Personalized Adaptive Meta Learning for Cold-start User Preference Prediction

Runsheng Yu,[1] Yu Gong, [2] Xu He, [1] Yu Zhu , [2] Qingwen Liu,[2] Wenwu Ou,[2] Bo An [1]
[1] School of Computer Science and Engineering, Nanyang Technological University
[2] Alibaba Group
runshengyu@gmail.com, gongyu.gy@alibaba-inc.com, hexu0003@e.ntu.edu.sg, boan@ntu.edu.sg, zy143829@alibaba-inc.com, xiangsheng.lqw@alibaba-inc.com, santong.oww@taobao.com

Abstract

A common challenge in personalized user preference prediction is the cold-start problem. Due to the lack of user-item interactions, directly learning from the new users' log data causes serious over-fitting problem. Recently, many existing studies regard the cold-start personalized preference prediction as a few-shot learning problem, where each user is the task and recommended items are the classes, and the gradient-based meta learning method (MAML) is leveraged to address this challenge. However, in real-world application, the users are not uniformly distributed (i.e., different users may have different browsing history, recommended items, and user profiles. We define the major users as the users in the groups with large numbers of users sharing similar user information, and other users are the minor users), existing MAML approaches tend to fit the major users and ignore the minor users. To address this cold-start task-overfitting problem, we propose a novel personalized adaptive meta learning approach to consider both the major and the minor users with three key contributions: 1) We are the first to present a personalized adaptive learning rate meta-learning approach to improve the performance of MAML by focusing on both the major and minor users. 2) To provide better personalized learning rates for each user, we introduce a similarity-based method to find similar users as a reference and a tree-based method to store users' features for fast search. 3) To reduce the memory usage, we design a memory agnostic regularizer to further reduce the space complexity to constant while maintain the performance. Experiments on MovieLens, BookCrossing, and real-world production datasets reveal that our method outperforms the state-of-the-art methods dramatically for both the minor and major users.

Introduction

Recommender Systems (RS) help people to discover the items they prefer (Guo et al. 2017; Qu et al. 2016). In order to train a well-performing personalized user preference predictor, enough interactions with users are indispensable. To address this challenge, many researchers take advantage of the offline supervised training methods, which leverage the historical data (log data) to train the model. However, when the log data is lacking, these supervised training methods cause the over-fitting issues (Vanschoren 2018), which is known as the *cold-start problem*.

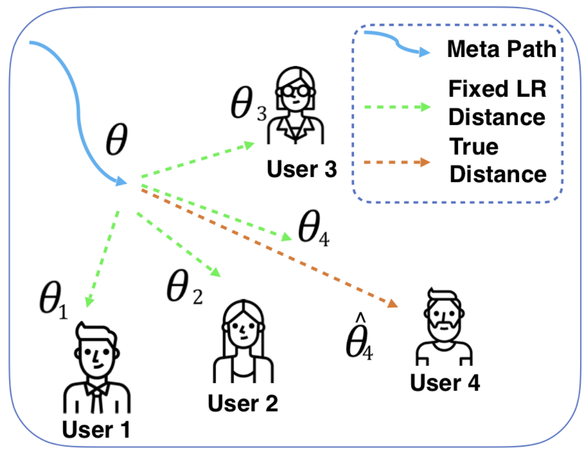

Figure 1: An example illustrating that MAML has limitations in the cold-start user imbalanced dataset. Users $1, 2, 3$ are the major users and user 4 is the minor user.

To train a well-performing model in the cold-start problem, meta learning-based approaches are introduced (Lee et al. 2019; Dong et al. 2020). Intuitively, learning with a few samples can be viewed as a few-shot learning problem and meta learning, especially gradient-based meta learning (e.g., Model Agnostic Meta Learning, MAML (Finn, Abbeel, and Levine 2017)), which aims to adapt to any task with only a few steps of parameter update, has been proven as one of the most successful approaches for these problems. Thanks to its good generalization ability, MAML has already been leveraged into various domains, including computer vision, natural language process, and robotics (Gui et al. 2018; Madotto et al. 2019; Finn et al. 2017). Recently, in the RS area, meta learning is introduced for the cold-start problem for either users or items, which treats the users/items as tasks, log data as samples, and learns to do fast adaptation when meeting new tasks (Dong et al. 2020; Pan et al. 2019; Lee et al. 2019; Luo et al. 2020).

However, these MAML methods assume that the distribution of tasks (users) is uniform. But in the recommendation systems, the user distribution is not always balanced, i.e., the user profile, browsing history, as well as the features of recommended items are not always uniformly distributed[1], which harms the performance of MAML. For example, regarding the user profile, as shown in Tab. 1, most values are imbalanced. In terms of the browsing histories, the distribution of the browsing histories is also non-uniform (Pan et al. 2019). Therefore, simply learning from these data will overfit the users with similar feature values since these users own the major similar features and fitting these users can already achieve good-enough *average* prediction performance, which is known as the cold-start user (task) overfitting problem. To illustrate the problem clearly, we regard a collection of users with similar feature values as a group and define the users as the major users when the number of users in these groups are large, and others users are the minor users. Now, we give an example to illustrate how imbalanced distribution harms the performance of MAML: as shown in Fig. 1, assuming that an MAML strategy aims to learn to do fast adaptation to the four cold-start users (three major users (users 1-3) and one minor user (user 4)). Different locations in the blue square indicate feature values (embeddings) for different users. Since three of them are near each other (have similar feature values), the model may *not* focus on the minor user (user 4) because fitting the major users can already achieve good prediction performance.

To address these limitations, in this paper, we propose a personalized adaptive meta learning approach to improve the performance of MAML in the cold-start user preference prediction problem. The main idea is to set different learning rates for different users so as to find task-adaptive parameters for each user. Our main contributions are as follows: 1) We are the first to introduce a novel user-adaptive learning rate based meta learning in RS to improve the performance of MAML by focusing on both the major and minor users. 2) To provide better-personalized learning rates, we introduce a method to find similar users as a reference and a tree-based method to store users' features for fast search. 3) To reduce the memory usage, we introduce a memory agnostic regularizer to reduce the space complexity while maintaining good prediction performance. Experiments on MovieLens-1M, BookCrossing, and real-world production datasets from one of the largest e-commerce platforms reveal that our method outperforms state-of-the-art methods dramatically for both the minor and the major users.

Related Works

In this section, we discuss some related works including gradient-based meta learning for the imbalanced dataset and the meta learning for cold-start recommendation..

Gradient-Based Meta Learning for Task Overfitting Problem. MAML based methods have been widely adopted for studying the few-shot learning problem (Finn, Abbeel, and Levine 2017; Li et al. 2017; Xu, van Hasselt, and Silver 2018; Chen et al. 2018; Ravi and Larochelle 2016; Lee and Choi 2018). To consider task-adaptive challenges, the vector of learning rates, the block-diagonal preconditioning matrix, latent embedding, and interleaving warp-layers are designed to learn parameters for different tasks respectively (Li et al. 2017; Park and Oliva 2019; Rusu et al. 2018; Flennerhag et al. 2020). However, those methods still do not explicitly consider the task overfitting problem. To take this problem into consideration, Bayesian TAML introduces different initiation parameters and inner gradient approaches to handle class imbalance, task imbalance, as well as out-of-distribution tasks together by higher-order statistics (Flennerhag et al. 2020). However, in RS, a users' feature space is high-dimensional and those statistics are still hard to represent the embedding completely. As far as we know, we are the *first* to propose a novel meta learning method focusing on the two key challenges in cold-start RS: 1) comparing with the common-seen few-shot learning problem which does not explicitly have the task information, the dimension of features for the task information (like the user profile) in our problem is large, and 2) task does not obey the uniform distribution (even long-tail) while the standard few-shot learning problem is assumed uniformly distributed.

The ratio of features			
Age	Gender	Zipcode	Occupations
72.4%	71.7%	56.7%	64.0%

Table 1: The ratio ($\frac{\text{the number of users having certain feature values}}{\text{total number of users}}$) of the top 30% largest number of feature values that users own[2], ratio of certain users ($\frac{\text{the number of users in certain group}}{\text{total number of users}}$), and the Mean Squared Errors (MSE) from one of the MAML methods (Lee et al. 2019) in MovieLens.

Meta Learning for Cold-start Recommendation. In RS, meta learning is mainly used to do fast adaptation with only few samples for users and items, including the item cold-start/warm-up problem (Vartak et al. 2017; Pan et al. 2019) and the user cold-start problem (Dong et al. 2020; Lee et al. 2019; Bharadhwaj 2019). For the item cold-start problem, linear weight adaptation is introduced to represent the items by the users who have viewed it (Vartak et al. 2017) while Meta-Embedding is given to generate different embeddings for cold-start processes (Pan et al. 2019). For the user cold-start approaches, MAML is used to address the challenge by learning fast adaptation to new users. But those approaches do not consider the task overfitting problems. Differently, we are the *first* to investigate the personalized learning rate based meta learning to address the cold-start user overfitting problem in RS, considering both major and minor users together.

Problem Formulation

This section discusses the gradient-based meta learning in RS and how user imbalance affects the performance of MAML. Formally, following (Lee et al. 2019; Finn, Abbeel, and Levine 2017), the objective of meta learning in RS scenario can be described as: $\arg\min_{\boldsymbol{\theta}} \left[\frac{1}{|\mathcal{T}|}\sum_{\mathcal{T}_i \in \mathcal{T}} L_{\mathcal{T}_i}(\boldsymbol{\theta}_i)\right]$ s.t. $\boldsymbol{\theta}_i = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L_{\mathcal{T}_i}(\boldsymbol{\theta})$, where i is the i-th index for user $u_i \in \mathcal{U}$ (or task, since each user is treated as a task in our RS settings), $L_{\mathcal{T}_i}$ is the loss function for user i (e.g., mean squared error) and the

[1]They are collectively named as the feature values.

item-feedback set $\tau_i = \{\langle \text{item}_i^j, \text{score}_i^j \rangle_{j=1}^{\hat{N}}\}$ is the set of items $\langle \text{item}_i^j \rangle_{j=1}^{\hat{N}}$ ($\hat{N}$ items in total) and their corresponding feedbacks $\langle \text{score}_i^j \rangle_{j=1}^{\hat{N}}$ (e.g., ranking score) by user u_i. L_τ is the loss for different tasks and $\tau \supseteq \tau_i$ is the union set of the item-feedback subsets for all the users. α is the (inner) Learning Rate (LR). For simplification, we use L_τ to replace $\frac{1}{|\tau|}\sum_{\tau_i \in \tau} L_{\tau_i}$. Before we discuss the user overfitting problem, we present a special case that helps us to analyze the problem better: users are clustered into two groups with densities p_1, p_2 ($p_1 \geq p_2$). Users' actual preferences are x_1, x_2 for group 1 and 2 respectively. Now, we show that MAML ignores the minor users in this case:

Lemma 1. *Suppose that the loss function is defined as* $L = \sum_{i=1}^{2} p_i(\theta_i - x_i)^2$ *, where* x_1, x_2 *are the actual preferences for task* 1 *and* 2*, and* $\theta_i = \theta - \alpha\nabla_\theta L_{\tau_i}(\theta)$*, when* $p_1 \geq p_2$*,* $x_1 \geq x_2$*, we have* $(\theta_1^* - x_1)^2 \leq (\theta_2^* - x_2)^2$*, where* θ^* *is the optimal parameter* θ*.*

All the proofs can be found in Appendix A. This case indicates that the MAML method tends to optimize the major users preferentially.

Methodology

This section presents a practical adaptive Learning Rate (LR)-based meta-learning method to address the user overfitting problem, including 1) why adaptive LR can solve the user overfitting problem and how to apply adaptive LR by end-to-end training, 2) a similarity-based learning approach to improve the performance by considering similar users and an approximated tree-based implementation for fast search, and 3) a memory agnostic regularizer to further reduce space complexity to constant.

Adaptive Learning Rate based MAML

To consider both major and minor users together, we propose an adaptive learning rate approach to design different gradient steps for different users :

$$\arg\min_{\boldsymbol{\theta}} \left[\frac{1}{|\tau|}\sum\nolimits_{\tau_i \in \tau} L_{\tau_i}(\boldsymbol{\theta}_i)\right], \text{ s.t. } \boldsymbol{\theta}_i = \boldsymbol{\theta} - \alpha(h_i)\nabla_{\boldsymbol{\theta}} L_{\tau_i}(\boldsymbol{\theta}), \tag{1}$$

where $h_i \in \mathbb{H}$ is the i-th user's feature embeddings[3] (or user embedding for short). The difference between ours and MAML is that the LR in our method is a mapping from the user embedding to a real number rather than a fixed LR. Intuitively, with an adaptive learning rate, the meta agent can fit any user even if it is far from the meta strategy (like the user 4 in Fig. 1). Here, an analysis is given to illustrate the adaptive learning rate can get better results in user imbalanced dataset:

Lemma 2. *Based on Lemma 1, we further defined as* $L' = \sum_{i=1}^{2} p_i(\theta_i' - x_i)^2$*, s.t.* $\theta_i' = \theta - \alpha(h_i)\nabla_\theta L_{\tau_i}(\theta)$*, (*$x_1, x_2$ *are the target values for task* 1 *and* 2*, where* $x_2 > x_1$*) there exists* α_1 *and* α_2 *that satisfying*

$$(\theta_2^* - x_2)^2 \geq (\theta_2^{*\prime} - x_2)^2 \textit{and } L^* \geq L^{*\prime},$$

where L^* *and* $L^{*\prime}$ *are the optimal loss functions for standard MAML and adaptive MAML respectively.* $\theta^{*\prime}$ *is the optimal* θ' *parameters.*

Lemma 2 indicates that learning a personalized adaptive learning rate provides a lower loss value for minor users than the standard MAML method, and is able to achieve a lower total loss value than standard method ($L^* \geq L^{*\prime}$). This coincides with our conjecture that the adaptive LR meta learning methods perform better than the fixed LR methods when facing an imbalanced distribution.

Therefore, the main challenge is to capture the relationship among the user distribution, the parameters of network, and user own learning rate. To construct it, we leverage the end-to-end data-driven learning approach to map each user's features into different high-dimensional embedding and obtain the individual learning rate adaptively. Based on this idea, we propose the Personalized Adaptive Meta Learning (PAML) method, a direct way to train the network by taking the gradient descent of Eq. (1):

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \beta\sum\nolimits_{\tau_i} \nabla_{\boldsymbol{\theta}} L_{\tau_i}(\boldsymbol{\theta}_i)\left(I - \alpha(h_i;\boldsymbol{\psi})\nabla^2_{\boldsymbol{\theta}} L_{\tau_i}(\boldsymbol{\theta})\right),$$

$$\boldsymbol{\psi} = \boldsymbol{\psi} + \beta\nabla_{\boldsymbol{\psi}} L_\tau(\boldsymbol{\theta})\left(\sum\nolimits_{\tau_i} \nabla_{\boldsymbol{\psi}}\alpha(h_i;\boldsymbol{\psi})L_{\tau_i}(\boldsymbol{\theta})\right).$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are the parameters for the model and learning rate[4]. β is the outer learning rate. The two equations above are drawn by the chain rule.

Approximated Tree-based PAML

The key for PAML is to get accurate personalized $\alpha(h)$ for different users. However, directly using Fully-Connected (FC) layers to learn $\alpha(h_i)$ is hard since LRs are related to the task distribution and FC layers are not capable to memorize a large number of the users. To address the challenge, one straightforward idea is to consider other similar users' features as a reference since users with similar feature embeddings share similar LRs. Based on this idea, we introduce the similarity-based method to find users with similar high-level features (the embedding) and interests.

Formally speaking, when a new user u_i with embedding $h_i \in \mathbb{H}$ comes, our goal is to find the users with the most similar embeddings to u_i. Here, we define a similarity function $s_k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ to estimate the embedding similarity between user u_i and user u_k. After computing top-K nearest (the most similar) users with the values of similarity, these values can be treated as a reference to obtain a personalized adaptive learning rate. Thus, we can get an LR function for a new user u_i: $\alpha(h_i) := \alpha'(h_i) + \tilde{\alpha}$, by considering the user information and the user similarities together, where $\tilde{\alpha} = \sum_k \frac{s_k}{\sum_j s_j + \sigma}\alpha(h_k)$ is the weighted average sum of the nearest users' LRs, s_k is the k-th existing nearest user's similarity value to user i, $\alpha'(h_i)$ is neural network modules mapping from user embedding to a real value, and σ is a small value (10^{-5}). Here, we set the Gaussian kernel function $s_k = \exp\left(-\delta\left\|h_i - h_k\right\|^2\right)$ as the similarity function,

[3] User's feature embeddings can include different user information like the user profile embeddings and user browsing history embeddings.

[4] For simplification, we use $\alpha(h_i)$ to replace $\alpha(h_i;\boldsymbol{\psi})$ if not specially mentioned.

where δ is a constant and $\|\cdot\|$ is the 2-norm. Due to the limited space, details about how the kernel-based function can be used to estimate the similarity can be found in Appendix B.

Since we need to find the top-K nearest users in every training step, it is necessary to find a fast searching approach to accelerate the training process. Here, we leverage the kd-tree (Muja and Lowe 2014) as the basic structure to store users' embeddings. Specifically, we first initialize the tree structure with several users' ($\geq K$) embeddings and its corresponding LRs (the warm-up stage). Then, when a new user u_i with embedding h_i comes, we search the K-th nearest users in the tree. After that, we add the new user embedding and its LR into that tree and rebuild that tree. We remove the least frequent used user's embedding and its LR when the tree-size is larger than the memory size we set manually. Based on this structure, the time complexity of searching each user can be reduced from $\mathcal{O}(n)$ (brute force method) to $\mathcal{O}(\sqrt{n})$ for two-dimension embedding cases (Yianilos 1993).

However, this structure induces bias because the user embedding layers are dynamically updated during the training process but the previous users' embeddings stored in the tree nodes are fixed, causing the new embeddings and the old embeddings unfitted. In order to coordinate them, we also let the tree nodes be dynamically updated by gradient descent: $\text{node}_j = \text{node}_j - \beta \nabla_{\text{node}_j} L_\tau$, where node_j indicates the j-th node value (the user's embedding). The tree-based structure will also be updated once the embeddings change. The learning-rates stored in nodes are also updated similarly.

To further reduce space usage and speed up, we use the approximated store and search methods in place of the exact (precious) ones. That is, we can take advantage of the randomized kd-tree algorithm (Muja and Lowe 2014) to achieve the approximation. These structures can be easily implemented by the open-source python package (e.g., Pyflann (Muja and Lowe 2013)). We name it as Approximated-Tree PAML (AT-PAML). The loss function is the same as Eq. (1) and the pseudo-code can be found in Algs. 1.

Algorithm 1: Personalized Adaptive Meta Learning

Input: User distribution $p(\mathcal{U})$, the learning rate β.
Initialize the meta-policy with parameters θ;
for *episode* T **do**
  sample N of users from $p(\mathcal{U})$;
  for *each user* $i \in U$ **do**
    Split the support set τ_i and query set $\hat{\tau}_i$ randomly;
    Extract user embedding h_i;
    if Approximated Tree-based method **then**
      if *Warm-up stage* **then**
        Set LR as α;
        Store_Node(h_i, α);
      else
        $\langle h_k \rangle_{k=1}^{K}, \langle \alpha(h_k) \rangle_{k=1}^{K}$ = Search_Tree(h_i);
        $s := \{s_k\}_{k=1}^{K} = \{\langle \phi(h_i), \phi(h_k) \rangle_{\mathcal{H}}^2\}_{k=1}^{K}$;
        Obtain $\alpha(h_i) = \sum_k \frac{s_k}{\sum_j s_j + \sigma} \alpha(h_k) + \alpha'(h_i)$;
        Store_Node(h_i, $\alpha(h_i)$);
    else if *Regularized-based method* **then**
      Obtain the learning rate α directly by $\alpha(h_i)$;
    Update $\theta^i \leftarrow \theta - \alpha(h_i) \nabla_\theta \mathcal{L}_{\tau_i}(;\theta)$ with support set τ_i;
  if *Regularized-based method* **then**
    update θ by Eq. (4) with query set $\hat{\tau} = \langle \hat{\tau}_i \rangle_{i=1}^{N}$;
  else if Approximated Tree-based method **then**
    update θ by Eqs. (5) & (6) with query set $\hat{\tau} = \langle \hat{\tau}_i \rangle_{i=1}^{N}$. (Eqs. (4), (5) and (6) are shown in Appendix C).
Output: a well-trained meta-strategy.

Regularizer-based PAML

However, for large-scale real-world applications (e.g., Netflix) which involve more than millions of users, even the linear space complexity methods (the space size is proportional to the number of user embeddings) to store user embedding is unacceptable. Thus, a constant space complexity algorithm needs to be proposed. A widely adopted method to achieve long-term memorization is the continual learning approach (Aljundi et al. 2018; Kirkpatrick et al. 2017), which designs losses and training schemes to memorize the previous tasks. Inspired by their works, we design an auxiliary loss function to help the model to remember the users it has seen. That is, for any two users u_i and u_j, memorizing the information of those two users means the loss values for those users are similar: i.e., $\|L_{\tau_i}(\boldsymbol{\theta}_i) - L_{\tau_j}(\boldsymbol{\theta}_j)\|$. Similarly, the multi-user loss is $\sum_{u_i,u_j \in \mathcal{U}; i>j} \|L_{\tau_i}(\boldsymbol{\theta}_i) - L_{\tau_j}(\boldsymbol{\theta}_j)\|$. It is easy to understand because forgetting a task means its loss value for that task is higher than the loss value of other tasks. If the model can obtain qualified and similar loss values for each user, we can guarantee that the model has memorized all existing users' information.

However, it is hard to implement it into practice since computing multi-user loss needs to calculate all the user-pair losses together. To address this challenge, we turn to optimize its upper bound:

$$\sum_{u_i,u_j \in \mathcal{U}; i>j} \|L_{\tau_i}(\boldsymbol{\theta}_i) - L_{\tau_j}(\boldsymbol{\theta}_j)\| \leq \sum_i (|\mathcal{U}| - 1) \|\frac{\partial L_{\tau_i}}{\partial \boldsymbol{\theta}}\|^2 \|\alpha(h_i)\| + C,$$

where C is a constant. *The proof of the upper bound can be found in Appendix A.2.* Notice that the sum of $\|\frac{\partial L_{\tau_i}}{\partial \boldsymbol{\theta}}\|^2 \|\alpha(h_i)\|$ can be divided into each task (user) loss as a regularizer to enable the network to remember each user information: i.e., for user i, the regularizer L_i^r is set as: $L_i^r = \|\frac{\partial L_{\tau_i}}{\partial \boldsymbol{\theta}}\|^2 \|\alpha(h_i)\|$. Intuitively, when $L_i^r = 0$, it indicates that the meta strategy has already obtained a good score

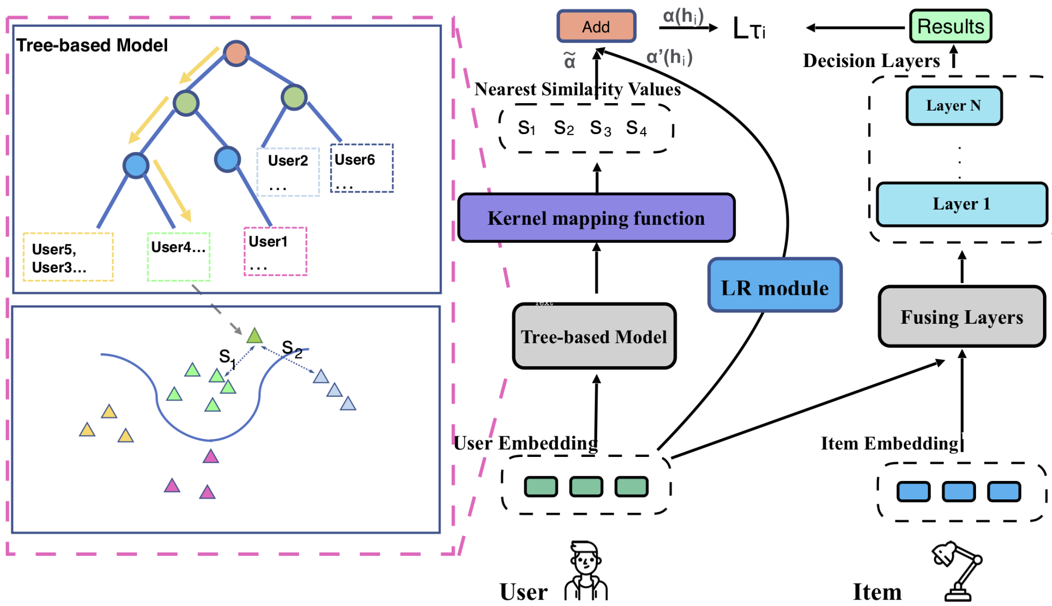

Figure 2: The network structure of the tree-based PAML. The right part is the network for inner update while the left part is the tree-based module. Specifically, *Fusing Layer* indicates the concatenation of user and item embeddings, *Add* is an adding function: $\alpha'(h_i) + \tilde{\alpha}$. $Layer1, \ldots N$ means fully connected layers and $S_1, \ldots S_4$ are similarity values.

for user i without training. Therefore, this term reinforces the model to remember the users it meets. Moreover, since $|\mathcal{U}|-1$ is a constant, we can replace it with a positive real number γ to balance $L^r_{\tau_i}$ and L_{τ_i}. We name the PAML with this REGularizer as *REG-PAML*. Since we do not need any extra space to store user embedding, the space complexity is *constant*. The total loss function for REG-PAML is $\mathcal{L} = \sum_i L_{\tau_i} + \gamma L^r_i$. The relationship to implicit MAML (Rajeswaran et al. 2019) can be found in Appendix C. The pseudo-code can also be found in Alg. 1.

Algorithm 2: Store and Search approaches

```
Def Store_Node(h_i, α):
    if Memory is full then
        Remove the embeddings and LR that are least
          recently used;
    Store the embeddings, LR, and build the standard
      kd-tree by embedding (Friedman, Bentley, and
      Finkel 1976);
    return the kd-tree;
Def Search_Tree(h_i):
    Search the nearest neighbours (Yianilos 1993) and
      their corresponding LRs;
    return the nearest neighbours and their
      corresponding LRs ;
```

Empirical Studies

This section validates our methods in various environments and tasks, including the rating prediction (in the MovieLens-1M dataset and the BookCrossing dataset) as well as the CTR prediction (the real-world production dataset). We also analyze the performance of our methods for both the minor and major users and conduct ablation studies.

Experimental Setup

Datasets. We use both open-source datasets as well as real-world production dataset to evaluate the performance of our methods in user imbalanced dataset, including MovieLens-1M (Harper and Konstan 2015), BookCrossing (Ziegler et al. 2005) as well as production dataset (collected from Taobao e-commerce platforms, which is somewhat similar to (Zhao et al. 2018; Guo et al. 2019)). The data pre-processing scheme is: 1) We rank the users with the number of their log data and choose the 80% users least log data as the cold-start users (the value 80% is based on the Pareto Principle). 2) We randomly split the users by $7:1:2$ for training, validation, and testing. 3) We tick out the users when they either have blank or wrong features (including the ages are less than 10 or larger than 100 as well as the location features with garble) or the items they viewed are less than a threshold (two items for rating prediction and two clicked items for CTR prediction). 4) We separate the support and query sets for each user with a ratio of 80%:20% randomly. Details about the features we use, and the statistics can be found in Appendix D.1 & D.3.

Major and minor users separating. To testify whether our methods provide better prediction results for both minor users and major users, we need to define a criteria to split the major and minor users. However, as we mentioned above, finding the minor users are hard because there is not an exact method to precisely cluster the users by their features. Here we define a simple but not completely precise rule to *approximately* split the minor and major users: since distribution of the feature embeddings is the key to classify whether a user is minor or major user, we split the minor and major users by the following criteria: if the user with more than two of features which are in the top 30% largest number of feature values set, we regard it as the major user. Otherwise, it is the minor user. The major and minor users are 65% and 35% in MovieLens, 72% and 28% in BookCrossing, and 71% and 29% in the Taobao dataset.

Baselines. To validate our methods, we compare them with several State-Of-The-Art (SOTA) methods, including MeLU (Lee et al. 2019), Meta-SGD (Li et al. 2017), and transfer learning (Tan et al. 2018). 1) MeLU (Lee et al. 2019). MeLU is one o f the state-of-the-art gradient-based meta-learning methods in RS. Since it is very similar to other user-based meta learning (Finn, Abbeel, and Levine 2017; Bharadhwaj 2019), we use it as the standard gradient-based meta-learning method. 2) Meta-SGD (Li et al. 2017). Meta-SGD is an adaptive learning rate meta-learning method. Comparing with standard MAML using a fixed (inner) learning rate, it uses different learning rates for each parameter. But those adaptive parameters do not rely on the user embedding. We use it as a baseline to testify the performance of our embedding-based adaptive LR methods. 3) Transfer-learning (Tan et al. 2018). We choose the standard transfer-learning approach which has been widely adopted to address the user imbalance issue as the baseline. Specifically, we train the model with all users in training data and fine-tune the trained model in each user support set in the test data.

Network structures. For the AT-PAML, as shown in Fig. 2, it contains five modules, the embedding modules (including the embedding layers for each feature respectively, we use an FC layer to concatenate them together), the tree-based module, the learning-rate module, the decision module as well as the fusing module. The embedding modules are the standard embedding layers (Pennington, Socher, and Manning 2014). The tree-based module is what we discussed in Sec. . The LR module is two FC layers with ReLU as activation function. The decision module (except for the last layer) is based on the FC layer with ReLU as activation function. For the CTR prediction, the last layer is a two-dimension softmax layer. For the rating prediction, it is a real value without any activation function. The fusing layer is to concatenate all user and item embeddings together. More details can be found in Appendix D.2. For the REG-PAML and PAML, except that they do not have the tree module, other modules are similar.

Our methods. We validate our proposed methods, including the *Approximated Tree-based PAML*(AT-PAML) as well as *Regularizer PAML* (REG-PAML). For ablations, we implement vanilla PAML and REG-PAML with different γ values (we use $\gamma = 10^{-3}$ as default gamma value). We use Pytorch[5] as the deep learning library and we use the ADAM as the optimizer (Kingma and Ba 2014). All the experiments are done on a single GeForce GTX 1080 Ti. The production dataset is obtained by the company's recommendation pipeline. The package for the two-tailed student's t-test is from[6]. he running time for REG-PAML and PAML are about 2 hours, and for AT-PAML is about 4 hours on a single GTX 1080 GPU for MovieLens. For memory space, the space size for AT-PAML is 2.12GB, while that of the REG-PAML/ PAML is about 669MB in MovieLens.

Evaluation metrics. Following (Lee et al. 2019; Ren et al. 2019), we leverage different evaluation metrics to testify different aspects of our methods, including Mean Squared Error (MSE) and average Normalized Discounted Cumulative Gain (NDCG) for rating prediction results; as well as Area Under the Curve (AUC) and weighted Negative Entropy Loss (NEL) for CTR prediction results. To evaluate the performance difference between the minor and major users, we use the two-tailed student's t-test for statistical hypothesis test. The null hypothesis is that 2 independent samples (minor users and major users) have identical average (expected) values. If the p-value is high, indicating that the expected values of the minor users and major users are not different. On the contrary, when the p-value is low, it shows that the expected values of the minor users and major users are different. Therefore, when a method focuses on both minor and major users, it does **not** have a low p-value. More detailed definitions can be found in Appendix D.6.

Results and Discussions

Rating Prediction. We first analyse the rating prediction results. As shown in Table 2, our methods (AT-PAML & REG-PAML) outperform other SOTA methods in different metrics, especially for REG-PAML, which has reached 3 out of 6 best results. For the AT-PAML, it has two best results and two second-best results, implying that this method is also well-performing. From the Figs. 5 and 6 in Appendix D.7, we can find that REG-PAML has fewer outliners, indicating the robustness of REG-PAML. Moreover, in Fig. 5, Meta-SGD has some outliners with large values, which might be the reason that Meta-SGD does not perform well in the BookCrossing dataset. Moreover, both the AT-PAML and REG-PAML have lower third quartile comparing with SOTAs, showing that our methods can achieve good-enough performance for most of the cold-start users.

Minor users. We conduct experiments to show whether our methods help to improve the performance of minor users. As shown in Tab. 3, AT-PAML outperforms all the SOTAs in both minor users and major users significantly. Also, from the two-tailed student's t-test analysis, we find that the AT-PAML has the highest p-value, indicating that with a high probability the expected values for minor users and major users are similar. We also notice that SOTA meta-learning methods (MeLU and Meta-SGD) have relatively low p-values, revealing that the expected values for minor and major users are different with high probability. The results reveal that our methods are not only well-performing in the whole datasets but also focus on both the minor and major users than the

[5]https://pytorch.org/

[6]https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

	MovieLens-1M			BookCrossing		
	MSE ↓	Avg nDCG@3 ↑	Avg nDCG@5 ↑	MSE ↓	Avg nDCG@3 ↑	Avg nDCG@5 ↑
MeLU	1.451±0.022	**0.793± 0.002**	0.800± 0.001	*4.019* ±0.101	0.864±0.002	0.914±0.002
Meta-SGD	1.340± 0.054	0.773± 0.007	0.794± 0.007	5.197± 0.089	*0.867± 0.001*	*0.921± 0.002*
Transfer-Learning	*1.308±0.016*	0.778± 0.001	0.796± 0.002	4.522±0.376	0.862±0.010	0.919±0.005
AT-PAML (ours)	1.322±0.007	*0.788±0.003*	**0.806*±0.006**	*3.991±0.114*	0.852±0.001	**0.930*±0.002**
REG-PAML (ours)	**1.210*± 0.029**	0.779± 0.002	*0.803± 0.001*	**3.928*± 0.176**	**0.868± 0.001**	0.917± 0.001

Table 2: Comparison of different methods on the MovieLens and the BookCrossing datasets. The best results are highlighted in bold and the second-best results are in italic. Avg means average. The mean and standard deviation are reported by 3 independent trials. $*$ denotes statistically significant improvement over the best baseline method (measured by t-test with p-value< 0.05).

	MovieLens-1M		
	MSE (MAU)	MSE (MU)	P-value
MELU	1.413± 0.025	1.506± 0.021	0.071
Meta-SGD	1.300± 0.059	1.403± 0.045	0.024
TL	1.288± 0.016	*1.340± 0.013*	0.227
REG-PAML	**1.194**± 0.019	**1.236± 0.011**	0.312
AT-PAML	*1.284± 0.005*	1.382± 0.011	0.018

Table 3: Results of different MSEs for MAjor Users (MAU) and Minor Users (MU) on MovieLens. The best results are highlighted in bold and the second-best results are in *italic*. P-value is drawn by two-tailed student's t-test between the minor user MSEs and major user MSEs over the same methods.

	MovieLens-1M		
	MSE (MAU)	MSE (MU)	P-value
REG-PAML (10^{-5})	1.464± 0.032	1.601± 0.021	0.008
REG-PAML (10^{-4})	1.210± 0.026	1.255± 0.024	0.226
REG-PAML (10^{-3})	*1.196± 0.019*	**1.183± 0.024**	0.312
REG-PAML (10^{-2})	**1.194± 0.014**	*1.236± 0.011*	0.738
PAML	1.471± 0.036	1.569± 0.031	0.038
AT-PAML	1.284± 0.005	1.382± 0.011	0.018

Table 4: Ablations of different methods for MAjor Users (MAU) and Minor Users (MU) on MovieLens. The best results are highlighted in bold and the second-best results are in *italic*. Avg means average. P-value is drawn by two-tailed student's t-test between the minor user MSEs and major user MSEs over the same methods. The values in brackets are the values of γ.

SOTA meta-learning methods. One interesting phenomenon is that during the experiments, we notice a *failure case* shown in Fig. 8 in Appendix D.7, which reveals when the method is not capable to distinguish the minor users and the major users, it cannot perform well. These results also empirically show the significance of different LRs for different users. Also, the t-SNE (Maaten and Hinton 2008) visualization of user embeddings (Fig. 11 in Appendix D.7) show that our methods can distinguish the minor and major users. Tab. 4 reveals that γ affects the performance of both major and minor users.

Ablations. We do ablation studies to evaluate whether our proposed modules work. PAML with tree-based structures and regularizer ($\gamma = 10^{-3}$) perform better than the vanilla one under the MSE and the Avg nDCG@3 metrics. Also, AT-PAML and REG-PAML achieve lower values in the MSE metric for both minor and major users (Tab. 4 in Appendix D.7). An interesting phenomenon is that with the decrease of parameter γ, the MSE values become higher while the p-values become lower, meaning that the model performs worse and focuses more on major users. This result reveals that the regularizer does play an important role to achieve better performance and drive the model to focus on both minor and major users.

Conclusion

In this paper, we propose a novel personalized adaptive meta-learning method to address the user overfitting problem in cold-start user preference prediction challenge with three key contributions: 1) We are the first to introduce a personalized adaptive learning rate based meta learning approach to improve the performance of MAML by focusing on both the major and minor users. 2) We build an approximated tree-based method to store different users' features for fast calculation and a similarity-based method to find similar users as a reference to obtain accurate personalized learning rates for different users. 3) To reduce the memory usage, we propose a memory agnostic regularizer to reduce the space complexity and maintain the memorizing ability by learning. Experiments on MovieLens-1M, BookCrossing, and real-world production dataset reveal that our method outperforms the state-of-the-art methods dramatically for both the minor users and the major users.

Acknowledgements

This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Joint Research Institute (JRI), Nanyang Technological University, Singapore.

The authors would like to thank Suming Yu, Zhenyu Shi, Rundong Wang, Xinrun Wang, Feifei Lin, Aye Phyu Phyu Aung, Hanzao Chen, Ziwen Jiang, Yi Cao, Yufei Feng for their helps. The authors would also thank the team of MELU for their codes.

Due to the rule of AAAI, we a completed version with supplementary material on arXiv.org.

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 139–154.

Bharadhwaj, H. 2019. Meta-Learning for User Cold-Start Recommendation. In *IJCNN*, 1–8.

Chen, F.; Luo, M.; Dong, Z.; Li, Z.; and He, X. 2018. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876* .

Dong, M.; Yuan, F.; Yao, L.; Xu, X.; and Zhu, L. 2020. MAMO: memory-augmented meta-optimization for cold-start recommendation. In *KDD*, 688–697.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.

Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; and Levine, S. 2017. One-Shot Visual Imitation Learning via Meta-Learning. In *CORL*, 357–368.

Flennerhag, S.; Rusu, A. A.; Pascanu, R.; Yin, H.; and Hadsell, R. 2020. Meta-Learning with Warped Gradient Descent. In *ICLR*.

Friedman, J. H.; Bentley, J. L.; and Finkel, R. A. 1976. An algorithm for finding best matches in logarithmic time. *TOMS* 3: 209–226.

Gui, L.-Y.; Wang, Y.-X.; Ramanan, D.; and Moura, J. M. 2018. Few-shot human motion prediction via meta-learning. In *ECCV*, 432–450.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *AAAI*, 1725–1731.

Guo, Q.; Li, Z.; An, B.; Hui, P.; Huang, J.; Zhang, L.; and Zhao, M. 2019. Securing the deep fraud detector in large-scale e-commerce platform via adversarial machine learning approach. In *WWW*, 616–626.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *TIIS* 5(4): 1–19.

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

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* 114(13): 3521–3526.

Lee, H.; Im, J.; Jang, S.; Cho, H.; and Chung, S. 2019. MeLU: Meta-Learned user preference estimator for cold-Start recommendation. In *KDD*, 1073–1082.

Lee, Y.; and Choi, S. 2018. Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace. In *ICML*, 2933–2942.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-SGD: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* .

Luo, M.; Chen, F.; Cheng, P.; Dong, Z.; He, X.; Feng, J.; and Li, Z. 2020. MetaSelector: meta-learning for recommendation with user-level adaptive model selection. In *WWW*, 2507–2513.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9: 2579–2605.

Madotto, A.; Lin, Z.; Wu, C.-S.; and Fung, P. 2019. Personalizing dialogue agents via meta-learning. In *ACL*, 5454–5459.

Muja, M.; and Lowe, D. 2013. FLANN-Fast Library for Approximate Nearest Neighbors User Manual. *University of British Columbia* .

Muja, M.; and Lowe, D. G. 2014. Scalable nearest neighbor algorithms for high dimensional data. *PAMI* 36(11): 2227–2240.

Pan, F.; Li, S.; Ao, X.; Tang, P.; and He, Q. 2019. Warm Up Cold-start Advertisements: Improving CTR predictions via learning to learn ID embeddings. In *SIGIR*, 695–704.

Park, E.; and Oliva, J. B. 2019. Meta-curvature. In *NeurIPS*, 3309–3319.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Qu, Y.; Cai, H.; Ren, K.; Zhang, W.; Yu, Y.; Wen, Y.; and Wang, J. 2016. Product-based neural networks for user response prediction. In *ICDM*, 1149–1154.

Rajeswaran, A.; Finn, C.; Kakade, S. M.; and Levine, S. 2019. Meta-learning with implicit gradients. In *NeurIPS*, 113–124.

Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning. In *ICLR*.

Ren, K.; Qin, J.; Fang, Y.; Zhang, W.; Zheng, L.; Bian, W.; Zhou, G.; Xu, J.; Yu, Y.; Zhu, X.; et al. 2019. Lifelong Sequential Modeling with Personalized Memorization for User Response Prediction. In *SIGIR*, 565–574.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-Learning with Latent Embedding Optimization. In *ICLR*.

Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; and Liu, C. 2018. A survey on deep transfer learning. In *ICANN*, 270–279.

Vanschoren, J. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* .

Vartak, M.; Thiagarajan, A.; Miranda, C.; Bratman, J.; and Larochelle, H. 2017. A meta-learning perspective on cold-start recommendations for items. In *NeurIPS*, 6904–6914.

Xu, Z.; van Hasselt, H. P.; and Silver, D. 2018. Meta-gradient reinforcement learning. In *NeurIPS*, 2396–2407.

Yianilos, P. N. 1993. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, volume 93, 311–321.

Zhao, M.; Li, Z.; An, B.; Lu, H.; Yang, Y.; and Chu, C. 2018. Impression Allocation for Combating Fraud in E-commerce Via Deep Reinforcement Learning with Action Norm Penalty. In *IJCAI*, 3940–3946.

Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *WWW*, 22–32.