Cross-Domain Meta-Learner for Cold-Start Recommendation

Renchu Guan, *Member, IEEE,* Haoyu Pang, Fausto Giunchiglia, Yanchun Liang and Xiaoyue Feng

Abstract—The cold-start problem is a major factor that limits the effectiveness of recommendation systems. Having too few available interaction records brings a series of challenges when predicting user preferences. At present, there are two main kinds of strategies for solving this problem from different perspectives. One is cross-domain recommendation (CDR), which introduces additional information by domain knowledge propagation with transfer learning. However, CDR methods follow traditional training processes in machine learning and cannot solve this typical few-shot problem from the perspective of optimization. The other type of method that has recently emerged is based on meta-learning. Most of these approaches focus only on generating a meta-model to perform better on new tasks and ignore improvements based on cross-domain information. Therefore, it is necessary to design a novel approach to solve this problem with both domain knowledge and meta-optimization. To achieve this goal, a novel cross-domain meta-learner for cold-start recommendation (MetaCDR) is proposed. In MetaCDR, we design a domain knowledge meta-transfer module to connect different domain networks. In addition, we introduce a pretraining strategy to ensure its efficiency. The experimental results show that MetaCDR performs significantly better than state-of-the-art models in a variety of scenarios.

Index Terms—Recommender systems, cold-start problem, transfer learning, meta-learning, cross-domain recommendation.

1 Introduction

FACED with an increasingly severe information overload problem, recommendation systems are playing essential roles in online services [8], [16], [19]. An excellent recommendation system can accurately and quickly discover users' personalized preferences, which provides convenience to users and brings substantial economic benefits to businesses [7], [53], [65]. Most recommendation systems learn a given user's preferences from the user's historical interaction information to generate recommendation results. However, in the real world, new users and items will constantly enter the system. These new users and items with little available data severely limit the performance of recommender systems; this is the well-known cold-start problem [15], [51].

An intuitive method for solving the cold-start problem is to introduce more available data to the system [17], [51], [70], such as by obtaining additional item features or the demographic information of the examined user during the data collection phase (e.g., extracting this information from knowledge graphs) [54], [57]. Instead of relying on the availability of additional information and incurring the cost of obtaining it manually, a more attractive approach is to improve the model structure [60] or build a mapping function [36] to introduce knowledge from other domains; this is called cross-domain recommendation (CDR) [68].

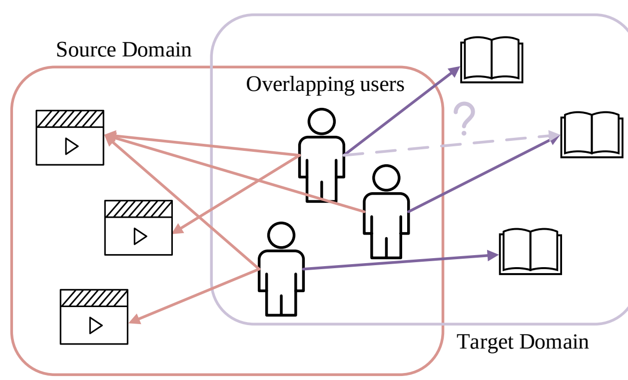

Fig. 1. An example of a cross-domain scenario including two domains: movies and books. We infer the users' personal preferences by integrating their interactions in the two domains.

Fig.1 shows an example of a cross-domain scenario. In the real world, the same users in multiple domains can be aligned, and the user interaction records in other domains are considered auxiliary information in the current domain. In this regard, deep fusion networks [37] with transfer learning [39] (e.g., cross-stitch networks (CSNs)) [22], [61] have achieved remarkable results. However, most of these works have focused on building more complex neural networks to achieve high-quality information transmission while ignoring the important role of model optimization strategies in solving this typical few-shot problem. The core problem of cold-start recommendation is that new users or items have only a small number of interactions in the recommender system with which to model their features. Similarly, the few-shot problem is that only a small number of samples per class are available [56]. Therefore, it is feasible to use

- R. C. Guan, H .Y. Pang, Y. C. Liang and X. Y. Feng (corresponding author) are with the Key Laboratory for Symbol Computation and Knowledge Engineering of the Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China.
- R.C. Guan and Y.C. Liang are with the Zhuhai Laboratory of the Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Zhuhai College of Jilin University, Zhuhai 519041, China.
- F. Giunchiglia is with the DISI, University of Trento, Trento, Italy.

Manuscript received xxx; revised xxx.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

few-shot learning methods to solve the cold-start problem.

Recent research on meta-learning [21] has provided new ways to solve this few-shot problem from an optimization perspective. Among them, gradient-based meta-learning (e.g., model-agnostic meta-learning) [12] learns the shared information among tasks to adapt to a new task with a few parameter update steps. This method has achieved great success in solving the cold-start problem for recommender systems. It treats each user as a single task and learns general characteristics among users. When a new user arrives, only a small amount of interactive information is needed to predict the user's preferences. However, most of the current meta-learning models focus on generating a meta-model to perform better on new tasks [9], [26], [59], and only simple MLPs are used as the basic model. Improving the extraction of cross-domain information is ignored, which leads to great restrictions on the usage scenarios of the resulting model.

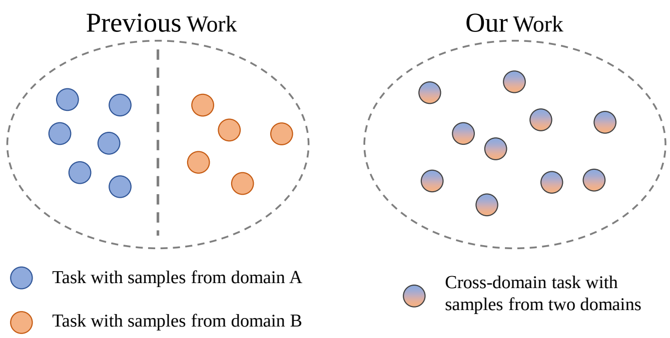

Fig. 2. The differences between the problems in our work and previous works.

Although some meta-learning algorithms have focused on solving the cross-domain problem [52], [58], they are different from those required to solve the problems we describe above. We show the difference in Fig.2. In previous works, although each task is obtained from a different domain, each task only contains samples from one domain. The challenge of these works is to determine the domains to which the tasks belong. However, in our work, each task contains samples from different domains. The challenge is to utilize domain relevance to transfer knowledge between different domains for better performance.

Therefore, to solve the cold-start problem, it is necessary to propose a model that makes full use of the advantages of cross-domain knowledge and introduce a model optimization strategy simultaneously. However, this task faces the following challenges: 1) How can the problem of cross-domain cold start be rationally redefined so that it can be applied to meta-optimization methods? Unlike traditional machine learning, meta-learning has special requirements for data and scenarios. Although the method of applying meta-learning to recommender systems has matured, determining how to transform the cold-start problem of cross-domain scenarios into a problem suitable for meta-optimization is still a challenge. 2) How can cross-domain knowledge transfer be achieved in a meta-learning setting? The current meta-learning-based models that can alleviate the cold-start problem are always based on simple MLP networks, which severely limit the expression ability yielded by the network and the obtained cross-domain knowledge. The introduction of transfer learning networks will inevitably face the complex problem of adaptation between transfer learning and meta-learning. 3) How can the efficiency of the resulting model be ensured? Meta-learning usually consumes more resources and time than traditional training methods. Moreover, with a more complex network structure and larger amounts of data, this problem becomes more serious [12].

Considering the above challenges, we propose a novel **c**ross-**d**omain **r**ecommendation model via **meta**-learning called **MetaCDR**, which solves the cold-start problem through two resources: cross-domain knowledge and an optimization model. We define the cold-start problem in cross-domain scenarios as a new few-shot problem and optimize it with model-agnostic meta-learning (MAML) [12]. In this model, a module called **DKMT** is designed based on a CSN [37] to perform **d**omain-**k**nowledge **t**ransfer in the **m**eta-learning setting. Finally, we propose a pretraining strategy to reduce the amount of computer resources and time required for model training, thereby enhancing the practicality of MetaCDR.

The contributions of this paper are as follows:

1) We design a novel recommendation model with transfer learning and meta-learning called MetaCDR to solve the cold-start problem. To the best of our knowledge, this is the first attempt to solve this problem from the viewpoint of both cross-domain knowledge and model optimization.
2) We propose a module called DKMT, which is designed specifically for recommender systems, to perform knowledge transfer between different domains.
3) We introduce a pretraining strategy for MetaCDR to reduce the amount of resources and time consumed while achieving similar effects.
4) A sufficient number of experiments are performed to prove that the results of MetaCDR are significantly better than those of several state-of-the-art methods in various scenarios. We also conduct an ablation experiment and detailed analysis to verify the impact of each component of MetaCDR and show the effectiveness of DKMT.

The structure of this paper is as follows: Section 2 introduces the related work. Section 3 defines the cold-start problem in cross-domain scenarios. Section 4 describes the structure and training process of MetaCDR in detail. Section 5 introduces the experimental settings and analyzes the results. In Section 6, we conclude this paper and introduce our future work.

2 Related Work

2.1 Cross-Domain Recommendation

Cross-domain recommendation (CDR) [68] is a commonly used method for solving the cold-start problem [41] by alleviating data sparseness. By transferring and sharing information across different domains, the relationships between the domains and semantics of user preferences can be explored to generate better recommendations [47], [48]. The key to this technology is the method of learning the

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

complex relationships between different domains. Recently, many CDR approaches have been proposed [13], [28], [55]. Man *et al.* [36] proposed embedding and mapping methods that model domain relationships through neural networks. Utilizing a dual-objective optimization method, Zhu *et al.* [66] achieved simultaneous performance improvements in both the source and target domains. Hu *et al.* [22] proposed a deep cross network to realize the two-way transfer of knowledge between the two domains. Liu *et al.* [33] extended CoNet with image information to extract users' aesthetic preferences. Zhao *et al.* [62] integrated like-minded users with an end-to-end framework to further enhance the effect of CDR. Krishnan *et al.* [25] leveraged the contextual invariance across domains to simultaneously develop cross-domain and cross-system recommendations. Bonab *et al.* [2] explored different market-adaptation techniques inspired by state-of-the-art domain adaptation and meta-learning approaches and proposed a neural approach for market adaptation. Li *et al.* [29] presented a debiasing learning-based cross-domain recommendation framework with causal embedding to correct the data selection bias in cross-domain scenarios with a generalized propensity score and to estimate the propensity score when domain-specific confounders are unobserved. Sahu *et al.* [44] utilized matrix factorization, by which a rating matrix is decomposed into several submatrices. Li *et al.* [27] proposed a novel CDR method via regression analysis for cold-start users who never rated items in the target domains.

However, most of the current transfer-learning-based methods are devoted to sharing information more effectively between domains by improving the complex structures of cross-domain networks while ignoring the importance of the optimization for solving the few-shot problem. In this paper, we propose a novel model that incorporates a transfer-learning-based CDR network with an optimization approach to enhance the ability of the overall model to solve the cold-start problem.

2.2 Meta-Learning Recommendation

Meta-learning [21] is also known as learning how to learn. Unlike traditional machine learning methods, a meta-learning model is trained through many separate tasks to learn their similarities and differences and to obtain a base model that can be adapted to new tasks with rapid updating [52], [58]. Common meta-learning methods can be divided into three categories: metric-based [6], [45], [50], memory-based [14], [46], and optimization-based [30], [38], [43] approaches. Previous works have tried to utilize a variety of meta-learning methods in recommender systems to solve the cold-start problem and achieve good results.

Vartak *et al.* [49] used a meta-learning-based method to predict users preferences for tweets based on their historical clicks. Du *et al.* [10] predicted user behavior via sequential recommendations in different domains using meta-learning. However, this method only learns common initialization parameters for each domain and does not consider the accurate alignment of fine-grained information across domains. Bharadhwaj [1] modeled each input user as a task with the optimization-based meta-learning method (MAML). Lee *et al.* [26] extended the above method by optimizing the parameters of the model in groups. Lu *et al.* [34] introduced heterogeneous information networks as additional information in a meta-learning environment to further alleviate the cold-start problem. Dong *et al.* [9] used a memory-augmented neural network to improve the model effect with respect to solving the cold-start problem. Zheng *et al.* [64] used a matching network to address the sequential recommendation cold-start problem without side information. Yu *et al.* [59] solved the problem of neglecting minor users through a meta-learning approach with a personalized adaptive learning rate. Lin *et al.* [32] further alleviated the cold-start problem with neural processes. Zhu *et al.* [69] reduced the bias toward limited overlapping users in the embedding and mapping approach via a meta-network. Feng *et al.* [11] developed a contextual modulation meta-learning framework for efficient and complete recommendation. Zhu *et al.* [71] proposed an embedding cold-start approach with meta-scaling and shifting networks to avoid the effects of noisy interactions. However, current methods do not consider the role of cross-domain knowledge transfer in meta-learning recommendation.

3 Preliminaries

In this section, we first provide a specific definition for the cold-start problem in CDR. Then, as the base model for the new work, a feature embedding method and a multilayer feed-forward neural network for personality prediction are introduced.

3.1 Problem Formulation

Two different domains (such as movies and books), both containing user features, item features, and interaction records, are called the source domain D_s and target domain D_t according to the interaction sparsity difference between them. The users who appear in both domains are called overlapping users. The features of these overlapping users are represented as a set U, and item features are represented as sets I_s and I_t. The interaction information between users and items can be expressed as $R_{u,s}$ and $R_{u,t}$ by implicit feedback (such as clicks, browsing or likes) [20] or explicit feedback (such as ratings) [26]. It is worth mentioning that interaction records for new users or items are often scarce; this is called the cold-start problem in CDR.

Our task is to use the features of users, items, and the interaction records among them to train a model to make predictions regarding the users' item ratings. The function is expressed as follows:

$$\hat{r}_{u,s}, \hat{r}_{u,t} = f_\theta(u, i_s, i_t), \tag{1}$$

where $\hat{r}_{u,s}$ and $\hat{r}_{u,t}$ are the predictions of the ratings of user $u \subset U$ for items $i_s \subset I_s$ and $i_t \subset I_t$ from source domain D_s and target domain D_t, respectively. f is the predicted model, and θ is the parameter of f.

In our model, we treat each domain as a separate recommendation task and adopt a single-domain method to model each task separately. Then, a cross-domain network is used to connect the two domains and perform knowledge transfer. A meta-learning strategy is used to train the domain models jointly. In the next section, the basic models are introduced before MetaCDR.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

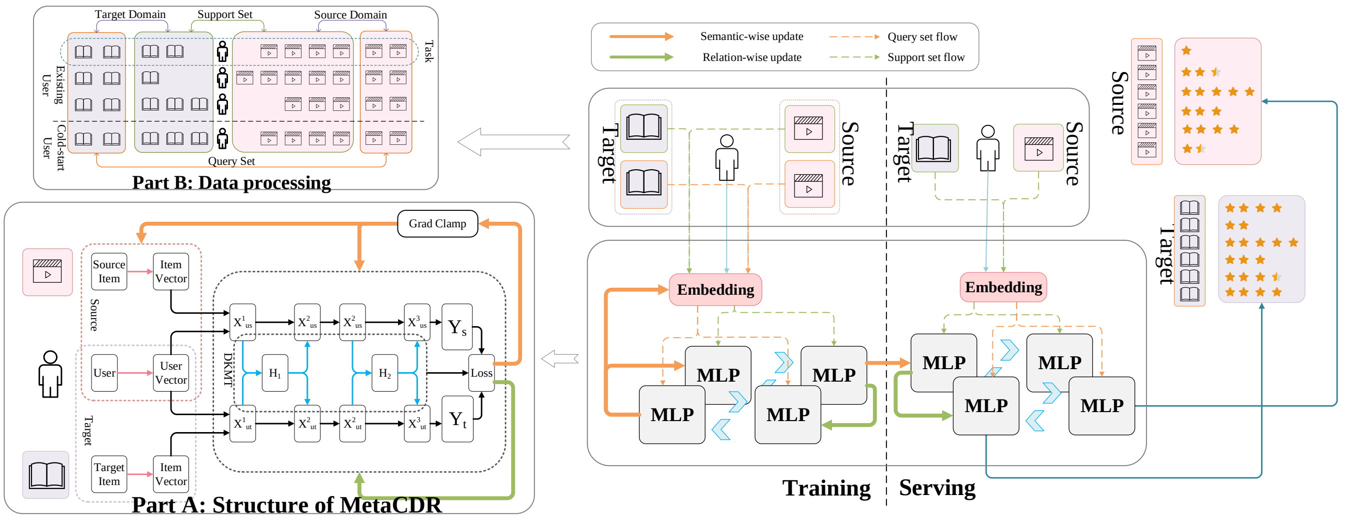

Fig. 3. The overall workflow, structural details, and data processing procedure of MetaCDR.

3.2 Embedding and Recommendation

In this section, we introduce the embedding method and the structure of the basic model for a single domain.

Embedding: $u \rightarrow e_u, i \rightarrow e_i$. Traditional recommender systems use one-hot vectors to represent the IDs of users and items, but these systems can only predict the interactions between existing users and items; they cannot learn user preference information. When faced with new users or items, a one-hot vector is helpless. Therefore, we use the demographic information of users (such as their ages, genders, and occupations) and the features of items (such as film directors or book types). These features provide users' potential preferences in the recommendation system. Specifically, we first divide the available numeric information into groups and represent it as integers, convert the category information into one-hot vectors, and then use a dimensional compression matrix for embedding as follows:

$$e_u = f_{\theta_u}(u) = [u_1 p_1; u_2 p_2; u_3 p_3; ......; u_N p_N]^T, \qquad (2)$$

where e_u is the embedding vector of user u. f is the embedding function for users with the parameters θ_u. u_n is an integer or a d_j-dimensional one-hot vector representing user feature $n \in \{1, ..., N\}$, and p_n is the d_e-by-d_j embedding matrix for the corresponding categorical content of user u. $[\cdot;\cdot]$ is the concatenation operation. The items are embedded in a similar way:

$$e_i = f_{\theta_i}(i) = [i_1 q_1; i_2 q_2; i_3 q_3; ......; i_M q_M]^T, \qquad (3)$$

where e_i is the embedding vector of item i. f is the embedding function for items with the parameters θ_i. i_m is an integer or a d_k-dimensional one-hot vector representing an item's feature $m \in \{1, ..., M\}$, and q_m is the d_e-by-d_k embedding matrix for the corresponding categorical content of item i. $[\cdot;\cdot]$ is the concatenation operation. Next, e_u and e_i are connected and fed into the recommendation model.

Recommendation: $(e_u, e_i) \rightarrow \hat{y}_{u,i}$. We use MLPs to model user preferences and predict the ratings, implicit feedback, or dwell times for items. The model can be expressed as:

$$\begin{aligned} \hat{r}_{u,i} &= MLP_{\phi}([e_u; e_i]^T) = W_k^T x_{k-1}, \\ x_{k-1} &= \sigma(W_{k-1}^T x_{k-2} + b_{k-1}), \\ &...... \\ x_1 &= \sigma(W_1^T x_0 + b_1), \\ x_0 &= [e_u; e_i]^T \end{aligned} \qquad (4)$$

where $\hat{r}_{u,i}$ is the model's prediction of user feedback, MLP stands for a multilayer perceptron, and ϕ is the set of its parameters, including a weight matrix W and a bias vector b. σ is the activation function; here, we use the rectified linear unit (ReLU). e_u and e_i are the embedding vectors of users and items, respectively.

4 METACDR

Fig.3 shows the overall workflow of MetaCDR. In this section, the details of MetaCDR are presented. First, we introduce the cross-domain combination method of the network, including the sharing of the user embedding network and the cross-domain connection between fully connected networks in the meta-learning environment. Second, we redefine the cold-start problem in the CDR scenario as a few-shot problem. Third, we propose a meta-optimization method for MetaCDR. In addition, we design a pretraining strategy to greatly reduce the time and resource consumption.

4.1 User Embedding Sharing

Part A of Fig.3 shows the structural details of MetaCDR, and the left side of its structure is the embedding part. To combine the networks of the source and target domains and share information, we first share the user embedding layer so that the same user feature has a consistent initial embedding in different domains, which can help the model focus on learning the mappings of the commodity characteristics

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

between domains. Our method of obtaining the embedding vector for each domain is as follows:

$$V_s = [f_{\theta_u}(u); f_{\theta_s}(i_s)]^T, V_t = [f_{\theta_u}(u); f_{\theta_t}(i_t)]^T, \tag{5}$$

where V_s is the input vector on the source domain side and V_t is the input vector on the target domain side. f represents the embedding functions, and θ_u, θ_s, and θ_t are the parameters of the embedding functions for users, source domain items, and target domain items, respectively.

4.2 Knowledge Transfer between Domains

Before introducing the structure of DKMT, we first review the deep neural network transfer model called CSN [37], which has achieved significant results for multitask learning problems in the field of computer vision. We also note the problems that need to be solved when performing cross-domain knowledge transfer in the recommendation system and meta-learning environment.

Given two convolutional neural network models, the CSN is used to connect the corresponding layers. Specifically, at location (i, j) in the activation map, we have:

$$\begin{bmatrix} \widetilde{x}_A^{ij} \\ \widetilde{x}_B^{ij} \end{bmatrix} = \begin{bmatrix} \alpha_{AA} & \alpha_{AB} \\ \alpha_{BA} & \alpha_{BB} \end{bmatrix} \begin{bmatrix} x_A^{ij} \\ x_B^{ij} \end{bmatrix} \tag{6}$$

where x_A^{ij} and x_B^{ij} represent the current-layer inputs of networks A and B, respectively; α_{AA}, α_{AB}, α_{BA}, and α_{BB} represent the cross-stitch parameters, which are used to implement knowledge transfer; and $\widetilde{x}_A^{ij}$ and $\widetilde{x}_B^{ij}$ are the inputs of the next layer of the networks.

Consider the following three issues: 1) The four weight parameters used by the CSN can only achieve content migration in the same dimensional space, but our model uses an N-layer fully connected neural network with different dimensions for each layer. 2) The CSN assumes that all dimensions of information are equally important. However, unlike images in computer vision, the importance of each user or item dimension in the recommendation system is different and requires an independent weight [5]. 3) The CSN assumes that all information is worth migrating, but it is evident that not every feature is helpful in other domains, at least in the recommendation system. We need to find a way to make the model transfer knowledge more conservatively and to better adapt the meta-learning framework.

To solve the above problems, we design a module to realize **d**omain **k**nowledge **m**eta-**t**ransfer, called DKMT. We show this module on the right side of *Part A* in Fig.3, and we propose solutions to the above three problems.

For the first two problems, we use weight matrices to replace the weight values in the CSN, which is equivalent to a cross-domain fully connected network. This structure is expressed as:

$$\begin{aligned} x_s^l &= W_{ss}^l x_s^{l-1} + H_{st}^l x_t^{l-1} + b_s^l, \\ x_t^l &= W_{tt}^l x_t^{l-1} + H_{ts}^l x_s^{l-1} + b_t^l \end{aligned} \tag{7}$$

where x_s^l and x_t^l represent the outputs of the cross network and are used as the inputs of the next layer of the network in the source and target domains. W_{ss}^l and W_{tt}^l represent the domain-specific weight matrices of the source domain and target domain, respectively, which are used to perform knowledge transfer within the domain. H_{st}^l and H_{ts}^l are the cross-domain weight matrices between the two domains, which are used to perform knowledge transfer from the source to the target domain and from the target to the source domain. x_s^{l-1} and x_t^{l-1} are the inputs of the cross network and are also the outputs from the last layer of the network with respect to the two domains. b_s^l and b_t^l are the biases in the two domains. In contrast to the CSN, H_{st}^l and H_{ts}^l are $d^l \times d^{l-1}$-dimensional parameter matrices, where d^l is the dimensionality of x_s^l and x_t^l, and d^{l-1} is the dimensionality of x_s^{l-1} and x_t^{l-1}. In this way, we perform knowledge transfer between different-dimensional layers.

Since the effectiveness of meta-learning can be significantly reduced on complex models, we set H_{st}^l and H_{ts}^l to the same matrix H^l to reduce the complexity of the model:

$$\begin{aligned} x_s^l &= W_{ss}^l x_s + H^l x_t^{l-1} + b_s^l, \\ x_t^l &= W_{tt}^l x_t + H^l x_s^{l-1} + b_t^l, \end{aligned} \tag{8}$$

where H^l is the shared parameter matrix used for knowledge transfer between the two domains. The other parameters are the same as those in formula (7).

For the last problem, we introduce a widely used sparsity-induced regularization method called the least absolute shrinkage and selection operator (LASSO) to the knowledge transfer matrix H^l. As usual, LASSO regularization can help the model filter more useful parameters through sparsity. The calculation of the regularization term can be expressed as:

$$\Omega(H^l) = \lambda \sum_{i=1}^{d^l} \sum_{j=1}^{d^{l-1}} \left| h_{ij}^l \right| \tag{9}$$

where Ω stands for LASSO regularization, H^l is the parameter of the knowledge transfer matrix in the l-th layer, $h_{i,j}^l$ is the (i, j)-th parameter in the matrix, and λ is the hyperparameter used to control the degree of sparsity.

4.3 Multilabel Loss Function

We define the loss as the sum of the mean square errors (MSEs) of the two networks and the regularization term. At this stage, the loss function of MetaCDR can be expressed as:

$$\mathcal{L} = \mathcal{L}_s(U, I_s, R_s) + \mathcal{L}_t(U, I_t, R_t) + \Omega((H^l)_{l=1}^L) \tag{10}$$

$$\mathcal{L}_s = \frac{1}{K} \sum_{k=1}^{K} \left(r_{u,i_s}^k - \hat{r}_{u,i_s}^k \right)^2 \tag{11}$$

$$\mathcal{L}_t = \frac{1}{K} \sum_{k=1}^{K} \left(r_{u,i_t}^k - \hat{r}_{u,i_t}^k \right)^2 \tag{12}$$

$$\hat{r_s}, \hat{r_t} = f_\theta(u, i_s, i_t) \tag{13}$$

where $\mathcal{L}$ represents the overall loss; $\mathcal{L}_s$ and $\mathcal{L}_t$ represent the MSE loss functions for the source and target domains, respectively; $u \in U$, $i_s \in I_s$, and $i_t \in I_t$ are the users and items from the two domains; $r_s \in R_s$ and $r_t \in R_t$ are the ratings of the two domains; and $\hat{r}_s$ and $\hat{r}_t$ are their predicted values. Ω represents the regularization term, and H^l $(l \in \{1, ..., L\})$ are the parameters of DKMT. K is the number of interaction records.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

Above, the network structure of MetaCDR is introduced. Next, we redefine the cold-start problem as a few-shot problem and introduce the utilized training and test processes based on meta-learning.

4.4 Data Processing

Part B of Fig.3 shows the employed data preprocessing method. In MetaCDR, each task includes a user, the items the user has rated in the source and target domains, and the corresponding ratings. Each domain of each task is divided into support sets and query sets.

To closely approximate a real scenario, we derive inspiration from the well-known meta-learning model matching network [50], use the latest fixed-length item sequence that the examined user has interacted with as the query set, and use the remaining items as the support set. Finally, we define the users or items that do not appear in the training phase as cold-start users or items.

4.5 Hierarchical Meta-Optimization

We utilize the idea of optimization-based meta-learning to optimize our model. We divide the input data into tasks according to different users. This can be understood as training a unique model for each user in the adaptation phase to better adapt to user interests and preferences.

As shown in Fig.3, we divide the training process into two parts: **relation-wise** and **semantic-wise** updating. We hierarchically update different parameters during meta-training. During relation-wise updating, the loss is computed via the task's support set and used to update the model based on a few steps of gradient descent for a task-adaptive model. Since the embedding matrix of the recommendation model occupies the vast majority of all parameters, updating all parameters at this stage will increase the computational cost and make it difficult to effectively approximate the task within a limited number of update steps. In addition, updating the parameters of the embedding layers during relation-wise updating will lead to frequent changes in user and item embeddings, which is not conducive to the model focusing on learning domain relations. Inspired by [40], we only update the parameters of MLP and DKMT in the relation-wise update.

We tried three different meta-optimization strategies; however, only the basic paradigm of gradient-based meta-learning is introduced here, and MetaCDR under each optimization strategy is introduced in Section 5.4.

Fig.3 shows the meta-optimization process of MetaCDR. To avoid repetition, we only show the equations in detail in Algorithm 1 and do not repeat them in the text. We divide the parameters that the model needs to optimize into three groups: 1) $\theta_e = \{\theta_u, \theta_s, \theta_t\}$ are the parameters of the embedding network. 2) $\theta_m = \{W^{\{1,\ldots,L\}}, b^{\{1,\ldots,L\}}\}$ are the parameters of the fully connected neural network. 3) $\theta_h = H^{\{1,\ldots,L-1\}}$ are the parameters of DKMT.

In the test phase, we use a small number of new user interaction records in the two domains to update the base model M_{base}. With the advantages of meta-learning and the transfer of knowledge between domains, the model can quickly and accurately adapt to user preferences. After that, the model can provide rating predictions for other items.

Algorithm 1: Training of MetaCDR.

Data: a set of meta-training tasks τ; each task $\tau_u \in \tau$ consists of two support sets τ^{sou}_{sup} and τ^{tar}_{sup} from different domains and two query sets τ^{sou}_{que} and τ^{tar}_{que} from different domains;
Input: semantic-wise and relation-wise update steps: s and r; global update and local update learning rates: α and β
Result: the trained base model;

Randomly initialize the base model M_{base} with the parameters $\theta = \{\theta_e, \theta_m, \theta_h\}$
while no convergence **do**
    sample a batch of tasks $\tau_u \sim p(\tau)$
    for task τ_u w.r.t. user u **do**
        do relation-wise update via task τ_u;
    end
    do semantic-wise update;
end

4.6 Pretraining Strategy

The combination of the complex network structure and meta-optimization and the large amount of data brought by the combination of two domains (with the Cartesian product) not only dramatically reduces the efficiency of the model (approximately 12 GB of GPU memory and 2 hours are required) but also induces a risk of nonconvergence. Therefore, we set a pretraining method to optimize the training process.

We first use a method similar to neural collaborative filtering to train two single-domain network parameters θ_e and θ_m with the traditional training method in an alternating manner. Then, we use the pretrained parameters to initialize the corresponding parameters in MetaCDR, randomly initialize the parameters θ_h of DKMT and fix the parameter θ_e. Here, we adopt a random strategy to select training samples from the combined data of the two domains. Finally, we obtain an evaluation model with a small number of training epochs.

The **p**re**t**rained model is called **MetaCDR-PT**. Section 5 proves that this pretraining method ensures the effectiveness of the model while greatly improving the training efficiency.

5 Experiments and Discussion

In this section, we summarize the experimental results and analyze them to answer the following research questions (RQs): **(RQ1)** How does MetaCDR perform compared to the state-of-the-art methods in various cold-start scenarios? **(RQ2)** How do the hyperparameters affect MetaCDR? **(RQ3)** How do meta-learning (MAML) and transfer learning (DKMT) affect MetaCDR? **(RQ4)** What is the time efficiency of MetaCDR-PT? **(RQ5)** How sensitive is MetaCDR to side information, feedback patterns, and network architecture? **(RQ6)** How can MAML and DKMT help to improve MetaCDR?

TABLE 1
Dataset Statistics.

Datasets		MovieLens	Douban
User Count		1485 (19.7%)	1208 (12.1%)
User Feature		gender, age, occupation, area	location
Item Count	Source	2,529	2,033
	Target	663	4,542
Item Feature	Source	rate, genre, director, actor	genre, director, language, actor, country
	Target	rate, genre, director, actor	theme, author, language
Rating Count	Source	79,193	98,142
	Target	42,585	77,712
Rating Range		1-5	1-5

5.1 Experimental Setup

5.1.1 Datasets

We choose two real-world datasets to evaluate our model: MovieLens 1M[1] [18] and Douban[2]. Table 1 shows the details of these datasets.

MovieLens 1M contains user rating records from the IMDB[3] for movies, as well as the features of users and movies. Similar to the method in [31], we divide the movies into a source domain (before 1998) and a target domain (after 1998) according to their release years, and the ratio is approximately 4:1. Then, to simulate a cold-start problem, we filter out the users with between 13 and 60 interaction records in each domain, and the average gap in the interaction counts between the domains is approximately 24.77. The last ten interaction records are used as query sets, and the rest are used as support sets. In particular, for fairness, we use the support set in the evaluation data for the meta-learner as the training data for the non-meta-learning methods.

Douban is a real-world dataset crawled from the Douban website [67]. It contains many user ratings on movies, music, books and other items. We select movies and books as the source domain and target domain, respectively, and select users with between 13 and 80 interactions in both domains as the available data; the average gap in the interaction counts is approximately 17.48. Similar to the processing method used for the MovieLens dataset, we divide the data into a support set and query set for each task (user). For the non-meta-learning methods, the support sets in the evaluation set are also used as their training data.

For each dataset, the division ratio of the training, validation and test sets is 7:1:2. We set up four scenarios on each dataset. 1) **Warm-Start**: The model is evaluated with existing users and items. 2) **User Cold-Start**: The model is evaluated with new users and existing items. 3) **Item Cold-Start**: The model is evaluated with existing users and new items. 4) **User-Item Cold-Start**: The model is evaluated with new users and new items.

In addition, we adopt a real-world dataset from the e-commerce platform Amazon[4] to study the effect of overlapping users, side information and feedback patterns on the models.

1. https://grouplens.org/datasets/movielens/
2. https://www.douban.com/
3. https://www.imdb.com/
4. https://www.amazon.cn/

5.1.2 Baselines

We compare MetaCDR with three categories of methods: **(1) traditional methods** (FM and NeuMF), **(2) cross-domain transfer methods** (EMCDR, CSN, and SCoNet), and **(3) meta-learning methods** (MeLU, MetaCS-DNN, and MAMO).

FM [42] is a classic method for recommendation based on the features of items and users. It can predict the personalized preferences of users by exploring the potential relationships between users and items through existing content and additional feature information.

NeuMF [20] is a state-of-the-art collaborative filtering model based on an MLP and generalized matrix factorization (GMF). We define its output module as a linear layer for rating prediction and embed the features of users and items as its inputs for the cold-start problem.

EMCDR [36] is an embedding and mapping approach for CDR. It first learns the embeddings of entities in the source domain and target domain and then uses a neural network to capture the mapping function between the embeddings of the same entity. In this paper, the two domains are set as the source domain and the target domain in turn.

MMoE [35] is a well-known multitask learning framework. It utilizes a gating network for each task on a mixture-of-experts structure. As suggested by [63], the embedding parameters are shared across all experts. The embedding vectors of users and items from the two domains are given to each expert. We set two towers to output scores for the two domains.

CSN [37] is a multitask model with a deep fusion network that was first applied in computer vision. Two networks are connected through cross-stitching to optimize the results with multitask learning.

SCoNet [22] is a state-of-the-art transfer learning model designed for CDR; it uses a parameter matrix to transfer knowledge between domains and uses Lasso to limit the degree of knowledge transfer.

MetaCS-DNN [1] is optimized with an N-layer fully connected network to obtain embeddings and ratings through an idea similar to that of MAML. By converting each user into a task, the cold-start problem is transformed into a few-shot problem.

MeLU [26] is designed with a similar idea to that of MetaCS-DNN, except that it optimizes its personalized recommender network at all stages and only optimizes the general embedding network during the global update stage. That is, the parameters of the embedding network are updated only as the model learns about the commonalities between users.

MAMO [9] is designed with a memory-augmented neural network to store the personalized user gradient information, further improving the accuracy of recommendations in cold-start scenarios.

TMCDR [69] is a meta-learning-based embedding and mapping approach for cross-domain recommendation. Unlike EMCDR, TMCDR utilizes a meta-network for the mapping stage. For fairness, we set an MLP as the base model

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

TABLE 2
The Experimental Results on MovieLens, with the Best Results Shown in Bold.

Scenario	Model	Source Domain (Before 1998)			Target Domain (After 1998)		
		MAE↓	RMSE↓	nDCG@5↑	MAE↓	RMSE↓	nDCG@5↑
Warm-Start	FM	1.0858	1.3146	0.7371	1.1721	1.6397	0.7047
	NeuMF	0.9176	1.1423	0.7945	0.9575	1.3826	0.7322
	EMCDR	0.9004	1.1305	0.8044	0.9571	1.3547	0.7326
	MMoE	0.8876	1.1145	0.7621	0.9233	1.3015	0.7473
	CSN	0.9332	1.2994	0.7529	0.9552	1.2974	0.7341
	SCoNet	0.8899	1.1298	0.8048	0.9096	1.2538	0.7425
	MetaCS-DNN	0.8298	1.0776	0.8129	0.8406	1.1711	0.7935
	MeLU	0.8024	1.0251	0.8285	0.8213	1.1556	0.8139
	MAMO	0.8001	1.0236	0.8207	0.8211	1.1203	0.8194
	TMCDR	0.8288	1.0864	0.8186	0.9412	1.2710	0.7664
	MetaCDR	**0.7885**	**1.0142**	**0.8358**	**0.8163**	1.1165	0.8211
	MetaCDR-PT	0.7832	1.0154	0.8331	0.8191	**1.0884**	**0.8215**
User Cold-Start	FM	1.0022	1.6843	0.6926	1.3459	1.749	0.6881
	NeuMF	0.9569	1.1127	0.7741	1.0588	1.3878	0.7502
	EMCDR	0.9433	1.0094	0.7820	1.1136	1.2355	0.7566
	MMoE	0.9486	1.1051	0.7793	1.1098	1.2087	0.7593
	CSN	1.1168	1.3112	0.7491	1.0108	1.3796	0.7168
	SCoNet	0.9474	1.0656	0.7724	0.9759	1.1714	0.7894
	MetaCS-DNN	0.8471	0.9832	0.8042	0.8413	1.0355	0.8202
	MeLU	0.8189	0.9941	0.8089	0.8496	1.0983	**0.8436**
	MAMO	0.8260	1.0105	0.7933	0.8551	1.1151	0.8333
	TMCDR	0.8231	1.0544	0.8008	0.8496	1.1048	0.8132
	MetaCDR	**0.7904**	**0.9706**	**0.8216**	**0.8201**	**0.9947**	0.8347
	MetaCDR-PT	0.7927	0.9758	0.8208	0.8257	0.9980	0.8306
Item Cold-Start	FM	1.2273	1.6056	0.7071	1.4003	1.6937	0.6681
	NeuMF	1.0289	1.2082	0.7571	1.3447	1.2634	0.6972
	EMCDR	1.1222	1.236	0.7384	1.0178	1.2148	0.7512
	MMoE	0.9590	1.2221	0.7525	0.9474	1.2018	0.7554
	CSN	1.1691	1.2973	0.7301	1.1366	1.2423	0.7268
	SCoNet	0.9677	1.1505	0.7622	0.9638	1.1533	0.7597
	MetaCS-DNN	0.9049	1.1277	0.7765	0.9221	1.1392	0.7825
	MeLU	0.8868	1.0478	0.8033	0.9127	1.1055	0.8261
	MAMO	0.8954	1.0992	0.7969	0.9212	1.1015	0.8111
	TMCDR	0.9117	1.1144	0.8016	0.9137	1.1191	0.7990
	MetaCDR	**0.8673**	**1.0241**	**0.8126**	0.8743	1.0362	**0.8469**
	MetaCDR-PT	0.8689	1.0366	0.8102	**0.8906**	**1.0312**	0.8483
User-Item Cold-Start	FM	1.3683	1.6969	0.6835	1.3379	1.7891	0.6768
	NeuMF	1.1753	1.3210	0.7588	1.1908	1.3542	0.7362
	EMCDR	0.9936	1. 2511	0.7704	1.0458	1.1965	0.7529
	MMoE	0.9847	1.2116	0.7855	1.0565	1.2114	0.7583
	CSN	1.0425	1.2223	0.7633	1.1147	1.2473	0.7396
	SCoNet	0.9791	1.1879	0.7742	0.9869	1.1892	0.7821
	MetaCS-DNN	0.9253	1.1334	0.8135	0.9523	1.1481	0.7905
	MeLU	0.9009	1.0746	0.8114	0.9291	1.1266	0.7971
	MAMO	0.8414	1.0668	0.8174	0.8804	1.1221	0.8070
	TMCDR	0.8811	1.0859	0.7868	0.9721	1.1563	0.8059
	MetaCDR	**0.8308**	**1.0494**	0.8241	**0.8575**	**1.1104**	**0.8183**
	MetaCDR-PT	0.8394	1.0616	**0.8258**	0.8592	1.1285	0.8022

in the transfer stage for each domain and obtain the embedding vectors based on the trained embedding layers. Similar to EMCDR, the two domains are set as the source domain and the target domain in turn.

The source codes of FM[5], NeuMF[6], EMCDR[7], SCoNet[8], MeLU[9] and MAMO[10] are openly available, and we modify their data processing and output components to apply them to our experiments. We implement MetaCS-DNN with the code of MeLU, which has a similar idea. We implement CSN ourselves.

5. https://github.com/lyst/lightfm
6. https://github.com/hexiangnan/neural_collaborative_filtering
7. https://github.com/MaJining92/EMCDR
8. http://home.cse.ust.hk/~ghuac/
9. https://github.com/hoyeoplee/MeLU
10. https://github.com/dongmanqing/Code-for-MAMO

5.1.3 Parameter Settings

For MetaCDR, we set MAML as the base meta-learner; the learning rates for semantic-wise and relation-wise updating are set to α=0.01 and β=0.001, respectively; the regularization parameter λ is set to 0.01; and the numbers of steps of relation-wise and semantic-wise updating are set to 5 and 1, respectively. The embedding dimensionality of each feature is set to 32. Two $[32 \times 8 \rightarrow 64 \rightarrow 64 \rightarrow 1]$ MLPs are used as the basic model. The rectified linear unit (ReLU) is employed as the activation function, and optimization is conducted by adaptive moment estimation (Adam) [4]. We also use batch normalization [23] to speed up the convergence of the model. We set the batch size to 16 tasks, and the maximum numbers of epochs are set to 30 and 20 for MetaCDR in MovieLens and Douban, respectively. For MetaCDR-PT, the maximum numbers of epochs are set to

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

TABLE 3
The Experimental Results on Douban, with the Best Result Shown in Bold.

Scenario	Model	Source Domain (Movie)			Target Domain (Book)		
		MAE↓	RMSE↓	nDCG@5↑	MAE↓	RMSE↓	nDCG@5↑
Warm-Start	FM	0.8222	0.8568	0.7990	0.8332	0.9106	0.7464
	NeuMF	0.7309	0.8063	0.8244	0.7084	0.8542	0.7836
	EMCDR	0.7241	0.8117	0.8001	0.7411	0.9011	0.7739
	MMoE	0.7010	0.8088	0.8052	0.7298	0.8621	0.7833
	CSN	0.7964	0.8245	0.7554	0.7262	0.9004	0.7791
	SCoNet	0.7044	0.7822	0.7549	0.7253	0.8856	0.7985
	MetaCS-DNN	0.6597	0.7596	**0.8582**	0.6829	0.8553	0.8496
	MeLU	0.6303	0.7555	0.8476	0.6638	0.8468	0.8542
	MAMO	0.6575	0.7681	0.8298	0.6977	0.8507	0.8558
	TMCDR	0.7079	0.7836	0.7666	0.7011	0.8768	0.7922
	MetaCDR	**0.6253**	**0.7494**	0.8505	**0.6413**	**0.7922**	**0.8631**
	MetaCDR-PT	0.6246	0.7461	0.8514	0.6439	0.7952	0.8626
User Cold-Start	FM	0.7016	0.8997	0.7479	0.8043	0.8527	0.7673
	NeuMF	0.7476	0.8323	0.7312	0.7279	0.8859	0.7946
	EMCDR	0.8215	0.8738	0.7325	0.8432	0.9023	0.8041
	MMoE	0.7884	0.8585	0.7236	0.7059	0.8522	0.8114
	CSN	0.8660	1.0012	0.6688	0.7414	0.8516	0.8335
	SCoNet	0.8053	0.8354	0.8283	0.7087	0.8302	0.8264
	MetaCS-DNN	0.6484	0.7979	0.8279	0.6795	0.8079	0.8319
	MeLU	0.6273	0.7337	0.8450	0.6712	0.7911	0.8530
	MAMO	0.6266	0.7588	0.8222	0.6777	0.8127	0.8331
	TMCDR	0.6655	0.8172	0.7540	0.7119	0.8377	0.7974
	MetaCDR	**0.6081**	**0.7322**	**0.8672**	**0.6697**	**0.7645**	**0.8658**
	MetaCDR-PT	0.6101	0.7359	0.8654	0.6718	0.7669	0.8638
Item Cold-Start	FM	0.8010	0.9567	0.7969	0.8149	0.9181	0.8098
	NeuMF	0.7604	0.8979	0.7044	0.8293	0.8666	0.7801
	EMCDR	0.7699	0.9003	0.6927	0.8959	0.9007	0.8066
	MMoE	0.8099	0.9167	0.6868	0.8104	0.9120	0.7158
	CSN	0.8326	0.9522	0.7542	0.8008	0.8962	0.7963
	SCoNet	0.7517	0.8532	0.8213	0.7876	0.8846	0.8012
	MetaCS-DNN	0.6914	0.8125	0.8698	0.7124	0.8313	0.8555
	MeLU	0.6545	0.7988	0.8672	0.6621	0.8154	0.8435
	MAMO	0.6877	0.8151	0.8633	0.6911	0.8142	0.8562
	TMCDR	0.6845	0.8221	0.8574	0.6891	0.8173	0.8298
	MetaCDR	**0.6354**	**0.7846**	**0.8881**	0.6869	0.8077	**0.8591**
	MetaCDR-PT	0.6393	0.7876	0.8811	**0.6856**	**0.7971**	0.8583
User-Item Cold-Start	FM	0.8415	0.9936	0.7757	0.8831	1.3012	0.7969
	NeuMF	0.8103	0.9111	0.8396	0.8028	0.9235	0.8112
	EMCDR	0.8274	0.9504	0.7961	0.8327	1.0158	0.8049
	MMoE	0.8206	0.9541	0.7802	0.8029	0.9816	0.8122
	CSN	0.8166	0.9386	0.7992	0.8283	0.9326	0.8092
	SCoNet	0.7533	0.8952	0.8433	0.8007	0.8822	0.8253
	MetaCS-DNN	0.6915	0.8181	0.8513	0.7584	0.8304	0.8204
	MeLU	0.6676	0.7761	0.8572	0.7236	0.7989	0.8355
	MAMO	0.6569	0.7715	0.8635	0.6895	0.7886	0.8385
	TMCDR	0.7337	0.8516	0.8447	0.8158	0.9002	0.8204
	MetaCDR	**0.6379**	**0.7505**	**0.8823**	**0.6593**	**0.7726**	**0.8461**
	MetaCDR-PT	0.6598	0.7889	0.8656	0.6612	0.7941	0.8409

only 10 and 5.

5.1.4 Evaluation Metrics

We adopt three evaluation metrics, the mean absolute error (MAE), root-mean-square error (RMSE), and normalized discounted cumulative gain at rank K (nDCG@K), to evaluate MetaCDR and the other baseline models. Here, we set $K = 5$. The specific calculation method is as follows:

$$MAE = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|N_u|} \sum_{i \in I_u} |r_{u,i} - \hat{r}_{u,i}| \tag{14}$$

$$RMSE = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|N_u|} \sqrt{\sum_{i \in I_u} (r_{u,i} - \hat{r}_{u,i})^2} \tag{15}$$

$$nDCG@K = \frac{1}{|U|} \sum_{u \in U} \frac{DCG@K}{IDCG@K} \tag{16}$$

$$DCG@K = \sum_{k=1}^{K} \frac{2^{r_{u,k}} - 1}{log_2(k+1)} \tag{17}$$

where U is the user set utilized in the test, I_u denotes the interaction record of user u, and $r_{u,i}$ and $\hat{r}_{u,i}$ are the real rating and the predicted rating, respectively. The IDCG calculates the best possible DCG for each user. The MAE and RMSE calculate the degree of error incurred when predicting ratings, and lower MAE and RMSE values correspond to better model performance. The NDCG represents the overall performance of the model for a certain user, and a higher NDCG indicates better performance.

5.1.5 Environment

All our experiments are conducted on a Linux server with a GPU (Tesla V100 with 32 GB of RAM) and CPU (Intel Xeon

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

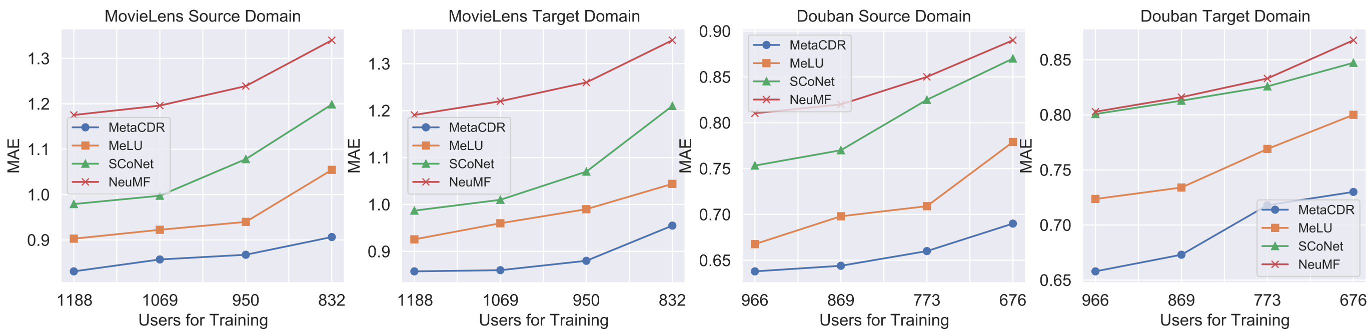

Fig. 4. Impact of the count of overlapping users.

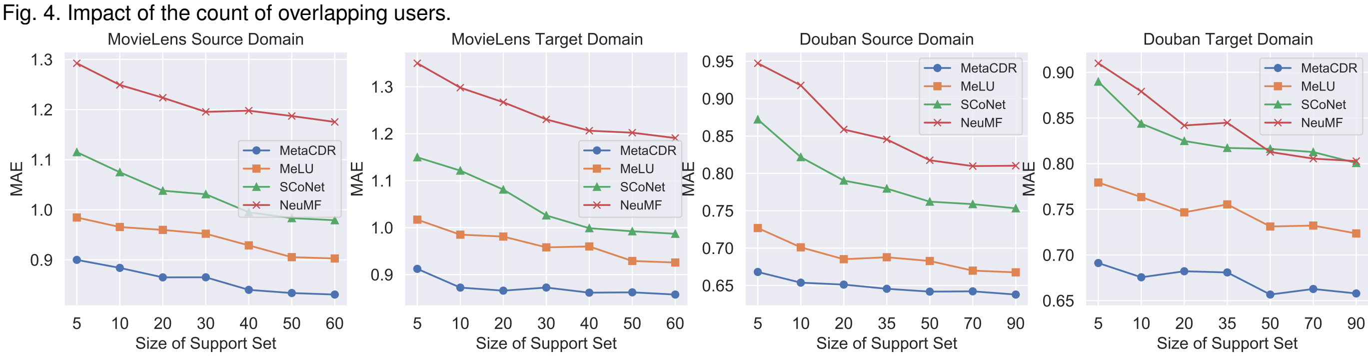

Fig. 5. Impact of the count of items in the support sets.

E5-2698 v4). The operating system is Ubuntu[11] 16.04.6, and Python[12] version 3.6.8 is used. The model is built based on the deep learning library PyTorch[13], version 1.4.1.

5.2 Performance Comparison (RQ1)

In this section, we compare MetaCDR and its pretrained version MetaCDR-PT with several state-of-the-art baseline models. We design four scenarios for each dataset: warm-start, user cold-start, item cold-start, and user-item cold-start.

Table 2 and Table 3 show the performance of all models in different domains of the two datasets for the four scenarios. It is obvious that MetaCDR and MetaCDR-PT outperform the state-of-the-art models for most of the datasets and scenarios, especially in cold-start scenarios with more severe conditions. Further analysis of the results shows that the meta-learning method usually performs better than the traditional methods and normal cross-domain methods when faced with a cold-start problem.

According to an in-depth analysis of the Amazon dataset, the average ratio of overlapping users to the total number of users in any two domains is less than 12% [24]. Fig.6 shows three pairs of domains as examples. To make the experiment more similar to the real world and evaluate the robustness of the models, we set three small-overlap scenarios for each dataset with fewer overlapping users in the training phase. To make the experiment fair, we use the same user-item cold-start data to evaluate each scenario. Fig.4 shows the performance of our method (MetaCDR), the meta-learning methods (MeLU), the cross-domain methods (SCoNet), and the traditional methods (NeuMF) in the four scenarios. It is evident that with the decrease in the number of overlapping users, MetaCDR exhibits stronger robustness (a smaller increase in MAE) than other baselines and achieves the best accuracy (lowest MAE).

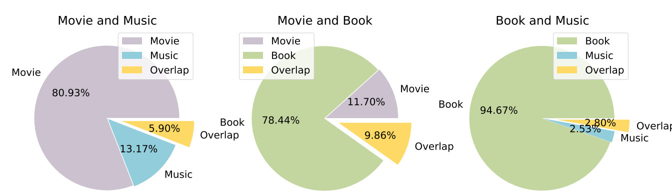

Fig. 6. The proportions of overlapping users among various domains in Amazon. We select three domains as examples: movies, books, and music. The yellow part in each figure represents overlapping users. Notably, the problem in which there is little overlap cannot be ignored.

To test the impacts of different degrees of cold-start problems on the model effects of the results, we test the performance of traditional methods (NeuMF), meta-learning methods (MeLU), cross-domain methods (SCoNet), and our MetaCDR by limiting the maximum length of each support set. The results in Fig.5 show that as the support set decreases, the performance of MetaCDR declines least among all models. Therefore, MetaCDR is sufficiently robust and can address the cold-start problem well when very limited data are available.

5.3 Hyperparameter Analysis (RQ2)

Next, we study the impacts of hyperparameters on MetaCDR by adjusting them. To better demonstrate the

11. https://ubuntu.com/
12. https://www.python.org/
13. https://pytorch.org/

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3208005

TABLE 4
The impacts of the locations and number of DKMTs for MovieLens.

DKMT			MAE	
H^1	H^2	H^3	Source Domain	Target Domain
×	×	×	1.3497	1.1828
✓	×	×	0.9831	0.9677
×	✓	×	1.1004	1.2323
×	×	✓	1.0375	1.3088
×	✓	✓	0.9241	1.1082
✓	×	✓	**0.8175**	0.8651
✓	✓	×	0.8308	0.8575
✓	✓	✓	0.8493	**0.8525**

effect of our model in the most challenging scenario, the experiments below are all performed in the **user-item cold-start** scenario. In this section, we analyze the impacts of three hyperparameters on the effectiveness of the model: the semantic-wise and relation-wise update steps and the regularization coefficient λ.

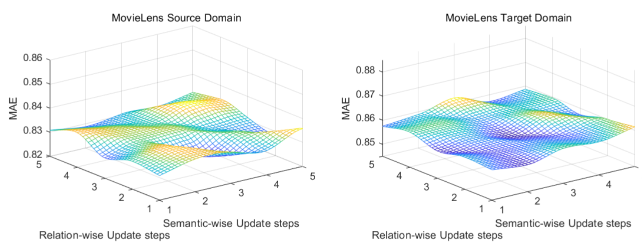

Fig. 7. The impact of the semantic-wise and relation-wise update steps on MetaCDR for MovieLens (user-item cold-start). The left panel is in the source domain, and the right panel is in the target domain.

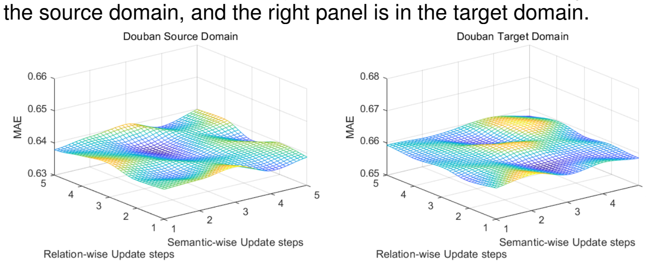

Fig. 8. The impact of the semantic-wise and relation-wise update steps on MetaCDR for Douban (user-item cold-start). The left panel is in the source domain, and the right panel is in the target domain.

Fig.7 and Fig.8 show the impact of the semantic-wise and relation-wise update steps on MetaCDR (MAE). Because similar experimental results are observed in terms of the RMSE and nDCG@5 metrics, we only report the MAE results, and the range of the number of update steps is from 1 to 5. The analysis shows that the impacts of the relation-wise and semantic-wise update steps on MetaCDR are relatively small in both the source and target domains. The model exhibits strong stability. The results show that the number of semantic-wise or relation-wise update steps has little effect on the model results in the user-item cold-start scenario. However, we still choose to perform 5 steps in the relation-wise update phase because in other scenarios, multiple relation-wise updates often bring some improvements to the model.

Fig.9 and Fig.10 show the impact of the regularization coefficient, i.e., the sparsity of the DKMT parameters, on

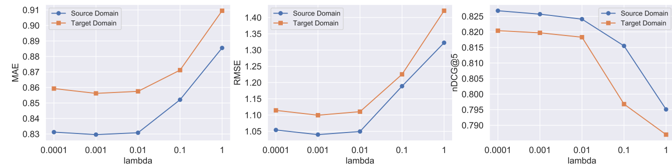

Fig. 9. The impact of λ on MetaCDR for MovieLens (user-item cold-start).

Fig. 10. The impact of λ on MetaCDR for Douban (user-item cold-start).

TABLE 5
The impacts of different meta-learners.

Meta-learner	MovieLens-MAE		Douban-MAE	
	Source	Target	Source	Target
-MAML	0.8308	**0.8575**	0.6379	0.6593
-Meta-SGD	**0.8257**	0.8669	**0.6321**	0.6848
-FOMAML	0.8329	0.8617	0.6499	**0.6504**
-Reptile	0.8562	0.8843	0.6536	0.6833

MetaCDR. We evaluate the effect of λ on the model for the MovieLens and Douban datasets (in terms of the MAE, RMSE, and nDCG@5). We select [0.0001, 0.001, 0.01, 0.1, 1] as the values of λ for the evaluation. Through the experimental results, it can be found that the effect is best when λ is set to approximately 0.01, and as it continues to decrease, the model effect changes slightly. However, when it increases, the model effect decreases significantly. This means that excessively sparse parameters also limit the transfer of cross-domain knowledge. Finally, we choose 0.01 as the regularization coefficient of MetaCDR.

We explore the optimal locations and number of DKMTs through further experiments. The basic model we use is a $[32 \times 8 \to 64 \to 64 \to 1]$ MLP, so we can add three DKMT structures: $H^1 \in \mathbb{R}^{256\times 64}$, $H^2 \in \mathbb{R}^{64\times 64}$, and $H^3 \in \mathbb{R}^{64\times 1}$. We successively change the locations and number of DKMTs and evaluate their impacts. The results are shown in Table 4. Although the trends are different in different domains, overall, the model with DKMTs mostly shows different degrees of improvement than the model without DKMTs. This shows that the DKMT structure is effective. Some models that have DKMTs with fewer parameters exhibit declines in performance after a few epochs, so we set fewer epochs for these models to obtain better results by stopping the process early. Similarly, enabling all DKMT structures does not significantly improve the effectiveness of the model, as the use of too many parameters reduces the computational efficiency of MetaCDR; thus, we set DKMTs in the first two layers and omit them in the last layer.

5.4 Impact of Meta-Learners

To test the impacts of different meta-learners for MetaCDR, we choose four meta-learners:

MAML [12] is a classic gradient-based meta-learner. It uses support sets for task-adaptive local updates and query sets for global updates. We improved it to suit the MetaCDR scenario. During the training procedure, first, all the parameters in the base model M_{base} are initialized, and a copy $M_{base}{}'$ of the parameter set is generated. Second, we randomly select a batch of tasks τ_u (for users u) and feed their support sets as inputs into the MetaCDR model to obtain the prediction results. Third, we calculate the loss $\mathcal{L}$ and gradient G through the results to update the parameters θ_m and θ_h of $M_{base}{}'$ and obtain the meta-model M_{meta}. This step explores the complex cross-domain relationship information, and it can be repeated many times to achieve the desired effect. It is worth noting that at this step, we have not changed the original model M_{base} but only updated M'_{base}. Fourth, the query sets are fed into the meta-model M_{meta} to obtain the loss $\mathcal{L}'$ and gradient G'. In this step, we clamp the gradient range so that the model can be adjusted more conservatively (this is essential for sensitive meta-learning models). Finally, all parameters in the base model M_{base} are updated with the gradient G'. In general, the overall form of the model update can be expressed as:

$$\begin{aligned}\theta = \theta - &\beta \nabla_\theta \mathcal{L}_q(\theta_e, \\ &\theta_m - \alpha \nabla_{\theta_m} \mathcal{L}_s(\theta_e, \theta_m, \theta_h), \\ &\theta_h - \alpha \nabla_{\theta_h} \mathcal{L}_s(\theta_e, \theta_m, \theta_h))\end{aligned} \tag{18}$$

where $\mathcal{L}_s$ and $\mathcal{L}_q$ are the losses in the support and query sets, respectively. Thus far, the model has completed a batch update, and this process is repeated until the model converges. Then, we obtain the trained meta-model.

Meta-SGD [30] is another gradient-based meta-learner based on MAML. The difference is that Meta-SGD can not only update the parameters of the network but also adaptively adjust the learning direction and learning rate in meta-optimization.

First-Order MAML (FOMAML) [38] implements meta-updates with the first-order gradient. Unlike MAML, FO-MAML is updated locally based on each task from a batch in turn, and the global update is based on parameters after the local update. FOMAML is faster and consumes less memory because it does not need to compute Hessian matrices.

Reptile [38] is a new first-order meta-learner. Unlike FOMAML, Reptile does not need a training-test split for each task, which makes it more flexible in certain scenarios. Specifically, Reptile uses the same set of samples from one task for multistep local and global updates. Here, we update the embedding parameters in both semantic-wise and relation-wise updates:

$$\begin{aligned}\theta &= \beta \frac{1}{K} \sum_{k=1}^{K} \theta'_k + (1-\beta)\theta \\ \theta'_k &= \theta - \alpha \nabla_\theta \mathcal{L}(\theta_e, \theta_m, \theta_h)\end{aligned} \tag{19}$$

where K is the number of tasks in each task batch.

According to the results shown in Table 5, in most scenarios, MAML is similar to MetaSGD and FOMAML, and MAML has the most stable effect. Reptile is faster, but it usually performs worse. Therefore, we adopt MAML in our model.

TABLE 6
Training Epochs and Time (Seconds) and Memory (MB) Consumption per Epoch on MovieLens for User Cold-Start. The Results of MetaCDR-PT/PT+ are Reported as Pretraining/Meta-training.

Model	MetaCDR	FOREC	-PT	-PT+
Random Sampling	×	×	✓	✓
Epochs	30	10	10/10	10/10
Time	966.2	139.4	**22.6/79.2**	27.5/77.8
Memory	11141.12	4580.8	**4334.9**	4664.7
MAE	**0.7904**	0.8841	0.7927	0.7920

5.5 Ablation Experiment (RQ3)

Finally, we study the impact of meta-learning (MAML) and transfer learning (DKMT) on MetaCDR with ablation experiments.

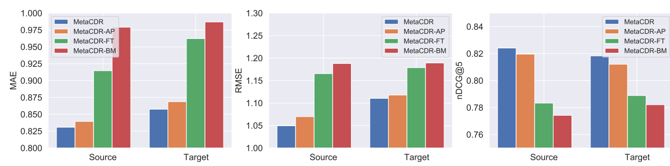

Fig. 11. Analysis of the impact of meta-learning on MetaCDR via various ablation models with MovieLens (user-item cold-start).

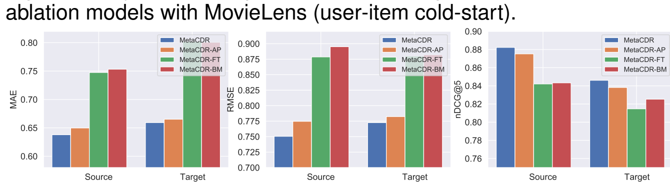

Fig. 12. Analysis of the impact of meta-learning on MetaCDR via various ablation models with Douban (user-item cold-start).

Impact of Meta-Learning: To study the impact of meta-learning, we design three ablation models for comparison with MetaCDR: 1) **A**ll **p**arameters are updated in the relation-wise update step: **MetaCDR-AP**. 2) As in transfer learning, all training data are used for pretraining as in the traditional method, and the base model is **f**ine-**t**uned when new users arrive: **MetaCDR-FT**. 3) Only traditional methods are used to optimize the **b**asic **m**odel, and the information of new users is also considered training data without adaptation: **MetaCDR-BM**.

Fig.11 and Fig.12 show the comparisons among the above ablation models and MetaCDR. With respect to the three metrics in the two domains, MetaCDR always performs best. Moreover, MetaCDR-AP performs slightly worse than MetaCDR because the impact of hierarchical parameter optimization, which allows the model to focus on different information, is noticeable. The effects of MetaCDR-FT and MetaCDR-BM lag significantly behind those of MetaCDR, which shows that meta-learning plays a vital role in the model.

Impact of Transfer Learning: To understand how DKMT impacts this model, similar to the previous analysis, we design three ablation models for comparison with MetaCDR: 1) Two **d**ifferent **c**ross-network parameters H_s and H_t are set for each DKMT: **MetaCDR-DC**. 2) DKMT is replaced with **CSN**; that is, the same weight is assigned to all information: **MetaCDR-CSN**. 3) Two recommendation networks are

TABLE 7
Experimental Results on Amazon for User Cold-Start, with the Best Result Shown in Bold.

Book→Movie	FM	NeuMF	MMoE	SCoNet	MeLU	TMCDR	MetaCDR	MetaCDR-PNN
Hit@10	0.3007	0.3225	0.3058	0.3544	0.4161	0.3390	**0.4242**	0.3834
AUC	0.6589	0.6204	0.6937	0.7215	**0.7420**	0.7078	0.7311	0.7022
MAP@10	0.3839	0.4066	0.4510	0.4583	0.4727	0.4664	**0.5154**	0.4868
Book→Music	FM	NeuMF	MMoE	SCoNet	MeLU	TMCDR	MetaCDR	MetaCDR-PNN
Hit@10	0.4584	0.4388	0.4675	0.4836	0.5379	0.5050	**0.5682**	0.5401
AUC	0.6884	0.6527	0.6716	0.6994	0.7524	0.7233	**0.7691**	0.7299
MAP@10	0.4278	0.4112	0.4397	0.4164	0.5215	0.4734	**0.5358**	0.5079

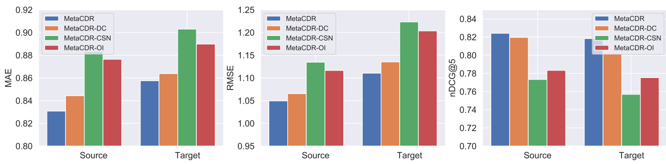

Fig. 13. Analysis of the impact of DKMT on MetaCDR via various ablation models with MovieLens (user-item cold-start).

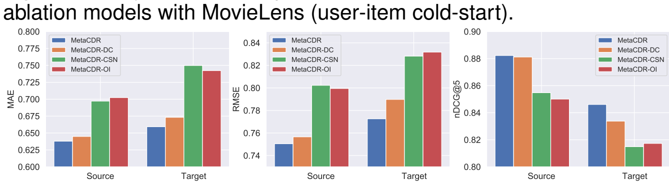

Fig. 14. Analysis of the impact of DKMT on MetaCDR via various ablation models with Douban (user-item cold-start).

optimized **i**ndependently in different domains, and only the user embedding is shared. The two networks are supervised by the labels obtained from the two domains separately and are optimized in an alternating manner: **MetaCDR-OI**.

Fig.13 and Fig.14 show the comparisons between MetaCDR and the above three ablation models. MetaCDR still performs best. The effect of MetaCDR-DC decreases because the introduction of many parameters obviously reduces the effect of the model. The results of MetaCDR-OI are worse than those of the first two models, indicating that the DKMT structure is necessary. It is worth mentioning that the MetaCDR-CSN model yields the worst effect in most scenarios and for most metrics because forcing all features to be given the same weight is not always conducive to the transfer of information.

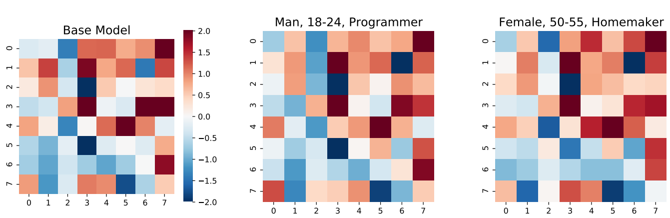

Fig. 15. Visualization of the DKMT parameters obtained from the base model after training, adapted separately for two users: (Male, 18-24, Programmer) and (Female, 50-55, Homemaker).

5.6 Impact of Pretraining (RQ4)

In this section, we demonstrate the effectiveness of the proposed pretraining strategy through experimental comparisons. Table 6 shows the time and space consumption of the four different methods. FOREC [3] is a cross-market recommendation algorithm with pretraining. In fact, FOREC's pretraining strategy is similar to the meta-training in MetaCDR. The difference is that FOREC only trains the meta-model on a single network and adapts it to different markets. Therefore, for fairness, FOREC's network structure is set to be the same as MetaCDR's single-domain network; i.e., it uses the same embedding and MLP. MetaCDR-PT+ is another pretraining MetaCDR that is trained with additional nonoverlapping users during single-domain pretraining. From the results, we find that the time and space efficiency of MetaCDR-PT/PT+ are much higher than those of the other two strategies.

5.7 Model Applicability Study (RQ5)

The Amazon dataset is used to evaluate the performance of MetaCDR in scenarios without side information and with implicit feedback. Unlike previous datasets, Amazon has no user or item features, so we only conduct experiments in the user cold-start scenario. Furthermore, due to the change in the feedback pattern, we introduce new metrics Hit@10, area under the curve (AUC), and mean average precision (MAP)@10 in this section. Furthermore, to study the impact of the network architecture on MetaCDR, we introduce MetaCDR-PNN in this experiment, which uses PNN as the base network.

The experimental results are shown in Table 7. The meta-learning methods still show excellent performance in scenarios without side information and with implicit feedback. MetaCDR achieves the best results on most metrics. Although the performance of MetaCDR-PNN is slightly worse than that of MetaCDR, it is still better than the other methods.

5.8 Visualization (RQ6)

To understand what information is transferred by DKMT and what is being adjusted via the adaptation of meta-learning, we show an 8×8 portion of the second layer of DKMT parameters in three cases in Fig.15. We find that DKMT learns a unique weight for each feature dimension. In Fig.15, the lighter the color, the closer the weight is to 0. A light-colored area indicates that the feature of this dimension plays a small role in cross-domain knowledge transfer, so the model learns a smaller weight for it. The dark blue and dark red areas indicate that the information of these dimensions has an important role, i.e., DKMT can use the differences in user preferences between different domains to capture the evolution of interest.

A comparative analysis of the heatmaps obtained under the above three conditions reveals the role of adaptation in meta-learning in MetaCDR. After updating the model with the information from two different users, the two DKMTs obtain different parameters. That is, in MetaCDR, a personalized model can be generated quickly for each user through meta-learning.

6 Conclusion

We construct a novel recommendation model called MetaCDR based on meta-learning and transfer learning to solve the cold-start problem through cross-domain knowledge and a model optimization strategy.

MetaCDR implements cross-domain knowledge transfer in a meta-learning setting through a DKMT module and updates its parameters hierarchically with meta-learning to learn the complex relationships between the given domains and the appropriate embedding method for user and item features. Moreover, we propose a novel pretraining strategy to make the developed model more applicable. The experimental results prove that the effect of MetaCDR is significantly better than those of the state-of-the-art models in various scenarios.

Acknowledgments

Our work is supported by the National Key Research and Development Program of China [2021YFF1201203 and 2021YFF1201205], the National Natural Science Foundation of China [Grants 61972174 and 62172187], the Science-Technology Development Plan Project of Jilin Province [Grants 20190303006SF, 20190302107GX, 20200801033GH, YDZJ202101ZYTS128], the Science and Technology Planning Project of Guangdong Province No. 2020A0505100018, Guangdong Universities' Innovation Team Project (2021KCXTD015) and Guangdong Key Disciplines Project (2021ZDJS138). Fausto Giunchiglia's work is funded by National Project PRIN 2017 Delphi.

References

[1] H. Bharadhwaj. Meta-learning for user cold-start recommendation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[2] H. Bonab, M. Aliannejadi, A. Vardasbi, E. Kanoulas, and J. Allan. Cross-market product recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 110–119, 2021.

[3] H. Bonab, M. Aliannejadi, A. Vardasbi, E. Kanoulas, and J. Allan. Cross-market product recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 110–119, 2021.

[4] C. Chen, D. Carlson, Z. Gan, C. Li, and L. Carin. Bridging the gap between stochastic gradient mcmc and stochastic optimization. In *Artificial Intelligence and Statistics*, pages 1051–1060. PMLR, 2016.

[5] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[7] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[8] M. F. Dacrema, P. Cremonesi, and D. Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019.

[9] M. Dong, F. Yuan, L. Yao, X. Xu, and L. Zhu. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 688–697, 2020.

[10] Z. Du, X. Wang, H. Yang, J. Zhou, and J. Tang. Sequential scenario-specific meta learner for online recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2895–2904, 2019.

[11] X. Feng, C. Chen, D. Li, M. Zhao, J. Hao, and J. Wang. Cmml: Contextual modulation meta learning for cold-start recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 484–493, 2021.

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[13] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 94–101, 2019.

[14] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2018.

[15] J. Gope and S. K. Jain. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 133–138. IEEE, 2017.

[16] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731, 2017.

[17] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[18] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[19] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648, 2020.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[21] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[22] G. Hu, Y. Zhang, and Q. Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 667–676, 2018.

[23] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[24] S. Kang, J. Hwang, D. Lee, and H. Yu. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1563–1572, 2019.

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

[26] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung. Melu: meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1073–1082, 2019.

[27] C.-T. Li, C.-T. Hsu, and M.-K. Shan. A cross-domain recommendation mechanism for cold-start users based on partial least squares regression. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(6):1–26, 2018.

[28] P. Li and A. Tuzhilin. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 331–339, 2020.
[29] S. Li, L. Yao, S. Mu, W. X. Zhao, Y. Li, T. Guo, B. Ding, and J.-R. Wen. Debiasing learning based cross-domain recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3190–3199, 2021.
[30] Z. Li, F. Zhou, F. Chen, and H. Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.
[31] J. Lian, F. Zhang, X. Xie, and G. Sun. Cccfnet: a content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 817–818, 2017.
[32] X. Lin, J. Wu, C. Zhou, S. Pan, Y. Cao, and B. Wang. Task-adaptive neural process for user cold-start recommendation. In *Proceedings of the Web Conference 2021*, pages 1306–1316, 2021.
[33] J. Liu, P. Zhao, F. Zhuang, Y. Liu, V. S. Sheng, J. Xu, X. Zhou, and H. Xiong. Exploiting aesthetic preference in deep cross networks for cross-domain recommendation. In *Proceedings of The Web Conference 2020*, pages 2768–2774, 2020.
[34] Y. Lu, Y. Fang, and C. Shi. Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1563–1573, 2020.
[35] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, 2018.
[36] T. Man, H. Shen, X. Jin, and X. Cheng. Cross-domain recommendation: an embedding and mapping approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2464–2470, 2017.
[37] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016.
[38] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.
[39] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
[40] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.
[41] K. Rama, P. Kumar, and B. Bhasker. Deep learning to address candidate generation and cold start challenges in recommender systems: A research survey. *arXiv preprint arXiv:1907.08674*, 2019.
[42] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 635–644, 2011.
[43] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2018.
[44] A. K. Sahu and P. Dwivedi. Knowledge transfer by domain-independent user latent factor for cross-domain recommender systems. *Future Generation Computer Systems*, 108:320–333, 2020.
[45] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4080–4090, 2017.
[46] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2440–2448, 2015.
[47] J. Sun, S. Lapuschkin, W. Samek, Y. Zhao, N.-M. Cheung, and A. Binder. Explanation-guided training for cross-domain few-shot classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7609–7616. IEEE, 2021.
[48] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, and M.-H. Yang. Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv:2001.08735*, 2020.
[49] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6907–6917, 2017.
[50] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3637–3645, 2016.
[51] M. Volkovs, G. Yu, and T. Poutanen. Dropoutnet: addressing cold start in recommender systems. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4964–4973, 2017.
[52] R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim. Multimodal model-agnostic meta-learning via task-aware modulation. *arXiv preprint arXiv:1910.13616*, 2019.
[53] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan. A content-based recommender system for computer science publications. *Knowledge-Based Systems*, 157:1–9, 2018.
[54] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426, 2018.
[55] Y. Wang, C. Feng, C. Guo, Y. Chu, and J.-N. Hwang. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 717–725, 2019.
[56] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.
[57] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu. Ckan: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 219–228, 2020.
[58] H. Yao, Y. Wei, J. Huang, and Z. Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, pages 7045–7054. PMLR, 2019.
[59] R. Yu, Y. Gong, X. He, Y. Zhu, Q. Liu, W. Ou, and B. An. Personalized adaptive meta learning for cold-start user preference prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10772–10780, 2021.
[60] W. Yu, X. Lin, J. Ge, W. Ou, and Z. Qin. Semi-supervised collaborative filtering by text-enhanced domain adaptation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2136–2144, 2020.
[61] F. Yuan, L. Yao, and B. Benatallah. Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. In S. Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4227–4233. ijcai.org, 2019.
[62] C. Zhao, C. Li, R. Xiao, H. Deng, and A. Sun. 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*, pages 229–238, 2020.
[63] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. Chi. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51, 2019.
[64] Y. Zheng, S. Liu, Z. Li, and S. Wu. Cold-start sequential recommendation via meta learner. *arXiv preprint arXiv:2012.05462*, 2020.
[65] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, 2018.
[66] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng. Dtcdr: A framework for dual-target cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1533–1542, 2019.
[67] F. Zhu, Y. Wang, C. Chen, G. Liu, and X. Zheng. A graphical and attentional framework for dual-target cross-domain recommendation. In *IJCAI*, pages 3001–3008. ijcai.org, 2020.
[68] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu. Cross-

domain recommendation: Challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696*, 2021.

[69] Y. Zhu, K. Ge, F. Zhuang, R. Xie, D. Xi, X. Zhang, L. Lin, and Q. He. 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*, pages 1813–1817, 2021.

[70] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(4):631–644, 2019.

[71] Y. Zhu, R. Xie, F. Zhuang, K. Ge, Y. Sun, X. Zhang, L. Lin, and J. Cao. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1167–1176, 2021.

Yanchun Liang received a Ph.D. degree in applied mathematics from Jilin University, Changchun, China, in 1997. He is currently a professor at the College of Computer Science and Technology, Jilin University, China. He was a visiting scholar at Manchester University in the U.K. from 1990 to 1991, a visiting professor at the National University of Singapore from 2000 to 2001, a guest professor at the Institute of High Performance Computing of Singapore from 2002 to 2004, a guest professor at Trento University, Italy, from 2006 to 2010, and a guest professor at the University of Missouri, USA, from 2011 to 2017. He has published over 400 papers. His research has been featured in Bioinformatics, IEEE TSMC, IEEE TKDE, the Journal of Micromechanics and Microengineering, Physical Review E, Neural Computing& Applications, Smart Materials and Structures, Artificial Intelligence in Medicine, Applied Artificial Intelligence, etc. He was the recipient of several grants from NSFC, the EU, etc. His research interests include computational intelligence, machine learning methods, text mining, MEMS modeling and bioinformatics.

Renchu Guan (M12) received a Ph.D. degree from Jilin University, Changchun, China, in 2010. He is currently a professor at the College of Computer Science and Technology, Jilin University, China. He has published over 40 papers. His research has been featured in IEEE TKDE, IEEE TGRS, Nature Communications, etc. He was the recipient of several grants from NSFC. His research interests include machine learning, bioinformatics, and knowledge engineering.

Haoyu Pang received a BSc degree in computer science and technology in 2020 from Jilin University, Changchun, China. He is currently working toward a graduate degree at the College of Computer Science and Technology, Jilin University, Changchun, China. He is supervised by Prof. Renchu Guan. His major research interests include machine learning and recommender systems.

Xiaoyue Feng received BSc and Ph.D. degrees in computer science from Jilin University, Changchun, China, in 2003 and 2008. She is currently an associate professor at the College of Computer Science and Technology, Jilin University, China. Her main interests include bioinformatics, machine learning, and text mining.

Fausto Giunchiglia received a Ph.D. degree in computer engineering from the University of Genoa, Faculty of Engineering. He is currently a full professor at the Faculty of Science, University of Trento, Italy. His research interests focus on artificial intelligence, formal methods, knowledge management and agent-oriented software engineering.