define by $\mathbf{A}_d^{(u)}$ and $\mathbf{A}_d^{(v)}$ the similarity matrix for the graph $G_d^{(u)}$ and $G_d^{(v)}$, respectively, which might be a cos function or a Laplacian function [24], [43]. Define $\mathbf{O}_d^{(v)} = diag(\sum_i (\mathbf{A}_d^{(v)})_{ij}$ to calculate the Laplacian matrix.

We have an individual deep structure to learn effective latent representations of users or items in each domain, which takes the ratings and/or the side information as input in various ways. In total, we design four deep structures for users and items in source and target domains. Based on the general block diagram, we propose a specific instantiations, where we employ SDAE as our deep structure.

IV. Methodologies

A. DTCFGP

TABLE I

Summary of primary notations, where INFO means information.

Notation	Description
m_d,n_d	Number of users and items in $\mathcal{D}_d$
$\mathbf{R}_d = [r_{d,ij}]_{m_d \times n_d}$	Rating matrix in $\mathcal{D}_d$
$\mathbf{C}_d = [c_{d,ij}]_{m_d \times n_d}$	Indicative matrix of $\mathbf{R}_d$ in $\mathcal{D}_d$
$\mathbf{U}_d = [u_{d,ij}]_{m_d \times k_1}$	Latent factors of users in $\mathcal{D}_d$
$\mathbf{u}_{d,i} \in \mathbb{R}^{1 \times k_1}$	Latent factor of user i in $\mathcal{D}_d$
$\mathbf{V}_d = [v_{d,ij}]_{n_d \times k_2}$	Latent factors of items in $\mathcal{D}_d$
$\mathbf{v}_{d,j} \in \mathbb{R}^{1 \times k_2}$	Latent factor of item j in $\mathcal{D}_d$
$\mathbf{H} \in \mathbb{R}^{k_1 \times k_2}$	Common latent factors
$\mathbf{x}_d^{(u)}$, $\mathbf{x}_d^{(v)}$	Rating vectors in terms of users/items
$\mathbf{F}_d^{(u)}$, $\mathbf{F}_d^{(v)}$	TFE results in terms of users/items
$\mathbf{p}_d^{(u)}$, $\mathbf{p}_d^{(v)}$	Side INFO of users/items in $\mathcal{D}_d$
$\mathbf{W}_d^{(u)} = [\mathbf{W}_{d,l}^{(u)}]_{1 \times L_d^{(u)}}$	Weights of users' SDAE in $\mathcal{D}_d$
$\mathbf{W}_d^{(v)} = [\mathbf{W}_{d,l}^{(v)}]_{1 \times L_d^{(v)}}$	Weighs of items' SDAE in $\mathcal{D}_d$
$\mathbf{b}_d^{(u)}$, $\mathbf{b}_{d,l}^{(u)}$, $\mathbf{b}_d^{(v)}$, $\mathbf{b}_{d,l}^{(v)}$	Biases of users/items' SDAE in $\mathcal{D}_d$
$\mathbf{h}_{d,o}^{(u)}$, $\mathbf{h}_{d,o}^{(v)}$	Latent representations in SDAE
$\mathbf{h}_{d,o}^{(u_i)}$, $\mathbf{p}_{d,i}^{(u)}$;$\mathbf{h}_{d,o}^{(v_j)}$, $\mathbf{p}_{d,j}^{(v)}$	Corresponding to $\mathbf{u}_{d,i}$, $\mathbf{v}_{d,j}$ in $\mathcal{D}_d$
$L_d^{(u)}$, $L_d^{(v)}$	Number of layers in SDAE
$\mathbf{z}_d^{(u)}$, $\mathbf{z}_{d,l}^{(u)}$, $\mathbf{z}_d^{(v)}$, $\mathbf{z}_{d,l}^{(v)}$	Weights for the secondary input
$\mathbf{A}_d^{(u)}$, $\mathbf{A}_d^{(v)}$	Similarity matrix for Graph $G_d^{(u)}$, $G_d^{(v)}$

In DTCFGP, as shown in Fig. 1, the SDAE deals with both the ratings and the side information, where the ratings are the primary input of the SDAE and the side information is the primary input. The information generated from the transformation and feature extraction (TFE) is taken as the primary input instead of the raw ratings to avoid the sparsity. The transformation is similar to [6] and feature extraction considers statistical characteristics. Specifically, we have $\mathbf{F}_d^{(u)} = [\mathbf{R}_d]_F$ and $\mathbf{F}_d^{(v)} = \left[\mathbf{R}_d^{\mathrm{T}}\right]_F$, where the operator $[\cdot]_F$ considers each row of the matrix and sequentially concatenates one-hot coding of the maximum, the minimum, the median, the mode, the quartiles and the rounding mean. Moreover, $\mathbf{f}_{d,i}^{(u)}$ and $\mathbf{f}_{d,j}^{(v)}$, the vector in $\mathbf{F}_d^{(u)}$ and $\mathbf{F}_d^{(v)}$, are obtained to feed SDAEs. The side information is regarded as a whole $\mathbf{p}_d^{(u)}$ (or $\mathbf{p}_d^{(v)}$), integrated by directly importing to all layers except the output layer. Therefore, considering the user's SDAE, the hidden representation at layer l and the outputs are obtained as

$$\begin{aligned}
\mathbf{h}_{d,l}^{(u)} &= g\left(\mathbf{W}_{d,l}^{(u)} \mathbf{h}_{d,l-1}^{(u)} + \mathbf{z}_{d,l}^{(u)} \tilde{\mathbf{p}}_{d,i}^{(u)} + \mathbf{b}_{d,l}^{(u)}\right) \\
\hat{\mathbf{f}}_{d,i}^{(u)} &= f\left(\mathbf{W}_{d,L_d^{(u)}}^{(u)} \mathbf{h}_{d,L_d^{(u)}}^{(u)} + \mathbf{b}_{d,L_d^{(u)}}^{(u)}\right) \\
\hat{\mathbf{p}}_{d,i}^{(u)} &= f\left(\mathbf{z}_{d,L_d^{(u)}}^{(u)} \mathbf{h}_{d,L_d^{(u)}}^{(u)} + \mathbf{b}_{d,n}^{(u)}\right) \qquad (1)
\end{aligned}$$

where $l \in \{1, 2, \cdots, L_d^{(u)} - 1\}$; $\tilde{\mathbf{p}}_{d,i}^{(u)}$ are the corrupted *side information*; $g(\cdot)$ and $f(\cdot)$ are activation functions for the hidden and output layers; $\mathbf{b}_{d,n}^{(u)}$ is the biases in the output layer for the side information. $\mathbf{f}_{d,i}^{(u)}$ is the input to the first layer and $\hat{\mathbf{f}}_{d,i}^{(u)}$ denotes the output. Similar results can be obtained for the item's SDAE by replacing (u) with (v), i with j.

1) Loss Function: The overall loss function of DTCFGP algorithm consists of the matrix tri-factorization loss, the loss of geometric structure, the reconstruction loss of the side information and the ratings, and the approximation error between deep representations and private latent factors.

The loss of matrix tri-factorization in source and target domains can be expressed as

$$\min_{\boldsymbol{\theta}_m} \mathcal{L}_m = \sum_{d \in \{s,t\}} \left\| \mathbf{C}_d \odot \left(\mathbf{R}_d - \mathbf{U}_d \mathbf{H} \mathbf{V}_d^{\mathrm{T}}\right) \right\|^2, \qquad (2)$$

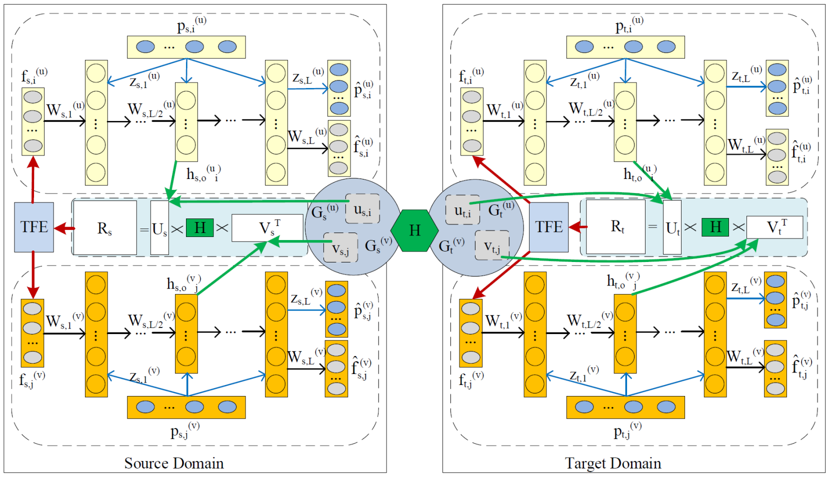


Fig. 1. Structure of the proposed DTCFGP.

where $\boldsymbol{\theta}_m = \{\mathbf{U}_s, \mathbf{V}_s, \mathbf{H}, \mathbf{U}_t, \mathbf{V}_t\}$, the binary matrix $\mathbf{C}_d$ is an indicator of sparsity and $\odot$ is the element-wise operation. Here, $\mathbf{U}_d\mathbf{H}\mathbf{V}_d^{\mathrm{T}}$ can be further written as

$$\begin{aligned}\left[\mathbf{U}_d\mathbf{H}\mathbf{V}_d^{\mathrm{T}}\right]_{ij} &= \left[\mathbf{u}_{d,i}\mathcal{H}_1^c, \mathbf{u}_{d,i}\mathcal{H}_2^c, \cdots, \mathbf{u}_{d,i}\mathcal{H}_{k2}^c\right]\mathbf{v}_{d,j}^{\mathrm{T}} \\ &= \mathbf{u}_{d,i}\left[\mathcal{H}_1^r\mathbf{v}_{d,j}^{\mathrm{T}}, \mathcal{H}_2^r\mathbf{v}_{d,j}^{\mathrm{T}}, \cdots, \mathcal{H}_{k_1}^r\mathbf{v}_{d,j}^{\mathrm{T}}\right], \quad (3)\end{aligned}$$

where $\mathcal{H}_k^c$ with $k \in \{1, 2, \cdots, k_1\}$ denotes the column of $\mathbf{H}$ and $\mathcal{H}_k^r$ with $k \in \{1, 2, \cdots, k_2\}$ denotes the row of $\mathbf{H}$. By defining $\bar{\mathbf{u}}_{d,i} \triangleq \left[\mathbf{u}_{d,i}\mathcal{H}_1^c, \cdots, \mathbf{u}_{d,i}\mathcal{H}_{k_2}^c\right]$ and $\bar{\mathbf{v}}_{d,j}^{\mathrm{T}} \triangleq \left[\mathcal{H}_1^r\mathbf{v}_{d,j}^{\mathrm{T}}, \cdots, \mathcal{H}_{k_1}^r\mathbf{v}_{d,j}^{\mathrm{T}}\right]$, (3) is simplified as

$$\left[\mathbf{U}_d\mathbf{H}\mathbf{V}_d^{\mathrm{T}}\right]_{ij} = \bar{\mathbf{u}}_{d,i}\mathbf{v}_{d,j}^{\mathrm{T}} = \mathbf{u}_{d,i}\bar{\mathbf{v}}_{d,j}^{\mathrm{T}}. \quad (4)$$

Accroding to [24], preserving the users and items' geometric structure in $\mathcal{D}_d$ is to minimize the energy of graphs $G_d^{(u)}$ and $G_d^{(v)}$

$$\begin{aligned}\min_{\boldsymbol{\theta}_m} \mathcal{L}_g &= \frac{1}{2}\sum_d\sum_{ij}\|\mathbf{u}_{d,i} - \mathbf{u}_{d,j}\|^2\left(\mathbf{A}_d^{(u)}\right) \\ &+ \frac{1}{2}\sum_d\sum_{ij}\|\mathbf{v}_{d,i} - \mathbf{v}_{d,j}\|^2\left(\mathbf{A}_d^{(v)}\right) \\ &= \sum_d tr\left(\mathbf{U}_d^{\mathrm{T}}(\mathbf{O}_d^{(u)} - \mathbf{A}_d^{(u)})\mathbf{U}_d\right) \\ &+ \sum_d tr\left(\mathbf{V}_d^{\mathrm{T}}(\mathbf{O}_d^{(v)} - \mathbf{A}_d^{(v)})\mathbf{V}_d\right). \quad (5)\end{aligned}$$

Furthermore, the reconstruction loss at both source and target domains can be expressed as

$$\begin{aligned}\min_{\boldsymbol{\theta}_r} \mathcal{L}_r &= \sum_d \alpha_d \sum_i\left(\mathbf{f}_{d,i}^{(u)} - \hat{\mathbf{f}}_{d,i}^{(u)}\right)^2 \\ &+ \sum_d \beta_d \sum_j\left(\mathbf{f}_{d,j}^{(v)} - \hat{\mathbf{f}}_{d,j}^{(v)}\right)^2 \\ &+ \sum_d (1-\alpha_d)\sum_i\left(\mathbf{p}_{d,i}^{(u)} - \hat{\mathbf{p}}_{d,i}^{(u)}\right)^2 \\ &+ \sum_d (1-\beta_d)\sum_j\left(\mathbf{p}_{d,j}^{(v)} - \hat{\mathbf{p}}_{d,j}^{(v)}\right)^2. \quad (6)\end{aligned}$$

where $\boldsymbol{\theta}_r = \left\{\mathbf{W}_s^{(u)}, \mathbf{b}_s^{(u)}, \mathbf{W}_s^{(v)}, \mathbf{b}_s^{(v)}, \mathbf{W}_t^{(u)}, \mathbf{b}_t^{(u)}, \mathbf{W}_t^{(v)}, \mathbf{b}_t^{(v)}\right\}$; α_d and β_d are penalty parameters.

Besides, the approximation error between deep representations and latent factor vectors can be expressed as

$$\begin{aligned}\min_{\boldsymbol{\theta}_a} \mathcal{L}_a &= \sum_d \rho_d \sum_i\left(\mathbf{u}_{d,i} - \mathbf{h}_{d,o}^{(u_i)}\right)^2 \\ &+ \sum_d \gamma_d \sum_j\left(\mathbf{v}_{d,j} - \mathbf{h}_{d,o}^{(v_j)}\right)^2, \\ &\qquad\qquad\qquad\qquad (7)\end{aligned}$$

where $\boldsymbol{\theta}_a = \left\{\cup_{d\in\{s,t\}}\boldsymbol{\theta}_d\right\}$ with

$$\boldsymbol{\theta}_d \triangleq \left\{\mathbf{U}_d, \mathbf{V}_d, \mathbf{W}_d^{(u)}, \mathbf{b}_d^{(u)}, \mathbf{W}_d^{(v)}, \mathbf{b}_d^{(v)}\right\},$$

ρ_d and γ_d are penalty parameters.

Consequently, the overall loss function of DTCFGP is finally obtained as

$$\min_{\boldsymbol{\Theta}} \mathcal{J} = \mathcal{L}_m + \lambda_g\mathcal{L}_g + \mathcal{L}_r + \mathcal{L}_a + \lambda f_{reg}, \quad (8)$$

where $\boldsymbol{\Theta} = \boldsymbol{\theta}_m \cup \boldsymbol{\theta}_r \cup \boldsymbol{\theta}_a$, and f_{reg} indicates the regularization term that prevents overfitting,

$$\begin{aligned}f_{reg} &= \sum_d\left\{\sum_i\|\mathbf{u}_{d,i}\|^2 + \sum_j\|\mathbf{v}_{d,j}\|^2\right\} \\ &+ \sum_d\left\{\|\mathbf{W}_d^{(u)}\|^2 + \|\mathbf{W}_d^{(v)}\|^2 + \|\mathbf{b}_d^{(u)}\|^2 + \|\mathbf{b}_d^{(v)}\|^2\right\}\end{aligned}$$

and λ is a penalty parameter.

2) Optimization: To solve this problem, an alternative optimization algorithm is considered by utilizing the following three-step procedure.

Step I: Given all weights $\mathbf{W}_d^{(u)}$, $\mathbf{W}_d^{(v)}$, and biases $\mathbf{b}_d^{(u)}$, $\mathbf{b}_d^{(v)}$ in source and target domains, the gradients of the overall loss function $\mathcal{J}$ in (8) with respect to $\mathbf{u}_{d,i}$ and $\mathbf{v}_{d,j}$, $d \in \{s,t\}$, can be obtained as

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{u}_{d,i}} &= -\sum_j c_{d,ij}\left(r_{d,ij} - \mathbf{u}_{d,i}\bar{\mathbf{v}}_{d,j}^{\mathrm{T}}\right)\bar{\mathbf{v}}_{d,j} \\
&\quad +\rho_d\left(\mathbf{u}_{d,i} - \mathbf{h}_{d,o}^{(u_i)}\right) \\
&\quad +\lambda_{reg}\left[\left(\mathbf{O}_d^{(u)} - \mathbf{A}_d^{(u)}\right)\mathbf{U}_d\right]_{i*} + \lambda\mathbf{u}_{d,i}, \\
\frac{\partial \mathcal{J}}{\partial \mathbf{v}_{d,j}} &= -\sum_i c_{d,ij}\left(r_{d,ij} - \bar{\mathbf{u}}_{d,i}\mathbf{v}_{d,j}^{\mathrm{T}}\right)\bar{\mathbf{u}}_{d,j} \\
&\quad +\gamma_d\left(\mathbf{v}_{d,j} - \mathbf{h}_{d,o}^{(v_j)}\right) \\
&\quad +\lambda_{reg}\left[\left(\mathbf{O}_d^{(v)} - \mathbf{A}_d^{(v)}\right)\mathbf{V}_d\right]_{j*} + \lambda\mathbf{v}_{d,j}, \qquad (9)
\end{aligned}
$$

where the binary $c_{d,ij}$ indicates whether the corresponding rating is observed ($= 1$) or not ($= 0$); $[\cdot]_{i*}$ denotes the row i of a matrix, and λ_{reg} is the penalty parameter of the graph. By using coordinate ascent similar to [39], we have

$$
\begin{aligned}
\mathbf{u}_{d,i} &= \left(\bar{\mathbf{V}}_d\mathbf{C}_{d,i}\bar{\mathbf{V}}_d^T + \left(\rho_d + \lambda_{reg}L_{d,ii}^{(u)} + \lambda\right)\mathbf{I}\right)^{-1} \\
&\quad \left(\bar{\mathbf{V}}_d\mathbf{C}_{d,i}\mathbf{R}_{d,i} + \rho_d\mathbf{h}_{d,o}^{(u_i)} + \lambda_{reg}L_{d,io}^{(u)}\mathbf{U}_d\right), \\
\mathbf{v}_{d,j} &= \left(\bar{\mathbf{U}}_d\mathbf{C}_{d,j}\bar{\mathbf{U}}_d^T + \left(\gamma_d + \lambda_{reg}L_{d,jj}^{(v)} + \lambda\right)\mathbf{I}\right)^{-1} \\
&\quad \left(\bar{\mathbf{U}}_d\mathbf{C}_{d,j}\mathbf{R}_{d,j} + \gamma_d\mathbf{h}_{d,o}^{(v_j)} + \lambda_{reg}L_{d,jo}^{(v)}\mathbf{V}_d\right), \qquad (10)
\end{aligned}
$$

with $\bar{\mathbf{U}}_d = [\bar{\mathbf{u}}_{d,i}]_1^{k_1}$, $\bar{\mathbf{V}}_d = [\bar{\mathbf{v}}_{d,j}]_1^{k_2}$, $\mathbf{C}_{d,i} = \mathrm{diag}(c_{i1}, \cdots, c_{ik_2})$, $\mathbf{C}_{d,j} = \mathrm{diag}(c_{1j}, \cdots, c_{k_1 j})$, $\mathbf{R}_{d,i} = (\mathbf{R}_{d,i1}, \cdots, \mathbf{R}_{d,ik_2})^{\mathrm{T}}$, $\mathbf{R}_{d,i} = (\mathbf{R}_{d,1j}, \cdots, \mathbf{R}_{d,k_1 j})^{\mathrm{T}}$; $L_{d,ii}^{(u)} \triangleq \left[\left(\mathbf{O}_d^{(u)} - \mathbf{A}_d^{(u)}\right)\mathbf{U}_d\right]_{ii}$ and $L_{d,io}^{(u)}$ is $\left[\left(\mathbf{O}_d^{(u)} - \mathbf{A}_d^{(u)}\right)\mathbf{U}_d\right]_{i*}$ with the i^{th} element replaced by 0; $L_{d,jj}^{(v)} \triangleq \left[\left(\mathbf{O}_d^{(v)} - \mathbf{A}_d^{(v)}\right)\mathbf{V}_d\right]_{jj}$ and $L_{d,jo}^{(v)}$ is $\left[\left(\mathbf{O}_d^{(v)} - \mathbf{A}_d^{(v)}\right)\mathbf{V}_d\right]_{j*}$ with the j^{th} element replaced by 0.

Step II: Fixed the private latent factors $\mathbf{U}_d$ and $\mathbf{V}_d$, $d \in \{s,t\}$, the update of common latent factors $\mathbf{H}$ can be obtained as

$$
\mathbf{H} \leftarrow \mathbf{H} \odot \sqrt{\frac{\sum_{d\in\{s,t\}}\mathbf{U}_d^{\mathrm{T}}\left(\mathbf{C}_d \odot \mathbf{R}_d\right)\mathbf{V}_d}{\sum_{d\in\{s,t\}}\mathbf{U}_d^{\mathrm{T}}\left(\mathbf{C}_d \odot (\mathbf{U}_d\mathbf{H}\mathbf{V}_d^{\mathrm{T}})\right)\mathbf{V}_d}}, \qquad (11)
$$

where $\odot$ is the element-wise operation defined as above.

Step III: Fixed the private latent factors $\mathbf{U}_d$, $\mathbf{V}_d$ and the common latent factors $\mathbf{H}$, $d \in \{s,t\}$, all weights $\mathbf{W}_d^{(u)}$, $\mathbf{W}_d^{(v)}$, and biases $\mathbf{b}_d^{(u)}$, $\mathbf{b}_d^{(v)}$, of SDAEs in both domains, can be learned by backpropagation with Adam method

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{W}_d^{(u)}} &= -\rho_d\sum_i\left(\mathbf{u}_{d,i} - \mathbf{h}_{d,o}^{(u_i)}\right)\frac{\partial \mathbf{h}_{d,o}^{(u_i)}}{\partial \mathbf{W}_d^{(u)}} \\
&+ \alpha_d\sum_i\left(\mathbf{f}_{d,i}^{(u)} - \hat{\mathbf{f}}_{d,i}^{(u)}\right)\frac{\partial \hat{\mathbf{f}}_{d,i}^{(u)}}{\partial \mathbf{W}_d^{(u)}} + \lambda\mathbf{W}_d^{(u)}, \\
\frac{\partial \mathcal{J}}{\partial \mathbf{W}_d^{(v)}} &= -\gamma_d\sum_j\left(\mathbf{v}_{d,j} - \mathbf{h}_{d,o}^{(v_j)}\right)\frac{\partial \mathbf{h}_{d,o}^{(v_j)}}{\partial \mathbf{W}_d^{(v)}} \\
&+ \beta_d\sum_j\left(\mathbf{f}_{d,j}^{(v)} - \hat{\mathbf{f}}_{d,j}^{(v)}\right)\frac{\partial \hat{\mathbf{f}}_{d,j}^{(v)}}{\partial \mathbf{W}_d^{(v)}} + \lambda\mathbf{W}_d^{(v)} \qquad (12)
\end{aligned}
$$

$\frac{\partial \mathcal{J}}{\partial \mathbf{b}_d^{(u)}}$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{b}_d^{(v)}}$ can be easily obtained by replacing $\mathbf{W}$ with $\mathbf{b}$ in (12).

The weights $\mathbf{z}_d^{(u)}$ for $\mathbf{x}_d^{(u)}$ and $\mathbf{z}_d^{(v)}$ for $\mathbf{x}_d^{(v)}$ can be updated and learned by

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{z}_d^{(u)}} &= -\rho_d\sum_i\left(\mathbf{u}_{d,i} - \mathbf{h}_{d,o}^{(u_i)}\right)\frac{\partial \mathbf{h}_{d,o}^{(u_i)}}{\partial \mathbf{z}_d^{(u)}} \\
&+ (1-\alpha_d)\sum_i\left(\mathbf{p}_{d,i}^{(u)} - \hat{\mathbf{p}}_{d,i}^{(u)}\right)\frac{\partial \hat{\mathbf{p}}_{d,i}^{(u)}}{\partial \mathbf{z}_d^{(u)}} + \lambda\mathbf{z}_d^{(u)}, \\
\frac{\partial \mathcal{J}}{\partial \mathbf{z}_d^{(v)}} &= -\gamma_d\sum_j\left(\mathbf{v}_{d,j} - \mathbf{h}_{d,o}^{(v_j)}\right)\frac{\partial \mathbf{h}_{d,o}^{(v_j)}}{\partial \mathbf{z}_d^{(v)}} \\
&+ (1-\beta_d)\sum_j\left(\mathbf{p}_{d,j}^{(v)} - \hat{\mathbf{p}}_{d,j}^{(v)}\right)\frac{\partial \hat{\mathbf{p}}_{d,j}^{(v)}}{\partial \mathbf{z}_d^{(v)}} + \lambda\mathbf{z}_d^{(v)}. \qquad (13)
\end{aligned}
$$

V. Experiments

A. Dataset

The MovieLens-100K dataset consists of 100K ratings of 943 users and 1682 movies while the MovieLens-1M dataset consists of 1 million ratings of 6040 users and 3706 movies, which are collected from different years [12]. Each rating is an integer in the range of 1 to 5. The ratings are highly sparse, where no ratings occupy 93.7% in MovieLens-100K dataset and 95.8% in MovieLens-1M dataset. The side information for users contains the user's age, gender, occupation and zipcode while the side information for items contains the category of movie genre and release date. The BookCrossing dataset [46] contains 1149780 books from 278858 users, where each rating is an integer from 0 to 10 and no ratings occupy 99.9%. Some attributes of books and users are also provided and being utilized as the side information.

To incorporate the side information in movie recommendation, the side information for users are encoded into a binary valued vector of length 29 in both domains. On the other hand, the side information for items are encoded into a binary valued vector of length 18 in both domains. Similarly, for book recommendation, the side information is encoded into binary vectors for users and items.

We organize MLK(s) vs MLM(t), MLM(s) vs MLK(t) and BC(s) vs MLK(t) as three pairs, where one acts as the relevant domain and the other acts as the target domain. For all compared methods, we train each compared method with

TABLE II
PERFORMANCE COMPARISON IN TERMS OF RMSE.

Algorithm	MLK(s) vs MLM(t)			MLM(s) vs MLK(t)			BC(s) vs MLK(t)		
	60%	80%	95%	60%	80%	95%	60%	80%	95%
NMF	1.0258	1.0127	1.0040	1.0381	1.0276	1.0195	1.0381	1.0276	1.0195
CDL	1.0207	1.0168	0.9984	1.0113	1.0027	0.9871	1.0207	1.0168	0.9984
aSDAE	0.9345	0.9272	0.9222	0.9933	0.9779	0.9702	0.9933	0.9779	0.9702
PMF	0.9204	0.9131	0.9100	0.9590	0.9380	0.9236	0.9590	0.9380	0.9236
RGCMF	0.9173	0.9123	0.9079	0.9585	0.9366	0.9213	0.9614	0.9371	0.9220
CMF	0.9090	0.8857	0.8746	0.9476	0.9232	0.9162	0.9476	0.9232	0.9162
DCF	0.8864	0.8632	0.8571	0.9348	0.9157	0.8981	0.9348	0.9157	0.8981
DTCF	0.8666	0.8527	0.8465	0.9260	0.9104	0.8992	0.9297	0.9109	0.9009
DTCFGP	**0.8640**	**0.8516**	**0.8454**	**0.9195**	**0.9034**	**0.8943**	**0.9190**	**0.9050**	**0.8949**

TABLE III
PERFORMANCE COMPARISON IN TERMS OF MAE.

Algorithm	MLK(s) vs MLM(t)			MLM(s) vs MLK(t)			BC(s) vs MLK(t)		
	60%	80%	95%	60%	80%	95%	60%	80%	95%
NMF	0.8241	0.8207	0.8169	0.8283	0.8249	0.8225	0.8283	0.8249	0.8225
CDL	0.8209	0.8187	0.8116	0.8173	0.8146	0.8061	0.8209	0.8187	0.8116
aSDAE	0.7475	0.7398	0.7347	0.8019	0.7848	0.7765	0.8019	0.7848	0.7765
PMF	0.7619	0.7553	0.7517	0.7903	0.7815	0.7694	0.7903	0.7815	0.7694
RGCMF	0.7232	0.7186	0.7124	0.7741	0.7702	0.7649	0.7843	0.7782	0.768
CMF	0.7214	0.7066	0.6993	0.7876	0.7652	0.7447	0.7876	0.7652	0.7447
DCF	0.7122	0.6918	0.6852	0.7632	0.7407	0.7236	0.7632	0.7407	0.7236
DTCF	0.6799	0.6686	0.6628	0.7247	0.7118	0.7040	0.7273	0.7122	0.7045
DTCFGP	**0.6785**	**0.6683**	**0.6627**	**0.7206**	**0.7073**	**0.7010**	**0.7198**	**0.7084**	**0.7003**

different percentages (60%, 80% and 95%) of ratings. We randomly select the training dataset from the whole dataset, and use the remaining data as the test dataset. We repeat the evaluation five times with different randomly selected training data and the average performance is reported.

B. Evaluation Metric

We employ the root mean squared error (RMSE), the mean absolute error (MAE) and Recall@K as evaluation metrics, which are defined respectively as

1) RMSE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_{\mathcal{T}}} \sum_{R_{ij}^t \in \mathcal{T}} \left(R_{ij}^t - \hat{R}_{ij}^t \right)^2}, \tag{14}$$

2) MAE:

$$\mathrm{MAE} = \frac{\sum_{R_{ij}^t \in \mathcal{T}} \left| R_{ij}^t - \hat{R}_{ij}^t \right|}{N_{\mathcal{T}}}, \tag{15}$$

where R_{ij}^t is the ground-truth rating of user i for item j, $\hat{R}_{ij}^t$ denotes the estimated rating of R_{ij}^t, and $N_{\mathcal{T}}$ is the total number of ratings in the test dataset $\mathcal{T}$.

3) Recall@K:

$$\mathrm{Recall@}K = \frac{\text{Number of Hits @K}}{|GT|}, \tag{16}$$

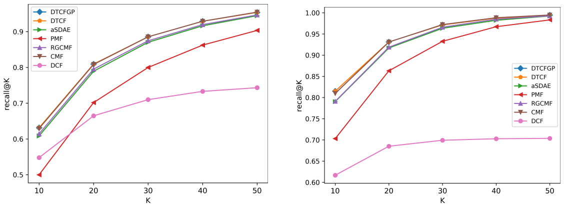
where Number of Hits @K is the number of test items that appear in the recommended list and GT is the ground-truth.

C. Baselines

In order to evaluate the performance of our proposed schemes, we consider the following various methods in our experiments.

- **NMF** - Conventional non-negative matrix factorization method [19];
- **CDL** - Collaborative deep learning [39] is a hierarchical deep Bayesian model to achieve deep representation learning for the item information and collaborative filtering for the user-item matrix.
- **aSDAE** - Additional denoising autoencoder [6] is a single-domain model, where both the side information and raw rating are fused by using an autoencoder.
- **PMF** - Probabilistic matrix factorization [29] is an effective model to factorize the user-item matrix to user and item factors. It assumes there exists Gaussian observation noise and Gaussian priors on the latent factors.
- **RGCMF** - Graph co-regularized collective matrix tri-factorization [24] is a transfer model which preserves the geometric structure in each domain. Revised GCMF (RGCMF) is a revision of the original GCMF. We improve this method by considering the data sparsity, where no side information is considered in the recommendation.
- **CMF** - Collective matrix factorization [33] is a model which simultaneously factorizes multiple sources, including the user-item matrix and matrices containing the additional side information.
- **DCF** - Deep collaborative filtering [20] is a recommendation model which combines PMF with marginalized denoising stacked autoencoders to achieve good recommendation.
- **DTCF** - Deep Transfer Collaborative Filtering [8] is a recommendation model integrating collective matrix factorization and deep transfer learning.
- **DTCFGP** - Our proposed DTCFGP scheme.

For our DTCFGP scheme, we set the parameters α_d and



(a) 60% of ratings

(b) 80% of ratings

Fig. 2. Performance comparison in terms of Recall@K.

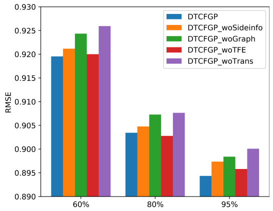


Fig. 3. Ablation Test Results of DTCFGP schemes

β_d as 0.95, the parameters γ_d, ρ_d as 2, the regularization coefficient λ as 0.3, and the weight of graph λ_{reg} as 0.9. We use a masking noise with a noise level of 0.1 to get the corrupted input from the side information. In terms of the SDAE, the number of layers for each encoder or decoder is set to 2 in our experiments. So the total number of layers for the autoencoder is thus equal to 5. Moreover, the dimensionality of learned latent factors is set to 30 for users and 100 for items. The size of the hidden layer is 80 for MovieLens-100K and MovieLens-1M and 120 for BookCrossing.

D. Summary of Experimental Results

1) Performance Comparison: Tables II and III shows respectively the average RMSE and MAE of NMF, CDL, aSDAE, PMF, RGCMF, CMF, DCF, DTCF and our DTCFGP schemes on three pairs of datasets, where the lowest value of each dataset is highlighted in boldface. From Tables II and III, it is observed that CMF, CDL, DCF, DTCF and our DTCFGP schemes achieve a better performance than PMF, indicating the effectiveness of incorporating the side information. Moreover, CDL, DCF, DTCF and our DTCFGP schemes outperform PMF and CMF, indicating that deep structures can admire better feature quality of the side information. Furthermore, it is noticed that our DTCFGP schemes obtain both a lower RMSE and a lower MAE than that of CDL, aSDAE, DCF and DTCF. In the end, DTCFGP is superior to DTCF, which is clearly due to the preserving of the geometric structure. We will further discuss the effect of geometric structure on the results in our model in the ablation study. Meanwhile, the decline becomes significant when the percentage of training data reduces, which validates the effectiveness of cross-domain learning for recommendation.

To sum up, the proposed DTCFGP schemes depict an evident superiority in comparison to the state-of-the-art methods in terms of the RMSE and MAE, which demonstrates the effectiveness of our schemes.

From Tables II and III, it is also observed that the performance on the data pair of BookCrossing vs MovieLens-100k is better than that on the data pair of MovieLens-1M vs MovieLens-100K. This may because the difference between BookCrossing and MovieLens-100K is much larger, so that cross-domain learning can transfer more information from closer domains.

2) Recall@K Analysis: Fig. 2 shows the Recall@K results on MovieLens dataset, where seven superior baselines are compared to our DTCFGP schemes in terms of RMSE. As seen, only the cases with 60% and 80% training data are considered, because in the case of 95% training data, these models performs approximately close to 1, indicating limited information.

Generally speaking, the conclusion drawn from the RMSE and MAE remains essentially unchanged when considering the Recall@K analysis. Moreover, it is observed that DTCFGP, DTCF and CMF have a high overlapping, indicating that they lie in the first class in terms of Recall@K. aSDAE and RGCMF have a large overlapping, implies that they lie in the second class, better than PMF. In both cases, DCF performs worst. The overlapping occurs because the model has a less impact than the structure of the data. Nevertheless, our model outperforms baselines in all cases, i.e. whatever K changes in a wide range of 10 to 50.

3) Ablation Study: To justify the effectiveness of our architecture design, a careful ablation study is conducted. Specifically, our model combines matrix tri-factorization with a shared weight matrix H and deep structure in both source and target domains. They are integrated to incorporate both cross-domain information and the side information.

we replace the side information in deep structure with the results of TFE block and name it as *DTCFGP_woSideinfo*; Specifically, we remove the graph structure and name it as *DTCFGP_woGraph*; we remove the knowledge transfer and name it as *DTCFGP_woTrans*; correspondingly, we remove the TFE block and name it as *DTCFGP_woTFE*.

Test results on movie and book datasets in terms of RMSE are shown in Fig. 3, where a couple of observations are worth being highlighted as follows

- In most cases, the removal of arbitrary component in DTCFGP schemes causes the performance drop, indicating that each component in DTCFGP is indispensable.
- Knowledge transfer also has a considerable influence on the performance improvement except the minority cases, indicating that the transferred knowledge from relevant domains is generally important. In Fig. 3, the transferred knowledge becomes the primary source of rich information because the TFE block misses providing abundant information.

Evaluation results using RMSE, MAE and Recall@K, and ablation analysis in experiments show that 1) DTCFGP outperforms baselines; 2) deep structure, the preservation of graph structure and knowledge transfer across domains indeed improve the performance; and 3) geometric preservation is of importance to domain transfer. Strictly speaking, please be noted that only movie vs book pair transfers across domains while other two pairs using movie datasets do not really owing to high relevance; *nevertheless, experiments demonstrate that DTCFGP works well for all cases.*

VI. Conclusion

Cross-domain deep collaborative filtering is proposed for recommendations by referring to the knowledge in relevant domains, where the loss of data geometric structure is minimized in the process of knowledge transfer to guarantee to predict labels smoothly. Non-negative matrix tri-factorization is integrated with deep structure in both source and target domains, where common latent factors construct a bridge between domains. The geometric structure is modeled by two designed graphs in source and target domains. Effective latent representations of users and items are learned by jointly optimizing the matrix tri-factorization, SDAEs and geometric structure. Extensive experimental results on movie and book datasets show that our proposed approach achieves a better performance in comparison to state-of-the-art related works, in terms of multiple evaluation metrics and ablation analysis.

References

[1] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *NIPS*, 2016, pp. 343–351.

[2] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *IJCAI*, 2009, pp. 1010–1015.

[3] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu, "Svdfeature: A toolkit for feature-based collaborative filtering," *JMLR*, vol. 13, no. 1, pp. 3619–3622, 2012.

[4] Y. Chen, H. Zhang, J. Wu, X. Wang, R. Liu, and M. Lin, "Modeling emerging, evolving and fading topics using dynamic soft orthogonal nmf with sparse representation," in *ICDM*, 2015, pp. 61–70.

[5] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *KDD*. ACM, 2006, pp. 126–135.

[6] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *AAAI*. AAAI Press, 2017, pp. 1309–1315.

[7] G. K. Dziugaite and D. M. Roy, "Neural network matrix factorization," *CoRR*, vol. abs/1511.06443, 2015.

[8] S. Gai, F. Zhao, Y. Kang, Z. Chen, D. Wang, and A. Tang, "Deep transfer collaborative filtering for recommender systems," in *PRICAI*. Springer, 2019, pp. 515–528.

[9] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *KDD*, 2009, pp. 359–368.

[10] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1087–1099, 2012.

[11] S. K. Gupta, D. Q. Phung, B. Adams, T. Tran, and S. Venkatesh, "Nonnegative shared subspace learning and its application to social media retrieval," in *KDD*, 2010, pp. 650–658.

[12] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *TIIS*, vol. 5, no. 4, p. 19, 2016.

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*. IW3C2, 2017, pp. 173–182.

[14] G. Hu, Y. Zhang, and Q. Yang, "Mtnet: A neural approach for cross-domain recommendation with unstructured text," in *KDD Deep Learning Day*, 2018, pp. 1–10.

[15] H. Kanagawa, H. Kobayashi, N. Shimizu, Y. Tagami, and T. Suzuki, "Cross-domain recommendation via deep domain adaptation," in *ECIR*. Springer, 2019, pp. 20–29.

[16] Y. Kang, S. Gai, F. Zhao, D. Wang, and Y. Luo, "Cross-domain deep collaborative filtering for recommendation," in *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2019, pp. 634–638.

[17] Y.-D. Kim and S. Choi, "Scalable variational bayesian matrix factorization with side information," in *AISTATS*, 2014, pp. 493–502.

[18] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer Journal*, pp. 30–37, 2009.

[19] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001, pp. 556–562.

[20] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *CIKM*, 2015, pp. 811–820.

[21] T. Li, Y. Ma, J. Xu, B. Stenger, C. Liu, and Y. Hirate, "Deep heterogeneous autoencoders for collaborative filtering," in *ICDM*, 2018, pp. 1164–1169.

[22] X. Ling, W. Dai, G. R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *KDD*, 2008, pp. 488–496.

[23] M. Long, W. Cheng, X. Jin, J. Wang, and D. Shen, "Transfer learning via cluster correspondence inference," in *ICDM*, 2011, pp. 917–922.

[24] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, "Transfer learning with graph co-regularization," in *AAAI*, 2012, pp. 1033–1039.

[25] S. Park, Y. D. Kim, and S. Choi, "Hierarchical bayesian matrix factorization with side information," in *IJCAI*, 2013, pp. 1593–1599.

[26] I. Porteous, A. U. Asuncion, and M. Welling, "Bayesian matrix factorization with side information and dirichlet process mixtures," in *AAAI*, 2010, pp. 563–568.

[27] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *ICML*, 2005, pp. 713–719.

[28] R. Salakhutdinov, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *ICML*, 2008, pp. 880–887.

[29] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS*, 2008, pp. 1257–1264.

[30] R. Salakhutdinov, A. Mnih, and G. E. Hinton, "Restricted boltzmann machines for collaborative filtering," in *ICML*, 2007, pp. 791–798.

[31] J. Shi, M. Long, Q. Liu, G. Ding, and J. Wang, "Twin bridge transfer learning for sparse collaborative filtering," in *PAKDD*. Springer, 2013, pp. 496–507.

[32] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix:a survey of the state of the art and future challenges," *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45, 2014.

[33] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *KDD*. ACM, 2008, pp. 650–658.

[34] ——, "A bayesian matrix factorization model for relational data," in *UAI*. AUAI Press, 2010, pp. 556–563.

[35] P. Sinno Jialin, I. W. Tsang, J. T. Kwok, and Y. Qiang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[36] T. T. Truyen, D. Q. Phung, and S. Venkatesh, "Ordinal boltzmann machines for collaborative filtering," in *UAI*, ser. UAI '09. Arlington, Virginia, United States: AUAI Press, 2009, pp. 548–556. [Online]. Available: http://dl.acm.org/citation.cfm?id=1795114.1795178

[37] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *IJCAI*, 2009, pp. 1273–1278.

[38] ——, "Heterogeneous domain adaptation using manifold alignment," in *IJCAI*, 2011, pp. 1541–1546.

[39] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *KDD*. ACM, 2015, pp. 1235–1244.

[40] H. Wang, H. Huang, F. Nie, and C. H. Q. Ding, "Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization," in *SIGIR*, 2011, pp. 933–942.

[41] X. Xu, F. Shen, Y. Yang, D. Zhang, H. T. Shen, and J. Song, "Matrix tri-factorization with manifold regularizations for zero-shot learning," in *CVPR*, 2017, pp. 2007–2016.

[42] S. Yang, C. Zhang, and Y. Wu, "Robust non-negative matrix factorization via joint sparse and graph regularization for transfer learning," *Neural Computing and Applications*, vol. 23, no. 2, pp. 541–559, 2013.

[43] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[44] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong, "Mining distinction and commonality across multiple domains using generative model for text classification," *TKDE*, vol. 24, no. 11, pp. 2025–2039, 2012.

[45] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi, "Exploiting associations between word clusters and document classes for cross-domain text categorization," in *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2011, pp. 100–114.

[46] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th International Conference on World Wide Web*, ser. WWW '05. New York, NY, USA: ACM, 2005, pp. 22–32. [Online]. Available: http://doi.acm.org/10.1145/1060745.1060754