Personalized Transfer of User Preferences
for Cross-domain Recommendation

Yongchun Zhu[1,3,4], Zhenwei Tang[1], Yudan Liu[4], Fuzhen Zhuang[2,5,*], Ruobing Xie[4], Xu Zhang[4],
Leyu Lin[4] and Qing He[1,3]

[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing
Technology, CAS, Beijing 100190, China
[2]Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
[4]WeChat Search Application Department, Tencent, China
[5]SKLSDE, School of Computer Science, Beihang University, Beijing 100191, China
{zhuyongchun18s, heqing}@ict.ac.cn, lilvjosephtang@gmail.com, {danydliu, ruobingxie, xuonezhang,
goshawklin}@tencent.com,zhuangfuzhen@buaa.edu.cn

ABSTRACT

Cold-start problem is still a very challenging problem in recommender systems. Fortunately, the interactions of the cold-start users in the auxiliary source domain can help cold-start recommendations in the target domain. How to transfer user's preferences from the source domain to the target domain, is the key issue in Cross-domain Recommendation (CDR) which is a promising solution to deal with the cold-start problem. Most existing methods model a common preference bridge to transfer preferences for all users. Intuitively, since preferences vary from user to user, the preference bridges of different users should be different. Along this line, we propose a novel framework named Personalized Transfer of User Preferences for Cross-domain Recommendation (PTUPCDR). Specifically, a meta network fed with users' characteristic embeddings is learned to generate personalized bridge functions to achieve personalized transfer of preferences for each user. To learn the meta network stably, we employ a task-oriented optimization procedure. With the meta-generated personalized bridge function, the user's preference embedding in the source domain can be transformed into the target domain, and the transformed user preference embedding can be utilized as the initial embedding for the cold-start user in the target domain. Using large real-world datasets, we conduct extensive experiments to evaluate the effectiveness of PTUPCDR on both cold-start and warm-start stages. The code has been available at https://github.com/easezyc/WSDM2022-PTUPCDR.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Cross-domain Recommendation; Cold-start Problem; Meta Network; Personalized Transfer

1 INTRODUCTION

Recommender systems are playing more and more important roles in web and mobile applications. In recent years, recommender systems have attracted a vast amount of interest from industries and academia, and researchers have conducted a great deal of research to improve the recommendation performance [1, 11]. However, most of these recommender systems are hard to provide satisfying recommendations for new users, i.e., cold-start users, who have no historical interactions.

Cross-domain Recommendation (CDR) [25] which aims to transfer knowledge from an informative source domain to the target domain is a promising solution to alleviate the cold-start problem. The core task of CDR is to bridge user's preferences in the source domain and the target domain, also called preference transfer [37]. To achieve preference transfer, many existing CDR methods [4, 9, 16, 39] assume that all users share the same relationships between user preferences in the source domain and the target domain, and learn a common preference bridge shared by all users, as shown in Figure 1 (a).

In practice, due to individual differences, the complex relationships between the user preferences of the source and target domains vary from user to user. Hence, it is hard for a single bridge to capture such complicated and various relationships, which may degrade these CDR methods' performance. To alleviate the drawback, it is necessary to use personalized bridges to model various relationships between user preferences in different domains. In other words, the process of preference transfer should be personalized.

Along this line, we propose a novel framework named Personalized Transfer of User Preferences for Cross-domain Recommendation (PTUPCDR). PTUPCDR learns a meta network that takes users'
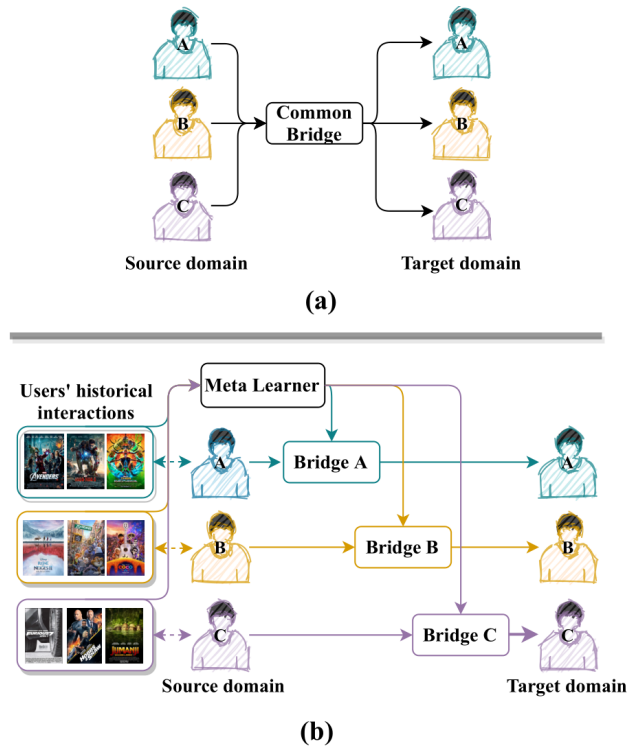
Figure 1: (a) In existing CDR methods: all users share the common bridge function. (b) The proposed PTUPCDR utilizes a meta network to generate personalized bridge functions for each user.

characteristic embeddings in the source domain as input and generates personalized bridges for each user, as shown in Figure 1(b). The generated bridge functions can be viewed as a model parameterized by the learned meta network. Note that the personalized bridge functions which depend on the users' characteristics vary from user to user, so the process of the preference transfer is personalized, which can capture preference relationships between different domains better than existing methods. After training, we feed user embeddings in the source domain into the meta-generated personalized bridge functions and obtain the transformed user embeddings. The transformed user embeddings are utilized as the initial embeddings in the target domain. With the initial embeddings, our method is effective for cold-start users who have no interactions in the target domain.

In practice, a high-level meta network is hard to optimize [18, 24], and how to optimize the meta network is another challenge. To learn the bridge function, existing bridge-based methods [4, 9, 16, 39] adopt a mapping-oriented optimization procedure to directly minimize the distance between the transformed users' embeddings from the informative source domain and the user's embedding in the target domain. In other words, with such an optimization procedure, the bridge function is sensitive to the quality of the users' embeddings. In practical recommender systems, it is pretty hard to learn reasonable embeddings [21, 42] for all users, which limits the performance of the bridge function learned with the mapping-oriented optimization. In addition, we find it is hard to learn the

meta network with the mapping-oriented optimization. Thus, to train the meta network, we take a task-oriented optimization procedure, which skips the users' embeddings in the target domain and directly utilizes the rating task as the optimization goal.

Most existing works in the literature [4, 9, 16, 37, 39] only testify their effectiveness by applying their methods upon simple base models (Matrix Factorization) in the extreme cold-start stage (users have no interactions in the target domain). Since such settings are far from real-world scenarios, we further explore how to use PTUPCDR in more practical scenarios to validate the compatibility and utility of PTUPCDR in real-world recommendations, e.g., warm-start scenarios, more complicated base models. Experimental results demonstrate that our proposed PTUPCDR is of good compatibility and utility in real-world recommendations.

The main contributions of our work are summarized into three folds:

- To solve the cold-start problem in CDR, we propose a novel method named PTUPCDR, utilizing a meta network to generate personalized bridge functions for each user, given the encoded users' characteristics in the source domain.
- To learn the meta network stably, we employ a task-oriented optimization procedure to alleviate the side effects of unreasonable users' embeddings.
- We conduct extensive experiments on three cross-domain tasks using Amazon review dataset to demonstrate the effectiveness and robustness of PTUPCDR for not only cold-start scenarios but also warm-start scenarios, while existing methods only testify their effectiveness in the cold-start scenarios.

2 RELATED WORK

2.1 Cross-domain Recommendation

Transfer learning aims to leverage knowledge from a source domain to improve the learning performance or minimize the number of labeled examples required in a target domain [30, 45], which led to an interest in many area, e.g., computer vision [29, 43, 44], natural language processing [19, 32]. Inspired by transfer learning, CDR is a promising solution to alleviate data sparsity and the cold-start problem in the target domain with the help of the auxiliary (source) domain. At the very beginning, CMF [25] assumes a shared global user embedding matrix for all domains, and it factorizes matrices from multiple domains simultaneously. CST [22] utilizes the user embedding in the source domain to initialize the embedding in the target domain and restricts them from being closed.

In recent years, researchers proposed many deep learning-based models to enhance knowledge transfer [5, 6, 8, 31, 34, 42]. CoNet [8] transfers and combines the knowledge by using cross-connections between feed-forward neural networks. MINDTL [6] combines the CF information of the target-domain with the rating patterns extracted from a cluster-level rating matrix in the source-domain. DDTCDR [15] develops a novel latent orthogonal mapping to extract user preferences over multiple domains while preserving relations between users across different latent spaces.

Another group of CDR methods focus on bridging user preferences in different domains [9, 16, 22, 36, 37, 39, 40], which is the most related work. CST [22] utilizes the user embedding learned in the source domain to initialize the user embedding in the target domain

and restricts them to being closed. Some methods [9, 16, 37, 40] explicitly model the preference bridge. Our study falls into the this bridge-based category. However, to the best of our knowledge, all of the bridge-based CDR works in the literature learn a shared bridge function for all users, while our PTUPCDR is the first to learn personalized bridges for each user.

2.2 Cold-start Recommendation

Providing recommendations for new users or items is challenging in recommender systems, also named the cold-start problem. There are two kinds of methods to solve the cold-start problems. The first type actively solves cold-start by designing a decision making strategy, such as using contextual-bandits [14, 20].

This paper belongs to the second type, which utilizes auxiliary information to help the cold-start stage. There are various kinds of auxiliary information could be exploited to improve cold-start recommendation performance, e.g., user attributes [13, 23], item attributes [17, 35, 42], knowledge graphs [28], samples in an auxiliary domain [16], etc. Usually, with samples in an auxiliary domain, the CDR methods can achieve much better results than other cold-start methods. Thus, in this paper, following most CDR work [9, 16], we only compare our method with CDR approaches.

2.3 Meta Learning

It is also named learning to learn, aiming to improve novel tasks' performance by training on similar tasks. There are various meta learning methods, e.g., metric-based methods [26, 27], gradient-based methods [3], and parameter-generating based methods [18]. The proposed PTUPCDR falls into the third group, which utilizes a meta learner to predict networks' parameters. Recently, researchers proposed many meta-based methods [12, 21, 41, 42] to improve recommender systems' performance. However, most of them fall into gradient-based methods and focus on single-domain recommendations, while we focus on cross-domain recommendations. The most related work is TMCDR [40] which utilizes meta-learning in CDR. However, TMCDR also trains a common bridge as existing bridge-based methods do.

3 MODEL

3.1 Problem Setting

In CDR, we have a source domain and a target domain. Each domain has a user set $\mathcal{U} = \{u_1, u_2, ...\}$, an item set $\mathcal{V} = \{v_1, v_2, ...\}$, and a rating matrix $\mathcal{R}$. $r_{ij} \in \mathcal{R}$ denotes the interaction between user u_i and item v_j. To distinguish these two domains, we denote the user, item sets, and the rating matrix of the source domain as $\mathcal{U}^s, \mathcal{V}^s, \mathcal{R}^s$, while $\mathcal{U}^t, \mathcal{V}^t, \mathcal{R}^t$ for the target domain. We define the overlapping users between the two domains as $\mathcal{U}^o = \mathcal{U}^s \cap \mathcal{U}^t$. In contrast, $\mathcal{V}^s$ and $\mathcal{V}^t$ are disjoint, which means there is no shared item between the two domains.

In latent factor models, the users and items are transformed into dense vectors, also called factors or embeddings. In this paper, $\boldsymbol{u}_i^d \in \mathbb{R}^k$ and $\boldsymbol{v}_j^d \in \mathbb{R}^k$ denote the embeddings of the user u_i^d and item v_j^d, respectively, where k denotes the dimensionality of embeddings and $d \in \{s, t\}$ represents the domain label. For each user u_i, we denote the list of her sequential interaction items in

---

Algorithm 1 Personalized Transfer of User Preferences for CDR (PTUPCDR)

Input: $\mathcal{U}^s, \mathcal{U}^t, \mathcal{V}^s, \mathcal{V}^t, \mathcal{U}^o, \mathcal{R}^s, \mathcal{R}^t$
Input: Meta network g_ϕ.
Input: Characteristic encoder h_θ.
Pre-training Stage:
  1. Learning a source model which contains $\boldsymbol{u}^s, \boldsymbol{v}^s$.
  2. Learning a target model which contains $\boldsymbol{u}^t, \boldsymbol{v}^t$.
Meta Stage:
  3. Learning a characteristic encoder h_θ and a meta network g_ϕ by minimizing Equation (7).
Initialization Stage:
  4. For a cold-start user u^t in the target domain, we use the transformed embedding $f_{u_i}(\boldsymbol{u}_i^s; \boldsymbol{w}_{u_i})$ as the user's initialized embedding in the target domain.

---

source domain by $\mathcal{S}_{u_i} = \{v_{t_1}^s, v_{t_2}^s, \cdots, v_{t_n}^s\}$, where n denotes the number of interacted items and $v_{t_n}^s$ denotes the interacted item in the source domain at timestamp t_n.

3.2 Characteristic Encoder

The first step to generate the personalized bridge function is to capture users' personalized transferable characteristics from interacted items. However, cold-start users have no interacted item in the target domain. Thus, it is essential to exploit the interacted items $\mathcal{S}$ in the source domain. Note that we need to find the transferable characteristics which are helpful for knowledge transfer.

Intuitively, various items have different contributions to knowledge transfer. The attention mechanism [33, 38] allows different parts to contribute differently when compressing them to a single representation. Therefore, we propose to employ the attention mechanism on item embeddings by performing a weighted sum:

$$\boldsymbol{p}_{u_i} = \sum_{v_j^s \in \mathcal{S}_{u_i}} a_j \boldsymbol{v}_j^s, \tag{1}$$

where $\boldsymbol{p}_{u_i} \in \mathbb{R}^k$ denotes the transferable characteristic embedding of user u_i, and a_j is the attention score for item v_j, which can be interpreted as the importance of v_j in predicting the personalized bridge function. For the target domain, an irrelevant item would has little help for personalized bridge functions of all users. Thus, we learn the attention score from the items' embeddings by an attention network. Formally, the attention network is defined as:

$$\begin{aligned} a_j' &= h(\boldsymbol{v}_j; \theta), \\ a_j &= \frac{\exp(a_j')}{\sum_{v_l^s \in \mathcal{S}_{u_i}} \exp(a_l')}, \end{aligned} \tag{2}$$

where $h(\cdot)$ denotes the attention network, and θ denotes the parameters of $h(\cdot)$. In this paper, $h(\cdot)$ is a two-layer feed-forward network. Note that the normalized attention score a_j is beneficial to find the useful interacted items for a specific user. After that, we could utilize each user's characteristics as input to guide the generation of personalized bridge function.
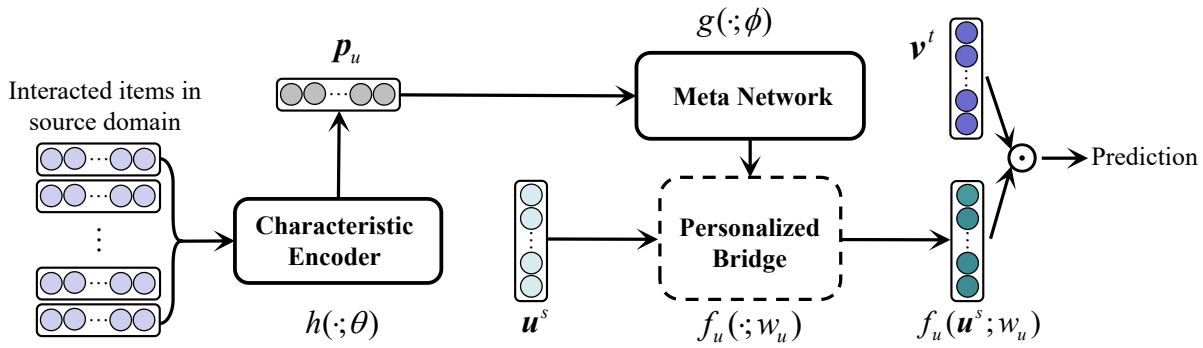
Figure 2: Personalized Transfer of User Preferences for Cross-domain Recommendation (PTUPCDR) utilizes a meta network with users' characteristic embeddings in the source domain as input to generate personalized bridge functions for each user. Then, with the personalized bridge function, we can obtain the transformed user's embeddings as the initial embeddings.

3.3 Meta Network

We have mentioned that the users' relationships between preferences of different domains vary from user to user. In other words, the process of preference transfer needs to be personalized. Intuitively, there exists a certain connection between the preference relationship and the user's characteristics. Based on this intuition, we propose a meta network which takes the user's transferable characteristics as input, and then generates a personalized bridge function between the user's embeddings in the source and target domains. The proposed meta network is formulated as:

$$w_{u_i} = g(p_{u_i}; \phi), \tag{3}$$

where $g(\cdot)$ is the meta network, which is parameterized by ϕ. In this paper, the meta network is a two-layer feed-forward network. The w_{u_i} is a vector whose size depends on the structure of the bridge function. The personalized bridge function is formulated as:

$$f_{u_i}(\cdot; w_{u_i}), \tag{4}$$

which utilizes w_{u_i} as the parameters of bridge function $f(\cdot)$. The bridge function can be defined as any structure. In this paper, for simplicity, we use a linear layer as $f(\cdot)$ following EMCDR [9, 16]. Thus, to fit the size of bridge's parameters, we reshape the vector $w_{u_i} \in \mathbb{R}^{k^2}$ into a matrix $w_{u_i} \in \mathbb{R}^{k \times k}$. Note that the w_{u_i} is used as the parameters of the bridge functions rather than input. The generated bridge function depends on user's characteristics and varies from user to user, and we call it the personalized bridge function.

With the personalized bridge function, we can obtain the personalized transformed user's embeddings:

$$\hat{u}_i^t = f_{u_i}(u_i^s; w_{u_i}), \tag{5}$$

where u_i^s denotes the embedding of user u_i in the source domain, and $\hat{u}_i^t$ represents the transformed embedding. Finally, we can utilize the transformed embedding $\hat{u}_i^t$ for prediction.

3.4 Task-oriented Optimization

To train the meta network and characteristic encoder, we can minimize the distance using the **mapping-oriented optimization**

procedure following existing bridge-based methods [4, 9, 16, 39]:

$$\mathcal{L} = \sum_{u_i \in \mathcal{U}^o} ||\hat{u}_i^t - u_i^t||^2, \tag{6}$$

where $\hat{u}_i^t$ denotes the transformed user embedding from u_i^s in the source domain, and u_i^t denotes the user embedding in target domain. The mapping-oriented optimization procedure would bring the transformed embedding $\hat{u}_i^t$ close to the target embedding u_i^t.

However, since some users only have limited interactions, the user's embedding u_i^t may be not reasonable and accurate enough. Learning towards the relatively unreasonable embeddings would lead to negative impact on the model. Thus, we propose a task-oriented optimization to train the meta network and characteristic encoder. The task-oriented training procedure directly utilizes the performance of the ultimate recommendation task as the optimization goal. In this paper, we focus on rating task, so the task-oriented loss can be formulated as:

$$\min_{\theta, \phi} \frac{1}{|\mathcal{R}_o^t|} \sum_{r_{ij} \in \mathcal{R}_o^t} (r_{ij} - f_{u_i}(u_i^s; w_{u_i})v_j)^2, \tag{7}$$

where $\mathcal{R}_o^t = \{r_{ij} | u_i \in \mathcal{U}^o, v_j \in \mathcal{V}^t\}$ denotes the interactions of overlapping users in the target domain.

Compared with the mapping-oriented procedure, task-oriented optimization has two advantages: (1) The task-oriented optimization can alleviate the effects of unreasonable embeddings. It directly uses the rating data, which is ground truth rather than approximate intermediate results. (2) The task-oriented learning procedure has more training samples, which can avoid overfitting. For example, with N overlapping users, each user has M ratings. The mapping-oriented process learns the mapping function with $|\mathcal{U}^o = N|$ samples as Equation (6), while the task-oriented learning procedure utilizes the $|\mathcal{R}_o^t| = M \times N$ user-item rating as Equation (7).

3.5 Overall Procedure

The overall structure of PTUPCDR is shown in Figure 2. The training procedure can be divided into three steps: pre-training, meta and initialization stages, as see Algorithm 1. After training, the method can work for both cold-start and warm-start stages.

Table 1: Statistics of the cross-domain tasks (*Overlap* denotes the number of overlapping users).

CDR Tasks	Domain		Item		User			Rating	
	Source	Target	Source	Target	Overlap	User Source	Target	Source	Target
Task1	Movie	Music	50,052	64,443	18,031	123,960	75,258	1,697,533	1,097,592
Task2	Book	Movie	367,982	50,052	37,388	603,668	123,960	8,898,041	1,697,533
Task3	Book	Music	367,982	64,443	16,738	603,668	75,258	8,898,041	1,097,592

Pre-training stage: This step is to learn latent spaces for each domain, respectively. The loss function is formulated as:

$$\min_{\boldsymbol{u},\boldsymbol{v}} \frac{1}{|\mathcal{R}|} \sum_{r_{ij} \in \mathcal{R}} (r_{ij} - \boldsymbol{u}_i \boldsymbol{v}_j)^2, \tag{8}$$

where $|\mathcal{R}|$ denotes the number of ratings. After the pre-training step, we can obtain the pre-trained embeddings $\boldsymbol{u}^s, \boldsymbol{u}^t, \boldsymbol{v}^s, \boldsymbol{v}^t$.

Meta stage: The existing methods directly train a common bridge function, while PTUPCDR trains the characteristic encoder and the meta network. The characteristic encoder and the meta network are optimized with Equation (7).

Initialization stage: When a new user comes (CDR assumes the new user has some interactions in the source domain), we use the transformed embedding $\hat{\boldsymbol{u}}_i^t = f_{u_i}(\boldsymbol{u}_i^s; \boldsymbol{w}_{u_i})$ to initialize the new user's embedding in the target domain.

Test stage: For the extreme cold-start users who have no interactions in the target domain, directly utilize the initial embedding $\hat{\boldsymbol{u}}_i^t = f_{u_i}(\boldsymbol{u}_i^s; \boldsymbol{w}_{u_i})$ for prediction. For the warm-start users who have some interaction in the target domain, it is convenient to fine-tune the initial embeddings with new interactions, and utilize the fine-tuned embeddings for prediction.

4 EXPERIMENTS

We conduct experiments to answer the following research questions: **RQ1** Why we need an auxiliary domain and why we need to introduce CDR? How does PTUPCDR perform in extremely cold-start scenarios comparing to state-of-the-art models with a CDR perspective? **RQ2** How does PTUPCDR perform in more practical scenarios of real-world recommendations? **RQ3** Why could PTUPCDR perform better?

4.1 Experimental Settings

Datasets. Following most existing methods [9, 37, 40], a real-world public dataset is adopted for experiments, namely the Amazon review dataset [1]. Specifically, we use the Amazon-5cores dataset in which each user or item has at least five ratings.

Following [9, 37], we choose 3 popular categories out of 24 in total: movies_and_tv (Movie), cds_and_vinyl (Music), and books (Book). We define 3 CDR tasks as Task 1: Movie → Music, Task 2: Book → Movie, and Task 3: Book → Music. As the details listed in Table 1, the number of ratings of the source domain is significantly large than the one in the target domain. While many existing works only select a part of the dataset for evaluation, we directly use all data to simulate the real-world application.

Evaluation Metrics. Amazon review dataset contains rating data (0 - 5 score). Following [16, 37] we adopt Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as the metrics.

Baselines. Since PTUPCDR falls into the bridge-based methods for CDR, we mainly compare PTUPCDR with the bridge-based methods. Therefore, we choose the following methods as baselines for comparison. 1) TGT denotes the target MF model, which is trained only using target domain data. 2) CMF [25] is an extension of MF. In CMF, the embeddings of users are shared across the source and target domains. 3) EMCDR [16] is a popular CDR method for cold-start. It adopts Matrix Factorization (MF) to learn embeddings first and then utilize a network to bridge the user embeddings from the auxiliary domain to the target domain. 4) DCDCSR [39] falls into the bridge-based methods, which considers the rating sparsity degrees of individual users in different domains. 5) SSCDR [9] is a semi-supervised bridge-based method.

Implementation Details. We implement our framework and the baselines using PyTorch. For each task and method, the initial learning rate for the Adam [10] optimizer are tuned by grid searches within {0.001, 0.005, 0.01, 0.02, 0.1}. In addition, we set the dimension of embeddings as 10. For all methods, we set mini-batch size of 512. We employ the same fully connected layer to facilitate comparison for the cross-domain bridge functions of EMCDR, DCDCSR, SSCDR, and PTUPCDR. Note that the personalized bridge function of PTUPCDR is generated by the meta network. The meta network is a two-layer network with hidden units $2 \times k$, where k denotes the embedding dimension, and the output's dimension of the meta network is $k \times k$. The attention network is a two-layer network with k hidden units.

Following [16], to evaluate the effectiveness of the proposed PTUPCDR on cross-domain recommendation, we randomly remove all the ratings of a fraction of overlapping users in the target domain and regard them as test users, and the samples of other overlapping users are used for training the bridge function. In our experiments, we set the proportions of test (cold-start) users β as 80%, 50%, and 20% of the total overlapping users, respectively. For the cold-start experiments in Table 2 and Figure 3, all ratings of test users are used as the test set. For the warm-start experiments in Figure 4, we divide the ratings of each test user into two parts with a ratio of 1:1. Note that we take the sequential timestamps into account to avoid information leakage. We use the first part as the cold-start set, and the other as the warm-start set. The process of model evaluation can be divided into three step: 1) Train the model on the training set. 2) Test extremely cold-start performance on the cold-start set. 3) Fine-tune the target model with the cold-start set, and evaluate the warm-start performance on the warm-start set. For each task, we report the mean results over five random runs.

[1] http://jmcauley.ucsd.edu/data/amazon/

Table 2: Cold-start results (MAE and RMSE) of 3 cross-domain tasks. We report the mean results over five runs. Best results are in boldface. ∗ indicates 0.05 **level, paired t-test of PTUPCDR vs. the best baselines.** *Improve* **denotes relative improvement over the best baseline.**

	β		Metric	TGT	CMF	DCDCSR	SSCDR	EMCDR	PTUPCDR	Improve
Task1	20%		MAE	4.4803	1.5209	1.4918	1.3017	1.2350	**1.1504***	6.86%
			RMSE	5.1580	2.0158	1.9210	1.6579	1.5515	**1.5195**	2.06%
	50%		MAE	4.4989	1.6893	1.8144	1.3762	1.3277	**1.2804***	3.57%
			RMSE	5.1736	2.2271	2.3439	1.7477	1.6644	**1.6380**	1.59%
	80%		MAE	4.5020	2.4186	2.7194	1.5046	1.5008	**1.4049***	6.39%
			RMSE	5.1891	3.0936	3.3065	1.9229	1.8771	**1.8234***	2.86%
Task2	20%		MAE	4.1831	1.3632	1.3971	1.2390	1.1162	**0.9970***	10.68%
			RMSE	4.7536	1.7918	1.7346	1.6526	1.4120	**1.3317***	5.69%
	50%		MAE	4.2288	1.5813	1.6731	1.2137	1.1832	**1.0894***	7.93%
			RMSE	4.7920	2.0886	2.0551	1.5602	1.4981	**1.4395***	3.91%
	80%		MAE	4.2123	2.1577	2.3618	1.3172	1.3156	**1.1999***	8.80%
			RMSE	4.8149	2.6777	2.7702	1.7024	1.6433	**1.5916***	3.15%
Task3	20%		MAE	4.4873	1.8284	1.8411	1.5414	1.3524	**1.2286***	9.15%
			RMSE	5.1672	2.3829	2.2955	1.9283	1.6737	**1.6085***	3.90%
	50%		MAE	4.5073	2.1282	2.1736	1.4739	1.4723	**1.3764***	6.51%
			RMSE	5.1727	2.7275	2.6771	1.8441	1.8000	**1.7447***	3.07%
	80%		MAE	4.5204	3.0130	3.1405	1.6414	1.7191	**1.5784***	3.84%
			RMSE	5.2308	3.6948	3.5842	2.1403	2.1119	**2.0510***	2.88%

4.2 Cold-start Experiments (RQ1)

This section presents experimental results and in-depth discussions of PTUPCDR on cold-start scenarios. Following the existing bridge-based methods [4, 9, 16, 37, 39, 40], we evaluate the performance of PTUPCDR on cold-start scenarios. We demonstrate the effectiveness of PTUPCDR on 3 CDR tasks under different values of β. As the experimental results are shown in Table 2, the best performance is shown in boldface, ∗ indicates 0.05 level paired t-test of PTUPCDR vs. the best baseline, *Improve* denotes relative improvement over the best baseline. From the experimental results, we have several findings: (1) TGT is a single-domain model that only uses data from the target domain, and its performance is unsatisfying. Compared with TGT, all other cross-domain methods could exploit data from the source domain, thus achieving better results. Therefore, utilizing data from an auxiliary domain is an effective way to alleviate data sparsity and improve the recommendation performance in the target domain. (2) CMF uses the auxiliary data by combining the data from different domains into a single domain, while CDR methods are specially designed to bridge the domains. We find that CDR methods can outperform CMF in most tasks. It is because CMF ignores the potential domain shift by regarding the data from both domains as the same. On the contrary, the bridge functions can transform the source embeddings into the target feature space, which effectively alleviates the influence of domain shift. Thus, it is essential to study CDR by using the auxiliary domain more effectively. (3) We find that PTUPCDR could outperform the best baseline significantly in most scenarios, which demonstrates that PTUPCDR is effective for cold-start recommendation.

4.3 More Practical Scenarios (RQ2)

In this section, we perform experiments and analysis on the compatibility of PTUPCDR with more complicated base models and its utility in the warm-start stage.

4.3.1 Generalization Experiments: Note that the bridge-based CDR methods [16, 39] focus on the bridge function itself, and works in the literature mainly apply their methods upon MF to conduct experimental evaluations. However, MF is a non-neural model, and it is probably too simple to achieve satisfying performance in large-scale real-world recommendations. Thus, to testify the compatibility of PTUPCDR as well as other bridge-based methods, we apply EMCDR and our PTUPCDR upon two more complicated neural models. In other words, we use other models to replace the MF: GMF [7] and YouTube DNN [1]. GMF assigns various weights for different dimensions in the dot-product prediction function, which can be regarded as a generalization of vanilla MF. YouTube DNN is a two-tower model. For GMF, the bridge function directly transforms the user embeddings. For YouTube DNN, the bridge function transforms the output of the user tower. For both GMF and YouTube DNN, we train the model with data from both domains. With $\beta = 0.2$, we conduct the generalization experiments on both non-neural (MF) and neural models (GMF, YouTube DNN). Other experimental settings are consistent with Section 4.1. From the results shown in Figure 3(a)(b)(c), we have several insightful observations: (1) The bridge-based CDR methods can be applied upon various base models. With different base models, both EMCDR and PTUPCDR effectively improve the recommendation performance
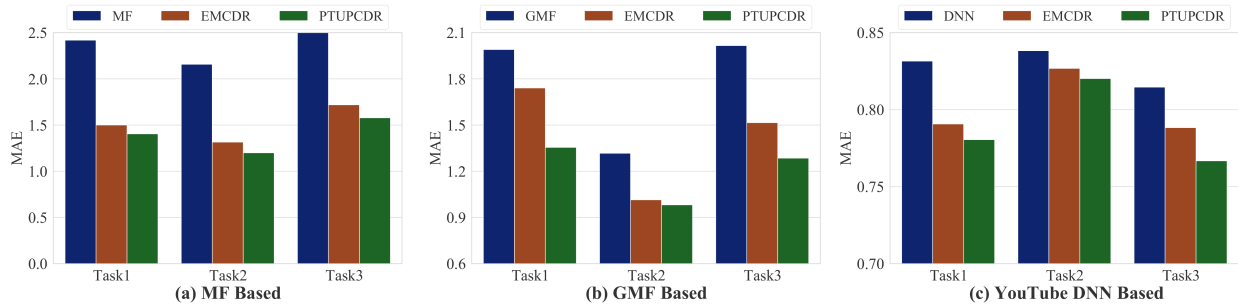
Figure 3: Generalization experiments: applying EMCDR and PTUPCDR upon three base models (a) MF, (b) GMF, and (c) YouTube DNN, and show the averaged results over five runs.
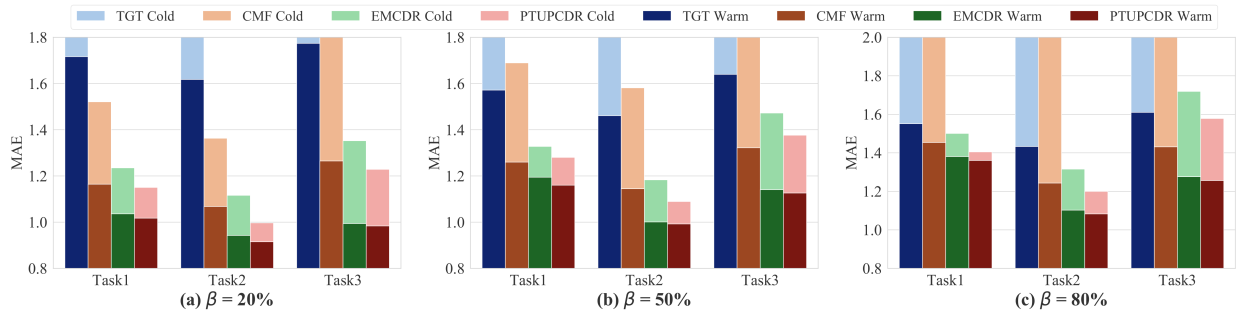

Figure 4: Warm-start experiments on TGT, CMF, EMCDR, and PTUPCDR for different proportions of test (cold-start) users β: (a) $\beta = 20\%$, (b) $\beta = 50\%$, and (c) $\beta = 80\%$. The light-colored histograms represent the performance of extreme cold-start scenario, while the dark-colored histograms represent the warm-start scenario.

for cold-start users in the target domain. Since GMF and YouTube DNN are two popular and well-designed models in large-scale real-world recommendations, they achieve better performance than the vanilla MF. (2) The generalized PTUPCDR could achieve satisfying performance. On the one hand, with various base models, the generalized PTUPCDR can constantly achieve the best results. On the other hand, as the cold-start problem is highly challenging, the achieved *MAE* is good enough to testify the effectiveness of generalized PTUPCDR in real-world scenarios.

4.3.2 Warm-start Experiments. The existing bridge-based CDR works in the literature [4, 9, 16, 37, 39] only conduct experiments on the extreme cold-start stage. Actually, bridge-based CDR methods are also highly helpful for the warm-start stage by using the mapped embeddings to initialize cold-start users' embeddings of TGT for further training. In real-world recommendations, such warming-up scenarios [21, 42] have great application value.

We conduct experiments on TGT, CMF, EMCDR, and our proposed PTUPCDR. In the warm-start training stage, i.e., warm-up process, CMF, EMCDR, and PTUPCDR can be viewed as pre-trained models for initialization. For CMF, we use the collectively trained embeddings to initialize both user and item embeddings of TGT. For EMCDR and PTUPCDR, we initialize the cold-start users' embeddings with mapped users' embeddings. From the results shown in Figure 4, we have the following observations:

Cold-start vs. Warm-start. More interactions can improve the performance of recommender systems. We find that all models

in the warm-start stage can achieve better performance than the cold-start stage, demonstrating that more interactions can help recommendation models understand the users better.

Utility. In the warm-start stage, with pre-trained embeddings as the initial embeddings, CMF, EMCDR, PTUPCDR can achieve better performance than TGT, which uses randomly initialized embeddings. PTUPCDR and EMCDR outperform CMF, demonstrating that embeddings pre-trained by bridge-based CDR methods could better exploit the source domain. Such utility of CDR methods on real-world warm-start scenarios is of great practical value.

Performance. In the warm-start stage, our PTUPCDR can still achieve the best results with various test ratio of β. Therefore, our PTUPCDR is useful and effective in both the cold-start stage and the warm-start stage.

4.4 Explanation of the Improvement (RQ3)

In this section, we conduct extensive experiments and present insightful discussions concerning three modules of PTUPCDR to explain the improvement brought by PTUPCDR and answer *RQ*3.

Latent Factor Visualization. We analyze embeddings on the target-domain feature space to further investigate the reason why PTUPCDR outperforms EMCDR and to show the capacity of the Meta Network to generate personalized bridge functions.

We employ the default setting of the t-SNE [2] in Scikit-learn to visualize the user embeddings learned by EMCDR and PTUPCDR on Task3 with $\beta = 0.2$. Figure 5 (a) and (b) denote the embeddings of
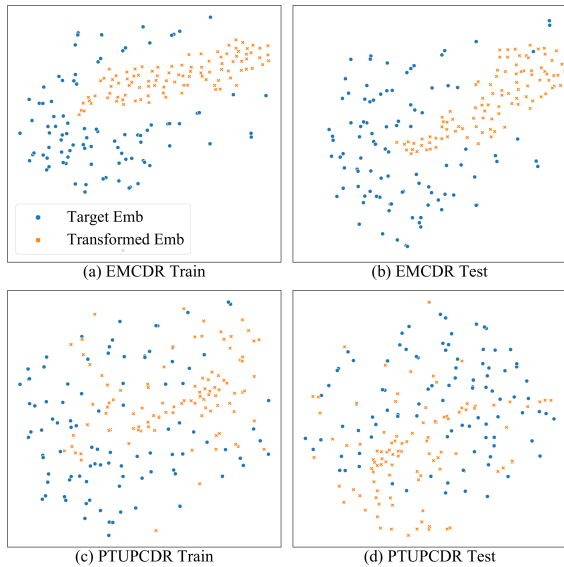
Figure 5: t-SNE visualization of randomly sampled user embeddings in target-domain feature space and transformed user embeddings. (a) and (b), (c) and (d) denotes the visualization results of EMCDR and PTUPCDR, respectively.

training and test users by EMCDR, while the visualized embeddings in Figure 5 (c) and (d) are learned by PTUPCDR. The blue points denote the target embeddings taken from the target model learned with both training and test users and are regarded as ground truths, while the orange points represent the transformed embeddings. For clarity, we randomly sample 100 training and test users respectively to plot. Note that PTUPCDR and EMCDR share the source and target models, and the only difference is whether the bridge function is personalized by our PTUPCDR or learned by EMCDR.

Ideally, the distributions of the transformed embeddings are the same as the target embeddings. From Figure 5 (a) and (b), we observe that the target embeddings (ground truths) are scattered, while the embeddings transformed by EMCDR are very concentrated. The main reason would be that the single bridge function is hard to capture the complex relationships between users' preferences in the source and target domains. As shown in Figure 5(c) and (d), PTUPCDR achieves better results. For one thing, the transformed embeddings by PTUPCDR are scattered across the target-domain feature space instead of being clustered as EMCDR, demonstrating the personalization capacity of the Meta Network and the PTUPCDR. More importantly, the distribution of embeddings transformed by PTUPCDR could better fit the target embeddings distributions, which could be the fundamental reason why PTUPCDR could achieve better overall performance.

Case Study. We present a case study to discuss the necessity and effectiveness of the attention-based Characteristic Encoder. As shown in Figure 6, our goal is to recommend CDs to a user who has not purchased any CD before, with the help of interacted movies of that user. Note that the darker color block below a movie represents a higher degree of predicted attention over that movie, and the shown 3 CDs are successfully recommended to this user.
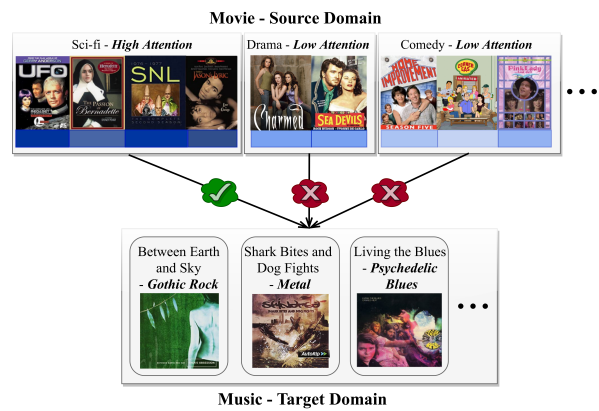
Figure 6: The color block below each movie represents the attention score. High attention items dominate the recommendation while others have little influence on results.

Those successfully recommended hard rock records are somewhat related to interacted science fiction films because they are both exciting. However, hard rock records are almost irrelevant to dramas and comedies. Thus, it is evident that the importance of different interacted items in the source domain should be modeled appropriately. Thus, it is necessary to adopt the attention mechanism to evaluate the items' different contributions to knowledge transfer automatically. At the same time, although the consumed CDs are related to only part of historical interacted movies, the proposed model still could provide relatively accurate recommendations regardless of the influence of noise from dramas and comedies, which demonstrates the effectiveness of the attention-based Characteristic Encoder. To summarize, the attention-based Characteristic Encoder could capture transferable individual characteristics, while existing bridge-based CDR methods ignore this point.

5 CONCLUSION

In this paper, we studied cross-domain recommendation (CDR) which aims to transfer user preferences from an auxiliary domain to the target domain. Many existing CDR methods learn a common preference bridge to transfer preferences. However, a single bridge function shared by all users is hard to capture various relationships between user preferences in source and target domains. Thus, we proposed a novel framework named Personalized Transfer of User Preferences for CDR (PTUPCDR). Specifically, a meta network fed with users' characteristic embeddings is learned to generate personalized bridge functions to achieve personalized transfer of user preferences. We conducted extensive experiments on real-world datasets to evaluate the proposed PTUPCDR, and the results validate the effectiveness of PTUPCDR on both cold-start and warm-start stages.

6 ACKNOWLEDGMENTS

REFERENCES

[1] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Recsys*. 191–198.

[2] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*. 647–655.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*. JMLR. org, 1126–1135.

[4] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation Systems. In *AAAI*, Vol. 33. 94–101.

[5] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial Feature Translation for Multi-domain Recommendation. In *KDD*. 2964–2973.

[6] Ming He, Jiuling Zhang, Peng Yang, and Kaisheng Yao. 2018. Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation. In *WSDM*. 225–233.

[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[8] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *CIKM*. 667–676.

[9] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *CIKM*. 1563–1572.

[10] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*, Vol. 5.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[12] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *KDD*. 1073–1082.

[13] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *AAAI*, Vol. 33. 4189–4196.

[14] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*. 661–670.

[15] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In *WSDM*. 331–339.

[16] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach.. In *IJCAI*. 2464–2470.

[17] Kaixiang Mo, Bo Liu, Lei Xiao, Yong Li, and Jie Jiang. 2015. Image feature learning for cold start problem in display advertising. In *IJCAI*.

[18] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *ICML*. 2554–2563.

[19] Qiong Nan, Juan Cao, Yongchun Zhu, Yanyan Wang, and Jintao Li. 2021. MD-FEND: Multi-domain Fake News Detection. In *CIKM*. 3343–3347.

[20] Feiyang Pan, Qingpeng Cai, Pingzhong Tang, Fuzhen Zhuang, and Qing He. 2019. Policy gradients for contextual recommendations. In *WWW*. 1421–1431.

[21] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *SIGIR*. 695–704.

[22] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*.

[23] Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. 2011. Personalised rating prediction for new users using latent factor models. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*. 47–56.

[24] Falong Shen, Shuicheng Yan, and Gang Zeng. 2018. Neural style transfer via meta networks. In *CVPR*. 8061–8069.

[25] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *KDD*. 650–658.

[26] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*. 4077–4087.

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NeurIPS*. 3630–3638.

[28] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*. 417–426.

[29] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. 2018. Visual domain adaptation with manifold embedded distribution alignment. In *ACMMM*. 402–410.

[30] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. 2021. Generalizing to Unseen Domains: A Survey on Domain Generalization. *arXiv preprint arXiv:2103.03097* (2021).

[31] Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the Sequential Dependence among Audience Multi-step Conversions with Multi-task Learning in Targeted Display Advertising. In *KDD*.

[32] Dongbo Xi, Fuzhen Zhuang, Ganbin Zhou, Xiaohu Cheng, Fen Lin, and Qing He. 2020. Domain adaptation with category attention network for deep sentiment analysis. In *WWW*. 3133–3139.

[33] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *IJCAI*.

[34] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2021. Contrastive Cross-domain Recommendation in Matching. *arXiv preprint arXiv:2112.00999* (2021).

[35] Ruobing Xie, Zhijie Qiu, Jun Rao, Yi Liu, Bo Zhang, and Leyu Lin. 2020. Internal and Contextual Attention Network for Cold-start Multi-channel Matching in Recommendation.. In *IJCAI*. 2732–2738.

[36] Yinan Zhang, Yong Liu, Peng Han, Chunyan Miao, Lizhen Cui, Baoli Li, and Haihong Tang. 2020. Learning Personalized Itemset Mapping for Cross-Domain Recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. 2561–2567.

[37] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-Domain Recommendation for Cold-Start Users via Aspect Transfer Network. In *SIGIR*. ACM, 229–238.

[38] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*. 1059–1068.

[39] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet Orgun, and Jia Wu. 2018. A deep framework for cross-domain and cross-system recommendations. In *IJCAI*.

[40] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-Meta Framework for Cross-domain Recommendation to Cold-Start Users. In *SIGIR*.

[41] Yongchun Zhu, Yudan Liu, Ruobing Xie, Fuzhen Zhuang, Xiaobo Hao, Kaikai Ge, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Expand Audience via Meta Hybrid Experts and Critics for Recommendation and Advertising. In *KDD*.

[42] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks. In *SIGIR*.

[43] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. 2019. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *AAAI*, Vol. 33. 5989–5996.

[44] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. 2021. Deep subdomain adaptation network for image classification. *TNNLS* 32, 4 (2021), 1713–1722.

[45] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.