REMIT: Reinforced Multi-Interest Transfer for Cross-Domain Recommendation

Caiqi Sun[1], **Jiewei Gu**[1,2] *, **BinBin Hu**[1], **Xin Dong**[1], **Hai Li**[1], **Lei Cheng**[1], **Linjian Mo**[1]

[1] Ant Group
[2] School of Data Science, Fudan University
{caiqi.scq, gujiewei.gjw, bin.hbb, zhaoxin.dx, tianshu.lh, lei.chenglei, linyi01}@antgroup.com

Abstract

Cold-start problem is one of the most challenging problems for recommender systems. One promising solution to this problem is cross-domain recommendation (CDR) which leverages rich information from an auxiliary source domain to improve the performance of recommender system in the target domain. In particular, the family of embedding and mapping methods for CDR is very effective, which explicitly learns a mapping function from source embeddings to target embeddings to transfer users' preferences. Recent works usually transfer an overall source embedding by modeling a common or personalized preference bridge for all users. However, a unified user embedding cannot reflect the user's multiple interests in the auxiliary source domain. In this paper, we propose a novel framework called **RE**inforced **M**ulti-**I**nterest **T**ransfer for CDR (REMIT). Specifically, we first construct a heterogeneous information network and employ different meta-path based aggregations to get users' multiple interests in the source domain, then transform different interest embeddings with different meta-generated personalized bridge functions for each user. To better coordinate the transformed user interest embeddings and the item embedding in the target domain, we systematically develop a reinforced method to dynamically assign weights to transformed interests for different training instances and optimize the performance of the target model. In addition, the REMIT is a general framework that can be applied upon various base models in the target domain. Our extensive experimental results on large real-world datasets demonstrate the superior performance and compatibility of REMIT.

Introduction

Recommender systems have been widely applied to many online services such as e-commerce, advertising, and social media to perform personalized information filtering (Covington, Adams, and Sargin 2016; Sun et al. 2022). However, most of these recommender systems are hard to provide satisfying recommendations for newly joining users and newly arriving items, forming the so-called cold-start problem.

Cross-domain recommendation (CDR) is a promising solution to address the cold-start problem, which aims to transfer knowledge from an informative source domain to the target domain. To address the problem, the Embedding and Mapping approach for CDR (EMCDR) (Man et al. 2017) is very effective, which encodes users' preferences of source and target domains on items into two embeddings, respectively, and then explicitly learns a bridge function from source embedding to target embedding with overlapping users. With the advantage of EMCDR, many bridge-based approaches (Kang et al. 2019; Zhu et al. 2020, 2022) have been proposed. For these methods, the key is to bridge users' preferences in the source domain and the target domain, also called preference transfer (Zhao et al. 2020). However, these methods give an overall source embedding for each user, which is hard to represent multiple interests in the source domain.

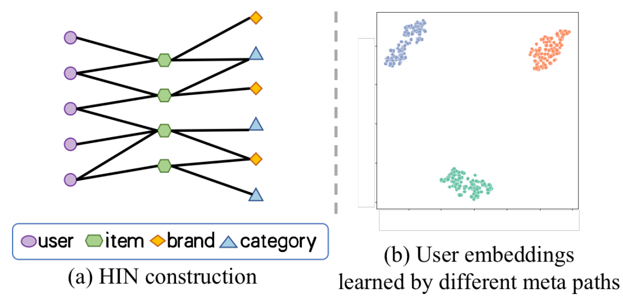


Figure 1: (a) gives an example of HIN. (b) plots user embeddings learned from meta-path *uiu*(purple), *uiciu*(green), and *uibiu*(orange), which capture users' interests.

More recently, Heterogeneous Information Networks (HIN) (Shi et al. 2016) have been leveraged to enrich user-item interactions with complementary heterogeneous information. As shown in Figure 1(a), a toy HIN can be constructed for movie recommendation, which captures how the items(movies) are related with each other via brand(actors) and category(genres). On the HIN, higher-order graph structures like meta-paths (Sun et al. 2011), a relation sequence connecting two objects, can effectively capture users' interests. For instance, the meta-path user–item–brand–item–user($uibiu$) incorporates movies starring the same actor as a facet of user preferences, which makes sense since the user might be a fan of an actor and prefers most movies played by the actor. Given different meta-paths, we can obtain multiple

*Work is done during internship at Ant Group.

interest embeddings for user nodes. For example, in Figure 1(b), we find users have clearly three interests when we set meta-path number to three. Intuitively, it is not necessary to assign the same weight to different interests when predicting a candidate item because only part of users' interests will influence his/her action. For example, a male basketball player will click a recommended NBA-themed phone case in the target domain mostly due to interests in NBA and smartphones, rather than cars in the source domain. Based on the above insights, there are two challenges in this paper. 1) How to transfer different source interest embeddings. 2) How to select different transformed user interest embeddings for each instance in the target domain task.

To address the challenges above, we propose a novel reinforced multi-interest transfer framework for CDR, dubbed REMIT. After constructing a heterogeneous information network and employing different meta-path based aggregations to get users' multiple interests in the source domain. We transform different interest embeddings with multiple preference bridges for each user, and each bridge is personalized at the user level using a meta network fed with users' characteristic embeddings, and employ a task-oriented optimization procedure to learn the meta network stably. For better performance in the target domain, we investigate how to assign appropriate weights to different transformed interests on various training samples. We formulate the interest selection problem under a reinforcement learning framework: the decision of the interest-selector agent is made based on the characteristics of training examples and the outputs of multiple personalized bridges, while the policy is learned towards maximizing the target domain performance as the return. Finally, we use the weights got from the agent to aggregate one transformed user embedding, which can be utilized as the initial embedding for the cold-start user in the target domain. To summarize, the main contributions of this paper are as follows:

- To the best of our knowledge, this is the first work to use multiple personalized bridges to transfer different source interests for different users in the family of embedding and mapping methods for CDR.
- A novel reinforcement learning framework is proposed to formulate the multi-interest selection problem, which dynamically assigns weights to transformed interests for different training instances.
- Extensive experiments show that REMIT achieves significant improvements over state-of-the-art models and can be applied upon various base models in the target domain. Further analysis presents insightful discussions concerning the necessity and effectiveness of the multi-personalized bridges and reinforced interest selector.

Related Work

Cross-domain recommendation (CDR) (Fu et al. 2019; Gao et al. 2021; Kang et al. 2019; Man et al. 2017), which aims to improve the recommendation performance by means of transferring information from the auxiliary domain to the target domain, is one of the promising ways to solve data sparsity and cold start problem. At the very beginning, CMF (Singh and Gordon 2008) proposes to achieve knowledge integration across domains by concatenating multiple rating matrices and sharing user factors across domains. In recent years, researchers proposed many deep learning-based models to enhance knowledge transfer (Hao et al. 2021; He et al. 2018; Hu, Zhang, and Yang 2018; Li and Tuzhilin 2020; Xi et al. 2021). CoNet (Hu, Zhang, and Yang 2018) is proposed to train a deep cross-stitch network for enhancing the recommendation on both domains simultaneously. MINDTL (He et al. 2018) combines the CF information of the target domain with the rating patterns extracted from a cluster-level rating matrix in the source domain. DDTCDR (Li and Tuzhilin 2020) 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 focuses on bridging user preferences in different domains (Kang et al. 2019; Man et al. 2017; Pan et al. 2010; Zhao et al. 2020; Zhu et al. 2020, 2021), which is the most related work. EMCDR (Man et al. 2017) explicitly maps user representations from different domains via a multi-layer fully connected neural network. With the advantage of EMCDR, many EMCDR-based approaches(Kang et al. 2019; Zhao et al. 2020; Zhu et al. 2020, 2022) have been proposed to explicitly model the preference bridge, which falls into four main types, e.g., improving the source embeddings representation capabilities (Zhao et al. 2020), handling situations where only a small amount of labeled data is available (Kang et al. 2019; Zhu et al. 2020), dealing with the inconsistency between the goals of the mapping function task and the task in target domain (Zhu et al. 2021), personalized modeling of mapping functions (Zhu et al. 2022). Our study falls into this bridge-based category. However, to the best of our knowledge, all of the bridge-based CDR methods only transfer an overall source embedding to the target domain for each user, while our REMIT is the first to learn multiple source interest embeddings transfer and selection for each user.

Proposed Method

In this section, we first introduce the problem formulation and then discuss the proposed framework in detail.

Problem Formulation

Assume we have a set of users $u \in U^d$, a set of items $i \in I^d$ and a set of ratings $r_{ui} \in R^d$, where $d \in \{s, t\}$ represents the source domain s and target domain t respectively. For each user u in source domain, we have his/her historical behavior sequence $S_u = \{i_1^s, i_2^s, \cdots i_n^s\}$, where n denotes the number of interacted items and i_t^s represents the t-th item interacted by user u in source domain. Each item i has its corresponding category $c_i \in C^s$ and brand $b_i \in B^s$. We define the overlapping users between the two domains as $U^o = U^s \cap U^t$. In contrast, I^s and I^t are disjoint, which means there is no shared item between the two domains. The pre-trained embeddings of user and item in target domain is $\mathbf{u}^t \in R^{1\times k}$ and $\mathbf{i}^t \in R^{1\times k}$ respectively, where k is the dimension of embeddings. Given rich behavioral data of the

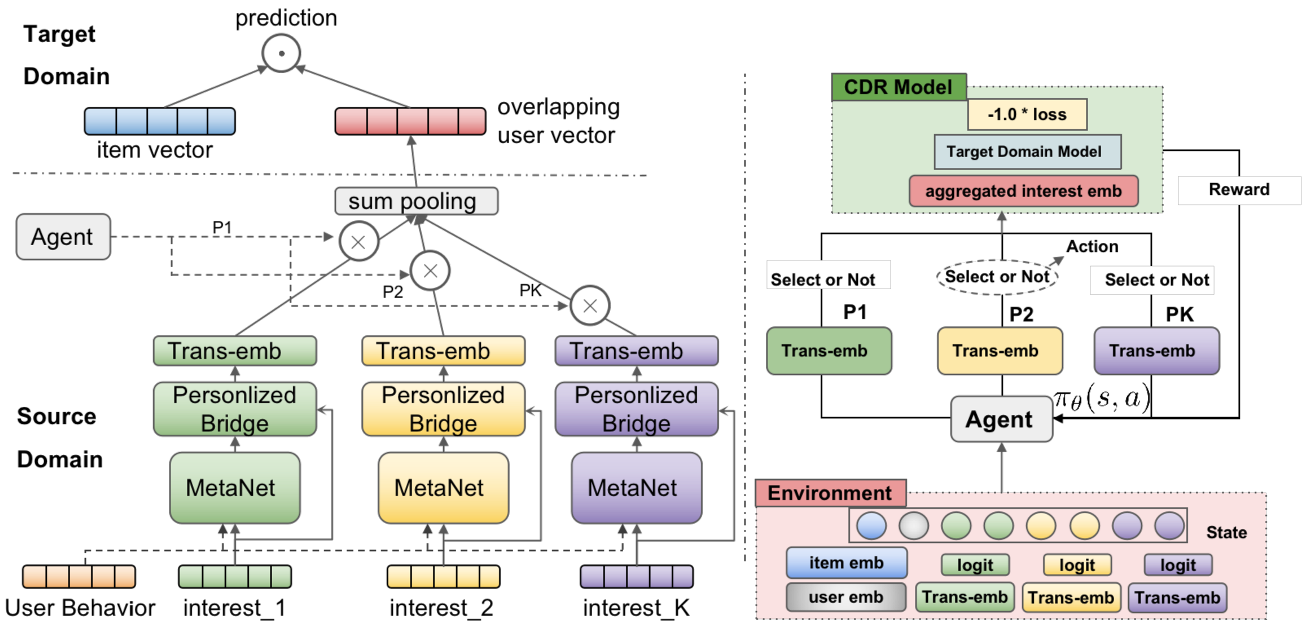


Figure 2: Details of REMIT. (a) illustrates the main procedure of REMIT, where users' source multiple interests are transformed by multi-bridge functions, personalized by the shared meta network with different inputs. Then, with interest selection agent's sampling probabilities, we can obtain the aggregated interest embedding as the initial embedding for each overlapping user in target domain and make prediction. (b) shows how interest selection agent works from a reinforcement learning perspective.

overlapping users in the source domain and the pre-trained embeddings in the target domain, the problem of bridge-based methods in CDR is to transfer users' interests from the source domain to the target domain and predict the item that cold-start user may interact with in target domain by using the transformed interest embedding.

Traditional bridge-based methods mainly rely on one overall source embedding as input for interest transfer. Our paper focuses on multi-interest embeddings transformation and selection. Given pre-trained multiple interest embeddings in source domain, we use multiple personalized bridges to transfer multiple interests. For each cold-start user in the target domain, we focus on leveraging reinforcement learning to select interests for improving the performance of the target domain task. The general network structure is illustrated in Figure 2. We first present the approach for multi-interest transfer in subsection , then illustrate reinforced interest selection method in subsection . Finally, we introduce the model training algorithm in subsection .

Multi-Interest Transfer (MIT)

We follow the HIN construction and meta-path based aggregation of HCDIR (Bi et al. 2020) to obtain the pre-trained user interests and item embeddings in the source domain. Specially, a HIN contains four types of nodes: users U^s, items I^s, categories C^s and brands B^s. Edges exist between U^s and I^s, between I^s and C^s, between I^s and B^s. Given the meta-path set $\{p_1, p_2, \cdots p_K\}$, after node aggregation, for node u, we can obtain K interest embeddings, denoted as $\{\mathbf{h}_u^{p_1}, \mathbf{h}_u^{p_2}, \cdots \mathbf{h}_u^{p_K}\}$, where $\mathbf{h}_u^{p_j} \in R^{1\times k}$ is the j-th interest embedding obtained by path p_j. We also get the item embedding $\mathbf{i}^s \in R^{1\times k}$ after updating the HIN node embeddings. Our approach in general can be applied to more complex HIN or other multi-interest extraction methods.

Multiple Personalized Bridges Given K interest embeddings, we transfer them using bridge functions $\{F_1, F_2, \cdots F_K\}$ that are personalized by the user's source features, where F_j means the j-th preference bridge. Intuitively, the user's historical behavior items S_u have different contributions to a specific bridge function F_j for transferring one source interest embedding $\mathbf{h}_u^{p_j}$. Therefore, we use target-aware attention (Sun et al. 2022) to make the interest embedding choose which item is used for each bridge. Specifically, given the sequence embedding, $V_u \in R^{n\times k}$, which represents all pre-trained $\mathbf{i}^s$ in S_u. we first calculate the correlations between each item in the sequence and one interest embedding $\mathbf{h}_u^{p_j}$. Weight vector $A^j \in R^{n\times 1}$ can be computed by:

$$A^j = \mathrm{softmax}(V_u(\mathbf{h}_u^{p_j})^\top) \tag{1}$$

Then we can obtain a weighted sum of item embedding $\mathbf{p}_u^j \in R^{1\times k}$ as follows:

$$\mathbf{p}_u^j = (A^j)^\top V_u \tag{2}$$

To personalize the bridge F_j, we feed $\mathbf{p}_u^j$ into a shared meta network (Zhu et al. 2022) $f(\cdot)$ to generate the parameters W_u^j for the j-th bridge function.

$$W_u^j = f(\mathbf{p}_u^j; \phi) \tag{3}$$

Where $f(\cdot)$ is a two-layer feed-forward network, which is parameterized by ϕ. Note that, The W_u^j is a vector whose size depends on the structure of the bridge function. The bridge function can be defined as any structure. In this paper, for simplicity, we use a linear layer following bridge-based methods (Man et al. 2017). Thus, to fit the size of the bridge's parameters, we reshape the vector into a matrix $W_u^j \in R^{k\times k}$. The output of bridge F_j is formulated as:

$$\mathbf{t}_u^{p_j} = F_j(\mathbf{h}_u^{p_j}; W_u^j) = \mathbf{h}_u^{p_j} W_u^j \tag{4}$$

where $\mathbf{t}_u^{p_j} \in R^{1\times k}$ denotes the j-th personalized transformed user's embedding. As multi-bridge functions depend on weights generated from meta network and vary from user to user, we call it multiple meta-generated personalized bridges.

With the multiple transformed interest embeddings, we can use Reinforcement Learning (RL) based interest selector to get a weighted sum of interests $\widehat{\mathbf{u}}^t \in R^{1\times k}$ as follows:

$$\widehat{\mathbf{u}}^t = \sum_{j=1}^{K} P_j \mathbf{t}_u^{p_j} \tag{5}$$

where P_j is the output of RL-based interest selector, which we will illustrate later. Finally, we can utilize the transformed embedding $\widehat{\mathbf{u}}^t$ for prediction.

Task-oriented Optimization For better training performance, we adopt the recently popular task-oriented optimization(Zhu et al. 2021, 2022), which directly utilizes the performance of the ultimate recommendation task as the optimization goal. In this paper, we focus on rating tasks, so the task-oriented loss can be formulated as:

$$Loss_{MIT} = \frac{1}{|R_o^t|} \sum_{r_{ui} \in R_o^t} (r_{ui} - \widehat{\mathbf{u}}^t (\mathbf{i}^t)^\top)^2 \tag{6}$$

where $R_o^t = \{r_{ui} | u \in U^o, i \in I^t\}$ denotes the interactions of overlapping users in the target domain.

RL Based Interest Selector (IS)

Inspired by the insights discussed in Introduction, we propose to dynamically assign weights to transformed interests at the instance level using a reinforced approach (Sutton 1992).

Figure 2 (b) illustrates the overview of our reinforcement learning based interest selection method. In each iteration, an agent interacts with the environment and receives the representation of a training instance, which is used for the agent's interest selection policy to calculate the sampling probabilities of multiple interests. The outputs of the policy will be used as the weight in equation 5 to participate in the muli-interest transfer (MIT) model. After an episode of training examples, the performance of the trained MIT model is used as the reward to update the policy parameters. This process iterates on episodes until the performance of the MIT model converges.

In general, a reinforcement learning approach involves elements in the form of (state, action, reward). The elements in our method are as follows.

State Our environment maintains a series of states $s \in S$ that summarize the characteristics of the input instances in the target domain and the transformed interests in the source domain. We design a state s as a vector $E_s \in R^{1\times((K+2)*k+K)}$, which is concatenated by two features.

The first feature is a vector representation $C \in R^{1\times((K+2)*k)}$ of each transformed interest $\mathbf{t}_u^{p_j}$, pre-trained user embedding $\mathbf{u}^t$ and item embedding $\mathbf{i}^t$ in target domain. To achieve the pre-trained embeddings in the target domain, there are many optional methods. In this paper, we explore three methods, MF (Koren, Bell, and Volinsky 2009), GMF (He et al. 2017) and YouTube DNN (Covington, Adams, and Sargin 2016)to get the pre-trained user and item embeddings in the target domain.

$$C = [\mathbf{t}_u^{p_1} \| \mathbf{t}_u^{p_2} \| \cdots \| \mathbf{t}_u^{p_K} \| \mathbf{u}^t \| \mathbf{i}^t] \tag{7}$$

where $\|$ represents a concatenate operation.

The second feature is the prediction (logit) vector $L \in R^{1\times K}$ that is concatenated by L_j of each transformed interest $\mathbf{t}_u^{p_j}$ and item $\mathbf{i}^t$ in target domain. In practice, the prediction is often derived from a dot-product between $\mathbf{t}_u^{p_j}$ and $\mathbf{i}^t$, that is,

$$\begin{aligned} L &= [L_1 \| L_2 \| \cdots \| L_K] \\ L_j &= \mathbf{t}_u^{p_j} (\mathbf{i}^t)^\top \end{aligned} \tag{8}$$

Action We use only one agent to sample all transformed interests. For each interest $\mathbf{t}_u^{p_j}$, the agent chooses between two possible actions, selecting the interest embedding or not for the current instance in the target domain. A policy function $\pi_\theta(s, a)$ parameterized by θ determines the distribution over the states, from which the action value of $a \in \{0, 1\}$ is sampled. Although the action value is sampled from $\{0, 1\}$, we can utilize the sampling probabilities $P = \{P_1, P_2, \cdots, P_K\}$ generated by the policy model $g(\cdot)$ as the weights of all transformed interests to avoid information loss.

$$\pi_\theta(s, a) = g(E_s; \theta) \tag{9}$$

where $g(\cdot)$ is a two-layer feed-forward network with relu activation function, which is parameterized by θ.

Reward The reward function is correlated with the performance of the MIT model trained with the selected transformed interest embeddings. We define an episode as one batch of training instances, that is, $D_b = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$, where b represents the batch ID, m is the batch size, x is the feature and y is the label. For each instance $(x, y) \in D_b$, we construct the state vector s_x and sample the action a_x^j for j-th interest according to policy $\pi_\theta(s_x, a_x^j)$ (Equation 9). We reshape all actions sampled as the weight in Equation 5 to train the muli-interest transfer (MIT) model.

Our reward calculation method is the minus of ground truth loss (Equation 6) of the MIT model, that is,

$$reward = -Loss_{MIT} \tag{10}$$

Please note that the reward is not given immediately after each step is taken. Instead, it is delayed until the training of the whole batch is completed.

Algorithm 1: Overall Training Procedure of REMIT

Input: $U^s, U^t, I^s, I^t, R^s, R^t, U^o$
Stage I: Pre-training
Pre-train a source model $\{\mathbf{h}_u^{p_1}, \mathbf{h}_u^{p_2}, \cdots \mathbf{h}_u^{p_K}\}, \mathbf{i}^s$ and a target model $\mathbf{u}^t, \mathbf{i}^t$
Stage II: Reinforced Multi-Interet Transfer
Pre-train the MIT model ϕ by minimizing task-oriented loss. ▷ Equation 6
Pre-train the IS policy θ by calculating the return under ϕ with all transformed interests selected.
Iteratively train MIT and IS in turn. ▷ Algorithm 2
Stage III: Initialization
For a cold-start u in target domain, we use IS policy's action to aggregate $\widehat{\mathbf{u}}^t$ as the user's initialized vector.

Optimization We follow the standard policy gradient method REINFORCE (Sutton 1992) to optimize the parameters, that is,

$$\theta \leftarrow \theta + \beta \sum_{x \in D_b} r \sum_{j=1}^{K} \nabla_\theta \pi_\theta(s_x, a_x^j) \tag{11}$$

where r is the reward defined by Equation 10 and β is the learning rate.

Algorithm 2: Joint Training of IS and MIT

Input: Epoch Number E; Training data $D = \{D_1, D_2, \cdots\}$; MIT and IS model initialized as ϕ, θ
for $epoch\ e = 1\ to\ E$ **do**
  Shuffle D to obtain a new training sequence.
  for $each\ batch\ D_b \in D$ **do**
    IS samples actions a_x for each instance $(x, y) \in D_b$ with θ to get the weights of multiple interests.
    Store (x, y, a_x) to the memory replay buffer M.
    Compute $\widehat{\mathbf{u}}^t$. ▷ Equation 5
    Update ϕ of MIT. ▷ Equation 6
  end for
  for each $(x, y, a_x) \in M$ **do**
    Compute delayed reward. ▷ Equation 10
    Update parameter θ of IS. ▷ Equation 11
  end for
end for

Model Training

Algorithm 1 shows the overall REMIT approach. Firstly, we initialize the MIT model ϕ by using the same fixed weights for each transformed interest embedding $\mathbf{t}_u^{p_j}$, that is, the agent's action a_x^j denotes the same sampling probability P_j for all interests. We then initialize parameter θ for the policy function using the same setting under ϕ. After initialization, we iteratively perform alternating training of multi-interest transfer and interest selection.

As described in Algorithm 2, in the MIT process, we fix the interest selection policy θ and learn the multi-interest transfer model ϕ. In the IS process, we fix ϕ to calculate the return and optimize the interest selection policy θ. The iteration continues for E epochs.

Tasks	Domain	Users		Items			Ratings
		#users	#overlap	#items	#categories	#brands	#ratings
Task 1	Moive	123,960	18,031	50,052	12164	472	1,697,533
	Music	75,258		64,443	-	-	1,097,592
Task 2	Book	603,668	37,388	367,982	123354	815	8,898,041
	Moive	123,960		50,052	-	-	1,697,533
Task 3	Book	603,668	16,738	367,982	123354	815	8,898,041
	Music	75,258		64,443	-	-	1,097,592

Table 1: Statistics of datasets in three tasks.

Experiments

In this section, we conduct extensive experiments to answer four research questions: **RQ1** How does REMIT perform in cold-start scenarios compared to state-of-the-art approaches from a bridge-based CDR perspective? **RQ2** Can REMIT generalize to different base models in the target domain? **RQ3** What's the effect of MIT and IS in REMIT? **RQ4** Why could REMIT perform better?

Experimental Setup

Datasets. Following most existing methods (Zhu et al. 2021, 2022), we test our algorithms on the Amazon review dataset[1]. Specifically, we pick 3 datasets out of 24, i.e., movies_and_tv (Movie), cds_and_vinyl (Music), and books (Book). Three cross-domain recommendation tasks are built upon these three datasets: Movie→Music (Task 1), Book → Movie (Task 2) and Book→Movie (Task 3). Statistics of these datasets are listed in Table 1.

Metrics. Amazon review dataset contains rating data (0 - 5 score). Following (Man et al. 2017; Zhu et al. 2022), we adopt Mean Absolute Error(MAE) and Rooted Mean Square Error(RMSE) to evaluate the performance.

Baselines. Since REMIT falls into the bridge-based methods for CDR, we mainly compare REMIT with the bridge-based methods. Therefore, we choose the following methods as baselines. 1) **TGT**, a naive target model trained only with data in the target domain. 2) **CMF** (Singh and Gordon 2008) shares user embeddings across the source domain and the target domain. 3) **EMCDR** (Man et al. 2017) employs MF to learn the embeddings and then utilizes a network to transfer user embeddings from the source domain to the target domain. 4) **SSCDR** (Kang et al. 2019) trains the bridge function in a semi-supervised manner. 5) **PTUPCDR** (Zhu et al. 2022) turns to the meta net fed with users' characteristic embeddings to produce personalized bridge functions for each user.

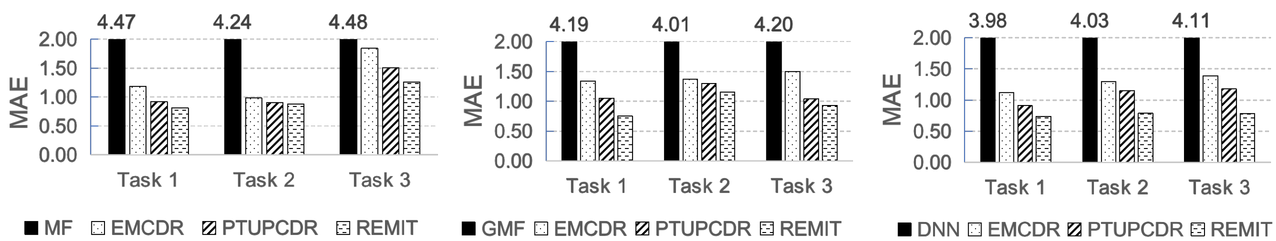
Implementation Details. The REMIT [2] is built based on the code repository of PTUPCDR using PyTorch and GPU. For both MIT and IS in each task and method, the initial learning rate for the Adam (Kingma and Ba 2014) optimizer is tuned by grid searches within {0.001, 0.005, 0.01,

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

[2]Code is available at https://github.com/mufusu21/REMIT

Scenarios	β	*Metric*	*TGT*	*CMF*	*SSCDR*	*EMCDR*	*PTUPCDR*	*REMIT*	*Improve*
Task 1	20%	*MAE*	4.4730	1.4128	1.2151	1.1963	1.0051	**0.9393***	6.55%
		RMSE	5.1615	1.8725	1.4946	1.4803	1.3498	**1.2709***	5.85%
	50%	*MAE*	4.5064	1.5179	1.3235	1.4570	1.1464	**1.0437***	8.96%
		RMSE	5.1774	2.0225	1.6794	1.8086	1.5987	**1.4580***	8.80%
	80%	*MAE*	4.5129	1.8609	1.6382	1.9397	1.4245	**1.2181***	14.49%
		RMSE	5.1983	2.4507	2.1576	2.3290	2.0338	**1.6601***	18.37%
Task 2	20%	*MAE*	4.2026	1.4857	1.2631	0.9834	0.9093	**0.8759***	3.67%
		RMSE	4.7677	1.9308	1.4700	1.2295	1.1914	**1.1650**	2.22%
	50%	*MAE*	4.2225	1.5974	1.3407	1.0556	0.9567	**0.9172***	4.13%
		RMSE	4.7890	2.0636	1.5607	1.3238	1.2712	**1.2379**	2.62%
	80%	*MAE*	4.2405	2.3416	1.2495	1.2249	1.0519	**1.0055***	4.41%
		RMSE	4.8201	2.8561	1.5572	1.5334	1.4338	**1.3772***	3.95%
Task 3	20%	*MAE*	4.4516	1.7873	1.5988	1.6121	1.4701	**1.3749***	6.48%
		RMSE	5.1455	2.3316	2.1146	2.1638	2.0707	**1.9940***	3.70%
	50%	*MAE*	4.4825	1.9348	1.8166	1.9050	1.5872	**1.4401***	9.27%
		RMSE	5.1585	2.5232	2.2718	2.3289	2.2279	**2.0495***	8.01%
	80%	*MAE*	4.5188	2.3989	2.1749	2.2192	1.8344	**1.6396***	10.62%
		RMSE	5.2001	3.0838	2.5652	2.6331	2.5235	**2.2653***	10.23%

Table 2: Mean results (MAE and RMSE) over five runs on 3 cross-domain cold-start tasks. Best results are labeled in bold and * indicates 0.05 level, paired t-test of REMIT vs. PTUPCDR. $Improve$ denotes relative improvement over PTUPCDR.



(a) MF based (b) GMF based (c) YouTube DNN based

Figure 3: Generalization experiments for EMCDR, PTUPCDR and REMIT upon three base models (a) MF, (b) GMF, (c) YouTube DNN. Mean results of MAE over five random seeds are reported.

0.02, 0.1$\}$. The epoch number is tuned by grid searchs between 5 and 15. In addition, we set the dimension of embeddings as 10 and the batch size as 512. The meta-path set is $\{uiu, uiciu, uibiu\}$, which represents users' interests of item, category and brand. The hidden size of meta network in MIT and policy network in IS is set to 50 and 25. We employ the same fully connected layer to facilitate comparison for the cross-domain bridge functions of EMCDR, SSCDR, PTUPCDR and REMIT. Following (Zhu et al. 2022), to evaluate the effectiveness of REMIT, 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 each task, we report the mean results over five random runs.

Cold-Start Experiments (RQ1)

The most concerning experimental result is how our framework performs in cold-start scenarios. In Table 2, we compare REMIT with five baseline models mentioned above to validate the effectiveness. We demonstrate the results on 3 CDR tasks under different settings of β, where β represents the percentage of cold-start users. Obviously, both MAE and RMSE increase as β grows since larger β leads to smaller training sets. Furthermore, we find that TGT performs worst because it relies only on interactions in the target domain. On the contrary, CMF takes advantage of information from the auxiliary source domain and thus achieves better results. However, CMF cannot differentiate information between different domains and thus ignores the domain shift. Having made up for this problem, the bridge-based method PTUPCDR stands out from existing methods by taking user personalities into consideration. But user preferences are usually too complicated to be characterized by a single representation. Thus instead of focusing on only one preference embedding in source domain, our framework, REMIT, transfers multiple source interest embeddings and employs an RL-based interest selector, achieving the best results.

Generalization Experiments (RQ2)

By default, we employ matrix factorization(MF) to learn embeddings for users and items in the target domain. However, MF might be too simple to fit the data in large-scale real-world recommendations. Besides, it is often the case that embeddings are trained by many quite different models, challenging the robustness and compatibility of bridge functions. Thus, to testify the compatibility of REMIT as well as other bridge-based methods (EMCDR, PTUPCDR), we switch the base model in the target domain to GMF (He et al. 2017) and YouTube DNN (Covington, Adams, and Sargin 2016) in turn. GMF is an advanced version of MF, where assigns various weights for different dimensions in the dot-product prediction function, we only use user embeddings during transfer. For YouTube DNN, the bridge function only transforms the output of the user tower. With $\beta = 20\%$, we conduct experiments over three tasks, and the results are demonstrated in Figure 3. On one hand, we can obviously see that the base models always yield the worst performances without absorbing auxiliary information. On the other hand, no matter how we switch the base model, these bridge-based methods continue to improve the performance, indicating their robustness and good generalization ability. Notably, changing the base model will not prevent REMIT from producing the best results, and it justifies the generalization ability of our framework.

Ablation Study (RQ3)

In this section, we aim to find out the key factors that affect performance most. To this end, we select PTUPCDR as the baseline and incrementally propose two variants, i.e., PTUP$^+$ and MIT. PTUP$^+$ augments the original PTUPCDR model by introducing multi-interest user embeddings, but all of them share a common personalized bridge. MIT further customizes interest-specific bridge functions for each user. Both PTUP$^+$ and MIT use the same weight for each transformed interest. Our framework, REMIT, distinguishes itself by weighing the importance of different interests. We test these algorithms on task 1 over different settings of β and report the results in Table 3. Firstly, it can be seen that simply incorporating multi-interest user embeddings negatively affects the performance, resulting from the fact that different interests share a common bridge, in which case a great challenge is posed to the bridge function since it is hard to capture users' diverse preferences. In MIT, we explicitly design different bridge functions for different interests in the source domain, and treat them equally, we can see that MIT successfully achieves better results. However, delivering the same weight to each interest will prevent the model from recognizing the interest relevance as we mentioned in the Introduction. Hence, by employing an RL-based interest selector upon MIT, we can notice that REMIT makes a significant improvement far beyond other algorithms.

Algorithms	$\beta = 20\%$		$\beta = 50\%$		$\beta = 80\%$	
	MAE	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
PTUPCDR	1.0051	1.3498	1.1464	1.5987	1.4245	2.0338
PTUP$^+$	1.0136	1.3629	1.1541	1.6116	1.4266	2.0413
MIT	0.9891	1.3313	1.1031	1.5744	1.2619	1.7920
REMIT	**0.9393**	**1.2709**	**1.0437**	**1.4580**	**1.2181**	**1.6601**

Table 3: Model performance of ablation study. The best results are underlined and highlighted in bold.

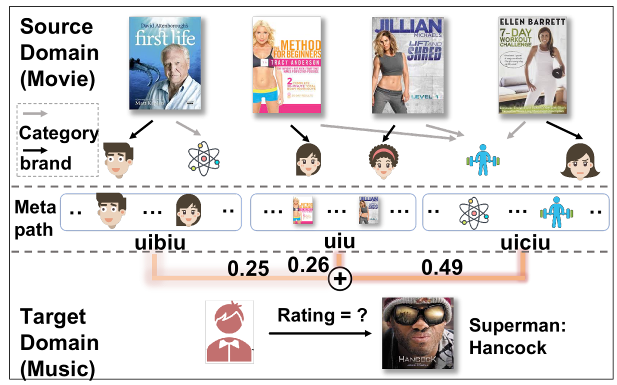


Figure 4: Case study of our REMIT.

Case Study (RQ4)

To explain the weight distribution over different interests, we pick out one instance specially and analyze the behavior of this cold-start user in Figure 4. In this instance, REMIT attempts to predict the rating of the user for the album "Superman: Hancock" in the target music domain. At the same time, we can see that in the source movie domain, the same user viewed a lot of movies and TV shows about sports and life science. As all items in both source and target domain are independent of each other and starred by different actors or singers, therefore low weights are assigned to embeddings obtained through meta-path uiu and $uibiu$. However, note that most items in the source domain are related to sports, strongly implying that the user prefers sports. The "Superman: Hancock" album in the target domain collects music played in the movie "Hancock", which is exciting and can cheer up people much during sports time, thus the embedding obtained by $uiciu$ can get higher weight, which helps make better predictions for target domain task.

Conclusions

In this paper, we tackle the problem of multi-interest transfer and selection in cross-domain recommendations when multiple source interests are available. After transforming different interest embeddings with different meta-generated personalized bridge functions for each user, we propose a novel RL-based interest selection method, which dynamically assigns weights to transformed interests at instance level to better coordinate the transformed user interest embeddings with the item embedding in the target domain. The extensive experiments confirm the effectiveness of our framework.

Acknowledgments

This work was supported by Ant Group through Ant Research Intern Program.

References

Bi, Y.; Song, L.; Yao, M.; Wu, Z.; Wang, J.; and Xiao, J. 2020. A heterogeneous information network based cross domain insurance recommendation system for cold start users. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2211–2220.

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.

Fu, W.; Peng, Z.; Wang, S.; Xu, Y.; and Li, J. 2019. Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 94–101.

Gao, C.; Lin, T.-H.; Li, N.; Jin, D.; and Li, Y. 2021. Cross-platform Item Recommendation for Online Social E-Commerce. *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

Hao, X.; Liu, Y.; Xie, R.; Ge, K.; Tang, L.; Zhang, X.; and Lin, L. 2021. Adversarial Feature Translation for Multi-Domain Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21, 2964–2973. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383325.

He, M.; Zhang, J.; Yang, P.; and Yao, K. 2018. Robust Transfer Learning for Cross-Domain Collaborative Filtering Using Multiple Rating Patterns Approximation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, 225–233. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355810.

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*, WWW '17, 173–182. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450349130.

Hu, G.; Zhang, Y.; and Yang, Q. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, 667–676. New York, NY, USA: Association for Computing Machinery. ISBN 9781450360142.

Kang, S.; Hwang, J.; Lee, D.; and Yu, H. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1563–1572.

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

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Li, P.; and Tuzhilin, A. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, 331–339. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368223.

Man, T.; Shen, H.; Jin, X.; and Cheng, X. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *IJCAI*, volume 17, 2464–2470.

Pan, W.; Xiang, E. W.; Liu, N. N.; and Yang, Q. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, 230–235. AAAI Press.

Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; and Philip, S. Y. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1): 17–37.

Singh, A. P.; and Gordon, G. J. 2008. Relational Learning via Collective Matrix Factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, 650–658. New York, NY, USA: Association for Computing Machinery. ISBN 9781605581934.

Sun, C.; Lu, P.; Cheng, L.; Cao, Z.; Dong, X.; Tang, Y.; Zhou, J.; and Mo, L. 2022. Multi-interest Sequence Modeling for Recommendation with Causal Embedding. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, 406–414. SIAM.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11): 992–1003.

Sutton, R. S. 1992. Reinforcement learning architectures. *Proceedings ISKIT*, 92.

Xi, D.; Chen, Z.; Yan, P.; Zhang, Y.; Zhu, Y.; Zhuang, F.; and Chen, Y. 2021. Modeling the Sequential Dependence among Audience Multi-step Conversions with Multi-task Learning in Targeted Display Advertising. *CoRR*, abs/2105.08489.

Zhao, C.; Li, C.; Xiao, R.; Deng, H.; and Sun, A. 2020. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 229–238.

Zhu, F.; Wang, Y.; Chen, C.; Liu, G.; Orgun, M.; and Wu, J. 2020. A deep framework for cross-domain and cross-system recommendations. *arXiv preprint arXiv:2009.06215*.

Zhu, Y.; Ge, K.; Zhuang, F.; Xie, R.; Xi, D.; Zhang, X.; Lin, L.; and He, Q. 2021. Transfer-meta framework for cross-domain recommendation to cold-start users. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1813–1817.

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 Fifteenth ACM International Conference on Web Search and Data Mining*, 1507–1515.