Preference-Adaptive Meta-Learning for Cold-Start Recommendation

Li Wang[1], **Binbin Jin**[1], **Zhenya Huang**[1], **Hongke Zhao**[2], **Defu Lian**[1], **Qi Liu**[1] and **Enhong Chen**[1*]

[1] Anhui Province Key Lab. of Big Data Analysis and Application, School of Data Science & School of Computer Science and Technology, University of Science and Technology of China

[2] College of Management and Economics, Tianjin University

{wl063, bb0725}@mail.ustc.edu.cn, {huangzhy, liandefu, qiliuql, cheneh}@ustc.edu.cn, {hongke}@tju.edu.cn

Abstract

In recommender systems, the cold-start problem is a critical issue. To alleviate this problem, an emerging direction adopts meta-learning frameworks and achieves success. Most existing works aim to learn globally shared prior knowledge across all users so that it can be quickly adapted to a new user with sparse interactions. However, globally shared prior knowledge may be inadequate to discern users' complicated behaviors and causes poor generalization. Therefore, we argue that prior knowledge should be locally shared by users with similar preferences who can be recognized by social relations. To this end, in this paper, we propose a Preference-Adaptive Meta-Learning approach (PAML) to improve existing meta-learning frameworks with better generalization capacity. Specifically, to address two challenges imposed by social relations, we first identify reliable implicit friends to strengthen a user's social relations based on our defined palindrome paths. Then, a coarse-fine preference modeling method is proposed to leverage social relations and capture the preference. Afterwards, a novel preference-specific adapter is designed to adapt the globally shared prior knowledge to the preference-specific knowledge so that users who have similar tastes share similar knowledge. We conduct extensive experiments on two publicly available datasets. Experimental results validate the power of social relations and the effectiveness of PAML.

1 Introduction

Benefiting from filtering out personalized irrelevant information for users, recommender systems can effectively remedy the information overload problem and are widely used in various kinds of web services [Li *et al.*, 2018; Huang *et al.*, 2019]. In most recommender systems, collaborative filtering (CF) is the mainstream method which makes predictions based on user-item interactions (e.g. ratings). However, when encountering new users, CF based approaches fail due to scarce interactions, leading to a decline in the new users' experience.

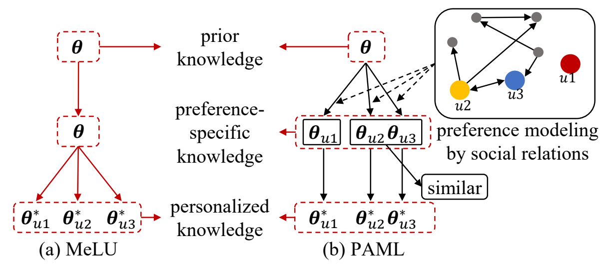

Figure 1: Comparison between MeLU and our proposed model.

To address the cold-start problem, an emerging direction adopts the meta-learning frameworks and some pioneering works prove its effectiveness [Vartak *et al.*, 2017]. Most existing works (e.g. MeLU [Lee *et al.*, 2019]) formulate each user as a task and aim to learn globally shared prior knowledge across all users. As shown in Fig. 1 (a), for a cold-start user, the learned prior knowledge can be quickly adapted to the personalized knowledge (i.e. initialization of parameters) based on her sparse interactions. However, globally shared prior knowledge may be inadequate to discern users' complicated behaviors and causes poor generalization [Dong *et al.*, 2020]. In this paper, instead of global sharing knowledge, we argue that users with similar preferences should locally share similar prior knowledge (named preference-specific knowledge) so that it can be easily generalized to these users. Thus, the key issue is how to recognize users with similar preferences and then generate the prior knowledge for them.

In recent years, with the prevalence of social platforms, users prefer to bond with each other and form the social network. Based on the homophily effect theory [Aral *et al.*, 2009], which states people tend to associate and bond with others that have similar preferences, it is natural to realize that the connected users in the social network are highly similar. Therefore, social relations can provide a guidance to recognize a bundle of users who have similar preferences and share similar knowledge as shown Fig. 1 (b).

Unfortunately, leveraging social relations imposes two challenges. (1) How to explore and strengthen social relations between users? In cold-start scenarios, social relations are almost as sparse as the user feedback. According to our statistics, most users only have less than 6 connected friends, who

*Corresponding Author

are referred to as *explicit friends*. In reality, except for these explicit friends, users also share similar tastes with other unknown ones and we refer to such users as *implicit friends*. However, it is difficult to identify credible implicit friends for a user from a large number of noisy ones. (2) How to recognize closely correlated users from a user's social relations? Although two connected users have similar preferences, the strength of the link between them varies at different rating levels. For example, two users both dislike action movies, leading to a strong tie at a low rating level. On the contrary, they favor different types of movies so that the strength is weak at a high rating level. Ignoring this may result in bad similarity discovery between a user and her friends, which arouses inaccurate user preferences. Therefore, how to wisely utilize social relations is another challenge.

Motivated by the above problems, in this paper, we propose a Preference-Adaptive Meta-Learning approach (PAML) for improving existing meta-learning frameworks with better generalization capacity. Particularly, we focus on recognizing users who have similar preferences and share similar knowledge (i.e. preference-specific knowledge in Fig. 1 (b)) by utilizing social relations. Specifically, to address the challenges imposed by social relations, we first define palindrome paths over the user-item-attribute graph and propose a measurement to identify reliable implicit friends, who have expressed same tastes on items or item attributes. Then, we propose a coarse-fine preference modeling method to accurately capture a user's preference which can also reflect relations with others. After that, a novel preference-specific adapter is designed to adapt the globally shared prior knowledge to the preference-specific knowledge so that users who have similar preferences can share similar knowledge. Under the preference-specific knowledge, optimal personalized knowledge can be learned and utilized to make predictions. We conduct extensive experiments on two publicly available datasets. Experimental results clearly demonstrate the power of social relations and validate the effectiveness of PAML.

2 Related Work

In our study, the related work involves two categories: meta-learning for recommendation and social recommendation.

2.1 Meta-learning for Recommendation

Meta-learning, enabling models to quickly learn a new task with scarce labeled data by utilizing prior knowledge learned from previous tasks, has been applied to solve the data sparsity problem in various fields, such as computer vision [Zhu *et al.*, 2020] and natural language processing [Mi *et al.*, 2019].

Since the cold-start problem is a typical data sparsity problem, meta-learning has been adopted to deal with it and achieved desired results [Vartak *et al.*, 2017]. For example, MeLU [Lee *et al.*, 2019] aimed to learn the initial weights of the neural networks for cold-start users based on MAML [Finn *et al.*, 2017]. Pan et al. [2019] proposed Meta-Embedding which was a content-based embedding generator for learning embeddings for new IDs. The above works learn globally shared parameters which are the same for all tasks. This setting may suffer from handling a sequence of tasks originated from different distributions [Yao *et al.*, 2019]. MAMO [Dong *et al.*, 2020] was the first attempt to get specific parameters by user profiles. MetaHIN [Lu *et al.*, 2020] took the advantage of HIN and proposed semantic-specific parameters. In contrast to existing works, we claim users with similar preferences should share similar prior knowledge and leverage social relations to achieve this goal. To the best of our knowledge, this is the first attempt.

2.2 Social Recommendation

Social relations are effective in advancing recommendation performance [Wang *et al.*, 2019]. Based on the homophily effect recognized by social scientists which declares that users' preferences are similar to their social neighbors [Aral *et al.*, 2009], previous studies for social recommendation attempt to model social effects in various ways. Ma et al. [2011] and Wang et al. [2013] formulated the social relations as a regularization term. Zhao et al. [2014] proposed a bayesian personalized framework together with social information. With the development of graph neural networks and graph embedding methods [Pei *et al.*, 2020], more complicated information hidden in the social networks can be captured. Along this line, GraphRec [Fan *et al.*, 2019] and DiffNet [Wu *et al.*, 2019] were two representative works.

In contrast to most existing works extracting explicit social relations, some researchers pay attention to identify reliable friends from unobserved social networks. Yu et al. [2018; 2019] proposed two approaches to recognize implicit friends. One was implemented on the HIN where users in the same meta path might have a strong connection. The other turned to the GAN [Goodfellow *et al.*, 2014; Jin *et al.*, 2020]. In cold-start scenarios, explicit friends are limited. Therefore, it is necessary to strengthen social relations by identifying implicit friends. Different from [Yu *et al.*, 2018] which mainly identifies implicit friends with the help of social networks, we adopt the user-item-attribute graph since the social network is sparse and propose a measurement to evaluate the similarity between users based on our defined palindrome paths.

3 Preliminaries

In this section, we first give an overview of our problem and then illustrate the HIN and palindrome paths.

3.1 Problem Overview

In this paper, we suppose U is the user set, I is the item set, R is the rating set. For a user $u \in U$ with profiles and a set of explicit friends $\{f_u|f_u \in U\}$, given a set of ratings on the items $\{r_{u,i}|u \in U, i \in I, r_{u,i} \in R\}$ where each item has several attributes, we aim to predict the unknown rating $\hat{r}_{u,i}$ between the user u and the item i.

In this paper, we formalize each user with historical interactions as a learning task which is similar to [Lee *et al.*, 2019], however with one important difference: we introduce users' social relations. More formally, a user *u* can be defined with $\mathcal{T}_u = (\mathcal{F}_u, \mathcal{S}_u, \mathcal{Q}_u)$, where $\mathcal{F}_u$ is the friend set of u, $\mathcal{S}_u$ is the support set containing the interacted items, and $\mathcal{Q}_u$ is the query set containing the items to be predicted. All tasks are split into meta-training tasks $\mathcal{T}^{tr}$ and meta-testing tasks

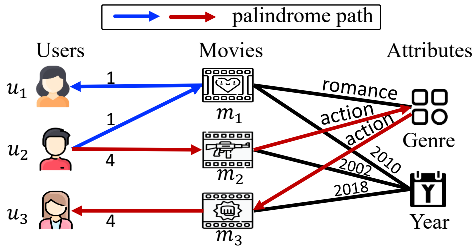

Figure 2: A toy example of the HIN and palindrome paths.

$\mathcal{T}^{te}$. Generally, $\mathcal{T}^{tr}$ are used to train the model while $\mathcal{T}^{te}$ are used to validate its performance. For each task in $\mathcal{T}^{tr}$ and $\mathcal{T}^{te}$, its friend set and support set are used to adapt the prior knowledge to preference-specific knowledge and personalized knowledge. Under the personalized knowledge, each item in the query set can be predicted. In addition, the query set in $\mathcal{T}^{tr}$ also plays a role in updating the prior knowledge.

3.2 User-Item-Attribute Graph

Since we identify implicit friends over the user-item-attribute graph which is a kind of heterogeneous information networks (HINs), we give definitions of the HIN and palindrome paths. Note that implicit friends are identified from a group of known users so that the HIN is constructed from all support sets in the meta-training tasks.

Definition 1 (Heterogeneous information network)**.** *A HIN is denoted as $\boldsymbol{\mathcal{G}} = (V, E, T)$, where each node $v \in V$ and each link $e \in E$ is associated with a mapping function $\phi(v) : V \to T_V$ and $\phi(e) : E \to T_E$, respectively. T_V and T_E denote the sets of node and relation types where $T = T_V \cup T_E$ and $|T_V| + |T_E| > 2$.*

Definition 2 (Palindrome path)**.** *Given the HIN $\boldsymbol{\mathcal{G}}$, a palindrome path of length l is defined as $p_l = v_0 \overset{e_0}{\to} \cdots \overset{e_{l-1}}{\to} v_l \overset{e_{l-1}}{\to} \cdots \overset{e_0}{\to} v_{2l}$, where $v_i \in V, \phi(v_i) = \phi(v_{2l-i})$ and $e_i \in E$.*

Fig. 2 shows a toy example of the HIN. There are three types of nodes (i.e., users, movies and attributes) as well as three types of relations (i.e., ratings, genres, years). "$u_2 \overset{1}{\to} m_1 \overset{1}{\to} u_1$" is a palindrome path of length 1 where u_1 and u_2 are both users who rate 1 score on the movie m_1.

4 Preference-adaptive Meta-learning

In this section, we will introduce the technical details of our proposed PAML, i.e., the approach to identifying implicit friends, social enhanced recommender with the coarse-fine preference modeling method and the entire meta-learning framework.

4.1 Identifying Implicit Friends over the HIN

In cold-start scenarios, most users have few explicit friends which restricts the power of social relations. Identifying implicit friends can strengthen the social relations. To achieve this goal, we design two palindrome paths over the heterogeneous information network $\boldsymbol{\mathcal{G}}$ we defined and assume users appeared in the same path share similar tastes since they express the same opinion on the item or the item attribute: (1) $p_1 = u \overset{r_{u,i}}{\to} i \overset{r_{u,i}}{\to} u$ where $u \in U, i \in I, r_{u,i} \in R$ (e.g. the blue path in Fig. 2 shows u_1 and u_2 both dislike the movie m_1 and give 1 score); (2) $p_2 = u \overset{r_{u,i}}{\to} i \overset{r_{i,a}}{\to} a \overset{r_{i,a}}{\to} i \overset{r_{u,i}}{\to} u$ where $r_{i,a}$ is a certain value of the attribute a (e.g. the red path in Fig. 2 shows u_2 and u_3 might have similar tastes since their favorite movies m_2 and m_3 both belong to action movies). Here, we only consider palindrome paths with a maximum length of 2 because a larger one entails more complicated semantics and does not contribute to finding implicit friends.

Then, we propose a measurement to evaluate the similarity between users based on palindrome paths. Suppose u_1 is connected to u_2 through $p_l = v_0(u_1) \overset{e_0}{\to} \cdots \overset{e_{2l-1}}{\to} v_{2l}(u_2)$, their similarity over p_l is formulated as follows:

$$Sim(u_1, u_2, p_l) = \prod_{i=0}^{2l-1} P(v_{i+1}|v_i, e_i), \tag{1}$$

$$P(v_{i+1}|v_i, e_i) = \frac{1}{N(v_i, e_i, v_{i+1})}, \tag{2}$$

where $N(v_i, e_i, v_{i+1})$ denotes the number of neighborhoods of the node v_i with the same relation value e_i and node type of v_{i+1}. Given a user u and her support set $\mathcal{S}_u$, we collect all paths denoted as $\mathcal{P}$ starting from $u \overset{r_{u,i}}{\longrightarrow} i, i \in \mathcal{S}_u$ over $\boldsymbol{\mathcal{G}}$ and calculate how similar she is to the others as $Sim(u, u') = \sum_{p \in \mathcal{P}} \mathbb{I}(u', p) Sim(u, u', p)$, where $\mathbb{I}(u', p)$ is an indicator function which equals 1 when u' locates at the end of p and equals 0 otherwise. Finally, we choose the top similar users as u's implicit friends to strengthen the social relations which can help to recognize the relations between users. Now, the friend set $\mathcal{F}_u$ of the user u consists of implicit and explicit friends which are denoted as $\mathcal{F}_u^i$ and $\mathcal{F}_u^e$, respectively.

4.2 Social Enhanced Recommender

Here, we will illustrate the social enhanced recommender including the coarse-fine preference modeling method.

Coarse-fine Preference Modeling

We aim to integrate a user's interactions and her friends to capture her preference which can also reflect the relations with others. Considering the strength of links between users varies at different rating levels, we propose the coarse-fine preference modeling method. Specifically, for the fine level, we distinguish the strength of social relations and combine them together at each rating score. For the coarse level, we learn an overall preference by integrating preferences obtained by the fine level.

First, we initialize user and item embeddings based on their features. Suppose there are N kinds of features for a user u, we define her embedding as follows:

$$\boldsymbol{u}^{ini} = [\boldsymbol{e}_1 \oplus \boldsymbol{e}_2 \oplus \cdots \oplus \boldsymbol{e}_N]^\top, \tag{3}$$

where $\oplus$ is the concatenation operation, and $\boldsymbol{e}_n$ is the n-th feature embedding extracted from the embedding matrix. For an item i, its embedding $\boldsymbol{i}^{ini}$ can be defined in the same way.

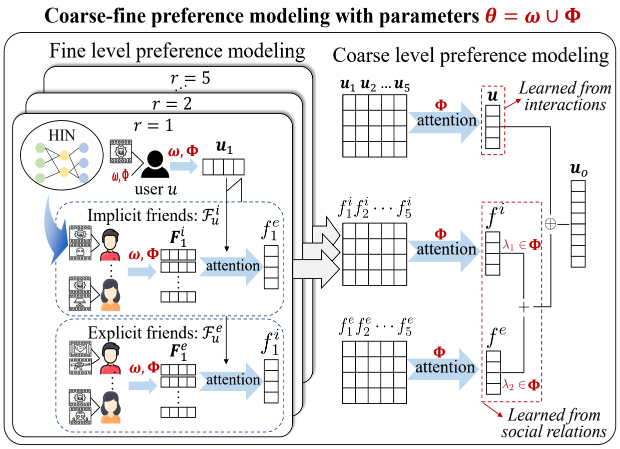

Figure 3: Graphical structure of coarse-fine preference modeling.

Then, we introduce the ***fine level preference modeling*** as shown in the left part of Fig. 3. Given a user u with her friend set $\mathcal{F}_u^i \cup \mathcal{F}_u^e$ and support set $\mathcal{S}_u$, we split items of $\mathcal{S}_u$ into several groups by rating scores and learn an item based user preference for u as follows:

$$\boldsymbol{u}_r = \tau(\boldsymbol{W}[\text{mean}_{\{i|i\in\mathcal{S}_u \wedge r_{u,i}=r\}}(\boldsymbol{i}^{ini}) \oplus \boldsymbol{u}^{ini}] + \boldsymbol{b}), \quad (4)$$

where $r \in R$ denotes a specific rating, $\text{mean}(\cdot)$ is mean pooling, $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and bias vector, and $\tau(\cdot)$ is the activation function which is stated as ReLU [Nair and Hinton, 2010]. Similarly, we can also get the item based preference for each friend in $\mathcal{F}_u^i$ and $\mathcal{F}_u^e$. We stack them by column and get the corresponding matrices denoted as $\boldsymbol{F}_r^i$ and $\boldsymbol{F}_r^e$, respectively. Note that although implicit and explicit friends are modeled independently, they share the same parameters in (4). For a specific rating r, different friends contribute differently to capturing u's preference. Therefore, we adopt the attention mechanism as follows:

$$\boldsymbol{f}_r^i = \boldsymbol{F}_r^i softmax(\boldsymbol{F}_r^{i\top} \boldsymbol{u}_r), \quad (5)$$

$$\boldsymbol{f}_r^e = \boldsymbol{F}_r^e softmax(\boldsymbol{F}_r^{e\top} \boldsymbol{u}_r). \quad (6)$$

Now, for a user, we obtain the item based preference $\boldsymbol{u}_r$, the implicit friends based preference $\boldsymbol{f}_r^i$, and the explicit friends based preference $\boldsymbol{f}_r^e$ for each rating r at a fine level.

Afterwards, we describe the ***coarse level preference modeling*** as shown in the right part of Fig. 3. Formally, we stack all $\boldsymbol{u}_r$ obtained from the fine level by column and get the matrix denoted as $\boldsymbol{U}$. An attention mechanism is applied to get the coarse level preference $\boldsymbol{u}$ which is learned from u's interactions as follows:

$$\boldsymbol{u} = \boldsymbol{U} softmax(\boldsymbol{W}_2 \tau(\boldsymbol{W}_1 \boldsymbol{U} + \boldsymbol{b}_1))^\top, \quad (7)$$

where $\boldsymbol{W}_1$ is the weight matrix, $\boldsymbol{b}_1$ is the bias vector, and $\boldsymbol{W}_2$ is the weight vector. Similarly, we can acquire the coarse level preferences $\boldsymbol{f}^i$ and $\boldsymbol{f}^e$ learned from implicit and explicit friends, respectively. Finally, we implement the following formula and obtain the overall preference $\boldsymbol{u}_o$:

$$\boldsymbol{u}_o = \boldsymbol{u} \oplus (\lambda_1 \boldsymbol{f}^i + \lambda_2 \boldsymbol{f}^e), \quad (8)$$

where λ_1 and λ_2 are learnable parameters that control the contributions of the implicit and explicit friends to modeling the user's preference.

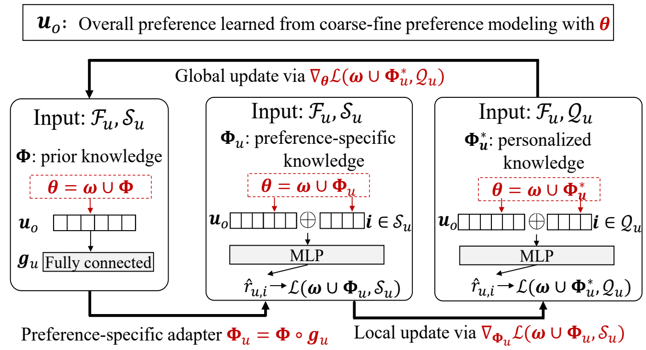

Figure 4: The flowchart overview of our meta-learning framework.

Prediction & Objective Function

Given the user's overall preference $\boldsymbol{u}_o$ and an unobserved item $\boldsymbol{i}$, we can predict the rating as follows:

$$\hat{r}_{u,i} = \text{MLP}(\boldsymbol{u}_o \oplus \boldsymbol{i}), \quad (9)$$

where MLP is a two-layer multilayer perceptron with ReLU activation functions. We minimize the following loss for the user u to optimize the parameters:

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_u) = \frac{1}{|\mathcal{D}_u|} \sum_{i \in \mathcal{D}_u} (r_{u,i} - \hat{r}_{u,i})^2, \quad (10)$$

where $\boldsymbol{\theta}$ includes all parameters, $\mathcal{D}_u$ is a set of items to be predicted, and $r_{u,i}$ is the actual rating of user u on item i.

4.3 Meta-learning Framework

Here, we describe the training procedure of our framework as well as our designed preference-specific adapter.

Preference-specific Adapter

Existing methods such as MeLU learn globally shared prior knowledge across all users, which may cause poor generalization. In contrast, we argue that closely correlated users may have similar preferences so they should locally share similar prior knowledge. Depend on that, we design a preference-specific adapter to customize the globally shared prior knowledge to preference-specific knowledge. Since the overall preference $\boldsymbol{u}_o$ can reflect the relations between users, it can help achieve this goal.

As shown in Fig. 4, we denote the parameters of feature embeddings as $\boldsymbol{\omega}$ and the rest as $\boldsymbol{\Phi}$, so that $\boldsymbol{\theta} = \boldsymbol{\omega} \cup \boldsymbol{\Phi}$. Note that $\boldsymbol{\Phi}$ is the prior knowledge. To generate similar knowledge for users with similar preferences, we design a series of preference-specific gates:

$$\boldsymbol{g}_u = \sigma(\boldsymbol{W}_g \boldsymbol{u}_o + \boldsymbol{b}_g), \quad (11)$$

where $\{\boldsymbol{W}_g, \boldsymbol{b}_g\} \in \boldsymbol{\Phi}$ are the weight matrix and bias vector, $\sigma(\cdot)$ is the sigmoid function, and $\boldsymbol{g}_u$ has the same shape with $\boldsymbol{\Phi}$. Then, the prior knowledge $\boldsymbol{\Phi}$ is adapted to the preference-specific knowledge $\boldsymbol{\Phi}_u$ of task $\mathcal{T}_u$ via the following equation:

$$\boldsymbol{\Phi}_u = \boldsymbol{\Phi} \circ \boldsymbol{g}_u, \quad (12)$$

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

Therefore, correlated users who share similar preferences will trigger similar parameter gates, resulting in similar model parameters and allowing more knowledge to be shared, while unrelated users are controlled to share less knowledge.

Algorithm 1: Training Procedure of PAML

Input: $\mathcal{T}^{\mathrm{tr}}$: meta-training tasks; α, β: learning rate
Output: Parameters $\boldsymbol{\theta}$
Randomly initialize $\boldsymbol{\theta} = \boldsymbol{\omega} \cup \boldsymbol{\Phi}$
while *not converge* **do**
    Sample a batch of tasks $\mathcal{B} = \{\mathcal{T}_u | \mathcal{T}_u \in \mathcal{T}^{\mathrm{tr}}\}$
    foreach *task* $\mathcal{T}_u = (\mathcal{F}_u^e, \mathcal{S}_u, \mathcal{Q}_u) \in \mathcal{B}$ **do**
        Identify implicit friends $\mathcal{F}_u^i$
        $\mathcal{T}_u = (\mathcal{F}_u^e \cup \mathcal{F}_u^i, \mathcal{S}_u, \mathcal{Q}_u)$
        Compute the overall preference $\boldsymbol{u}_o$ in (8)
        Compute preference-specific knowledge $\boldsymbol{\Phi}_u$ in (12)
        Local update: $\boldsymbol{\Phi}_u^* = \boldsymbol{\Phi}_u - \alpha \nabla_{\boldsymbol{\Phi}_u} \mathcal{L}\left(\boldsymbol{\omega} \cup \boldsymbol{\Phi}_u, \mathcal{S}_u\right)$
    Global update: $\boldsymbol{\theta} = \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\mathcal{T}_u \in \mathcal{B}} \mathcal{L}\left(\boldsymbol{\omega} \cup \boldsymbol{\Phi}_u^*, Q_u\right)$
return $\boldsymbol{\theta}$

Meta Optimization

Under the preference-specific knowledge, we aim to generate the personalized knowledge $\boldsymbol{\Phi}_u^*$ (i.e. initialization of parameters for u). Given the preference-specific knowledge $\boldsymbol{\Phi}_u$ and support set $\mathcal{S}_u$, $\boldsymbol{\Phi}_u^*$ is obtained by locally updating $\boldsymbol{\Phi}_u$ through several gradient descent steps:

$$\boldsymbol{\Phi}_u^* = \boldsymbol{\Phi}_u - \alpha \nabla_{\boldsymbol{\Phi}_u} \mathcal{L}\left(\boldsymbol{\omega} \cup \boldsymbol{\Phi}_u, \mathcal{S}_u\right), \tag{13}$$

where α is the learning rate.

During the inference stage, $\boldsymbol{\Phi}_u^*$ is adopted to make predictions for items in the query set $\mathcal{Q}_u \in \mathcal{T}^{te}$. During the training stage, we sample a batch of tasks $\mathcal{B}$ from meta-training tasks $\mathcal{T}^{tr}$. For each task, the learned personalized knowledge is utilized to calculate the loss on the query set $\mathcal{Q}_u$ and optimize all parameters. Overall, all parameters are globally updated as follows:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\mathcal{T}_u \in \mathcal{B}} \mathcal{L}\left(\boldsymbol{\omega} \cup \boldsymbol{\Phi}_u^*, Q_u\right), \tag{14}$$

where β is another learning rate. The training procedure is shown in Algorithm 1.

5 Experiments

In this section, we first introduce the experimental setup in details and then report the experimental results from three perspectives.

5.1 Experimental Setup

Datasets Description

We conduct experiments on two real-world datasets: Bouban Book[1] and Yelp[2], which are from publicly accessible repositories. Both datasets provide a large quantity of ratings, social relations and information of users and items. The rating scale is from 1 to 5, where higher score means stronger preference.

Considering cold-start scenarios, we first separate the users and items into two groups (existing/new) with a ratio of 8:2 for each dataset according to user joining time (or first user action time) and item releasing time. Then, for each dataset, we divide it into meta-training tasks and meta-testing tasks. The former only contain existing users and existing items. The latter include three kinds of cold-start scenarios, i.e., UC denotes the scenario where only the users are new, IC denotes the scenario where only the items are new, and UIC denotes the scenario where both users and items are new. Moreover, we randomly extract 10% of meta-training tasks as the traditional recommendation scenario which is denoted as NC.

Dataset	Douban Book	Yelp
# Users	6,576	25,783
# Items	20,547	33,105
# Ratings	326,419	727,259
Rating Sparsity	99.76%	99.91%
Avg. friends of each user	6.0	3.8
# Users without friends	1,314	10,867

Table 1: The statistics of two datasets.

To construct the support and query sets, we first keep users whose interaction history length is between 13 and 100 for Bouban Book, and keep users whose interaction history length is between 20 and 50 for Yelp. Then, for each user, 10 interacted items in history are randomly chosen to be the query set (i.e. $\mathcal{Q}_u$), the rest make up the support set (i.e. $\mathcal{S}_u$) , and social connected friends who belong to meta-training tasks are used as explicit friends (i.e. $\mathcal{F}_u^e$). We use the information of all support sets in the meta-training tasks to construct the HIN and identify implicit friends (i.e. $\mathcal{F}_u^i$). Especially, we use the attribute set $\{$*Publisher*$\}$ of items to construct the HIN for Douban Book and the attribute set $\{$*Stars*, *PostalCode*$\}$ of items to construct the HIN for Yelp. Table 1 lists the detailed statistics of two datasets.

Baselines

To validate the effectiveness of PAML, we choose three kinds of baselines. (1) Traditional methods, including FM [Rendle, 2010], NeuMF [He *et al.*, 2017] and Wide & Deep [Cheng *et al.*, 2016]. They are classic and widely used for recommendation. (2) Social based methods, including SoReg [Ma *et al.*, 2011] which used social regularization with the assumption that connected users would share similar latent embeddings and DiffNet [Wu *et al.*, 2019] which used the GCN method to propogate social relations. (3) Meta-learning based methods, including MetaEmb [Pan *et al.*, 2019], MeLU [Lee *et al.*, 2019] and MAMO [Dong *et al.*, 2020]. They are designed for solving the cold-start problem by meta-learning algorithms.

Evaluation Metrics

We adopt two popular metrics. One is the root mean square error (RMSE) which is used to evaluate the predictive accuracy [Liu *et al.*, 2019]. Smaller value of RMSE indicates better predictive accuracy. The other is the normalized discounted cumulative gain at rank K (nDCG@K) which is used to evaluate top-K ranking performance. In this paper, we set K = 5 and the larger the value, the better the performance.

Parameters Setting

For parameters in PAML, they are randomly initialized following the xavier normal distribution [Glorot and Bengio, 2010]. We set the dimension of the feature embeddings to

[1] https://book.douban.com
[2] https://www.yelp.com/dataset

Scenario	Model	Douban Book		Yelp		Scenario	Model	Bouban Book		Yelp	
		RMSE	nDCG@5	RMSE	nDCG@5			RMSE	nDCG@5	RMSE	nDCG@5
UC	FM	0.8884	0.8287	1.1856	0.7581	IC	FM	0.8853	0.8314	1.0936	0.7573
	NeuMF	0.8311	0.8364	1.1761	0.7644		NeuMF	0.8003	0.8288	1.0471	0.7751
	Wide & Deep	0.7988	0.8483	1.1635	0.7765		Wide & Deep	0.7923	0.8497	1.0449	0.7802
	SoReg	0.8741	0.8114	1.1654	0.7614		SoReg	0.8705	0.8344	1.0593	0.7655
	DiffNet	0.7758	0.8653	1.1094	0.8032		DiffNet	0.7810	0.8536	1.0528	0.7910
	MetaEmb	0.7662	0.8833	1.1087	0.8262		MetaEmb	0.7624	0.8836	1.0178	0.8266
	MeLU	0.7650	0.8920	1.0972	0.8320		MeLU	0.7666	0.8890	1.0107	0.8356
	MAMO	0.7849	0.8583	1.1431	0.8213		MAMO	0.7592	0.8766	1.0243	0.8330
	PAML	**0.7246**	**0.9117**	**1.0506**	**0.8412**		**PAML**	**0.7355**	**0.9108**	**0.9404**	**0.8505**
UIC	FM	0.9094	0.8296	1.0774	0.7649	NC	FM	0.8888	0.8470	1.0328	0.7953
	NeuMF	0.8735	0.8304	1.0660	0.7718		NeuMF	0.8380	0.8521	1.0059	0.8016
	Wide & Deep	0.8027	0.8510	1.0391	0.7838		Wide & Deep	0.8097	0.8677	0.9993	0.8133
	SoReg	0.8930	0.8084	1.0469	0.7713		SoReg	0.8757	0.8495	1.0009	0.7986
	DiffNet	0.7888	0.8541	1.0037	0.8033		DiffNet	0.7750	0.8812	0.9471	0.8455
	MetaEmb	0.7651	0.8805	1.0008	0.8270		MetaEmb	0.7819	0.8880	0.9582	0.8475
	MeLU	0.7795	0.8903	0.9942	0.8323		MeLU	0.7793	0.9144	0.9621	0.8559
	MAMO	0.7679	0.8702	0.9945	0.8421		MAMO	0.8072	0.8893	0.9929	0.8479
	PAML	**0.7496**	**0.9146**	**0.9536**	**0.8489**		**PAML**	**0.7604**	**0.9284**	**0.8997**	**0.8662**

Table 2: Performance comparisons in four scenarios on two datasets.

Model	Bouban Book		Yelp	
	RMSE	nDCG@5	RMSE	nDCG@5
PAML	**0.7496**	**0.9146**	**0.9536**	**0.8489**
PAML-I-E-A*	0.7795	0.8903	0.9942	0.8323
PAML-I	0.7732	0.9056	0.9743	0.8406
PAML-E	0.7512	0.9097	0.9866	0.8392
PAML-I-E	0.7743	0.9045	0.9858	0.8343
PAML-A	0.7538	0.9036	0.9659	0.8364

* PAML-I-E-A is equivalent to MeLU.

Table 3: Ablation study in UIC.

32 and batch size to 64. Two layers used for the prediction are with 64 nodes each. We set the local and global learning rate (i.e., α, β) to 0.001 and 0.001 for Bouban Book, 0.001 and 0.0005 for Yelp, respectively. For two datasets, the number of implicit friends is empirically fixed to 5 by default , and the number of local updates is fixed to 1 by default. The sensitivity of some important hyper-parameters is discussed in Section 5.2. In addition, the hyper-parameters for baselines are set as stated in the corresponding papers and tuned carefully to achieve the best performance for fair comparisons.

In this paper, our proposed PAML is implemented by Pytorch and trained on a Linux system (2.10GHz Intel Xeon Gold 6230 CPUs and a Tesla V100 GPU).

5.2 Experimental Results

Overall Performance

In this experiment, we compare the overall performance of all methods on two datasets. Specifically, Table 2 shows the comparison results on both datasets in three cold-start scenarios and the non-cold-start scenario.

From the results, we observe our PAML consistently yields the best performance among all methods on two datasets. For instance, PAML relatively improves over the best baseline w.r.t. RMSE by 1.9-5.3% on Bouban Book and 4.1-6.9% on Yelp. By comparing the results in different scenarios, we can also find it is more difficult to make predictions in cold-start scenarios than that in the non-cold-start one. Among baselines, meta-learning based methods (i.e., MAMO, MeLU and MetaEmb) perform better than the other two kinds of methods, especially in cold-start scenarios, which validates the advantage of the meta-learning frameworks in alleviating the cold-start issue. In addition, DiffNet is a competitive model since it adopts GCN to diffuse the social influence, enriching social relations to a certain extent, and achieves the best performance among the social based methods and traditional methods, especially in the non-cold-start scenario. The rest of baselines are least competitive because they suffer from limited capacity to express user preferences by features and scarce labeled data.

Ablation Study

We conduct an ablation study to investigate the impact of different components in PAML. Here, we only report the performance in the typical scenario UIC. As for the others, the trends are similar. As shown in Table 3, "I" denotes the component of implicit friends while "E" denotes that of explicit friends. "A" denotes the preference-specific adapter. "-" denotes removing the following component. Note that MeLU is a special case of PAML which is equivalent to PAML-I-E-A.

We first study the effect of social relations. Since we use both implicit friends and explicit friends, we consider three variants of PAML including PAML-I, PAML-E and PAML-I-E. The results clearly demonstrate that the social relations could contribute to modeling the user's preference so that facilitating the performance. In addition, we realize the implicit friends contribute more than explicit ones on Bouban Book, but it becomes opposite on Yelp.

We then explore the effect of preference-specific adapter. As the preference-specific adapter plays a pivotal role in our model, we give the results of the variant PAML-A which directly adapts the prior knowledge to personalized knowledge.

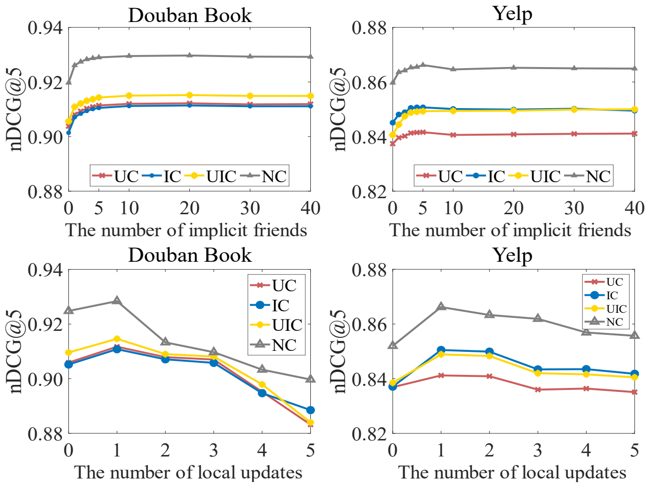

Figure 5: Parameter sensitivity in different settings.

The results not only prove our claim that users with similar preferences should locally share prior knowledge is reasonable but also demonstrate our proposed preference-specific adapter is effective.

Parameter Sensitivity

Finally, we conduct parameter sensitivity experiments on two datasets. As mentioned in Section 4.1, the top similar users are chosen as implicit friends of the user u. Therefore, we explore how the number of implicit friends would impact on the performance. As shown in the top half of Fig. 5, for the Bouban Book dataset, nDCG@5 increases quickly from 0 to 5 and then reaches to a stable level. For the Yelp dataset, increasing the number of implicit friends does not lead to continuous improvements but a slight drop. We guess the reason is that more implicit friends may introduce noise.

In addition, we analyze the effect of the number of local updates in the meta-learning process. The bottom half of Fig. 5 shows nDCG@5 of PAML for varying the number of local updates from 0 to 5. The results reach the optimal performance at one local update, and then as the number of local updates increases, nDCG@5 gradually decreases, which may be due to overfitting on the support set.

6 Conclusion

In this paper, we proposed a Preference-Adaptive Meta-Learning approach (PAML) for improving existing meta-learning frameworks with better generalization capacity. By leveraging users' social relations and our proposed preference-specific adapter, correlated users who share similar preferences could trigger similar knowledge. Benefits from that, the meta-learning algorithm could have better generalization capacity, so the prior knowledge could be quickly adapted to new users with sparse interactions. The proposed method was evaluated on two real-world datasets, showing that PAML outperforms the competing baselines. The ablation study demonstrated the power of social relations and the effectiveness of the preference-specific adapter.

Acknowledgments

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. U20A20229, 61922073, 61976198 and 62022077), the National Key R&D Program of China under Grant No. 2020AAA0103800, and the Fundamental Research Funds for the Central Universities (Grant No.WK2150110021 and WK2150110017).

References

[Aral *et al.*, 2009] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[Cheng *et al.*, 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa 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.

[Dong *et al.*, 2020] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming 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.

[Fan *et al.*, 2019] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426, 2019.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1126–1135, 2017.

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

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NeurIPS'14*, volume 27, pages 2672–2680, 2014.

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

[Huang *et al.*, 2019] Zhenya Huang, Qi Liu, Chengxiang Zhai, Yu Yin, Enhong Chen, Weibo Gao, and Guoping Hu. Exploring multi-objective exercise recommendations in online education systems. In *Proceedings of the 28th ACM*

International Conference on Information and Knowledge Management, pages 1261–1270, 2019.

[Jin *et al.*, 2020] Binbin Jin, Defu Lian, Zheng Liu, Qi Liu, Jianhui Ma, Xing Xie, and Enhong Chen. Sampling-decomposable generative adversarial recommender. In *Advances in Neural Information Processing Systems*, volume 33, pages 22629–22639, 2020.

[Lee *et al.*, 2019] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee 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.

[Li *et al.*, 2018] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1734–1743, 2018.

[Liu *et al.*, 2019] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, 2019.

[Lu *et al.*, 2020] Yuanfu Lu, Yuan Fang, and Chuan 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.

[Ma *et al.*, 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296, 2011.

[Mi *et al.*, 2019] Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3151–3157, 2019.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[Pan *et al.*, 2019] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704, 2019.

[Pei *et al.*, 2020] Hongbin Pei, Bingzhe Wei, Kevin Chang, Chunxu Zhang, and Bo Yang. Curvature regularization to prevent distortion in graph embedding. In *Advances in Neural Information Processing Systems*, volume 33, pages 20779–20790. Curran Associates, Inc., 2020.

[Rendle, 2010] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[Vartak *et al.*, 2017] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Proceedings of NeurIPS'17*, pages 6904–6914, 2017.

[Wang *et al.*, 2013] Hao Wang, Binyi Chen, and Wu-Jun Li. Collaborative topic regression with social regularization for tag recommendation. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[Wang *et al.*, 2019] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. Mcne: An end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1064–1072, 2019.

[Wu *et al.*, 2019] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 235–244, 2019.

[Yao *et al.*, 2019] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7045–7054, 2019.

[Yu *et al.*, 2018] Junliang Yu, Min Gao, Jundong Li, Hongzhi Yin, and Huan Liu. Adaptive implicit friends identification over heterogeneous network for social recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 357–366, 2018.

[Yu *et al.*, 2019] Junliang Yu, Min Gao, Hongzhi Yin, Jundong Li, Chongming Gao, and Qinyong Wang. Generating reliable friends via adversarial training to improve social recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 768–777. IEEE, 2019.

[Zhao *et al.*, 2014] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 261–270, 2014.

[Zhu *et al.*, 2020] Yaohui Zhu, Chenlong Liu, and Shuqiang Jiang. Multi-attention meta learning for few-shot fine-grained image recognition. In *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence*, pages 1090–1096, 2020.