Rebuttal of *Online Item Cold-Start Recommendation with Popularity-Aware Meta-Learning*, KDD 2024, Submission 123

Anonymous Author(s)

REVIEWER XXHZ

Thank you for the thoughtful review of our work! Please allow us to address your concerns and answer the questions.

W1: Unclear classification of existing works.

In related works, we categorize the existing work into presentation-based and side-information-based following [1]. We consider these to represent two ways of obtaining the cold-start item parameters: The former (*e.g.* few-shot learning) learns to obtain the information from a few interaction samples between users and items, and the latter obtains the information from the existing users and items in the system, such as multi-modal information, attibutes, *etc.*. As you say, these two methods are not not opposed to each other, many novel approaches already combine few-shot learning with side-information in an effort to recommend cold-start items accurately.

W2.1: Inapplicable of existing meta-based works.

In existing meta-learning methods, each item or user is treated as a task to generate personalized parameters for them. Note that the parameters here include not only the embedding features but also network weight parameters (*e.g.* FC layers). In online systems, it's not feasible to fine-tune the personalized parameters for each new item, which is time and memory consuming.

W2.2&Q2: About side-information.

PAM is also an approach that incorporates side-information, we will clarify this in camera-ready version. Side-information that is manually labeled is completely unavailable online due to time consumption issues. In PAM, more information leads to better recommendations, yet the simplest side-information without manual labelling (*e.g.*, the category of an item) is feasible.

W3: Computation and storage costs of PAM.

As mentioned above, traditional cold-start few-shot learning methods require fine-tuning on each new item to generate personalized parameters, which is very time and storage consuming at a moment. In contrast, PAM generates network parameters specifically for cold-start items from the arriving data at moment t, which can be stored directly for recommendation at moment $t + 1$ without requiring any fine-tuning as new items arrive.

Q1: Difference between ID embedding and behavior-based embeddings.

Thank you for pointing this out, and we note that many reviewers have raised similar questions, and we will furthur exprain it more clearly in camera-ready version. According to the definition of behavior-based embedding, ID embedding is also a kind of behavior-based embedding. Due to the property that ID embedding naturally updates in the arriving interaction data, we consider that ID embedding most directly stores the information of items, and thus we utilize it as a supervisory signal for the cold-start task enhancer for training, and distinguish it from the rest of the behavior-based embeddings, such as embeddings of historical interacted users.

W4: Lack the comparision of latest works.

Thanks for pointing that out. The field of online recommendations and incremental updates to recommendation models is a small research field with less work. We have selected a latest work in 2023 IMSR[2] as a baseline for comparison. We conducted the experiment based on MovieLens dataset. The results of the experiment are displayed in Table 1:

Method	Recall@20	NDCG@20
IMSR	0.3631	0.1990
PAM-F	0.5966	0.3833

Table 1: PAM's performance compared with IMSR.

The results show that PAM also outperforms IMSR. just like online incremental updating methods such as SML, IMSR also has the property of relying on the initial state of the model, and pre-training thus plays an important role (line 695 in the article), which is also manifested in its poorer performance results in the scheme without pre-training.

[1] Ma et al. Cross-Modal Content Inference and Feature Enrichment for Cold-Start Recommendation. IJCNN '23

[2] Wang et al. Incremental Learning for Multi-Interest Sequential Recommendation. ICDE 2023

REVIEWER RE7C

Thank you for the thoughtful review of our work! Please allow us to address your concerns and answer the questions.

W1: Difference between offline and online scenarios.

Thank you for pointing out this, we noticed that this is an issue and we will add an image to camera-ready to clarify the difference between offline and online recommendations.

Briefly, the differences are as follows:

(1) Real-time: Since new users and items are constantly entering the system, the system needs to be able to recommend new items or generate recommendation parameters (whether these parameters are universal or personalized) in a small time and computational overhead.
(2) Streaming data: In an online system, the consumption data arrives as streaming data, and usually the data is used for training only once in an online system. The system is required to capture real-time changes in user interests using the streaming data.
(3) Complexity: online systems have a huge amount of data and a high frequency of requests, so online systems need to take into account all kinds of overheads, which makes many of the methods used in offline systems no longer applicable.

W2: Comparision of content-based models.

The reason we did not compare with content-based methods is that our proposed meta-learning-based training paradigm is **model-agnostic**, and therefore side-information-based methods can also be applied to PAM, and so we focus on comparisons with different online training paradigms without considering the differences introduced by the inputs. To verify this, we select a multi-modal side-information-based method SLMRec[1] on MovieLens dataset, apply PAM and present the results in Table 2.

Method	Recall@10	NDCG@10
SLMRec	0.2706	0.1868
PAM-F	0.3226	0.2611

Table 2: PAM's performance compared with SLMRec.

The results show that applying PAM in a information-based approach can further enhance its effectiveness in online systems.

Q: Difference between ID embedding and behavior-based embeddings.

Thank you for pointing this out, and we note that many reviewers have raised similar questions, and we will furthur exprain it more clearly in camera-ready version. According to the definition of behavior-based embedding, ID embedding is also a kind of behavior-based embedding. Due to the property that ID embedding naturally updates in the arriving interaction data, we consider that ID embedding most directly stores the information of items, and thus we utilize it as a supervisory signal for the cold-start task enhancer for training, and distinguish it from the rest of the behavior-based embeddings, such as embeddings of historical interacted users.

REVIEWER OBKX

Thank you for the thoughtful review of our work! Please allow us to address your concerns and answer the questions.

W1: Confusing definitions of the symbols.

Thank you for pointing this out, other reviews have also mentioned similar issues, we will fix all symbol definitions and usage issues in the camera-ready version.

W2&Q1: Different notation between article and figure

Thanks for pointing this out, it's one of the problems we have when writing. In the figure of cold-start enhancer, we identify ID embedding separately to indicate its specificity, and we use ID embedding as a supervised signal for self-supervised learning in the enhancer. And in the writing of the article, we consider ID embedding as a part of behavior-based embedding, so in practice, we concatenate ID embedding, other behavior-based embedding, and content-based embedding together.

W3&Q2: Unnecessity of fine-tuning.

In the article, we mentioned the advantage that PAM does not require fine-tuning (line 130). Here, we are describing the fact that the PAM method does not require real-time fine-tuning of new items at the moment they enter the system, which brings additional fine-tuning time consuming. PAM generates network parameters specifically for cold-start items from the arriving data at moment t, which can be stored directly for recommendation at moment $t+1$ without requiring any fine-tuning as new items arrive (This can be seen in the pseudo-code at line 594).

Q3: Larger improvements of Tmall dataset.

We believe that this phenomenon is related to the processing of the datasets: the user-item interaction data of the MovieLens and Yelp datasets contained rating data, so we used ratings of 4 and above as positive samples and ratings of 3 and below as negative samples. In contrast, the data in the Tmall dataset did not contain rating information, so we randomly sampled one user for each item of consumption data as a negative sample, and it is likely that this data processing caused the traditional online methods to perform poorly on the Tmall dataset. The results are displayed in Table 3.

Method	Recall@20	NDCG@20
PF	0.0996	0.0386
PAM-M	0.2189	0.0896

Table 3: PAM's performance compared with PF, on partial randomly negatively sampled MovieLens dataset.

From the results, we can see that although the smaller dataset results in poorer recommendations, the baseline method in the randomly negatively sampled MovieLens dataset is still more different from PAM, with a magnitude similar to that of the Tmall dataset.

[1] Tao et al. Self-Supervised Learning for Multimedia Recommendation. TMM '22

REVIEWER AI1I

Thank you for the thoughtful review of our work! Please allow us to address your concerns and answer the questions.

W1: Missing of related works.

Thanks for pointing out. Meta-learning, the primary method used in this paper, plays an important role in cold-start recommendations. Some citations of related work are missing, and we will supplement them in detail in camera-ready version. As a relatively novel meta-learning cold-start approach at the time, LWA&NLBA[1] generate personalized parameters based on users' historical behaviors, and demonstrated strengths on Twitter data. M2EU[2] differed from previous meta-learning schemes by focusing on user-side enhancement, which was achieved by selecting similar users as features for meta-learning user cold-start. However, as mentioned in line 101 of the article, such schemes that consider new users or new items as tasks encounter the problem of time-consuming fine-tuning in online systems.

W2: Comparing with cold-start online recommendation methods.

To the best of our knowledge, the cold-start problem in online recommendation scenarios is still relatively understudied, and the only solution based on few-shot learning is FORM[3].

However, as mentioned in line 212 of the paper, due to its property of generating personalized parameters for each new user, FORM can only be applied in online scenarios such as advertisement recommendation with low frequency of parameter updates. In scenarios such as short video recommendation where a large number of new items enter the system, it suffers from computationally time-consuming problems, and thus cannot be well applied to online recommender systems.

In scenarios such as short video recommendation where a large number of new items enter the system in real time, it suffers from computational time-consuming problems, and thus FORM cannot be well applied to online recommender systems. Therefore, we did not select it for comparison.

W3: Comparing partial baselines of popular items.

Since the performance of popular items was not the focus of the article and due to space constraints, we did not give all baseline comparisons. For the detailed results, please refer to Table 4.

Methods	Recall@20	NDCG@20
PF	0.4285	0.2850
s^2Meta	0.4287	0.2861
IncCTR	0.4247	0.2838
SML	0.4049	0.2643
ASMG	0.3999	0.2588
MeLON	0.5139	0.3173
PAM	0.4840	0.3198

Table 4: PAM's performance on popular items of MovieLens.

From the results we can see similar results to those in the paper (lines 790-799), with the rest of the baseline showing an increase in performance on popular items compared to cold-start items, while PAM shows a decrease in performance, reflecting its advantages in cold-start item recommendations.

[1] Vertak et al. A Meta-Learning Perspective on Cold-Start Recommendations for Items, NIPS '17

[2] Wu et al. M2EU: Meta Learning for Cold-start Recommendation via Enhancing User Preference Estimation, SIGIR '23

[3] Sun et al. FORM: Follow the Online Regularized Meta-Leader for Cold-Start Recommendation. SIGIR '21

REVIEWER XWYL

Thank you for the thoughtful review of our work! Please allow us to address your concerns and answer the questions.

W1: Technical contributions are not clear enough.

We would highlight our technical contributions as follows:

(1) We proposed a new paradigm for meta-learning applied to the online recommendation cold-start problem, where tasks are partitioned in terms of popularity for items and trained using a meta-learning approach. Instead of focusing on the generalization performance of meta-learning on new tasks, this novel task partitioning approach focuses on the ability of meta-learning to generate parameters individually while sharing information across different tasks.
(2) We proposed a novel way of simulating cold-start items using popular items, and the unique cold-start task enhancer based on data augmentation and self-supervised learning is furthur designed. By concatenating the earlier behavior-based embeddings of popular items and the current content-based embeddings, a cold-start item at the current moment is simulated and used for subsequent training.

W2.1&Q1.1: Not compare with FORM.

As described in line 212 of the paper and in **W2** of rebuttal of Reviewer Ai1i, the FORM approach requires personalized recommendation parameters to be generated for each new user, and such time-consuming fine-tuning is unacceptable in a large online system where many new items enter the system streamingly, which is likewise why the traditional meta-learning approach is not applicable to online systems.

W2.2&Q1.2: Computation and storage costs of PAM.

As mentioned above, traditional cold-start few-shot learning methods require fine-tuning on each new item to generate personalized parameters, which is very time and storage consuming at a moment. In contrast, PAM generates network parameters specifically for cold-start items from the arriving data at moment t, which can be stored directly for recommendation at moment $t + 1$ without requiring any fine-tuning as new items arrive.

W3: Symbol issues.

Thank you for pointing this out, the symbol issue you described is the result of an oversight in our work and we will fix the issue you mentioned in camera-ready version. For ⑤, it do have the same meaning in InfoNCE.

Q2: PAM's model agnosticism.

Since the proposed PAM is an online cold-start meta-learning training process, it is model-agnostic. To verify the effectiveness of PAM on other base models, we selected DIN[1] as the base model (Embedding&MLP in the article). The results are summarized by Table 5:

Method	Recall@5	NDCG@5
PF	0.6474	0.4876
PAM-M	0.8629	0.7208

Table 5: PAM's performance compared with PF with base model DIN.

As can be seen from the results, PAM still has a similar magnitude of improvement compared to the baseline under different base models.

Q3: Rationale of dividing embeddings.

We divide the embeddings for two reasons:

(1) These types of embeddings play different roles in the recommendation of cold-start and popular items, as can be seen from the experiments in Section 5.4: content-based embedding is more important for cold-start items, whereas popular items value behavioral-based embedding more.
(2) We distinguish these two kinds of embeddings to describe the data argumentation for cold-items (**W1** 2.). We propose a **novel** data argumentation for cold-start items, in which content embeddings of cold items are reused together with behavior embeddings from subsequent interations.

[1] Zhou et al. Deep Interest Network for Click-Through Rate Prediction. KDD '18.

Reviewer myQM

Thank you for the thoughtful review of our work! Please allow us to address your concerns and answer the questions.

Q1: Categotization of cold-start problems.

As a few-shot learning method, meta-learning[1] aim to learn the ability to fine-tune a personalized parameters from a small amount of samples. Therefore, samples is necessary for a meta-learning method, without which there is no fine-tuning to generate personalized parameters, and thus the meta-learning method is not able to handle strict cold-start scenarios.

In the start of the system, all items are cold like you said. However, item's popularity gradually increased over time, and after entering the testing phase, we calculated metrics for all cold-start items, not all strictly cold-start items with 0 popularity. Besides, our definition for cold-start items is based on the popularity threshold, which is a common way of defining cold-start items (e.g. "divide the cold-start set with few interactions" in [1] and "As for the items, we regard items with less than 10 ratings as cold items" in [2]). Our evaluation metrics are also suitable and wide applied to cold-start items, as it evaluate the rank of candidate users to an item.

[1] Ma et al. Cross-Modal Content Inference and Feature Enrichment for Cold-Start Recommendation. IJCNN '23

[2] Dong et al. MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation. KDD '20

Q2: Small fraction of cold-start samples.

This is exactly the long-tail effect of cold-start items. What the paper describes is that the **consumption data** generated by cold-start items makes up a small portion of the total data, while the cold-start **items** make up the majority of the items. In other words, popular items make up only a minority of the total, but produce the majority of the consumption traffic.

Q3: Same features of same popularity.

The formulation here is not clear enough, what we want to express is that items of the same popularity are recommended using the same network weight parameters (different for items of different popularity), and the embedding parameters are shared between tasks. We'll fix it in camera-ready.

Q4: Task attribution of u-i pairs.

A u-i pair will only belong to a single task. Here the piece-wise function means that pairs are incrementally divided into tasks by popularity p: pairs with a $p \in (50, 200]$ go to task 2, those with $p \in (200 - 1000]$ to task 3, *etc.*.

Q5: Support and query sets.

Support sets and query sets are concepts in meta-learning. As mentioned above, meta-learning requires a small number of samples in a task for generating personalized parameters, these samples are support set. The remaining samples form the query set and are used to compute the loss on the generated parameters. In PAM, we compute the parameters on each popularity task via the support set, and the query set is used to compute the loss and optimize the initialization parameters.

Q6: Different characteristics of tasks.

In our setting, we consider the data in different tasks have different characteristics, and this is a prerequisite to divide the tasks to learn through a meta-learning scheme. For example, ID embeddings of cold-start items are somehow useless; whereas those of popular items are significant. So we definitely convince that the recommendations of items with different popularity are truly different tasks.

[1] Finn et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, ICML '17