Cold & Warm Net: Addressing Cold-Start Users in Recommender Systems

Xiangyu Zhang *[1], Zongqiang Kuang *[1], Zehao Zhang[1,2], Fan Huang †[1], and Xianfeng Tan[3]

[1] Tencent, Shenzhen, China
{altairzhang,devinkuang,sinohuang}@tencent.com
[2] Tsinghua University, Beijing, China
zhangzeh20@mails.tsinghua.edu.cn
[3] Tencent, Beijing, China
victan@tencent.com

Abstract. Cold-start recommendation is one of the major challenges faced by recommender systems (RS). Herein, we focus on the user cold-start problem. Recently, methods utilizing side information or meta-learning have been used to model cold-start users. However, it is difficult to deploy these methods to industrial RS. There has not been much research that pays attention to the user cold-start problem in the matching stage. In this paper, we propose Cold & Warm Net based on expert models who are responsible for modeling cold-start and warm-up users respectively. A gate network is applied to incorporate the results from two experts. Furthermore, dynamic knowledge distillation acting as a teacher selector is introduced to assist experts in better learning user representation. With comprehensive mutual information, features highly relevant to user behavior are selected for the bias net which explicitly models user behavior bias. Finally, we evaluate our Cold & Warm Net on public datasets in comparison to models commonly applied in the matching stage and it outperforms other models on all user types. The proposed model has also been deployed on an industrial short video platform and achieves a significant increase in app dwell time and user retention rate.

Keywords: recommender systems · cold-start · Cold & Warm Net.

1 Introduction

As online information in social media and e-commerce platforms grows explosively, large-scale recommender systems (RS) [11] play an important role in solving the problem of information overload for users. Industrial RS [12] typically contains two stages: matching and ranking. In the matching stage, thousands of items potentially relevant to the user's interests are retrieved from a large-scale candidate pool, which is required to quickly find as many items that satisfy the

* indicates equal contributions.

† indicates corresponding author.

user's interests as possible. after that, the ranking stage is performed to precisely predict the probability of a user interacting with an item.

Recently, the matching stage in recommenders [3] has been paid increasing attention. Various methods have been applied in the matching stage. Conventional collaborative filtering (CF) method [11] depends on the similarity of interacted items between users for the recommendation. State-of-the-art methods based on reinforcement learning [14], graph network [15] and Multi-Interest network [1] focus on user behavior sequence representation owned solely by active users with much interaction behavior. However, these models fail to learn high-quality embeddings for the cold-start users with sparse interaction behavior.

Faced with the cold-start problem, side information [16] has been used to provide a better recommendation. However, methods utilizing side information can only benefit part of the users. There are some attempts [4] to introduce meta-learning into recommender systems, which requires the computation of second-order gradients. Therefore, it cannot meet the scalability required by the matching stage of real-world recommendation scenarios. Scalability is the ability to process large-scale information efficiently.

In this paper, modeling cold-start users in the matching stage is our purpose. The core mission of modeling cold-start users is to learn collaborative information between old and cold-start users and train models effectively. Herein, we propose an implicit embedding net based on cold-start and warm-up experts which solves the problems mentioned above efficiently. According to the frequency of interaction, users can be briefly divided into three categories: cold-start users, warm-up users, and active users. The category of users is dynamically changing with the accumulation of interests and behavior, so it is not suitable to use the same strategies for different types of users. Our embedding net based on cold-start and warm-up experts models the dynamic process of cold-start users towards warm-up and active users without compulsory strategies. Overall, the main contributions of this work can be concluded as follows.

- Dynamic handling of samples. With the division of cold-start and warm-up experts, our Cold & Warm Net can dynamically represent the users' interest in cold-start and warm-up phases. Through the gate network, the net can automatically incorporate the results from two experts according to user type and user state. Cold-start and warm-up experts can learn the differences between samples.

- Flexible teacher selector. Dynamic knowledge distillation is applied to cold-start and warm-up experts using a teacher selector. The selector chooses the right teacher for the cold-start expert according to prediction accuracy. By applying dynamic knowledge distillation, it avoids the underfitting of the cold-start expert while preventing the assimilation of two experts after training, which enables learning sufficient information from cold-start users.

- Explicit modeling of behavior bias. Using a bias net to model the behavior bias of cold-start users explicitly. By utilizing mutual information, user features highly relevant to user behavior are selected. With the combination of the similarity score from the original net and the bias score from the bias net, information hidden behind user behavior is thoroughly considered.

2 RELATED WORK

In this section, we review the two-tower models based on embedding which are applied in the matching stage and models targeting the cold-start problem.

2.1 Two-tower models in the matching stage

One of the challenges faced by RS in the matching stage is that the representations of users and items are not in the same latent space. Models based on embedding learn how to map the sparse user and item vectors in high-dimensional space into dense vectors in low-dimensional space and calculate the inner product or cosine similarity between user and item vectors to obtain a relevance score. The idea of deep learning has been applied in the two-tower models. DSSM [8] is a well-known two-tower model utilizing two deep neural networks that map queries and documents into a common space to achieve better search satisfaction. YouTube [3] proposes a deep candidate generation model which can effectively learn the embedding of user and item features. [1] exploits both user profile and behavior information for candidate matching. To tackle sample bias in the matching stage. [7] uses random sampling to acquire negative samples, which successfully bridges the gap in data distribution between training and testing. However, these two-tower models all require close user-item interaction and are incompetent to model cold-start users with rare interaction behavior.

2.2 Cold-start problem

The cold-start problem has been one of the long-standing challenges faced by RS. Traditional methods rely on side information to alleviate the cold-start problem, e.g. utilizing social networks among users [16]. Transfer learning-based method [6] is also used to deal with the cold-start problem. [4] uses meta-learner to generate cold-start user embedding. However, most of the existing works focus on the cold-start problem in the ranking stage. [16] applies an attention mechanism in multi-channel matching to extract useful feature interactions. [2] uses an extra adversarial network to generate cold-start item embedding. [5] simulates the cold-start scenarios from the users/items with sufficient interactions and takes the embedding reconstruction as the pretext task. To our knowledge, we are the first to propose an embedding net that dynamically models different types of users in the matching stage.

3 METHOD

3.1 Problem description

The objective of the matching stage for RS is to retrieve Top K relevant items from a large-scale candidate pool $\mathcal{I}$ for each user $u \in \mathcal{U}$. To achieve this target, a matching model is built. The input of model is a tuple $(\mathcal{X}_u, \mathcal{X}_i)$, where $\mathcal{X}_u$ denotes user features and $\mathcal{X}_i$ denotes item features. Modeling cold-start users is tough due to the lack of behavior. The core task of Cold & Warm Net is to learn a function that can map original features into user representations. The function can be formulated as:

$$\overrightarrow{\boldsymbol{e}}_u = f_{user}\left(\mathcal{X}_u\right) \tag{1}$$

where $\overrightarrow{\boldsymbol{e}}_u \in \mathbb{R}^{1\times d}$ denotes the representation vector of user u, d the dimensionality. In addition, the representation vector of target item i is obtained by a function:

$$\overrightarrow{\boldsymbol{e}}_i = f_{item}\left(\mathcal{X}_i\right) \tag{2}$$

where $\overrightarrow{\boldsymbol{e}}_i \in \mathbb{R}^{1\times d}$ denotes the representation vector of item i. Finally, The top K relevant items are retrieved according to the scoring function:

$$f_{score}\left(\overrightarrow{\boldsymbol{e}}_u, \overrightarrow{\boldsymbol{e}}_i\right) = \overrightarrow{\boldsymbol{e}}_u \cdot \overrightarrow{\boldsymbol{e}}_i \tag{3}$$

3.2 Cold & Warm Net

As shown in Figure 1, our Cold & Warm Net consists of two subnets: original cold & warm net and bias net. The original cold & warm net uses user features $\mathcal{X}_u$ and item features $\mathcal{X}_i$ as input while the bias net takes bias features $\mathcal{X}_b$ as input. We divide user features into two categories: user profile features $\mathcal{X}_{up}$(e.g., gender and age) and user action features $\mathcal{X}_{ua}$(also called user behavior). Taking different user features as input, we attain output embedding: user profile embedding $\overrightarrow{\boldsymbol{e}}_{up} \in \mathbb{R}^{1\times d}$ and user action embedding $\overrightarrow{\boldsymbol{e}}_{ua} \in \mathbb{R}^{1\times d}$. Besides, user group embedding $E_{ug} \in \mathbb{R}^{m\times d}$ is provided as prior information for all users. Firstly, We use a pre-trained model for getting all active-user embeddings. Then, taking all active-user embeddings as input, the k-means algorithm is used to attain m clusters. Finally, we use average pooling to aggregate the embeddings of each cluster for generating E_{ug}. The three parts are defined as U_a, U_b and U_c, which are fed into user cold & warm embedding layer to generate user embedding $\overrightarrow{\boldsymbol{e}}_u$. Along with item embedding $\overrightarrow{\boldsymbol{e}}_i$, similarity score y_{sim_score} is calculated as follows.

$$y_{sim\_score} = \frac{\overrightarrow{\boldsymbol{e}}_u \cdot \overrightarrow{\boldsymbol{e}}_i}{\|\overrightarrow{\boldsymbol{e}}_u\|\|\overrightarrow{\boldsymbol{e}}_i\|} \tag{4}$$

The similarity score y_{sim_score} from the original cold & warm net and the bias score y_{bias_score} from the bias net constitute our final output:

$$y = sigmoid(y_{sim\_score} + y_{bias\_score}) \tag{5}$$

User cold & warm embedding layer As shown in Figure 2, our user cold & warm embedding layer is mainly composed of two experts: the cold-start expert and the warm-up expert. To extract and learn valid information that matches the current user, we use the attention mechanism to retrieve the prior information

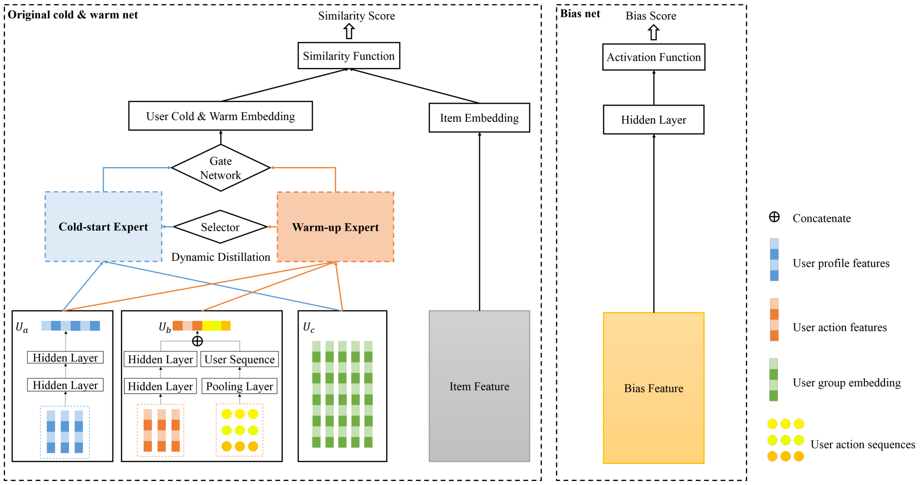

Fig. 1. Cold & Warm Net.

from E_{ug} which contains all user group pre-trained embedding. Cold-start expert takes input from U_a and U_c which contain user profile information and user group information. Attention embedding takes prior user group information to assist in modeling cold-start users, which can be formulated as:

$$\overrightarrow{e}^{a}_{cold} = softmax(\frac{\overrightarrow{e}_{up}E_{ug}^{T}}{\sqrt{d}})E_{ug} \tag{6}$$

where $\overrightarrow{e}^{a}_{cold} \in \mathbb{R}^{1\times d}$ means pre-trained embedding retrieved from E_{ug} using attention mechanism [9]. The output embedding of cold-start expert $\overrightarrow{e}_{cold}$ is:

$$\overrightarrow{e}_{cold} = mlp(\overrightarrow{e}_{up}; \overrightarrow{e}^{a}_{cold}) \tag{7}$$

Where $\overrightarrow{e}_{cold} \in \mathbb{R}^{1\times d}$. The warm-up expert takes input from U_a, U_b and U_c. It is designed for users who possess user profile features $\mathcal{X}_{up}$ and user action features $\mathcal{X}_{ua}$. Taking $\mathcal{X}_{up}$ and $\mathcal{X}_{ua}$ as input, we attain embedding $\overrightarrow{e}_{ut} \in \mathbb{R}^{1\times d}$. Through the assistance of attention anchor embedding, the output embedding of warm-up expert $\overrightarrow{e}_{warm}$ is defined as follows.

$$\overrightarrow{e}^{a}_{warm} = softmax(\frac{\overrightarrow{e}_{ut}E_{ug}^{T}}{\sqrt{d}})E_{ug} \tag{8}$$

$$\overrightarrow{e}_{warm} = mlp(\overrightarrow{e}_{ut}; \overrightarrow{e}^{a}_{warm}) \tag{9}$$

where $\overrightarrow{e}_{warm} \in \mathbb{R}^{1\times d}$. A gate network is used to produce weights for experts.

$$w_{cold}, w_{warm} = f_{weight}(\mathcal{X}_{us}) \tag{10}$$

where $\mathcal{X}_{us}$(such as login state and active degree) denotes state feature. The output user cold & warm embedding is:

$$\overrightarrow{e}_{u} = w_{cold} \cdot \overrightarrow{e}_{cold} + w_{warm} \cdot \overrightarrow{e}_{warm} \tag{11}$$

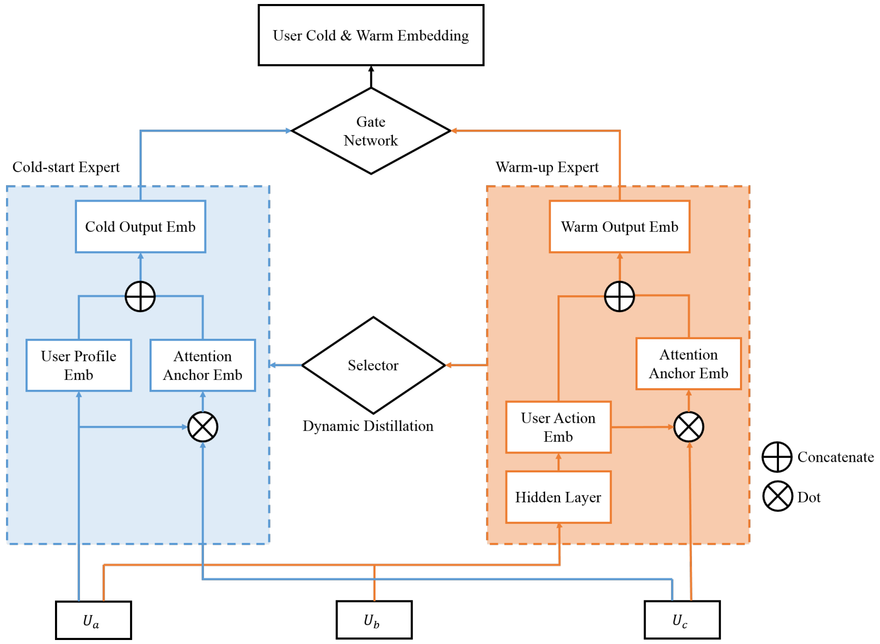

Fig. 2. User cold & warm embedding layer.

The above expression can be understood as a weighted summation of cold-start expert and warm-up expert.

Dynamic knowledge distillation With a mix of experts, cold-start expert suffers from underfitting due to limited information from cold-start users. The reason is that the warm-up expert learns better for active users which own rich behavior features. To avoid underfitting of the cold-start expert, we invent dynamic knowledge distillation(DKD) to distill information from the warm-up expert to the cold-start expert. Binary cross entropy is selected as our loss function. The major loss function L:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right) \tag{12}$$

where N denotes the number of samples, y_i is the label for each sample. $\hat{y}_i$ denotes the predicted result. L is the loss function except for the cold-start expert. Besides, The auxiliary loss function L_d from dynamic knowledge distillation:

$$L_d = -\frac{1}{N}\sum_{i=1}^{N} l_d \tag{13}$$

where l_d is the loss function of DKD for each sample, which is shown in Algorithm 1. For each sample, we compare the cross entropy loss of cold-start expert

Algorithm 1: Dynamic knowledge distillation.

```
foreach sample from batch samples do
    Calculate $l(\hat{y_i}^c, y_i)$, $l(\hat{y_i}^w, y_i)$
    if $l(\hat{y_i}^c, y_i) \leq l(\hat{y_i}^w, y_i)$ then
        $l_d = l(\hat{y_i}^c, \hat{y_i}^w)$;
    else
        $l_d = 0$;
    end
end
```

$l(\hat{y_i}^c, y_i)$ and warm-up expert $l(\hat{y_i}^w, y_i)$. If $l(\hat{y_i}^c, y_i) \leq l(\hat{y_i}^w, y_i)$, it indicates that output from the cold-start expert taught by the label is better and there is no need to distill information from the warm-up expert. Otherwise, the cold-start expert taught by the label is not enough and it is necessary to learn from the warm-up expert. $\hat{y_i}^c$ denotes the predicted label for the cold-start expert and $\hat{y_i}^w$ is the predicted label for the warm-up expert. The teacher for knowledge distillation is dynamic to make sure the cold-start expert could learn effective information from the teacher. The loss function of the cold-start expert is defined as L_o.

$$L_o = L + \alpha \cdot L_d \tag{14}$$

where α is hyperparameter. α determines the strength of distillation from the warm-up expert.

Bias net. To solve the behavior bias when modeling cold-start users, an additional bias net is applied. The reason why bias net is effective is that the behavior bias is large between cold-start users and active users in real-world recommendation scenarios. For example, active users have several times more click rates than new users, so a bias net is indispensable for describing the bias. We aim to find a set of user features $\mathcal{X}_b$ which are highly relevant to user behavior and feed them into the bias net to get y_{bias_score}. To mine the key features of the behavior bias, mutual information is used to measure the relevance between user features and behaviors. We select the top β relevant features as our bias features $\mathcal{X}_b$. The output bias score y_{bias_score} is:

$$y_{bias\_score} = f_{bias}(\mathcal{X}_b) \tag{15}$$

y_{bias_score} is used to characterize the bias of target behaviors of people.

4 Experiments

4.1 Offline Evaluation

In this section, we compare our Cold & Warm Net with existing methods applied in the matching stage in terms of recommendation accuracy on two datasets.

Table 1. Statistics of the datasets.

Dataset	# User	# Items	# Interaction
MovieLens 1M	6040	3706	1,000,209
Little-World	433,549	406,140	15,200,286

Datasets and experimental setup Two datasets are chosen to evaluate the recommendation performance in the matching stage. One is MovieLens 1M[4], which is one of the most common datasets used for recommendations. Also, we collect a real-world large-scale dataset from the Little-World[5]. The statistics of datasets are shown in Table 1. Hit rate (HR) and Normalized discounted cumulative gain (NDCG) are adopted as the main metric to evaluate the performance of models in the matching stage, define as:

$$HitRate@K = \sum_{(u,i)\in T} \frac{I(target\, items\, occur\, in\, topK)}{|T|} \tag{16}$$

$$NDCG@K = \frac{1}{|U|} \sum_{u\in U} \frac{DCG_k^u}{IDCG_k^u} \tag{17}$$

$$DCG_k^u = \sum_{r=1}^{k} \frac{2^{R_{ur}} - 1}{\log_2(1+r)} \tag{18}$$

where T denotes the test set containing pair of user and item and I denotes the indicator function. R_{ur}, U, and $IDCG_k^u$ are the real rating of user u for the r-th ranked item, a set of users in the test data and the best possible DCG_k^u for user u, respectively. Specially, in the matching stage, $R_{ir} \in \{0, 1\}$.

Comparing methods The following methods widely applied in the matching stage in industry RS are used to compare with our Cold & Warm Net.

- FM [13] A model that utilizes the feature vectors of query and item and feeds them into FM layer.
- YouTubeDNN [3] One of the most commonly used models in the recommendation industry which applies deep neural network to generate item and user embedding.
- DSSM [8] A popular model applied in the matching stage which makes use of rich content features of user and item.

[4] https://grouplens.org/datasets/movielens/1m/

[5] Little-World is a short video platform in QQ, which allows users to create and share micro-videos. Note that we anonymize the data and conduct strict desensitization processing. The data may be made public in the future.

Table 2. Performance comparison of different models in terms of HR and NDCG

(a)Results on full users

Models	MovieLens 1M				Little-World			
	HR@50	HR@100	NDCG@10	NDCG@50	HR@50	HR@100	NDCG@10	NDCG@50
FM	0.0969	0.1922	0.0099	0.0262	0.0513	0.0754	0.0100	0.0173
YouTubeDNN	0.1399	0.2548	0.0153	0.0378	0.0862	0.1461	0.0110	0.0245
DSSM	0.2013	0.3151	0.0226	0.0520	0.0913	0.1511	0.0116	0.0260
Mind	0.2019	0.3322	0.0238	0.0612	0.0917	0.1530	0.0118	0.0262
UMI	0.2348*	0.3697*	0.0305*	0.0664*	0.0920*	0.1546*	0.0119*	0.0270*
Cold & Warm	**0.2556**	**0.3932**	**0.0369**	**0.0750**	**0.1122**	**0.1792**	**0.0155**	**0.0325**
%improve.	8.86%	6.35%	20.98%	12.95%	21.95%	15.91%	30.25%	20.37%

(b)Results on cold-start users

Models	MovieLens 1M				Little-World			
	HR@50	HR@100	NDCG@10	NDCG@50	HR@50	HR@100	NDCG@10	NDCG@50
FM	0.1568	0.2953	0.0211	0.0461	0.0710	0.1047	0.0147	0.0242
YouTubeDNN	0.2444	0.3849	0.0236	0.0639	0.1088	0.1768	0.0138	0.0311
DSSM	0.3666*	0.5356*	0.0657*	0.1173*	0.1109*	0.1775*	0.0159*	0.0326*
Mind	0.3259	0.4807	0.0485	0.1008	0.1074	0.1771	0.0132	0.0300
UMI	0.3360	0.4705	0.0493	0.1002	0.1103	0.1671	0.0143	0.0281
Cold & Warm	**0.4094**	**0.5866**	**0.0678**	**0.1265**	**0.1435**	**0.2200**	**0.0215**	**0.0418**
%improve.	11.67%	9.52%	3.19%	7.84%	29.39%	23.94%	35.22%	28.22%

– Mind [10] the first attempt in representing a user with multiple interest vectors via deep neural network structures.

– UMI [1] State-of-the-art model that relies on multiple user interest representations to achieve superior recommendation accuracy.

The above models are implemented by Tensorflow and Faiss is used to retrieve the top K items from the item pool. The embedding dimension and batch size are set to 32 and 256 respectively for all models. To ensure a fair comparison, for each model, hyperparameters are tuned to achieve the best performance. For Cold & Warm Net, hyperparameters α, β are set to 5e-2 and 10 respectively.

Experimental results Table 2 summarizes the performance of Cold & Warm Net in comparison with different models applied in the matching stage in terms of HR@K(K=50, 100) and NDCG@K(K=10, 50). Obviously, Cold & Warm Net achieves the best recommendation performance among all models on different user categories. FM performs worst among all models revealing the power of deep learning. UMI and Mind which utilize multiple interest representations of user generally performs better than YouTubeDNN which only uses single-user interest representation. UMI performs better than Mind due to exploiting both user profile and behavior information for candidate matching. However, Mind and UMI perform worse than the DSSM model for cold-start users, which may be because cold-start users lack abundant interests. Results on two types of users justify the performance of Cold & Warm Net on different types of users, effectively solving the user cold-start problem faced in recommender systems.

Table 3. Ablation study of Cold & Warm Net.

Models	MovieLens 1M				Little-World			
	Full users		Cold-start users		Full users		Cold-start users	
	HR@100	NDCG@10	HR@100	NDCG@10	HR@100	NDCG@10	HR@100	NDCG@10
Cold & Warm	**0.3932**	**0.0369**	**0.5866**	**0.0678**	**0.1792**	**0.0155**	**0.2200**	**0.0215**
w/o DKD	0.3869	0.0318	0.5540	0.0581	0.1703	0.0144	0.1987	0.0190
w/o Bias Net	0.3930	0.0367	0.5682	0.0632	0.1761	0.0147	0.2122	0.0203

Table 4. Influence of Dynamic Knowledge Distillation on weights (w_{cold}, w_{warm}).

Metrics	Cold-start expert	Warm-up expert
Weights (w/o DKD)	0.0410	0.9590
Weights (DKD)	0.3140	0.6860

Table 3 summarizes the result of the ablation study. It is conducted to evaluate the contribution of the dynamic knowledge distillation(DKD) module and bias net. DKD and bias net designed for cold-start users contribute mainly to solving the problem of modeling cold-start users. For cold-start users, applying DKD brings an increase of 5.88% and 10.72% in HitRate@100 on two datasets while adding bias net brings an increase of 3.24% and 3.68% in HitRate@100. The major boost from DKD may be due to the fact that the cold-start expert learns better user representation with the assistance of the warm-up expert.

4.2 Analysis of Dynamic Knowledge Distillation

In this section, the influence of DKD has been analyzed based on the Little-World dataset. AUC is chosen as the evaluation metric. As shown in Table 4, by applying dynamic knowledge distillation, it greatly improves the weight of cold-start expert w_{cold} from 0.0410 to 0.3140, which allows the cold-start expert to learn sufficient information either from warm-up expert or label and avoids underfitting of cold-start expert. It can be seen from Table 5 that after applying DKD, the AUC of the cold-start expert increases obviously, which demonstrates the effect of DKD on enabling sufficient training. The AUC of the warm-up expert decreases on the train set because DKD reduces losses flowing to the warm-up expert. Therefore, the major contribution of AUC comes from sufficient learning of the cold-start expert. Improved AUC of the whole model on the test set also justifies the influence of DKD on cold-start users.

4.3 Online Experiment

Finally, we deploy Cold & Warm Net in the real-world recommending scenario of Little-World. User retention rate (URR) and app dwell time (APT) are used as

Table 5. Influence of Dynamic Knowledge Distillation on AUC.

Metrics	w/o DKD			DKD		
	Cold-start expert	Warm-up expert	Whole	Cold-start expert	Warm-up expert	Whole
full users	0.5770	0.9255	0.9267	**0.8772**	0.8993	**0.9279**
cold-start users	0.5675	0.7279	0.7281	**0.7384**	0.7434	**0.7548**

Table 6. Online experimental results. Cold & Warm Net performs better in terms of VPI and VSR on cold-start users. DSSM model as the baseline.

Models	VPI	VSR
Cold & Warm	**+23.34%**	**−14.30%**
Mind	-2.05%	+2.76%

the main metrics for cold-start users. All online experimental results are averaged over a week's A/B test on Little-World. The result shows that Cold & Warm Net brings an increase of 3.27% in APT and 1.01% in URR for cold-start users. Meanwhile, we compare Mind and Cold & Warm Net using the DSSM model as the baseline. Video play integrity (VPI) and video skip rate (VSR) are used as the main metrics for user satisfaction evaluation. The result in Table 6 shows that Cold & Warm Net outperforms the DSSM model on both VPI and VSR, which indicates successfully modeling cold-start users and improving user satisfaction.

5 Conclusion

User cold-start problem in the matching stage is a critical challenge faced by RS. However, the solutions are rare both in academia and industry. In this paper, we propose Cold & Warm Net which effectively solves the problem for cold-start users, while in the meantime satisfying the scalability required by the billion-scale matching stage. We first construct our network with two experts and incorporate a gate network to combine results according to the user state. Bias net and DKD module responsible for modeling cold-start users are incorporated. Finally, we evaluate our model through offline and online experiments and it achieves an obvious increase in recommendation performance.

References

1. Chai, Z., Chen, Z., Li, C., Xiao, R., Li, H., Wu, J., Chen, J., Tang, H.: User-aware multi-interest learning for candidate matching in recommenders. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1326–1335 (2022)
2. Chen, H., Wang, Z., Huang, F., Huang, X., Xu, Y., Lin, Y., He, P., Li, Z.: Generative adversarial framework for cold-start item recommendation. In: Proceedings

of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2565–2571 (2022)

3. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM conference on recommender systems. pp. 191–198 (2016)
4. Dong, M., Yuan, F., Yao, L., Xu, X., Zhu, L.: Mamo: Memory-augmented meta-optimization for cold-start recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 688–697 (2020)
5. Hao, B., Zhang, J., Yin, H., Li, C., Chen, H.: Pre-training graph neural networks for cold-start users and items representation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 265–273 (2021)
6. Hu, G., Zhang, Y., Yang, Q.: Conet: Collaborative cross networks for cross-domain recommendation. In: Proceedings of the 27th ACM international conference on information and knowledge management. pp. 667–676 (2018)
7. Huang, J.T., Sharma, A., Sun, S., Xia, L., Zhang, D., Pronin, P., Padmanabhan, J., Ottaviano, G., Yang, L.: Embedding-based retrieval in facebook search. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2553–2561 (2020)
8. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)
9. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 197–206. IEEE (2018)
10. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., Lee, D.L.: Multi-interest network with dynamic routing for recommendation at tmall. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2615–2623 (2019)
11. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing **7**(1), 76–80 (2003)
12. Lv, F., Jin, T., Yu, C., Sun, F., Lin, Q., Yang, K., Ng, W.: Sdm: Sequential deep matching model for online large-scale recommender system. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2635–2643 (2019)
13. Rendle, S.: Factorization machines. In: 2010 IEEE International conference on data mining. pp. 995–1000. IEEE (2010)
14. Wang, P., Fan, Y., Xia, L., Zhao, W.X., Niu, S., Huang, J.: Kerl: A knowledge-guided reinforcement learning model for sequential recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 209–218 (2020)
15. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 974–983 (2018)
16. Zhang, Y., Shi, Z., Zuo, W., Yue, L., Liang, S., Li, X.: Joint personalized markov chains with social network embedding for cold-start recommendation. Neurocomputing **386**, 208–220 (2020)