Data Science and Engineering
https://doi.org/10.1007/s41019-024-00245-y

RESEARCH PAPER

A Meta-adversarial Framework for Cross-Domain Cold-Start Recommendation

Yufang Liu[1] · Shaoqing Wang[1] · Xueting Li[1] · Fuzhen Sun[1]

Received: 27 July 2023 / Revised: 28 November 2023 / Accepted: 1 February 2024

Abstract

The cold-start problem in recommender systems has been facing a great challenge. Cross-domain recommendation can improve the performance of cold-start user recommendations in the target domain by using the rich information of users in the source domain. In cross-domain cold-start recommendation, users in target domain lack sufficient historical behaviors. Existing meta-learning-based methods depend on the feature distribution of training data and limit the adaptability in new tasks. To address these issues, we propose a meta-adversarial framework for cross-domain cold-start recommendation (MAFCDR) . Specifically, we employ a multi-level feature attention mechanism for independently learning the weights of long-term and short-term features to construct preferences of users in source domain. To migrate user representations, we train a meta-adversarial network that utilizes feature embeddings in the source domain as input and enhances the robustness and stability of the model. Then, the personalized bridge function transfers the user preferences in the source domain to the target domain. We build three cross-domain tasks using Amazon dataset and conduct extensive experiments, which demonstrate the effectiveness of the proposed model in cold-start user recommendation.

Keywords Cross-domain recommendation · Cold-start recommendation · Multi-level attention · Meta-learning · Adversarial learning

1 Introduction

As the amount of information rapidly increases, recommendation systems have become an important tool for information filtering. They help users discover products and services that they may be interested in. Recommendation systems have achieved great success in modeling their preferences and intentions by taking advantage of users' recent and long-term behaviors. Existing methods usually use recurrent neural networks [1, 2] and attention mechanisms [3, 4] to model user preferences based on historical interaction sequences. However, there are two serious challenges in recommendation systems, namely data sparsity and cold-start problems [5]. Cold-start users have few or no interaction items, and cross-domain recommendation [6] has attracted widespread attention from academia and industry by leveraging the rich user behaviors in the source domain to help the target domain make recommendations, thereby alleviating the data sparsity and cold-start problems.

In recent years, a promising solution in cross-domain recommendation is to connect the source domain and the target domain by learning the bridge function which can transfer the appropriate knowledge across two domains. For example, the EMCDR [6] model trains a bridge function shared by all users to achieve knowledge transfer from the source domain to the target domain. However, it utilizes a shared bridge function, which not only fails to reflect users' personalized recommendations, but also reduces the accuracy of recommendations, and can be considered as a coarse-grained method. On the one hand, the relationship between user preferences in source domain and that in target domain is complex and changeable, and it is difficult for a single bridge function to accurately capture the relationships of

✉ Shaoqing Wang
wsq0533@163.com

Yufang Liu
Liuyf0820@163.com

Xueting Li
1403529162@qq.com

Fuzhen Sun
13005068@qq.com

[1] School of Computer Science and Technology, Shandong University of Technology, Zibo, China

Published online: 05 March 2024

all users. On the other hand, it is unstable to use only some active users and popular items to train the bridge function, and ignores a large part of important users and items, which makes the generalization ability weak. In order to alleviate these shortcomings, PTUPCDR [7] proposes to leverage meta-network to learn personalized bridge functions for each user, and achieves good results.

Despite the validity of existing approaches, these studies have some limitations: (1) Most cross-domain approaches only consider long-term interests as users' overall preferences while ignoring sequential features, and they are limited in modeling dynamic short-term interests. For a new user, his current interests through short-term behaviors should be modeled, and then long-term interests will be continually complemented and extended to the recommendations. For an old user, recommender can model both long-term interests and short-term behaviors so as to capture their latest interest changes. It is important and challenging to adaptively fuse these two aspects. (2) Bridge functions learned from meta-learning are usually unstable and cannot be adapted to new tasks. In addition, personalized bridge functions may be too fine-grained and lead to overfitting problems. Meta-learning requires to training the model on a large number of similar training tasks; however, the data of cold-start users is very sparse and the meta-learning-based methods rely heavily on the feature distribution of the training data, it ignores the ability to enhance adaptation to new tasks. Then, the bridge functions learned from meta-learning cannot accurately migrate user features across two domains.

Therefore, it is an urgent problem to provide more accurate recommendations for cold-start users in the target domain while ensuring good processing granularity and avoiding overfitting problems.

Considering the dynamic representation of users' sequential interactions over the past period of time and taking advantage of meta-learning, we propose a meta-adversarial framework for cross-domain cold-start recommendation (MAFCDR) to solve the cold-start user recommendation problem. Specifically, we used Gated Recurrent Unit (GRU) gating to extract long short-term preferences of users in source domain. Then, we constructed a multi-level feature attention mechanism to independently learn the weights of long-term and short-term features. By weighing these features, we can build the users' interest representations. To transfer users' representations, we train an adversarial meta-network with the user's feature embedding in the source domain as the input, and we obtain the model parameters which can be applied to different tasks through the adversarial game of generators and discriminators so as to enhance the robustness and stability of the model. The obtained personalized parameters are used as initialization parameters of the bridge function, which can capture the preference relationships between different domains. After training, we input the user embeddings from the source domain to the bridge function generated by the meta-network, and obtain the transformed user embeddings. The transformed user embeddings are used as the initial embeddings in the target domain. With these initial embeddings, our method is effective for cold-start users in the target domain.

The contribution of this paper can be summarized as follows:

- We propose a meta-adversarial framework for cross-domain cold-start user recommendation. Personalized bridge functions for each user are generated by our model.
- We design multi-level feature attention to extract the long short-term preferences of users in the source domain, and transfer it to the preference representations of users in the target domain.
- We conduct extensive experiments on three cross-domain tasks using the Amazon dataset to validate the effectiveness of the proposed model.

2 Related Work

There are three lines of work that are most related to our work in this paper: cross-domain recommendations, cold-start recommendations and meta-learning.

2.1 Cross-Domain Recommendations

Cross-domain recommendations (CDRs) provide effective solutions for data sparsity and cold-start challenges. The basic idea of CDRs is to utilize the richer training data in source domain to improve the recommendation accuracy in the sparse target domain. Most of the existing CDR methods are based on collaborative filtering, while others use transfer learning-based CDR methods. Transfer learning-based CDR methods solve recommendation tasks in the target domain by transferring auxiliary knowledge that is different from but related to the target domain and improve the recommendation performance of the target domain. EMCDR [6] learns the mapping function on overlapping users, which maps user preferences across domains. SSCDR [8] uses overlapping users as anchors to calculate the preference characteristics of cold-start users through semi-supervised learning and k-nearest neighbor clustering to achieve cross-domain recommendation. DDTCDR [9] developed a new potential orthogonal mapping to extract users' preferences in multiple domains while retaining the relationship between users in different potential spaces. Similar to the multi-task approach [10], these approaches focus on well-designed deep network structures. In this paper, we design a framework that can explicitly model knowledge transfer between different domains, rather than using a special deep network structure to implicitly transfer knowledge.

2.2 Generative Adversarial Network

Generative adversarial networks (GANs) [11] are becoming increasingly popular in cross-domain recommendation. Traditional GAN models usually require a lot of training data to learn the data distribution, especially in the case of high-dimensional space and complex data distribution, and the training process is often unstable. ACDN [12] dynamically generates adversarial samples during training to improve the generalization ability of cross-domain recommender systems. GAR [13] tricks the recommender by adversarially training the generator so that the generated cold-item embeddings have a similar distribution as the warm item embeddings. ELECRec [14] trains the generator as an auxiliary model with the discriminator for reasonable alternative items are sampled, and the trained discriminator is considered as the final model.

2.3 Meta-learning

In recent years, the meta-learning in recommendation systems [15] has attracted people's attention. Most of these works focus on scenarios with few training samples, because it is natural to turn these tasks into few-shot learning problems. The inspiration for meta-learning comes from the human learning process, which can quickly learn new tasks based on a small number of examples. In existing meta-learning works, metric-based methods [16] learn metrics or distance functions on tasks, while model-based methods [17] aim to design an architecture or training process for fast generalization across tasks. Finally, the optimization-based approach [18] directly adapts the optimization method to achieve fast adaptation. In contrast, we consider the concept of meta-learning based on parameter optimization for recommendation system, which can serve the personalized recommendation model by reflecting each user's item interactions. The optimization-based meta-learning algorithm considers the distribution of model N and tasks $p(T)$. It attempts to find the ideal parameters of model N, as shown in Fig. 1. The optimization-based meta-learning algorithm performs local and global updates. Starting from random initial parameters θ, the algorithm extracts several tasks from the distribution of tasks $T_i \sim p(T)$. For each task $i = 1, \dots T$, the algorithm updates the parameter θ to θ^* locally by gradient $\nabla \theta L_i(f_\theta)$, where T is the number of sampling tasks, and $L_i(f_\theta)$ represents the training loss of task i. After local updating, for all sampling tasks, the algorithm updates the parameter θ globally based on $L'_i(f_{\theta^*})$, i.e., the test loss of the parameter θ^* of task i, so that the globally updated parameters are suitable for various tasks.

The optimization-based meta-learning algorithm uses two sets for each task, namely support set and query set. The support set and query set are used to calculate the training loss and test loss on each task, respectively. In the local update process, the algorithm adjusts the parameters of model in support set (learning process). In the global update process, the algorithm trains the parameter (from learning to learning) by minimizing the loss of the adaptive parameter on the query set. When the learning-learning process reaches the termination condition of the previous task, the algorithm only accepts the support set of the new task. Using a support set, the model can adapt to new tasks. Note that the algorithm allows not to store the parameters of each task. On the contrary, these parameters are calculated by the support set.

We regard each task as estimating user preferences in the recommendation system. Inspired by this, we propose a MAML-based recommendation system that can quickly estimate the preferences of new users based on only a few user-item interactions. The MAML-based recommendation system considers that different users have different optimal parameters. Therefore, our MAML-based model provides

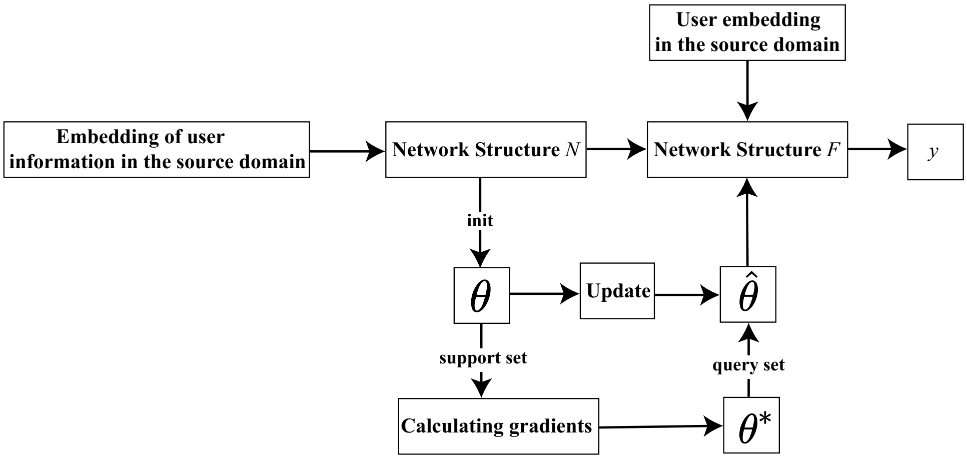


Fig. 1 Diagram of the optimization-based meta-learning algorithm

personalized recommendations for each user based on their unique consumption history.

3 Preliminaries

The CDR problem studied in this paper includes a source domain and a target domain. Each domain has a user set $U = \{u_1, u_2, \dots\}$, an item set $V = \{v_1, v_2, \dots\}$ and a rating matrix R. $r_{ij} \in R$ represents the interaction between user u_i and item v_j. In order to distinguish these two domains, the user set, item set and rating matrix of the source domain are represented as U^s, V^s, R^s, respectively, and the target domain is represented as U^t, V^t, R^t. The set of overlapping user between the two domains is defined as $U^o = U^s \cap U^t$. For items, V^s and V^t are disjoint, which means that there are no overlapping items between the two domains.

This paper leverages the embedding method to convert users and items into low-dimensional dense vectors. $u_i^d \in \mathbb{R}^k$ and $v_j^d \in \mathbb{R}^k$ represent the embedding of user u_i and item v_j, respectively, where k represents the embedding dimension and $d \in \{s, t\}$ represents the domain label. Given users' behavior sequences, we can generate dense vectors that encode the users' preferences and can be used (along with other rich features) to predict users' preference scores for items in the target domain.

4 The Proposed Model

4.1 Model Framework

Inspired by adversarial learning and meta-learning research, we propose a novel model framework as shown in Fig. 2. The framework attempts to combine the long-term (static) and short-term (dynamic) preferences of the user for the next item recommendation [19].

Given user u, we first obtain his/her long and short-term preferences representations, i.e., $\boldsymbol{p_l}$ and $\boldsymbol{p_e}$, according to his/her behavior sequence in the source domain. Then, a multi-level attention structure is adopted to fuse long-term and short-term features and we obtain a generalized user representation $\boldsymbol{p_u}$ where the contribution of long-term and short-term features is determined by dynamic learnable weight. In order to capture the complex relationship between different user preferences in source and target domains, we propose an adversarial training framework containing a generator that generates internal model initialization parameters and a discriminator that maintains meta-task invariance. Through adversarial training, a personalized bridge function is generated between users' embeddings in source and target domains. After training, we input the user embedding in the source domain into the bridge function and can obtain the transformed user embedding, which is used as the initial embedding in the target domain. Through the initial embedding, our method is effective for cold-start users who do not interact in the target domain.

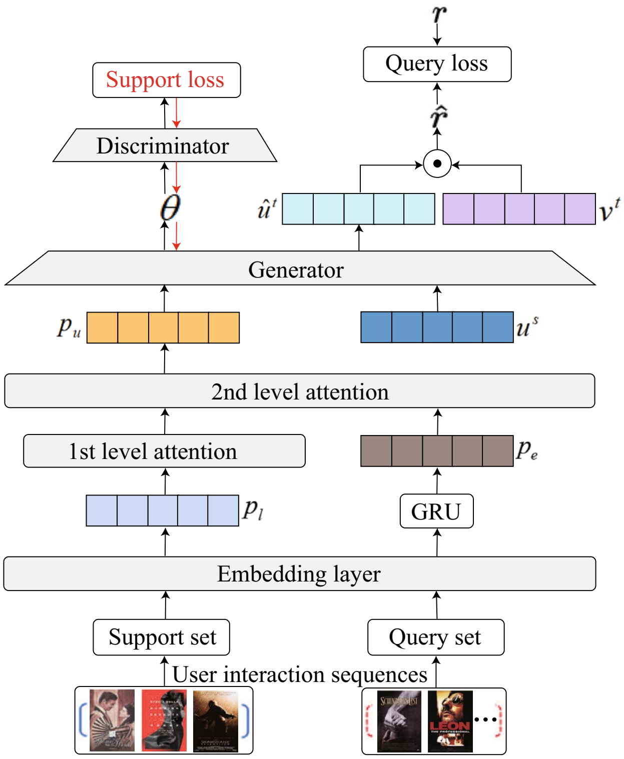


Fig. 2 Model framework

4.2 User Preference Learning

To capture the users' personalized preferences, we extract long-term and short-term feature vectors from the users' interactive sequences. In this way, our model can not only obtain stable long-term preferences of users, but also mine dynamic short-term preferences, which can improve the diversity and novelty of recommendations.

In this module, the users' long-term sequences are used as input, and the users' short-term preferences are captured by GRU gating. In order to integrate the long-term and short-term preferences of users, we design a multi-level attention structure that can determine the contribution of long-term and short-term features of users through dynamic learnable weighting factors.

4.2.1 Long Short-Term User Representation Learning

Due to the excellent performance of RNN in user sequential behavior modeling, it has attracted great attention in academia and industry in recent years. The update process can be defined as follows:

$$\boldsymbol{h_k} = g\left(W\boldsymbol{x_k} + U\boldsymbol{h_{k-1}} + b\right), \tag{1}$$

where g is the activation function, $\boldsymbol{x_k}$ is the latest user behavior, $\boldsymbol{h_{k-1}}$ is the last hidden state, b is the bias term, W and U are trainable parameters. Among all RNN-based models, LSTM (long short-term memory) and GRU are most commonly used in RS. Compared with LSTM, GRU has less 'gating' inside and fewer parameters than LSTM, but can achieve the same function as LSTM, and is easier to train in comparison, which can greatly improve the training efficiency. Therefore, we apply GRU to users' historical interactive sequences to extract short-term preferences. The equations are as follows:

$$\begin{aligned}
&\boldsymbol{r_i} = \sigma\left(W_r \cdot \left[\boldsymbol{h_{i-1}}, S_u\right]\right), z_i = \sigma\left(W_z \cdot \left[\boldsymbol{h_{i-1}}, S_u\right]\right),\\
&\widetilde{\boldsymbol{h}}_{\boldsymbol{i}} = \tanh\left(W_{\tilde{h}} \cdot \left[\boldsymbol{r_i} \otimes \boldsymbol{h_{i-1}}, S_u\right]\right),\\
&\boldsymbol{h_i} = \left(1 - z_i\right) \otimes \boldsymbol{h_{i-1}} + z_i \otimes \widetilde{\boldsymbol{h}}_{\boldsymbol{i}},
\end{aligned} \tag{2}$$

where $\boldsymbol{r_i} \in R^d$ and $z_i \in R^d$ are gates controlling past and present information, $W_r, W_z, W_{\tilde{h}} \in R^{d\times(d+1)}$ are learnable weights, $\sigma(\cdot)$ is a sigmoid function, $[\cdot,\cdot]$ denotes a connection, $\otimes$ denotes element-wise multiplication, and the initial hidden state h_0 is zero-initialized.

The short-term user preferences extracted by GRU are obtained by linear transformation using the output hidden layer state $\boldsymbol{h_i}$ as follows:

$$\boldsymbol{p_e} = W \cdot \boldsymbol{h_i} + b, \tag{3}$$

where $W \in R^d$ and $b \in R^d$ are learnable parameters.

As long-term preferences are inherent and static, we directly use the item sequence embedding of user interaction as the user's long-term preference, denoted as $\boldsymbol{p_l}$.

4.2.2 Long Short-Term Preference Fusion

The users' long-term preferences and short-term preferences reflect different aspects of information, and their dimensions are not exactly the same. We cannot simply use weighted summation to fuse them.

In this work, we use attention mechanisms [20] to address this problem. Attention-based models can not only capture the relationships between different components, but also selectively construct features to emphasize key information and weaken redundant information. In this paper, to describe user interests more carefully, we design a multi-level attention structure to determine the contribution of long-term and short-term features by dynamically learnable weighting factors.

We first augment the long-term feature representation using a first-level attention mechanism to capture key item information, then apply second-level attention to assign different weights, thus fusing the user representations are obtained. The first-order attention formula is as follows:

$$\boldsymbol{p_l'} = W_1\boldsymbol{p_l}, \tag{4}$$

where $\boldsymbol{p_l'}$ denotes the embedding obtained after the first-level attention mechanism and W_1 denotes the dynamically learned weight parameter.

Long-term user representations correspond to long-term preferences, while short-term user representations indicate dynamic and recent preferences. These two types of representations are complementary and their fusion may have a stronger expressive power. After obtaining the enhanced long-term user preferences, in order to further determine the proportion of cross-domain long-term preferences and short-term preferences, i.e., which of them occupies the majority in users' preferences, we use second-level attention to help make judgements.

$$\boldsymbol{p_u} = W_2\boldsymbol{p_l}' + W_3\boldsymbol{p_e}, \tag{5}$$

For more details, we calculate these weighting parameters using the following equations.

$$\begin{aligned}
&W_1 = \exp\left({h_1}^T\mathrm{ReLU}\left(V_1 \cdot \boldsymbol{p_l}\right) + \varphi_1\right),\\
&W_2 = \exp\left({h_2}^T\mathrm{ReLU}\left(V_2 \cdot [\boldsymbol{p_l}' \oplus \boldsymbol{p_e}]\right) + \varphi_2\right),\\
&W_3 = \exp\left({h_3}^T\mathrm{ReLU}\left(V_3 \cdot [\boldsymbol{p_l}' \oplus \boldsymbol{p_e}]\right) + \varphi_3\right),
\end{aligned} \tag{6}$$

where $V_1, V_2, V_3 \in R^{D_h\times D_{p_l}}$ are matrix parameters that implement the dimensional mapping, $h_1, h_2, h_3 \in R^{D_h}$ are vector parameters, and $\varphi_1, \varphi_2, \varphi_3$ are scalar parameters.

It is worth noting that the weight is calculated with $\exp(\cdot)$, which makes W_* may be greater than 1. This is a relatively benign consideration, because these weights can compensate for dimension differences to some extent. Of course, dynamic learning can be less than 1 if necessary.

4.3 Meta-adversarial Training Process

The relationship between user preferences in different domains varies from user to user, and thus the process of preference transfer needs to be personalized. Intuitively, there is some connection between preference relationships and user characteristics, and existing approaches use meta-networks to capture this relationship [7, 21, 22]. However, the distribution of tasks for meta-learning is often complex and variable, with very sparse data in each task, which makes the parameters learned by the meta-network unstable and cannot be adapted to new tasks, thus reducing the recommendation performance. To address this problem, we propose a meta-adversarial network in which the generator takes the users' transferable features as input and generates different model parameters for different tasks, while the discriminator discriminates whether the generated parameters have 'task invariance,' i.e., whether the parameters can maintain a certain stability across tasks. If the 'task invariance' is maintained, the discriminator will give positive feedback; otherwise, the discriminator will give negative feedback. We use a multilayer perceptron (MLP) to construct the encoder and discriminator.

Formally, for a given user feature $\boldsymbol{p}_{\boldsymbol{u}_i}$, we apply the following procedure to obtain the initial parameters of the personalized bridge function:

$$p_G\left(\theta \mid \boldsymbol{p}_{\boldsymbol{u}_i}\right)\text{MLP}_{\text{enc}}\left(\boldsymbol{p}_{\boldsymbol{u}_i};\phi\right), \tag{7}$$

where the generator is a two-layer feedforward network parameterized by ϕ

The loss of generators is:

$$\mathcal{L}_{gen} = -\prod_{i=1}^{N}\left(\theta = u_i^t\right)\log\left(p_G(\theta)\right). \tag{8}$$

In order to make the generated personalized parameters applicable to various tasks, we let the discriminator identify θ as false parameters to deceive the generator so that the generator learns shared features of multiple tasks during training, and is able to continuously improve the generator's ability, and continuously adapt to the specific needs of the current task by iterative adversarial methods. Then, the bridge function can more accurately transfer user preferences to the target domain in the face of new tasks.

The discriminator is defined as Eq. 9

$$p_D\left(\theta, u_i^t\right) = \text{MLP}_{\text{dis}}\left(\theta;\varphi\right), \tag{9}$$

where the discriminator is a two-layer feedforward network parameterized by φ, and θ is the parameter generated by the generator.

The purpose of the discriminator is to predict whether θ is a 'real' or 'generated' parameter. Since the generated parameters are eventually used to transfer the user representation, we directly use the embedded representation of the user in the target domain as the real sample to train the discriminator. The discriminator is trained with a binary cross-entropy loss as follows:

The discriminator loss is calculated as Eq. 10.

$$\begin{aligned}\mathcal{L}_{\text{dis}} = \sum_{i=1}^{N} &-\prod\left(\theta = u_i^t\right)\log\left(p_D\left(u_i^t\right)\right)\\ &-\prod\left(\theta \neq u_i^t\right)\log\left(1 - p_D(\theta)\right).\end{aligned} \tag{10}$$

The adversarial loss is shown in Eq. 11.

$$\mathcal{L}_{\text{adv}} = \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{dis}}. \tag{11}$$

With the adversarial approach, the meta-generator and the meta-discriminator can contribute to each other and improve the generative power of the generator. After obtaining the personalization parameters for the adaptation task, a bridge function is used to transfer the user preferences in the source domain to the target domain. The bridge function can be defined as any structure, and since the multilayer perceptron (MLP) can learn more complex features, improve training speed and accuracy, and also fine-tune on less data, it can perform better in new domain. We migrate user preferences directly using the trained generator as a bridge function and θ will be used as a parameter to the generator instead of an input. The generated bridge function varies from user to user and depends on the user's characteristics.

The users' embedding representations in the source domain are sent to the bridge function to obtain the transformed user embedding representation. The transformed embedding representation is considered as the initial embedding of the user in the target domain.

The transformed personalized user embedding can be obtained through the bridge function:

$$\hat{u}_i^t = \text{MLP}_{\text{enc}}\left(u_i^s;\theta\right), \tag{12}$$

where u_i^s denotes the embedding of user u_i in the source domain, $\hat{u}_i^t$ denotes the transformed user personalized embedding. Finally, $\hat{u}_i^t$ is used for prediction.

Existing bridge-based methods [21, 23] directly utilize the transformed user embedding $\hat{u}_i^t$ to minimize the loss. However, due to the limited number of items for some cold-start user interactions, the user embedding u_i^t may be unreasonable and inaccurate, and the learned unreasonable embedding can negatively affect the model. Therefore, we utilizes a task-oriented optimization approach to optimize the whole model.

To train the model, we use a task-oriented training procedure directly using the ratings of the final recommendation task as the optimization objective. The loss function can be formulated as follows:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|R_o^t|}\sum_{r_{ij}\in R_o^t}\left(r_{ij} - \hat{r}_{ij}\right)^2, \tag{13}$$

where $R_o^t = \left\{r_{ij} | u_i \in U^o, v_j \in V^t\right\}$ denotes the interaction of overlapping users in the target domain, r_{ij} the true rating of user i on item j, and $\hat{r}_{ij}$ is the prediction rating.

In the end, we combine the two loss functions by a linear interpolation to obtain the hybrid loss function:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{rec}} + (1 - \alpha)\mathcal{L}_{\text{adv}}, \tag{14}$$

where α is a hyper-parameter to control the relative importance of each loss function.

4.4 Training

The meta-network should be optimized for a large number of training tasks. We put this concept into the model to reflect personalized user preferences with only a small amount of interaction. The model of this paper considers the user's consumption history, constructs $M(M > 10)$ group training task. We randomly select 10 items in the sequence as the query set, and the rest is the support set. That is, in order to reflect the user's interest, the model updates the parameters in the meta-adversarial network according to the user's unique consumption history. In addition, unlike MAML [24], this paper extends the idea of matching networks [25] without limiting the length of the project consumption history, i.e., the length of the support set is not fixed.

Algorithm 1 Meta-adversarial networks (MAN) for parameter-generated meta-learning

```
**Input:** $f_\theta$ A pre-trained model; $G, D$: Two initialize networks;
**Output:** Model Parameters $\theta$;
Randomly initialize $G, D$;
**while** not converge **do**
    Sample a batch of users $B \sim p(B)$;
    **for** user $u$ in $B$ **do**
        Obtain interaction sequence embeddings $S_u$;
        Obtain user preferences $\boldsymbol{p_u}$ by Equ. 5;
        Evaluate $\mathcal{L}$ with $S_u$ and $\boldsymbol{p_u}$ by Equ. 14;
        Local update $G, D$ by minimizing $\mathcal{L}$;
        Evaluate $\mathcal{L}_{\text{rec}}$ with $S_u$ by Equ. 13;
        Global update $f_\theta$ by minimizing $\mathcal{L}_{\text{rec}}$;
    **end for**
**end while**
```

We denote the parameters of the generator and discriminator as ϕ and φ, respectively, and during each meta iteration, a meta-batch is first sampled from the meta-training dataset, then trained internally on that task, locally updating the parameters of the generator and discriminator by computing $\mathcal{L}$ and performing a gradient descent step on the support set as follows:

$$\begin{aligned} \phi &\longleftarrow \phi - \alpha \nabla_\phi \sum_{i=B} \frac{\partial \mathcal{L}^{\text{adv}}}{\partial \phi}, \\ \varphi &\longleftarrow \varphi - \beta \nabla_\varphi \sum_{i=B} \frac{\partial \mathcal{L}^{\text{adv}}}{\partial \varphi}, \end{aligned} \tag{15}$$

where $\alpha > 0$ and $\beta > 0$ are the step sizes (learning rate) of the gradient descent. This local update can be considered as a personalized iteration, which can be repeated several times. Now we have new generators and discriminators and then globally update the pre-trained model on the new interaction sequence based $\mathcal{L}_{\text{rec}}$, the purpose of this process is to find the ideal parameters to obtain good recommendation performance after several local updates for all users.

The meta-optimization is performed on the generator and discriminator parameters, i.e., ϕ and φ, while the goal is to use the updated generator to generate the personalized parameters θ used for migrating preference. In fact, the purpose of the meta-phase is to optimize the parameters of the task-oriented meta-adversarial network so that a set of one or a small number of gradient steps simulating a cold-start user will yield the most effective behavior on a real-world cold-start user. Finally, we obtain the overall training algorithm, i.e., Algorithm 1, for the model, which allows updating the meta-parameters by small batches of stochastic gradient descent.

5 Experiments

This section evaluates the proposed framework for solving cold-start user problems under different tasks. Firstly, the experimental setup and baselines are introduced. Then, extensive experiments are conducted on the Amazon dataset.

5.1 Experimental Setup

5.1.1 Datasets

The Amazon review dataset [26] is one of the most widely used public datasets for e-commerce recommendations, and this paper uses the Amazon-5 core dataset with at least five ratings per user or item. The dataset contains 24 different item domains. Three popular categories are chosen for this paper: movies_and_tv (movies), cds_and_vinyl (music) and books (books). Three CDR tasks are defined: task 1: movies→music, task 2: book→movies and task 3: books→music. As shown in Table 1, the number of ratings in the source domain is significantly larger than the number of ratings in the target domain. Unlike many existing works that select only a portion of the dataset for evaluation, this paper uses all the data directly to simulate real-world applications.

5.1.2 Evaluation Metrics

Firstly, to measure the regression predictive ability of the model, we select MAE and RMSE as evaluation metrics. Then, to verify the ranking ability of the model, we select AUC and NDCG@10 as evaluation metrics. They are widely used in recommender system to evaluate the performance of model.

Table 1 Cross-domain task information

CDR tasks	Domain		Item		User			Rating	
	Source	Target	Source	Target	Overlap	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

5.1.3 Baseline Models

The baselines can be divided into two groups: single domain and cross domain. In the first group, the source and target domains are considered as single domains respectively, and the popular matrix factorization (MF) method is utilized. The second group includes state-of-the-art CDR methods for cold-start users. As the proposed model belongs to the bridge-based CDR methods, this paper focuses on comparing the proposed model with the bridge-based methods. Therefore, the following methods are chosen as the baselines for comparison.

Single domain:

- TGT: TGT [27] is a MF model, trained using only target domain data.
- CMF: CMF [28] is an extension of MF. In CMF, the user's embedding vector can be shared across source and target domains. Cross-domain:
- SSCDR: SSCDR [8] is a method based on semi-supervised bridge.
- DCDCSR: DCDCSR [21] belongs to a bridge-based approach, which considers the sparsity of individual users' ratings in different domains.
- EMCDR: EMCDR [6] is a commonly used cold-start CDR method. MF is first used to learn the embedding, and then the network is used to connect the user embedding from source domain to target domain.
- RecGURU: RecGURU [29] learns users' long short-term preference through adversarial training, achieving information sharing and cross-domain collaboration in user representations.
- ELECRec: ELECRec [14] is a generative task. The generator is trained as an auxiliary model with the discriminator to sample reasonable alternatives, and the trained discriminator is considered as the final RS model.
- PTUPCDR: PTUPCDR [7] belongs to the bridge-based cold-start CDR approach, which generates a personalized bridge function by using a meta-network of user feature embeddings and enables personalized preference transfer for each user.

5.1.4 Implementation Details

The proposed framework is implemented by PyTorch. For each task and method, the initial learning rate of the Adam [30] optimizer is tuned by a grid search in the range {0.001, 0.005, 0.01, 0.02, 0.1}. In addition, the dimensionality of the embedding is set to 10. For all methods, the small batch size is set to 512. The same fully connected layer is used to facilitate comparison of EMCDR, DCDCSR, SSCDR, PTUPCDR and MAFCDR, where the mapping function for MAFCDR is generated by a meta-network. The meta-network is a two-layer linear model with hidden cells of $2 \times k$, where k denotes the embedding dimension, and the output dimension of the meta-network is $k \times k$.

To evaluate the performance of the proposed model, a portion of the overlapping users are then removed from the target domain and they are used as test users, while the rest of the overlapping user samples are used to train the meta-learner. In the experiments, the proportion of test (cold-start) users β is set to 20%, 50% and 80% of the total overlapping users. Overlapping users with item consumption history lengths between 13 and 100 are selected in the training data. For each overlapping user in the training data, 10 random items from the interactive sequences are used as the query set and the rest of the items are used as the support set, i.e., length of item consumption history is the value between 3 and 90, which shows good performance even though the length of the support set is not fixed.

5.2 Comparative Experiments

Tables 2 and 3 show the performance of MAFCDR on the three cross-domain recommendation tasks. For each task, we report the average results of five random runs. The best performance is shown in bold. $*$ indicates 0.05 level paired t test of MAFCDR against the best baseline. The Improve column indicates improvement relative to the best baseline. The following observations can be made from the experimental results.

- TGT is a single-domain model that uses only data from the target domain and its performance is not satisfactory. Compared to TGT, all other cross-domain methods can utilize data from the source domain, resulting in better results. Therefore, utilizing data from source domains

Table 2 Regression performance comparison of different models on 3 cross-domain tasks

	β (%)	Metric	TGT	CMF	DCDCSR	SSCDR	EMCDR	RecGURU	ELECRec	PTUPCDR	MAFCDR	Improve (%)
Task 1	20	MAE	4.4803	1.5209	1.4918	1.3017	1.2350	1.2320	1.3084	1.1504	**0.9998***	13.10
		RMSE	5.1580	2.0158	1.9210	1.6579	1.5515	1.5545	1.5539	1.5195	**1.3421***	11.68
	50	MAE	4.4989	1.6893	1.8144	1.3762	1.3277	1.3696	1.4662	1.2903	**1.1740***	9.02
		RMSE	5.1736	2.2271	2.3439	2.0105	1.8021	1.6640	1.8258	1.7546	**1.6266***	7.30
	80	MAE	4.5020	2.4186	2.7194	1.8247	1.6851	1.7154	1.7624	1.6006	**1.5013***	6.21
		RMSE	5.1891	3.0936	3.3065	2.5901	2.3154	2.2160	2.2208	2.2107	**2.1363***	3.37
Task 2	20	MAE	4.1831	1.3632	1.3971	1.2390	1.1162	1.0404	1.3350	1.0740	**0.9479***	11.75
		RMSE	4.7536	1.7918	1.7346	1.6526	1.4120	1.2598	1.5544	1.3829	**1.2314***	10.96
	50	MAE	4.2288	1.5813	1.6731	1.2137	1.1832	1.2434	1.7134	1.1014	**0.9770***	11.30
		RMSE	4.7920	2.0886	2.0551	1.5602	1.4981	1.4921	1.9152	1.4395	**1.2896***	10.42
	80	MAE	4.2123	2.1577	2.3618	1.3172	1.3156	1.2243	1.6470	1.1999	**1.0718***	10.68
		RMSE	4.8149	2.6777	2.7702	1.7024	1.6433	1.5801	1.9126	1.5916	**1.4585***	8.37
Task 3	20	MAE	4.4873	1.8284	1.8411	1.5414	1.3524	1.4467	1.5834	1.2681	**1.0736***	15.34
		RMSE	5.1672	2.3829	2.2955	1.9283	1.7853	1.7188	1.7989	1.6944	**1.4873***	12.23
	50	MAE	4.5073	2.1282	2.1736	1.4739	1.4723	1.7496	1.8369	1.4347	**1.1759***	18.04
		RMSE	5.1727	2.7275	2.6771	2.2823	2.0159	2.0766	2.2221	1.9303	**1.6569***	14.17
	80	MAE	4.5204	3.013	3.1405	2.0846	1.8850	1.8535	1.9910	1.7550	**1.5143***	13.72
		RMSE	5.2308	3.6948	3.5842	2.7068	2.5194	2.3401	2.4432	2.3883	**2.1599***	9.57

Table 3 Ranking performance comparison of different models on 3 cross-domain tasks

	β (%)	Metric	TGT	CMF	EMCDR	RecGURU	ELECRec	PTUPCDR	MAFCDR	Improve (%)
Task 1	20	AUC	0.7737	0.7769	0.8014	0.8112	0.8172	0.8114	**0.8316***	1.76
		NDCG@10	0.2049	0.2066	0.2255	0.2672	0.3400	0.3403	**0.3629***	6.64
	50	AUC	0.7440	0.7430	0.7853	0.7818	0.8018	0.8113	**0.8228***	1.41
		NDCG@10	0.1706	0.1872	0.2068	0.2233	0.2440	0.2559	**0.2672***	4.41
	80	AUC	0.7391	0.7336	0.7620	0.7999	0.8190	0.8199	**0.8286***	1.06
		NDCG@10	0.1872	0.1722	0.2305	0.2368	0.2326	0.2429	**0.2506***	3.17
Task 2	20	AUC	0.7437	0.7697	0.8175	0.8219	0.8154	0.8275	**0.8335***	0.72
		NDCG@10	0.1913	0.2002	0.2403	0.2652	0.2590	0.2706	**0.2801***	3.51
	50	AUC	0.6914	0.6939	0.7211	0.7201	0.7196	0.7303	**0.7492***	2.58
		NDCG@10	0.2193	0.2104	0.2414	0.2459	0.2652	0.2702	**0.2869***	6.18
	80	AUC	0.6235	0.6259	0.6793	0.6831	0.7100	0.7097	**0.7267***	2.35
		NDCG@10	0.2144	0.2089	0.2444	0.2565	0.2480	0.2611	**0.2696***	3.25
Task 3	20	AUC	0.7199	0.7295	0.7936	0.7800	0.8024	0.8135	**0.8351***	2.42
		NDCG@10	0.2150	0.2049	0.2270	0.2392	0.2462	0.2169	**0.2595***	5.40
	50	AUC	0.6396	0.6384	0.6788	0.7004	0.7255	0.7115	**0.7348***	1.70
		NDCG@10	0.1525	0.1596	0.1633	0.1867	0.2016	0.1725	**0.2092***	3.76%
	80	AUC	0.6369	0.6292	0.6734	0.6799	0.6784	0.6955	**0.7111***	2.24%
		NDCG@10	0.1474	0.1446	0.1575	0.1850	0.1972	0.1913	**0.1995***	1.16%

is an effective way to alleviate data sparsity and can improve the performance of target domain recommendations.

- CMF uses auxiliary data by combining data from different domains into a single domain, while the CDR approach is specifically designed. It can be observed that the CDR method can outperform CMF for most tasks, this is because CMF ignores potential domain shifts by treating the data from both domains as identical. In contrast, the bridge function can transform the source embedding into the target feature space, effectively alleviating the effect of domain shifts. It is therefore essential to investigate CDR by making more effective use of auxiliary domains.

- By observing the results of the *t* test with a 95% confidence level, it can be seen that MAFCDR significantly outperforms the best baseline in most cases, indicating that MAFCDR is an effective solution for cold-start recommendations.

5.3 Ablation Experiments

The ablation experiments further explore the impact of the various components of the proposed MAFCDR model on performance. Specifically, the following models will be evaluated.

- -Mulatt: It replaces the multi-level feature attention structure in the model with a self-attentive mechanism and uses long-term sequential user features as input to the self-attentive mechanism.
- -GAN: The GAN is removed from the model and is replaced with a two-layer linear network as a meta-network.
- -MAN: The meta-network is removed from the model, and we transfer the user preferences learned through the multi-level attention structure to the target domain through simple matrix multiplication.
- -TOO: We replace the task-oriented optimization loss with a mapping-oriented optimization process to minimize the distance, using the transformed user embedding $\hat{u}_i^t$ to approach the target embedding u_i^t.

Table 4 shows the results of the ablation experiments for the introduced variants on the three cross-domain recommendation tasks. Differences in overall recommendation performance can be observed when sub-modules or features are gradually subtracted from the complete model. It indicates the effectiveness of the individual modules for cold-start cross-domain recommendations.

5.4 Parameter Experiments

We explore the impact of the number of local updates on Task 1. Figure 3 shows the performance of our method on two metrics by varying the number of personalized iterations. Even with few local updates, the model achieves significant improvements on both metrics. After a single iteration of the data, the method achieves significantly lower MAE and RMSE values. After one iteration, a slightly different result is observed by increasing the number of local updates, contrary to the results of the existing MAML [31], whose performance improves with increase in number of iterations. Our model can be adapted quickly to the user, as just one local update is sufficient. Fast adaption allows the proposed method to be applied to online recommendations based on user ratings.

5.5 Generalization Experiments

The comparison experiments mainly applied to MF for experimental evaluation. However, MF is a non-neural model, and the matrix decomposition algorithm is one of the most effective methods in recommendation recommendations. Therefore, to demonstrate the compatibility of MAFCDR with other bridge-based methods, i.e., EMCDR, PTUPCDR and MAFCDR are applied to two more complex neural models: GMF [32] and YouTube DNN [33]. GMF assigns different weights to different dimensions in the dot product prediction function, which can be seen as a generalization of the ordinary MF. YouTube DNN is a two-tower model. For GMF, the parameters trained by meta-learning can directly transfer the user embedding to the target

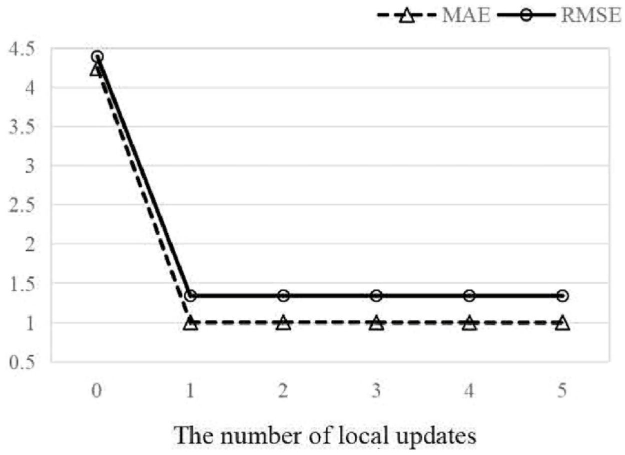
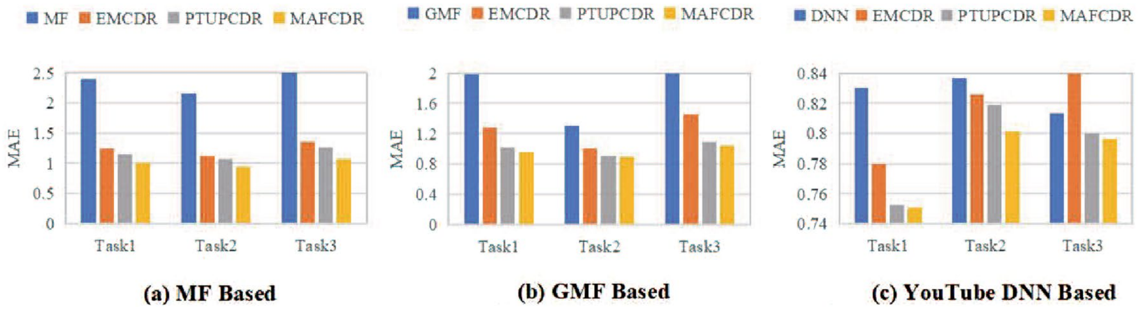


Fig. 3 The performance of our model according to vary the number of local updates on the task1

Table 4 Ablation experiments on three cross-domain tasks

	-Mulatt	-GAN	-MNet	-Too	Task 1	Task 2	Task 3
MAFCDR	–	–	–	–	0.9998	0.9479	1.0736
	√	–	–	–	1.0904	1.0078	1.2016
	√	√	–	–	1.1504	1.0740	1.2681
	√	√	√	-	2.8827	3.6706	2.7425
	√	√	√	√	3.0284	3.7532	2.9409



(a) MF Based

(b) GMF Based

(c) YouTube DNN Based

Fig. 4 Generalization experiment on three basic models **a** MF, **b** GMF, and **c** YouTube DNN. The average results of five runs are shown in the figure

domain. For YouTube DNN, the bridge function will transform the output of the user tower. Generalization experiments are conducted on the non-neural model (MF) and the neural model (GMF, YouTube DNN). From the results shown in Fig. 4, the following conclusions can be obtained:

- The bridge-based CDR approaches can be applied to a variety of baseline models. For different baseline models, EMCDR, PTUPCDR and MAFCDR are effective in improving the performance of recommendations for cold-start users in the target domain. As GMF and YouTube DNN are two popular and well-designed models in large-scale real-world recommendations, they achieve better performance than that of ordinary MF.
- The generalized MAFCDR can achieve satisfactory performance. On the one hand, with various base models, generalized MAFCDR can consistently achieve better results. On the other hand, the cold-start problem is highly challenging and the results of MAE are sufficient to demonstrate the effectiveness of the generalized MAFCDR in cold-start scenarios.

6 Conclusion

To better transfer user preferences from the source domain to the target domain, we proposed to train a meta-learning parameter for each user using a meta-adversarial framework. A meta-generator containing user feature embeddings was learnt to obtain personalized parameters that vary from user to user, and a bridge function was used to initialize the user embeddings to enable personalized transfer of user preferences. Extensive experiments were conducted on real datasets to evaluate the proposed model, and the results validate the effectiveness of the proposed model for cold-start cross-domain recommendation. In the future, we plan to integrate more content information into the framework to further alleviate the cold-start problem.

Acknowledgements This work was supported by Shandong Provincial Natural Science Foundation, China (ZR2020MF147, ZR2021MF017), Science and Technology Support Plan for Youth Innovation of Colleges and Universities of Shandong Province of China (2021KJ031).

Author contributions Liu Yufang was involved in methodology, writing—original draft, data analysis and paper writing. Wang Shaoqing helped in supervision, project administration, funding acquisition. Li Xueting contributed to formal analysis, investigation, resources. Sun Fuzhen assisted in writing—review & editing.

Funding Shandong Provincial Natural Science Foundation (ZR2020MF147, ZR2021MF017), Science and Technology Support Plan for Youth Innovation of Colleges and Universities of Shandong Province of China (2021KJ031).

Data availability The datasets used or analyzed in this study are openly available in http://jmcauley.ucsd.edu/data/amazon/links.html The implementation code is available online: https://github.com/LiuerH/MAFCDR.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Chen X, Xu H, Zhang Y, Tang J, Cao Y, Qin Z, Zha H (2018) Sequential recommendation with user memory networks. In: Proceedings of the Eleventh ACM international conference on web

search and data mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, pp 108–116

2. Hidasi B, Karatzoglou A (2018) Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International conference on information and knowledge management, CIKM 2018, Torino, Italy, October 22-26, 2018, pp 843–852
3. Ying H, Zhuang F, Zhang F, Liu Y, Xu G, Xie X, Xiong H, Wu J (2018) Sequential recommender system based on hierarchical attention networks. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, pp 3926–3932
4. Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P (2019) Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management, CIKM 2019, Beijing, China, November 3-7, 2019, pp 1441–1450
5. Li J, Jing M, Lu K, Zhu L, Yang Y, Huang Z (2019) From zero-shot learning to cold-start recommendation. In: AAAI, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pp 4189–4196
6. Man T, Shen H, Jin X, Cheng X (2017) Cross-domain recommendation: An embedding and mapping approach. In: IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pp 2464–2470
7. Zhu Y, Tang Z, Liu Y, Zhuang F, Xie R, Zhang X, Lin L, He Q (2022) Personalized transfer of user preferences for cross-domain recommendation. In: WSDM ’22 Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022, pp 1507–1515
8. Kang S, Hwang J, Lee D, Yu H (2019) Semi-supervised learning for cross-domain recommendation to cold-start users. In: CIKM 2019, Beijing, China, November 3-7, 2019, pp 1563–1572
9. Li P, Tuzhilin A (2020) DDTCDR: deep dual transfer cross domain recommendation. In: WSDM ’20 Houston, TX, USA, February 3-7, 2020, pp 331–339
10. Bai T, Xiao Y, Wu B, Yang G, Yu H, Nie J (2022) A contrastive sharing model for multi-task recommendation. In: WWW ’22 Virtual Event, Lyon, France, April 25 - 29, 2022, pp 3239–3247
11. Li Y, Xu JJ, Zhao P-P, Fang JH, Chen W, Zhao L (2020) Atlrec: An attentional adversarial transfer learning network for cross-domain recommendation. J Comput Sci Technol 35:794–808
12. Yan H, Zhao P, Zhuang F, Wang D, Liu Y, Sheng VS (2020) Cross-domain recommendation with adversarial examples. In: DASFAA 2020, Jeju, South Korea, September 24-27, 2020, Proceedings, Part III, pp 573–589
13. Chen H, Wang Z, Huang F, Huang X, Xu Y, Lin Y, He P, Li Z (2022) Generative adversarial framework for cold-start item recommendation. In: SIGIR ’22 Madrid, Spain, July 11 - 15, 2022, pp 2565–2571
14. Chen Y, Li J, Xiong C (2022) Elecrec: Training sequential recommenders as discriminators. In: SIGIR ’22 Madrid, Spain, July 11 - 15, 2022, pp 2550–2554
15. Hospedales TM, Antoniou A, Micaelli P, Storkey AJ (2022) Meta-learning in neural networks: a survey. IEEE Trans Pattern Anal Mach Intell 44(9):5149–5169
16. Lake BM (2019) Compositional generalization through meta sequence-to-sequence learning. In: NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp 9788–9798
17. Vuorio R, Sun S, Hu H, Lim JJ (2019) Multimodal model-agnostic meta-learning via task-aware modulation. In: NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp 1–12
18. Rajeswaran A, Finn C, Kakade SM, Levine S (2019) Meta-learning with implicit gradients. In: NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp 113–124
19. Xu M, Liu F, Xu W (2019) A survey on sequential recommendation. In: 2019 6th international conference on information science and control engineering (ICISCE), pp 106–111
20. Shaw P, Uszkoreit J, Vaswani A (2018) Self-attention with relative position representations. In: NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (ShortPapers), pp 464–468
21. Zhu F, Wang Y, Chen C, Liu G, Orgun MA, Wu J A deep framework for cross-domain and cross-system recommendations. CoRR (2020) arXiv:abs/2009.06215
22. Zhu, Y., Xie, R., Zhuang, F., Ge, K., Sun, Y., Zhang, X., Lin, L., Cao, J. (2021): Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In: SIGIR ’21 Virtual Event, Canada, July 11-15, 2021, pp 1167–1176
23. Fu, W., Peng, Z., Wang, S., Xu, Y., Li, J.: Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In: The Thirty-Third AAAI conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, The Ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pp. 94–101 (2019)
24. Finn, C., Abbeel, P., Levine, S. (2017): Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th international conference on machine learning - Vol 70, pp 1126–1135
25. Cai, Q., Pan, Y., Yao, T., Yan, C., Mei, T. (2018): Memory matching networks for one-shot image recognition. In: CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp 4080–4088
26. Ni, J., Li, J., McAuley, J.J. (2019): Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp 188–197
27. Salakhutdinov, R., Mnih, A. (2007): Probabilistic matrix factorization. In: advances in neural information processing systems 20, Proceedings of the twenty-first annual conference on neural information processing systems, Vancouver, British Columbia, Canada, December 3-6, 2007, pp 1257–1264
28. Singh, A.P., Gordon, G.J. (2008): Relational learning via collective matrix factorization. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, Las Vegas, Nevada, USA, August 24-27, 2008, pp 650–658
29. Li, C., Zhao, M., Zhang, H., Yu, C., Cheng, L., Shu, G., Kong, B., Niu, D. (2022): Recguru: Adversarial learning of generalized user representations for cross-domain recommendation. In: WSDM ’22: The Fifteenth ACM international conference on web search and data mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022, pp. 571–581
30. Kingma, D.P., Ba, J. (2015): Adam: A method for stochastic optimization. In: ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings
31. Finn, C., Abbeel, P., Levine, S. (2017): Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, pp 1126–1135
32. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. (2017): Neural collaborative filtering. In: WWW 2017, Perth, Australia, April 3-7, 2017, pp 173–182
33. Covington, P., Adams, J., Sargin, E. (2016): Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM conference on recommender systems, Boston, MA, USA, September 15-19, 2016, pp 191–198