Deep Transfer Tensor Decomposition with Orthogonal Constraint for Recommender Systems

Zhengyu Chen[1,2], **Ziqing Xu**[3], **Donglin Wang**[*2]

[1] College of Computer Science & Technology, Zhejiang University
[2] Machine Intelligence Lab (MiLAB), AI Division, School of Engineering, Westlake University
[3] Department of Statistics, University of Chicago
chenzhengyu@zju.edu.cn, simonxu@uchicago.edu, {chenzhengyu,wangdonglin}@westlake.edu.cn

Abstract

Tensor decomposition is one of the most effective techniques for *multi-criteria* recommendations. However, it suffers from data sparsity when dealing with three-dimensional (3D) user-item-criterion ratings. To mitigate this issue, we consider effectively incorporating the side information and cross-domain knowledge in tensor decomposition. A deep transfer tensor decomposition (DTTD) method is proposed by integrating deep structure and Tucker decomposition, where an *orthogonal constrained* stacked denoising autoencoder (OC-SDAE) is proposed for alleviating the scale variation in learning effective latent representation, and the side information is incorporated as a compensation for tensor sparsity. Tucker decomposition generates users and items' latent factors to connect with OC-SDAEs and creates a common core tensor to bridge different domains. A cross-domain *alignment* algorithm (CDAA) is proposed to solve the rotation issue between two core tensors in source and target domain. Experiments show that DTTD outperforms state-of-the-art related works.

Introduction

With the data explosion in recent years, recommendations are becoming increasingly attractive. Traditional single-criterion recommendation typically operates on two-dimensional (2D) user-item ratings (Gai et al. 2019; Xiao, Liang, and Meng 2019a). In single-criterion recommendation, there are two primary categories of algorithms: content-based methods and collaborative filtering (CF) based methods, where matrix factorization is effective in learning effective latent factors for users and items (Xiao, Liang, and Meng 2019b; Xiao et al. 2019). However, they cannot work well for multi-criteria recommendations that contain multiple criterion-specific ratings. With the emergence of multimodal data, multi-criteria recommendation becomes more important(Lakiotaki, Matsatsinis, and Tsoukias 2011). Figure 1 shows an example in TripAdvisor: customers rate hotels using multiple criteria (in red) such as value, service, atmosphere, food and overall, where the side information of customers and hotels is provided (in blue).

Prior multi-criteria techniques can be briefly classified into three categories: heuristic neighborhood-based approaches (Lakiotaki, Matsatsinis, and Tsoukias 2011), aggregation-based approaches (Lakiotaki, Tsafarakis, and Matsatsinis 2008), and model-based approaches (Sahoo et al. 2012). Heuristic neighborhood-based approaches attempt to use various multi-criteria similarity metrics to collect the neighbors of a targeted user, and then estimate unknown ratings based on the known ratings of those neighbors (Mikeli, Apostolou, and Despotis 2013). Aggregation-based approaches aim to build a mapping to aggregate multiple criterion-specific ratings by assuming that there is a certain relation between the overall rating and other criterion-specific ratings (Lakiotaki, Tsafarakis, and Matsatsinis 2008). Model-based approaches learn a model by leveraging the observed multi-criteria ratings and then employing the model to execute prediction (Sahoo et al. 2012).

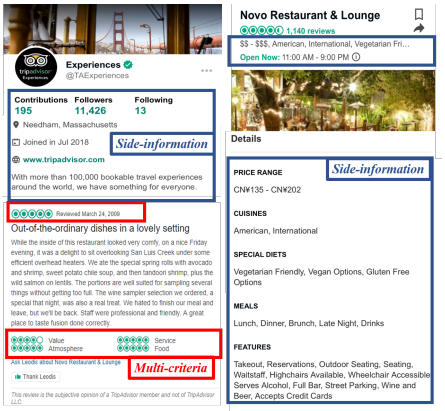


Figure 1: Multi-criteria ratings and the side information

Tensor decomposition is a milestone of model-based techniques. Many related techniques have been developed for multi-criteria recommendations (Bhargava et al. 2015; Yao et al. 2015), but all these suffer from sparsity problem. Although single-criterion recommendations using matrix factorization consider to absorb the side information and the knowledge from relevant domains to enrich priors and confront sparsity (Chen, Wang, and Yin 2021; Dong et al. 2017), *no prior work* in multi-criteria recommendations incorporates such information into tensor decomposition.

In this paper, we integrate the side information and knowl-

*Corresponding author.

edge transfer into Tucker decomposition to solve the sparsity issue in multi-criteria recommendations. The key *challenges* are as follows: 1) How to transfer knowledge from source to target domain in cross-domain Tucker decomposition? 2) How to make sure that effective latent representations learned via deep structure in different domains have an approximately identical scale? 3) And how to overcome the rotation between two core tensors that are decomposed respectively in source and target domains?

In this paper, we propose a deep transfer tensor decomposition (DTTD) scheme. To overcome the first challenge, the core tensor in tucker decomposition is taken as a bridge to transfer knowledge from source to target domain. Previous works (Malik and Becker 2018) show that the Tucker decomposition is generally not unique. This intuitively follows from the fact that the core tensor can be arbitrarily structured and might allow interactions between any component. Imposing orthogonal constraint can therefore lead to more relaxed uniqueness properties. Moreover, we propose an *orthogonal constrained* stacked denoising autoencoder (OC-SDAE) to learn effective latent representation.For the third challenge, a cross-domain *alignment* algorithm (CDAA) is proposed to rotate the core tensor in target domain to accomplish alignment with that in source domain. In DTTD, Tucker decomposition in each of domains generates private users and items' latent factors that are used to connect with two OC-SDAEs, and learns a common core tensor to connect different domains. To the best of our knowledge, no prior work incorporates the side information and cross-domain knowledge in Tucker decomposition based multi-criteria recommendations. The contribution of this paper can be summarized as follows: 1) To solve data sparsity problem in multi-criteria ratings, we propose DTTD to integrate deep structure and cross-domain Tucker decomposition; 2) OC-SDAE is proposed to learn effective latent representation that has a small scale variation; CDAA is proposed to rotate the core tensor in target domain to accomplish alignment with that in source domain; 3) An alternative optimization algorithm is proposed because joint-optimization is unavailable in this case; 4) Experiment results on three datasets demonstrate the effectiveness of our proposed DTTD.

Related Work

Multi-criteria recommendation has been studied over decades and can be briefly grouped into three categories: heuristic neighborhood-based approaches (Adomavicius and Kwon 2007; Lakiotaki, Matsatsinis, and Tsoukias 2011; Mikeli, Apostolou, and Despotis 2013), aggregation-based approaches (Lakiotaki, Tsafarakis, and Matsatsinis 2008; Jannach, Karakaya, and Gedikli 2012), and model-based approaches (Sahoo et al. 2012). Model-based approaches aim to learn a predictive model and then employ the model to estimate the ratings. Many techniques have been proposed for recommendations, including a probabilistic mixture algorithm (Sahoo et al. 2012), an adaptive neuro-fuzzy inference and self-organizing map clustering (Nilashi, Ibrahim, and Ithnin 2014), and a multi-linear singular value decomposition (Li, Wang, and Geng 2008).

Tensor decomposition: This is a milestone of model-based approaches and various variants are developed for wide applications (Lakiotaki, Matsatsinis, and Tsoukias 2011). A tensor decomposition based ranking is presented in (Rendle et al. 2009) to predict tags for users. Based on tensor decomposition, Bhargava et al. (Bhargava et al. 2015) tackle context-aware collaborative recommendation by tensor while Yao et al. (Yao et al. 2015) present an application in point-of-interest recommendations. And Zhang et al.(Zhang and Aeron 2017) present a tensor SVD (t-SVD) that can perfectly recover a tensor with low tubal-rank under certain standard incoherent condition.

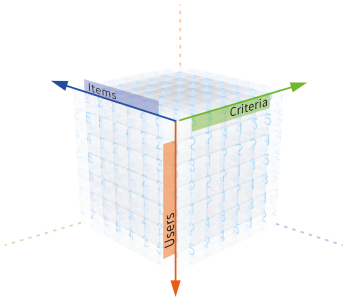


Figure 2: Rating tensor

Deep learning: (Hamada and Hassan 2018) uses networks to learn an aggregation function in multi-criteria recommendation. Its training process is accomplished via particle swarm optimization. Tallapally et al. (Tallapally et al. 2018) utilize an autoencoder to learn the relationship between criteria and overall rating. Chen et al. (Chen, Gai, and Wang 2019) propose a CP factorization based model which combines side information by integrating deep representation learning. However, these works only consider the single-domain multi-criteria ratings.

Preliminary

Problem Definition. This paper aims to cope with the data sparsity problem in multi-criteria recommendation and the rotation problem between two core tensors of source and target domains. Based on various specific criterion, multi-criteria recommender systems are to leverage multiple categories of ratings to make recommendation (Jannach, Zanker, and Fuchs 2014; Lakiotaki, Matsatsinis, and Tsoukias 2011). Figure 3 shows an example of a 3D *user-item-criterion* rating tensor, where each user rates on various criterion of a given item, and the mark "?" means an unobserved rating. Each rating, r_{ijl}, corresponds to the case that the user i rates on the criterion l of item j. Given the sparse *user-item-criterion* rating tensor, the goal is to simultaneously learn user latent factor, item latent factor and criteria latent factor, and finally predict the unobserved ratings.

Tucker Decomposition. For convenience, define by $\mathcal{D}_s$ source domain and $\mathcal{D}_t$ target domain. *And the domain indices are denoted as* $d \in \{s, t\}$. In a recommendation setting, the user-item-criterion matrix $\mathbf{R}_d \in \mathbb{R}^{I_d \times J_d \times L}$ can be decomposed as a sum of rank-1 tensors across all users as in Figure 3. So we have

$$arg \min_{\mathscr{G}, \mathbf{U}_d, \mathbf{V}_d, \mathbf{C}} \|\mathbf{R}_d - \mathscr{G} \times_1 \mathbf{U}_d \times_2 \mathbf{V}_d \times_3 \mathbf{C}_d\|^2, \quad (1)$$

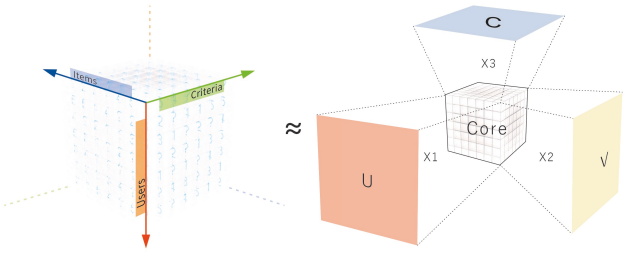


Figure 3: Tucker decomposition

where $\mathbf{U}_d \in \mathbb{R}^{I_d \times f_1}$, $\mathbf{V}_d \in \mathbb{R}^{J_d \times f_2}$ and $\mathbf{C} \in \mathbb{R}^{L \times f_3}$ represent the latent factor matrix for users, items and criteria, respectively; $\mathscr{G} \in \mathbb{R}^{f_1 \times f_2 \times f_3}$ is the *core tensor* with showing the level of interaction between different components; f is the dimension of latent factor space; and the operator $\times_n$ is the *n-mode (matrix) product* of a tensor with a matrix.

Specifically, $\mathscr{G} \times_n \mathscr{M}$ indicates the product of a tensor $\mathscr{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \cdots \times I_n}$ with a matrix $\mathscr{M} \in \mathbb{R}^{J \times I_n}$, where the resulting size becomes $I_1 \times I_2 \times I_3 \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. Elementwise, we have

$$(\mathscr{G} \times_n \mathscr{M})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} g_{i_1 i_2 \cdots i_N} m_{j i_n}. \quad (2)$$

The source domain $\mathcal{D}_s$ is connected with the target domain $\mathcal{D}_t$ via common core tensor $\mathscr{G}$.

Methodology

OC-SDAE. The difference between OC-SDAE and conventional SDAE lies in the *orthogonal constraint* in the loss function. In DTTD, the OC-SDAE takes as input the side information. Considering the OC-SDAE for users in Figure 4, the representation $h_{d,l}^{(u)}$ at each hidden layer and the output at layer $L^{(u)}$ can be obtained as

$$\begin{aligned} \mathbf{h}_{d,l}^{(u)} &= g\left(\mathbf{W}_{d,l}^{(u)} \mathbf{h}_{d,l-1}^{(u)} + \mathbf{b}_{d,l}^{(u)}\right), \\ \hat{\mathbf{p}}_{d,i}^{(u)} &= f\left(\mathbf{W}_{d,L^{(u)}}^{(u)} \mathbf{h}_{d,L^{(u)}}^{(u)} + \mathbf{b}_{d,L^{(u)}}^{(u)}\right), \end{aligned} \quad (3)$$

where $l \in \{1, 2, \cdots, L_d^{(u)} - 1\}$; $g(\cdot)$ and $f(\cdot)$ are activation functions for the hidden and output layers. The corrupted side information $\tilde{\mathbf{p}}_{d,i}^{(u)}$ is the input to the first layer, $\mathbf{h}_{d,r}^{(u_i)}$ denotes deep representations from the middle layer and $\hat{\mathbf{p}}_{d,i}^{(u)}$ denotes the output of the users' OC-SDAE. OC-SDAE utilizes the orthogonal constraint to alleviate the scale variation of effective latent factors $\mathbf{h}_{d,r}^{(u_i)}$ at the middle layer. By considering the orthogonal constraint, OC-SADE reconstructs the input signal using a novel loss function

$$\min_{\mathbf{W}_d^u, \mathbf{b}_d^u} \mathcal{L}_{d,o}^{(u)} = \sum_i \left(\mathbf{p}_{d,i}^{(u)} - \hat{\mathbf{p}}_{d,i}^{(u)}\right)^2 + \lambda_o \left\| I - \mathbf{h}_{d,r}^{(u)^T} \mathbf{h}_{d,r}^{(u)} \right\|^2, \quad (4)$$

where I is the identity matrix. Similar results can be obtained for the items' OC-SDAE by replacing (u) with (v). In Figure 4, the users' (or items') OC-SDAE takes as input the side information of users (or items) to learn the latent representation $\mathbf{h}_{d,r}^{(u_i)}$ (or $\mathbf{h}_{d,r}^{(v_j)}$) that is used to compensate latent factor vectors $\mathbf{u}_{d,i}$ (or $\mathbf{v}_{d,j}$) in tensor decomposition.

Cross-Domain Decomposition with CDAA. Cross-domain Tucker decomposition takes the core tensor as a bridge to connect different domains. However, as shown in Figure 5, which looks like a part of Figure 4 but emphasizes more on the misalignment issue, it is likely that users (or items) with the similar preference in source and target domains are placed in different position of rating tensor. In such case, the knowledge transfer through the core tensor will lead to a misalignment problem. Therefore, we define an orthogonal transformation matrix in the target domain to align with the core tensor in the source domain. In order to solve this problem, the CDAA is proposed by considering a rotation of core tensor in the target domain. With CDAA, the loss of cross-domain Tucker decomposition on a rating tensor is

$$\begin{aligned} \min_{\boldsymbol{\theta}_t} \mathcal{L}_t &= \sum_d \|\mathbf{Z_d} \odot (\mathbf{R}_d - \mathscr{G} \times_1 \mathbf{U}_d \times_2 \mathbf{V}_d \times_3 \mathbf{C}_d)\|^2 \\ &\quad + \left\|\mathbf{U}_s - \mathbf{U}_t \mathbf{O}^{(u)}\right\|^2 + \left\|\mathbf{V}_s - \mathbf{V}_t \mathbf{O}^{(v)}\right\|^2, \\ \text{s. t.} &\quad \left(\mathbf{O}^{(u)}\right)^T \mathbf{O}^{(u)} = I, \left(\mathbf{O}^{(v)}\right)^T \mathbf{O}^{(v)} = I, \end{aligned} \quad (5)$$

where $\boldsymbol{\theta}_t = \left\{\mathbf{U_d}, \mathbf{V_d}, \mathbf{C_d}, \mathscr{G}, \mathbf{O}^{(u)}, \mathbf{O}^{(v)}\right\}$; the binary tensor $\mathbf{Z}_d \in \mathbb{R}^{I_d \times J_d \times L_d}$ is an indicator of sparsity, in which each element indicates whether the corresponding rating is observed ($= 1$) or not ($= 0$); $\odot$ is the element-wise production; and I is identity matrix.

Loss of DTTD. DTTD learns users, items and criteria' latent factor through the following objective function

$$\min_{\boldsymbol{\Theta}} \mathcal{J} = \frac{1}{2}\left(\mathcal{L}_t + \mathcal{L}_r + \mathcal{L}_a + \lambda f_{reg}\right), \quad (6)$$

where the overall loss function $\mathcal{J}$ consists of the loss of tensor decomposition $\mathcal{L}_t$, the reconstruction loss of the side information $\mathcal{L}_r$, the approximation error between deep representation and latent factors $\mathcal{L}_a$, and the regularization term f_{reg} preventing overfitting. Firstly, the loss of Tucker decomposition on a sparse rating tensor is given by Eq. (5). Secondly, by using the proposed OC-SDAE, the reconstruction loss of the side information for both users and items is

$$\min_{\boldsymbol{\theta}_r = \{\mathbf{W}_d^u, \mathbf{b}_d^u, \mathbf{W}_d^v, \mathbf{b}_d^v\}} \mathcal{L}_r = \sum_d \left(\alpha_d \mathcal{L}_{d,o}^{(u)} + \beta_d \mathcal{L}_{d,o}^{(v)}\right), \quad (7)$$

where $\boldsymbol{\theta}_r = \{\mathbf{W}_d^u, \mathbf{b}_d^u, \mathbf{W}_d^v, \mathbf{b}_d^v\}$, α_d and β_d are penalty parameters. Furthermore, the approximation error of deep representation and latent factor vector for users and items is

$$\begin{aligned} \min_{\boldsymbol{\theta}_a} \mathcal{L}_a &= \sum_d \rho_d \sum_i \left(\mathbf{u}_{d,i} - \mathbf{h}_{d,r}^{(u_i)}\right)^2 \\ &\quad + \sum_d \gamma_d \sum_j \left(\mathbf{v}_{d,j} - \mathbf{h}_{d,r}^{(v_j)}\right)^2, \end{aligned} \quad (8)$$

where $\boldsymbol{\theta}_a = \left\{\mathbf{U}_d, \mathbf{V}_d, \mathbf{W}_d^{(u)}, \mathbf{b}_d^{(u)}, \mathbf{W}_d^{(v)}, \mathbf{b}_d^{(v)}\right\}$, ρ_d and γ_d are penalty parameters; $\mathbf{h}_{d,r}^{(u_i)}$ and $\mathbf{h}_{d,r}^{(v_j)}$ are latent representation of users and items' side information, which is extracted through the OC-SDAE.

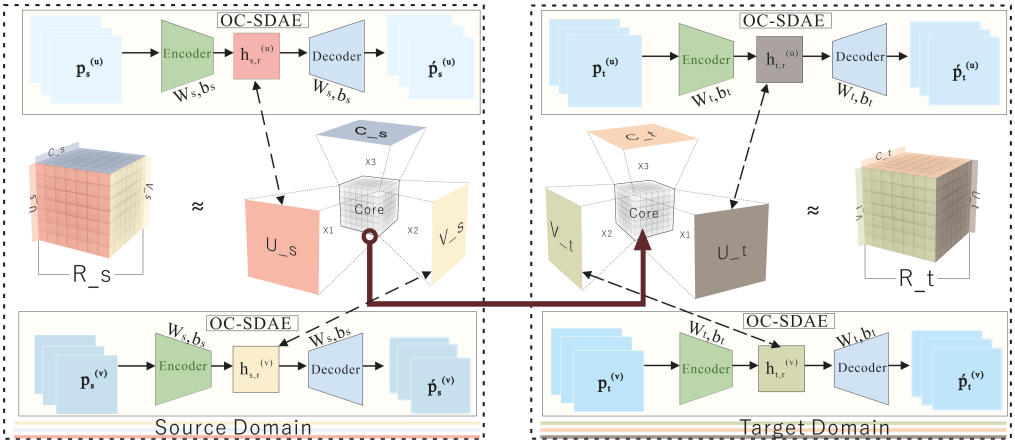


Figure 4: Block diagram of the proposed DTTD

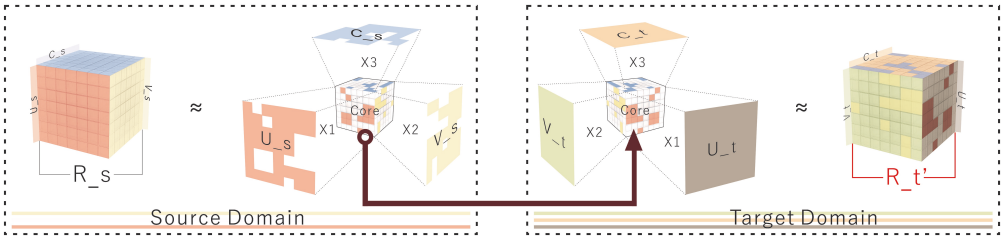


Figure 5: Misalignment in cross-domain decomposition

The last term denotes the regularization term f_{reg} as

$$f_{reg} = \sum_d \left(\sum_i \|\mathbf{u}_{d,i}\|^2 + \sum_j \|\mathbf{v}_{d,j}\|^2 \right) + \sum_d \left(\|\mathbf{W}_d^{(u)}\|^2 + \|\mathbf{W}_d^{(v)}\|^2 + \|\mathbf{b}_d^{(u)}\|^2 + \|\mathbf{b}_d^{(v)}\|^2 \right), \tag{9}$$

Optimization of DTTD. Since the latent matrices U and V are coupled with deep representation, the gradient of core tensor cannot be derived so the joint-training is unavailable in this case. Because the core tensor is decoupled from other components, it can be solved using an alternative optimization (Malik and Becker 2018).

To minimize $\mathcal{J}$, we propose an alternative optimization algorithm that utilizes the following four-step procedure.

Step I: Given all weights $\mathbf{W}_d$ and biases $\mathbf{b}_d$, the gradients of $\mathcal{J}$ in (6) with respect to $\mathbf{u}_{d,i}$, $\mathbf{v}_{d,j}$, $\mathbf{c}_{d,l}$ can be obtained as

$$\begin{aligned}
\widehat{r_{d,ijl}} &= \mathbf{g} \times_1 \mathbf{u}_{d,i} \times_2 \mathbf{v}_{d,j} \times_3 \mathbf{c}_{d,l}, \\
\frac{\partial \mathcal{J}}{\partial \mathbf{u}_{d,i}} &= -\sum_j \sum_l z_{d,ijl} \left(r_{d,ijl} - \widehat{r_{d,ijl}} \right) + \lambda \mathbf{u}_{d,i} \\
&\quad * (\mathbf{g} \times_2 \mathbf{v}_{d,j} \times_3 \mathbf{c}_{d,l}) + \rho_d \left(\mathbf{u}_{d,i} - \mathbf{h}_{d,r}^{(u_i)} \right), \\
\frac{\partial \mathcal{J}}{\partial \mathbf{v}_{d,j}} &= -\sum_i \sum_l z_{d,ijl} \left(r_{d,ijl} - \widehat{r_{d,ijl}} \right) + \lambda \mathbf{v}_{d,j} \\
&\quad * (\mathbf{g} \times_1 \mathbf{u}_{d,i} \times_3 \mathbf{c}_{d,l}) + \gamma_d \left(\mathbf{v}_{d,j} - \mathbf{h}_{d,r}^{(v_j)} \right), \\
\frac{\partial \mathcal{J}}{\partial \mathbf{c}_{d,l}} &= -\sum_i \sum_j z_{d,ijl} \left(r_{d,ijl} - \widehat{r_{d,ijl}} \right) + \lambda \mathbf{c}_{d,l} \\
&\quad * (\mathbf{g} \times_1 \mathbf{u}_{d,i} \times_2 \mathbf{v}_{d,j}),
\end{aligned} \tag{10}$$

where the binary $z_{d,ijl}$ indicates whether the corresponding rating is observed (=1) or not (=0).

Step II - Update from CDAA: The transformation matrix is obtained by solving the following optimization problem

$$\begin{aligned}
\mathbf{O}^{(u)} &= \arg\min_{\mathbf{O}^{(u)}} \left\| \mathbf{U}_s - \mathbf{U}_t \mathbf{O}^{(u)} \right\|^2 \\
\text{s. t.} \quad & \left(\mathbf{O}^{(u)} \right)^T \mathbf{O}^{(u)} = I.
\end{aligned} \tag{11}$$

Update user's latent representation $\mathbf{U}_t$ in target domain as $\tilde{\mathbf{U}}_t = \mathbf{U}_t \mathbf{O}^{(u)}$. Similarly, we can update item's latent representation $\mathbf{V}_t$ in the target domain as $\tilde{\mathbf{V}}_t = \mathbf{V}_t \mathbf{O}^{(v)}$.

Step III: Fixed the private latent factors $\mathbf{U_d}$, $\mathbf{V_d}$ and $\mathbf{C_d}$, the core tensor $\mathscr{G}$ can be updated as

$$\begin{aligned}
\hat{\mathbf{R}}_\mathbf{d} &= (\mathscr{G} \times_1 \mathbf{U_d} \times_2 \mathbf{V_d} \times_3 \mathbf{C_d}) \\
\mathscr{G} &\leftarrow \mathscr{G} \odot \sqrt{\frac{\sum_{d \in (s,t)} \mathbf{U_d}^{\mathrm{T}} (\mathbf{Z_d} \odot \mathbf{R_d}) \mathbf{V_d}^{\mathrm{T}} \mathbf{C_d}^{\mathrm{T}}}{\sum_{d \in (s,t)} \mathbf{U_d}^{\mathrm{T}} \left(\mathbf{Z_d} \odot \hat{\mathbf{R}}_\mathbf{d} \right) \mathbf{V_d}^{\mathrm{T}} \mathbf{C_d}^{\mathrm{T}}}}.
\end{aligned} \tag{12}$$

Step IV: Fixed the latent factor matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{C}$, all weights and biases in OC-SDAEs can be learned using stochastic gradient decent (SGD) method. So, we have

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}_d^{(u)}} = -\rho_d \sum_i \left(\mathbf{u}_{d,i} - \mathbf{h}_{d,r}^{(u_i)} \right) \frac{\partial \mathbf{h}_{d,r}^{(u_i)}}{\partial \mathbf{W}_d^{(u)}}$$

$$+ \quad \alpha_d \sum_i \left(\mathbf{p}_{d,i}^{(u)} - \hat{\mathbf{p}}_{d,i}^{(u)} \right) \frac{\partial \hat{\mathbf{p}}_{d,i}^{(u)}}{\partial \mathbf{W}_d^{(u)}} + \lambda \mathbf{W}_d^{(u)}$$

$$- \quad \underbrace{\lambda_1 \sum_i \left(I - \mathbf{h}_{d,r}^{(u_i)^T} \mathbf{h}_{d,r}^{(u_i)} \right) \frac{\partial \mathbf{h}_{d,r}^{(u_i)}}{\partial \mathbf{W}_d^{(u)}}}_{\text{Effect of user's orthogonal constraint}},$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}_d^{(v)}} = -\gamma_d \sum_j \left(\mathbf{v}_{d,j} - \mathbf{h}_{d,r}^{(v_j)} \right) \frac{\partial \mathbf{h}_{d,r}^{(v_j)}}{\partial \mathbf{W}_d^{(v)}}$$

$$+ \quad \beta_d \sum_j \left(\mathbf{p}_{d,j}^{(v)} - \hat{\mathbf{p}}_{d,j}^{(v)} \right) \frac{\partial \hat{\mathbf{p}}_{d,j}^{(v)}}{\partial \mathbf{W}_d^{(v)}} + \lambda \mathbf{W}_d^{(v)}$$

$$- \quad \underbrace{\lambda_2 \sum_i \left(I - \mathbf{h}_{d,r}^{(v_i)^T} \mathbf{h}_{d,r}^{(v_i)} \right) \frac{\partial \mathbf{h}_{d,r}^{(v_i)}}{\partial \mathbf{W}_d^{(v)}}}_{\text{Effect of item's orthogonal constraint}}, \qquad (13)$$

where $\lambda_1 = \alpha_d * \lambda_o$, and $\lambda_2 = \beta_d * \lambda_o$. $\frac{\partial \mathcal{J}}{\partial \mathbf{b}_d^{(u)}}$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{b}_d^{(v)}}$ can be easily obtained by replacing $\mathbf{W}_d$ with $\mathbf{b}_d$ in (13). Iterate four steps above until convergence.

Experiments

Datasets: To evaluate our proposed algorithm, we use three public datasets, two from TripAdvisor (TA) and one from RateBeer (RB). They are commonly used for evaluating the performance of recommendation (Jannach, Karakaya, and Gedikli 2012; McAuley, Leskovec, and Jurafsky 2012). All three datasets are independent from each other. **TripAdvisor-12M (TA-12M):** This dataset contains $177,614$ records given by $1,475$ users based on 4 criteria including $value$, $location$, $service$, and $overall$ for $3,447$ hotels. Each user gives at least 2 ratings. The sparsity is around 99.12%. **TripAdvisor-20M (TA-20M):** This dataset contains $246,698$ records given by $4,503$ users based on 8 criteria including $value, location$, $cleanness$, $checkin$, $business$, $rooms$, $service$, and $overall$ for $6,702$ hotels. The sparsity is around 99.89%. **RateBeer(RB):** This dataset contains $1,326,451$ records given by $2,167$ users for $3,109$ beers based on 5 criteria including $appearance$, $aroma$, $palate$, $taste$ and $overall$. The sparsity is around 96.20%.

Side information: For RB dataset, the side information of users contains hometown, registration date, favorite style and 10 different scores, which are encoded into a binary vector of length 130, by using the method in (Wang et al. 2018). Similarly, the side information of items contains the item's brewery, ABV, and style, which are encoded into a binary vector of length 30. For TA datasets, the user and item additional matrices are generated similarly as RB. The side information of users contains hometown, registration date, zip code, contributions and followers, which are encoded into a binary vector of length 106. And the side information of items contains the item's address, the overall rating, and 14 details of hotel amenities, which are encoded into a binary vector of length 134.

Baselines: In order to evaluate the performance, we consider the following baselines in our experiments: **AFBM**: Aggregation function based method (Adomavicius and Kwon 2007) employs a matrix factorization to factor the observed user-criterion rating data, and then uses the learned model to estimate the ratings of a user on the individual criterion (excluding special overall criterion); **CMF**: Collective matrix factorization(Singh and Gordon 2008) is a model which simultaneously factorizes multiple sources, including the user-item matrix and matrices containing the additional side information; **HCF**: HCF is a hybrid collaborative filtering model (Dong et al. 2017) which unifies aSDAE model with matrix factorization; **DCF**: Deep collaborative filtering (Li, Kawale, and Fu 2015) is a recommendation model which combines probabilistic matrix factorization with marginalized denoising stacked autoencoders to achieve recommendation; **t-SVD**: Tensor Singular Value Decomposition (Zhang and Aeron 2017) is a model to generalize MF approaches to higher dimensional;

Alg.	RB(s) vs TA12M(t)		RB(s) vs TA20M(t)	
	60%	80%	60%	80%
AFBM	1.194	1.152	1.284	1.166
CMF	1.182	1.139	1.268	1.098
HOSVD	1.150	1.059	1.184	1.084
DCF	1.161	1.069	1.216	1.092
HCF	1.083	1.062	1.122	1.070
CCCFNet	1.053	1.043	1.058	1.048
t-SVD	1.149	1.038	1.176	1.061
DTF	0.936	0.922	1.040	0.962
DTTD	**0.886**	**0.876**	**0.892**	**0.880**

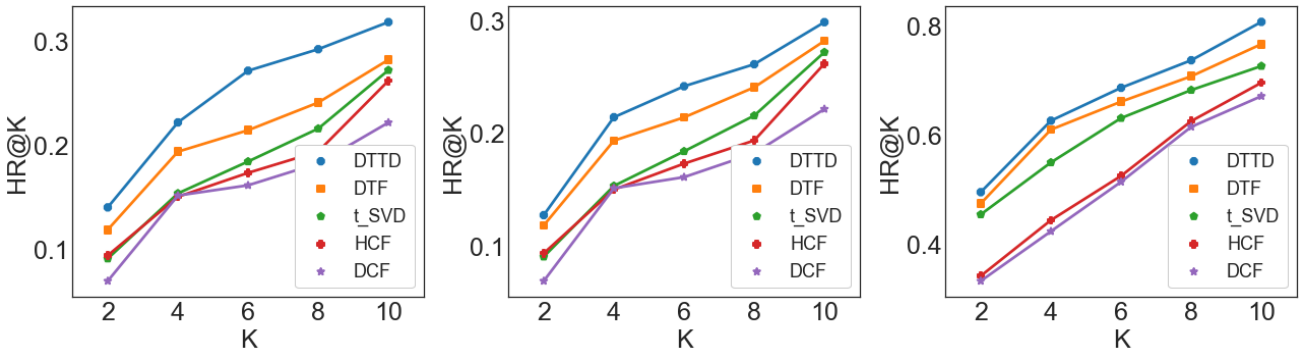
Table 1: Performance comparison in terms of RMSE.

Alg.	RB(s) vs TA12M(t)		RB(s) vs TA20M(t)	
	60%	80%	60%	80%
AFBM	0.872	0.806	0.891	0.828
CMF	0.861	0.783	0.886	0.785
HOSVD	0.787	0.721	0.798	0.750
DCF	0.816	0.770	0.825	0.775
HCF	0.738	0.720	0.747	0.744
CCCFNet	0.720	0.713	0.740	0.737
t-SVD	0.747	0.710	0.805	0.744
DTF	0.677	0.663	0.716	0.690
DTTD	**0.621**	**0.608**	**0.666**	**0.662**

Table 2: Performance comparison in terms of MAE.

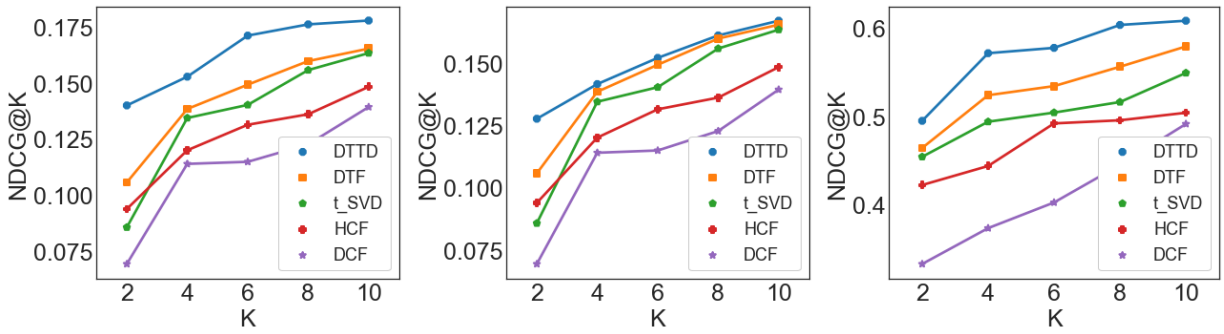
Evaluation metric: Root mean squared error (RMSE), mean absolute error (MAE), hit ratio (HR) and the normalized discounted cumulative gain (NDCG) (Chen et al. 2021; Chen and Wang 2021; Xiao and Wang 2021) are used as metrics. We organize TA12M(s) vs TA20M(t), RB(s) vs TA20M(t) and RB(s) vs TA12M(t) as three pairs for evaluation, where the former acts as the source domain and the latter acts as the target domain. For all comparison methods, we train each of them with 60% and 80% percentage 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.

Performance Comparison Tables 1 and 2 illustrate the



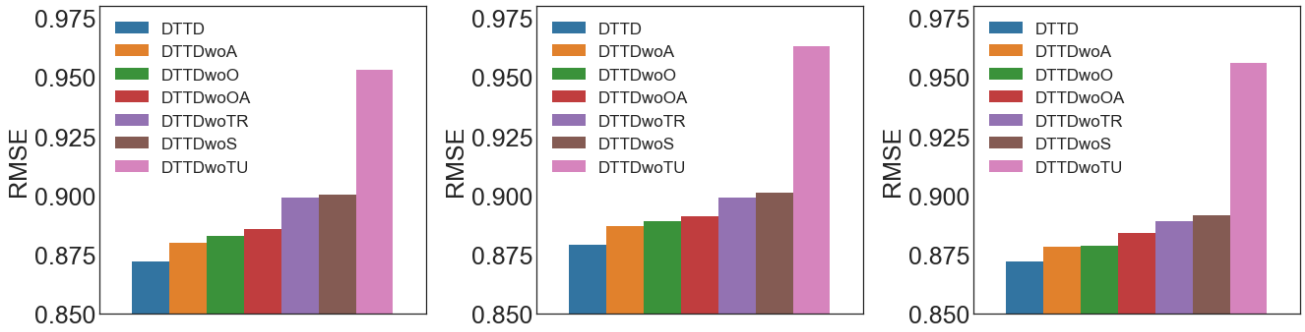
(a) TA12M(s) vs TA20M(t) (b) RB(s) vs TA20M(t) (c) RB(s) vs TA12M(t)

Figure 6: Top-K recommendation in terms of HR@K.



(a) TA12M(s) vs TA20M(t) (b) RB(s) vs TA20M(t) (c) RB(s) vs TA12M(t)

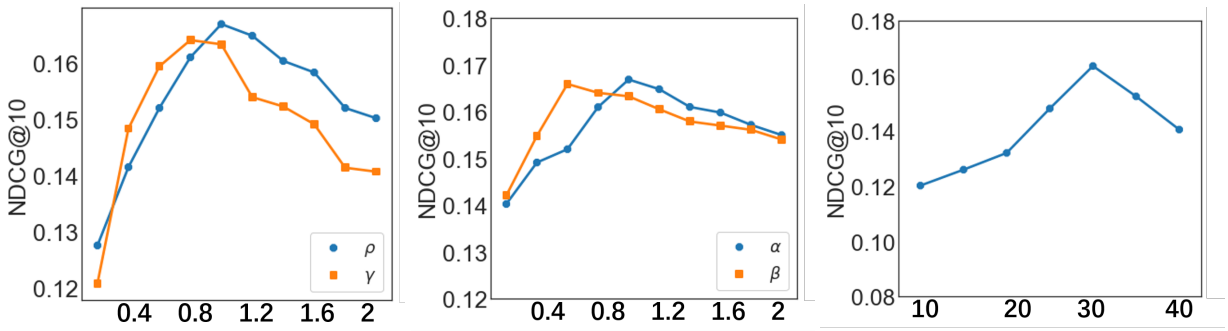
Figure 7: Top-K recommendation in terms of NDCG@K.



(a) TA12M(s) vs TA20M(t) (b) RB(s) vs TA20M(t) (c) RB(s) vs TA12M(t)

Figure 8: Ablation Study of DTTD.

performance of all methods in terms of the average RMSE and MAE, where the lowest RMSE/MAE in each dataset is highlighted in boldface. It is observed that the proposed DTTD achieves the *best* performance for all cases. As shown, CCCFNet, HCF, DCF and CMF outperform AFBM in general cases, and DTTD outperforms t-SVD and HOSVD, which demonstrate the effectiveness of incorporating the side information into the multi-criteria ratings (i.e. 3D rating tensor). That DTTD, CCCFNet, HCF and DTF outperform CMF indicates that deep structure can acquire a better feature of side information. HCF, DCF, CMF and AFBM only consider the correlation between arbitrary two of three dimensions so DTTD and DTF outperform these methods. And CCCFNet has better performance in RB(s) vs TA20M(t) and RB(s) vs TA12M(t)(60%) than t-SVD, indicating the effectiveness of cross-domain knowledge and side information. However, when the multi-criteria ratings increase and the sparsity of the dataset decreases, t-SVD has better performance in RB(s) vs TA12M(t)(80%) case. It shows that multi-criteria ratings play a key role in the prediction of multi-criteria recommendation. That DTTD outperforms CCCFNet, DCF and HCF indicates that Tucker



(a) (b) (c)

Figure 9: Analysis of Parameters in terms of NDCG@10:(a) γ and ρ; (b) α and β (c) Dimension of latent factor f.

decomposition method can effectively learn intrinsic interactions among three dimensions, indicating a good fit for multi-criteria recommendations. DTTD outperforms DTF, indicating that cross-domain knowledge under orthogonal constraint can effectively overcome the data sparsity issue.

Top-K Analysis To further evaluate the Top-K item recommendation, the experimental results of HR@K and NDCG@K on three datasets with respect to the number of latent factors are shown in Figure 6 and 7, where K varies from 2 to 10. It is observed that a similar conclusion could be drawn from Figure 6 and 7 demonstrating that DTTD achieves the best performance on three datasets for all cases. Furthermore, instead of a fixed length of recommendation list K, Figure 6 and 7 evaluate the Top-K item recommendation by considering a variant K from 2 to 10. Here, the K we choose is small because users usually pay high attention to just a few top recommendations. It is observed that as K increases, the resulting performance gets improved as expected, because the probability that the target item appears in the Top-K list is enhanced.

Parameter Analysis To evaluate the effect of various hyperparameters, we set $\alpha, \beta, \gamma, \rho$ in $\{0.2, 0.4, \ldots, 2.0\}$, and we consider the dimension of latent factor f in $\{10, 15, \ldots, 40\}$. In Figure 9(a), NDCG firstly rises up with the increment of γ, ρ, and then goes down, indicating that the side information improves accuracy, but it becomes a hold-back when getting more attention. Figure 9 (b) depicts the importance of OC-SDAE. The best performance is obtained when α is set to 0.6 and β is set to 1.0, indicating that users' side information affects more on NDCG compared to items. Figure 9 (c) shows that a high dimension of latent factor is not good to capture implicit information and a relatively low dimension of latent factors leads to better performance.

Ablation Analysis A careful ablation study is conducted to justify the effectiveness of DTTD, where each method is defined as follows:

- DTTDwoTR: DTTD without the knowledge transfer;
- DTTDwoS: DTTD without OC-SDAEs;
- DTTDwoA: DTTD without the alignment algorithm;
- DTTDwoO: Replacing OC-SDAE with SDAE;
- DTTDwoAO: DTTDwoO without alignment algorithm;
- DTTDwoTU: Replacing Tucker decomposition with matrix factorization in DTTD;

All test results of ablation analysis in terms of RMSE are shown in Figure 8 and a couple of observations are worth being highlighted as: The best performance is obtained by DTTD for all cases, indicating that each of components does contribute to the effectiveness and robustness of the whole model; DTTD achieves a better performance than DTTDwoO, indicating that OC-SDAE with the reduction of the scale variation outperforms SDAE; DTTD achieves a better performance than DTTDwoO, indicating that OC-SDAE with the reduction of the scale variation outperforms SDAE; DTTD performs better than DTTDwoAO, and DTTDwoAO performs close to DTTDwoTR indicating that CDAA and OC-SDAE play a key role in the knowledge transfer via core tensor, as shown in Figure 5; Removing the side information leads to a significant drop on the performance especially for TA datasets, which indicates that incorporating the side information using OC-SDAE is essential to handle the sparsity problem in multi-criteria recommendations; Consistent with the transfer analysis above, the knowledge transfer via core tensor plays a key role in the performance improvement for all cases.

Conclusion

In this paper,we propose deep transfer tensor decomposition (DTTD) method to solve sparsity problem in Tucker decomposition based recommendations. By considering an orthogonal constraint, OC-SDAE has been proposed to acquire effective latent representation with a small scale-variation. CDAA has been presented to overcome the rotation between two core tensors in different domains. The proposed DTTD can learn effective latent factors by making full use of the rating tensor, the side information and cross-domain knowledge. We evaluate our DTTD using RMSE, MAE and Top-K recommendations. Experimental results including various detailed analysis demonstrate the effectiveness of cross-domain Tucker decomposition for sparse multi-criteria recommendations, exhibiting a superior performance in comparison to state-of-the-art techniques.

Acknowledgments

The authors gratefully acknowledge funding support from the Westlake University and Bright Dream Joint Institute for Intelligent Robotics.

References

Adomavicius, G.; and Kwon, Y. 2007. New Recommendation Techniques for Multicriteria Rating Systems. *IEEE Intelligent Systems* 22(3): 48–55. ISSN 1541-1672. doi:10.1109/MIS.2007.58.

Bhargava, P.; Phan, T.; Zhou, J.; and Lee, J. 2015. Who, What, When, and Where: Multi-Dimensional Collaborative Recommendations Using Tensor Factorization on Sparse User-Generated Data. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, 130–140. doi:10.1145/2736277.2741077. URL https://doi.org/10.1145/2736277.2741077.

Chen, Z.; Gai, S.; and Wang, D. 2019. Deep Tensor Factorization for Multi-Criteria Recommender Systems. In *2019 IEEE International Conference on Big Data (Big Data)*, 1046–1051. IEEE.

Chen, Z.; Ge, J.; Zhan, H.; Huang, S.; and Wang, D. 2021. Pareto Self-Supervised Training for Few-Shot Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021*.

Chen, Z.; and Wang, D. 2021. Multi-Initialization Meta-learning With Domain Adaptation. In *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Chen, Z.; Wang, D.; and Yin, S. 2021. Improving Cold-Start Recommendation via Multi-Prior Meta-Learning. In *Advances in Information Retrieval 43rd European Conference on IR Research, ECIR 2021*.

Dong, X.; Yu, L.; Wu, Z.; Sun, Y.; Yuan, L.; and Zhang, F. 2017. A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 1309–1315. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14676.

Gai, S.; Zhao, F.; Kang, Y.; Chen, Z.; Wang, D.; and Tang, A. 2019. Deep Transfer Collaborative Filtering for Recommender Systems. In *PRICAI 2019: Trends in Artificial Intelligence - 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III*, 515–528. doi:10.1007/978-3-030-29894-4_42. URL https://doi.org/10.1007/978-3-030-29894-4_42.

Hamada, M.; and Hassan, M. 2018. Artificial neural networks and particle swarm optimization algorithms for preference prediction in multi-criteria recommender systems. In *Informatics*, volume 5, 25.

Jannach, D.; Karakaya, Z.; and Gedikli, F. 2012. Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, 674–689. doi:10.1145/2229012.2229065. URL https://doi.org/10.1145/2229012.2229065.

Jannach, D.; Zanker, M.; and Fuchs, M. 2014. Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations. *Information Technology & Tourism* 14(2): 119–149.

Lakiotaki, K.; Matsatsinis, N. F.; and Tsoukias, A. 2011. Multicriteria User Modeling in Recommender Systems. *IEEE Intelligent Systems* 26: 64–76.

Lakiotaki, K.; Tsafarakis, S.; and Matsatsinis, N. F. 2008. UTA-Rec: a recommender system based on multiple criteria analysis. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008*, 219–226. doi:10.1145/1454008.1454043. URL https://doi.org/10.1145/1454008.1454043.

Li, Q.; Wang, C.; and Geng, G. 2008. Improving personalized services in mobile commerce by a novel multicriteria rating approach. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, 1235–1236. doi:10.1145/1367497.1367743. URL https://doi.org/10.1145/1367497.1367743.

Li, S.; Kawale, J.; and Fu, Y. 2015. Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, 811–820. doi:10.1145/2806416.2806527. URL https://doi.org/10.1145/2806416.2806527.

Malik, O. A.; and Becker, S. 2018. Low-rank tucker decomposition of large tensors using tensorsketch. In *Advances in Neural Information Processing Systems*, 10096–10106.

McAuley, J. J.; Leskovec, J.; and Jurafsky, D. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, 1020–1025. doi:10.1109/ICDM.2012.110. URL https://doi.org/10.1109/ICDM.2012.110.

Mikeli, A.; Apostolou, D.; and Despotis, D. K. 2013. A Multi-criteria Recommendation Method for Interval Scaled Ratings. In *2013 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Atlanta, Georgia, USA, 17-20 November 2013, Workshop Proceedings*, 9–12. doi:10.1109/WI-IAT.2013.141. URL https://doi.org/10.1109/WI-IAT.2013.141.

Nilashi, M.; Ibrahim, O. b.; and Ithnin, N. 2014. Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications* 41(8): 3879–3900. ISSN 0957-4174. doi:10.1016/j.eswa.2013.12.023. URL http://www.sciencedirect.com/science/article/pii/S0957417413009986.

Rendle, S.; Marinho, L. B.; Nanopoulos, A.; and Schmidt-Thieme, L. 2009. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the*

15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009, 727–736. doi:10.1145/1557019.1557100. URL https://doi.org/10.1145/1557019.1557100.

Sahoo, N.; Krishnan, R.; Duncan, G. T.; and Callan, J. 2012. Research Note - The Halo Effect in Multicomponent Ratings and Its Implications for Recommender Systems: The Case of Yahoo! Movies. *Inf. Syst. Res.* 23(1): 231–246. doi:10.1287/isre.1100.0336. URL https://doi.org/10.1287/isre.1100.0336.

Singh, A. P.; and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, 650–658. doi:10.1145/1401890.1401969. URL https://doi.org/10.1145/1401890.1401969.

Tallapally, D.; Sreepada, R. S.; Patra, B. K.; and Babu, K. S. 2018. User preference learning in multi-criteria recommendations using stacked auto encoders. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, 475–479. doi:10.1145/3240323.3240412. URL https://doi.org/10.1145/3240323.3240412.

Wang, H.; Zhang, F.; Hou, M.; Xie, X.; Guo, M.; and Liu, Q. 2018. SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, 592–600. doi:10.1145/3159652.3159666. URL https://doi.org/10.1145/3159652.3159666.

Xiao, T.; Liang, S.; and Meng, Z. 2019a. Dynamic Collaborative Recurrent Learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 1151–1160. doi:10.1145/3357384.3357901. URL https://doi.org/10.1145/3357384.3357901.

Xiao, T.; Liang, S.; and Meng, Z. 2019b. Hierarchical Neural Variational Model for Personalized Sequential Recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 3377–3383. doi:10.1145/3308558.3313603. URL https://doi.org/10.1145/3308558.3313603.

Xiao, T.; Liang, S.; Shen, W.; and Meng, Z. 2019. Bayesian Deep Collaborative Matrix Factorization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 5474–5481. doi:10.1609/aaai.v33i01.33015474. URL https://doi.org/10.1609/aaai.v33i01.33015474.

Xiao, T.; and Wang, D. 2021. A General Offline Reinforcement Learning Framework for Interactive Recommendation. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*.

Yao, L.; Sheng, Q. Z.; Qin, Y.; Wang, X.; Shemshadi, A.; and He, Q. 2015. Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, 1007–1010. doi:10.1145/2766462.2767794. URL https://doi.org/10.1145/2766462.2767794.

Zhang, Z.; and Aeron, S. 2017. Exact Tensor Completion Using t-SVD. *IEEE Trans. Signal Processing* 65(6): 1511–1526.