Computer Science and Information Systems 21(1):1–20 https://doi.org/10.2298/CSIS230522067G

FSASA: Sequential Recommendation Based on Fusing Session-Aware Models and Self-Attention Networks

Shangzhi Guo[1], Xiaofeng Liao[1], Fei Meng[1], Qing Zhao[1], Yuling Tang[2], Hui Li[3] and Qinqin Zong[4,⋆]

[1] College of Computer Science, Chongqing University
Chongqing 400044, China
{20211401018g, xfliao, 20211401024g, 20211401020g}@cqu.edu.cn
[2] Hunan Creator Information Technologies CO., LTD.
Changsha 410000, China
yuling.tang@chinacreator.com
[3] Jiangxi Institute of Land and Space Survey and Planning
Nanchang 330000, China
lihcool@126.com
[4] Jiangxi Biological Vocational College
Nanchang 330200, China
honeybabyqinqin@gmail.com

Abstract. The recommendation system can alleviate the problem of “information overload”, tap the potential value of data, push personalized information to users in need, and improve information utilization. Sequence recommendation has become a hot research direction because of its practicality and high precision. Deep Neural Networks (DNN) have the natural advantage of capturing comprehensive relations among different entities, thus almost occupying a dominant position in sequence recommendation in the past few years. However, as Deep Learning (DL)-based methods are widely used to model local preferences under user behavior sequences, the global preference modeling of users is often underestimated, and usually, only some simple and crude user latent representations are introduced. Therefore, this paper proposes a sequential recommendation based on Fusing Session-Aware models and Self-Attention networks (FSASA). Specifically, we use the Self-Attentive Sequential Recommendation (SASRec) model as a global representation learning module to capture long-term preferences under user behavior sequences and further propose an improved session-aware sequential recommendation model as a local learning representation module from user model the user’s dynamic preferences in the historical behavior, and finally use the Gated Recurrent Unit (GRU) module to calculate their weights. Experiments on three widely used recommendation datasets show that FSASA outperforms state-of-the-art baselines on two commonly used metrics.

Keywords: Recommendation Systems, Sequential Recommendation, Session-Aware Recommendation, Self-Attention, Gated Recurrent Unit.

⋆ Corresponding author

1. Introduction

In recent years, with the rapid development of the Internet, especially the mobile Internet, Internet information has also shown an explosive growth trend. Faced with massive amounts of information, the time and cost for users to obtain the content they need have increased significantly. The recommendation system, as an effective means to solve the problem of information overload, has become the core of many e-commerce and multimedia platforms [60]. Personalized recommendation services can help the platform to attract users' attention, increase the number of user visits, and provide a steady stream of power for the development of network platforms. Its commercial value has also attracted the attention of industry and academia [19].

Collaborative Filtering (CF) is the most widely used recommendation system in the early stage. The core idea is to synthesize the explicit feedback information of users and items and filter out items the target users may be interested in for recommendation [42]. Different from CF, the goal of sequence recommendation is to combine a personalized model of user behavior (based on historical information) with some concept of "context" based on the user's recent behavior by understanding and analyzing the user's interaction history as a sequence information, to push the matching items of user interest [46, 52, 35]. Early work on sequential recommendation usually uses Markov Chain (MC) [8, 7, 10, 11], but its disadvantage is also obvious. Due to the Markov property, it is assumed that the current interaction only depends on one or a few recent interactions. Only short-term dependencies are captured, while long-term dependencies are ignored. As one of the important research directions of sequence recommendation, session-aware recommendation takes each session as the basic input unit, which can capture the user's short-term preference and the dynamic preference reflected by the interest transfer between sessions, thereby improving the accuracy and timeliness of recommendation [43, 49]. Deep neural network has the natural advantage of capturing the comprehensive relationship between different entities, which can alleviate the problem of insufficient expressive ability of traditional recommendation models, so it has almost occupied the dominant position in sequence recommendation in the past few years [28, 44, 61, 24, 25, 9, 4]. However, most of DNN-based methods also do not pay enough attention to the long-term relationship between sequences, and the user's global preference modeling is often underestimated. Attention mechanisms, which can reveal syntactic and semantic patterns between words in a sentence, have also become an important component of sequence recommendation [23, 2, 45, 30]. Among them, SASRec [18] stacks multiple self-attention blocks, which can effectively capture the long-term preferences of users within a sequence. Now that users' long-term and short-term preferences have been well explored in previous studies, an intuitive way to develop sequential recommendation methods is to model local dynamic preferences and combine them with global preferences to more comprehensively predict users' true preferences [23, 38, 27, 54, 59].

Inspired by the above work, this paper proposes a novel solution named sequential recommendation based on fusing session-aware models and self-attention networks (FSASA). FSASA can consider the user's long-term static preference and short-term dynamic preference simultaneously and more fully express the user's real intention. Specifically, our model contains three main components, a global representation learning module, a local representation learning module, and a GRU module. For global representation learning, we follow SASRec [18] based on the self-attention mechanism because it achieves excellent performance in capturing users' long-term preferences. For global representation learning, we introduce implicit features from users' historical behaviors based on sequential recommendation [43] to model users' dynamic preferences accurately. Finally, we use GRU to balance the weights of the global representation learning module and the local representation learning module. In addition, we conduct a comprehensive ablation study to show the impact of crucial modules and parameters on recommendation performance.

The main contributions of the proposed FSASA are as follows:

- We propose a novel sequential recommendation based on session-aware models and self-attention networks to capture the dynamic preferences beneath users' behavior sequences, and improving recommendation performance.
- We design a session-aware local representation learning module for mining the implicit features in the user's historical behavior to model the user's dynamic preferences accurately.
- The GRU module is used to balance the contribution of the global representation learning module and the local representation learning module and more comprehensively predict the user's real preferences.
- To verify the performance of FSASA, we also conducted simulation experiments on three commonly used datasets. Experimental results show that FSASA significantly outperforms five state-of-the-art baselines. We also perform ablation studies and discuss details of local and gating units.

The rest of this paper is structured as follows. In the next section, a brief review is given of recent investigations on general recommendation, sequential recommendation, and session-aware recommendation. We propose sequential recommendation based on fusing session-aware models and self-attention networks in Section 3. Section 4 describes experiments based on three real datasets and analyzes the results. Finally, Section 5 presents the main conclusions and future work.

2. Related Work

In this section, we will briefly review several lines of works closely related to ours, including general recommendation, sequential recommendation, and session-aware recommendation, respectively, and point out the relationship and differences between our FSASA and those works.

2.1. General Recommendation

Collaborative filtering [42] was the most widely used recommendation algorithm in the early days. It mainly finds users' preferences through deep mining of their past behavioral data, groups users based on different preferences, and recommends items with similar tastes to other users in the group [20, 6, 51, 55]. Among them, the matrix decomposition [31, 21, 12, 1, 16, 62] algorithm uses Singular Value Decomposition (SVD), Eigenvalue Decomposition (ED), and other methods to decompose the co-occurrence matrix to generate an implicit vector for the user and the project, respectively, and uses the implicit vector to represent the user's interest and the project's attributes, to dig the deep potential relationship between the user and the project—excellent performance in user rating prediction task. Collaborative filtering and matrix decomposition algorithms only utilize user-project interaction information. At the same time, Logistic Regression (LR) [39] can integrate user portrait features, item attributes, and context information, transform features into numerical vectors, input them into the network for training, learn the weight of each feature, and predict the probability of positive samples in the output layer. However, LR has limited characterization ability and does not carry out a cross combination of multiple features, which affects prediction accuracy. Rendle [37] proposed a Factorization Model (FM) by adding a second-order cross-feature combination based on logistic regression. The Facebook team [13] combined the gradient lifting decision tree with logistic regression and used the combined model to complete the recommendation task.

Combining deep learning and recommendation system can alleviate the problem of insufficient expression ability of the traditional recommendation model. Multi-Layer Perceptron (MLP) is a neural network with feed-forward structure. The data flows through the input layer and multiple hidden layers into the output layer to calculate the final result. The recommendation system often uses it to mine the crossover of high-order features and learn potential data patterns [40, 5, 3, 29].

2.2. Sequential Recommendation

Different from traditional collaborative filtering and content filtering-based recommendation systems, sequential recommendation attempts to model and understand user sequential behavior, interactions between users and items, and the evolution of user preferences and item popularity over time [50]. Early works on sequential recommendation usually use Markov chains, and MC's natural advantage in modeling sequential dependencies provides an intuitive solution for sequential recommendation [8, 7, 10, 11]. Nevertheless, its shortcomings are also obvious. Due to the Markov property, it is assumed that the current interaction only depends on one or a few recent interactions, so it can only capture short-term dependencies and ignore long-term dependencies.

As mentioned in the previous section, deep neural networks have the natural advantage of capturing comprehensive relations among different entities (e.g., users, items, interactions). They thus have almost dominated sequential recommendation in the past few years. Recurrent Neural Network (RNN) is a deep network structure commonly used to process time series data. RNN can perform feed-forward calculation, maintain the information of the previous moment, and use historical state data and current state to predict output so that it can process sequence data such as text and audio [28, 44, 61, 24, 25]. To solve the problem of information loss caused by too long time intervals and the problem of gradient disappearance and explosion, RNN has also constructed new variants: Long Short-Term Memory network (LSTM) [9] and gated recurrent unit [4]. Our work uses GRU to fuse two representation learning modules, which will be discussed further in Section 3. RNN is not perfect, and it may only capture point dependencies and ignore set dependencies (e.g., several interactions collaborating to influence the next one). Since Convolutional Neural Networks (CNN) do not have strong sequential assumptions about the interactions in sequences, the above-mentioned shortcomings of RNN in sequence recommendation can be compensated to some extent. The CNN first puts all the embedding elements of the interaction into a matrix, then uses this matrix as an "image" in time and latent space, and finally, learns sequential patterns as local features of the image, using convolutional filters for subsequent recommendations [47, 58, 57]. However, due to the limited size of filters used in CNN, CNN-based sequence recommendations cannot effectively capture long-term dependencies, which limits their applications. With the rapid development of Graph Neural Networks (GNN), numerous researchers utilize GNN to model and capture complex transition sequences of user-item interactions [26, 34, 52, 53]. This method fully exploits the advantages of GNN to capture complex relations in structured relational datasets, showing great potential for explainable recommendation, which is still in the early exploration stage.

The attention model is also often used in sequence recommendation to emphasize those genuinely relevant and essential interactions in the sequence while ignoring those interactions that are irrelevant to the next interaction, allowing the model to focus on more important information, reducing the impact of data noise on the impact of the results [18, 23, 2, 45, 30]. In this paper, we base the global representation learning module on the Self-Attention Sequential Recommendation (SASRec) model [18], which is an excellent sequential recommendation model. Note that [23] bases the local representation learning module on the SASRec model, which is similar to our FSASA and will be further discussed in Section 4.

2.3. Session-aware Recommendation

Sequential recommendation considers that all historical interaction information is equally important for predicting user's current preference. However, user preference may change over time, which is dynamic rather than static. Therefore, Session-Based Recommendation Systems (SBRS) have been proposed to bridge these gaps in recent years. SBRS takes each session as the basic input unit, which can capture the user's short-term preferences and the dynamic preferences reflected by the interest transfer between sessions, thereby improving the accuracy and timeliness of recommendations [49]. Unlike session-based, the session-aware recommendation is a method that uses the relationship between sessions for each user and makes recommendations by structurally decoupling long-term and short-term preferences from a slightly more diverse perspective [22]. [17] propose an early SBRS emphasizing the importance of considering recently observed user behavior when making recommendations. [36] proposed one of the earliest deep learning techniques for the session-aware recommendation, where the authors used two parallel GRU layers to model information across sessions. In the same year, Ruocco et al. [41] proposed the IIRNN model, which, like [36], uses the RNN architecture and extends session-based techniques to model inter-session and intra-session notifications. RNN were later also used in NSAR models [33] to encode session patterns combined with user embeddings to represent long-term user preferences across different sessions. Hu et al. [15] combine inter-session and intra-session context with a joint context encoder for item prediction. In [56], the authors utilize a two-layer hierarchical attention network to model short-term and long-term user interests. In [14], the authors are inspired by language modeling methods such as word2vec to treat items as words and recommend related items based on contextual information.

As the session-aware method is widely used in local and dynamic preference modeling under user behavior sequences, the user's global and static preference modeling is often underestimated. Usually, only some simple and crude user potential representations are introduced. In our work, the respective advantages of session-aware and self-attention-based sequential recommendation methods are fully combined to model the user's short-term and long-term preferences, respectively. This means that our FSASA can more comprehensively represent the user's true intent, which we discuss further in the next section.

3. Proposed Method: FSASA

This section proposes our FSASA, i.e., sequential recommendation based on fusing session-aware models and self-attention networks. For sequential recommendation, we are given a user's action sequence $\mathcal{S}^u = \{\mathcal{S}_1^u, \mathcal{S}_2^u, ..., \mathcal{S}_t^u, ..., \mathcal{S}_{|\mathcal{S}^u|}^u\}$, $u \in \mathcal{U}$, $\mathcal{S}_t^u \in \mathcal{I}$, where $\mathcal{U}$ denote a set of users and $\mathcal{I}$ denote a set of items. Given the interaction history $\mathcal{S}_t^u$, sequential recommendation aims to predict the item that user u will interact with at time step $\mathcal{S}_{t+1}^u$. In this paper, we use capital letters in bold to denote matrices and their lowercase form to denote the corresponding row vectors.

3.1. Global Representation Learning

First of all, we fix the input sequence of each user u by extracting his/her latest n behaviors, which is abbreviated as $\mathcal{S}^u = \{s_1, s_2, ..., s_n\}$, where n represents the maximum length that can handle. If the sequence length exceeds n, we consider the most recent n actions. If the sequence length is less than n, we repeatedly add a padding item **0** on the left until the length is n. Let $\mathbf{M} \in \mathbb{R}^{|\mathcal{I}| \times d}$ denote the learnable item embedding matrix with d as the latent dimensionality. We can then represent the input sequence as an embedding matrix $\mathbf{E} \in \mathbb{R}^{n \times d}$, where $\mathbf{E}_i = \mathbf{M}_{s_i}$. A constant zero vector **0** is used as the embedding for the padding item.

Following [18], since the self-attention model does not include any recurrent or convolutional module, it is unaware of the positions of previous items. Hence we inject a learnable position embedding matrix $\mathbf{P} = [p_1; p_2; ...; p_n] \in \mathbb{R}^{n \times d}$ to the input embedding matrix $\mathbf{E} \in \mathbb{R}^{n \times d}$, and obtain an input matrix $\mathbf{X}^{(0)} = [x_1; x_2; ...; x_n] \in \mathbb{R}^{n \times d}$ for the self-attention network:

$$\mathbf{x}_i^{(0)} = \mathbf{m}_{s_i} + \mathbf{p}_i, i \in \{1, 2, ..., n\} \tag{1}$$

Then, we feed the sequence $\mathbf{X}^{(0)} \in \mathbb{R}^{n \times d}$ into a series of stacked self-attention blocks (SABs). The output of the b-th block is as follows:

$$\mathbf{X}^{(b)} = SAB^{(b)}\mathbf{X}^{(b-1)}, b \in \{1, 2, ..., B\} \tag{2}$$

Omitting the normalization layers with residual connection, each self-attention block can be viewed as a self-attention layer $SAL(\cdot)$ followed by a feed-forward layer $FFL(\cdot)$ as follows:

$$SAB(\mathbf{X}) = FFL(SAL(\mathbf{X})) \tag{3}$$

$$\mathbf{X}^{'} = SAL(\mathbf{X}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\Delta \cdot \mathbf{V} \tag{4}$$

$$FFL(\mathbf{X}^{'}) = ReLU(\mathbf{X}^{'}\mathbf{W}_1 + \mathbf{1}^T\mathbf{b}_1)\mathbf{W}_2 + \mathbf{1}^T\mathbf{b}_2 \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the position-aware input matrix, $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ and $\mathbf{V} = \mathbf{X}\mathbf{W}_V$ with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$ are the projected query, key and value matrices, respectively, to improve the flexibility. Note that $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{1 \times d}$ are weights and biases for the two layers of convolution, $\mathbf{1}$ is a unit row vector of size $1 \times n$ and Δ is the causality mask (i.e., a unit lower triangular matrix of size $n \times n$), to preserve the transitions from previous steps only.

[23] uses a simple location-based attention mechanism as a global representation learning module to model the user's long-term static preferences. Its performance is not as good as that of SASRec, that stacks multiple self-attention blocks. It is shown in [18] that hierarchy is important for global representation. Specifically, self-attention blocks at the bottom tend to capture long-term dependencies, while higher blocks may focus on more recent dependencies. In this module, we use the bottom self-attention block $SAB^{(1)}(\cdot)$ as the global representation learning module of FSASA.

3.2. Local Representation Learning

While the self-attention block at the top of [18] can also be used to model a user's short-term dynamic preferences, in many online services, user interactions are often grouped by sessions where preferences are likely to be shared, and this is where session-aware is needed to establish connections for each user's session. Inspired by [46], we added user rating information embedding based on [43] to accurately model short-term dynamic preferences of users. Note that the local representation learning module is independent of the global one.

The simplest way to distinguish sessions within item sequences is to insert a separator between item sequences [27]. A learnable extra token called Session Token (ST) is inserted between sessions as if it were an item embedding. Unlike the fill marker, it is not excluded from attention, and it has the effect of moving the embedding one position per session. The advantage of this method is that it can indicate at inference time whether the input is a new session or not.

User rating information is one of the important criteria for modeling user preferences, but most current recommendation algorithms define rating as positive feedback, which is unreasonable. For a user, scoring an item only means that the user has browsed the item rather than that the user is interested in the item. For example, suppose a user gives an item a shallow score. In that case, the user is not interested in the item, and the recommendation system should recommend fewer such items.

In FSASA, the rating information is fine-grained, and the learnable Rating Segment Embedding (RSE) is used, representing the session's importance and providing a sequence hierarchy. Note that, similar to session tokens, scoring information can also indicate whether it is a new session or not at inference time. For p-th item i in j-th session of a user, our input representation becomes: $x = IE_i + PE_p + RSE_j$, where IE is an item embedding, PE is a positional embedding from BERT, and RSE is a session segment embedding. The maximum number of sessions is limited so that only the most recent m sessions are considered. As in the implementation of positional embedding, ordinals are attached in the most recent order and padding is filled to match the model input length L.

For a timestamp t, we define a Temporal Encoding (TE) as follows:

$$TE(t) = [\cos(\omega_1 t + \theta_1) \cdots \cos(\omega_{d_T} t + \theta_{d_T})]^\top \tag{6}$$

where d_T is a temporal dimension, and ω_i, θ_i are learnable parameters. We concatenate temporal encoding vectors **t** to the input representation $\mathbf{X}$, which gives us a Temporal Self-Attention (TAS) as follows:

$$TSA(\mathbf{X},\mathbf{t}) = softmax(\frac{[\mathbf{Xt}][\mathbf{Xt}]^\top}{\sqrt{d_\mathbf{X} + d_\mathbf{t}}})\mathbf{X} \tag{7}$$

where $d_\mathbf{X}$ is an input dimension of $\mathbf{X}$. Here we can see that the attention weight a_{ij} between (x_i, t_i) and (x_j, t_j) is calculated as:

$$a_{ij} = x_i^\top x_j + TE(t_i)^\top TE(t_j) \tag{8}$$

The weight becomes sum of self-attentiveness and temporal attentiveness [54]. For multi-layered and multi-headed Transformer layers, we concatenate TE on each layer and head. Note that TE can be trained on each layer or head separately, but empirically no significant improvements were found.

[23] uses SASRec as a local representation learning module to model users' short-term preferences, while SASRec is better at modeling recent activities of sparse datasets, and it is difficult to model normal or dense Recent activity on the dataset accurately. FSASA uses BERT-STR as a local representation learning module and adds user rating information. Thanks to the excellent representation ability of session-aware, it can accurately model the short-term dynamic preferences of users.

The input representation layer including all proposed methods is shown in Figure 1. The rest part of the model is identical to BERT4Rec [46]. Note that the difference from SASRec [18], which uses an autoregressive decoder, is that information other than item embedding such as positional embedding, session segment embedding, and temporal encoding can be utilized at inference time for the to-be-predicted item.

3.3. Gating Unit

To combine the local representation and the global representation, we may naturally think of concatenation or summation. Many researchers suggest a weighted summation to balance the two representations by considering the consistency of the item lists (corresponding to the sequences in our case), which performs better in their cases. Inspired by [23, 4], we use GRU to combine the weights of global and local representation learning module, and the fusion equation is as follows:

$$x(t) = x_{global} \otimes r + x_{local} \otimes (1 - r) \tag{9}$$

GRU is a variant of traditional RNN. Like LSTM, it can effectively capture the semantic association between long sequences and alleviate the phenomenon of gradient disappearance or explosion. At the same time, its structure and calculation are simpler than LSTM. Its core structure is composed of update gate $z(t)$ and reset gate $r(t)$:

$$z(t) = \sigma(\mathbf{W}_z \cdot [h_{t-1}, x_t]) \tag{10}$$

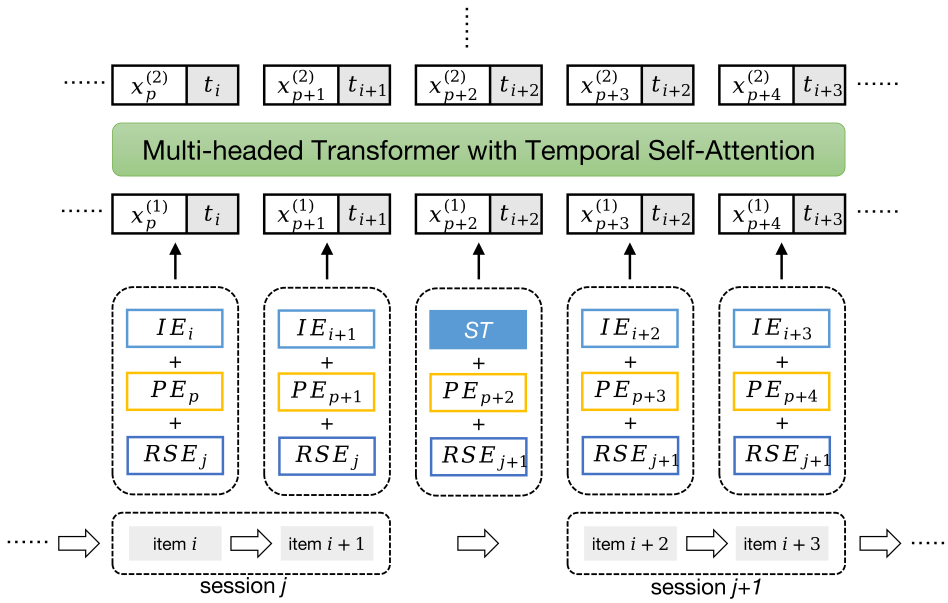


Fig. 1. Input layer

$$r(t) = \sigma(\mathbf{W}_r \cdot [h_{t-1}, x_t]) \tag{11}$$

where h_{t-1} denote the hidden state layer output of the previous time, in FSASA represents the user preference learned at the previous time, and x_t denote the input at the current time, which in FSASA represents the current local/short-term preference. After getting the gating signal, first use the reset gating to get the data after "RESET":

$$h_{t-1}^{'} = h_{t-1} \odot r_t \tag{12}$$

The representative controls how much information from the last time can be used. Then use this reset $h_{t-1}^{'}$ to perform basic RNN calculations, i.e., splicing with x_t for linear change, and after $tanh$ activation obtain $h_t^{'}$:

$$h_t^{'} = \tanh(\mathbf{W} \cdot [h_{t-1}^{'}, x_t]) \tag{13}$$

The gate value z_t of the last update gate will act on the $h_t^{'}$, and $1 - z_t$ will act on h_{t-1}, and then add the results of the two to get the final hidden state output h_t:

$$h_t = z_t \odot h_t^{'} + (1 - z_t) \odot h_{t-1} \tag{14}$$

The range of the gating signal (i.e., z_t) is 0 to 1, the closer the gating signal is to 1, the more data is "remember", and the closer to 0 is the more "forget". FSASA uses GRU to "forget" the unimportant information in the user sequence, "remember" the important information in the user sequence, and more comprehensively and accurately represent the user's true intention.

We abandon the way of compressing the long-term or short-term preference representation as the initial hidden state h_0, as this would compress the representation vector to very low dimensions and lose information.

[23] uses item similarity models to add user's uncertain intention information to make recommendations, although it can to a certain extent, the recommendation performance in the absence of user sequence information is improved. However, at the same time, noise may be introduced to affect the overall recommendation performance. FSASA directly uses GRU to balance the weight of the global representation module and the local representation module. Although the improvement of the recommendation performance is limited, it will certainly not have a negative impact on the recommendation performance and has robust scalability.

4. Performance Evaluation

To verify the performance of our proposed FSASA, this section introduces the details of the datasets, evaluation indicators, baseline methods, and parameter settings used in the simulation experiments. It conducts many ablation experiments to explore the impact of relevant hyperparameters on the performance of FSASA.

4.1. Datasets Description and Preprocessing

- **Steam[5][18, 48, 32]:** This dataset contains data from October 2010 to January 2018 of Steam, a large online video game distribution platform. These include 2,567,538 users, 15,474 games, and 7,793,069 user reviews. The dataset also provides rich hidden information such as user's game time, price information, purchase information, media ratings, categories, product bundles, developers, etc.
- **ML-1M[6]:** The dataset contains 1 million ratings of 4,000 movies from 6,000 users. This data includes movie ratings, movie metadata (genre and year), and user demographic data (age, zip code, gender, occupation, etc.).
- **ML-20M[7]:** The dataset contains 20,000,263 ratings and 465,564 tags for 27,278 movies from 138,493 users. Users are randomly selected, and each selected user has rated at least 20 movies. There is no demographic information, and each user is only given an ID, and no other private information is involved.

For sequential recommendation, we preprocess these datasets as follows:

1) To improve the dataset's quality, we delete items with less than 5 interactions and delete users with less than 5 interactions; 2) When users have no new interactions within a day, use unix timestamp units to divide sessions. Each user has at least 2 sessions, each session contains at least 2 items, and only uses 200 recently interacted items; 3) In the preprocessing step, each comment or rating information is considered as a There is a hidden positive interaction, so this paper retains the rating in the data set when constructing the training set, which is an important improvement of this paper. For each user, the last

[5] https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data

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

[7] https://grouplens.org/datasets/movielens/

item is used as the test item, the second closest item is used as the validation, and the remaining items are used as the training set.

The statistics of the processed datasets are summarized in Table 1.

Table 1. Dataset statistics after preprocessing

Dataset	#Users	#Items	#Interactions	Avg. Length	Density
Steam	6330	4331	49163	7.77	0.18%
ML-1M	1196	3327	158496	132.52	3.98%
ML-20M	23404	12239	1981866	84.68	0.69%

4.2. Evaluation Metrics

We evaluate the recommendation performance via two standard metrics, i.e., recall (Recall@10, R@10) and normalized discounted cumulative gain (NDCG@10, N@10). Recall is how much of the information the user interested is predicted. The NDCG is a standardized DCG that considers the list of recommendations and the number of truly valid results in each search. The definition of Recall@10 and NDCG@10 are as follows:

$$Recall@10 = \frac{\sum_{u \in U} \mid R(u) \cap T(u) \mid}{\sum_{u \in U} \mid T(u) \mid} \tag{15}$$

$$NDCG@10 = \frac{DCG@10}{IDCG@10} \tag{16}$$

$$DCG@10 = \sum_{i=1}^{10} \frac{2^{rel_i} - 1}{\log_2 (i+1)} \tag{17}$$

Where $R(u)$ represents the $Top-10$ recommendation list made to the user according to the user's behavior in the training set, and $T(u)$ represents the item set actually selected by the user after the system recommends the item to the user. rel_i stands for correlation degree of items in position i, IDCG@10 stands for ideal DCG, i.e., DCG under perfect result.

4.3. Baselines

To verify the effectiveness of FSASA, we compare it with the following five representative baselines:

- **SASRec[18]:** It addresses the sequential recommendation problem by introducing a self-attention mechanism that adaptively assigns weights to previous entries at each time, tending to consider long-term dependencies on dense datasets while focusing on recent activities on sparse datasets. It is also a global representation learning module in our FSASA.

- **B4Rec[46]:** It employs deep bidirectional self-attention to model user behavior sequences. Each layer utilizes all the information of the previous layer and can capture the information of the entire field.
- **BERT-ST[43]:** It proposes three ways to utilize session information in the BERT-based model to improve sequential recommendation performance. We use the method with the best comprehensive performance among them to compare with FSASA.
- **BERT-STR:** The rating information is added based on BERT-ST, which is also the local representation learning module in our FSASA.
- **FISSA[23]:** From the global and local time series perspective, SASRec is used as a local learning module, and then a position-based attention layer is used as a global module, and their weights are balanced by gating.

4.4. Implementation Details

We perform all the experiments on a single server with ADM Ryzen5 3600x CPU and Nvidia 3070 GPU. The software environment includes Cuda 11.4, Cudnn 8.2, Miniconda 3, Python 3.7, deep learning framework Pytorch 1.10, and tensorboardx 2.5. All hyper-parameters were tuned through grid search, and we report the one with the best performance in the final result.

The FSASA model comprised [18] and improved [43], respectively, to act as a global representation learning module and a local representation learning module. The former is used to capture long-term dependencies between items, while the latter is used to obtain short-term associations between items, using GRUs to balance their weight. We use the AdamW optimizer to calculate and update the model parameters to minimize the objective function, the learning rate is initialized to 0.001, and the dropout is set to 0.2 to avoid model overfitting. Limited by hardware conditions, the layer of SASRec is uniformly set to 1, and the parameter settings for different datasets are shown in Table 2.

Table 2. Initialization parameters for the three datasets

Dataset	MaxLength	Layers	Hidden_dim	Heads	Batch_size
Steam	15	2	128	2	1024
ML-1M	200	2	256	2	128
ML-20M	100	4	256	4	128

4.5. Overall Performance Comparison

Table 3 presents the recommendation performance of all methods on the three datasets. As we can see, here following observations would be found:

Compared with all baselines, our FSASA achieves the best performance on all three datasets, which clearly demonstrates the superiority of FSASA (note that the Steam dataset does not provide user rating information, so we use GRU to fuse the original SASRec and BERT-ST, but its performance still has a noticeable improvement). The second best performance is obtained by BERT-ST or BERT-STR, which is consistent with the observations of previous studies [46, 43, 23], which also show the advantages of session-aware in modeling user dynamic preferences. In addition, BERT-STR performs slightly better than BERT-ST, illustrating that adding user rating information helps to accurately model truly relevant and important interactions in user sequences.

It is worth noting that FISSA uses a structure similar to FSASA, but its recommendation performance is not outstanding, only better than SASRec and basic B4Rec. We analyze the reasons from the components of FISSA and FSASA: 1) FISSA uses SASRec as a local representation learning module to model users' short-term preferences, while SASRec is better at modeling recent activities of sparse datasets, and it is difficult to model normal or dense Recent activity on the dataset accurately. FSASA uses BERT-STR as a local representation learning module and adds user rating information. Thanks to the excellent representation ability of session-aware, it can accurately model the short-term dynamic preferences of users; 2) FISSA uses a simple location-based attention mechanism as a global representation learning module to model the user's long-term static preferences. Its performance is not as good as that of SASRec, that stacks multiple self-attention blocks; 3) FISSA uses item similarity models to add user's uncertain intention information to make recommendations, although it can to a certain extent, the recommendation performance in the absence of user sequence information is improved. However, at the same time, noise may be introduced to affect the overall recommendation performance. FSASA directly uses GRU to balance the weight of the global representation module and the local representation module. Although the improvement of the recommendation performance is limited, it will certainly not have a negative impact on the recommendation performance and has robust scalability.

4.6. Ablation Study

We discuss the impact of relevant parameters on the performance of FSASA in this section.

(1) Effect of representation learning ratio

To explore the impact of the representation learning module in FSASA on recommendation performance, we manually set the proportion of the global representation learning module involved in FSASA. As shown in Figure 2, on the whole, with the increase of the proportion of the global representation learning module involved, the performance of FSASA fluctuates slightly before 0.5, and the performance decreases with the increase of the proportion after 0.5. Between 0.3 to 0.5, the performance of FSASA is optimal. It shows that the session-aware-based local representation learning module we proposed is dominant in FSASA. However, it also needs the assistance of the global representation learning module to more fully represent the user's true intentions.

(2) Effect of gating unit

To explore the impact of the gating unit in FSASA on the recommendation performance, we fixed the proportion of the global representation learning module and the local representation learning module at 0.5 for comparative experiments. As shown in Figure 3, the two metrics of FSASA using GRU outperform FSASA without GRU on all three datasets. The analysis in 4.6.1 shows that although the local representation learning module is dominant in FSASA, the larger the proportion, the better. GRU needs to be dynamically adjusted according to different scenarios to give full play to the greatest advantages of FSASA.

Table 3. Recommendation performance of FSASA and five baselines on three datasets. The best performing method in each row is bolded, and the second best performing method in each row is underlined.

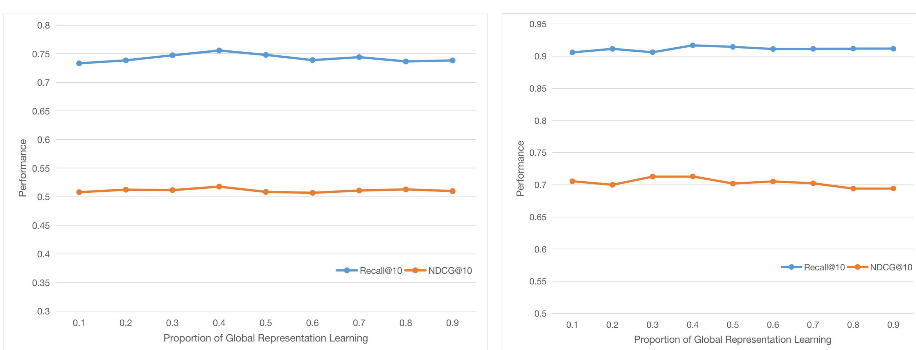
Dataset	Metric	Item	SASRec	B4Rec	BERT-ST	BERT-STR	FISSA	FSASA
Steam [1]	R@10	Ran[2]	0.7834	0.7987	0.8120	\	0.8015	**0.8139**
		Pop [3]	0.5523	0.5670	0.6030	\	0.5735	**0.6056**
		All [4]	0.5164	0.5313	0.5616	\	0.5425	**0.5718**
	N@10	Ran	0.6726	0.6915	0.7093	\	0.7002	**0.7165**
		Pop	0.5007	0.5196	0.5596	\	0.5316	**0.5666**
		All	0.4610	0.4782	0.5187	\	0.4885	**0.5282**
ML-1M	R@10	Ran	0.7199	0.7341	0.7291	0.7400	0.7380	**0.7558**
		Pop	0.4189	0.4725	0.4841	0.4849	0.4731	**0.5192**
		All	0.1480	0.1129	0.1731	0.1697	0.1528	**0.1811**
	N@10	Ran	0.4962	0.5100	0.5113	0.5115	0.5112	**0.5177**
		Pop	0.2674	0.3011	0.3105	0.3093	0.3021	**0.3417**
		All	0.0742	0.0508	0.0838	0.0873	0.0806	**0.0915**
ML-20M	R@10	Ran	0.9014	0.9053	0.9114	0.9113	0.9083	**0.9168**
		Pop	0.4370	0.4729	0.4799	0.4822	0.4735	**0.5195**
		All	0.1389	0.1381	0.1439	0.1393	0.1499	**0.1555**
	N@10	Ran	0.6954	0.6944	0.6910	0.6964	0.6998	**0.7130**
		Pop	0.2839	0.3051	0.3125	0.3155	0.3062	**0.3588**
		All	0.0707	0.0724	0.0754	0.0755	0.0785	**0.0846**

[1] Note that the Steam dataset does not provide user ratings.

[2] Randomly select 100 non-repeated items from the items that the user unclicked for recommendation.

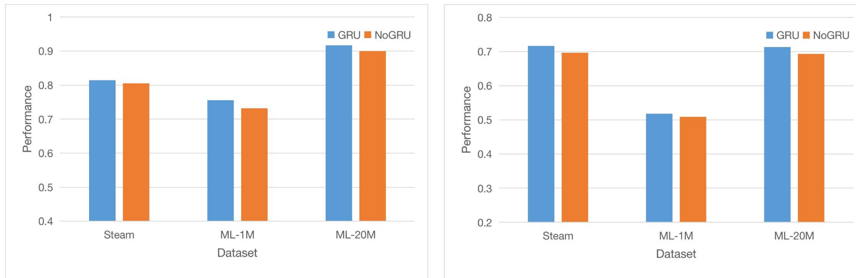
[3] Sort the item list in descending order, and continuously extract 100 unclicked and non-repeated items for recommendation.

[4] The recommended label space is all items.



(a) ML-1M Dataset (b) ML-20M Dataset

Fig. 2. Effect of representation learning ratio



(a) Recall@10 (b) NDCG@10

Fig. 3. Effect of gating unit

(3) Effect of rating

We add user rating information based on [43] to serve as the local representation learning module of FSASA. In order to verify the rationality of our work, we compare it with FSASA using the BERT-ST module. As shown in Figure 4, both metrics of FSASA using BERT-STR slightly outperform FSASA using BERT-ST on both datasets. Because BERT-STR adds the implicit feature of user rating information, it can more accurately model the user's short-term dynamic preferences.

(a) Recall@10 (b) NDCG@10

Fig. 4. Effect of rating

5. Conclusions And Future Work

With the rapid development of information technology, recommendation system plays an important role in alleviating the problem of information overload. Compared with traditional recommendation algorithms, deep learning enhances the model's scalability and representation ability, allowing the model to incorporate more diverse features. However, users' global preferences and modeling are often underestimated. Therefore, in this paper, we propose a new sequential recommendation method (FSASA) based on the session-aware model and self-attention network to capture global preferences and dynamic preferences under user behavior sequences. Precisely, our model consists of three main components, the local presentation learning module, the global presentation learning module, and the GRUs module. We used the SASRec model as a global presentation learning module to capture long-term preferences under user behavior sequences and proposed an improved session-aware sequential recommendation model as a local learning presentation module to model users' dynamic preferences from users' historical behaviors. Finally, the Gated Recurrent Unit module is used to balance the weights of the two modules. We compared the FSASA model with various mainstream algorithms on three publicly available data sets. The results showed that the FSASA model was superior to the existing mainstream recommended model. We also conducted many ablation experiments and quantitative studies to demonstrate the rationality of the FSASA model. In future work, we plan to extend the model by incorporating rich contextual information, exploiting session information more thoroughly and consistently to predict users' future preferences accurately.

Acknowledgments. This work is supported by the Postgraduate Scientific Research and Innovation Foundation of Chongqing under Grant No. CYB22064.

References

1. Chen, C., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Efficient neural matrix factorization without sampling for recommendation. ACM Transactions on Information Systems (TOIS) 38(2), 1–28 (2020)
2. Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.S.: Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 335–344 (2017)
3. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 7–10 (2016)
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
5. 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)
6. Cui, Z., Xu, X., Fei, X., Cai, X., Cao, Y., Zhang, W., Chen, J.: Personalized recommendation system based on collaborative filtering for iot scenarios. IEEE Transactions on Services Computing 13(4), 685–695 (2020)
7. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence. pp. 2069–2075. ACM (2015)
8. Garcin, F., Dimitrakakis, C., Faltings, B.: Personalized news recommendation with context trees. In: Proceedings of the 7th ACM Conference on Recommender Systems. pp. 105–112 (2013)
9. Graves, A., Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks pp. 37–45 (2012)
10. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: Proceedings of the eleventh ACM conference on recommender systems. pp. 161–169 (2017)

11. He, R., McAuley, J.: Fusing similarity models with markov chains for sparse sequential recommendation. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 191–200. IEEE (2016)
12. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 549–558 (2016)
13. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al.: Practical lessons from predicting clicks on ads at facebook. In: Proceedings of the eighth international workshop on data mining for online advertising. pp. 1–9 (2014)
14. Hu, L., Cao, L., Wang, S., Xu, G., Cao, J., Gu, Z.: Diversifying personalized recommendation with user-session context. In: IJCAI. pp. 1858–1864 (2017)
15. Hu, L., Chen, Q., Zhao, H., Jian, S., Cao, L., Cao, J.: Neural cross-session filtering: Next-item prediction under intra-and inter-session context. IEEE Intelligent Systems 33(6), 57–67 (2018)
16. Hwang, W.S., Li, S., Kim, S.W., Lee, K.: Data imputation using a trust network for recommendation via matrix factorization. Computer Science and Information Systems 15(2), 347–368 (2018)
17. Jannach, D., Ludewig, M., Lerche, L.: Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Modeling and User-Adapted Interaction 27, 351–392 (2017)
18. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM). pp. 197–206. IEEE (2018)
19. Kim, D., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM conference on recommender systems. pp. 233–240 (2016)
20. Koren, Y., Bell, R.: Advances in collaborative filtering, recommender systems handbook, editör: Ricci, f., rokach, l., shapira, b (2015)
21. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
22. Latifi, S., Mauro, N., Jannach, D.: Session-aware recommendation: A surprising quest for the state-of-the-art. Information Sciences 573, 291–315 (2021)
23. Lin, J., Pan, W., Ming, Z.: Fissa: fusing item similarity models with self-attention networks for sequential recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems. pp. 130–139 (2020)
24. Liu, B., Tang, R., Chen, Y., Yu, J., Guo, H., Zhang, Y.: Feature generation by convolutional neural network for click-through rate prediction. In: The World Wide Web Conference. pp. 1119–1129 (2019)
25. Liu, H., Lu, J., Yang, H., Zhao, X., Xu, S., Peng, H., Zhang, Z., Niu, W., Zhu, X., Bao, Y., et al.: Category-specific cnn for visual-aware ctr prediction at jd. com. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2686–2696 (2020)
26. Ma, C., Ma, L., Zhang, Y., Sun, J., Liu, X., Coates, M.: Memory augmented graph neural networks for sequential recommendation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 5045–5052 (2020)
27. Ma, Y., Narayanaswamy, B., Lin, H., Ding, H.: Temporal-contextual recommendation in real-time. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2291–2299 (2020)
28. Medsker, L., Jain, L.C.: Recurrent neural networks: design and applications. CRC press (1999)
29. Merabet, F.Z., Benmerzoug, D.: Qos prediction for service selection and recommendation with a deep latent features autoencoder. Computer Science and Information Systems 19(2), 709–733 (2022)

30. Ouyang, W., Zhang, X., Li, L., Zou, H., Xing, X., Liu, Z., Du, Y.: Deep spatio-temporal neural networks for click-through rate prediction. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2078–2086 (2019)
31. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD cup and workshop. vol. 2007, pp. 5–8 (2007)
32. Pathak, A., Gupta, K., McAuley, J.: Generating and personalizing bundle recommendations on steam. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1073–1076 (2017)
33. Phuong, T.M., Thanh, T.C., Bach, N.X.: Combining user-based and session-based recommendations with recurrent neural networks. In: Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25. pp. 487–498. Springer (2018)
34. Qiu, R., Li, J., Huang, Z., Yin, H.: Rethinking the item order in session-based recommendation with graph neural networks. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 579–588 (2019)
35. Quadrana, M., Cremonesi, P., Jannach, D.: Sequence-aware recommender systems. ACM Computing Surveys (CSUR) 51(4), 1–36 (2018)
36. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: proceedings of the Eleventh ACM Conference on Recommender Systems. pp. 130–137 (2017)
37. Rendle, S.: Factorization machines. In: 2010 IEEE International conference on data mining. pp. 995–1000. IEEE (2010)
38. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 811–820 (2010)
39. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on World Wide Web. pp. 521–530 (2007)
40. Ruck, D.W., Rogers, S.K., Kabrisky, M.: Feature selection using a multilayer perceptron. Journal of Neural Network Computing 2(2), 40–48 (1990)
41. Ruocco, M., Skrede, O.S.L., Langseth, H.: Inter-session modeling for session-based recommendation. In: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems. pp. 24–31 (2017)
42. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295 (2001)
43. Seol, J.J., Ko, Y., Lee, S.g.: Exploiting session information in bert-based session-aware sequential recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2639–2644 (2022)
44. Shen, X., Yi, B., Zhang, Z., Shu, J., Liu, H.: Automatic recommendation technology for learning resources with convolutional neural network. In: 2016 international symposium on educational technology (ISET). pp. 30–34. IEEE (2016)
45. Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., Tang, J.: Autoint: Automatic feature interaction learning via self-attentive neural networks. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 1161–1170 (2019)
46. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 1441–1450 (2019)
47. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the eleventh ACM international conference on web search and data mining. pp. 565–573 (2018)

48. Wan, M., McAuley, J.: Item recommendation on monotonic behavior chains. In: Proceedings of the 12th ACM conference on recommender systems. pp. 86–94 (2018)
49. Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D.: A survey on session-based recommender systems. ACM Computing Surveys (CSUR) 54(7), 1–38 (2021)
50. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q.Z., Orgun, M.: Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830 (2019)
51. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 165–174 (2019)
52. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 346–353 (2019)
53. Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., Fang, J., Zhou, X.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI. vol. 19, pp. 3940–3946 (2019)
54. Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K.: Self-attention with functional time representation learning. Advances in neural information processing systems 32 (2019)
55. Ye, H., Li, X., Yao, Y., Tong, H.: Towards robust neural graph collaborative filtering via structure denoising and embedding perturbation. ACM Transactions on Information Systems 41(3), 1–28 (2023)
56. Ying, H., Zhuang, F., Zhang, F., Liu, Y., Xu, G., Xie, X., Xiong, H., Wu, J.: Sequential recommender system based on hierarchical attention network. In: IJCAI International Joint Conference on Artificial Intelligence (2018)
57. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 729–732 (2016)
58. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: Proceedings of the twelfth ACM international conference on web search and data mining. pp. 582–590 (2019)
59. Zhang, M., Wu, S., Gao, M., Jiang, X., Xu, K., Wang, L.: Personalized graph neural networks with attention mechanism for session-aware recommendation. IEEE Transactions on Knowledge and Data Engineering 34(8), 3946–3957 (2020)
60. Zhang, Y.: The application of e-commerce recommendation system in smart cities based on big data and cloud computing. Computer Science and Information Systems 18(4), 1359–1378 (2021)
61. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the tenth ACM international conference on web search and data mining. pp. 425–434 (2017)
62. Zhenzhen, X., Jiang, H., Kong, X., Kang, J., Wang, W., Xia, F.: Cross-domain item recommendation based on user similarity. Computer Science and Information Systems 13(2), 359–373 (2016)

Shangzhi Guo received the B.Sc. degree in software engineering from Hunan Normal University in 2005, the M.Sc. degree in software engineering from Hunan University in 2018. He is currently studying for a Ph.D. in mechanical engineering from Chongqing University as a senior engineer. His research interests are artificial intelligence and intelligent manufacturing, intelligent recommendation, system architecture, and big data application.

Xiaofeng Liao received the B.Sc. degree in computational mathematics from Sichuan University in 1986, the M.Sc. degree in computational mathematics from Sichuan University in 1992, Ph.D. in circuits and systems from University of Electronic Science and Technology of China in 1997, IEEE Fellow, AIAA Fellow. His research interests include computational intelligence and information security, artificial neural networks, dynamic theory, cryptography.

Fei Meng received the B.Sc. degree in physical technology from Huaqiao University in 2012, the M.Sc. degree in computer technology engineering from Chongqing University in 2018. She is currently studying for a Ph.D. in transportation from Chongqing University. She research interests include spatiotemporal trajectory data processing and big data application.

Qing Zhao received the B.Sc. degree in electrical engineering from Chongqing University in 2001, the M.Sc. degree in software engineering from Fudan University in 2009. He is currently studying for a Ph.D. in mechanical engineering from Chongqing University. His research interests include artificial intelligence and intelligent manufacturing, data security, network security, and big data applications.

Yuling Tang received the B.Sc. degree in computer science and technology from Tianjin Vocational and Technical Normal University in 2010. She research interests include medical big data, auxiliary medicine, and smart medicine.

Hui Li received the B.Sc. degree in management from Jiangxi Normal University in Nanchang, China in 2019, the M.Sc. degree in computer technology from Jiangxi Normal University in 2022. He is currently working on data integration and mining at the Jiangxi Provincial Institute of Land and Space Survey and Planning , nanchang, China. His research interests includes natural language processing and data mining.

Qinqin Zong received the B.Sc. degree in software engineering from Jiangxi Normal University in 2018, the M.Sc. degree in computer science and technology from Jiangxi Normal University in 2021. She is currently working at Jiangxi Biotechnology Vocational College. She research interests include natural language processing and recommendation systems.

Received: May 22, 2023; Accepted: August 20, 2023.