Feature-level Deeper Self-Attention Network for Sequential Recommendation

Tingting Zhang[1,2], **Pengpeng Zhao**[1,2*], **Yanchi Liu**[3], **Victor S. Sheng**[4], **Jiajie Xu**[1], **Deqing Wang**[5], **Guanfeng Liu**[6] and **Xiaofang Zhou**[7,2]
[1]Institute of AI, School of Computer Science and Technology, Soochow University, China
[2]Zhejiang Lab, China
[3]Rutgers University, New Jersey, USA
[4]University of Central Arkansas, Conway, USA
[5]School of Computer, Beihang University, Beijing, China
[6]Department of Computing, Macquarie University, Sydney, Australia
[7]The University of Queensland, Brisbane, Australia

Abstract

Sequential recommendation, which aims to recommend next item that the user will likely interact in a near future, has become essential in various Internet applications. Existing methods usually consider the transition patterns between items, but ignore the transition patterns between features of items. We argue that only the item-level sequences cannot reveal the full sequential patterns, while explicit and implicit feature-level sequences can help extract the full sequential patterns. In this paper, we propose a novel method named Feature-level Deeper Self-Attention Network (FDSA) for sequential recommendation. Specifically, FDSA first integrates various heterogeneous features of items into feature sequences with different weights through a vanilla attention mechanism. After that, FDSA applies separated self-attention blocks on item-level sequences and feature-level sequences, respectively, to model item transition patterns and feature transition patterns. Then, we integrate the outputs of these two blocks to a fully-connected layer for next item recommendation. Finally, comprehensive experimental results demonstrate that considering the transition relationships between features can significantly improve the performance of sequential recommendation.

1 Introduction

With the quick development of the Internet, sequential recommendation has become essential in various applications, such as ad click prediction, purchase recommendation and web page recommendation. In such applications, each user behavior can be modeled as a sequence of activities in chronological order, with his/her following activity influenced by the previous activities. And sequential recommendation aims to recommend the next item that a user will likely interact by capturing useful sequential patterns from user historical behaviors.

Increasing research interests have been put in sequential recommendation with various models proposed. For modeling sequential patterns, the classic Factorizing Personalized Markov Chain (FPMC) model has been introduced to factorize the user-specific transition matrix by considering the Markov Chains [Rendle *et al.*, 2010]. However, the Markov assumption has difficulty in constructing a more effective relationship among factors [Huang *et al.*, 2018]. With the success of deep learning, Recurrent Neural Network (RNN) methods have been widely adopted in sequential recommendation [Hidasi *et al.*, 2016; Zhao *et al.*, 2019]. These RNN methods usually employ the last hidden state of RNN as the user representation, which is used to predict the next action. Despite the success, these RNN models are difficult to preserve long-range dependencies even using the advanced memory cell structures like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [Chung *et al.*, 2014]. Besides, RNN-based methods need to learn to pass relevant information forward step by step, which makes RNN hard to parallelize [Al-Rfou *et al.*, 2019]. Recently, self-attention networks (SANs) have shown promising empirical results in various NLP tasks, such as machine translation [Vaswani *et al.*, 2017], natural language inference [Shen *et al.*, 2018], and question answering [Li *et al.*, 2019]. One strong point of self-attention networks is the strength of capturing long-range dependencies by calculating attention weights between each pair of items in a sequence. Inspired by self-attention networks, Kang et al. [Kang and McAuley, 2018] proposed Self-Attentive Sequential Recommendation model (SASRec) that applied a self-attention mechanism to replace traditional RNNs for sequential recommendation and achieved the state-of-the-art performance. However, it only considers the sequential patterns between items, ignoring the sequential patterns between features that are beneficial for capturing the user's fine-grained preferences.

Actually, our daily activities usually present transition patterns at the item feature level, i.e., explicit features like category or other implicit features. For example, a user is more likely to buy shoes after buying clothes, indicating that the

*Pengpeng Zhao is the corresponding author and his email is ppzhao@suda.edu.cn

next product's category is highly related to the category of the current product. Here we refer to the user's evolving appetite for structured attributes (e.g., categories) as explicit feature transition. Moreover, an item may also contain some other unstructured attributes, like description texts or image, which present more details of the item. Therefore, we want to mine the user's potential feature-level patterns from these unstructured attributes, which we call implicit feature transition. However, explicit and implicit feature transitions among item features are often overlooked by existing methods. We argue that only the item-level sequences cannot reveal the full sequential patterns, while the feature-level sequences can help achieve this goal better. To this end, in this work, we propose a novel feature-level deeper self-attention network for sequential recommendation. For capturing explicit feature-level transition patterns, instead of using the combined representation of item and its features, we apply separated self-attention blocks on item sequences and feature sequences, respectively, to capture the item-item and feature-feature relationships. Then, we combine the contexts at the item-level and the feature-level to make a recommendation. Moreover, we further investigate how to capture meaningful implicit feature-level transition patterns from heterogeneous attributes of items. We additionally utilize vanilla attention to assist feature-based self-attention block to adaptively select essential features from the various types of attributes of items and further learn potential implicit feature transition patterns. Then, we combine item transition patterns with implicit feature transition patterns to a fully-connected layer for the recommendation. Finally, we conduct extensive experiments on two real-world datasets of a famous E-commerce platform. Experimental results demonstrate that considering feature-level transition patterns can significantly improve the performance of recommendation.

The main contributions of this paper are summarized as follows:

- We propose a novel framework, Feature-level Deeper Self-Attention Network (FDSA), for sequential recommendation. FDSA applies self-attention networks to integrate item-level transitions with feature-level transitions for modeling user's sequential intents.
- Explicit and implicit feature transitions are modeled by applying different self-attention blocks on item sequences and feature sequences, respectively. For obtaining implicit feature transitions, a vanilla attention mechanism is added to assist feature-based self-attention block to adaptively select important features from various item attributes.
- We conduct extensive experiments on two real-world datasets to demonstrate the effectiveness of our proposed method.

2 Related Work

In this section, we review closely related work from two perspectives, which are sequential recommendation and attention mechanisms.

2.1 Sequential Recommendation

Many sequential recommendation methods strove to capture meaningful sequence patterns more efficiently. Most existing sequential approaches focused on Markov Chain based methods and Neural network-based methods. Markov Chain based methods estimated an item-item transition probability matrix and used it to predict the next item given the last interaction of a user. FPMC fused matrix factorization and first-order Markov Chains to capture long-term preferences and short-term item-item transitions respectively [Rendle *et al.*, 2010]. All these Markov Chain based methods have the same deficiency that these models only model the local sequential pattern between every two adjacent items. With the success of neural network, recurrent neural network (RNN) methods are widely adopted in sequence modeling. [Hidasi *et al.*, 2016] proposed GRU4Rec approach to model item transition patterns using Gated Recurrent Unit (GRU). Though RNN is an efficient way to model sequential patterns, it still suffers from several difficulties, such as hard to parallelize, time-consuming, and hard to preserve long-term dependencies even using the advanced memory cell structures like LSTM and GRU.

2.2 Attention Mechanisms

Attention mechanisms are popular in many tasks, such as image/video caption [Chen *et al.*, 2017], machine translation [Chen *et al.*, 2018] and recommendation [He *et al.*, 2018]. Recently, self-attention networks have achieved promising empirical results in machine translation task [Vaswani *et al.*, 2017]. Inspired by Transformer, [Zhou *et al.*, 2018] proposed an attention-based user behavior modeling framework ATRank, which projected user behavior representation into multiple latent spaces and then used the self-attention network to model the influences brought by other behaviors. [Huang *et al.*, 2018] proposed a unified framework CSAN that modeled multiple types of behaviors and various modal items into a common latent space and then applied the self-attention mechanism to extract different aspects of user's behavior sequence. [Zhou *et al.*, 2018; Huang *et al.*, 2018] focused on modeling multiple types of actions, but collecting multiple behaviors in many applications is difficult, so here we only consider modeling single-type behavior. [Kang and McAuley, 2018] applied self-attention network to model sequential recommendation, confirming that self-attention based methods have achieved better performance than RNN.

Different from the above approaches in that they only model the item-level sequences, but we employ separated self-attention blocks on the item-level sequences and the feature sequences, respectively, to learn item transition patterns and feature transition patterns and the experimental results show the significant effects of our model.

3 Feature-level Deeper Self-Attention Network for Sequential Recommendation

In this section, we first describe the problem statement in our work, and then present the architecture of our feature-level deeper self-attention network (FDSA) for next item recommendation.

3.1 Problem Statement

Before going into the details of our proposed model, we first introduce notations used in this paper and define the sequential recommendation problem. We denote a set of users as $\mathcal{U}$ = $\{u_1, u_2, ..., u_N\}$ and a set of items as $\mathcal{I}$ = $\{i_1, i_2, ..., i_M\}$, where N and M are the numbers of users and items, respectively. We use $\mathcal{S}$ = $\{s_1, s_2, ..., s_{|\mathcal{S}|}\}$ to denote a sequence of items in chronological order that a user has interacted with before, where $s_i \in \mathcal{I}$. Each item i has some attributes, such as category, brand, and description text. Here we take category as an example, the category of item i is denoted as $c_i \in \mathcal{C}$, where $\mathcal{C}$ is the set of categories. The goal of sequential recommendation is to recommend the next item that the user may act on, given the user historical activities on items.

3.2 The Network Architecture of Feature-level Deeper Self-Attention (FDSA)

As we mentioned before, daily human activities usually present feature-level (e.g., category-level) transition patterns. In this paper, we propose a novel feature-level deeper self-attention network for sequential recommendation (FDSA). FDSA utilizes not only the item-based self-attention block to learn item-level sequence patterns but a feature-based self-attention block to search for feature-level transition patterns. As shown in Figure 1 FDSA consists of five components, i.e., Embedding layer, Vanilla Attention layer, Item-based self-attention block, Feature-based self-attention block, and Fully-connected layer. Specifically, we first project the sparse representation of items and discrete attributes of items (i.e., one-hot representation) into low-dimensional dense vectors. For text attributes of items, we employ a topic model to extract the topical keywords of these texts, and then apply Word2vector to gain the word vector representation of these keywords. Due to the features (attributes) of item are often heterogeneous and come in different domains and data types. Hence, we utilize a vanilla attention mechanism to assist the self-attention network in selecting important features from the various features of items adaptively. After that, a user's sequence patterns are learned through two self-attention networks, in which the item-based self-attention block is applied to learn item-level sequence patterns, and the feature-based self-attention block is used to capture feature-level transition patterns. Finally, we integrate the outputs of these two blocks to a fully-connected layer for getting the final prediction. Next, we will introduce the details of each component of FDSA.

Embedding layer. Due to the number of user's action sequence is not fixed, we take a fixed-length sequence $s = (s_1, s_2, ..., s_n)$ from user's history sequence to calculate user's historical preferences, where n denotes the maximum length that our model handles. If a user's action sequence is less than n, we add zero-padding to the left side of the sequence to convert the user' action sequence to a fixed-length. If a user's sequence length is greater than n, we take the most recent n behaviors. Similarly, we process the feature sequence in the same way. Let us use the category information as an example. Since each item corresponds to a category, we get a fixed-length category sequence $c = (c_1, c_2, ..., c_n)$. Then, we apply a lookup layer to transform the one-hot vectors of action sequence s and its corresponding category sequence c into dense vector representations. For other categorical features (such as brand, seller), the same way is applied. For the textual features (i.e., description text, title), we first utilize the widely-used topic model to extract the topical keywords of texts, then apply Word2vector model to learn textual semantic representations. In this paper, we extract five topical keywords from the description text and title of each item, and then apply the Mean Pooling method to fuse five topical keyword vectors into a vector representation.

Vanilla attention layer. Since the characteristics of items are often heterogeneous, it is difficult to know which features will determine a user's choice. Therefore, we employ vanilla attention to assist the feature-based self-attention block in capturing the user's varying appetite toward attributes (e.g., categories, brands). Given an item i, its attributes can be embedded as $A_i = \{vec(c_i), vec(b_i), vec(item_i^{text})\}$, where $vec(c_i)$ and $vec(b_i)$ represent the dense vector representation of category and brand of item i, respectively. Also, $vec(item_i^{text})$ denotes the textual feature representation of item i. Formally, the attention network is defined as follows.

$$\boldsymbol{\alpha}_i = softmax(\mathbf{W}^f \mathbf{A}_i + \mathbf{b}^f), \tag{1}$$

where $\mathbf{W}^f$ is $d \times d$ matrice and $\mathbf{b}^f$ is d-dimensional vector. Finally, we compute the feature representation of item i as a sum of the item i's attribute vector representations weighted by the attention scores as follows.

$$\mathbf{f}_i = \boldsymbol{\alpha}_i \mathbf{A}_i. \tag{2}$$

It is worth noting that if item i only considers one feature (e.g., category), then the feature representation of item i is $vec(c_i)$.

Feature-based self-attention block. Since the item-based self-attention block and the feature-based self-attention block only differ in their inputs, we focus on illustrating the process of the feature-based self-attention block in detail. From the above attention layer, we can get a feature representation $\mathbf{f}_i$ for item i. Thus, given a user, we get the feature sequence $f = \{f_1, f_2, ..., f_n\}$. To model category-level transition patterns, we utilize the self-attention network proposed by [Vaswani *et al.*, 2017], which can keep the sequential contextual information and capture the relationships between categories in the category sequence, regardless of their distance. Though the self-attention network can ensure computational efficiency and derive long-term dependencies, it ignores the positional information of the sequential input [Gehring *et al.*, 2017]. Hence, we inject a positional matrix $\mathbf{P} \in \mathbb{R}^{n \times d}$ into the input embedding. Namely, the input matrix of the feature-based self-attention block is

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 + \mathbf{p}_1 \\ \mathbf{f}_2 + \mathbf{p}_2 \\ \cdots \\ \mathbf{f}_n + \mathbf{p}_n \end{bmatrix}. \tag{3}$$

The scaled dot-product attention (SDPA) proposed by [Vaswani *et al.*, 2017] is defined as below:

$$SDPA(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{4}$$

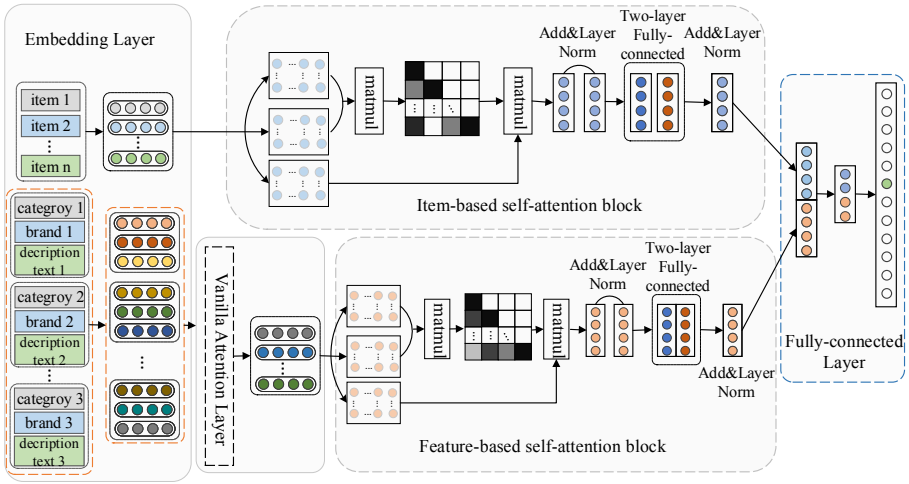

Figure 1: The Network Architecture of FDSA.

where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ represent query, key, and value, respectively, d denotes feature dimension of each feature. Here, query, key and value in the feature-based self-attention block equal to $\mathbf{F}$, we first convert it to three matrices through linear transformation, and then feed them into the SDPA as follows.

$$\mathbf{H}_f = SDPA(\mathbf{F}\mathbf{W}^Q, \mathbf{F}\mathbf{W}^K, \mathbf{F}\mathbf{W}^V), \tag{5}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d\times d}$ are the projection matrices. In order to enable the model to jointly attend to information from different representation subspaces at different positions [Vaswani *et al.*, 2017], the self-attention adopts multi-head attention (MH). The multi-head attention is defined as follows.

$$\begin{aligned} \mathbf{M}_f &= MH(\mathbf{F}) = Concat(h_1, h_2, ..., h_{l_f})\mathbf{W}^O, \\ h_i &= SDPA(\mathbf{F}\mathbf{W}_i^Q, \mathbf{F}\mathbf{W}_i^K, \mathbf{F}\mathbf{W}_i^V), \end{aligned} \tag{6}$$

where $\mathbf{W}^O, \mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ are parameters to be learned and l_f is the number of heads in the feature-based self-attention block. Also, the self-attention network employs a residual connection, a layer normalization and two-layer fully-connected layer with a ReLU activation function to strengthen the performance of the self-attention network. Finally, the output of the feature-based self-attention block is defined as follows.

$$\begin{aligned} \mathbf{M}_f &= LayerNorm(\mathbf{M}_f + \mathbf{F}), \\ \mathbf{O}_f &= ReLU((\mathbf{M}_f\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2), \\ \mathbf{O}_f &= LayerNorm(\mathbf{O}_f + \mathbf{M}_f), \end{aligned} \tag{7}$$

where $\mathbf{W}_*, \mathbf{b}_*$ are model parameters. For the sake of simplicity, we define the entire self-attention block as follows.

$$\mathbf{O}_f = SAB(\mathbf{F}). \tag{8}$$

After the first self-attention block, $\mathbf{O}_f$ essentially aggregates all previous features' embedding. However, it may need to capture more complex feature transitions via another self-attention block based on $\mathbf{O}_f$. Thus, we stack the self-attention block and the q-th $(q > 1)$ block is defined as follows.

$$\mathbf{O}_f^{(q)} = SAB(\mathbf{O}_f^{(q-1)}), \tag{9}$$

where $\mathbf{O}_f^{(0)} = \mathbf{F}$.

Item-based self-attention block. The goal of the item-based self-attention block is to learn meaningful item-level transition patterns. Given a user, we can get an item action sequence s whose corresponding matrix is $\mathbf{S}$. Thus, the output of the stack item-based self-attention block is constructed as follows.

$$\mathbf{O}_s^{(q)} = SAB(\mathbf{O}_s^{(q-1)}), \tag{10}$$

where $\mathbf{O}_s^{(0)} = \mathbf{S}$.

Fully-connected layer. To capture the transition patterns of items and categories simultaneously, we concatenate the output of item-based self-attention block $\mathbf{O}_s^{(q)}$ and the output of feature-based self-attention block $\mathbf{O}_f^{(q)}$ together and project them into a fully-connected layer.

$$\mathbf{O}_{sf} = \left[\mathbf{O}_s^{(q)}; \mathbf{O}_f^{(q)}\right]\mathbf{W}_{sf} + \mathbf{b}_{sf}, \tag{11}$$

where $\mathbf{W}_{sf} \in \mathbb{R}^{2d\times d}$, $\mathbf{b}_{sf} \in \mathbb{R}^d$. Finally, we calculate the user's preference for items through a dot product operation.

$$y_{t,i}^u = \mathbf{O}_{sf_t}\mathbf{N}_i^T, \tag{12}$$

where $\mathbf{O}_{sf_t}$ denotes the t-th line of $\mathbf{O}_{sf}$, $\mathbf{N} \in \mathbb{R}^{M\times d}$ is an item embedding matrix, $y_{t,i}$ is the relevance of item i being the next item given the previous t items $(i.e., s_1, s_2, ..., s_t)$. It is worth noting that the model inputs a sequence $(i_1, i_2, ..., i_{n-1})$ and its expected output is a 'shifted' version of the same sequence: $(i_2, i_3, ..., i_n)$ during training process. In the test process, we take the last row of matrix $\mathbf{O}_{sf}$ to predict the next item.

3.3 The Loss Function for Optimization

In this subsection, to effectively learn from the training process, we adopt the binary cross-entropy loss as the optimization objective function of our FDSA model, which is defined as:

$$L = -\sum_{i\in s}\sum_{t\in[1,2,...,n]}[log(\sigma(y_{t,i})) + \sum_{j\notin s} log(1 - \sigma(y_{t,j}))]. \tag{13}$$

Moreover, for each target item i in each action sequence, we randomly sample a negative item j.

4 Experiments

In this section, we conduct experiments to evaluate the performance of our proposed method FDSA on two real-world datasets. We first briefly introduce the datasets and baseline methods, then we compare FDSA with these baseline methods. Finally, we analyze our experimental results.

Dataset	Tmall	Toys and Games
# users	16,257	35,124
# items	18,678	28,351
# avg. actions/user	15.98	5.51
# Ratings	276,117	228,650

Table 1: Datasets statistics

4.1 Dataset

We perform experiments on two publicly available datasets, i.e., Amazon [1] [Zhou *et al.*, 2018] and Tmall [2] [Tang and Wang, 2018]. Amazon is an E-commerce platform and is widely used for product recommendation evaluation. We adopt a sub-category: Toys and Games. For Toys and Games dataset, we filter users who rated less than 5 items and items that are rated by less than 10 users [Kang and McAuley, 2018]. The feature set of each item contains category, brand, description text on Toys and Games dataset. Tmall, the largest B2C platform in China, is a user-purchase data obtained from IJCAI 2015 competition. We remove items that are observed by less than 30 users and eliminate users who rated less than 15 items [Kang and McAuley, 2018]. The characteristics of each item are category, brand, and seller on Tmall dataset. The statistics of two datasets are summarized in Table 1.

4.2 Evaluation Metrics and Implementation Details

To evaluate the performance of each model for sequential recommendation, we apply two widely used evaluation metrics, i.e., hit ratio (Hit) and normalized discounted cumulative gain (NDCG). Hit ratio measures the accuracy of the recommendation, and NDCG is a position-aware metric which assigns larger weights on higher positions[Yuan *et al.*, 2019]. In our experiments, we choose $K = \{5, 10\}$ to illustrate different results of Hit@K and NDCG@K. Without a special mention in this text, we fix the embedding size of all models to 100 and the batch size to 10. Also, the maximum sequence length n is set to 50 on the two datasets.

4.3 Baseline Methods

We will compare our model FDSA with following baseline methods, which are briefly described as follows.

- **PopRec** ranks items according to their popularity. The most popular items are recommended to users.
- **BPR** [Rendle *et al.*, 2009] is a classic method for building recommendation from implicit feedback data, which proposes a pair-wise loss function to model the relative preferences of users.
- **FPMC** [Rendle *et al.*, 2010] fuses matrix factorization and first-order Markov Chains to capture long-term preferences and short-term item-item transitions, respectively, for next item recommendation.
- **TransRec** [He *et al.*, 2017] regards users as a relational vector acting as the junction between items.
- **GRU4Rec** [Hidasi *et al.*, 2016] applies GRU to model user click sequences for session-based recommendation.
- **CSAN** [Huang *et al.*, 2018] can model multi-type actions and multi-modal contents based on the self-attention network. Here we only consider content and behavior in datasets.
- **SASRec** [Kang and McAuley, 2018] is a self-attention-based sequential model, and it can consider consumed items for next item recommendation.
- **SASRec+** is our extension to the SASRec method, which concatenates item vector representations and category vector representations together as the input of the item-level self-attention network.
- **SASRec++** is our extension of SASRec method, which splices item representations and various heterogeneous features of items together as the input of the item-level self-attention mechanism.
- **CFSA** is a simplified version of our proposed method, which only considers a category feature. It applies separated self-attention blocks on the item-level sequences and the category-level sequences, respectively.

4.4 Performance Comparison

We compare the performance of FDSA with ten baselines regarding Hit and NDCG with cutoffs at 5 and 10. Table 2 reports their overall experimental performances on the two datasets. We summarize the experimental analysis as follows.

Firstly, both BPR and GRU4Rec outperform PopRec on the two datasets. This suggests the effectiveness of personalized recommendation methods. Among the baseline methods, the sequential model (e.g., FPMC and TransRec) usually perform better than the non-sequential model (i.e., BPR) on the two datasets. This demonstrates the importance of considering sequential information in next item recommendation.

Secondly, compared with FPMC and TransRec, SASRec performs better performance in terms of the two metrics. This confirms the advantages of using a self-attention mechanism to model a sequence pattern. Although CSAN splices the heterogeneous features of the item in the item representation to help the self-attention mechanism learn the sequential patterns, the self-attention mechanism may only be able to better model temporal order information. However, SASRec employs not only self-attention mechanism to capture long-term preferences but also considers short-term preferences (i.e., last action) through a residual connection.

Thirdly, SASRec+ and SASRec++ achieve a better result than SASRec on the Toys and Games dataset and perform worse than SASRec on the Tmall dataset. This phenomenon

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

[2] https://tianchi.aliyun.com/competition

Dataset	Method	@5		@10	
		Hit	NDCG	Hit	NDCG
Tmall	PopRec	0.1532	0.0988	0.2397	0.1267
	BPR	0.1749	0.1129	0.2647	0.1418
	FPMC	0.2731	0.2034	0.3680	0.2339
	TransRec	0.2652	0.1854	0.3773	0.2214
	GRU4Rec	0.1674	0.1217	0.2446	0.1465
	CSAN	0.3481	0.2440	0.4787	0.2863
	SASRec	0.3572	0.2531	0.4840	0.2940
	SASRec+	0.3427	0.2415	0.4714	0.2829
	SASRec++	0.3550	0.2534	0.4785	0.2932
	CFSA	0.3836	0.2724	0.5152	0.3149
	FDSA	**0.3940**	**0.2820**	**0.5197**	**0.3226**
Toys and Games	PopRec	0.1952	0.1287	0.3058	0.1643
	BPR	0.2096	0.1394	0.3219	0.1756
	FPMC	0.2983	0.2261	0.3833	0.2535
	TransRec	0.3135	0.2255	0.4206	0.2600
	GRU4Rec	0.2039	0.1359	0.3118	0.1705
	CSAN	0.2327	0.1601	0.3404	0.1947
	SASRec	0.3292	0.2334	0.4441	0.2705
	SASRec+	0.3367	0.2410	0.4510	0.2776
	SASRec++	0.3394	0.2428	0.4544	0.2799
	CFSA	0.3391	0.2411	0.4538	0.2782
	FDSA	**0.3571**	**0.2572**	**0.4738**	**0.2949**

Table 2: Experimental results of FDSA and baselines. The best performance of each column (the larger is the better) is in bold.

can be explained that the sequential patterns may not be stably modeled by concatenating items' representations and items' feature representations together as input vectors of the self-attention mechanism. Moreover, the performance of CFSA is better than SASRec+, and FDSA surpasses SASRec++. This demonstrates that applying separated self-attention blocks on item-level sequences and feature-level sequences, respectively, to capture item transition patterns and feature transition patterns (i.e., CFSA and FDSA) is more effective than splicing item representations and its feature representations as the input to a self-attention mechanism (i.e., SASRec+ and SASRec++). The above experiments demonstrate that modeling item and feature transition patterns through two separate independent item-level and feature-level sequences is valuable and meaningful for sequential recommendation.

Finally, regardless of the datasets and the evaluation metrics, our proposed FDSA achieves the best performance. Our degenerated model CFSA consistently beats most baseline methods. This shows the effectiveness of modeling independent category-level sequences by the self-attention network. FDSA performs better than CFSA, indicating the effectiveness of modeling more features in feature-level sequences.

Dataset	model	NDCG l_s \ l_f	$l_f = 2$	$l_f = 4$
Tmall	CFSA	$l_s = 2$	0.3058	0.3060
		$l_s = 4$	**0.3149**	0.3146
	FDSA	$l_s = 2$	0.3120	0.3176
		$l_s = 4$	**0.3226**	0.3211
Toys and Games	CFSA	$l_s = 2$	0.2600	**0.2782**
		$l_s = 4$	0.2764	0.2729
	FDSA	$l_s = 2$	0.2759	**0.2949**
		$l_s = 4$	0.2799	0.2791

Table 3: The Performance of FDSA and CFSA with varying l_s and l_f in terms of NDCG@10 on two datasets.

4.5 Influence of Hyper-parameters

We investigate the influence of hyper-parameters, such as the embedding size d, the number of heads in item-based self-attention block l_s and the number of heads in feature-based self-attention block l_f. Due to space limitation, we only show the experimental results of NDCG@10. We have obtained similar experimental results on the Hit@10 metric.

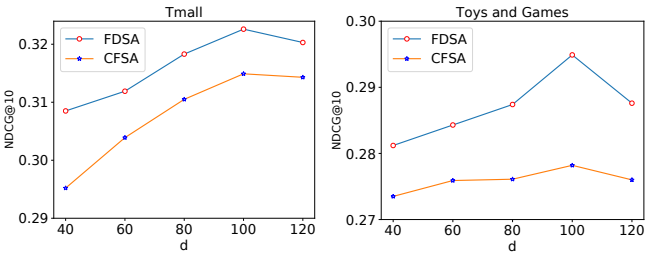

Figure 2: The performance of FDSA and CFSA under difference choices of d.

Influence of embedding size d. Figure 2 shows the performance of our model with different embedding sizes d on the two datasets. As we can see from Figure 2, high dimensions can model more information for items, but when the dimension exceeds 100, the performance of FDSA and CFSA degrade. This demonstrates that over-fitting may occur when the implicit factor dimension of the model is too high.

Influence of the number of heads l_s and l_f. We conduct experiments to study the performance of our model with varying l_s and l_f on the two datasets. Table 3 demonstrates the experimental result in term of NDCG@10. We can observe that CFSA and FDSA achieve the best performance with the setting $l_s = 4$, $l_f = 2$ on the Tmall dataset, while they get the best result with the setting $l_s = 2$, $l_f = 4$ on the Toys and Games dataset. This may be because our model needs more heads to capture the transition relationships between features due to each item contains a descriptive text and a title in the Toys and Games dataset, while the single data type of the features of these items on Tmall dataset may not require too complicated structures to model the relationships between the features.

5 Conclusion

In this paper, a novel method named Feature-level Deeper Self-Attention Network (FDSA) is proposed for sequential recommendation. FDSA modeled the transition patterns between items through an item-based self-attention block, and it also learned the transition patterns between features by a feature-based self-attention block. Then, the outputs of these two blocks are integrated into a fully-connected layer for next item prediction. Extensive experimental results have shown that our model outperformed the state-of-the-art baseline methods.

Acknowledgments

This research was partially supported by NSFC (No. 61876117, 61876217, 61872258, 61728205), Major Project of Zhejiang Lab (No. 2019DH0ZX01), Open Program of Key Lab of IIP of CAS (No. IIP2019-1) and PAPD.

References

[Al-Rfou *et al.*, 2019] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. *In AAAI*, 2019.

[Chen *et al.*, 2017] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6298–6306. IEEE, 2017.

[Chen *et al.*, 2018] Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. Syntax-directed attention for neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[He *et al.*, 2017] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 161–169. ACM, 2017.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2354–2366, 2018.

[Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *In ICLR*, 2016.

[Huang *et al.*, 2018] Xiaowen Huang, Shengsheng Qian, Quan Fang, Jitao Sang, and Changsheng Xu. Csan: Contextual self-attention network for user sequential recommendation. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 447–455. ACM, 2018.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[Li *et al.*, 2019] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010.

[Shen *et al.*, 2018] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573. ACM, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[Yuan *et al.*, 2019] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 582–590. ACM, 2019.

[Zhao *et al.*, 2019] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Zhixu Li, Jiajie Xu, and Victor S Sheng. Where to go next: A spatio-temporal lstm model for next poi recommendation. *In AAAI*, 2019.

[Zhou *et al.*, 2018] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. Atrank: An attention-based user behavior modeling framework for recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.