M2EU: Meta Learning for Cold-start Recommendation via Enhancing User Preference Estimation

Zhenchao Wu
Gaoling School of Artificial Intelligence,
Renmin University of China
Beijing, China
wuzhenchao@ruc.edu.cn

Xiao Zhou*
Gaoling School of Artificial Intelligence,
Renmin University of China
Beijing, China
xiaozhou@ruc.edu.cn

ABSTRACT

The cold-start problem is commonly encountered in recommender systems when delivering recommendations to users or items with limited interaction information and can seriously harm the performance of the system. To cope with this issue, meta-learning-based approaches have come to the rescue in recent years by enabling models to learn user preferences globally in the pre-training stage followed by local fine-tuning for a target user with only a few interactions. However, we argue that the user representation learned in this way may be inadequate to capture user preference well since solely utilizing his/her own interactions may be far from enough in cold-start scenarios. To tackle this problem, we propose a novel meta-learning method named M2EU to enrich the representations of cold-start users by incorporating the information from other similar users who are identified based on the similarity of both inherent attributes and historical interactions. In addition, we design an attention mechanism according to the variances of ratings in the aggregation of similar user embeddings. To further enhance the capability of user preference modeling, we devise different neural layers to generate user or item embeddings at the rating level and utilize the weight-sharing strategy to guarantee adequate parameters learning of neural layers in our meta-learning approach. In meta-training with mini-batching, we adopt an incremental learning scheme to learn a set of generalized parameters for all tasks. Experimental results on the public benchmark datasets demonstrate that M2EU outperforms state-of-the-art methods through extensive quantitative evaluations in various cold-start scenarios.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Recommender systems; Cold-start problem; Meta learning; User preference estimation

*Corresponding author.

ACM Reference Format:
Zhenchao Wu and Xiao Zhou. 2023. M2EU: Meta Learning for Cold-start Recommendation via Enhancing User Preference Estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591719

1 INTRODUCTION

Recommender systems play a vital role in many real-world applications, including e-commerce platforms, news portals, and online advertising. Most recommender systems recommend candidate items for users according to their interactive history and help them discover what they want in less time. However, these systems are not always useful in addressing the cold-start problem, which arises when there is a lack of user-item interactions. As a result, research on cold-start recommendations is both challenging and crucial.

Previous studies have attempted to address the cold-start problem through data augmentation, which can be accomplished using two strategies. One approach involves randomly dropping out certain user-item interactions or user/item attributes, while still requiring the model to use the limited information to recover the complete contents, including the deleted parts. For example, Volkovs *et al.* [24] thought the cold-start issue means that preference information is missing and applied dropout [20] to input mini-batches and trained a DNN-based latent model that generalizes to missing input. Zheng *et al.* [30] also regarded the cold start as a missing user-item interaction problem and built the Multi-view Denoising Graph Auto-Encoders to randomly drop out some user-item interactions and force the decoder to recover the full views only relying on these limited views. For another, some data augmentation-based methods solve the cold start problem by incorporating auxiliary information into the recommender models, such as content-based recommender systems [16, 21, 26]. In [26], the authors proposed a hybrid model to combine item features learned from the item descriptions into a collaborative filtering model timeSVD++. In [21], a graph auto-encoder framework was proposed based on differentiable message passing on the bipartite interaction graph, where side information is included in the dense hidden layer in the form of user and item feature vectors. Although such approaches enrich user or item representations to some extent, recommender systems tend to recommend the same items to users with similar side information, neglecting fine distinctions.

The core idea of recommender systems is to discern user interests based on inherent user attributes and interaction information, especially user-item interactions. In cold-start scenarios, most recommender systems have difficulties in predicting user preferences due to sparse user-item interactions, and thus often bring some poor recommendations to users. Recently, meta-learning-based methods have been extensively studied to alleviate the cold-start problem with the superiority that the pre-trained model can be fine-tuned to rapidly adapt to new users only with a few interactions. In this line of research, [22] introduced a meta-learning formulation into the recommendation problem and elaborated the rationality of a meta-learning perspective for cold start. Inspired by this, more meta-learning-based methods were proposed to alleviate the cold-start problem, leading to the meta-learning paradigm as one of the most feasible learning frameworks for cold-start recommendations.

Although the meta-learning paradigm has achieved promising results in the cold-start recommendation, there are still some problems to solve or further optimize in such methods. Firstly, user representations usually stand for user comprehensive characteristics and thus are critical for recommender models to estimate user preferences. However, to represent user preferences by solely utilizing new users' own information consisting of inherent attributes and historical interactions is far from enough, especially in cold-start scenarios where user interaction information is lacking. This phenomenon calls for a challenging but crucial task to enrich user representations. Secondly, when user representations are incorporated with user-item interactions, they are unable to reflect user preferences clearly if the items rated differently are addressed with the same neural layers, such as [11]. Some researchers try to set different neural layers at the rating level. However, the ratings on different levels are unequal in quantity, where the neural layers corresponding to the ratings with fewer interactions may be learned inadequately in meta-training. Thirdly, recommender models are usually trained with the mini-batching strategy for fewer memory resources. By this means, the trained models tend to forget the previously learned knowledge and are unable to learn a set of generalized parameters well for all tasks.

In this paper, we present a novel **Me**ta-learning method via **E**nhancing **U**ser preference estimation for the cold-start recommendation, named **M2EU**. To tackle the above challenges, we enrich user representations by incorporating the information of some similar users. To identify these similar users, we select the top K similar users based on the similarity of both inherent attributes and interaction information from the customers who rated a mutually interacted item similarly with the cold-start user. We then design an attention mechanism based on variances of ratings to aggregate the information of the selected top K similar users. To reflect user preferences more clearly, we set different neural layers at the rating level to generate user and item embeddings and introduce a weight-sharing strategy to avoid learning the parameters of embedding layers inadequately. To improve the generalization of the trained model, we adopt an incremental learning scheme to make the model learn the new task batch while simultaneously keeping the memory of the previously learned knowledge in meta-training.

The major contributions in this paper are summarized as follows:

- We incorporate the information of the most similar users to enrich user representation, which is a novel method combing meta-learning and data augmentation for cold-start recommendations.
- We set different neural layers at the rating level to generate user/item embeddings in our meta-learning method. Meanwhile, we introduce a weight-sharing mechanism to avoid learning the parameters of embedding layers inadequately on some levels.
- We adopt an incremental learning scheme to improve the model generalization with the mini-batching strategy in meta-training.
- The extensive experiments on three widely adopted datasets show the proposed M2EU outperforms the state-of-the-art methods on some public benchmarks. The source code is available at https://github.com/zhenchaowu/M2EU.

2 RELATED WORK

2.1 Cold-start Recommendation

The cold-start problem is a common issue in recommender systems that tends to occur when few user-item interactions exist and can harm the recommendation performance. Data augmentation is a traditional solution to deal with this issue by incorporating auxiliary information into the recommender systems. Such methods can enhance user preference estimation by enriching user or item representations, such as using user attributes [16, 19, 28], item attributes [4, 17, 18], or relational data [10, 27, 29]. Besides, other data augmentation-based methods [24, 30, 32] regard cold start as a missing data problem where user content, item content, or user-item interactions are missing, and take the measure that randomly drops out some useful information while forcing the model to recover the previous full contents. Although user and item representations can be enriched with data augmentation, the slight preference differences are ignored among the users with similar side information. In addition, some transfer learning-based methods [6, 7, 9] alleviate the cold-start problem by utilizing the data and knowledge from a source domain to train a model for the target domain. Recently, some methods [2, 8, 11, 13] try to solve the cold-start problem based on the meta-learning paradigm. Here, the global parameters of the model are learned with the existing user data and locally updated to rapidly adapt to the new user preferences with a few interactions. In this sense, meta-learning-based methods are only applicable in the incomplete user cold-start scenario.

2.2 Meta Learning

The goal of meta-learning is to understand how learning itself can be flexible according to the domain or task under study [23]. The meta-learner can work at the both example level and across-task level. The pre-trained model can be fine-tuned by the meta-learner to adapt to a task only with a few examples belonging to this task.

Given the cold-start problem usually occurs in the case of sparse interactions, the pre-trained recommender model can be fine-tuned to rapidly adapt to the cold-start user preference in the same way as meta-learning naturally. For example, Lee *et al.* [8] developed a meta-learning-based recommender system to estimate new user preferences with a few consumed items. Pan *et al.* [13] designed a meta-learning-based model to improve CTR predictions by generating desirable initial embeddings for new items. Lu *et al.* [11] presented a meta-learning approach to address the cold-start problem on heterogeneous information networks using a co-adaptation meta-learner with both semantic- and task-wise adaptations. Dong *et al.* [2] constructed two memory matrices that can store task-specific memories and feature-specific memories to guide the model

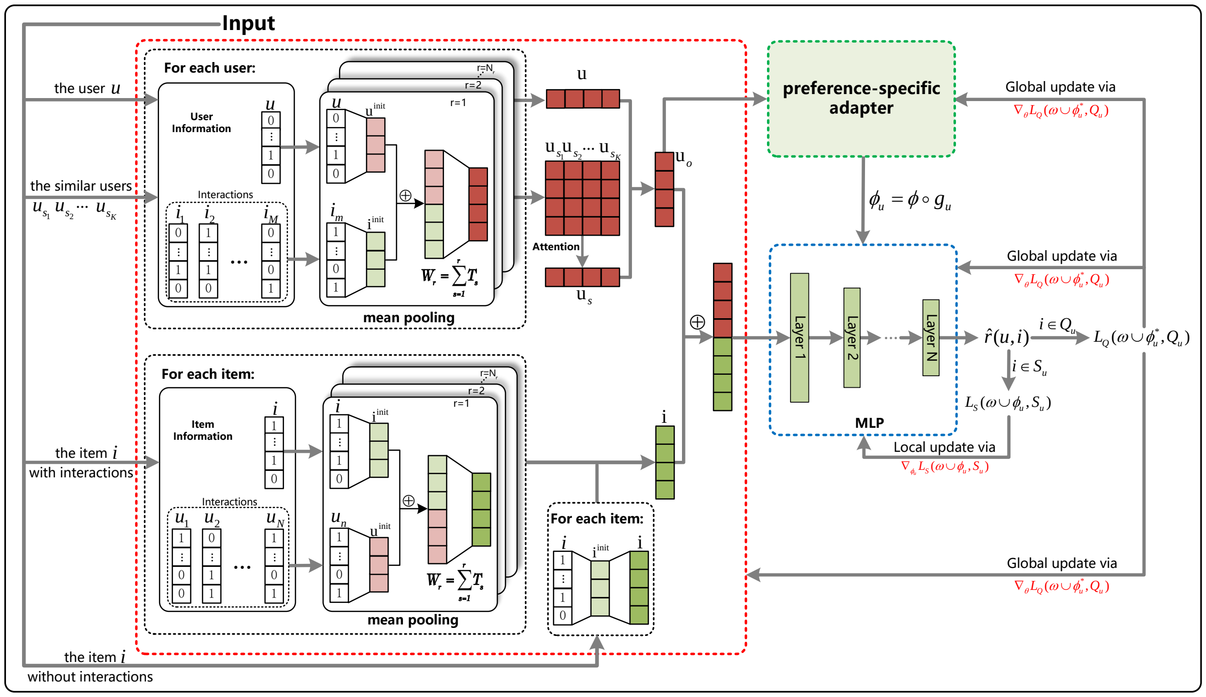

Figure 1: The overall architecture of the proposed M2EU framework. The representation builders (marked with red dotted box) aim to generate the user representation $\mathbf{u}_o$ and item representation $\mathbf{i}$. In the preference-specific adapter (marked with green dotted box), user representation $\mathbf{u}_o$ is used to generate the weights g_u for the prediction module parameters ϕ (marked with blue dotted box) that predicts the rating score $\hat{r}_{u,i}$ for the input user u and item i. The reweighting parameter ϕ_u is locally updated on the support set $\mathcal{S}_u$ while the global parameter θ is updated on the query set Q_u.

to initialize the personalized parameters and predict the user preferences respectively. Wang *et al.* [25] proposed a meta-learning approach that can enhance user preferences with social relations and a preference-specific adapter. However, [2, 8, 13] take no more effective measures to enrich user representations while [11, 25] incorporate user-item interactions into user representations with the same neural layers on different rating levels.

3 PROBLEM FORMULATION

In this paper, we aim to predict the unknown rating score $\hat{r}_{u,i}$ between the user u and item i in cold-start scenarios. To reduce the effect of sparse user-item interactions, we introduce the meta-learning paradigm into the recommender system to better adapt to user preference. In our method, each user can be regarded as a learning task, whose interactions are equivalent to the examples of this task. We separate each task $\mathcal{T}_u$ containing the interacted items into the support set $\mathcal{S}_u$ and query set Q_u that are mutually exclusive. Here we look upon the support set as the item set rated by the user, and the query set as the item set to be predicted in the meta-learning framework. In the phase of meta-training, for each task, $\mathcal{T}_u$ in the meta-training task set $\mathcal{T}^{tr}$, only the prediction module parameters ϕ is fine-tuned to rapidly adapt to the preference of this task *w.r.t.* the loss on the support set $\mathcal{S}_u$. Then the global parameter θ is updated through the backpropagation of the loss on the query set Q_u, which is calculated under the above-updated parameters ϕ. During meta-testing, for each task, $\mathcal{T}_u$ in the meta-test tasks $\mathcal{T}^{te}$, the support set $\mathcal{S}_u$ is still used to update the prediction module parameters ϕ to rapidly adapt to the preference of this test task, but the rating scores of the items in the query set Q_u will be directly predicted using the just locally updated parameters on the support set $\mathcal{S}_u$. Since we hope that M2EU can alleviate the cold-start problem in various scenarios, we consider the following three situations: (1) **UC**: the user cold-start scenario, (2) **IC**: the item cold-start scenario and (3) **UIC**: the user and item cold-start scenario. The details will be illustrated in the section 5.1.1.

4 THE PROPOSED APPROACH

In this section, we introduce the proposed meta-learning approach M2EU for cold-start recommendations.

4.1 Overview

As illustrated in Fig. 1, M2EU consists of three parts: user and item representation builders (red dotted box), preference-specific adapter (green dotted box), and prediction module (blue dotted box).

In representation builders, we make use of user-item interactions to generate user and item representations, where each user representation is enriched by incorporating the aggregation of the top K similar user embeddings. Then we set the weights g_u related to user representation in the preference-specific adapter, aiming to adjust the prediction module parameters ϕ to the preference-specific knowledge ϕ_u in each task. During both meta-training and meta-testing, ϕ_u can be fine-tuned on the support set $\mathcal{S}_u$ to rapidly adapt to user preference. The global parameter θ is updated according to the loss on the query set Q_u during meta-training while the rating scores of the items in the query set Q_u will be directly predicted based on the locally updated parameters ϕ_u^* during meta-testing.

4.2 Identifying Similar Users

In general, the cold-start users have a few user-item interactions, which produce poor user representations where the interactions may be incorporated. To overcome this problem, we enrich user representations with the information of other users similar to the cold-start users. Therefore, similar users are closely related to the quality of user representations and have to be selected cautiously with a very reasonable strategy.

In our method, we think that the user u_i and u_j are similar if they gave the common item the same rating score. Based on this observation, we can seek out many users similar to the cold-start users, but some of them may rate other mutually interacted items differently or have a certain amount of differences in user content from the cold-start users. With this strategy, we may be unable to get real similar users in fact. To identify the high-quality similar users, we further select the top K similar users from the above similar users according to both inherent attributes and interaction information, referring to Eq. (1). Compared with the cold-start user, each of the top K similar users is asked to not only have high similarity in user content but also give similar rating scores to most mutually interacted items. The similarity difference between the user u_i and u_j is formulated as

$$E_{sim}\left(u_i, u_j\right) = \alpha \frac{\mathrm{HD}(a_{u_i}, a_{u_j})}{N} + (1-\alpha)\frac{\sum_{l=1}^{L}\mathrm{ABS}(r_{il} - r_{jl})}{L}, \quad (1)$$

where HD $(\cdot,\cdot)$ and ABS $(\cdot)$ represent the operators of **Hamming Distance** and **Absolute Value**, respectively. a_{u_i} and a_{u_j} are the feature vectors of the user u_i and u_j. Let N represent the number of user features and L represent the number of the items rated by both u_i and u_j. r_{il} and r_{jl} represent the rating scores of the lth mutually interacted item given by u_i and u_j, respectively. α is a manually set constant that controls the influences of the two factors on the similarity between u_i and u_j. Although the two factors have different scales, their value ranges are bounded and approximate. The experimental results in the section 5.2 could verify the feasibility of this strategy in combining the two factors.

4.3 Recommender Model

Here, we will illustrate the recommender model of M2EU, which enriches user representations with the information of the top K similar users at the data level and updates the local parameters to rapidly adapt to user preferences at the model level.

4.3.1 Embedding Builders. Recommender systems can predict the rating scores between users and items from the input user and item representations. In most cases, recommender systems are used to recommend items to the users according to the user's preferences. Thus we propose a novel scheme to enrich user representations by incorporating the information of the top K similar users for the cold-start recommendation. Below we will elaborate on the procedure of enriching user representations in detail.

Firstly, M2EU performs the process of embedding initialization with the strategy of [8]. It is known that a user has N features. The model transforms each feature to the one-hot or multi-hot vector, the embedding of which is generated with the corresponding feature embedding matrix and has a fixed dimension. Afterward, the model concatenates all the feature embeddings to generate the initial user embedding $\mathbf{u}^{init}$, which is shown as

$$\mathbf{u}^{init} = [\mathbf{e}_1 \oplus \mathbf{e}_2 \oplus \cdots \oplus \mathbf{e}_N], \quad (2)$$

where $\oplus$ is the concatenation operation, and $\mathbf{e}_n$ is the nth feature embedding. Similarly, M2EU can generate the initial item embedding $\mathbf{i}^{init}$ for the item i.

Then, M2EU incorporates user-item interactions to generate the user and item embeddings. In the phase of generating each user embedding, our model concatenates the initial user embedding and each initial interacted item embedding respectively, then generates the embedding of user-item interaction with the weight matrix corresponding to the rating between the user and item. Generally speaking, the users should have different preferences on the items rated differently. If all the interacted items are addressed in the same way, the interaction information would be unable to succeed in expressing user preferences. Given a user u and the corresponding support set $\mathcal{S}_u$, the user embedding $\mathbf{u}$ is defined as:

$$\mathbf{u} = \mathrm{MEAN}_{\{i|i\in\mathcal{S}_u\}}\tau(\mathbf{W}_r(\mathbf{u}^{init} \oplus \mathbf{i}^{init})), \quad (3)$$

where $\mathbf{W}_r$ is the weight matrix to generate the embeddings of user-item interactions with the rating r, MEAN$(\cdot)$ is mean pooling and τ is the activation function (we use ReLU). However, for both the users and items, there are different numbers of ratings among the rating levels. Therefore, the parameters of weight matrix $\mathbf{W}_r$ may be learned inadequately if there are only a small amount of user-item interactions with the rating r in the training set. To overcome this problem, we introduce a **weight sharing** mechanism [21] to represent the weight matrix $\mathbf{W}_r$ with the shared parameters on most rating levels. Following [31], the weight matrix $\mathbf{W}_r$ is redefined as $\mathbf{W}_r = \sum_{s=1}^{r} T_s$. Here, the sequence of the shared weight matrices $\{T_1, T_2, \ldots, T_{N_r}\}$ is actually learned in meta-training, in which N_r is the number of all the rating levels. In M2EU, the item embedding $\mathbf{i}$ is generated with the same strategy.

Then the user representations are enriched by incorporating the information of the top K similar users selected with the strategy in the section 4.2. Suppose that the embedding set of the top K similar users is $\{\mathbf{u}_{s_1}, \mathbf{u}_{s_2}, \ldots, \mathbf{u}_{s_K}\}$, then we propose a novel attention mechanism to aggregate these embeddings. Empirically, the rating score represents the user preference level for the item. If a user gave more categories of rating levels to the interacted items, the generated user embedding can reflect the user preference more clearly. The fact is observed and verified that the larger the variance of a series of digits is, the more the categories of these digits are. Inspired by this prior, the attention mechanism in M2EU is based on variances of ratings about each similar user. Here, we denote the set of rating variances about top K similar users by $\{var_{s_1}, var_{s_2}, \ldots, var_{s_K}\}$, then the attention score of the kth similar user embedding is calculated by the following formula:

$$\alpha_k = \frac{\exp(var_{s_k})}{\sum_{i=1}^{K} \exp(var_{s_i})}. \tag{4}$$

Then the aggregation of the top K similar user embeddings is

$$\mathbf{u}_s = \sum_{k=1}^{K} \alpha_k \mathbf{u}_{s_k}. \tag{5}$$

Finally, we incorporate the aggregation $\mathbf{u}_s$ into the user embedding $\mathbf{u}$ to generate final user representation $\mathbf{u}_o$, which is calculated by Eq. (6) or Eq. (7) as follows:

$$\mathbf{u}_o = \mathbf{W}(\lambda_1 \mathbf{u} + \lambda_2 \mathbf{u}_s) + \mathbf{b} \tag{6}$$

$$\mathbf{u}_o = \lambda_1(\mathbf{W}\mathbf{u} + \mathbf{b}) + \lambda_2(\mathbf{W}\mathbf{u}_s + \mathbf{b}), \tag{7}$$

where $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the bias vector, which are the parameters to be learned in meta-training, while λ_1 and λ_2 are the parameters that control the contributions of the user embedding $\mathbf{u}$ and the similar user aggregation $\mathbf{u}_s$ to generate final user representation $\mathbf{u}_o$. Similarly, they are also learnable in meta-training.

The experiments have validated that Eq. (6) and Eq. (7) have different performance on different datasets (see section 5).

4.3.2 Prediction. After obtaining the final user representation $\mathbf{u}_o$, M2EU can predict the rating score $\hat{r}_{u,i}$ between $\mathbf{u}_o$ and an unobserved item $\mathbf{i}$ as follows:

$$\hat{r}_{u,i} = \mathrm{MLP}(\mathbf{u}_o \oplus \mathbf{i}), \tag{8}$$

where MLP is a two-layer multilayer perceptron with the ReLU activation function. Inspired by the preference-specific adapter [25], we set the weights related to user representation for the MLP parameters, which help the model customize the globally shared prior knowledge to preference-specific knowledge. The reweighting scheme on the prediction module is implemented by

$$g_u = \sigma(\mathbf{W}_g \mathbf{u}_o + \mathbf{b}_g), \tag{9}$$

where g_u is the generated weights, which keeps the same shape with the MLP parameters ϕ, $\mathbf{W}_g$, and $\mathbf{b}_g$ are the weight matrix and bias vector, and $\sigma(\cdot)$ is the sigmoid activation function. For the task $\mathcal{T}_u$, the MLP parameters ϕ is adjusted to the preference-specific knowledge ϕ_u via the following formula:

$$\phi_u = \phi \circ g_u, \tag{10}$$

where $\circ$ is the element-wise product operation.

4.3.3 Loss Function. In the meta-learning paradigm, the parameters are locally and globally updated *w.r.t.* the losses on the support set and query set, respectively. Here, we first define the loss function on the support set as

$$\mathcal{L}_S(\omega \cup \phi_u, \mathcal{S}_u) = \frac{1}{|\mathcal{S}_u|} \sum_{i \in \mathcal{S}_u} (r_{u,i} - \hat{r}_{u,i})^2, \tag{11}$$

where ω is all the parameters of the model except the prediction module parameters ϕ, *i.e.*, $\theta = \omega \cup \phi$, and $r_{u,i}$ and $\hat{r}_{u,i}$ are the real and predicted rating scores between the user u and item i.

Different from the loss function on the support set, we introduce two regularizers weighted by the factors γ_1 and γ_2 to the loss function on the query set:

$$\begin{aligned}\mathcal{L}_Q(\omega \cup \phi_u^*, Q_u) = \frac{1}{|Q_u|} \sum_{i \in Q_u} (r_{u,i} - \hat{r}_{u,i})^2 \\ + \gamma_1 \left\| w(\mathbf{u}_o - \mathbf{u}_o^{'}) \right\|_2^2 + \gamma_2 \frac{1}{|\tilde{Q}_u|} \sum_{i \in \tilde{Q}_u} \left\| \mathbf{i} - \mathbf{i}^{'} \right\|_2^2,\end{aligned} \tag{12}$$

where w is set to 1 when the information of the user u is partially dropped out, otherwise set to 0, and $\tilde{Q}_u \subset Q_u$ contains the selected items whose interactions are all dropped out. $\mathbf{u}_o^{'}$ is the generated user representation with randomly dropping out some user-item interactions and similar users, and $\mathbf{i}^{'}$ is the generated item embedding only with the item content. The first regularizer encourages the model to generate the rich user representation with sparse user-item interactions, while the second regularizer aims to solve the generating item embedding problem without any interactions.

4.3.4 Dropout. In meta-training, we randomly select some tasks in each task batch, then drop out some interactions and similar users of the selected tasks. This operation aims to force the model to produce high-quality user representations in spite of the lack of interactions and encourage the pre-trained model to be fine-tuned only with a few interactions to adapt to the cold-start user preferences. In addition, this operation also can avoid over-fitting.

In the meta-learning recommender system, the users have to own some interactions, with which the pre-trained model can be fine-tuned to rapidly adapt to user preferences. However, it is necessary for us to consider how meta-learning models work in the complete item cold-start scenario. To generate the embeddings of the items without any interactions, we design a neural network module to produce accurate item embedding based on the item content. In M2EU, we randomly select some items from each task, then drop out all the interactions of the selected items. The model is forced to utilize item content to generate the item embedding similar to that based on interactions.

4.3.5 Incremental Learning. To reduce the memory requirement in meta-training, we introduce the strategy of mini-batching to train the model, where the task batches are learned one by one. However, the model tends to learn the new task batch while simultaneously forgetting the previously learned knowledge, which is unable to learn a set of generalized parameters. To solve this problem, we adopt the scheme of incremental learning that utilizes a momentum-based dynamic controller η to adjust the contributions of the previous and new knowledge. After the pth task batch is learned, the global parameters θ_p can be obtained by

$$\theta_p = \eta \tilde{\theta}_p + (1 - \eta)\theta_{p-1}, \tag{13}$$

where $\tilde{\theta}_p$ is the global parameters learned just based on pth task batch, and θ_{p-1} is the global parameters before the pth task batch is learned. The dynamic controller η is defined like [14]:

$$\eta = \exp(-\beta \frac{p}{P}), \tag{14}$$

where P is the number of task batches in meta-training, and β is the decay rate. Referring to Eq. (14), we can realize that η keeps declining but the speed of descent is reduced continuously during each epoch of meta-training.

4.4 Meta Optimization

In the meta-learning recommender system, the preference-specific knowledge ϕ_u is fine-tuned to rapidly adapt to user preference with a few user-item interactions on the support set $\mathcal{S}_u$. The personalized knowledge ϕ_u^* is calculated by

$$\phi_u^* = \phi_u - \mu_1 \nabla_{\phi_u} \mathcal{L}_S(\omega \cup \phi_u, \mathcal{S}_u), \tag{15}$$

where μ_1 is the learning rate in the local update.

Then, the model will predict the rating scores of the items in the query set $\mathcal{Q}_u$ based on the updated parameters ϕ_u^*. The global parameters θ can be updated on the query set $\mathcal{Q}_u$ by

$$\theta = \theta - \mu_2 \nabla_\theta \sum_{\mathcal{T}_u \in \mathcal{B}} \mathcal{L}_Q(\omega \cup \phi_u^*, \mathcal{Q}_u), \tag{16}$$

where μ_2 is the learning rate in global update, and $\mathcal{B}$ is the current task batch. The overall process of meta-training is shown in Algorithm 1.

Algorithm 1 : The overall process of meta-training

```
Input: The meta-training tasks $\mathcal{T}^{tr}$,
       The interactions of each item in $\mathcal{T}^{tr}$,
       The top $K$ similar users of each user in $\mathcal{T}^{tr}$,
       The learning rate $\mu_1$ and $\mu_2$.
Output: The learned global parameters $\theta$.
1. Randomly initialize $\theta = \omega \cup \phi$
2. While not converge do
3.     Sample the $p$th batch of tasks $\mathcal{B}_p = \{\mathcal{T}_u | \mathcal{T}_u \in \mathcal{T}^{tr}\}$
4.     Randomly drop out some user information referring to 4.3.4
5.     foreach task $\mathcal{T}_u \in \mathcal{B}_p$ do
6.         Randomly drop out some item information referring to 4.3.4
7.         Compute the final user representation $\mathbf{u}_o$ according to (6) or (7)
8.         Compute the preference-specific knowledge $\phi_u$ according to (10)
9.         Locally update to obtain $\phi_u^*$ according to (15)
10.    Globally update to obtain $\tilde{\theta}_p$ according to (16)
11.    Obtain $\theta_p$ by incremental learning according to (13)
12. return $\theta$
```

5 EXPERIMENTS

In this section, we first introduce the details of the experimental setup, then show and analyze the experimental results in three aspects: (1) the overall performance of M2EU and the state-of-the-art approaches; (2) the ablation study about the different components applied in M2EU; (3) the impacts of parameters on recommendation performance.

5.1 Experimental Setup

5.1.1 Dataset. We conduct experiments on three public benchmark datasets: Douban Book[1], MovieLens[2] and Yelp[3], which are accessible from the websites. Table 1 shows the statistics of the preprocessed datasets.

To obtain the meta-training and meta-testing tasks, we first divide the users and items into two groups (existing/new) with a ratio of 8:2 for each dataset with the strategy from [11]. Then we split all the tasks into four scenarios, *i.e.* **NC** denotes the scenario where there are only existing users and items, **UC** denotes the scenario where there are new users and existing items, **IC** denotes the scenario where there are existing users and new items and **UIC** denotes the scenario where there are only new users and items. The tasks in the first scenario are regarded as meta-training data, and the rest are meta-testing data in cold-start scenarios. Moreover, we randomly extract 10% of meta-training data as the fourth meta-testing task to evaluate the performance of our model in the traditional scenario.

For meta-training and meta-testing data, we select the tasks including between 14 and 100 interacted items on Douban Book and MovieLens, while selecting the tasks containing between 20 and 50 interacted items on Yelp like [25]. For a task with more than 100 or 50 rated items, we randomly select 100 or 50 items as experimental data for this task. In each task, we randomly extract 10 items as the query set $\mathcal{Q}_u$, and the other items are collected as the support set $\mathcal{S}_u$. For the users in meta-training, we seek similar users for them with the interaction information which is only from the support set of meta-training. However, similar users of the meta-testing tasks are selected with the interaction information from both the whole meta-training set and the support set of meta-testing.

[1]https://book.douban.com
[2]https://grouplens.org/datasets/movielens/
[3]https://www.yelp.com/dataset/challenge

5.1.2 Baselines. To evaluate the effectiveness of the proposed model, we compare our method with three kinds of baselines. (1) **Traditional methods**, including FM, NeuMF, Wide&Deep and GC-MC. (2) **Data augmentation based methods**, including DropoutNet and Heater. (3) **Meta-learning methods**, including MeLU, MetaEmb, MAMO, MetaHIN, PAML, and GME. Here we redefine the output units of some baselines as a linear layer for rating prediction.

- **FM** [15] is a feature-based baseline that is able to model all interactions between variables using factorized parameters. Here, we incorporate the user and item content as the input features.
- **NeuMF** [5] is a neural network architecture for collaborative filtering consisting of a generalized matrix factorization component and a MLP component.
- **Wide&Deep** [1] is an advanced learning framework combining wide linear models and deep neural networks, which improves the memorization and generalization of recommender systems.
- **GC-MC** [21] is a graph auto-encoder framework for the matrix completion task in recommender systems, where the user and item embeddings are generated by GCN.
- **DropoutNet** [24] is a neural network-based model for the cold-start problem, which regards cold start as a missing data problem.
- **Heater** [32] is a combined separate training and joint-training framework, which avoids the error superimposition issue and improves the model effectiveness.
- **MeLU** [8] is a meta-learning method, which estimates new users' preferences with a few consumed items. The method alleviates the user cold-start problem with the strategy of MAML.
- **MetaEmb** [13] is a meta-learning method for CTR prediction, which generates initial embeddings for new ad IDs with the user and item content.
- **MAMO** [2] is a meta-learning method, which makes use of two memory matrices to guide the model to initialize the personalized parameters and fast predict user preferences.

Table 1: Statistics of the preprocessed datasets.

	Douban Book	MovieLens-1M	Yelp
Users	10,592	6,040	51,624
Items	21,192	3,881	34,259
Ratings	649,381	1,000,209	1,301,869
Sparsity	99.7107 %	95.7331%	99.9262%
User content	Location	Age, Occupation, Gender, Zip code	Fans, Avg.Rating, Year jointed Yelp
Item content	Publisher, Publication year	Rate, Genre Publication year	Stars, PostalCode
Range of ratings	$1 \sim 5$	$1 \sim 5$	$1 \sim 5$

- **MetaHIN** [11] is a novel meta-learning approach to address cold-start recommendation on HINs, whose meta-learner is equipped with semantic- and task-wise adaptations.
- **PAML** [25] is a meta-learning approach, which designs a novel preference-specific adapter to adapt the globally shared prior knowledge to the preference-specific knowledge.
- **GME** [12] utilizes graph neural networks and meta learning to generate desirable initial embeddings for new ad IDs, which presents three specific GMEs from different perspectives. Here we select the optimal one as the baseline, *i.e.*, GME-A.

5.1.3 Evaluation Metrics. We adopt two widely-employed evaluation metrics like [2] in our experiments. One is mean absolute error (MAE), which measures the differences between the predicted and real rating scores. The other is normalized discounted cumulative gain at rank K (NDCG@K), which evaluates the top K ranking accuracy of the predicted rating scores. Here, we set K as 5.

5.1.4 Environment and Parameter Settings. In this paper, the experiments were carried out on a Linux platform with a 2.2GHz Intel(R) Xeon(R) Silver 4214 CPU and a Tesla T4 GPU. Our method is currently implemented using Pytorch 1.9.0.

In M2EU, the learnable parameters are randomly initialized following the Xavier normal distribution [3]. For the hyper-parameters, we introduce the initialization details here. The parameter α is set as 0.2, 0.5, and 0.3 for Douban Book, MovieLens, and Yelp, respectively. The number of similar users K is set as 10 for Douban Book and MovieLens and 15 for Yelp. Besides, we set the user and item embedding dimensions as 16 on Douban Book and Yelp and 32 on MovieLens. But the batch size of meta-training is all set as 64. The global meta-learning rate is all set as 0.005 for the three datasets, and the local meta-learning rate is set as 0.00005 for Douban Book and 0.0005 for MovieLens and Yelp. The number of local updates is set as 5 and the parameter β is set as 2 for the three datasets. The dropout rates on users and items are set as 0.3 and that on similar users is set as 0.2 for Douban Book. And the corresponding dropout rates are set as 0.4 and 0.3 for MovieLens and 0.7 and 0.3 for Yelp. The parameters γ_1 and γ_2 are set as 0.1 and 0.01 for Douban Book and MovieLens while 0.1 and 0.02 for Yelp.

5.2 Experimental Results

5.2.1 Overall Performance. We compare M2EU with the baselines on the recommendation performance in four scenarios, *i.e.* three cold-start scenarios and the non-cold-start scenario. The quantitative evaluations on the three datasets are shown in Table 2.

From the experimental results, we observe that M2EU consistently yields the best performance among all methods on the three datasets. In the first three scenarios, *i.e.* the cold-start scenarios, M2EU improves over the best baseline *w.r.t.* MAE by 2.93-5.72%, 6.26-7.75% and 0.41-3.34% on Douban Book, MovieLens and Yelp, respectively. Among these baselines, the performance of DropoutNet [24] is least competitive roughly though it is a specialized method for cold-start recommendation. The reason might be that the U and V obtained by matrix decomposition are inaccurate due to the sparsity and large size of the interaction matrix between users and items, and the product of U and V is treated as the ground truth in the training process. Heater [32] yields a good performance similar to the meta-learning methods on Douban Book but a mediocre performance on MovieLens and Yelp. The traditional methods have incorporated the user and item content into recommender models, which achieve better performance than that just relying on the user and item IDs. However, most meta-learning methods can yield better results than these methods in most cases, owning to the superiority of updating the local parameters of the models to rapidly adapt to specific user preferences. To be specific, these meta-learning methods basically achieve good performances on Movielens, but some of them work mediocrely on Douban Book and Yelp. For example, MeLU [8] performs worse than other meta-learning methods [2, 11, 25] *w.r.t.* MAE on Douban Book in each scenario. It may be because fewer user contents are utilized to enrich user representations and the strategy of the local parameter update is simple. In addition, there are undesirable results for MetaEmb [13] on Douban Book in the scenarios (*i.e.* IC and UIC) and on Yelp in the scenarios (*i.e.* UC, IC and UIC), and for GME [12] on Douban Book in the scenarios (*i.e.* UC, UIC, and NC). The reason for this phenomenon may be that over-fitting occurs in the meta-training phases and hinders the generated models from generalizing to some cold-start scenarios. Finally, experimental results in NC have a similar tendency to that in cold-start scenarios on Douban Book and Yelp. But these meta-learning methods yield similar performances with other methods on Yelp in this scenario. This might be because the superiority of meta-learning is inconspicuous on Yelp in the traditional scenario compared with the cold-start scenarios.

Although most meta-learning methods perform better than other kinds of baselines, they still underperform M2EU in all scenarios. MeLU, MetaEmb, and MAMO incorporate the content information into the user and item embeddings, but they take no more measures to enrich user and item representations. MetaHIN captures the HIN-based semantics to enrich user representations, however, user representations are aggregated with the information of items rated differently which are generated with the same weight matrix. GME refines the initial item ID embeddings using neighbor attributes, but user representations are obtained only with the user contents. PAML utilizes social relations to enrich user representations, whose performances are good but inferior to MeLU.

5.2.2 Ablation Study. We conduct some ablation experiments to investigate the effects of two factors in identifying similar users and the contributions of the components applied in the recommender

Table 2: Experimental results of all the methods on the three datasets in four scenarios.

	Model	**UC**		**IC**		**UIC**		**NC**	
		MAE ↓	NDCG@5 ↑	MAE ↓	NDCG@5 ↑	MAE ↓	NDCG@5 ↑	MAE ↓	NDCG@5 ↑
Douban Book	FM	0.6633	0.8980	0.7190	0.8950	0.7345	0.8897	0.6505	0.8940
	NeuMF	0.6384	0.9003	0.6502	0.8970	0.6643	0.8922	0.6349	0.8974
	Wide&Deep	0.6633	0.8937	0.6805	0.8955	0.6830	0.8898	0.6610	0.8898
	GC-MC	0.6595	0.8720	0.6413	0.8764	0.6534	0.8764	0.6505	0.9164
	DropoutNet	0.7715	0.8827	0.7746	0.8879	0.7738	0.8848	0.7564	0.8696
	Heater	0.6296	0.8809	0.6397	0.8886	0.6445	0.8848	0.6397	0.8712
	MeLU	0.6741	0.8930	0.6838	0.8901	0.6900	0.8830	0.6754	0.8869
	MetaEmb	0.6296	0.9162	0.7028	0.9006	0.7380	0.8896	0.6204	0.9202
	MAMO	0.6173	0.8834	0.6182	0.8965	0.6141	0.8955	0.6445	0.8722
	MetaHIN	0.6034	0.8944	0.6163	0.8949	0.6156	0.8913	0.6374	0.8855
	PAML	0.6018	0.8958	0.6135	0.8951	0.6170	0.8973	0.6229	0.9082
	GME	0.6736	0.8984	0.6135	0.9209	0.6609	0.9086	0.6905	0.8832
	M2EU	**0.5758**	**0.9233**	**0.5784**	**0.9271**	**0.5961**	**0.9191**	**0.5824**	**0.9221**
MovieLens-1M	FM	0.8668	0.8646	0.9110	0.8271	0.9424	0.8203	0.8565	0.8689
	NeuMF	0.8826	0.8628	0.8919	0.8418	0.9370	0.8401	0.8679	0.8611
	Wide&Deep	0.9115	0.8582	0.9745	0.8254	0.9679	0.8250	0.9187	0.8514
	GC-MC	0.8824	0.8929	0.9691	0.8354	1.0112	0.8313	0.8756	0.8941
	DropoutNet	0.9655	0.8477	1.0008	0.8233	1.0021	0.8240	0.9628	0.8471
	Heater	0.8880	0.8464	0.9583	0.7979	0.9496	0.8028	0.8969	0.8459
	MeLU	0.8160	0.8387	0.8959	0.8324	0.8833	0.8274	0.8087	0.8408
	MetaEmb	0.8010	0.8781	0.8934	0.8533	0.8926	0.8559	0.7846	0.8838
	MAMO	0.8002	0.8479	0.8899	0.8291	0.8735	0.8225	0.7970	0.8470
	MetaHIN	0.7824	0.8657	0.8660	0.8432	0.8516	0.8423	0.7814	0.8651
	PAML	0.8046	0.8632	0.8764	0.8794	0.8665	0.8766	0.8032	0.8637
	GME	0.8414	0.8681	0.8769	0.8798	0.8659	0.8723	0.8682	0.8564
	M2EU	**0.7334**	**0.8940**	**0.7989**	**0.8844**	**0.7916**	**0.8830**	**0.7284**	**0.8953**
Yelp	FM	0.9317	0.8485	0.9377	0.8375	0.9557	0.8430	0.7682	0.8536
	NeuMF	0.9315	0.8479	0.8532	0.8414	0.8710	0.8472	0.7644	0.8550
	Wide&Deep	0.9680	0.8445	0.9014	0.8342	0.9106	0.8419	0.7824	0.8511
	GC-MC	0.9288	0.8612	0.8285	0.8325	0.8615	0.8278	0.7603	0.8771
	DropoutNet	0.9418	0.8284	0.8390	0.8432	0.8444	0.8314	0.8084	0.8370
	Heater	0.9383	0.8293	0.8168	0.8495	0.8276	0.8360	0.7875	0.8311
	MeLU	0.8907	0.8484	0.7857	0.8386	0.7967	0.8443	0.7750	0.8468
	MetaEmb	0.9284	0.8571	0.9101	0.8379	0.9205	0.8388	0.7604	0.8682
	MAMO	0.8662	0.8204	0.7581	0.8384	0.7667	0.8367	0.7520	0.8207
	MetaHIN	0.8572	0.8244	0.7483	0.8409	0.7583	0.8348	0.7445	0.8381
	PAML	0.8943	0.8225	0.7989	0.8289	0.8076	0.8312	0.7694	0.8814
	GME	0.9011	0.8463	0.7730	0.8553	0.7928	0.8465	0.7581	0.8451
	M2EU	**0.8286**	**0.8711**	**0.7452**	**0.8610**	**0.7468**	**0.8654**	**0.7074**	**0.8828**

model. Similar to the work [25], we only evaluate the impacts of these components on experimental results in the scenario UIC.

In Fig. 2, we can observe that results are optimal when both inherent attributes and interaction information are taken into account to identify similar users. Due to the sparsity of user-item interactions in cold-start scenarios, inherent attributes play more important roles in identifying similar users than interaction information. Besides, we explain some concepts defined in Table 3: "S" denotes the top K similar users, "A" denotes the proposed attention mechanism, "RW" denotes the different weight matrices at the rating level, "WS" denotes the strategy of weight sharing, "P" denotes the preference-specific adapter, "D" denotes the dropout action, "I" denotes incremental learning and "-" denotes removing the following component. Obviously, all the components in the model play positive roles in cold-start scenarios.

Firstly, we evaluate the impact of incorporating the information of the top K similar users into user representations. When the top K similar users are removed, M2EU-S yields the worse performance on these datasets. Besides, the information of the top K similar users might be unable to make positive effects on experimental results if aggregated by mean pooling, *i.e.*, M2EU-A. Fortunately, we present a novel attention mechanism to aggregate the information of the top K similar users, which avoids this problem effectively.

Table 3: Ablation study in UIC.

Model	Douban Book		MovieLens		Yelp	
	MAE	NDCG@5	MAE	NDCG@5	MAE	NDCG@5
M2EU	**0.5961**	**0.9191**	**0.7916**	**0.8830**	**0.7468**	**0.8654**
M2EU-S	0.6083	0.9155	0.8163	0.8781	0.7790	0.8511
M2EU-A	0.6186	0.9114	0.7969	0.8816	0.7705	0.8605
M2EU-RW	0.6336	0.8846	0.8937	0.8085	0.8443	0.8392
M2EU-WS	0.6033	0.9166	0.7954	0.8793	0.7634	0.8610
M2EU-P	0.6093	0.9065	0.8017	0.8798	0.7554	0.8564
M2EU-D	0.6296	0.8946	0.8124	0.8798	0.7607	0.8606
M2EU-I	0.6340	0.8772	0.8089	0.8748	0.7778	0.8491

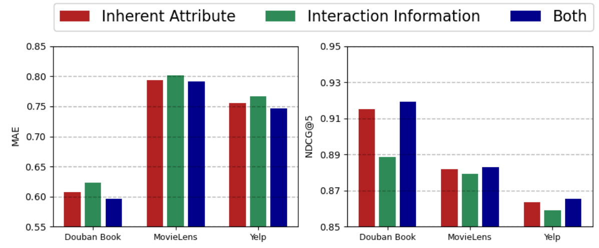

Figure 2: Impact of two factors in identifying similar users.

Then, we study the efficiency of the different weight matrices corresponding to the different ratings. M2EU-RW addresses the items rated differently with the same weight matrix, which underperforms M2EU evidently. In addition, we compare M2EU with the model M2EU-WS, whose results verify the validity of the weight-sharing strategy. In a sense, the less meta-training data is, the more important role the weight-sharing strategy will play.

At last, we explore the contributions of the other three components adopted in M2EU. M2EU-P locally updates the parameters ϕ instead of ϕ_u, which implies that M2EU-P is unable to further improve the prediction performance with user representations. M2EU-D trains the tasks without dropout action in user-item interactions and similar users, which not only hardly generalizes to the cold-start scenarios, but easily causes over-fitting. M2EU-I cuts off the relation of the parameters learned from the new and previous task batches, which tends to make the trained model only suitable for the new tasks. The experimental results in Table 3 indicate that the three components are all indispensable parts of our method.

5.2.3 Parameter Analysis. We also investigate the impacts of key parameters by analyzing how they affect the experimental results on the three datasets.

Our model M2EU enriches user representations by incorporating the information of the top K similar users, where the K value has an important influence on the performance of the recommender system. Therefore, we plot the experimental results of M2EU $w.r.t.$ MAE in four scenarios as K is set at 5 intervals in Fig. 3 (a). With the K value continuing to grow, the value has an ascent trend after an initial decline. We observe that M2EU achieves optimal performance when the K value is set to 10 on Douban Book and MovieLens while 15 on Yelp, which indicates incorporating a limited number of similar users leads to optimal performances in our model.

We also study the impact of the number of local updates on the performance of the recommender system. Fig. 3 (b) shows the

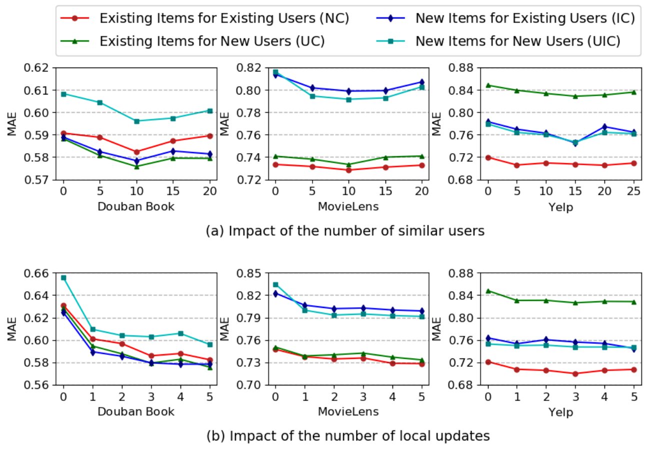

Figure 3: Parameter analysis in different settings.

tendency of MAE obtained from M2EU with the number of local updates varying from 0 to 5. It is evident that the MAE in M2EU without fine-tuning is larger than that in the model undergoing some local updates, which proves the meta-learning strategy is effective for the cold-start problem. The MAE in four scenarios gradually decreases slowly after one gradient update. Here, we set the maximum value as 5 to ensure that M2EU reaches the optimal performance. According to the tendency of MAE in four scenarios, there may not be a significant improvement in performance but complex calculations when the number of local updates is over 5.

6 CONCLUSION

In this paper, we propose a novel meta-learning approach called M2EU to alleviate the cold-start problem. In M2EU, user representations are enriched by incorporating the information of the top K similar users. Besides, we design an attention mechanism to aggregate these similar user embeddings, which promotes user representations to reflect user preferences more clearly. For the user and item embeddings, we set different neural layers to generate them at the rating level. To avoid learning the parameters inadequately on some rating levels, we introduce the weight-sharing strategy to our meta-learning model, which makes most shared weight matrix parameters learned through each user-item interaction. We also utilize a preference-specific adapter for prediction and train the task batches with the incremental learning strategy to further improve the recommendation performance in cold-start scenarios. Experimental results on three public datasets show M2EU achieves better results compared with state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 62106274 and the Fundamental Research Funds for the Central Universities, Renmin University of China under Grant No. 22XNKJ24. We also wish to acknowledge the support provided by China Unicom Innovation Ecological Co-operation Plan and Intelligent Social Governance Platform, Major Innovation Planning Interdisciplinary Platform for the "Double-First Class" Initiative. Xiao Zhou is the corresponding author.

REFERENCES

[1] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and et al. 2016. Wide & deep learning for recommender systems. *In Proceedings of the 1st workshop on deep learning for recommender systems* (2016), 7–10.

[2] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation. *In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2020).

[3] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. *In Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), 249–256.

[4] Q Gu, Z Jie, and CHQ Ding. 2010. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. *In Proceedings of the 2010 SIAM international conference on data mining* (2010), 199–210.

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. *In Proceedings of the 26th international conference on world wide web* (2017), 173–182.

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

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

[8] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. *In The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2019).

[9] Cheng-Te Li, Chia-Tai Hsu, and Man-Kwan Shan. 2018. A Cross-Domain Recommendation Mechanism for Cold-Start Users Based on Partial Least Squares Regression. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2018), 1–26.

[10] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2013. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. *In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (2013), 283–292.

[11] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation. *In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2020).

[12] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Li Li, Kun Zhang, Jinmei Luo, Zhaojie Liu, and Yanlong Du. 2021. Learning Graph Meta Embeddings for Cold-Start Ads in Click-Through Rate Prediction. *In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).

[13] Feiyang Pan, Shuokai Li, and Xiang Ao. 2019. Warm Up Cold-start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. *In SIGIR'19: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

[14] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. 2020. iTAML : An Incremental Task-Agnostic Meta-learning Approach. *In CVPR* (2020).

[15] Steffen Rendle. 2010. Factorization Machines. *In ICDM* (2010).

[16] Sujoy Roy and Sharath Chandra Guntuku. 2016. Latent factor representations for cold-start video recommendation. *In Proceedings of the 10th ACM conference on recommender systems* (2016), 99–106.

[17] M. Saveski and A. Mantrach. 2014. Item cold-start recommendations: learning local collective embeddings. *In Proceedings of the 8th ACM Conference on Recommender systems* (2014), 89–96.

[18] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. *In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (2002), 253–260.

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

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* (2014).

[21] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[22] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A Meta-Learning Perspective on Cold-Start Recommendations for Items. *31st Conference on Neural Information Processing Systems* (2017), 6904–6914.

[23] Ricardo Vilalta and Youssef Drissi. 2002. A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review* (2002), 77–95.

[24] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. *In Advances in neural information processing systems* (2017), 4957–4966.

[25] Li Wang, Binbin Jin, Zhenya Huang, Hongke Zhao, Defu Lian, Qi Liu, and Enhong Chen. 2021. Preference-Adaptive Meta-Learning for Cold-Start Recommendation. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence* (2021).

[26] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2016. Collaborative filtering and deep learning based hybrid recommendation for cold start problem. *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)* (2016), 974–877.

[27] Yuan Yao, Hanghang Ton, Guo Yan, Feng Xu, Xiang Zhang, Boleslaw K. Szymanski, and Jian Lu. 2014. Dual-regularized oneclass collaborative filteringn. *In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management* (2014), 759–768.

[28] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Yue. 2014. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. *In Proceedings of the 37th international ACM SIGIR conference on Research development in information retrieval* (2014), 73–82.

[29] Wayne Xin Zhao, Sui Li, Yulan He, Edward Y. Chang, Ji-Rong Wen, and Xiaoming Li. 2016. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering* 28, 5 (2016), 1147–1159.

[30] Jiawei Zheng, Qianli Ma, Hao Gu, and Zhenjing Zheng. 2021. Multi-view Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-start Recommendation. *In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2021), 2338–2348.

[31] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. *In Proceedings of the 33nd International Conference on Machine Learning* (2016), 764–773.

[32] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for New Users and New Items via Randomized Training and Mixture-of-Experts Transformation. *In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).