FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation

Jing Lin
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
linjing2018@email.szu.edu.cn

Weike Pan*
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
panweike@szu.edu.cn

Zhong Ming*
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

ABSTRACT

Sequential recommendation has been a hot research topic because of its practicability and high accuracy by capturing the sequential information. As deep learning (DL) based methods being widely adopted to model the local and dynamic preferences beneath users' behavior sequences, the modeling of users' global and static preferences tends to be underestimated that usually, only some simple and crude users' latent representations are introduced. Moreover, most existing methods hold an assumption that users' intention can be fully captured by considering the historical behaviors, while neglect the possible uncertainty of users' intention in reality, which may be influenced by the appearance of the candidate items to be recommended. In this paper, we thus focus on these two issues, i.e., the imperfect modeling of users' global preferences in most DL-based sequential recommendation methods and the uncertainty of users' intention brought by the candidate items, and propose a novel solution named fusing item similarity models with self-attention networks (FISSA) for sequential recommendation. Specifically, we treat the state-of-the-art self-attentive sequential recommendation (SASRec) model as the local representation learning module to capture the dynamic preferences beneath users' behavior sequences in our FISSA, and further propose a global representation learning module to improve the modeling of users' global preferences and a gating module that balances the local and global representations by taking the information of the candidate items into account. The global representation learning module can be seen as a location-based attention layer, which is effective to fit in well with the parallelization training process of the self-attention framework. The gating module calculates the weight by modeling the relationship among the candidate item, the recently interacted item and the global preference of each user using an MLP layer. Extensive empirical studies on five commonly used datasets show that our FISSA significantly outperforms eight state-of-the-art baselines in terms of two commonly used metrics.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Sequential Recommendation, Self-Attention, Item Similarity Models, Gating Networks

ACM Reference Format:
Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383313.3412247

1 INTRODUCTION

A recommendation system is an intelligent tool to alleviate the problem of information overload especially when users' intents are uncertain. Traditional recommendation systems deal with general recommendation only, in which the user-item interaction records can be placed in a 2-D rating matrix, so that predictions are achieved by filling the vacancies of this matrix. Different from general recommendation, sequential recommendation treats users' historical records as sequences of items rather than sets of items, in order to predict exactly the next item that they will interact with. Sequential recommendation is now widely studied, because it is more consistent with real-world situations and is expected to obtain more accurate result with more information considered.

Now that the global and static preference of users have been well explored in general recommendation, an intuitive way to develop a sequential recommendation method is to model the local and dynamic preference and combine it with the global one. This is exactly what the state-of-the-art factorized personalized Markov chains (FPMC) [32] does. Specifically, FPMC consists of two parts, i.e., the traditional matrix factorization (MF) model that factorizes the one-class feedback matrix, and a novel MF model that factorizes the transition matrix generated through personalized Markov chains. An improved model called Fossil[8] replaces the former component of FPMC with the factored item similarity model (FISM) [15], extends the latter component to a higher-order version by including multiple transition matrices and also introduces some personalized weighting factors to balance these global and local preferences.

Recently, more and more deep learning (DL) based methods are adopted to model the dynamic interest. One of the earliest works that apply recurrent neural networks (RNNs) for sequential recommendation is GRU4Rec [13], which progressively learns the user

[*] co-corresponding authors

RecSys '20, September 22–26, 2020, Virtual Event, Brazil
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412247

preference at each step. Caser [38] utilize convolutional neural networks (CNNs) to learn complex (i.e., point-level, union-level and skipping) short-term interest by sliding filters. In many other works, the attention mechanism has become a remarkable component to discover users' intention unshackled from the fixed order. SASRec [16] stacks multiple self-attention blocks to capture both the long- and short-term transitions within sequences efficiently. However, most of these DL-based methods do not pay enough attention to the static representation learning, let along the combination of the static and dynamic representations, e.g., in Caser, the learnable users' latent representations and the concatenation operation are simply adopted, which makes them still be challenged by models with well-designed global representations and balanced combination approach like Fossil.

Besides, almost all of the existing sequential recommendation methods rely on an idealistic assumption that users' intention can be fully captured by using their historical behaviors, i.e., the interaction between the final representation of a user's preference and a new item (or a candidate item to be recommended) is usually executed at the last step of the algorithms, before which the information of the candidate item is never used for the preference learning. Actually, users' intention can be uncertain especially when they are involved in a longstanding habitual behavior (e.g., purchase) sequence rather than a short-lived activity (e.g., listen to music) session. A proper way to tell whether a new item can attract a user is to consider how it can arouse different parts of the user's interest (i.e., the short-term one and the long-term one).

Based on the above analysis, in this paper, we propose a novel solution named fusing item similarity models with self-attention networks (or FISSA in short) for sequential recommendation. Our FISSA not only joins the effective global representation learning to the well-established method, i.e., self-attentive sequential recommendation (SASRec) [16], but also balances a user's short-term and long-term interest for each candidate item. Specifically, our model contains three main components, i.e., a local representation learning module, a global representation learning module, and a gating module to balance these two kinds of representations. For local representation learning, we follow SASRec because it has achieved excellent performance, and enhancing the dynamic interest modeling is not our focus in this paper. For global representation learning, we apply a location-based attention layer to achieve an attentive version of FISM [15], in which a query vector shared by all the sequences are introduced, so as to distinguish the importance of different items for generating the global representation of the sequences. Inspired by neural attentive item similarity (NAIS) [9] that weighs items by considering their relations to the candidate item, we design a gating network based on a multilayer perceptron (MLP) that decides the contribution ratio of the local and global representations by considering the relationship among the candidate item, the recently interacted item and the global preference of a target user.

We summarize our main contributions as follows:

- We propose a novel solution named FISSA to deal with two issues, i.e., the imperfect modeling of users' global preferences in most DL-based sequential recommendation methods and the uncertainty of users' intention which may be influenced by the candidate items.
- We design a global representation learning module to effectively capture users' global preferences in our FISSA, which can be seen as a location-based attention layer that fits in well with the parallelization training process of the self-attention framework.
- We design an MLP-based gating module in our FISSA, which balances the local and global representations by taking the information of the candidate items into account, so as to deal with the uncertainty of the users' intention at the same time.
- We conduct extensive empirical studies on five commonly used datasets and show that our FISSA significantly outperforms eight state-of-the-art baselines. In particular, our FISSA surpasses SASRec by 10.11% and 10.05% on average in terms of Rec@10 and NDCG@10, respectively. We also conduct ablation studies and discuss some options for the details of the global and gating modules, etc.

2 REALTED WORK

In this section, we review the state-of-the-art methods for general recommendation and sequential recommendation, respectively, and point out the relationship and differences between our FISSA and those works, as well as how our FISSA significantly advances the closely related works for the studied problem.

2.1 General Recommendation

Collaborative filtering (CF) methods often treat the users' behavior history as a set of user-item interaction pairs. There are three main branches of CF methods, i.e., neighborhood-based methods [1, 34], matrix factorization (MF) based methods [28, 31] and hybrid methods [15, 17]. MF-based methods have been popular because of their high efficiency and accuracy. In early works of MF-based methods [28, 31], user-specific and item-specific latent representation vectors are directly learned by executing singular value decomposition (SVD) on the rating matrix, and a predicted rating is obtained via the inner product of the two corresponding vectors. Later, there are suggestions [15, 17] (e.g., FISM [15]) to obtain the representation of a user by summarizing the representation of his/her interacted items. In this way, the predicted rating can be regarded as the factored similarity between the user's historical items and the candidate item, which provides the MF-based model with good interpretability as neighborhood-based models. Also, the composite user representation is more informative to tackle the limited number of users' records. Lately, deep learning (DL) based methods [2, 10, 21, 42] are adopted to enhance the above methods. For example, neural collaborative filtering (NCF) [10] uses multilayer perceptron (MLP) to learn the user-specific and item-specific latent representation vectors and can be easily combined with the traditional MF models. In another model called attentive collaborative filtering (ACF) [2], the attention mechanism is applied to weigh different historical items and build a more comprehensive model. Neural attentive item similarity (NAIS) [9] also applies the attention mechanism but focuses on distinguishing more important items for the candidate item rather than for the user. There are also other DL-based methods that use an autoencoder (AE) [21, 22, 42] or restricted Boltzmann machines (RBMs) [33] for general recommendation. In this paper, we achieve an attentive form of FISM [15] to obtain the global representation of a user's behavior sequence, and design an item similarity gating for balancing the local and global representations by modeling the relationship among the candidate item, the recently interacted item and the global preference of the user.

2.2 Sequential Recommendation

The earliest works on sequential recommendation use Markov chains (MCs) to model the first-order transition between items [49], or a Markov decision processes (MDP) [35] to handle long-term effects. Later, factorized personalized MCs [32] are proposed and extended to a higher-order version [8], which are inspired by and combined with general MF-based methods [15, 31]. To maintain the triangle inequality for the sparse transition data, metric embedding [5] and translation-based [7, 19] methods are proposed. While recently most researchers follow the crowd to adopt DL-based methods to capture the nonlinearity and dynamic features for sequential recommendation. RNN-based models [4, 12, 13, 20, 30, 37, 47] are nearly the first to be adopted because of their natural instincts to model sequences step-by-step. To avoid the vanishing gradient problem brought by RNNs, other DL-based methods that use CNNs [38, 48] are also carried out, with additional characteristics such as multiple and flexible filter sizes to refine the short-term features. Based on RNNs and CNNs, the applications of some emerging network models come into vogue. For example, memory networks [3, 14], graph neural networks (GNN) [26, 29, 41, 43] that cooperate with the attention mechanism are used to extract short-term features with more consistency or adjacency consideration. Note that the attention mechanism is also proved to be effective on its own with proper hierarchical structure [16, 23, 46]. In this paper, we base our local representation learning module on the self-attentive sequential recommendation (SASRec) model [16], which is found to be an outstanding sequential recommendation model with satisfactory conciseness and efficiency. Note that different from other works that improve SASRec by introducing graph neural networks [43] or bidirectional structure [36], which still focus on the local and dynamic preference modeling, our proposed FISSA aims at combining SASRec with an effective global and static preference learning model in a balanced way. Another improved work of SASRec (i.e., consistency-aware recommendation (CAR) [11]) that is similar to our FISSA is further discussed in Section 4.

For dealing with the uncertainty of users' intention in sequences, existing works mainly focus on distinguishing the importance of the items in sequences. For example, in a recent work, a model named streaming session-based recommendation (SSR) [6] focuses on streaming session data and introduces an MF-based attention into the RNN-based session encoder, so that a user's intention in the current session is related to the long-term preferences from the historical sessions. Different from this work, we focus on some longstanding habitual behaviors (e.g., review, check-in, purchase, etc.), and thus only consider one sequence for each user. Moreover, in addition to learning the importance of each item in the sequence through a self-attentive model, we separately model the short-term and long-term preferences of a user and then balance them according to different candidate items, which means that our FISSA captures the changing intention influenced by the appearance of the candidate items and is more well rounded.

3 PROPOSED METHOD

In this section, we propose our FISSA, i.e., fusing item similarity models with self-attention networks for sequential recommendation. Without loss of generality, we have a recommendation system with implicit feedback given by a set of users $\mathcal{U}$ to a set of items $\mathcal{I}$. For sequential recommendation, we denote the records of each user $u \in \mathcal{U}$ as an item sequence (ordered by the interaction time) as $\mathcal{S}^u = \{s_1^u, s_2^u, \ldots, s_{|\mathcal{S}^u|}^u\}$, $s_\cdot^u \in \mathcal{I}$. Our goal is to provide a recommendation list for each user u, in which we expect the real next interacted item $s_{|\mathcal{S}^u|+1}^u \in \mathcal{I} \backslash \mathcal{S}^u$ to appear and be ranked as high as possible.

We illustrate our FISSA in Figure 1, which contains three main components, including a local representation learning module, a global representation learning module, and a gating module to balance these two kinds of representations. In this paper, we use capital letters in bold to denote matrices and their lowercase form to denote the corresponding row vectors.

3.1 Local Representation Learning

First of all, we fix the input sequence of each user u by extracting his/her latest L behaviors, which is abbreviated as $\mathcal{S}^u = \{s_1, s_2, \ldots, s_L\}$ (usually a relatively large value of L, e.g., $L = 50$ for our studied datasets, is chosen to reserve the whole sequences of most users, and padding items are appended at the beginning of the sequences when needed). Let $\boldsymbol{M} \in \mathbb{R}^{|\mathcal{I}|\times d}$ denote the learnable item embedding matrix with d as the latent dimensionality. We can then represent the input sequence as an embedding matrix $\boldsymbol{E} = [\boldsymbol{m}_{s_1}; \boldsymbol{m}_{s_2}; \ldots; \boldsymbol{m}_{s_L}] \in \mathbb{R}^{L\times d}$.

Following [16], we use a hierarchical self-attention network to capture both the short-term and long-term item transitions in the sequence. In order to capture the influence of the position, we add a learnable position embedding matrix $\boldsymbol{P} = [p_1; p_2; \ldots; p_L] \in \mathbb{R}^{L\times d}$ to the input embedding matrix $\boldsymbol{E} \in \mathbb{R}^{L\times d}$, and obtain an input matrix $\boldsymbol{X}^{(0)} = [x_1; x_2; \ldots; x_L] \in \mathbb{R}^{L\times d}$ for the self-attention network:

$$\boldsymbol{x}_\ell^{(0)} = \boldsymbol{m}_{s_\ell} + \boldsymbol{p}_\ell, \ell \in \{1, 2, \ldots, L\}. \tag{1}$$

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

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

where the self-attention block $SAB^{(b)}(\cdot)$ is first introduced in [39]. 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:

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

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

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

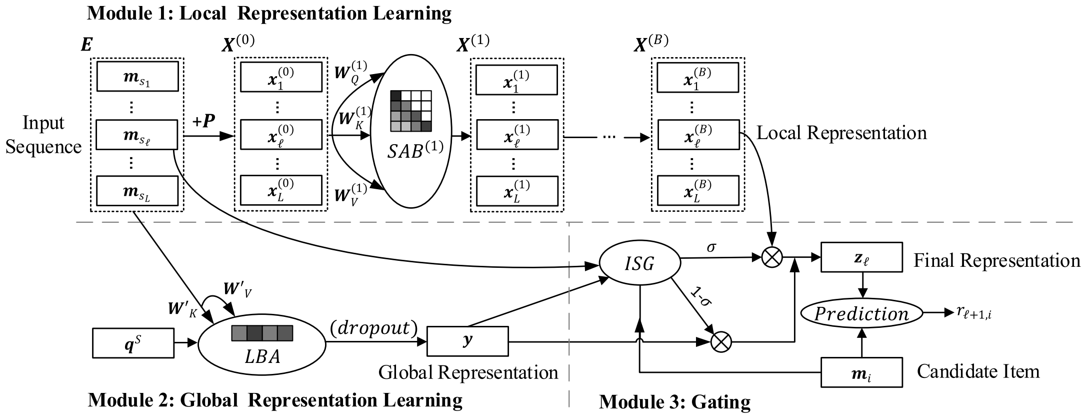


Figure 1: The network architecture of our proposed FISSA. At the beginning, the input sequence (in the upper left corner) is represented as an embedding matrix E, in which each item embedding $m_\cdot$ is from the item embedding matrix $M \in \mathbb{R}^{|\mathcal{I}|\times d}$ that works as a dictionary. The local representation learning module (in the top half) consists of a series of stacked self-attention blocks $SAB(\cdot)$ (see Eqs.(2∼5)). The global representation learning module (in the bottom left corner) is actually a location-based attention layer $LBA(\cdot)$ (see Eq.(7)). In the gating module (in the bottom right corner), the item similarity gating function $ISG(\cdot,\cdot,\cdot)$ (see Eq.(9)) outputs the balanced weights of the local and global representations by taking the representations of the candidate item m_i, the recently interacted item m_{s_ℓ} and the user's global preference y as inputs.

where $X \in \mathbb{R}^{L\times d}$ is the position-aware input matrix, $Q = XW_Q$, $K = XW_K$ and $V = XW_V$ with $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are the projected query, key and value matrices, respectively, to improve the flexibility. Note that $W_1, W_2 \in \mathbb{R}^{d\times d}$ and $b_1, b_2 \in \mathbb{R}^{1\times d}$ are weights and biases for the two layers of convolution, $\mathbf{1}$ is a unit row vector of size $1 \times L$ and Δ is the causality mask, i.e., a unit lower triangular matrix of size $L \times L$, to preserve the transitions from previous steps only. The normalization and dropout layers we use in this module are the same with that in [39].

In this module, we take the output vector $x_\ell^{(B)} \in \mathbb{R}^{1\times d}$ from the top self-attention block as the local representation, which stands for the dynamic preference at the ℓth step in the user behavior sequence. It is shown in [16] that the hierarchical structure is important for the local representation. Specifically, the bottom self-attention block (i.e., $SAB^{(1)}(\cdot)$) tends to capture the long-term dependencies, while higher blocks may pay attention to more recent ones.

3.2 Global Representation Learning

Though applying the attention mechanism to avoid rigorous ordering of the previous items, the local representation still ignores the variable ordering of the current item and its subsequent items. A simple way to deal with this issue is to generate a global and non-causal representation of each user's behavior sequence, so that more available information from the future can be utilized for the prediction at each step in the sequence during training.

By referring to factored item similarity model (FISM) [15] for general recommendation, we propose an attentive version of FISM for global representation learning, which is well adapted to the parallelization training process of the self-attention framework (i.e., including all the steps of a sequence in one training sample). Note that the global representation learning module is independent of the local one.

In FISM [15], the preference of user u on the interacted item $s_{\ell+1}^u$ at the $(\ell+1)$th step is generated as the uniform aggregation of the representation of the other interacted items:

$$\tilde{y}_{\ell+1}^u = \frac{1}{\sqrt{|\mathcal{S}^u \backslash \{s_{\ell+1}^u\}|}} \sum_{i' \in \mathcal{S}^u \backslash \{s_{\ell+1}^u\}} m_{i'}. \tag{6}$$

In this way, the predicted rating $r_{\ell+1,s_{\ell+1}^u}^u = \tilde{y}_{\ell+1}^u m_{s_{\ell+1}^u}^T$ can be regarded as a factored similarity between the historical items of user u and the candidate item $s_{\ell+1}^u$. Another understanding of FISM is that with the benefit of utilizing the shared item representation, sequences with similar items tend to have similar representations. We believe that this effect can be enhanced if more representative items are noticed. So instead of an aggregation with average weighting, we introduce a learnable query vector $q^S \in \mathbb{R}^{1\times d}$ shared by all sequences to figure out the most representative items in the sequences. Omitting the superscript u, the global representation of the sequence can then be formalized as follows:

$$y = LBA(E) = softmax(q^S(EW'_K)^T)EW'_V, \tag{7}$$

where $E \in \mathbb{R}^{L\times d}$ is the initial input matrix as mentioned in the first paragraph of Section 3.1, $W'_K, W'_V \in \mathbb{R}^{d\times d}$ are projection matrices to be learned, similar to W_Q, W_K, W_V in Eq.(4). Such an attention layer is also known as a location-based attention layer $LBA(\cdot)$ described in [24], in which neither personalized nor contextual information is embedded in the query vector. Note that besides the query condition, the main differences of the attention layer in Eq.(7) and Eq.(4) also include that the position information P and the causality constraint Δ are abandoned here.

It is worth mentioning that in our case, the global representation of the sequence is the same to all steps, which means that the corresponding parameters in Eq.(7) are updated only once in a training epoch even for predictions in many (e.g., L) steps. So a dropout layer is very important during training to generalize the global representation to all steps, i.e., we have the global representation matrix $\boldsymbol{Y} \in \mathbb{R}^{L \times d}$ as follows:

$$\boldsymbol{y}_\ell = Dropout(\boldsymbol{y}), \ell \in \{1, 2, \ldots, L\}. \tag{8}$$

In a related work to ours that also improves SASRec by introducing the global representation [11], the authors still keep the position information and the causality constraint, which is found to be a sub-optimal choice in our empirical studies in Session 4.

3.3 Item Similarity Gating

To combine the local representation and the global representation, we may naturally think of concatenation or summation. Our early attempts have shown that summation is always better than concatenation. In [11], the authors 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. However, these approaches of combination are still based on the historical information only, which may be idealistic as discussed in Section 1.

To deal with the the issue of the uncertainty of users' intention in sequential recommendation and inspired by neural attentive item similarity (NAIS) [9], we propose an item similarity gating module, which calculates the weight of the local representation and global representation by modeling the item similarity between the candidate item $i \in \mathcal{I}$ and the recently interacted item s_ℓ, as well as the item similarity between the candidate item i and the aggregation of the historical items $i' \in \mathcal{S}^u$. To simplify the model, we 1) determine the output value of the gating function g as the weight of the local representation and restrict it to $0 < g < 1$, so that the weight of global representation is automatically obtained as $1 - g$; and 2) feed the three kinds of 'items' (i.e., the candidate item i, the recently interacted item s_ℓ and the aggregation of the historical items $i' \in \mathcal{S}^u$) into the gating function by one step, which means that the two pairs of item similarities are integrated into the relationship among the candidate item, the recently interacted item and the global preference of user u.

Specifically, the representation of the candidate item $i \in \mathcal{I}$ and the interacted item at the most recent step s_ℓ are taken from the primitive item embedding matrix $\boldsymbol{M}$, i.e., $\boldsymbol{m}_i$ and $\boldsymbol{m}_{s_\ell}$, respectively. The global preference is represented as the aggregated representation of the historical items $i' \in \mathcal{S}^u$, which is exactly the learned global representation $\boldsymbol{y}$. The individual-level gating is then written as an MLP as follows:

$$g = \sigma(ISG(\boldsymbol{m}_{s_\ell}, \boldsymbol{y}, \boldsymbol{m}_i)) = \sigma([\boldsymbol{m}_{s_\ell}, \boldsymbol{y}, \boldsymbol{m}_i]\boldsymbol{W}_G + b_G), \tag{9}$$

where $ISG(\cdot,\cdot,\cdot)$ is the item similarity gating function, $[\cdot,\cdot,\cdot]$ denotes the ternary concatenation operation, $\boldsymbol{W}_G \in \mathbb{R}^{3d \times 1}$ and $b_G \in \mathbb{R}$ are the weights and bias to be learned, respectively. We use the sigmoid function $\sigma(\xi) = 1/(1 + e^{-\xi})$ as the activation function, so that g is restricted to $(0, 1)$.

For other options, similar to [9], we also try introducing the two element-wise products (representing the two pairs of item similarities) as "$[\boldsymbol{m}_{s_\ell} \otimes \boldsymbol{m}_i, \boldsymbol{y} \otimes \boldsymbol{m}_i]$" for the input of the MLP, which makes no difference to the prediction performance and may cause some information loss in theory [9]. We can also design a feature-level gating to output a vector rather than a scalar, which is included in our empirical studies in Section 4.2.4.

The final representation of the sequence at the ℓth step is obtained by the weighed sum of the corresponding local representation $\boldsymbol{x}_\ell^{(B)}$ and global representation $\boldsymbol{y}$ as follows:

$$\boldsymbol{z}_\ell = \boldsymbol{x}_\ell^{(B)} \otimes g + \boldsymbol{y} \otimes (1 - g), \tag{10}$$

where $\otimes$ denotes the element-wise operation with the broadcasting mechanism in TensorFlow. Note that the weight of the local representation in our FISSA can be extended from $(0, 1)$ to $[0, 1]$ to include both SASRec ($\boldsymbol{z}_\ell^L = \boldsymbol{x}_\ell^{(B)}$ when $g = 1$, i.e., our local representation learning module in Section 3.1) and the attentive version of FISM ($\boldsymbol{z}_\ell^G = \boldsymbol{y}$ when $g = 0$, i.e., our global representation learning module in Section 3.2) as special cases.

Finally, we predict the preference of item i being the $(\ell + 1)$th item in the sequence as follows:

$$r_{\ell+1, i} = \boldsymbol{z}_\ell (\boldsymbol{m}_i)^T. \tag{11}$$

We train our FISSA by minimizing the binary cross-entropy loss with the Adam optimizer. The loss function is as follows:

$$\mathcal{L} = -\sum_{u \in \mathcal{U}} \sum_{\ell=1}^{L-1} \delta(s_{\ell+1}^u)[\log(\sigma(r_{\ell+1, s_{\ell+1}^u})) + \log(1 - \sigma(r_{\ell+1, j}^u))], \tag{12}$$

where $j \in \mathcal{I} \backslash \mathcal{S}^u$ is a negative item randomly sampled for each prediction. The indicator function $\delta(s_{\ell+1}^u) = 1$ only if $s_{\ell+1}^u$ is not a padding item, otherwise 0.

4 EXPERIMENTS

In this Section, we present experimental settings and results to answer the following research questions: **RQ1**) Does our FISSA achieve the state-of-the-art performance? **RQ2**) What is the impact of different components in our FISSA? **RQ3**) How does the key parameters such as the dimensionality d and the number of blocks B affect the performance of our FISSA? **RQ4**) What is the impact of some options for the design of the global representation module (e.g., the consideration of causality) and the gating module (e.g., the input and output of the MLP layer) in our FISSA?

4.1 Settings

4.1.1 Datasets. We conduct experiments on five public datasets from four real-world scenarios, i.e., Amazon[1], Steam[2], Foursquare[3] and Tmall[4]. Amazon and Steam are review datasets collected by [16, 27] from the eponymous e-commerce and video game platform, respectively. Note that we follow [16] and choose two categorized datasets from Amazon, i.e., 'Beauty' and 'Games'. Foursquare contains check-in records collected by [18] from the eponymous location-based social application. Tmall is another e-commerce dataset with multiple behaviors (including clicks, purchases, etc.) recorded and is published for the IJCAI Competition 2015. For sequential recommendation, we preprocess these datasets as follows:

[1] http://jmcauley.ucsd.edu/data/amazon/

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

[3] https://archive.org/details/201309_foursquare_dataset_umn

[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42

Dataset	# Users	# Items	# Interactions	Avg. Length	Density
Beauty	40,226	54,542	353,962	8.80	0.02%
Games	29,341	23,464	280,945	9.58	0.04%
Steam	281,428	13,044	3,488,899	12.40	0.10%
Foursquare	22,748	11,146	145,106	6.38	0.06%
Tmall	201,139	97,636	1,936,790	9.63	0.01%

Table 1: Statistics of the processed datasets.

1) we treat the presence of review, check-in and purchase behaviors as positive feedback and order them by the timestamps; 2) we discard later duplicated user-item pairs in order to predict new items; 3) we successively discard items and users with fewer than 5 records to maintain sequentiality; and 4) we adopt the leave-one-out evaluation by splitting each dataset into three parts, i.e., the last interaction of each user for test, the penultimate one for validation and the rest for training. Note that cold-start items in the test and validation data are also removed. The statistics of the processed datasets are shown in Table 1. The source codes for preprocessing the datasets are released together with the implementation code of our FISSA (see Section 4.1.4).

4.1.2 Evaluation Metrics. We evaluate the recommendation performance via two common metrics, i.e., recall (Rec@10, equivalent to hit ratio because there is exactly one preferred item by each user in our case) and normalized discounted cumulative gain (NDCG@10). Rec@10 refers to the ratio of the real next items presenting in the top-10 recommendation lists, while NDCG@10 cares more about the exact ranking positions of the target items in these lists. To reduce computation, we follow [10, 16] to prearrange a candidate list with 100 randomly sampled un-interacted items for each user.

4.1.3 Baselines. We adopt eight competitive baselines, including four MF-based methods and four DL-based methods as follows:

- BPRMF [31]. A general recommendation model that simply factorizes the user-item interaction matrix via a pairwise loss.
- FISM [15]. Another general model that indirectly obtains the users' representation by factorizing an item-to-item similarity matrix.
- FPMC [32]. A pioneer method for sequential recommendation that models first-order Markov chains (MCs) in a factorization way and combines it with BPRMF [31].
- Fossil [8]. An improved model of FPMC by utilizing FISM for global preference learning, extending factored MCs to higher orders, and introducing some personalized weighting factors to balance these two components.
- GRU4Rec+ [12]. An updated version of the session-based RNN model GRU4Rec [13] by adopting a listwise loss function (e.g., BPR-max) and an additional sampling strategy. Note that GRU4Rec is known as one of the earliest works to introduce DL-based methods (i.e., RNN) for sequential recommendation.
- Caser [38]. A CNN-based model which applies horizontal and vertical convolutional filters to capture the point-level, union-level and skipping patterns of the short-term preferences in sequences.
- SASRec [16]. A hierarchical self-attention network for sequential recommendation, which also works as the local representation learning module in our FISSA.
- CAR [11]. A similar model to ours, which also improves SASRec by introducing the global preferences of users and a consistency-aware gating. Note that unlike the others, this model aims at addressing the user-generated item list continuation problem defined in [11].

4.1.4 Implementation Details. We implement the MF-based models with the codes provided by [8] for the research of Fossil[5], and run the DL-based methods GRU4Rec+[6] and Caser[7] with the codes released by the authors of the original papers. Our code of FISSA[8] is an adaption from the published code of SASRec[9], in which the attention module is modified and a new MLP for our gating module is added. We also adapt our code for CAR. So we run the experiments of SASRec and CAR via our adapted codes rather than the original ones. For fair comparison, we select the item embedding dimensionality d in all models as 50 from a common range $\{10, 20, 30, 40, 50\}$ because we observe that the referenced baselines generally perform better with a larger value of d on such sparse datasets [16, 38]. Other key parameters such as the MC orders ($\in \{1, 2, \ldots, 9\}$ for Fossil and Caser), negative sampling numbers (2048 for GRU4Rec+), filter sizes (4 and 16 for the vertical and horizontal filters, respectively in Caser) and so on are all tuned on the validation data according to the suggestions in the corresponding papers. For our FISSA, following [16], we set the sequence length L to 50, the batch size to 128, the learning rate to 0.001 and the dropout rate to 0.5, and use single-head self-attention layers. The number of blocks B is important for the performance of SASRec, CAR and our FISSA, which is searched from $\{1, 2, 3\}$.

4.2 Results

4.2.1 Performance Comparison (RQ1). The recommendation performance of our FISSA and eight baselines on five datasets is shown in Table 2. The best result in each row is marked in bold, and the second best one is marked with an underline.

We can see that our FISSA achieves the best performance on all of the five datasets compared with all the baselines, which clearly demonstrates the superiority of our proposed model. On average of the five datasets, our FISSA improves SASRec by 10.11% in terms of Rec@10 and 10.05% in terms of NDCG@10. The second best performance is obtained by SASRec or CAR, which is consistent with the observations in previous studies [11, 16]. These results also show the advantage of the self-attention network for dynamic preference modeling. Moreover, we find that CAR does not improve much over SASRec ($\leq 1.89\%$) and is even defeated on Games and Tmall. Some possible reasons are as follows: 1) the global preference representation learned in CAR still maintains the causality constraint, which makes it redundant to the local one; and 2) the consistency-aware gating is designed for the user-generated item lists rather than for

[5] https://cseweb.ucsd.edu/~jmcauley/
[6] https://github.com/hidasib/GRU4Rec
[7] https://github.com/graytowne/caser_pytorch
[8] http://csse.szu.edu.cn/staff/panwk/publications/FISSA/
[9] https://github.com/kang205/SASRec

Dataset	Metric	MF-based				DL-based					FISSA vs.
		BPRMF	FISM	FPMC	Fossil	GRU4Rec+	Caser	SASRec	CAR	FISSA	SASRec
Beauty	Rec@10	0.2498	0.3533	0.2397	0.3570	0.2729	0.2809	0.3609	$\underline{0.3660}$	**0.4164**	15.38%
	NDCG@10	0.1148	0.1942	0.1093	0.2108	0.1683	0.1610	0.2173	$\underline{0.2214}$	**0.2484**	14.35%
Games	Rec@10	0.3454	0.4791	0.3665	0.4800	0.4825	0.4810	$\underline{0.6009}$	0.5947	**0.6743**	12.22%
	NDCG@10	0.1981	0.2631	0.2115	0.2708	0.2906	0.2857	$\underline{0.3685}$	0.3644	**0.4134**	12.19%
Steam	Rec@10	0.1023	0.3183	0.1735	0.2926	0.3177	0.2686	0.3886	$\underline{0.3927}$	**0.4294**	11.79%
	NDCG@10	0.0468	0.1703	0.0849	0.1546	0.1707	0.1342	0.2144	$\underline{0.2164}$	**0.2420**	13.91%
Foursquare	Rec@10	0.2659	0.3977	0.3641	0.4223	0.4324	0.4043	0.4808	$\underline{0.4868}$	**0.5106**	6.20%
	NDCG@10	0.1287	0.2025	0.1873	0.2303	0.2375	0.2108	0.2611	$\underline{0.2629}$	**0.2794**	7.04%
Tmall	Rec@10	0.1744	0.2149	0.1739	0.2303	0.3526	0.2813	$\underline{0.4204}$	0.4184	**0.4412**	4.94%
	NDCG@10	0.0825	0.1043	0.0821	0.1142	0.2072	0.1531	$\underline{0.2385}$	0.2380	**0.2451**	2.78%

Table 2: Recommendation performance of our FISSA and eight baselines on five datasets.

Dataset Architecture	Beauty	Games	Steam	Foursquare	Tmall
L_1	0.3268	0.5646	0.3699	0.4650	0.3874
L_3	0.3609	0.6009	0.3886	0.4808	$\underline{0.4204}$
G	0.3407	0.5447	0.2981	0.4830	0.3423
L + G	$\underline{0.3851}$	$\underline{0.6301}$	$\underline{0.4010}$	**0.5267**	0.4197
L + G + C	0.3727	0.6152	0.3973	$\underline{0.5199}$	0.4198
L + G + I	**0.4046**	**0.6712**	**0.4305**	0.4972	**0.4237**

Table 3: Recommendation performance (Rec@10) in ablation studies with different architectures on five datasets.

the interaction sequences, which means that it is more suitable for the prediction of longer sequences with different consistencies.

For the four MF-based methods, we observe that: 1) FISM beats BPRMF on all the five sparse datasets, which indicates the effectiveness of the item similarity model for generating users' global representations on these sparse datasets; and 2) FPMC surpasses BPRMF on three of the five datasets and Fossil performs the best among these four MF-based methods except on Steam, which shows the rationality to consider the high-order sequential information, as well as the importance to balance the dynamic short-term preferences with the static long-term preferences.

Moreover, we notice that though having achieved very promising performance, GRU4Rec+ and Caser are still challenged by Fossil and FISM in some cases, i.e., GRU4Rec+ and Caser on Beauty, and Caser on Steam and Foursquare. This actually justifies our motivation to generate better global representation for DL-based sequential recommendation models.

4.2.2 *Ablation Study (RQ2)* .

In order to figure out the contribution of different components to the performance of our FISSA, we conduct an ablation study as shown in Table 3. Note that we only present the results on Rec@10 because the variation tendency of the NDCG@10 is similar to that of Rec@10. We compare the separate effect of the local representation learning module (i.e., SASRec, $\boldsymbol{z}_\ell^L = \boldsymbol{x}_\ell^{(B)}$, denoted as 'L_1' for $B = 1$ and 'L_3' for $B = 3$) and the global representation learning module (i.e., $\boldsymbol{z}_\ell^G = \boldsymbol{y}$, denoted as 'G'). We also examine the joint effect ($B = 1$) with different approaches of combination, i.e., normal summation ('L+G', $\boldsymbol{z}_\ell^{L+G} = \boldsymbol{x}_\ell^{(B)} + \boldsymbol{y}$), weighted summation with consistency-aware gating as in CAR [11] ('L+G+C') and weighted summation with our proposed item similarity gating ('L+G+I', i.e., our FISSA).

We have the following observations.

- **G vs. L.** SASRec with three blocks (i.e., 'L_3') wins on most of these datasets except on Foursquare, while SASRec with only one block (i.e., 'L_1') performs worse on two datasets (i.e., Beauty and Foursquare), which demonstrates the importance of the hierarchical structure and the competitive effectiveness of our global representation learning module.
- **'L + G' vs. L or G.** The straightforward hybrid model always significantly outperforms the separate ones (except on Tmall), which shows the complementary effect between the local and global representations in our FISSA and inspires us to look for more suitable approaches of combination.
- **'L + G + I' or 'L + G + C' vs. 'L + G'.** The consistency-aware gating 'C' does not work well in our cases, e.g., it pulls down the results from normal summation 'L + G' on almost all the datasets. In contrast, adopting our item similarity gating 'I' improves the recommendation accuracy on four of the five datasets (the exception on Foursquare may probably be due to overfitting, as latter discussed in Section 4.2.3). These results show that our gating network that concerns about the candidate item is more effective in balancing the local and global representations for sequential recommendation.

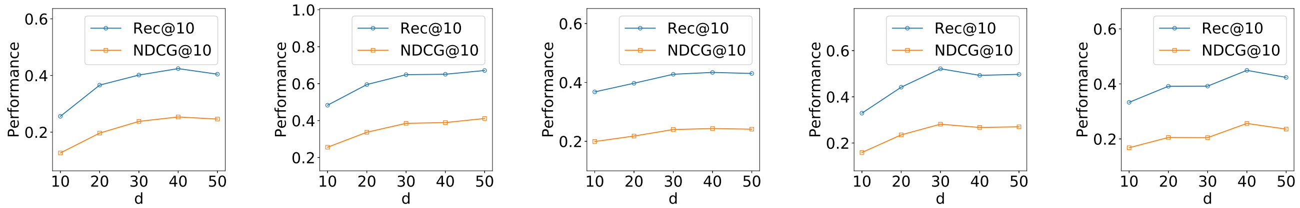
Note that some researchers successfully introduce the global preference into their local preference learning models by concatenating (e.g., [38]) or summation (e.g., [25, 26]) the learnable users' latent representations. However, as mentioned in both [16] and [11], adopting these approaches are not beneficial to the self-attention models, so we do not include them in our experiments.

4.2.3 *Quantitative Study (RQ3)* .

We study the effect of two hyperparameters, i.e., the dimensionality d that affects the representational capacity of the model and the number of block B for the hierarchical local representation learning module. We report the results in Figure 2 and Figure 3.

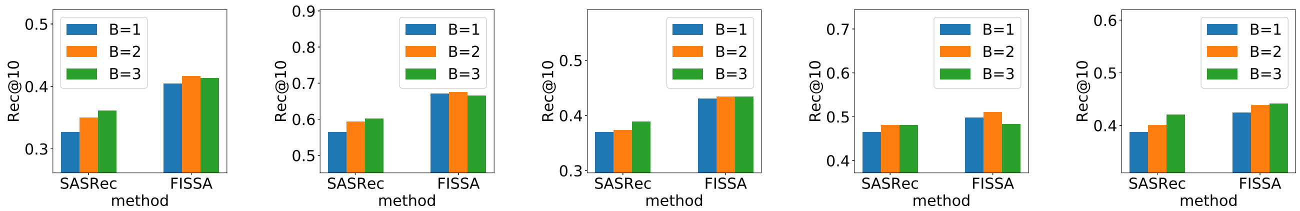
From Figure 2, we can see that our FISSA achieves better results as the dimensionality d gets bigger on Games and Steam, but is more easier to become overfitting on Beauty, Foursquare and Tmall, for which $d = 40$, $d = 30$ and $d = 40$ perform the best, respectively.

From Figure 3, we can see that unlike SASRec, setting the number of blocks $B = 2$ is enough for our FISSA to achieve the best results in most cases (except on Tmall), and adopting more blocks may backfire. This is because that though the hierarchical structure is still useful, the global representation learned in our FISSA is



(a) Beauty (b) Games (c) Steam (d) Foursquare (e) Tmall

Figure 2: Recommendation performance of SASRec and our FISSA with different dimensionalities d on five datasets ($B = 1$).



(a) Beauty (b) Games (c) Steam (d) Foursquare (e) Tmall

Figure 3: Recommendation performance (Rec@10) of SASRec and our FISSA with different numbers of blocks B on five datasets ($d = 50$).

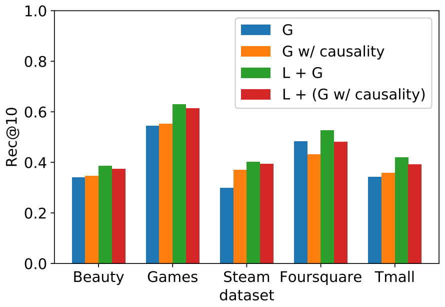


Figure 4: Recommendation performance of having and not having the causality constraint for the global representation leaning ($B = 1$, $d = 50$). Note that 'G' and 'L+G' denote the global representation learning module ($z_\ell^G = \boldsymbol{y}$) and the combination of the local and global representation modules ($z_\ell^{L+G} = \boldsymbol{x}_\ell^{(B)} + \boldsymbol{y}$), respectively, and the other two architectures are achieved by replacing the proposed global representation module 'G' with its causality-constrained version as 'G w/ causality'.

actually a novel substitute for the long-term transitions learned in the bottom block of SASRec.

4.2.4 Exploratory Study (RQ4). In the following, we present some results of different designs for the global representation module and the gating module.

The non-causality of global representation. As stated in Section 3.2, the global representation is learned with and shared by all steps in a sequence, which makes future information available during training. We replace it by masking the future steps and learn a relatively global (to the known history) representation $\boldsymbol{y'}_\ell = LBA(\boldsymbol{E}_{1:\ell})$ for each step ℓ (denoted as 'G w/ causality'). As shown in Figure 4, though a relatively global representation that only considers the historical interactions may be more suitable than a time unaware one (i.e., 'G w/ causality' is better than 'G' on Beauty, Games, Steam and Tmall), when joined with a well-established local representation learning model (also with causality), the causality consideration for global representation learning becomes redundant, i.e., 'L+G' is better than 'L + (G w/ causality)' on all the five datasets. This also demonstrates the advantage of introducing the future information for global representation learning in our FISSA.

The input and output of gating . In Figure 5, we make some changes to the MLP layer for the item similarity gating (see Eq.(9)) by 1) removing the global preference (i.e., $\boldsymbol{y}$) or the recently interacted item (i.e., $\boldsymbol{m}_{s_\ell}$) from the inputs of the MLP layer; and 2) switching the aforementioned individual-level gating to a feature-level one by setting $\boldsymbol{W}_G \in \mathbb{R}^{3d \times d}$ and $\boldsymbol{b}_G \in \mathbb{R}^{1 \times d}$ to obtain an output vector $\boldsymbol{g} \in \mathbb{R}^{1 \times d}$, which provides different weights for different dimensions. From Figure 5, we can see that introducing a single type of historical representation (i.e., the global preference $\boldsymbol{y}$ or the recent interaction $\boldsymbol{m}_{s_\ell}$) into the gating function is usually enough to achieve the excellent results. Specifically, on Beauty and Steam, introducing $\boldsymbol{y}$ only is more effective, while on the other three datasets, introducing $\boldsymbol{m}_{s_\ell}$ only is helpful. Considering that sometimes (e.g., on Games) introducing both $\boldsymbol{y}$ and $\boldsymbol{m}_{s_\ell}$ still increases the performance, we keep both $\boldsymbol{y}$ and $\boldsymbol{m}_{s_\ell}$ in the standard item similarity gating function for universality. From Figure 5, we can also see that a feature-level gating brings worse results on four

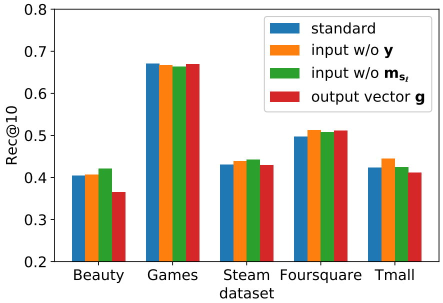


Figure 5: Recommendation performance (Rec@10) of some changes to the gating MLP (see Eq.(9), $B = 1, d = 50$). Note that $\boldsymbol{y}$ and $\boldsymbol{m}_{s_\ell}$ denote the representation of the user's global preference and the recently interacted item, respectively, which are two parts of the inputs of Eq.(9), and $\boldsymbol{g}$ is an output in the form of a 1-D vector.

datasets (except on Foursqaure), though it is expected to refine the weights for different dimensions. Actually, in our experiments we find that a feature-level gating makes the model more unstable, which tends to be trapped into local optimal.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel solution named fusing item similarity models with self-attention networks (FISSA) for sequential recommendation. Specifically, our model contains three main components, i.e., a local representation learning module, a global representation learning module and a gating module to balance these two kinds of representations. We base the local representation learning module on the SASRec [16] model, and design an attentive version of FISM [15] for global representation learning to fill the gap of the deficient consideration on global preference learning in most DL-based sequential recommendation methods (e.g., using simple and crude users' latent representations). We also design a gating network, which takes the relationship among the candidate item, the recent interaction and the global preference of each user into consideration, to deal with the possible uncertainty of users' intention. Extensive empirical studies on five public datasets show that our FISSA achieves the state-of-the-art performance compared with several very competitive baselines. Some ablation and quantitative studies showcase the rationality of our design of the global and gating modules.

In the future, we plan to explore the application of federated machine learning [45] on sequential recommendation and generalize our FISSA to a privacy-aware version. Additionally, more contextual information such as knowledge graph of the items [40, 44] is also worth being incorporated into our FISSA.

ACKNOWLEDGMENTS

We thank the support of National Natural Science Foundation of China Nos. 61872249, 61836005 and 61672358.

REFERENCES

[1] Fabio Aiolli. 2013. Efficient Top-N Recommendation for Very Large Scale Binary Rated Datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. 273–280.

[2] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. 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 (SIGIR '17)*. 335–344.

[3] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*. 108–116.

[4] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*. 152–160.

[5] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI '15)*. 2069–2075.

[6] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming Session-based Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 1569–1577.

[7] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based Recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*. 161–169.

[8] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM '16)*. 191–200.

[9] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. 173–182.

[11] Yun He, Yin Zhang, Weiwen Liu, and James Caverlee. 2020. Consistency-Aware Recommendation for User-Generated Item List Continuation. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 250–258.

[12] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. 843–852.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR '16)*.

[14] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 505–514.

[15] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored Item Similarity Models for top-N Recommender Systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. 659–667.

[16] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *Proceedings of the 18th IEEE International Conference on Data Mining (ICDM '18)*. 197–206.

[17] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. 426–434.

[18] Justin J. Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F. Mokbel. 2012. LARS: A Location-Aware Recommender System. In *Proceedings of the 28th IEEE International Conference on Data Engineering (ICDE '12)*. 450–461.

[19] H. Li, Y. Liu, N. Mamoulis, and D. S. Rosenblum. 2020. Translation-based Sequential Recommendation for Complex Users on Sparse Data. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2020), 1639–1651.

[20] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 1419–1428.

[21] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '17)*. 305–314.

[22] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018*

World Wide Web Conference (WWW '18). 689–698.

[23] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1831–1839.

[24] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP '15)*. 1412–1421.

[25] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical Gating Networks for Sequential Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19)*. 825–833.

[26] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory Augmented Graph Neural Networks for Sequential Recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI '20)*. 5045–5052.

[27] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. 43–52.

[28] Arkadiusz Paterek. 2007. Improving Regularized Singular Value Decomposition for Collaborative Filtering. In *Proceedings of KDD Cup and Workshop*. 39–42.

[29] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi YIn. 2019. 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 (CIKM '19)*. 579–588.

[30] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*. 130–137.

[31] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*. 452–461.

[32] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. 811–820.

[33] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann Machines for Collaborative Filtering. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*. 791–798.

[34] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. 285–295.

[35] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-based Recommender System. *Journal of Machine Learning Research* 6 (2005), 1265–1295.

[36] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. 1441–1450.

[37] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '16)*. 17–22.

[38] Jiaxi Tang and Ke Wang. 2018. Personalized top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*. 565–573.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS '17)*. 6000–6010.

[40] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 950–958.

[41] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI '19)*. 346–353.

[42] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM '16)*. 153–162.

[43] 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 *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19)*. 3940–3946.

[44] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product Knowledge Graph Embedding for E-Commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 672–680.

[45] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 12:1–12:19.

[46] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System Based on Hierarchical Attention Network. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18)*. 3926–3932.

[47] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. 729–732.

[48] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. 582–590.

[49] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using Temporal Data for Making Recommendations. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI '01)*. 580–588.