Towards Unbiased Information Extraction and Adaptation in Cross-Domain Recommendation

Yibo Wang[1,2,§], **Yingchun Jian**[3], **Wenhao Yang**[1,2,§], **Shiyin Lu**[3], **Lei Shen**[3], **Bing Wang**[3], **Xiaoyi Zeng**[3], **Lijun Zhang**[1,2,*]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
[3] Alibaba Group, Hangzhou 310052, China

Abstract

Cross-Domain Recommendation (CDR) leverages additional knowledge from auxiliary domains to address the long-standing data sparsity issue. However, existing methods typically acquire this knowledge by minimizing the average loss over all domains, overlooking the fact that different domains possess different user-preference distributions. As a result, the acquired knowledge may contain biased information from data-rich domains, leading to performance degradation in data-scarce domains. In this paper, we propose a *novel* CDR method, which takes domain distinctions into consideration to extract and adapt unbiased information. Specifically, our method consists of two key components: Unbiased Information Extraction (UIE) and Unbiased Information Adaptation (UIA). In the UIE, inspired by distributionally robust optimization, we optimize the worst-case performance across all domains to extract domain-invariant information, preventing the potential bias from auxiliary domains. In the UIA, we introduce a *new* user-item attention module, which employs domain-specific information from historically interacted items to attend the adaptation of domain-invariant information. To verify the effectiveness of our method, we conduct extensive experiments on three real-world datasets, each of which contains three extremely sparse domains. Experimental results demonstrate the considerable superiority of our proposed method compared to baselines.

Introduction

Over past decades, recommendation systems have played a significant role in a variety of applications, such as stream media (Covington, Adams, and Sargin 2016), e-commerce (Smith and Linden 2017) and social networking (Naumov et al. 2019). Most existing studies on recommendation systems follow the classical collaborative filtering framework, which captures user preferences based on historical user-item interactions (Sarwar et al. 2001; Linden, Smith, and York 2003; Yi et al. 2017). However, in practical applications, it is common to suffer the data sparsity issue where available interactions fall short of revealing underlying preferences, which in turn hampers the effectiveness of existing methods (Ricci, Rokach, and Shapira 2015).

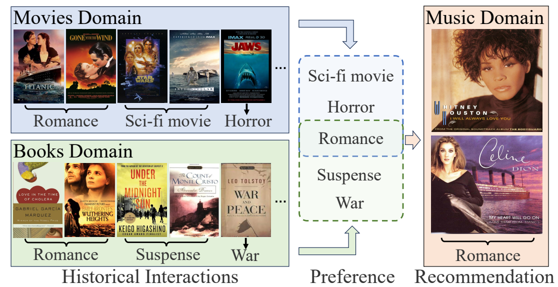

Figure 1: A simple illustration of Cross-Domain Recommendation over three domains: movies, books and music.

To alleviate this issue, Cross-Domain Recommendation (CDR) has been proposed, leveraging additional knowledge learned from auxiliary domains to enhance the recommendation performance in data-scarce domains (Zhu et al. 2021). The underlying rationale is that users typically have similar preferences across different domains. For example, as shown in Figure 1, although our understanding of user preferences in the music domain is limited, we infer that the user likely prefers romantic music based on his common preference, i.e., *Romance*, observed in the movies and books domains. According to the number of target domains, prior research on CDR can be generally divided into three groups: single-target CDR (Singh and Gordon 2008; Tan et al. 2014; Lu, Dong, and Smyth 2018), dual-target CDR (Zhu et al. 2019; Li and Tuzhilin 2020; Liu et al. 2020; Li and Tuzhilin 2021; Zhu et al. 2022), and multi-target CDR (Yang et al. 2017; Krishnan et al. 2020; Li et al. 2023; Yang et al. 2024).

Single-target and dual-target CDR primarily focus within two domains, whereas multi-target CDR aims to boost performance across three or more domains simultaneously. In multi-target CDR, a classical paradigm typically involves two steps: extraction and adaptation (Zang et al. 2022). Specifically, the extraction focuses on generating a global user embedding for each user from all participating domains, to capture the common preference. The adaptation aims to employ domain-specific information, such as local user embedding derived solely from the target domain, to establish linkages between the target domain and the global user embedding. Based on this paradigm, several multi-target CDR methods have been proposed recently (Kim et al. 2019; Zhu

[§]The work is completed during the internship at Alibaba Group,
[*]Corresponding author, zhanglj@lamda.nju.edu.cn

et al. 2023a; Li et al. 2023; Ning et al. 2023).

Despite extensive developments in multi-target CDR, we observe that existing studies still suffer two limitations. First, during the extraction, previous methods follow a conventional strategy that learns the domain-invariant knowledge by directly minimizing the average loss across all participating domains, overlooking a crucial fact that different domains typically possess different user-preference distributions. Consequently, such an average-case minimization strategy may introduce biased information from data-rich domains and compromise performances in data-scarce domains. In other words, optimizing the average loss may not handle the domain distinctions in multi-target CDR well. Second, during the adaptation, existing studies predominantly utilize user-side information to facilitate the transfer of global user embeddings into target domains, overlooking item-side information which harbors potential domain-specific user preferences from a complementary perspective.

To address the above two limitations, in this paper, we propose a *novel* multi-target CDR method, namely Unbiased Information Extraction and Adaptation (UIEA), which takes domain distinctions into consideration to extract and transfer unbiased information. Specifically, our method consists of three parts: pretraining, Unbiased Information Extraction (UIE) and Unbiased Information Adaptation(UIA). First, in the pretraining, we learn the local user embeddings and local item embeddings by exploiting the Bayesian matrix factorization method (Rendle et al. 2009), to capture the domain-specific information in each individual domain. Then, in the UIE, we employ distributionally robust optimization to prevent the potential bias from auxiliary domains. To be more precise, we develop a *novel* extraction loss that incorporates a reconstruction component to guide the generation of global user embeddings, and a fine-tuning component to infuse domain-specific information into the global user embeddings from each domain. To prevent potential biases, we optimize for the worst-case performance, specifically targeting the highest extraction loss across all domains. Additionally, we implement a joint training strategy that concurrently updates model parameters and importance weights for different domains, thereby generating global embeddings that are refined with domain-invariant information from all participating domains. In the UIA, we propose a *new* user-item attention module that leverages the information not only from users but also from their historically interacted items to facilitate the adaptation of global user embeddings. Our proposed UIA presents a comprehensive view of user preferences derived from their past interactions with items and is capable of refining global user embeddings using the items themselves, preventing the potential biased information.

Empirically, we conduct extensive experiments over three real-world datasets: Amazon (Ni, Li, and McAuley 2019), Douban (Zhu et al. 2019) and IE datasets. Experimental results demonstrate that UIEA exhibits significant performance improvements over its competitors.

Related Work

In this section, we briefly review recent studies on CDR and distributionally robust optimization.

Cross-Domain Recommendation

The single-target CDR aims to extract additional knowledge from one source domain and transfer it into a different target domain (Singh and Gordon 2008; Tan et al. 2014; Lu, Dong, and Smyth 2018; Sopchoke, ichi Fukui, and Numao 2018). The dual-target CDR treats both source and target domains equally and seeks to improve performances in two domains concurrently (Zhu et al. 2019; Liu et al. 2020; Li and Tuzhilin 2021; Zhu et al. 2022, 2023b).

Although various methods have been proposed in single-target and dual-target CDR, they are limited in modeling the pairwise relations between domains. When extended into the situation with n domains, these methods suffer high computational costs of handling at least $\binom{n}{2}$ relations (Cui et al. 2020). To address this issue, multi-target CDR is introduced, which aims to boost the performance in more than two domains simultaneously. Compared with previous two problems, multi-target CDR presents greater challenges, as more participating domains are considered concurrently, increasing the difficulty of extracting and transferring knowledge across different domains (Zang et al. 2022).

In recent years, there are several studies focus on multi-target CDR. Vartak et al. (2017) employ meta-learning to capture user preferences from item interactions. Yang et al. (2017) propose a generative model to capture the domain-invariant preference with user behaviors across all domains. Similarly, Kim et al. (2019) also investigate multi-domain user behaviors but adopt recurrent neural networks to mine the underlying preference. Ma et al. (2019) utilize disentangled representation learning to capture user preferences (prototype) at a macro level while disentangling item features affecting preferences at a micro level. Krishnan et al. (2020) combine meta-learning and transfer learning to extract contextual invariants, and integrate them with domain-specific user and item embeddings. Cui et al. (2020) and Zhu et al. (2023a) represent user-item interactions across domains as a shared graph and apply graph embedding algorithms to extract the cross-domain preference. Ning et al. (2023) explicitly disentangle domain-invariant and domain-specific knowledge, and employ a random walk-based strategy for knowledge transfer.

As previous mentioned, existing methods for multi-target CDR typically utilize the average-case optimization strategy to extract domain-invariant information, overlooking distribution distinctions across different domains. This neglect could potentially cause the extracted information being biased by data-rich domains, subsequently leading to performance degradation in data-scarce domains. Furthermore, the item information that encapsulates user preferences is also ignored during the adaptation.

Distributionally Robust Optimization

Classical machine learning methods commonly minimize the average loss on a training set, facing significant performance degradation when the test distribution differs from the training distribution (Koh et al. 2021). In contrast, Distributionally Robust Optimization (DRO) focuses on minimizing the worst-case loss over an uncertainty distribution

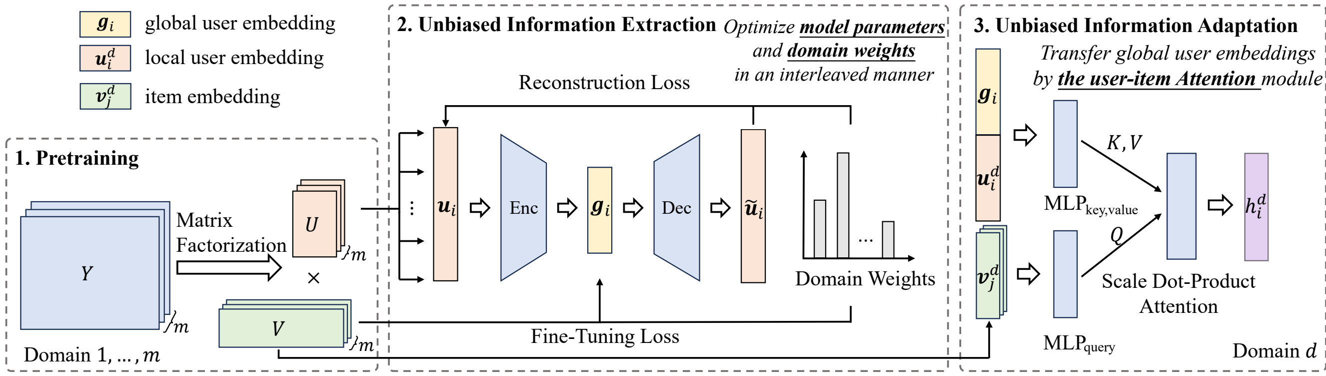

Figure 2: An illustration of UIEA framework. In the pretraining, UIEA pretrains local user and item embeddings in each domain. In the UIE, we take local user embeddings $\{\mathbf{u}_i^d \in U_d\}$ as inputs and generate the global user embedding $\boldsymbol{g}_i$ for each user $i \in \mathcal{U}$. In the UIA, we exploit item embeddings $\{\mathbf{v}_j^d | j \in \mathcal{P}_i^d\}$ to transfer the global embedding $\boldsymbol{g}_i$ into the target domain d.

set $\mathcal{S}$ to improve the robustness (Ben-Tal et al. 2013). Mathematically, it can be formulated as the following minimax optimization problem

$$\min_{\mathbf{w}\in\mathcal{W}} \sup_{\mathcal{P}\in\mathcal{S}} \left\{\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[\ell(\mathbf{w};\mathbf{z})]\right\}, \tag{1}$$

where $\mathbf{w} \in \mathcal{W}$ denotes the model parameters, $\mathbf{z}$ denotes the input data randomly sampled from $\mathcal{P}$, and $\ell(\cdot;\cdot)$ denotes the loss function that measures the performance. Over recent decades, plenty of efforts have been made to solve (1) over the uncertain set $\mathcal{S}$ with infinite distributions (Namkoong and Duchi 2016, 2017; Kuhn et al. 2019; Levy et al. 2020) and finite distributions (Oren et al. 2019; Sagawa et al. 2020; Zhang et al. 2023, 2024; Yu et al. 2024).

In this paper, we mainly focus on the latter one, which is also referred to as group DRO. To be precise, given an uncertain set with m distributions, i.e., $\mathcal{S} = \{\mathcal{P}_1, \cdots, \mathcal{P}_m\}$, the original problem (1) becomes

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{i\in[m]} \left\{\mathbb{E}_{\mathbf{z}\sim\mathcal{P}_i}[\ell(\mathbf{w};\mathbf{z})]\right\}. \tag{2}$$

By minimizing the worst-case performance, DRO has provided an effective way to prevent overfitting in the scenarios with different distributions, such as federated learning (Mohri, Sivek, and Suresh 2019) and distribution shift (Duchi, Hashimoto, and Namkoong 2023).

Method

In this section, we first formulate multi-target CDR and then present UIEA, of which the framework is shown in Figure 2.

Problem Formulation

In multi-target CDR, each domain contains a unique item set while sharing a common user set. Specifically, we denote $\mathcal{U}$ as the common user set and $\mathcal{V}_d$ as the domain-specific item set for each domain $d \in \{1, \cdots, m\}$. User-item interactions in the domain d are represented by a matrix $Y_d \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}_d|}$, where $|\mathcal{U}|$ denotes the number of common users and $|\mathcal{V}_d|$ denotes the number of items. In this paper, we focus on the implicit feedback setting (Zhuang and Zhang 2024): each entry y_{ij}^d in the matrix Y_d is chosen from $\{0, 1\}$, where the value indicates whether there is an interaction between the user $i \in \mathcal{U}$ and the item $j \in \mathcal{V}_d$. Formally, we have:

$$y_{ij}^d = \begin{cases} 1, & \text{if the interaction } (i, j) \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

Given multiple interaction matrices $Y_1, \cdots, Y_m$, our goal is to improve performances over m domains simultaneously.

Pretraining

To begin with, we pretrain the local user embedding to extract domain-specific information from each domain. Concretely, for each domain $d \in \{1, \cdots, m\}$, we employ the Bayesian matrix factorization method (Rendle et al. 2009) to factorize the interaction matrix Y_d into a user embedding matrix $U_d \in \mathbb{R}^{|\mathcal{U}|\times n}$ and an item embedding matrix $V_d \in \mathbb{R}^{|\mathcal{V}_d|\times n}$, where n denotes the embedding dimension. In this way, we obtain m local embeddings for each user $i \in \mathcal{U}$. The preference score is computed based on the inner product, i.e., $r_{ij}^d = \langle \mathbf{u}_i^d, \mathbf{v}_j^d \rangle$ for the user $i \in \mathcal{U}$ with local embedding $\mathbf{u}_i^d \in U_d$ and the item $j \in \mathcal{V}_d$ with local embedding $\mathbf{v}_j^d \in V_d$. In this part, we aim to minimize the following Bayesian Personalized Ranking (BPR) loss (Rendle et al. 2009):

$$\begin{aligned} \mathcal{L}_{bpr}^d = &\sum\nolimits_{i\in\mathcal{U}} \sum\nolimits_{j\in\mathcal{I}_i^d} \sum\nolimits_{k\notin\mathcal{I}_i^d} -\log\sigma\left(r_{ij}^d - r_{ik}^d\right) \\ &+ \lambda_U \sum\nolimits_{i\in\mathcal{U}} \|\mathbf{u}_i^d\|_2^2 + \lambda_V \sum\nolimits_{j\in\mathcal{V}_d} \|\mathbf{v}_j^d\|_2^2, \end{aligned} \tag{3}$$

where σ denotes the sigmoid function, $\mathcal{I}_i^d$ denotes the item set that user i have interacted with in the domain d, λ_U and λ_V are the regularization hyper-parameters. By minimizing (3), for each user $i \in \mathcal{U}$, we obtain m local embeddings $\mathbf{u}_i^1, \cdots, \mathbf{u}_i^m$ that capture the domain-specific preference in individual domains.

Unbiased Information Extraction

In the extraction, we employ an Autoencoder model as the embedding generator, which takes local embeddings from all domains as inputs and generates the global embedding with domain-invariant information. Inspired by the spirit of DRO, we optimize for the worst-case performance across all domains to handle domain distinctions, and adopt a joint training strategy where both model parameters and domain weights are updated concurrently.

Specifically, for a user $i \in \mathcal{U}$, we first concatenate the pretrained local embeddings $\{\mathbf{u}_i^d \in U_d\}$ from each domain $d \in \{1, \cdots, m\}$ as the input, i.e., $\mathbf{u}_i = [\mathbf{u}_i^1, \cdots, \mathbf{u}_i^m]$. Then, we map $\mathbf{u}_i$ into a latent space and reconstruct the input embeddings $\tilde{\mathbf{u}}_i = [\tilde{\mathbf{u}}_i^1, \cdots, \tilde{\mathbf{u}}_i^m]$ as follows

$$\boldsymbol{g}_i = \mathrm{MLP}_{\mathrm{enc}}\left\{[\mathbf{u}_i^1, \cdots, \mathbf{u}_i^m]\right\} \quad (4)$$

$$[\tilde{\mathbf{u}}_i^1, \cdots, \tilde{\mathbf{u}}_i^m] = \mathrm{MLP}_{\mathrm{dec}}\left\{\boldsymbol{g}_i\right\}, \quad (5)$$

where $\boldsymbol{g}_i$ denotes the global embedding of user i and shares the same dimension as the local embedding, i.e., $\boldsymbol{g}_i \in \mathbb{R}^n$.

To train our embedding generator, we introduce a novel extraction loss, which consists of two parts: a reconstruction loss and a fine-tuning loss. The former one measures the differences between the local embedding $\mathbf{u}_i^d$ and the reconstructed embedding $\tilde{\mathbf{u}}_i^d$ in the domain d, as shown below:

$$\mathcal{L}_{rec}^d = \|\mathbf{u}_i^d - \tilde{\mathbf{u}}_i^d\|_2. \quad (6)$$

The latter one fine-tunes the global user embedding $\boldsymbol{g}_i$ in the domain d:

$$\mathcal{L}_{ft}^d = \sum_{j \in \mathcal{I}_i^d} \sum_{k \notin \mathcal{I}_i^d} -\log \sigma\left(\langle \boldsymbol{g}_i, \mathbf{v}_j^d - \mathbf{v}_k^d \rangle\right) + \lambda_U \|\boldsymbol{g}_i\|_2^2. \quad (7)$$

It should be noticed that although (7) shares a similar form to (3) used in the pretraining, the fine-tuning loss (7) measures the performance of global user embedding $\boldsymbol{g}_i$ in the domain d, rather than the local user embedding $\mathbf{u}_i^d$. Moreover, in (7), only the global user embedding is learnable, and item embeddings remain fixed.

Putting (6) and (7) together, we obtain the extraction loss over the domain d:

$$\mathcal{L}_{ext}^d = \alpha \mathcal{L}_{rec}^d + (1-\alpha)\mathcal{L}_{ft}^d, \quad (8)$$

where α is the hyper-parameter that controls the balance between the reconstruction and fine-tuning. To train the embedding generator, a conventional approach is to minimize the average loss over all domains, i.e.,

$$\bar{\mathcal{L}}_{ext} = \frac{1}{m} \sum\nolimits_{d=1}^{m} \mathbb{E}_{\mathbf{u}_i^d \sim \mathcal{P}_d}[\mathcal{L}_{ext}^d(\mathbf{w}; \mathbf{u}_i^d)], \quad (9)$$

where $\mathcal{P}_d$ denotes the underlying distribution in the domain d. However, this approach implicitly assumes that all domains are equal, neglecting the distinctions among different domains and potentially introducing biased information from data-rich domains, which can result in performance degradation in data-scarce domains.

To tackle this issue, we propose to optimize the embedding generator for the worst-case performance. Concretely, we optimize the model parameters $\mathbf{w}$ within the domain that suffers the highest extraction loss. Formally, it can be formulated as the following min-max optimization problem:

$$\min_{\mathbf{w} \in \mathcal{W}} \max_{d \in [m]} \left\{\mathbb{E}_{\mathbf{u}_i^d \sim \mathcal{P}_d}[\mathcal{L}_{ext}^d(\mathbf{w}; \mathbf{u}_i^d)]\right\}. \quad (10)$$

To facilitate optimizations, we further cast (10) as a stochastic saddle-point problem below (Nemirovski et al. 2009):

$$\min_{\mathbf{w} \in \mathcal{W}} \max_{\mathbf{q} \in \Delta_m} \left\{\sum\nolimits_{d=1}^{m} q_d \mathbb{E}_{\mathbf{u}_i^d \sim \mathcal{P}_d}[\mathcal{L}_{ext}^d(\mathbf{w}; \mathbf{u}_i^d)]\right\}, \quad (11)$$

where q_d denotes the weight of domain d and will be updated during the training. To solve (11), we adopt a joint training strategy—concurrently optimizing model parameters $\mathbf{w}$ and domain weights $\mathbf{q}$. Specifically, at round t, we first draw a user $i \in \mathcal{U}$ with the local embedding $\mathbf{u}_i^d$ from the distribution $\mathcal{P}_d$ in the domain $d \in \{1, \cdots, m\}$. Then, we construct the stochastic gradients with respect to the model parameters $\mathbf{w}$ as shown below:

$$\mathbf{g}_w(\mathbf{w}_t, \mathbf{q}_t) = \sum\nolimits_{d=1}^{m} q_{t,d} \nabla \mathcal{L}_{ext}^d(\mathbf{w}_t; \mathbf{u}_i^d), \quad (12)$$

and the domain weights $\mathbf{q}$:

$$\mathbf{g}_q(\mathbf{w}_t, \mathbf{q}_t) = [\mathcal{L}_{ext}^1(\mathbf{w}_t; \mathbf{u}_i^1), \cdots, \mathcal{L}_{ext}^m(\mathbf{w}_t; \mathbf{u}_i^m)]^\top. \quad (13)$$

Next, we adjust $\mathbf{w}$ and $\mathbf{q}$ according to the updating rules:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{\eta_w \langle \mathbf{g}_w(\mathbf{w}_t, \mathbf{q}_t), \mathbf{w} - \mathbf{w}_t \rangle + B_w(\mathbf{w}, \mathbf{w}_t)\right\}$$

$$\mathbf{q}_{t+1} = \arg\min_{\mathbf{q} \in \Delta_m} \left\{\eta_q \langle -\mathbf{g}_q(\mathbf{w}_t, \mathbf{q}_t), \mathbf{q} - \mathbf{q}_t \rangle + B_q(\mathbf{q}, \mathbf{q}_t)\right\},$$

where $\eta_w, \eta_q > 0$ are learning rates, and $B_w(\cdot, \cdot)$ and $B_q(\cdot, \cdot)$ are the Bregman divergence in $\mathcal{W}$ and Δ_m, respectively. For brevity, we choose two common formulations of Bregman divergence: $B_w(\mathbf{w}_1, \mathbf{w}_2) = 2^{-1}\|\mathbf{w}_1 - \mathbf{w}_2\|_2^2$ and $B_q(\mathbf{q}_1, \mathbf{q}_2) = \sum_{d=1}^{m} q_{1,d} \log(q_{1,d}/q_{2,d})$. Therefore, the updating rules can be rewritten as:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left[\mathbf{w}_t - \eta_w \mathbf{g}_w(\mathbf{w}_t, \mathbf{q}_t)\right] \quad (14)$$

$$q_{t+1,d} = \frac{q_{t,d} \exp\left(\eta_q \mathcal{L}_{ext}^d(\mathbf{w}_t; \mathbf{u}_i^d)\right)}{\sum_{k=1}^{m} q_{t,k} \exp\left(\eta_q \mathcal{L}_{ext}^k(\mathbf{w}_t; \mathbf{u}_i^k)\right)}, \quad (15)$$

where $\Pi_{\mathcal{W}}[\cdot]$ denotes the projection operation that finds the nearest point in $\mathcal{W}$. From (15), we observe that during the update, the weight q_t^d is adaptively updated based on the current performance in domain d. For example, domains with higher extraction losses are assigned with higher weights in subsequent rounds, thereby preventing overfitting to any specific domain.

After T rounds, we obtain the final model parameters $\mathbf{w}_T$ and domain weights $\mathbf{q}_T$. For each user $i \in \mathcal{U}$, we exploit the embedding generator with parameters $\mathbf{w}_T$ to produce the global embedding $\boldsymbol{g}_i$ according to (4).

Unbiased Information Adaptation

After the extraction, we obtain global embeddings refined from all participating domains. However, a direct application of them in the target domain is inappropriate, as they only capture domain-invariant information and can not adapt to the target domain well due to domain distinctions. For this reason, we propose a novel user-item attention module, which leverages domain-specific information from historically interacted items to adequately transfer global embeddings into the target domain.

Algorithm 1: The UIEA framework

Input: Common user set $\mathcal{U}$, and item sets $\{\mathcal{V}_1, \cdots, \mathcal{V}_m\}$ for domains $\{1, \cdots, m\}$.

Pretraining:

1: Learn the local domain-specific embeddings $\{U_d, V_d\}$ by minimizing (3) in each domain $d \in \{1, \cdots, m\}$.

Unbiased Information Extraction:

1: Initialize model parameters $\mathbf{w}_0$, and domain weights $\mathbf{q}_0 = [1/m, \cdots, 1/m]^\top \in \mathbb{R}^m$.
2: **for** $t = 1, \cdots, T$ **do**
3: Draw a user $i \in \mathcal{U}$ with its local embeddings $\mathbf{u}_i^d$ in each domain $d \in \{1, \cdots, m\}$.
4: Calculate the gradient with regard to $\mathbf{w}$ and $\mathbf{q}$ according to (12) and (13), respectively.
5: Update $\mathbf{w}_t$ and $\mathbf{q}_t$ according to (14) and (15), respectively.
6: **end for**
7: Generate the global embedding $\boldsymbol{g}_i$ for each user $i \in \mathcal{U}$ according to (4).

Unbiased Information Adaptation:

1: In domain d, for each user $i \in \mathcal{U}$, compute the adapted user embedding $\mathbf{h}_i^d$ according to (18).
2: Optimize parameters of the user-item attention module by minimizing (3) with the adapted user embedding $\mathbf{h}_i^d$.

Specifically, for a user $i \in \mathcal{U}$, we first compute the average embeddings of all historically interacted items $\{\mathbf{v}_j^d | j \in \mathcal{P}_i^d\}$ in the domain d. Then, we take the average item embedding as *query*:

$$Q = \text{MLP}_{\text{query}} \left\{ \text{avg} \left(\mathbf{v}^d_{j \in \mathcal{P}_i^d} \right) \right\}, \tag{16}$$

and the concatenation of global user embedding $\boldsymbol{g}_i$ and local user embedding $\mathbf{u}_i^d$ as *key, value*:

$$K = \text{MLP}_{\text{key}} \left\{ [\boldsymbol{g}_i, \mathbf{u}_i^d] \right\}, \; V = \text{MLP}_{\text{value}} \left\{ [\boldsymbol{g}_i, \mathbf{u}_i^d] \right\}. \tag{17}$$

Next, we compute the adapted user embedding $\mathbf{h}_i^d$ via the scaled-dot-product attention (Vaswani et al. 2017):

$$\mathbf{h}_i^d = \text{softmax} \left(QK^\top / \sqrt{n} \right) V, \tag{18}$$

where n is the embedding dimension. For any item $j \in \mathcal{V}_d$, we compute $r_{ij} = \langle \mathbf{h}_i^d, \mathbf{v}_j^d \rangle$ as the preference score. Furthermore, to train the user-item attention module, we fix the local user and item embeddings and only update the final user embedding $\mathbf{h}_i^d$ by minimizing the BPR loss (3) in each target domain d. The detailed process of UIEA is summarized in Algorithm 1.

Experiments

In this section, we conduct empirical studies to answer the following questions:

- **Q1**: Does the proposed UIEA avoid performance degradation caused by domain distinctions when compared to the single-domain method? How does our model perform compared to existing cross-domain methods?

Dataset	Domains	#Users	#Items	#Interactions	Density
Amazon	Books	33,561	58,071	671,407	1.72×10^{-5}
	Movies		20,585	494,116	7.16×10^{-4}
	Elec		21,830	325,308	4.58×10^{-5}
Douban	Books	1,122	8,630	75,013	1.16×10^{-4}
	Movies		20,168	674,182	2.98×10^{-3}
	Music		7,107	66,664	1.40×10^{-4}
IE	BR	117,453	18,147	811,119	8.51×10^{-6}
	KR	38,706		310,769	4.43×10^{-4}
	US	55,270		434,638	1.81×10^{-5}

Table 1: Statistics of Datasets

- **Q2**: How does UIEA perform in the worst-case and average-case performance?
- **Q3**: How do the sub-modules introduced in UIEA contribute to the performance improvement?

Experimental Settings

We first introduce experimental settings including datasets, evaluation protocols, baselines and implementations.

Datasets. Our experiments are conducted on three real-world datasets, each of which includes three domains: Amazon (Books, Movies and Elec) (Ni, Li, and McAuley 2019), Douban (Books, Movies and Music) (Zhu et al. 2019), and IE (BR, KR, US).[1] All datasets are randomly divided into training, validation and test sets with the ratio of 7:1:2. Moreover, following the common practice (Zhu et al. 2019, 2020), we retain users and items with at least 5 interactions. The statistics of each dataset is summarized in Table 1.

Evaluation Protocols. According to the classical protocol (Krichene and Rendle 2020), we adopt the leave-one-out strategy to ensure an unbiased performance evaluation. Specifically, for each test user, we reserve one interacted item as the positive sample and randomly draw 99 irrelevant items as negative samples. Then, we rank the 100 items according to prediction scores and evaluate recommendation performance on top-k ranking results. Furthermore, we choose three widely-used metrics for performance evaluation: MRR@k, NDCG@k, and HR@k.

Baseline Methods. We compare UIEA with the single-domain methods: BPRMF (Rendle et al. 2009) and NeuMF (He et al. 2017), and the cross-domain methods: CMF (Singh and Gordon 2008), HeroGraph (Cui et al. 2020), GA-MTCDR (Zhu et al. 2023a), EDDA (Ning et al. 2023) and CAT-ART (Li et al. 2023).

Implementations. In the pretraining, we set the embedding dimension $n = 64$ and hyper-parameters $\lambda_U = \lambda_V = 10^{-5}$ in the BPR loss (3), and the mini-batch size $N = 2048$. In the UIE module, we configure the encoder of the embedding generator with layer sizes $[3 \times n, 2 \times n, n]$, and the

[1]IE is a real-world dataset collected by AliExpress, a leading global online marketplace under Alibaba International Digital Commerce Group, including user-item interactions from Brazil, Korea, and the United States.

Datasets		Metric	BPRMF	NeuMF	CMF	HeroGraph	GA-MTCDR	EDDA	CAT-ART	UIEA	Impr.
Amazon@10	Books	MRR	24.17 ± 0.17	21.08 ± 0.46	23.20 ± 0.14	20.05 ± 0.08	19.91 ± 0.24	$\underline{25.65} \pm 0.15$	24.91 ± 0.21	$\mathbf{29.57}^* \pm 0.11$	15.28%
		HR	43.17 ± 0.11	41.73 ± 0.79	44.41 ± 0.16	39.66 ± 0.10	38.80 ± 0.46	$\underline{47.39} \pm 0.07$	44.51 ± 0.43	$\mathbf{52.06}^* \pm 0.09$	9.85%
		NDCG	28.64 ± 0.14	25.93 ± 0.55	28.19 ± 0.14	24.66 ± 0.08	24.36 ± 0.29	$\underline{30.76} \pm 0.12$	29.52 ± 0.26	$\mathbf{34.87}^* \pm 0.10$	13.36%
	Movies	MRR	29.46 ± 0.20	25.76 ± 0.27	26.60 ± 0.10	24.93 ± 0.06	26.19 ± 0.08	28.21 ± 0.10	$\underline{29.62} \pm 0.14$	$\mathbf{34.13}^* \pm 0.14$	15.23%
		HR	56.31 ± 0.04	53.44 ± 0.35	53.80 ± 0.12	52.15 ± 0.08	53.52 ± 0.22	$\underline{55.37} \pm 0.17$	53.65 ± 0.23	$\mathbf{62.56}^* \pm 0.17$	12.99%
		NDCG	35.80 ± 0.15	32.27 ± 0.29	33.01 ± 0.10	31.34 ± 0.05	32.63 ± 0.11	34.61 ± 0.12	$\underline{35.29} \pm 0.16$	$\mathbf{40.86}^* \pm 0.14$	15.78%
	Elec	MRR	17.19 ± 0.15	17.46 ± 0.16	$\underline{17.05} \pm 0.03$	16.45 ± 0.06	15.50 ± 0.31	16.29 ± 0.06	15.32 ± 0.34	$\mathbf{19.66}^* \pm 0.21$	15.31%
		HR	34.51 ± 0.48	35.12 ± 0.28	37.86 ± 0.17	$\underline{40.78} \pm 0.11$	32.25 ± 0.48	34.45 ± 0.14	30.87 ± 0.54	$\mathbf{41.83}^* \pm 0.31$	2.57%
		NDCG	21.25 ± 0.23	22.55 ± 0.19	21.92 ± 0.06	$\underline{22.14} \pm 0.05$	19.42 ± 0.35	20.54 ± 0.02	18.95 ± 0.39	$\mathbf{24.86}^* \pm 0.23$	12.29%
Douban@10	Books	MRR	18.67 ± 0.50	17.65 ± 0.60	18.27 ± 0.47	17.00 ± 0.96	16.67 ± 0.68	18.21 ± 0.79	$\underline{18.30} \pm 0.79$	$\mathbf{21.21}^* \pm 0.40$	15.90%
		HR	40.19 ± 0.94	38.09 ± 1.20	$\underline{40.63} \pm 0.80$	36.97 ± 0.93	36.88 ± 0.58	39.16 ± 1.22	39.28 ± 0.71	$\mathbf{45.20}^* \pm 0.73$	11.25%
		NDCG	23.71 ± 0.49	22.43 ± 0.66	$\underline{23.50} \pm 0.55$	21.66 ± 0.95	21.39 ± 0.67	23.12 ± 0.89	23.22 ± 0.75	$\mathbf{26.84}^* \pm 0.47$	14.21%
	Movies	MRR	27.87 ± 0.31	27.11 ± 0.18	26.19 ± 0.39	22.36 ± 0.29	23.23 ± 0.19	26.27 ± 0.57	$\underline{26.35} \pm 0.07$	$\mathbf{29.32}^* \pm 0.08$	11.27%
		HR	58.93 ± 0.84	60.00 ± 0.16	56.49 ± 0.35	51.98 ± 0.48	51.94 ± 0.73	$\underline{57.61} \pm 0.91$	57.29 ± 0.40	$\mathbf{63.04}^* \pm 0.93$	9.42%
		NDCG	35.17 ± 0.40	35.07 ± 0.12	33.31 ± 0.34	29.34 ± 0.34	29.98 ± 0.27	33.62 ± 0.66	$\underline{33.63} \pm 0.15$	$\mathbf{37.25}^* \pm 0.17$	10.76%
	Music	MRR	15.50 ± 0.75	14.61 ± 0.47	$\underline{15.23} \pm 0.59$	13.64 ± 0.80	13.38 ± 0.55	13.44 ± 0.28	15.03 ± 0.52	$\mathbf{18.63}^* \pm 0.42$	22.32%
		HR	35.91 ± 1.74	34.82 ± 0.83	$\underline{36.07} \pm 1.74$	33.23 ± 1.36	32.32 ± 1.60	31.77 ± 1.15	35.13 ± 1.32	$\mathbf{40.60}^* \pm 0.57$	12.56%
		NDCG	20.26 ± 0.97	19.32 ± 0.56	$\underline{20.09} \pm 0.85$	18.20 ± 0.92	17.79 ± 0.79	17.72 ± 0.45	19.71 ± 0.67	$\mathbf{23.76}^* \pm 0.44$	18.27%
IE@10	BR	MRR	31.00 ± 0.11	23.49 ± 0.39	24.54 ± 0.20	29.36 ± 0.41	28.88 ± 0.10	28.32 ± 0.30	$\underline{30.60} \pm 0.06$	$\mathbf{35.17}^* \pm 0.19$	14.93%
		HR	56.35 ± 0.11	47.40 ± 0.66	47.87 ± 0.26	$\underline{58.99} \pm 0.53$	50.24 ± 0.18	52.75 ± 0.22	55.94 ± 0.09	$\mathbf{62.08}^* \pm 0.16$	5.24%
		NDCG	37.01 ± 0.10	29.11 ± 0.45	30.04 ± 0.20	36.35 ± 0.43	33.89 ± 0.12	34.09 ± 0.27	$\underline{36.61} \pm 0.04$	$\mathbf{41.56}^* \pm 0.17$	13.52%
	KR	MRR	25.65 ± 0.24	20.29 ± 0.24	21.12 ± 0.13	$\underline{24.18} \pm 0.56$	18.64 ± 0.17	19.96 ± 0.46	23.19 ± 0.04	$\mathbf{27.13}^* \pm 0.13$	12.20%
		HR	51.12 ± 0.20	42.38 ± 0.23	45.40 ± 0.20	$\underline{52.37} \pm 0.94$	41.62 ± 0.27	41.89 ± 0.86	48.02 ± 0.11	$\mathbf{53.86}^* \pm 0.18$	2.85%
		NDCG	31.65 ± 0.22	25.48 ± 0.23	26.82 ± 0.14	$\underline{30.80} \pm 0.66$	24.01 ± 0.20	25.11 ± 0.55	29.05 ± 0.05	$\mathbf{33.44}^* \pm 0.08$	8.57%
	US	MRR	23.88 ± 0.10	19.52 ± 0.63	21.53 ± 0.15	21.62 ± 0.43	14.79 ± 0.21	20.22 ± 0.89	$\underline{22.94} \pm 0.14$	$\mathbf{25.51}^* \pm 0.08$	11.20%
		HR	47.80 ± 0.21	42.01 ± 0.51	45.56 ± 0.13	$\underline{47.56} \pm 0.64$	34.86 ± 0.31	41.77 ± 1.08	46.97 ± 0.08	$\mathbf{50.95}^* \pm 0.16$	7.13%
		NDCG	29.52 ± 0.10	24.80 ± 0.72	27.17 ± 0.14	27.68 ± 0.48	19.47 ± 0.24	25.27 ± 0.93	$\underline{28.60} \pm 0.12$	$\mathbf{31.51}^* \pm 0.09$	10.17%

Table 2: Performance (%) on three datasets. * denotes that the best-performing method significantly outperforms the best cross-domain baseline (indicated by the underline) on the paired t-test (p-value < 0.05). *Impr.* denotes the improvement of UIEA over the best-performing cross-domain baseline method.

decoder with layer sizes $[n, 2 \times n, 3 \times n]$. Additionally, the hyper-parameter in (8) is set as $\alpha = 0.1$. In the UIA module, both MLP_{key} and $\text{MLP}_{\text{value}}$ have layer sizes $[2 \times n, n]$, and $\text{MLP}_{\text{query}}$ has layer sizes $[n, n]$. All methods are implemented by the Pytorch framework, and we employ Adam (Kingma and Ba 2014) with default parameters as the optimizer. The learning rate is chosen from $\{10^{-4}, \cdots, 10^{-1}\}$ for each method. All experiments are conducted on a single machine equipped with Tesla V100 GPUs. Each experiment is repeated 5 times with different random seeds, and we report the mean and standard deviation as results.

Performance Comparisons

We summarize the performance of UIEA and baseline methods on three datasets in Table 2.

Overall, UIEA significantly outperforms single-domain and cross-domain baselines (**Q1**). First, compared with single-domain methods BPRMF and NeuMF, our UIEA exhibits better performance in both data-rich domains (e.g., Books of Amazon) and data-scarce domains (e.g., Elec of Amazon). This improvements demonstrate that our method not only takes advantage of external information from auxiliary domains, but also prevents the potential performance degradation brought by domain distinctions. Second, compared with cross-domain methods, UIEA also exhibits significant performance improvements, especially in sparser domains. For example, our method achieves the best performance with the improvements of 15.23%, 12.99% and 15.78% in the Movies of Amazon, and 22.32%, 12.56% and 18.27% in the Music of Douban. We summarize this into two reasons: (i) our proposed method is designed to optimize the worst-case performance, ensuring that the global user embedding does not overfit to data-rich domains; (ii) we fully utilize domain-specific information from interacted items to transfer global embeddings into the target domain.

Furthermore, we conduct refined analysis on the results in Table 2, and investigate the worst-case and average-case performances across three domains of all datasets (**Q2**). To make it clear, we take the metric: HR as an example, and present the formally definitions of the worst-case and average-case performance as shown below

$$\text{avg}_{\text{HR}} = (\text{HR}_1 + \text{HR}_2 + \text{HR}_3)/3$$
$$\text{worst}_{\text{HR}} = \min(\text{HR}_1, \text{HR}_2, \text{HR}_3).$$

The experimental results are presented in Figure 3. We observe that in both worst and average cases, UIEA exhibits superior performance compared to all baselines. Specially, in the worst case on Amazon dataset, our method achieves performance improvements of 14.37%, 5.47% and 12.29% in MRR@10, HR@10 and NDCG@10, respectively. This is reasonable since we explicitly consider the domain distinctions and optimize the worst-case performance to prevent the potential overfitting. In the average case, our UIEA also outperforms the best baseline method with the improvements of 17.71%, 14.01% and 17.04% in MRR@10, HR@10 and NDCG@10, respectively in the Amazon dataset. This can

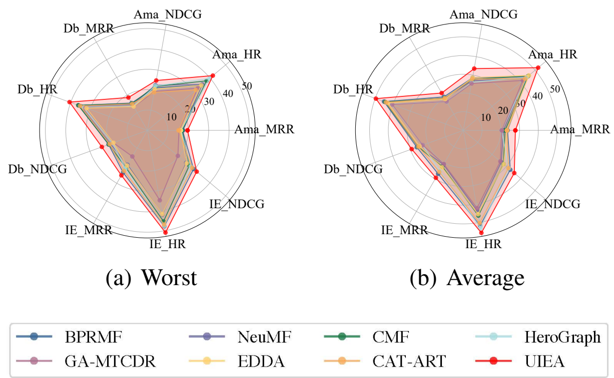

Figure 3: The worst-case and average performance (%).

be attributed to the user-item attention module that effectively utilizes the domain-specific information from interacted items for the adaptation. More comprehensive discussions are provided in the ablation study.

Ablation Study

The strengths of UIEA mainly come from two crucial components: (i) Unbiased Information Extraction (UIE), which optimizes the worst-case performance across all domains; (ii) Unbiased Information Adaptation (UIA), which employs historically interacted items through a user-item attention module to facilitate the adaptation of global embeddings. To investigate the impacts brought by UIE and UIA, we study the following ablation variants (**Q3**):

- Rec + MLP, which merely utilizes the reconstruction loss
$$\mathcal{L}_{rec} = \sum\nolimits_{d=1}^{m} \|\mathbf{u}_i^d - \tilde{\mathbf{u}}_i^d\|_2$$
for extraction and MLP layers for adaptation, i.e., $\mathbf{h}_i^d = \text{MLP}\left\{[\boldsymbol{g}_i, \mathbf{u}_i^d]\right\}$;
- AVG + MLP, which employs the average-case optimization strategy, i.e., minimizing the average loss (9) over all domains, for extraction and MLP layers for adaptation;
- UIE + MLP, which employs UIE for extraction and MLP layers for adaptation;
- AVG + UIA, which exploits the average-case optimization strategy for extraction and UIA for adaptation.

Table 3 summarizes the ablation results on Amazon. Specifically, compared with the single domain method BPRMF, Rec + MLP achieves a slight improvement in the Books domain, but suffers a significant decrease in other two domains, especially in Movies. This indicates that Rec + MLP overfits to the data-rich domain, and suffers heavy performance degradation in data-scarce domains. Furthermore, compared with Rec + MLP, AVG + MLP additionally fine-tunes global embeddings in each domain by minimizing (7) in the extraction stage and thereby, improves the performance over all domains. However, in two data-scarce domains, i.e., Movies and Elec, the performance is slightly worse than BPRMF,

Model	Amazon Domain	Metric@10 MRR	HR	NDCG
BPRMF	Books	24.17 ± 0.17	43.17 ± 0.11	28.64 ± 0.14
	Movies	29.46 ± 0.20	56.31 ± 0.04	35.80 ± 0.15
	Elec	17.19 ± 0.15	34.51 ± 0.48	21.25 ± 0.23
Rec + MLP	Books	25.07 ± 0.16	47.15 ± 0.15	30.27 ± 0.16
	Movies	26.87 ± 0.09	54.02 ± 0.17	33.27 ± 0.11
	Elec	15.83 ± 0.13	34.93 ± 0.26	20.42 ± 0.16
AVG + MLP	Books	26.76 ± 0.22	48.48 ± 0.20	31.86 ± 0.22
	Movies	28.32 ± 0.36	55.35 ± 0.28	35.32 ± 0.34
	Elec	16.72 ± 0.28	34.58 ± 0.73	21.08 ± 0.38
UIE + MLP	Books	26.98 ± 0.16	48.99 ± 0.19	31.91 ± 0.16
	Movies	31.72 ± 0.07	59.98 ± 0.05	38.40 ± 0.05
	Elec	18.28 ± 0.02	37.52 ± 0.18	22.79 ± 0.03
AVG + UIA	Books	27.85 ± 0.15	50.27 ± 0.25	33.14 ± 0.17
	Movies	30.08 ± 0.07	57.68 ± 0.19	36.60 ± 0.09
	Elec	17.48 ± 0.12	35.40 ± 0.19	21.68 ± 0.13
UIEA	Books	$\mathbf{29.57} \pm 0.11$	$\mathbf{52.06} \pm 0.09$	$\mathbf{34.87} \pm 0.10$
	Movies	$\mathbf{34.13} \pm 0.14$	$\mathbf{62.56} \pm 0.17$	$\mathbf{40.86} \pm 0.14$
	Elec	$\mathbf{19.66} \pm 0.21$	$\mathbf{41.83} \pm 0.31$	$\mathbf{24.86} \pm 0.23$

Table 3: Ablation results (%) on the Amazon dataset.

which indicates that AVG + MLP still can not handle the domain distinctions well.

Moreover, we investigate other two variants: UIE + MLP and AVG + UIA. In the former one, we observe considerable improvements in Movies and Elec compared to AVG + MLP. This observation illustrates that leveraging the worst-case optimization strategy could avoid the potential overfitting to the data-rich domain (i.e., Books) and hence, improve performances in data-scarce domains. In the latter one, the performances of all domains are enhanced compared to AVG + MLP, which demonstrates that incorporating historically interacted items could effectively facilitate the adaption of domain-invariant information, boosting the recommendation performance in the target domain.

Conclusion

In this paper, we propose a novel multi-target CDR method named UIEA, which takes domain distinctions into consideration to achieve unbiased information extraction and adaptation. Specifically, in the extraction, we optimize the worst-case performance across all domains to generate global embeddings with domain-invariant information. In the adaptation, we introduce a novel user-item attention module, which utilizes domain-specific information from historically interacted items to facilitate an adequate adaptation of global embeddings. Extensive experiments on three real-world datasets have demonstrated the significant improvements of UIEA over baseline methods.

Acknowledgments

This work was partially supported by NSFC (U23A20382), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and Alibaba Group through Alibaba Innovative Research Program.

References

Ben-Tal, A.; den Hertog, D.; Waegenaere, A. D.; Melenberg, B.; and Rennen, G. 2013. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2): 341–357.

Covington, P.; Adams, J.; and Sargin, E. 2016. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198.

Cui, Q.; Wei, T.; Zhang, Y.; and Zhang, Q. 2020. HeroGRAPH: A heterogeneous graph framework for multi-target cross-domain recommendation. *Proceedings of the 3rd Workshop on Online Recommender Systems and User Modeling co-located with the 14th ACM Conference on Recommender Systems*.

Duchi, J.; Hashimoto, T.; and Namkoong, H. 2023. Distributionally robust losses for latent covariate mixtures. *Operations Research*, 71(2): 649–664.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182.

Kim, D.; Kim, S.; Zhao, H.; Li, S.; Rossi, R. A.; and Koh, E. 2019. Domain switch-aware holistic recurrent neural network for modeling multi-domain user behavior. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 663–671.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *ArXiv e-prints*, arXiv:1412.6980.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; Lee, T.; David, E.; Stavness, I.; Guo, W.; Earnshaw, B. A.; Haque, I. S.; Beery, S.; Leskovec, J.; Kundaje, A.; Pierson, E.; Levine, S.; Finn, C.; and Liang, P. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, 5637–5664.

Krichene, W.; and Rendle, S. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1748–1757.

Krishnan, A.; Das, M.; Bendre, M.; Yang, H.; and Sundaram, H. 2020. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1081–1090.

Kuhn, D.; Esfahani, P. M.; Nguyen, V. A.; and Shafieezadeh-Abadeh, S. 2019. Wasserstein distributionally robust optimization: Theory and applications in machine learning. *Operations Research & Management Science in the Age of Analytics*, 130–166.

Levy, D.; Carmon, Y.; Duchi, J. C.; and Sidford, A. 2020. Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems 33*, 8847–8860.

Li, C.; Xie, Y.; Yu, C.; Hu, B.; Li, Z.; Shu, G.; Qie, X.; and Niu, D. 2023. One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, 366–374.

Li, P.; and Tuzhilin, A. 2020. DDTCDR: Deep dual transfer cross domain recommendation. In *Proceedings of the 13rd ACM International Conference on Web Search and Data Mining*, 331–339.

Li, P.; and Tuzhilin, A. 2021. Dual metric learning for effective and efficient cross-domain recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 321–334.

Linden, G.; Smith, B.; and York, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1): 76–80.

Liu, M.; Li, J.; Li, G.; and Pan, P. 2020. Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 885–894.

Lu, Y.; Dong, R.; and Smyth, B. 2018. Why I like it: multi-task learning for recommendation and explanation. In *Proceedings of the 12nd ACM Conference on Recommender Systems*, 4–12.

Ma, J.; Zhou, C.; Cui, P.; Yang, H.; and Zhu, W. 2019. Learning disentangled representations for recommendation. In *Advances in Neural Information Processing Systems 32*, 5712–5723.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning*, 4615–4625.

Namkoong, H.; and Duchi, J. C. 2016. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems 29*, 2216–2224.

Namkoong, H.; and Duchi, J. C. 2017. Variance-based regularization with convex objectives. In *Advances in Neural Information Processing Systems 30*, 2971–2980.

Naumov, M.; Mudigere, D.; Shi, H.-J. M.; Huang, J.; Sundaraman, N.; Park, J.; Wang, X.; Gupta, U.; Wu, C.-J.; Azzolini, A. G.; Dzhulgakov, D.; Mallevich, A.; Cherniavskii, I.; Lu, Y.; Krishnamoorthi, R.; Yu, A.; Kondratenko, V.; Pereira, S.; Chen, X.; Chen, W.; Rao, V.; Jia, B.; Xiong, L.; and Smelyanskiy, M. 2019. Deep learning recommendation model for personalization and recommendation systems. *ArXiv e-prints*, arXiv:1906.00091.

Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 188–197.

Ning, W.; Yan, X.; Liu, W.; Cheng, R.; Zhang, R.; and Tang, B. 2023. Multi-domain recommendation with embedding disentangling and domain alignment. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1917–1927.

Oren, Y.; Sagawa, S.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 4227–4237.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 452–461.

Ricci, F.; Rokach, L.; and Shapira, B. 2015. *Recommender systems handbook*. Springer.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, 285–295.

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 & Data Mining*, 650–658.

Smith, B.; and Linden, G. 2017. Two decades of recommender systems at Amazon. *IEEE International Computing*, 21(3): 12–18.

Sopchoke, S.; ichi Fukui, K.; and Numao, M. 2018. Explainable cross-domain recommendations through relational learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 8159–8160.

Tan, S.; Bu, J.; Qin, X.; Chen, C.; and Cai, D. 2014. Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, 127: 124–134.

Vartak, M.; Thiagarajan, A.; Miranda, C.; Bratman, J.; and Larochelle, H. 2017. A meta-learning perspective on cold-start recommendations for items. In *Advance in Neural Information Processing Systems 30*, 6904–6914.

Vaswani, A.; Shazeer, N.; Niki Parmar, J. U.; Jones, L.; Gomez, A. N.; Łukasz Kaiser; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Yang, C.; Yan, H.; Yu, D.; Li, Y.; and Chiu, D. M. 2017. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 175–184.

Yang, W.; Jian, Y.; Wang, Y.; Lu, S.; Wang, B.; Tang, H.; and Zhang, L. 2024. Not all embeddings are created equal: towards robust cross-domain recommendation via contrastive learning. In *Proceedings of the ACM Web Conference 2024*, 3195–3206.

Yi, J.; Hsieh, C.-J.; Varshney, K. R.; Zhang, L.; and Li, Y. 2017. Scalable demand-aware recommendation. In *Advance in Neural Information Processing Systems 30*, 2409–2418.

Yu, D.; Cai, Y.; Jiang, W.; and Zhang, L. 2024. Efficient algorithms for empirical group distributionally robust optimization and beyond. In *Proceedings of the 41st International Conference on Machine Learning*, 57384–57414.

Zang, T.; Zhu, Y.; Liu, H.; Zhang, R.; and Yu, J. 2022. A survey on cross-domain recommendation: taxonomies, methods, and future directions. *ACM Transactions on Information Systems*, 41(42): 1–39.

Zhang, L.; Bai, H.; Tu, W.-W.; Yang, P.; and Hu, Y. 2024. Efficient stochastic approximation of minimax excess risk optimization. In *Proceedings of the 41st International Conference on Machine Learning*, 58599–58630.

Zhang, L.; Zhao, P.; Zhuang, Z.-H.; Yang, T.; and Zhou, Z.-H. 2023. Stochastic approximation approaches to group distributionally robust optimization. In *Advances in Neural Information Processing Systems 36*, 52490–52522.

Zhu, F.; Chen, C.; Wang, Y.; Liu, G.; and Zheng, X. 2019. DTCDR: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1533–1542.

Zhu, F.; Wang, Y.; Chen, C.; Liu, G.; and Zheng, X. 2020. A graphical and attentional framework for dual-target cross-domain recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 3001–3008.

Zhu, F.; Wang, Y.; Chen, C.; Zhou, J.; Li, L.; and Liu, G. 2021. Cross-domain recommendation: challenges, progress, and prospects. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 950–958.

Zhu, F.; Wang, Y.; Zhou, J.; Chen, C.; Li, L.; and Liu, G. 2023a. A unified framework for cross-domain and cross-system recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 1171–1184.

Zhu, J.; Wang, Y.; Zhu, F.; and Sun, Z. 2023b. Domain disentanglement with interpolative data augmentation for dual-Target cross-domain recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 515–527.

Zhu, Y.; Tang, Z.; Liu, Y.; Zhuang, F.; Xie, R.; Zhang, X.; Lin, L.; and He, Q. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, 1507–1515.

Zhuang, Z.-H.; and Zhang, L. 2024. Soft contrastive learning for implicit feedback recommendations. In *Advances in Knowledge Discovery and Data Mining*, 219–230.