CBML: A Cluster-based Meta-learning Model for Session-based Recommendation

Jiayu Song
jysongsuda@stu.suda.edu.cn
School of Computer Science and Technology, Soochow University
Suzhou, China

Jiajie Xu*
xujj@suda.edu.cn
School of Computer Science and Technology, Soochow University
Suzhou, China

Rui Zhou
rzhou@swin.edu.au
Swinburne University of Technology
Australia

Lu Chen
luchen@swin.edu.au
Swinburne University of Technology
Australia

Jianxin Li
jianxin.li@deakin.edu.au
Deakin University
Australia

Chengfei Liu
cliu@swin.edu.au
Swinburne University of Technology
Australia

ABSTRACT

Session-based recommendation is to predict an anonymous user's next action based on the user's historical actions in the current session. However, the cold-start problem of limited number of actions at the beginning of an anonymous session makes it difficult to model the user's behavior, i.e., hard to capture the user's various and dynamic preferences within the session. This severely affects the accuracy of session-based recommendation. Although some existing meta-learning based approaches have alleviated the cold-start problem by borrowing preferences from other users, they are still weak in modeling the behavior of the current user. To tackle the challenge, we propose a novel cluster-based meta-learning model for session-based recommendation. Specially, we adopt a soft-clustering method and design a parameter gate to better transfer shared knowledge across similar sessions and preserve the characteristics of the session itself. Besides, we apply two self-attention blocks to capture the transition patterns of sessions in both item and feature aspects. Finally, comprehensive experiments are conducted on two real-world datasets and demonstrate the superior performance of CBML over existing approaches.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

session-based recommendation; meta-learning; soft-clustering; content information

*Corresponding author

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482239

ACM Reference Format:
Jiayu Song, Jiajie Xu, Rui Zhou, Lu Chen, Jianxin Li, and Chengfei Liu. 2021. CBML: A Cluster-based Meta-learning Model for Session-based Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482239

1 INTRODUCTION

Recommender systems play an important role in providing users required information in a timely and effective manner. Most existing recommendation methods assume that user profiles and past activities are constantly recorded. However, in many scenarios, a user is usually anonymous and only a few user historical actions in an ongoing session can be used to predict the user's next click. This motivates session-based recommendation, which has become an important sub-area of recommender systems. Existing solutions utilize deep neural networks like improved LSTM [13], GRU [2] and self-attention mechanism [31] to capture user preferences within sessions. However, since anonymous sessions tend to contain few interactions, this cold-start problem severely limits the performance of session-based recommendation.

As a representative few-shot learning method, meta-learning is proposed in [5] and shows promising results in many cold-start applications, such as few-shot image classification [7]. Inspired by this, some recent studies [3, 12] have adopted meta-learning in recommendation tasks for addressing cold-start problems. In these models, each user is regarded as a learning task. These models first learn well-generalized global parameters that can reasonably initialize the parameters of all tasks. When processing each recommendation to a user [3], local updates are conducted on initialized parameters using the user's own data to derive personalized parameters, which represents user-specific preferences and enables meaningful recommendation. Since meta-learning turns out to be effective in cold-start scenarios [3, 12, 37], it provides great opportunity for session-based recommendation. However, in session-based recommendation where each session becomes a learning task, directly applying existing meta-learning based recommendation methods would incur inaccuracy due to two limitations discussed below.

First, previous meta-learning based recommendation models [3, 4, 12, 37] for cold-start problems assume that the same global parameters are used to guide parameter initialization for all tasks.

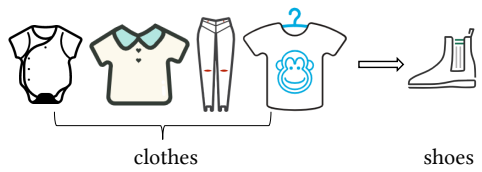


Figure 1: An example of category relationship between items.

However, although a globally shared parameter setting may be superior to others in the overall performance, it is unlikely to achieve better performance for every session. Recently, [3] tries to resolve this problem by designing memory matrices with users' historical actions to guide the model with personalized parameter initialization. Yet, this method is not suitable for session recommendation because users are anonymous and there may be only a few interactions in an ongoing session. In fact, although different sessions are originated from different distributions, the shared knowledge can be transferred across similar sessions, because similar users' preferences are also similar, indicating the generalization among closely correlated sessions. Thus, in order to avoid the impact of divergence between different sessions and better transfer shared knowledge across similar sessions, it would be beneficial to divide sessions into several clusters and learn shared knowledge in each cluster respectively.

Second, the content-based methods [23, 29], using auxiliary information, such as labels, comments, etc, are helpful to address the cold-start problem of recommendation system. Existing meta-learning methods [3, 12] for cold-start problems, which are primarily for general situations, embed user features without further considering contents. However, for the severe cold start problem, it is essential for session recommendation to take auxiliary information into consideration. For example, as is shown in Figure 1, after clicking shirts and trousers (clothes), a user is more likely to click shoes to see if it fits the chosen clothes, indicating the category between items is also important for next item prediction. Thus, meta-learning methods for session-based recommendation should be extended to model the collaboration sequence of both the feature layer and the item layer. In this way, sequential patterns of items and features can be captured simultaneously for next item prediction, such that auxiliary information is utilized more sufficiently to alleviate the cold-start problem.

To address the above challenges, in this paper, we propose a cluster-based meta-learning model (CBML) for session based recommendation, which is the first to adopt meta-learning to address the cold start problem in session-based recommendation. Specifically, to better share the generalized knowledge among similar sessions, whose users are similar, instead of being affected by the divergence between different sessions, we adopt a soft-clustering method on training sessions to derive clusters, through which each session obtains a cluster enhanced representation to contain shared characteristics of the clusters the session belongs to. Then, we utilize a carefully-designed parameter gate to guide the initialization of the globally shared parameters to each cluster, such that the initialization can serve for all sessions belonging to the cluster. Moreover, CBML integrates an item-based self-attention block and a feature-based self-attention block in meta-learning to capture the transition patterns of sessions in both item and feature aspects. The main contributions of this paper are summarized as follows:

- We propose a novel cluster-based meta-learning for session recommendation, which is the first to deal with the cold start problem in session recommendation with meta-learning.
- We adopt a soft-clustering method in meta-learning and design a parameter gate to better transfer shared knowledge across similar sessions.
- The proposed meta-learning model fully considers content information to capture more fine-grained sequential intents of a user, which includes item-level sequence patterns and feature-level transition patterns.
- We conduct extensive experiments on two real-world datasets to demonstrate the effectiveness of our method.

The remainder of this paper is organized as follows. We firstly review the related work in Section 2. Then, we present our problem in Section 3. Next, in Section 4, we present our proposed method CBML for addressing user cold-start problems in session-based recommendation in detail. In Section 5, we compare our model with the state-of-the-art methods and ablation experiments to confirm the effectiveness of our model. Finally, in Section 6, we conclude the general idea of this paper.

2 RELATED WORK

2.1 Session-based Recommendation

Session-based recommendation selects required information for users based on anonymous behavior sequences, including implicit feedbacks instead of explicit preferences. Therefore, the model-based methods which make use of user profiles cannot be applied for session-based recommendation.

In this scenario, early works on session-based recommendation focused on employing item-to-item relations, such as transition relation and co-occurrence relation. For example, typical Markov chain-based methods [8, 9] map the recurrent session into a Markov chain, and then rely on the last element in the session to infer a user's next action. [22] captures long-term preferences and short-term item-item transitions respectively for recommendation by fusing matrix factorization and first-order Markov chains.

In recent years, many studies utilize deep neural networks like GRU [2, 27] and attention mechanism [10] to model the sequential patterns within sessions. More recently, graph neural networks (GNN) [30, 31, 34] are embedded in the sequential models to further capture the pairwise item transitions by modeling the session as a graph structure. What is more, the self-attention method has also become popular in many areas for sequential patterns, such as natural language processing [14] and recommendation areas [16, 31]. For example, Xu et al. [31] and Luo et al. [16] utilize multi-head self-attention methods to capture the item-item transitions and global dependencies between the whole input sequence without regard to the distances of items. This can make full use of user's historical records to capture user's preferences for better recommendation.

Although these neural network methods can effectively model sequential patterns for recommendation, their performances still

have more room to improve in session scenarios, because there are just a few historical interactions in one ongoing anonymous session and content information can also be helpful to capture user preference.

2.2 Meta-learning for Recommendation

Meta-learning [11], also called learning-to-learn, aims to learn the general knowledge through several tasks that can rapidly be adapted to new tasks. It is usually divided into three categories: memory-based meta-learning method [18, 19, 24], metric-based meta-learning method [25, 26, 28] and optimization-based meta-learning method [1, 5, 6].

Recently, meta-learning methods have been applied in many areas, such as few-shot learning in computer vision [33, 35], and natural language processing [17, 20]. Inspired by few-shot problem, some recent studies [3, 4, 12, 15, 37] have adopted meta-learning methods in recommendation tasks for addressing cold-start problems. Most existing meta-learning works for recommendation utilize optimization-based strategies. They learn a meta-optimizer that learns globally shared initial parameters through several tasks so that these initial parameters can make new tasks quickly adapt to the model's best point. For example, [12] globally updates the whole model to get well-generalized initial parameters for all users and locally updates the prediction layers to get personalized recommender. [15] utilizes the similar idea of [5] to learn well-generalized initial parameters by locally and globally updating the whole heterogeneous information networks (HINs). [3] tries to embed user representation into meta-learning to guide the model with personalized parameter initialization, which considers user's personalized preferences based on [12].

Although existing works try to address the cold-start problem of general recommendation, which provides great opportunities for session-based recommendation, they cannot well support session-based recommendation because of lacking user attributes in anonymous sessions which are important for user embedding [3, 12, 15]. To this end, this paper aims to propose a cluster-based meta-learning model, which particularly ensures shared knowledge among similar sessions, whose users are similar, to be transferred and design suitable initial parameters for each session to provide an accurate recommendation.

3 PROBLEM DEFINITION AND PRELIMINARIES

3.1 Problem Statement

Session-based recommendation aims to predict which item the anonymous user will click next based on the current session. Here, in our model, we make $V=\{v_1, v_2, ..., v_{|V|}\}$ denote all the unique items and also $U=\{u_1, u_2, ..., u_{|U|}\}$ be a set of anonymous sessions. A session with m click actions can be denoted as $u_i=\{x_1, x_2, ..., x_m\}$ in chronological order, where $x_t \in V$ represents a clicked item at time step t. Each item v_i has some attributes. We mainly consider category, brand, and seller in this paper, while other attributes can be incorporated if needed.

The goal of session-based recommendation is to predict the next click (i.e., x_{m+1}) given a session u_i. Formally, we consider the recommendation for a session as one task. Given a session $u_i=\{x_1, x_2, ..., x_m\}$, our recommendation model aims to calculate the probabilities $\hat{y}=\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_{|V|}\}$ of all candidate items and then choose N candidate items with the highest probabilities for recommendation.

3.2 Meta-learning Setting

Here for the meta-learning setting, we consider one session as one task and divide the sessions into a training set T^{train} for meta-training and a test set T^{test} for meta-testing. For each session $u_i=\{x_1, x_2, ..., x_m\} \in T^{train} \cup T^{test}$, we generate $m-2$ sub-sequences $(\{x_1\}, x_2)$, $(\{x_1, x_2\}, x_3)$, ..., and $(\{x_1, x_2, ..., x_{m-2}\}, x_{m-1})$ forming a support set D_u^{train}, and take the original sequence $\{x_1, x_2, ..., x_m\}$ as a singleton query set D_u^{test}. The support set and query set are used to update model parameters locally and globally, which are detailed in Section 4.2.

4 APPROACH

4.1 Overview

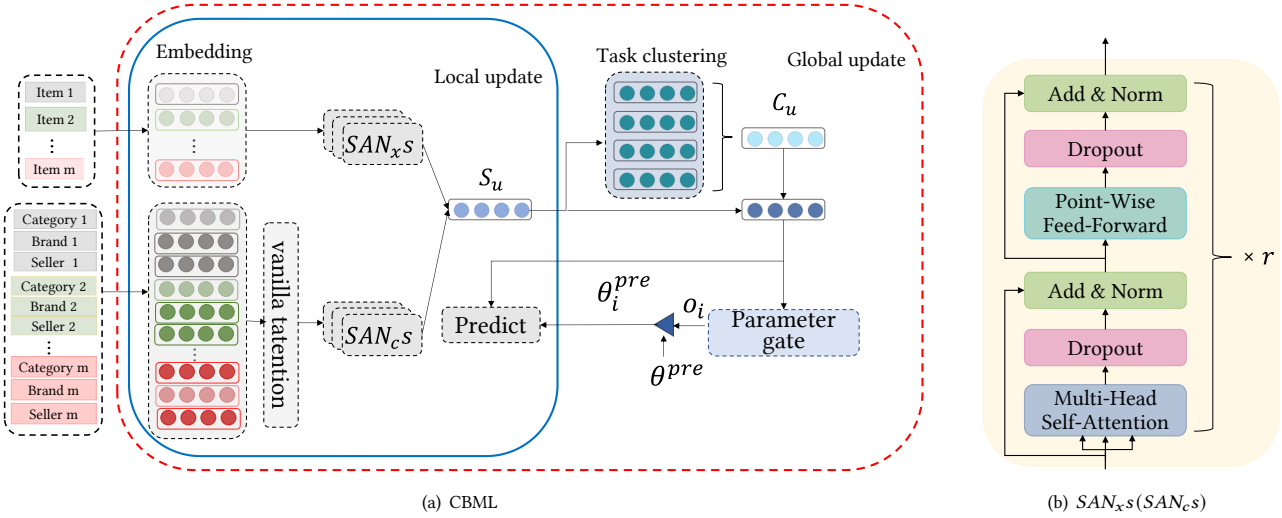
In this section, we will detail our proposed method a cluster-based meta-learning model for session-based recommendation (CBML) in Figure 2(a). First, we will introduce a base model IF-SAN for the session-based recommendation, which integrates an item-based self-attention block and a feature-based self-attention block to capture the transition patterns of sessions in both item aspects and feature aspects for more fine-grained sequential intents of a user. Then, we will present the details of the designed meta-learning framework of CBML model to learn the suitable initial parameters for each session to address the problem of user cold-start. The model CBML considers the situation that different sessions require different initial parameters for better recommendation on the basis of the traditional optimization-based meta-learning method [1, 5, 6].

4.2 Session-based Recommender

For session-based recommendation, the state-of-the-art models [30, 31, 34] can be directly applied as a base model for meta-learning framework. However, they consider sequential patterns between items only, ignoring the sequential patterns between features that are crucial for sufficient utilization of auxiliary information for recommendation. As a result, in our base mode IF-SAN, we further exploit feature-level transition patterns on the basis of the state-of-the-art models [30, 31, 34] for alleviating the cold-start problem to some extent.

Specifically, our proposed session-based recommender IF-SAN with parameters $\theta^* = \{\theta^e, \theta^s, \theta^{pre}\}$ is composed of four components, i.e., embedding layers θ^e for both item-level and feature-level embedding, item-based self-attention layers and feature-based self-attention layers θ^s for transition patterns of both item-item and feature-feature, and prediction layers θ^{pre} to select next items for users. Next, we will discuss the implementation of each component in detail.

4.2.1 Embedding Layers. Since the length of the different sessions are not equal, we take a fixed-length vector to represent a session embedding, $u = \{x_1, x_2, ..., x_n\}$, where n is a predefined size of the sequence, and adopt a zero-padding at the front of the sequence if a sequence does not have enough items. Then, we apply a lookup



(a) CBML (b) $SAN_x s(SAN_c s)$

Figure 2: (a) the framework of CBML. The embedding layers, SANs and prediction layers (marked with grey box) constitute the recommendation model for sessions. The task-clustering layer and parameter gate (marked with blue box) are designed to guide the initial parameters to serve for each session separately. We locally update the recommendation model based on the support set of each session and globally updated all layers based on the query set of all sessions. (b) the composed $SAN_x s(SAN_c s)$ blocks.

layer to transform the one-hot vectors of item sequence into a dense vector representation. Thus, the embedding of item x_i can be obtained, denoted as e_{x_i}.

Similarly, the attribute sequences are processed in the same way. Given an item x_i, its attributes can be embedded as $a_i = \{vec(r_i), vec(b_i), vec(l_i)\}$, where $vec(r_i)$, $vec(b_i)$ and $vec(l_i)$ represent the dense vector representations of category, brand and seller of item x_i.

Since different attributes are often heterogeneous and have different effects on a user's decision, we follow the way in [36] to adopt a vanilla attention mechanism to transform the weighted sum of the item x_i's attribute vector representations into a feature representation e_{ai}. The attention mechanism is as follows,

$$\alpha_i = softmax(W^f a_i + b^f) \tag{1}$$

where W^f is $d \times d$ matrice and d^f is $d-$dimensional vector.

Finally, the feature representation of item i can be computed as:

$$e_{a_i} = \alpha_i a_i \tag{2}$$

Note that if there is only one attribute (e.g., category) in item i, the vanilla attention mechanism is unnecessary and the feature representation of item i is $e_{a_i} = vec(r_i)$.

4.2.2 Feature-based Self-Attention Layers. Self-attention has been applied in many areas. It can capture the global dependence of the whole sequence regardless of the distance of the sequential input and output. Thus, we employ two self-attention blocks on item sequence and feature sequence for item-item and feature-feature transition patterns. Since the difference between item-based self-attention block and feature-based self-attention block is only their input, we illustrate the process of feature-based self-attention block in detail. We adopt a multi self-attention block [31] on the feature sequence to obtain the feature-feature transitions across the entire input and output sequence itself. The framework of the self-attention network ($SAN_* s$) can be seen in Figure 2(b). The $SAN_x s$ blocks represent self-attention network for item-level and the $SAN_c s$ blocks represent self-attention network for feature-level.

After the embedding layers and the vanilla attention mechanism, we can obtain the embeddings of all the attributes in the session u, i.e., $E_{ua} = \{e_{a_1}, e_{a_2}, ..., e_{a_n}\}$. Then, we adopt a self-attention network to better capture the global preference from the input sequence,

$$F_{ua} = SAN(E_{ua}) \tag{3}$$

Since different layers can capture different types of features, we stack the self-attention block to obtain complex feature transitions. Thus, the l-th ($l > 1$) self-attention layer is:

$$F_{ua}^l = SAN(F_{ua}^{l-1}) \tag{4}$$

where $F_{ua}^1 = F_{ua}$ and F_{ua}^l is the final output of the multi-layer self-attention network.

Then, in order to better describe the feature-level sequence patterns, we combine the long-term preference and the current interest of the session u as the session representation in the feature-level,

$$S_{ua}^l = w_a F_{ua}^l + (1 - w_a) e_{a_m} \tag{5}$$

where w_a is a weighting parameter and a_m is the last element in the original feature sequence of session u which is not padded by zero.

4.2.3 Item-based Self-Attention Layers. Since the difference between the item-based self-attention block and the feature-based self-attention block is only their input, it is easy to see that the

item-based self-attention block can be constructed like the feature-based self-attention block. Thus, the representation of the session u, in the item-level can be obtained by,

$$F_{ux}^{l} = SAN(F_{ux}^{l-1}) \tag{6}$$

$$S_{ux}^{l} = w_x F_{ux}^{l} + (1 - w_x) e_{x_m} \tag{7}$$

where w_x is the weighting parameter, $F_{ux}^{1} = F_{ux}$ and F_{ux}^{l} is the final output of the item-based multi-layer self-attention network, x_m is the last element in the original item sequence of session u before zero-padding.

4.2.4 Prediction Layers. In order to capture the transition patterns of items and features at the same time, we combine the session representation in item-level S_{ux}^{l} and feature-level S_{ua}^{l} as the final *session specific representation*, and feed them into a fully-connected layer for the final calculation of user preferences for items.

$$\hat{y}_{ui} = FC_{\theta^{pre}}(S_u) e_{v_i}^{T} \tag{8}$$

where $S_u \in \mathbf{R}^{2d}$, $FC_{\theta^{pre}}(\cdot)$ is a fully-connected layer with parameters θ^{pre} for prediction, e_{v_i} ($v_i \in V$) is the embedding of the $i\text{-}th$ item and $\hat{y}_{ui}$ is the probability of the $i\text{-}th$ item to be the next click in the session u.

4.3 Cluster-based Meta Optimization

In this section, we elaborate a cluster-based meta-learning framework with parameters ϕ, which addresses the problem that different sessions are expected to have different initial parameters on the basis of traditional optimization-based meta-learning method [1, 5, 6] for better recommendation. The framework consists of two components, task clustering and cluster-aware parameter gate. The former component is to adopt a soft-clustering method on the training sessions to derive clusters. Since similar users have similar preferences, indicating the generalization among closely correlated sessions, each session can obtain a *cluster enhanced representation* that contains the shared characteristics of the clusters the session is classified to. The latter component aims to guide the initialization of the globally shared parameters to each cluster, such that the initialization can serve for all the sessions belonging to the cluster. At last, we will introduce the process of the local updating for each session and global updating for the initialization of the globally shared parameters.

4.3.1 Task Clustering. Since there exists impact from the divergence between different sessions and the shared knowledge can be transferred across similar sessions, where users are similar, we propose a clustering method to locate the cluster the session belongs to in order to better transfer shared knowledge among similar sessions for each session. Although there already exist meta-learning methods and clustering methods for the recommendation problems, no one has combined the advantages of the two methods. There are no sufficient records for a cold user in a session to obtain sufficient preferences. However, the clustering method can help transfer shared knowledge across similar sessions and the meta-learning method can learn general knowledge through several sessions that can rapidly be adapted to new sessions. Thus, it is suitable to combine the clustering method with the meta-learning method to solve the cold-start problem. In this framework, we adopt soft-clustering method instead of hard-clustering method. There are mainly two reasons. First, in reality, session groups are overlapping and the sharing knowledge between sessions often exists because anonymous users from the same crowd have similar hobbies and preferences. Second, soft assignment can guarantee differentiability while hard assignment cannot.

Here we follow the traditional method of clustering. First, we conduct a cluster assignment to each cluster for each session. In particular, as introduced in Section 4.1, we use embedding layers and two self-attention blocks in both item-level and feature-level to obtain a *session specific representation* S_u, reflecting the anonymous user's preference, and then linearly project the representation to get the query vector q_u for the clusters, which is formulated as follows:

$$q_u = W_q S_u + b_q \tag{9}$$

where $W_q \in \mathbf{R}^{2d \times d}$ and $b_q \in \mathbf{R}^{d}$ are learned parameters.

Next we compute the soft-assignment probability vector p_u^k by calculating the distance between the query vector $q_u \in \mathbf{R}^{d}$ and each learned cluster center $\{g_k\}_{k=1}^{K}$, i.e.,

$$p_u^k = \frac{\exp(\langle q_u, g_k \rangle)}{\sum_{k=1}^{K} \exp(\langle q_u, g_k \rangle)} \tag{10}$$

where K denotes the number of clusters and determining K will be discussed in experiments.

Finally, the *cluster enhanced representation* of the session u, which contains shared characteristics of the clusters this session belongs to, can be calculated as:

$$C_u = \sum_{k=1}^{K} p_u^k \cdot g_k \tag{11}$$

Here $\cdot$ is the multiplication.

Note that cluster centers are being updated continuously, because new training sessions are coming in due to new clicks or newly established conversations, and outdated training sessions are being discarded. We randomly initialize each cluster center at the beginning.

4.3.2 Cluster-aware Parameter Gate. Since different sessions are originated from different distributions (denoting different preferences or hobbies), it is irrational to utilize a single globally shared parameter for the recommendation of all sessions. [32] indicates that similar meta-parameters can be shared across similar tasks. Thus, in order to preserve the personalization of each session and include the generalization among similar sessions, we propose to combine the *session specific representation* S_u and the *cluster enhanced representation* C_u. And we design a cluster-aware parameter gate to guide the globally shared initial parameters to suitable initial parameters for each session to achieve better performance. The parameter gate is designed as follows,

$$o_u = FC^{\sigma}(S_u \oplus C_u) \tag{12}$$

where $\oplus$ means the tensors concatenation, and FC^{σ} is a fully-connected layer activated by a sigmoid function σ. Then the globally shared initial parameters $\theta^{*} = \{\theta^{e}, \theta^{s}, \theta^{pre}\}$ of the session-based recommendation model IF-SAN can be guided for session u as follows,

$$\theta_u^{e} \leftarrow \theta^{e}, \theta_u^{s} \leftarrow \theta^{s} \tag{13}$$

$$\theta_u^{pre} \leftarrow \theta^{pre} \cdot o_u \tag{14}$$

Therefore, the final prediction for next item in the session u can be calculated as,

$$\hat{y}_{ui} = FC_{\theta_u^{pre}}(S_u \oplus C_u) \cdot e_{v_i}^T \tag{15}$$

Here $FC_{\theta_u^{pre}}$ is the prediction layer in Section 4.1 with guided initial parameters, $v_i \in V$ is candidate item for session u and $e_{v_i}^T$ is the transpose of the embedding of the candidate item v_i.

4.3.3 Local Update. Typically, the parameters of a neural network are randomly initialized, and then the initialization of the parameters converges to a good local optimum by minimizing prediction loss based on the training set. Similar to the neural network, in meta-learning, we aim to update the local parameters for each session by minimizing the prediction loss of the recommendation based on the support set of a single session. After we guide the globally shared initial parameters for a session in the cluster-aware parameter gate, we can locally update the recommender parameters to minimize the prediction loss of session u. Here the local parameters $\theta_u^* = \{\theta_u^e, \theta_u^s, \theta_u^{pre}\}$ of the session-based recommender IFSAN for session u can be updated as follows,

$$\hat{\theta}_u^* \leftarrow \theta_u^* - \beta \cdot \nabla_{\theta_u^*} \mathcal{L}_u(\phi, \theta_u^*) \tag{16}$$

where β is the learning rate for updating the local parameters of recommender for each session and $\mathcal{L}_u(\cdot,\cdot)$ is the prediction loss of support set D_u^{train} in session u.

4.3.4 Global Update. In optimization-based methods [5, 21], they learn a well-generalized model through several tasks by updating the shared initial parameters. Similar to these methods, in our meta-optimization process, we take one-step gradient descent to update the global parameters, $\theta^* = \{\theta^e, \theta^s, \theta^{pre}\}$ and ϕ, according to the sum of the loss on query set D_u^{test} of each session $u \in T^{train}$ after the local updating on support set D_u^{train},

$$\theta^* \leftarrow \theta^* - \gamma \sum_{u \in T^{train}} \nabla_{\theta^*} \mathcal{L}_u^{'}(\phi, \hat{\theta}_u^*) \tag{17}$$

$$\phi \leftarrow \phi - \gamma \sum_{u \in T^{train}} \nabla_{\phi} \mathcal{L}_u^{'}(\phi, \hat{\theta}_u^*) \tag{18}$$

Here, γ is the learning rate for updating the initialization of globally shared parameters. $\mathcal{L}_u^{'}(\cdot,\cdot)$ is the prediction loss of query set in session u.

5 EXPERIMENTS

In this section, we conduct extensive experiments on two real-world datasets to evaluate the performance of our proposed method CBML. Here we first describe the experimental setup and compared methods. Then we compare our method CBML with other variants of CBML. Finally, we analyze the influence of different experimental settings in CBML.

5.1 Experimental Setup

5.1.1 Datasets. Table 1 shows the statistics of the two representative real-world datasets, Yoochoose[1] and Diginetica[2]. The Yoochoose dataset is a public dataset released by the RecSys Challenge 2015, which contains 6 months of a stream of user clicks on an e-commerce website. The Diginetica dataset is obtained from CIKM Cup 2016, and we only utilize its transactional data. For our model, we set the session data of last week as the test set for meta-testing, and the remaining as a training set for meta-training. And we follow the way in [30] to generate sub-sequences of each session as introduced in Section 3.

[1]http://2015.recsyschallenge.com/challege.html
[2]http://cikm2016.cs.iupui.edu/cikm-cup

Table 1: Statistics of datasets used in the experiments

Dataset	Yoochoose1/64	Yoochoose1/4	Diginetica
#of train	65,172	1,090,115	133,724
#of test	9,347	40,618	11,446
#of items	37,487	37,487	43,097
Averg length	4.95	4.66	4.98

5.1.2 Parameter Setup and Metrics. Through several comparisons of different experimental settings in Section 5.5~5.7, we choose the settings with the best performance as our model CBML settings. We set the dimension of latent vectors as 100, the number of self-attention layers as 2. And the number of clusters is set as 8. The initial learning rates of local update and global update are set as 0.001 and 0.001 respectively. We utilize Adam as an optimizer to evaluate models and adopt three common metrics, i,e., Hit Rate (Hit@N), Mean Reciprocal Rank (MRR@N), and Normalized Discounted Cumulative Gain (NDCG@N). The former one is an evaluation of unranked retrieval results, while the latter two are evaluations of ranked lists. Here, we consider Top-N (N = 5) for the recommendation.

5.2 Baselines

To evaluate the effectiveness of our model, CBML, we compare it with the state-of-the-art session-based recommendation models (SR-GNN, GC-SAN, TAGNN) and our proposed base model IF-SAN. In addition, we also compare our model with methods that deal with cold-start problems, e.g., transfer learning based model Multi-FT and meta-learning based models (MeLU, MAMO, MetaHIN). This further confirms the effectiveness of our model for alleviating cold-start problems. To be fair, we use the same base model IF-SAN for Multi-FT, MeLU, MAMO, and our model CBML and we add the same self-attention blocks that are from IF-SAN for MetaHIN to make this model suitable for our session-based recommendation problem.

- **SR-GNN** [30] models the sessions as a graph to capture complex item interactions and combines the user's global preferences and current interests through an attention mechanism. It only considers interactions on item-level.
- **GC-SAN** [31] follows a similar idea of SR-GNN to generate complex item interactions and then utilizes a self-attention network to represent each session, which just considers item-level transitions.
- **TAGNN** [34] builds the model based on item-level transitions and adopts similar methods of SR-GNN to capture the user's global preferences and current interests and additionally utilizes target attentive network to activate the users' diverse interests in sessions.

Table 2: The performance of CBML compared with other baseline methods.

datasets	Diginetica			Yoochoose 1/64			Yoochoose 1/4		
method	Hit@5	MRR@5	NDCG@5	Hit@5	MRR@5	NDCG@5	Hit@5	MRR@5	NDCG@5
SR-GNN	10.6238	6.7439	5.3291	34.7170	21.7888	17.6824	32.9840	21.2593	18.1067
TAGNN	10.5713	6.7518	5.3608	35.1129	21.7958	17.9213	33.0129	21.9036	18.4814
GC-SAN	11.0377	6.7981	5.4416	35.5710	21.7277	18.0992	33.2085	22.1026	18.6842
IF-SAN	12.3362	6.9917	5.5532	36.8246	21.8294	18.2128	35.3381	23.8499	19.3483
Multi-FT	21.2476	13.6995	11.2924	40.6654	26.9981	20.6601	41.5497	27.0685	21.7091
MeLU	24.0665	16.1092	12.3892	41.3181	28.5153	21.6111	41.6478	27.6303	21.9064
MAMO	23.6495	15.6417	12.2010	41.7781	28.4535	22.0710	41.6818	28.1440	21.9601
MetaHIN	24.2852	16.8975	12.9186	42.2595	28.6755	22.2812	42.9977	28.5743	22.2757
CBML	**25.9869**	**18.0444**	**13.8275**	**44.2923**	**29.9502**	**23.0022**	**46.1218**	**30.8459**	**23.7872**
Improv.	7.01%	6.79%	7.04%	4.81%	4.45%	3.24%	7.27%	7.95%	6.79%

Table 3: The performance of CBML compared with variants of CBML.

datasets	Diginetica			Yoochoose 1/64			Yoochoose 1/4		
method	Hit@5	MRR@5	NDCG@5	Hit@5	MRR@5	NDCG@5	Hit@5	MRR@5	NDCG@5
CBML-G	23.3840	16.0934	13.0412	41.1897	28.3094	22.0884	43.8435	29.0522	22.6139
CBML-C	23.7696	16.2853	13.0605	43.6660	29.2582	22.5653	44.6453	29.9803	23.0246
CBML-S	24.5453	17.0526	13.2658	42.8112	29.0723	22.3395	44.2970	29.3859	23.0103
CBML-F	20.9593	15.0459	12.5786	40.6768	27.6011	21.8482	42.2095	27.7935	21.3036
CBML	**25.9869**	**18.0444**	**13.8275**	**44.2923**	**29.9502**	**23.0022**	**46.1218**	**30.8459**	**23.7872**

- **IF-SAN** is our base model. It utilizes two self-attention blocks on item-level and feature-level to capture item-level sequential patterns and feature-level sequential patterns. More details can be seen in Section 4.2.
- **Multi-FT** is a transfer learning-based method, which trains IF-SAN based on training sessions and fine-tunes the model for test sessions.
- **MeLU** [12] adopts the traditional optimization-based meta-learning method. It feeds the item embeddings into fully connected layers for recommendation. It locally updates the parameters of the fully connected layers for personalized recommendation and globally updates the parameters of the whole model for all users.
- **MAMO** [3] is an improvement on MeLU. It designs two memory matrices with user profiles to guide the model with personalized parameter initialization. In our setting, since users are anonymous, we remove the memory matrices with user's profiles.
- **MetaHIN** [15] combines MAML with HINs to exploit the power of meta-learning at the model level and HINs at the data level simultaneously to alleviate the cold-start problem. The rich semantic of HINs provides a fine-grained prior which is beneficial to fast adaptations of new tasks.

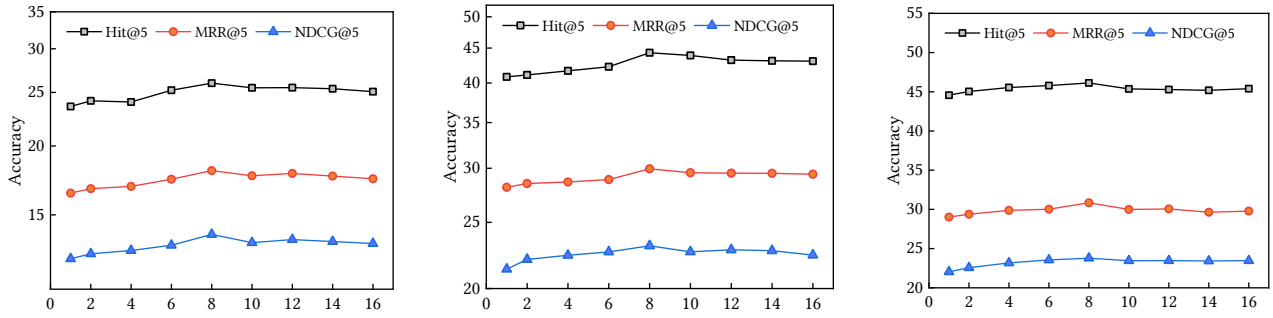
5.3 Comparisons of Performance

For a fair comparison, we run three times of each method and take the average value as the final result. Table 2 illustrates the experimental results of all methods on both datasets and we have the following observations.

5.3.1 Base Model Comparison. To evaluate the performance of our base model IF-SAN, we compare it with the start-of-the-art session-based recommendation methods, including SR-GNN, GC-SAN, and TAGNN. As shown in Table 2, GC-SAN performs better than SR-GNN and TAGNN, which indicates that the self-attention mechanism is effective for sequential patterns. The IF-SAN model, which is our base model and considers item-level and feature-level sequential patterns at the same time, further improves the Hit@5, MRR@5, and NDCG@5 metrics. This can be explained that the more sufficient use of auxiliary information can make the model obtain more fine-grained sequential intents of a user, which helps alleviate the cold start problem. Therefore, we consider IF-SAN as the most effective meta-learning base model for session-based recommendation.

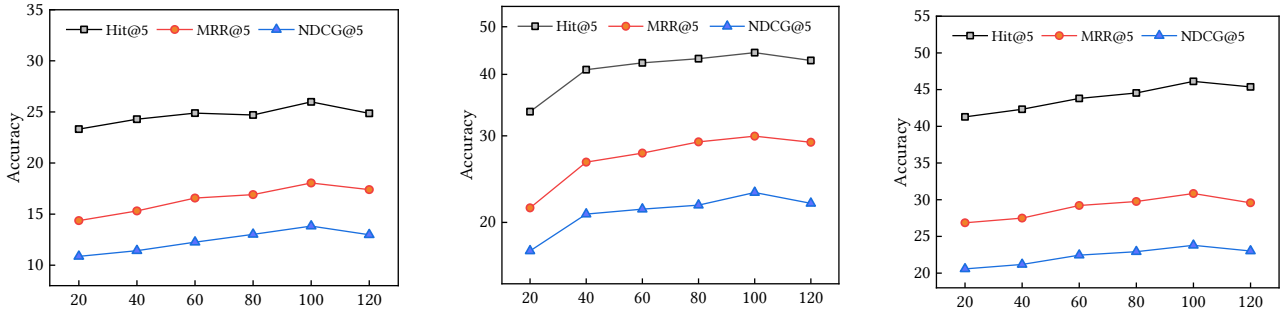
5.3.2 Meta-learning Strategy Comparison. Then we evaluate the effectiveness of the transfer learning method (Multi-FT) and different meta-learning strategies (MeLU, MAMO, MetaHIN, and CBML). From Table 2, we can observe that the transfer learning-based approach Multi-FT significantly outperforms all the base models(SR-GNN, TAGNN, GC-SAN, and IF-SAN), indicating that the transfer learning method can alleviate the cold-start problem to some extent. In addition, the meta-learning based models (MeLU, MAMO, MetaHIN, and CBML) further perform better than Multi-FT on all datasets. This demonstrates that the meta-learning method may be more preferred than the transfer learning method for session-based recommendation because the main idea of all meta-learning methods is to capture the generalization among different sessions.

When comparing the meta-learning based strategies, we can see that MeLU performs better than MAMO on Diginetica, while it performs worse on Yoochoose1/64 and Yoochoose1/4. This means



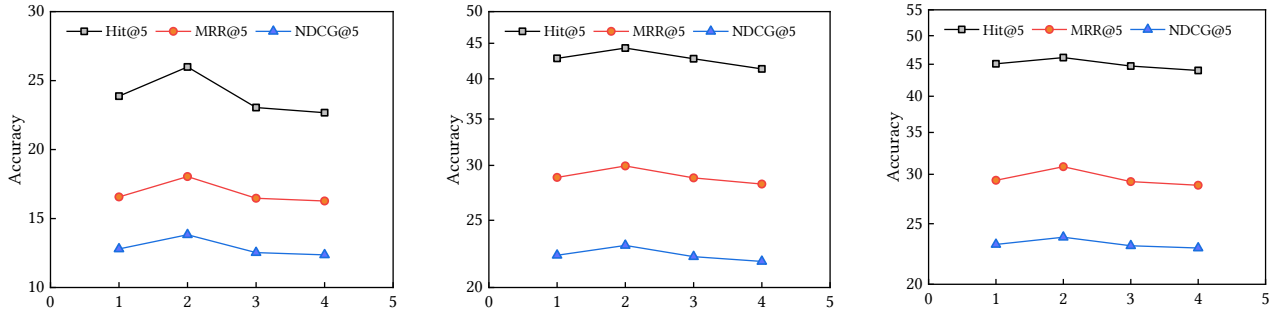
(a) Diginetica (b) Yoochoose1/64 (c) Yoochoose1/4

Figure 3: The performance under different cluster numbers K.



(a) Diginetica (b) Yoochoose1/64 (c) Yoochoose1/4

Figure 4: The performance under different embedding sizes d.



(a) Diginetica (b) Yoochoose1/64 (c) Yoochoose1/4

Figure 5: The performance under different number of stacked self-attention blocks l.

that MeLU and MAMO are suitable for different datasets. Then we can find MetaHIN, which enhances the representations of new users or items, performs better than all existing meta-learning based methods, indicating that auxiliary data is important to capture fine-grained sequential intents of a user in recommendation. However, as we can see in Table 2, we can find that CBML achieves the best performance than MeLU, MAMO, and MetaHINs on all datasets, and the improvement of Hit@5, MRR@5, and NDCG@5 are in the range of 3%~8%. Here the improvement is computed as the difference of CBML and the best state-of-the-art method over the performance of the best state-of-the-art method on that metric, shown in percentage. The performance demonstrates the superiority of CBML, which can be explained as that cluster-based meta-learning method can ensure the shared knowledge among similar sessions be better

transferred and suitable initial parameters can be guided for each session. Since the existing meta-learning methods are designed to alleviate the cold-start problem and our model CBML outperforms all of them, it is confirmed that CBML is effective in alleviating the cold-start problem.

5.4 Influence of the Cluster-aware Parameter Gate

In this section, we aim to evaluate the effectiveness of the cluster-aware parameter gate, the usefulness of *session specific* representation and *cluster enhanced* representation as input of the gate (according to Formula 12), and the importance of feature-level transition patterns in user's preferences. Thus, we compare the performance of CBML with several variants of CBML, including

- **CBML-G** is the version that removes the cluster-aware parameter gate from CBML.
- **CBML-C** takes the *session specific* representation S_u as input of the parameter gate only, without considering the effects of clusters.
- **CBML-S** takes the *cluster enhanced* representation C_u as input of the parameter gate only, without considering the *session specific* representation.
- **CBML-F** only considers item-level transition patterns to capture sequential intents of a user, ignoring feature-level transition patterns.

In Table 3, we can find that CBML-F performs the worst, indicating that auxiliary information, such as labels, comments, etc, is also significant to capture more fine-grained sequential intents of a user to alleviate the cold-start problem of recommendation system. CBML-G performs worse than CBML-C and CBML-S because a global initialization is directly shared by all sessions, which confirms that the cluster-aware parameter gate is helpful to guide suitable initial parameters for each session. For the parameter gate, CBML-S performs better on Diginetica and CBML-C performs better on Yoochoose1/64 and Yoochoose1/4. This indicates that the session specific representation and the cluster enhanced representation may have varying importance on different datasets. By considering feature-level transition patterns in the base model and combining the two types of representations as input, CBML successfully achieves almost 1.43%∼5.87% improvements over CBML-*.

We can thus conclude that the cluster-aware parameter gate is necessary to improve the accuracy of recommendation by capturing suitable initial parameters for each session, and the *session specific* representation and the *cluster enhanced* representation are both required in cluster-aware parameter gate. In addition, feature-level transition patterns are also useful to alleviate the cold-start problem to some extent.

5.5 Influence of the Number of Clusters K

In this section, we evaluate the effectiveness of cluster numbers. Figure 3 shows the performance of our model with different numbers of clusters **K** on Diginetica, Yoochoose1/64, and Yoochoose1/4. We can see that when **K** is smaller than 4, the performance increases as the number of clusters increases. Then when **K** is greater than 4, the performance of CBML is better than baseline methods, improving almost 2.3%∼7.77% on Diginetica, 0.47%∼4.81% on Yoochoose1/64, and 3.73%∼7.95% on Yoochoose1/4. In addition, Figure 3 also shows that increasing the number of clusters does not change the performance very much, with **K** = 8 winning slightly. It shows that when the number of clusters is greater than 4, the performance is not sensitive to the number of clusters in reality.

5.6 Influence of the embedding size d

In Figure 4, we conduct several experiments to investigate the performance of varying the embedding size **d** ranging from 20 to 120 on the two real-world datasets. As we can see, when the embedding size is smaller than 100, the performance increases as the number of embedding sizes increases. This is because that the embedding size determines the complexity of the model, which can extract more sequential intents of a user from input sequences. However, once a proper value is exceeded, the performance of CBML does not grow very much, with **d** = 100 winning slightly. Thus, in this analysis, we set the number of embedding size **d** in our model CBML as 100 for the best performance.

5.7 Influence of the number of self-attention blocks l

In this section, we investigate how many levels of self-attention layers can benefit most for CBML. Figure 5 shows the experimental results of self-attention blocks with **l** ranging from 1 to 4. On all datasets, we find that when increasing the number of **l** (**l**≤ 2), the performance of CBML also improves. This is because more different types of sequential features can be captured from more self-attention layers. However, when **l** gets to properly value, increasing the number of self-attention layers may result in worse performance. The reason is that using more blocks (**l** >2) would make CBML easier to lose low-layer information.

6 CONCLUSION

In this paper, we propose a novel cluster-based meta-learning model (CBML) to address cold-start problem. Specifically, in order to avoid the impact from the divergence between different sessions and better transfer shared knowledge among similar sessions, we adopt a soft-clustering method and design a parameter gate to guide the initialization of the globally shared parameters to better serve for each session. Besides, in order to obtain more fine-grained sequential intents of a user, we apply two self-attention blocks to capture the transition patterns of sessions in both item and feature aspects. Finally, extensive experiments are conducted on two real-world datasets to verify our proposed model performs better than the state-of-the-art methods on two real public datasets.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China projects under grant numbers (No.61872258, No.61772356, No.62072125), the major project of natural science research in universities of Jiangsu province under grant number 20KJA520005, the priority academic program development of Jiangsu higher education institutions, young scholar program of Cyrus Tang Foundation, the Australian Research Council Discovery Projects under grant numbers (DP170104747, DP200103700), and the Australian Research Council Linkage Project under grant number LP180100750.

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *NIPS*. 3981–3989.

[2] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014).

[3] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation. In *KDD*. 688–697.

[4] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential Scenario-Specific Meta Learner for Online Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2895–2904.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, Vol. 70. PMLR, 1126–1135.

[6] Chelsea Finn, Kelvin Xu, and Sergey Levine. 2018. Probabilistic Model-Agnostic Meta-Learning. In *NIPS*. 9537–9548.

[7] Ahmed Frikha, Denis Krompaß, Hans-Georg Köpken, and Volker Tresp. 2020. Few-shot one-class classification via meta-learning. *arXiv preprint arXiv:2007.04146* (2020).

[8] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized news recommendation with context trees. In *RecSys*. 105–112.

[9] Qi He, Daxin Jiang, Zhen Liao, Steven CH Hoi, Kuiyu Chang, Ee-Peng Lim, and Hang Li. 2009. Web query recommendation via sequential query prediction. In *ICDE*. IEEE, 1443–1454.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR 2016*.

[11] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439* (2020).

[12] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *KDD*. 1073–1082.

[13] David Lenz, Christian Schulze, and Michael Guckert. 2018. Real-Time Session-Based Recommendations Using LSTM with Neural Embeddings. In *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th*. 337–348.

[14] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding. In *ICLR 2017*.

[15] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *KDD*. 1563–1573.

[16] Anjing Luo, Pengpeng Zhao, Yanchi Liu, Fuzhen Zhuang, Deqing Wang, Jiajie Xu, Junhua Fang, and Victor S. Sheng. 2020. Collaborative Self-Attention Network for Session-based Recommendation. In *IJCAI 2020*. 2591–2597.

[17] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing Dialogue Agents via Meta-Learning. In *ACL 2019*. 5454–5459.

[18] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141* (2017).

[19] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. 2018. Rapid adaptation with conditionally shifted neurons. In *ICML*. PMLR, 3664–3673.

[20] Kun Qian and Zhou Yu. 2019. Domain Adaptive Dialog Generation via Meta Learning. In *ACL 2019*. 2639–2649.

[21] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-Shot Learning. In *ICLR 2017*.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[23] Sujoy Roy and Sharath Chandra Guntuku. 2016. Latent factor representations for cold-start video recommendation. In *RecSys*. 99–106.

[24] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *ICML*. 1842–1850.

[25] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*. 4077–4087.

[26] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1199–1208.

[27] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *RecSyc*. ACM, 17–22.

[28] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*. 3630–3638.

[29] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2016. Collaborative filtering and deep learning based hybrid recommendation for cold start problem. In *2016 IEEE*. IEEE, 874–877.

[30] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI*, Vol. 33. 346–353.

[31] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*. 3940–3946.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML 2015*, Vol. 37. 2048–2057.

[33] Han-Jia Ye, Xiang-Rong Sheng, and De-Chuan Zhan. 2020. Few-shot learning with adaptively initialized task optimizer: a practical meta-learning approach. *Mach. Learn.* 109, 3 (2020), 643–664.

[34] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. In *SIGIR 2020*. ACM, 1921–1924.

[35] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. 2018. MetaGAN: An Adversarial Approach to Few-Shot Learning. In *NeurIPS 2018*. 2371–2380.

[36] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation.. In *IJCAI*. 4320–4326.

[37] Liang Zhao, Yang Wang, Daxiang Dong, and Hao Tian. 2019. Learning to Recommend via Meta Parameter Partition. *arXiv preprint arXiv:1912.04108* (2019).