逐次推薦のための特徴量レベルの深い自己アテンションネットワーク

Tingting Zhang[1,2], Pengpeng Zhao[1,2], Yanchi Liu[3], Victor S. Sheng[4],
Jiajie Xu[1], Deqing Wang[1], Guanfeng Liu[5] and Xiaofang Zhou[6]
[1] 中国・スーチョー大学コンピュータ科学技術学部AI研究所
[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] マシーナ大学計算機学部、シドニー、オーストラリア
The University of Queensland, Brisbane, Australia

Abstract

ユーザが近い将来交流する可能性が高い次のアイテムを推薦することを目的とした逐次推薦は、様々なインターネットアプリケーションにおいて不可欠となっている。既存の手法は、通常、アイテム間の遷移パターンを考慮するが、アイテムの特徴間の遷移パターンは無視する。我々は、アイテムレベルのシーケンスのみでは完全な逐次パターンを明らかにすることができず、明示的および暗黙的な特徴レベルのシーケンスは完全な逐次パターンの抽出に役立つと主張する。本論文では、逐次推薦のためのFeature-level Deeper SelfAttention Network (FDSA)と名付けた新しい手法を提案する。具体的には、まず、FDSAはバニラアテンション機構により、アイテムの様々な異種特徴を異なる重みの特徴列に統合する。その後、FDSAはアイテムレベルのシーケンスと特徴レベルのシーケンスに対して、それぞれ分離した自己注意ブロックを適用し、アイテムの遷移パターンと特徴の遷移パターンをモデル化する。そして、この2つのブロックの出力を完全連結層に統合し、次のアイテム推薦を行う。最後に、包括的な実験結果により、特徴量間の遷移関係を考慮することで、逐次推薦の性能を大幅に向上させることができること

1 Introduction

インターネットの急速な発展に伴い、広告クリック予測、購入推薦、ウェブページ推薦など、様々なアプリケーションで逐次推薦が不可欠となっている。このようなアプリケーションでは、各ユーザーの行動は時系列に並んだ活動のシーケンスとしてモデル化され、その後の行動は以前の活動に影響される。また、逐次推薦では、ユーザが次に対話する可能性のあるアイテムを推薦することを目的としている。

^{}} Pengpeng Zhaoはcorresponding authorであり、彼の電子メールはppzhao@suda.edu.cnで、ユーザーの過去の行動から有用な連続パターンをキャプチャしています。

様々なモデルが提案され、逐次推薦の研究関心が高まっている。逐次パターンのモデル化のために、古典的な因子分解パーソナライズドマルコフ連鎖(FPMC)モデルが、マルコフ連鎖を考慮してユーザー固有の遷移行列を因子分解するために導入されている[Rendle et al., 2010]。しかし、マルコフ仮定は、因子間のより効果的な関係を構築することが困難である[Huang et al., 2018]。深層学習の成功に伴い、リカレントニューラルネットワーク(RNN)法は逐次推薦で広く採用されている[Hidasi et al., 2016; Zhao et al., 2019]。これらのRNN手法は通常、RNNの最後の隠れ状態をユーザ表現として採用し、次の行動を予測するために使用される。成功にもかかわらず、これらのRNNモデルは、Long Short-Term Memory (LSTM) や Gated Recurrent Units (GRU) などの高度なメモリセル構造を用いても長距離依存性を保持することが困難である [Chung et al., 2014]。その上、RNNベースの手法は、関連する情報を段階的に前に渡すことを学習する必要があり、RNNの並列化が困難である[Al-Rfou et al., 2019]。最近、自己注意ネットワーク(SAN)は、機械翻訳[Vaswani et al., 2017]、自然言語推論[Shen et al., 2018]、質問応答[Li et al., 2019]など、様々なNLPタスクで有望な経験的結果を示している。自己注意ネットワークの強力なポイントの1つは、シーケンス内のアイテムの各ペア間の注意重みを計算することによって長距離依存性を捉えることの強さである。自己注意ネットワークに触発され、Kang et al.[Kang and McAuley, 2018] は、逐次推薦のために従来の RNN を置き換えるために自己注意メカニズムを適用し、最先端の性能を達成した自己注意型逐次推薦モデル（SASRec）を提案した。しかし、これはアイテム間の逐次的なパターンのみを考慮し、ユーザのきめ細かい嗜好を捉えるのに有益な特徴間の逐次的なパターンを無視するものである。

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

実際、我々の日常的な活動では、通常、アイテムの特徴量レベルでの遷移パターン、すなわち、カテゴリや他の暗黙の特徴量などの明示的な特徴量を提示する。例えば、あるユーザは服を買った後に靴を買う可能性が高く、次の商品のカテゴリーが現在の商品のカテゴリーと高い関連性を持っていることを示している。ここでは、ユーザが進化する構造化属性(例えば、カテゴリ)に対する欲求を明示的特徴遷移と呼ぶことにする。さらに、アイテムには、説明文や画像など、より詳細なアイテムを提示する非構造化属性も含まれることがある。そこで、これらの非構造化属性から、ユーザの潜在的な特徴レベルのパターンを抽出することを、暗黙的特徴遷移と呼ぶ。しかし、項目特徴間の明示的・暗黙的な特徴量の遷移は、既存の手法では見落とされがちである。我々は、項目レベルのシーケンスだけでは完全な順序パターンを明らかにできず、特徴レベルのシーケンスはこの目標の達成に役立つと主張する。このため、本研究では、逐次推薦のための新しい特徴レベルの深い自己注意ネットワークを提案する。明示的な特徴レベルの遷移パターンを捉えるために、項目とその特徴を組み合わせた表現を用いるのではなく、項目列と特徴列に対してそれぞれ分離した自己注意ブロックを適用し、項目-項目、特徴-特徴の関係を捉える。そして、項目レベルの文脈と特徴レベルの文脈を組み合わせて推薦を行う。さらに、項目の異種属性から意味のある暗黙的な特徴レベルの遷移パターンを捉える方法をさらに検討する。さらに、バニラアテンションを利用して、特徴に基づく自己アテンションブロックを支援し、アイテムの様々な属性から本質的な特徴を適応的に選択し、潜在的な暗黙的特徴遷移パターンをさらに学習させる。そして、暗黙的特徴遷移パターンを持つ項目遷移パターンを組み合わせて、推薦のための完全連結層とする。最後に、有名なEコマースプラットフォームの2つの実世界データセットに対して、広範な実験を行った。実験の結果、特徴レベルの遷移パターンを考慮することで、推薦の性能を大幅に向上させることができることが示された。本論文の主な貢献は以下のようにまとめられる。

- 我々は、逐次推薦のための新しいフレームワーク、Feature-level Deeper Self-Attention Network (FDSA)を提案する。FDSAは自己注意ネットワークを適用し、アイテムレベルの遷移と特徴レベルの遷移を統合し、ユーザの逐次的な意図をモデル化する。

- 明示的特徴遷移と暗黙的特徴遷移は、それぞれ項目列と特徴列に異なる自己アテンションブロックを適用してモデル化される。暗黙的な特徴遷移を得るために、バニラアテンション機構を追加し、特徴に基づく自己アテンションブロックを支援し、様々な項目属性から重要な特徴を適応的に選択する。

- 提案手法の有効性を実証するために、2つの実世界データセットで広範な実験を行った。

2 Related Work

本節では、逐次推薦と注意のメカニズムという二つの観点から、密接に関連する研究をレビューする。

2.1 Sequential Recommendation

多くの逐次推薦手法は、意味のあるシーケンスパターンをより効率的に捉えることを目的としていた。既存の逐次推薦手法の多くは、マルコフ連鎖に基づく手法とニューラルネットワークに基づく手法に着目していた。マルコフ連鎖に基づく手法は、アイテム-アイテム遷移確率行列を推定し、それを用いてユーザの最後のインタラクションが与えられたときに次のアイテムを予測する。FPMCは長期的な嗜好と短期的なアイテム-アイテム遷移をそれぞれ捉えるために、行列分解を融合し、1次マルコフ連鎖を用いる[Rendle et al., 2010]。これらのマルコフ連鎖に基づく手法はすべて、これらのモデルが隣接する2つのアイテム間の局所的な順序パターンのみをモデル化するという欠点がある。ニューラルネットワークの成功により、リカレントニューラルネットワーク(RNN)法がシーケンスモデリングに広く採用されるようになった。[Hidasi et al., 2016]は、Gated Recurrent Unit (GRU)を用いてアイテム遷移パターンをモデル化するGRU4Recアプローチを提案した。RNNは逐次パターンを効率的にモデル化する方法であるが、LSTMやGRU

2.2 Attention Mechanisms

注意メカニズムは、画像/映像キャプション[Chen et al., 2017]、機械翻訳[Chen et al., 2018]、推薦[He et al., 2018]など多くのタスクで人気がある。近年、自己注意ネットワークは機械翻訳タスクにおいて有望な実証結果を達成している[Vaswani et al., 2017]。Transformerに触発され、[Zhou et al., 2018]は、ユーザーの行動表現を複数の潜在空間に投影し、自己注意ネットワークを用いて他の行動によってもたらされる影響をモデル化する、注意に基づくユーザー行動モデリングフレームワークATRankを提案した。[Huang et al., 2018]は、複数のタイプの行動と様々なモードアイテムを共通の潜在空間にモデル化し、その後、自己注意メカニズムを適用してユーザーの行動シーケンスの異なる側面を抽出する統一フレームワークCSANを提案した。[Zhou et al., 2018; Huang et al., 2018]は、複数のタイプのアクションのモデル化に焦点を当てたが、多くのアプリケーション[Kang and McAuley, 2018] は、逐次推薦のモデルに自己注意ネットワークを適用し、自己注意に基づく方法がRNNよりも良い性能を達成していることを確認した。上記のアプローチとは異なり、アイテムレベルのシーケンスのみをモデル化するが、アイテムレベルのシーケンスと特徴シーケンスにそれぞれ分離した自己注意ブロックを採用し、アイテム遷移パターンと特徴遷移パターンを学習し、実験結果は我々のモデルの有意な効果を示している。

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

本節では、まず、我々の研究で問題提起を行った後、次アイテム推薦のための特徴量レベルの深層自己注意ネットワーク(FDSA)のアーキテクチャを紹介する。

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

提案モデルの詳細に入る前に、まず本論文で使用する表記法を紹介し、逐次推薦問題を定義する。ユーザの集合を U = {u_1, u_2, ・・・、u_N }、アイテムの集合を I = {i_1, i_2, ・・・、i_M } とし、N と M はそれぞれユーザとアイテムの数である。ここで、S = {s_1, s_2, ・・・, s_{S_} } を用いて、ユーザが以前に交流したアイテムの並びを時系列で表す(s_i ∈ I)。各アイテムiは、カテゴリ、ブランド、説明文などの属性を持つ。ここでは、カテゴリを例にとると、アイテムiのカテゴリはc_i∈Cと表記され、Cはカテゴリの集合である。逐次推薦の目的は、アイテムに関するユーザの過去の活動から、ユーザが次に行動しうるアイテムを推薦することである。

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

前述したように、人間の日常的な活動は、通常、特徴レベル(例えば、カテゴリレベル)の遷移パターンを提示する。本論文では、逐次推薦のための新しい特徴レベルの深い自己注意ネットワーク(FDSA)を提案する。FDSAは、アイテムレベルのシーケンスパターンを学習するためにアイテムベースの自己注意ブロックを利用するだけでなく、特徴ベースの自己注意ブロックを利用して、特徴レベルの遷移パターンを探索する。図1に示すように、FDSAは、埋め込み層、バニラ注意層、アイテムベース自己注意ブロック、特徴ベース自己注意ブロック、完全連結層の5つのコンポーネントから構成されている。具体的には、まず、アイテムの疎な表現とアイテムの離散的な属性(すなわち、ワンショット表現)を低次元の密なベクトルに投影する。アイテムのテキスト属性については、トピックモデルを用いてこれらのテキストのトピックキーワードを抽出し、Word2vectorを適用してこれらのキーワードの単語ベクトル表現を得る。アイテムの特徴(属性)は異種であることが多く、ドメインやデータの種類が異なるため、このような特徴を利用する。そこで、バニラアテンション機構を利用して、アイテムの様々な特徴から重要な特徴を適応的に選択する自己アテンションネットワークを支援する。その後、アイテムベースの自己アテンションブロックを適用してアイテムレベルのシーケンスパターンを学習し、特徴ベースの自己アテンションブロックを用いて特徴レベルの遷移パターンを捕捉する2つの自己アテンションネットワークにより、ユーザのシーケンスパターンを学習する。最後に、この2つのブロックの出力を完全連結層に統合し、最終的な予測を得る。次に、FDSAの各コンポーネントの詳細を紹介する。

埋め込み層。ユーザの行動シーケンスの数は固定ではないので、ユーザの履歴列から固定長のシーケンスs = (s_1, s_2, --, s_n) を取り出し、ユーザの履歴嗜好を計算する(nは我々のモデルが扱う最大長を表す)。ユーザの行動シーケンスがnより小さい場合、シーケンスの左側にゼロパディングを追加し、ユーザの行動シーケンスを固定長に変換する。ユーザのシーケンス長がnより大きい場合、直近のn個の行動を取る。同様に、特徴列も同様に処理する。カテゴリ情報を例にとってみよう。各項目はカテゴリに対応するので、固定長のカテゴリ列c = (c_1, c_2, --, c_n) が得られる。次に、ルックアップ層を適用して、行動列sとそれに対応するカテゴリ列cのワンホットベクトルを密なベクトル表現に変換する。他のカテゴリ特徴(ブランド、売り手など)についても、同様の方法が適用される。テキスト特徴量(説明文、タイトル)については、まず、広く用いられているトピックモデルを利用してテキストのトピックキーワードを抽出し、次にWord2vectorモデルを適用してテキストの意味表現を学習する。本論文では、説明文と各項目のタイトルから5つのトピックキーワードを抽出し、Mean Pooling法を適用して5つのトピックキーワードベクトルをベクトル表現に融合させる。バニラアテンション層。アイテムの特性はしばしば異質であるため、どの特徴がユーザの選択を決定するかを知ることは困難である。そこで、バニラアテンションを用いて、特徴に基づく自己アテンションブロックを支援し、ユーザの属性(カテゴリ、ブランドなど)に対する様々な嗜好を捉える。アイテムiが与えられたとき、その属性は

embedded as $A_i = \{vec(c_i), vec(b_i), vec(item_i^{\quad})\}$, where $vec(c_i)$ and $vec(b_i)$ represent the dense vector representa-

は、それぞれカテゴリーとアイテムiのブランドの表現である。また $vec(item_i^{text})$ denotes the textual feature representation of item i. Formally, the attention network is defined as follows.

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

ここで、W^fはd×d行列、b^fはd次元ベクトルである。最後に、アイテムiの特徴表現を、アイテムiの属性ベクトル表現を注目スコアで重み付けした和として、以下のように計算する。

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

ここで注目すべきは、アイテム i が一つの特徴(例えば、カテゴリ)しか考慮しない場合、アイテム i の特徴表現は vec(c_i) であることである。特徴量に基づく自己アテンションブロック。アイテムベース自己アテンションブロックと特徴ベース自己アテンションブロックは入力が異なるだけなので、特徴ベース自己アテンションブロックのプロセスを詳細に説明することに焦点を当てる。上記の注意層から、アイテムiの特徴表現f_iを得ることができる。したがって、ユーザから与えられた特徴列f = {f_1, f_2, --, f_n }を得ることができる。カテゴリレベルの遷移パターンをモデル化するために、我々は[Vaswani et al., 2017]によって提案された自己注意ネットワークを利用し、カテゴリシーケンス内の連続した文脈情報を保持し、カテゴリ間の距離に関係なく関係を捉えることができる。自己注意ネットワークは計算効率を確保し、長期依存性を導出することができるが、逐次入力の位置情報を無視する[Gehring et al., 2017]。そこで、入力埋め込みに位置行列P∈$R^{n×d}$を注入する。すなわち、特徴量に基づく自己注意ブロックの入力行列は

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

Vaswani et al., 2017]によって提案されたスケーリングドットプロダクトアテンション(SDPA)は、以下のように定義される。

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

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:

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

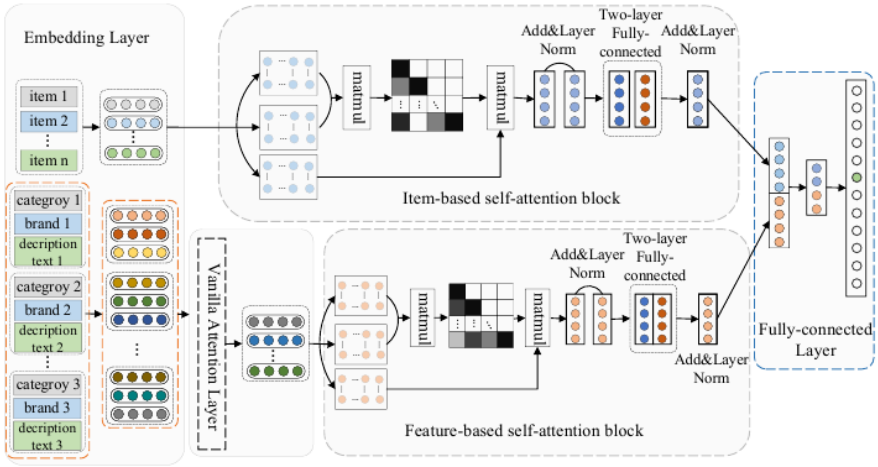

図1:FDSAのネットワークアーキテクチャ

ここで、Q, K, V はそれぞれ query, key, value を表し、d は各特徴の特徴次元を表す。ここで、特徴ベース自己アテンションブロックにおける query, key, value は F に等しいので、まず線形変換により 3 つの行列に変換し、以下のように SDPA に投入する。

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

ここで、$\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V \in \mathbb{R}^{d\times d}$ は射影行列である。異なる位置の異なる表現部分空間からの情報にモデルが共同で注意することを可能にするために[Vaswani et al., 2017]、自己注意はマルチヘッド注意(MH)を採用する。マルチヘッド注意は以下のように定義される。

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

ここで、$\mathbf{W}^O$, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ は学習するパラメータ、l_f は特徴ベース自己アテンションブロックのヘッド数である。また、自己アテンションネットワークは、自己アテンションネットワークの性能を強化するために、残差接続、層正規化、2層完全連結層とReLU活性化関数を採用している。最後に、特徴量に基づく自己アテンションブロックの出力は以下のように定義される。$\mathbf{M}_f = \text{LayerN orm}(\mathbf{M}_f + \mathbf{F})$,

$$\mathbf{O}_f = ReLU((\mathbf{M}_f\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2), \tag{7}$$

$\mathbf{O}_f = \text{LayerN orm}(\mathbf{O}_f + \mathbf{M}_f)$, ここで W, b はモデルパラメータである。簡単のために、自己アテンションブロック全体を以下のように定義する。

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

最初の自己アテンションブロックの後、$\mathbf{O}_f$は本質的に以前のすべての特徴の埋め込みを集約する。しかし、$\mathbf{O}_f$ に基づく別の自己アテンションブロックによって、より複雑な特徴遷移を捉える必要がある場合がある。 そこで、自己アテンションブロックを積み重ね、q番目の(q > 1)ブロックを以下のように定義する。

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

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

項目ベース自己アテンションブロック。アイテムベース自己アテンションブロックの目標は、意味のあるアイテムレベルの遷移パターンを学習することである。ユーザがあれば、対応する行列がSであるアイテムアクション列sを得ることができる。したがって、スタックアイテムベース自己アテンションブロックの出力は以下のように構成される。

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

ここで、$\mathbf{O}_s^{(0)} = \mathbf{S}$. 完全連結層。アイテムとカテゴリの遷移パターンを同時に捉えるために、アイテムベースの自己注意ブロック $\mathbf{O}_s^{(q)}$ の出力と(q)特徴ベースの自己注意ブロック $\mathbf{O}_f$ の出力を連結して完全連結層に射影する。

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

ここで、$\mathbf{W}_{sf} \in \mathbb{R}^{2d\times d}$, $\mathbf{b}_{sf} \in \mathbb{R}^d$. 最後に、ドットプロダクト操作により、ユーザのアイテムに対する嗜好を計算する。

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

ここで、$\mathbf{O}_{sft}$は$\mathbf{O}_{sf}$のt番目の行を表し、$\mathbf{N}\in\mathbb{R}^{M\times d}$は項目埋め込み行列、$y_{t,i}$は項目iが前のt個の項目を与えられた次の項目であることの関連性(すなわち、$s_1, s_2, --, s_t$)である。注目すべきは、モデルがシーケンス($i_1, i_2, --, i_{n-1}$)を入力し、その期待出力が同じシーケンスの「シフト」版:($i_2, i_3, --, i_n$)を学習過程で入力していることである。テストプロセスでは、行列$\mathbf{O}_{sf}$の最後の行を取り、次の項目を予測する。

3.3 最適化のための損失関数

このサブセクションでは、学習プロセスから効果的に学習するために、我々のFDSAモデルの最適化目的関数として、バイナリクロスエントロピー損失を採用し、以下のように定義される。$L = -\sum_{i\in s}\sum_{t\in[1,2,...,n]} [\log(y_{t,i})) + \sum_{j\notin s} \log(1 - \sigma(y_{t,j}))]$.

(13)

さらに、各行動列の各対象項目iについて、負の項目jをランダムにサンプリングする。

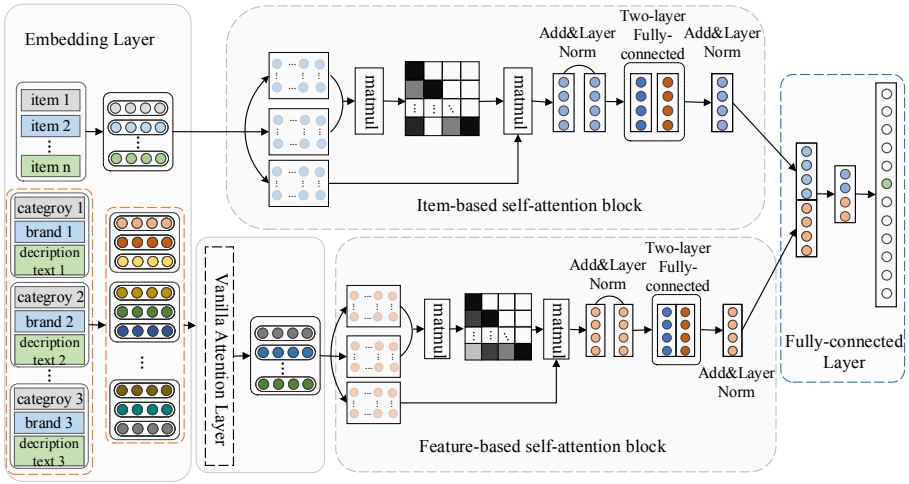

Figure 1: The Network Architecture of FDSA.

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

本節では、提案手法FDSAの性能を2つの実世界データセットで評価するための実験を行う。まず、データセットとベースライン手法を簡単に紹介し、次にFDSAとベースライン手法を比較する。最後に、実験結果について分析する。

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

我々は、2つの一般に利用可能なデータセット、すなわち、Amazon[1] [Zhou et al., 2018] とTmall[2] [Tang and Wang, 2018] に対して実験を行う。AmazonはEコマースプラットフォームであり、商品推薦の評価に広く利用されている。サブカテゴリを採用する。ToysとGamesである。ToysとGamesデータセットについては、10人未満のユーザーによって評価されたアイテムとアイテムが5未満のユーザーをフィルタリングします[Kang and McAuley, 2018]。各アイテムの特徴セットには、ToysとGamesのデータセットにおけるカテゴリ、ブランド、説明文が含まれています。中国最大のB2CプラットフォームであるTmallは、IJCAI 2015コンペティションから取得したユーザー購入用データである。30 人未満のユーザが観測したアイテムを削除し、15 人未満のアイテムを評価したユーザーを排除する[Kang and McAuley, 2018]。各アイテムの特徴は、Tmall データセットにおけるカテゴリ、ブランド、セラーである。2 つのデータセットの統計量を

4.2 Evaluation Metrics and Implementation Details

逐次推薦に対する各モデルの性能を評価するために、広く用いられている2つの評価指標、すなわち、ヒット率(Hit)と正規化割引累積利得(NDCG)を適用する。Hit ratioは推薦の精度を測定し、NDCGは高い位置に大きな重みを割り当てる位置考慮型メトリックである[Yuan et al., 2019]。我々の実験では、Hit@KとNDCG@Kの異なる結果を説明するために、K = {5, 10}を選択する。本文中で特に言及しないが、全てのモデルの埋め込みサイズを100に固定し、バッチサイズを10に固定する。また、2つのデータセットにおいて、最大配列長nは50に設定されている。

4.3 Baseline Methods

本モデルのFDSAを以下のベースライン手法と比較する。

- PopRec は、アイテムの人気度に応じてランク付けを行う。最も人気のあるアイテムは、ユーザに推薦される。- BPR [Rendle et al., 2009] は、暗黙のフィードバックデータから推薦を構築するための古典的な手法であり、以下のような特徴がある。は、ユーザの相対的な嗜好をモデル化するために、ペアワイズ損失関数を提案している。FPMC [Rendle et al., 2010] は、行列分解と一次マルコフ連鎖を融合し、それぞれ長期的な嗜好と短期的な項目-項目遷移を捉え、次の項目推薦を行う。TransRec [He et al., 2017] は、ユーザーをアイテム間の接続として作用する関係ベクトルと見なす。GRU4Rec [Hidasi et al., 2016] は、セッションベースの推薦のために、ユーザーのクリックシーケンスをモデル化するためにGRUを適用する。CSAN [Huang et al., 2018]は、自己注意ネットワークに基づいて、マルチタイプアクションとマルチモーダルコンテンツをモデル化することができる。ここでは、データセットに含まれるコンテンツと行動のみを考慮する。SASRec [Kang and McAuley, 2018] は自己注意に基づく逐次モデルであり、消費されたアイテムを次アイテム推薦のために考慮することができる。

- SASRec+はSASRec法の拡張であり、アイテムレベルの自己注視ネットワークの入力として、アイテムベクトル表現とカテゴリベクトル表現を連結したものである。

- SASRec++は、SASRec法を拡張したもので、アイテムレベルの自己注 意メカニズムの入力として、アイテム表現とアイテムの様々な異種特徴を一緒にスプライスするものである。

- CFSA は提案手法の簡略版であり、カテゴリ特徴のみを考慮する。アイテムレベルシーケンスとカテゴリレベルシーケンスに対して、それぞれ分離された自己アテンションブロックを適用する。

4.4 Performance Comparison

HitとNDCGに関して、FDSAの性能を10のベースラインと比較し、5と10でカットオフした場合の性能を比較する。表2は、2つのデータセットにおける実験全体の性能を報告する。実験解析の結果を以下のようにまとめる。まず、BPRとGRU4Recは、2つのデータセットにおいて、PopRecを上回った。これは、パーソナライズされた推薦手法の有効性を示唆している。ベースライン手法のうち、逐次モデル(FPMCやTransRecなど)は、通常、2つのデータセットにおいて、非逐次モデル(BPRなど)よりも良い性能を示す。これは、次アイテム推薦において逐次情報を考慮することの重要性を示している。次に、FPMCやTransRecと比較して、SASRecは2つの指標においてより良い性能を発揮する。これは、シーケンスパターンをモデル化するために自己注意メカニズムを用いることの利点を確認するものである。CSANはアイテム表現においてアイテムの異種特徴をスプライスし、自己注意メカニズムが逐次パターンを学習するのを助けるが、自己注意メカニズムは時間順序情報をよりよくモデル化することしかできないかもしれない。しかし、SASRecは長期的な嗜好を捉えるために自己注意メカニズムを用いるだけでなく、残差接続による短期的な嗜好(すなわち、最後の行動)も考慮する。第三に、SASRec+とSASRec++は、ToysとGamesのデータセットでSASRecより良い結果を達成し、TmallのデータセットではSASRecより悪い結果となった。

[1] http://jmcauley.ucsd.edu/data/amazon/links.html
[2] https://tianchi.aliyun.com/competition

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**

表2:FDSAとベースラインの実験結果。各列の最高性能(大きいほど良い)は太字で表示。

この現象は、アイテムの表現とアイテムの特徴表現を連結して自己注意メカニズムの入力ベクトルとして、順序パターンを安定的にモデル化できない可能性があると説明できる。さらに、CFSAはSASRec+よりも性能が良く、FDSAはSASRec++を凌駕している。これは、アイテムレベルのシーケンスと特徴レベルのシーケンスにそれぞれ分離した自己注意ブロックを適用し、アイテムの遷移パターンと特徴の遷移パターン(すなわち、CFSAとFDSA)を捉えることが、アイテム表現とその特徴表現を自己注意メカニズムの入力としてスプライシングする(すなわち、SASRec+とSASRec++)より効果的であることを示している。以上の実験から、項目レベルと特徴レベルの独立した2つのシーケンスを通して、項目と特徴の遷移パターンをモデル化することが、逐次推薦に有用かつ有意義であることが示された。最後に、データセットや評価指標に関わらず、提案するFDSAは最も良い性能を達成する。我々の縮退したモデルCFSAは、ほとんどのベースライン手法に一貫して勝っている。これは、自己アテンションネットワークによる独立したカテゴリレベルのシーケンスのモデリングの有効性を示している。また、FDSAはCFSAよりも性能が良く、特徴量レベルのシーケンスにおいてより多くの特徴量をモデル化することの有効性を示している。

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

表 3: 2 つのデータセットにおける、l_s と l_f を NDCG@10 で変化させた FDSA と CFSA の性能。

4.5 Influence of Hyper-parameters

埋め込みサイズd、アイテムベース自己注意ブロックのヘッド数l_s、特徴ベース自己注意ブロックのヘッド数l_fなどのハイパーパラメータの影響を調査した。

スペースの関係上、NDCG@10の実験結果のみを示す。Hit@10指標についても同様の実験結果が得られている。

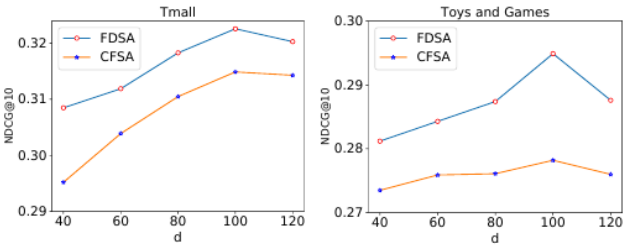

図2:dの違いによるFDSAとCFSAの性能比較

埋め込みサイズdの影響 図2は、2つのデータセットにおいて、埋め込みサイズdを変えた場合の本モデルの性能を示している。図2からわかるように、高次元はより多くの情報をモデル化することができるが、次元が100を超えると、FDSAとCFSAの性能は低下する。これは、モデルの暗黙の因子次元が高すぎる場合に、オーバーフィッティングが発生する可能性があることを示している。

ヘッド数 l_s と l_f の影響. 2つのデータセットにおいて、l_s と l_f を変化させた場合の本モデルの性能を調べる実験を行った。表3はNDCG@10の観点からの実験結果である。Tmallデータセットでは、CFSAとFDSAが$l_s = 4$, $l_f = 2$と最高の性能を達成し、ToysとGamesデータセットでは$l_s = 2$, $l_f = 4$と最高の結果を得ていることが観察されます。これは、Tmallデータセットではこれらのアイテムの特徴の単一データ型が、特徴間の関係をモデル化するためにあまり複雑な構造を必要としないのに対し、各項目が説明テキストとタイトルを含むため、特徴間の遷移関係を捉えるために我々のモデルはより多くのヘッドを必要とするためと思われます。

5 Conclusion

本論文では、逐次推薦のためのFeature-level Deeper Self-Attention Network (FDSA)と名付けた新しい手法を提案する。FDSAはアイテムベースの自己注意ブロックを通してアイテム間の遷移パターンをモデル化し、また特徴ベースの自己注意ブロックによって特徴間の遷移パターンを学習する。そして、この2つのブロックの出力を完全連結層に統合し、次のアイテムを予測する。広範な実験結果により、我々のモデルは最先端のベースライン手法を凌駕することが示された。

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.

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.

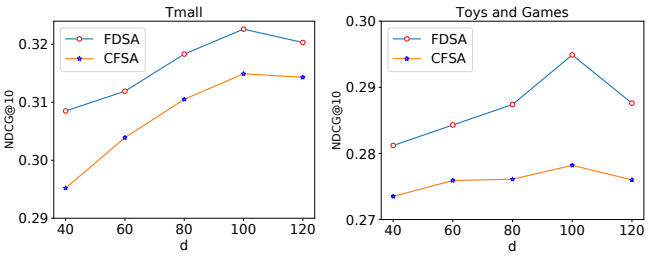

Figure 2: The performance of FDSA and CFSA under difference choices of d.

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. より深い自己注意による文字レベルの言語モデリング。AAAI, 2019 にて。

[Chen et al., 2017] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: 画像キャプションのための畳み込みネットワークにおける空間的およびチャネル的な注意。In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6298-6306. IEEE, 2017.

[Chen et al., 2018] Kehai Chen, Rui Wang, Utiyama, Eiichiro Sumita, and Tiejun Zhao. ニューラル機械翻訳のための構文指向の注意。30-Second AAAI Conference on Artificial Intelligence, 2018 にて。

[Chung et al., 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. シーケンスモデリングに関するゲート型リカレントニューラルネットワークの経験的評価。CoRR, abs/1412.3555, 2014.

[Gehring et al., 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin.

[He et al., 2017] Ruining He, Wang-Cheng Kang, and Julian McAuley. 翻訳ベースの推薦。第11回ACM推薦システム会議論文集, ページ161-169. ACM, 2017.

[He et al., 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: 推薦のための神経気配りアイテム類似度モデル. 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. リカレントニューラルネットワークによるセッションベースのレコメンデーション。ICLR, 2016 にて。

[Huang et al., 2018] Xiaowen Huang, Shengsheng Qian, Quan Fang, Jitao Sang, and Changsheng Xu. Csan: ユーザー逐次推薦のための文脈的自己注意ネットワーク. In 2018 ACM Multimedia Conference on Multimedia Conference, pages 447-455. ACM, 2018.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. 自己注意型逐次推薦。2018 IEEE International Conference on Data Mining (ICDM), page 197-206 にて。IEEE, 2018.

[Li et al., 2019] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: ビデオ質問応答のための共注意を伴う位置的自己注意。30third AAAI Conference on Artificial Intelligence, 2019 にて。

[また、このような場合にも、「逐次刊行物」を利用することができる。Bpr: 暗黙のフィードバックからのベイズ型パーソナライズドランキング。人工知能における不確実性に関する第25回会議論文集, ページ452-461. AUAI Press, 2009.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenタラー、ラース・シュミット-ティーム. 次バスケット推薦のためのパーソナライズド・マルコフ連鎖の因数分解. 第19回世界的なウェブに関する国際会議の議事録、ページ811-820。ACM, 2010.

[Shen *et al.*, 2018] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: 方向性自己注意ネットワークによるrnn/cnn-自由言語理解. 第30回AAAI人工知能会議, 2018にて。

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Person-畳み込みシーケンス埋め込みによる上位n個の逐次推薦のalized。第11回ACM国際ウェブ検索・データマイニング会議論文集, ページ565-573. ACM, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Nikiパーマー、ヤコブ・ウズコレイト、リオン・ジョーンズ、アダン・N・ゴメス、Łukaszカイザー、イリア・ポロスチン。注意力があればよい。神経情報処理システムの進歩、ページ5998-6008、2017年。

[Yuan *et al.*, 2019] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 次の項目推薦のための単純な畳み込み生成ネットワーク。ACM, 2019.

[Zhao *et al.*, 2019] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Zhixu Li, Jiajie Xu, and Victor S Sheng. 次に行くべき場所。次のポイ推薦のための時空間lstmモデル。AAAI, 2019 にて。

[Zhou *et al.*, 2018] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. Atrank: 推薦のための注意に基づくユーザ行動モデリングフレームワーク. 30-Second AAAI Conference on Artificial Intelligence, 2018 にて。

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.