Information Processing and Management 59 (2022) 102814

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Combining Non-sampling and Self-attention for Sequential Recommendation

Guangjin Chen [a], Guoshuai Zhao [a,*], Li Zhu [a], Zhimin Zhuo [b], Xueming Qian [c,d]

[a] School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China
[b] Beijing Institute of Electronic System Engineering, 100854, Beijing, China
[c] School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China
[d] Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, Xi'an, 710049, Shaanxi, China

ARTICLE INFO

Keywords:
Non-sampling mechanism
Self-attention
Sequential recommendation
User preference modeling

ABSTRACT

With the rapid development of social media and big data technology, user's sequence behavior information can be well recorded and preserved on different media platforms. It is crucial to model the user preference through mining their sequential behaviors. The goal of sequential recommendation is to predict what a user may interact with in the next moment based on the user's historical record of interactive sequence. However, existing sequential recommendation methods generally adopt a negative sampling mechanism (e.g. random and uniform sampling) for the pairwise learning, which brings the defect of insufficient training to the model, and decrease the evaluation performance of the entire model. Therefore, we propose a Non-sampling Self-attentive Sequential Recommendation (**NSSR**) model that combines non-sampling mechanism and self-attention mechanism. Under the premise of ensuring the efficient training of the model, NSSR model takes all pairs in the training set as training samples, so as to achieve the goal of fully training the model. Specifically, we take the interactive sequence as the current user representation, and propose a new loss function to implement the non-sampling training mechanism. Finally, the state-of-the-art result is achieved on three public datasets, Movielens-1M, Amazon Beauty and Foursquare_TKY, and the recommendation performance increase by about 29.3%, 25.7% and 42.1% respectively.

1. Introduction

In the context of the explosion of social media data, recommendation systems have played an irreplaceable role in alleviating network information overload (Fang, Zhang, Shu, & Guo, 2020; He et al., 2017; Xu, 2018). As a branch of the recommendation system, sequential recommendation is to predict the user's behavior at the next moment based on the user's historical interaction behavior records. For example, in e-commerce shopping, users are likely to buy a mobile phone case after purchasing a mobile phone. Because it is based on the implicit feedback data of the user's historical interaction behavior to model user preferences, and this implicit feedback data is easy to obtain, so sequential recommendation plays a very important role on many social media platforms, e.g., e-commerce products recommendation (He, & McAuley, 2016b; Kang, & McAuley, 2018; Rendle, Freudenthaler, & Schmidt-Thieme, 2010), app recommendation (Cao et al., 2017), music recommendation (Cheng, & Shen, 2016; Cheng, Shen, Zhu, Kankanhalli, & Nie, 2017) and next POI recommendation (Bao, Zheng, Wilkie, & Mokbel, 2015; Cheng, & Shen, 2014; Cui, Shen, Nie, Hong, & Ma, 2017; Feng et al., 2018; Jiang, Qian, Mei, & Fu, 2016; Zhao et al., 2020, 2020), etc.

* Corresponding author.
E-mail address: guoshuai.zhao@xjtu.edu.cn (G. Zhao).

https://doi.org/10.1016/j.ipm.2021.102814
Received 25 May 2021; Received in revised form 2 October 2021; Accepted 4 November 2021
Available online 14 January 2022

Different from the score prediction task of traditional recommendation system (Yang, Hsu, Hua, & Cheng, 2019), sequential recommendation is essentially to mine the co-occurrence rules between items in the sequence. Therefore, previous researchers generally use matrix factorization and Markov chains algorithms to mine the transfer relationship (He, Kang and McAuley, 2017; He & McAuley, 2016b; Li, Lu, Cheema, Shou, & Chen, 2020a; Rendle et al., 2010), these methods can fully take into account the items that the user has recently interacted with. Recently, with the rapid development of deep learning, Hidasi et al. firstly used recurrent neural networks in session-based recommendation (Hidasi, Karatzoglou, Baltrunas, & Tikk, 2015); Liu, Q., Wu et al. used recurrent neural networks to model the spatio-temporal context attributes in POI recommendations to achieve better POI recommendation performance (Liu, Wu, Wang, & Tan, 2016; Qian, Feng, Zhao, & Mei, 2013; Shen, Wang, Yan, & Cui, 2013; Sun, Zhao, & Zhang, 2018; Wu, Li, Zhao, & Xueming, 2020); Zhao, P., Luo et al. proposed a spatio-temporal gated recurrent neural network to realize POI recommendations (Zhao, Lou, Qian and Hou, 2020; Zhao, Luo et al., 2020). Tang, J., & Wang, K. et al. modeled the transfer law between items in a sequence using convolutional neural networks (Li et al., 2020; Tang, & Wang, 2018). For the first time, Wu, S., Tang, Y. et al. used graph neural networks in session-based sequence recommendation (Wu et al., 2019). Xu, C., Zhao et al. used graph neural networks to model interactive sequences and the self-attention mechanism is added to the above, which can extract more abstract and accurate feature expressions to further improve the recommendation performance (Xu et al., 2019). Ma, C., Ma, L. et al.et al. took the user id embedding representation as a memory unit to model the user's long-term preference, in which the graph neural network models the user's short-term interest (Hao, Dun, Zhao, Wu, & Qian, 2021; Ma, Ma, Zhang, Sun, Liu et al., 2020a; Tang, Zhao, Bu and Qian, 2021). Recently, with the great success of the self-attention mechanism in machine translation, the self-attention mechanism has also been used in the field of sequential recommendation, such as Kang and McAuley (2018), Li, Wang and McAuley (2020), Wang, and Han (2021), Liu, Zhang, and Gulla (2020) and Ren et al. (2020). And these methods have achieved good results in sequential recommendation.

However, these sequential recommendation methods mentioned above have a common shortcoming. The model training methods are all based on a negative sampling mechanism, that is, some negative samples are randomly selected from all non-positive instances, then build the training pairs with the positive samples to realize pairwise learning. The advantage of negative sampling mechanism is that the model training speed is fast and the effect is good. But random negative sampling cannot consider all the negative samples in the dataset, so it is not conducive to the full training of the model (Wang, Hu, Wang, Cao, Sheng et al., 2019). If the model samples all non-positive instances as negative samples, it will bring about time-consuming and inefficient model training (Hidasi et al., 2015).

In order to solve the above challenges, we propose the NSSR model, which combines non-sampling mechanism and self-attention mechanism. Specifically, based on the self-attention mechanism, our NSSR model takes the interactive sequence as the current user representation and uses the whole data to build training pairs to fully train the model, so as to improve the recommendation performance of the model. At the same time, we propose a new loss function to reduce the time complexity of the model, and compared to the traditional non-sampling training method, the training efficiency of our NSSR model is guaranteed. Finally, extensive empirical studies on three public datasets show our NSSR model is better than the existing sequence modeling baseline models.

The major contributions of this paper are summarized as follows:

(1) We propose a sequential recommendation method NSSR model that combines non-sampling and self-attention mechanisms for the first time. Our NSSR model uses all the training samples of the dataset during the training process, so as to achieve the purpose of sufficient training.

(2) We propose a new non-sampled loss function in our NSSR model, and solve the problem of high time complexity caused by traditional non-sampling training methods, and improve the efficiency of model training.

(3) Our NSSR model achieves the best performance results on the three public sequential recommendation datasets Movielens-1M, Amazon Beauty and Foursquare_TKY, which increased by approximately 29.3%, 25.7% and 42.1%, respectively.

The remainder of this paper is organized follows: In Section 2, we introduce the related work; Section 3 we describe the implementation of our NSSR model in detail; Section 4 we carry out experimental verification and result analysis; Finally, we give the conclusion and future outlook in Section 5.

2. Related works

In this section, we introduce latest research progress about the existing sequential recommendation methods, next POI recommendation methods and negative sampling mechanism.

2.1. Sequential recommendation

In the early days, many existing sequential recommendation methods generally mine the co-occurrence rules of the items in the sequence by modeling the item-item transfer matrix. For example, the Factorized Personality Markov Chain(FPMC) (Rendle et al., 2010) model uses matrix factorization and Markov chains to capture long-term and short-term preferences of users, respectively. In fact, since the last item of user interaction has a decisive effect on the current recommendation, the model based on the first-order Markov chain can better model the transfer relationship between items in the sequence (He, Kang et al., 2017), while the model based on the higher-order Markov chain considers the common influence of multiple items that the user has interacted with recently. In addition, some Markov chain models consider the effect of similarity between items (He & McAuley, 2016b). In recent years, many sequential recommendation methods based on deep learning have been proposed. For example, the GRU4Rec

model (Hidasi et al., 2015; Zhao, Liu, Chao and Qian, 2021) uses the natural sequence modeling advantages of the recurrent neural network to achieve good results in the session-based recommendation. The Caser model (Tang & Wang, 2018) is a model based on a convolutional neural network. It regards the representations of n items of the user's recent interaction as a picture, and uses convolution operations to model the transfer relationship between items in the sequence. Recently, SASRec model (Kang & McAuley, 2018) uses the self-attention mechanism to mine the influence of the user's previous interactive items on the current sequence, and has achieved the best sequential recommendation performance.

2.2. Next POI recommendation

Most of the early research on POI recommendation used collaborative filtering algorithms to characterize users' interest preferences, such as matrix factorization techniques incorporating various contextual information (Cui et al., 2017; Gao, Tang, Hu, & Liu, 2015; Jiang, Qian, Shen, Fu, & Mei, 2015; Lian, Zhao, Xie, Sun, Chen et al., 2014; Liu, Pham, Cong, & Yuan, 2017; Yao, Fu, Liu, Liu, & Xiong, 2016). However, these models only model static user preferences and cannot capture user dynamic interest changes. Recently, due to the successful application of deep learning in the field of recommendation systems, a large number of POI recommendation methods based on neural networks have also been proposed (Ma et al., 2020b; Yang, Bai, Zhang, Yuan, & Han, 2017; Yin, Wang, Wang, Chen, & Zhou, 2017).

The task of next POI recommendation is to use the user's previous historical check-in data to predict the next point of interest. Cheng, Yang, Lyu, and King (2013) proposed a method of using matrix factorization to realize personalized embedded Markov Chain. Inspired by the application of RNN in sequential recommendation (Ding, Quan, Yao, Li, & Jin, 2020), RNN-based POI recommendation methods have also been proposed (Li, Shen, & Zhu, 2018; Manotumruksa, Macdonald, & Ounis, 2017). For example, the ST-RNN model (Liu et al., 2016) extends RNN to local spatio-temporal context for modeling. The CARA model (Manotumruksa, Macdonald, & Ounis, 2018) proposes a gated GRU unit, whose purpose is to capture the dynamic changes of users' points of interest. Both TMCA model (Li et al., 2018) and STGN model (Zhao, Luo et al., 2020) are based on LSTM by adding a gating mechanism to learn spatio-temporal information features. DeepMove model (Feng et al., 2018) designed a multi-modal type of RNN to learn the transfer relationship in the sequence. And ASSPA model (Zhao, Zhang et al., 2020) takes into account the sub-sequence features to better model the POI sequence pattern. Therefore, next POI recommendation is essentially a sequential recommendation that incorporates contextual information.

2.3. Negative sampling mechanism

In sequential recommendation or next POI recommendation, the model training data is modeled based on the implicit feedback data of the interaction between the user and the item, such as purchases, or check-ins. However, because implicit feedback data often lack negative feedback samples. In order to overcome this difficulty, previous research mainly used the following two strategies: one is a negative sampling mechanism (Chen, Yeh, & Ma, 2021; Kang & McAuley, 2018; Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2012; Tang, Zhao, Wu and Qian, 2021; Zhao, Song, Xie, He, & Zhuang, 2015); the other strategy is based on the whole data training (Hu, Koren, & Volinsky, 2008; Liang, Charlin, McInerney, & Blei, 2016).

The negative sampling strategy (Kang & McAuley, 2018; Ma et al., 2020a; Rendle et al., 2012; Tang & Wang, 2018) refers to sampling negative instances from the data that the user has not interacted with. For example, The Bayesian Personalized Ranking(BPR) ranking model can essentially be regarded as an improvement of negative sampling (Rendle et al., 2012). The core idea is to randomly select negative instance entries from all training samples, and maximize the observed distance between positive and negative instances during model training. Since the number of negative samples is limited, the time complexity of model training will not be very high, and the training cost of the total model is acceptable (Ding et al., 2020), but the disadvantage is that each sampling cannot cover all training samples in the training iteration, which will make the model training insufficient and reduce the convergence speed of the model, so the model performance also highly depends on the design of the sampler (Ding et al., 2020; He, Zhang, Kan and Chua, 2016).

The whole data training strategy is to use all the training data for training. For example, the Weighted Matrix Factorization(WMF) model (Hu et al., 2008) assigns the missing values in the user–item score matrix to a label of 0, and then uses a point-to-point regression mechanism to give a lower sample weight. Although the whole data training strategy has a higher coverage of negative instance modeling, and the model training will be more adequate. But the disadvantage is that each iteration of training needs to calculate all samples, thus the learning method will be very slow.

3. Our method

3.1. Problem formulation

Fig. 1 shows the architecture of our proposed model NSSR. First, we input the user historical interaction sequence into the embedding layer to obtain the embedding representation of each item; Then, use the self-attention network to model the transfer relationship between the items in the sequence and get the sequence representation of each time; Finally, at each time step t, the model predicts the next item based on the sequence representation at the time t through the prediction layer. Specifically, we provide the user $u \in U$ with an interactive sequence $S_u = (S_1^u, S_2^u, \dots, S_{|S^u|}^u)$, where $|S^u|$ is real length of the sequence. The input of our NSSR model is $(S_1^u, S_2^u, \dots, S_{|S^u|-1}^u)$, and the output ground truth at each time is the input at the next time, denoted as $(S_2^u, S_3^u, \dots, S_{|S^u|}^u)$. In the following sections, we describe how to build our NSSR model through the embedding layer, self-attention network, prediction layer and the non-sampling training mechanism. Table 1 shows the mathematical symbols and their definitions used in this article.

Table 1
Notation and description.

Notation	Description
U, I	User and item set
Z	All users' interaction sequences set
S_B	Batch of users' interaction sequences
S^u	The user u's historical interaction sequence
d	Item embedding size
L	Maximum length of model input sequence
a	Number of stacked self-attention modules
h	Parameters of prediction layer
E	Embedding matrix of item representation
P	Embedding matrix of position representation
Y	Output of the embedding layer
$\hat{R}$	Prediction score set of all candidate items
O^a	Sequence representation after the a-th self-attention module
F^a	Sequence representation after the a-th feed-forward network

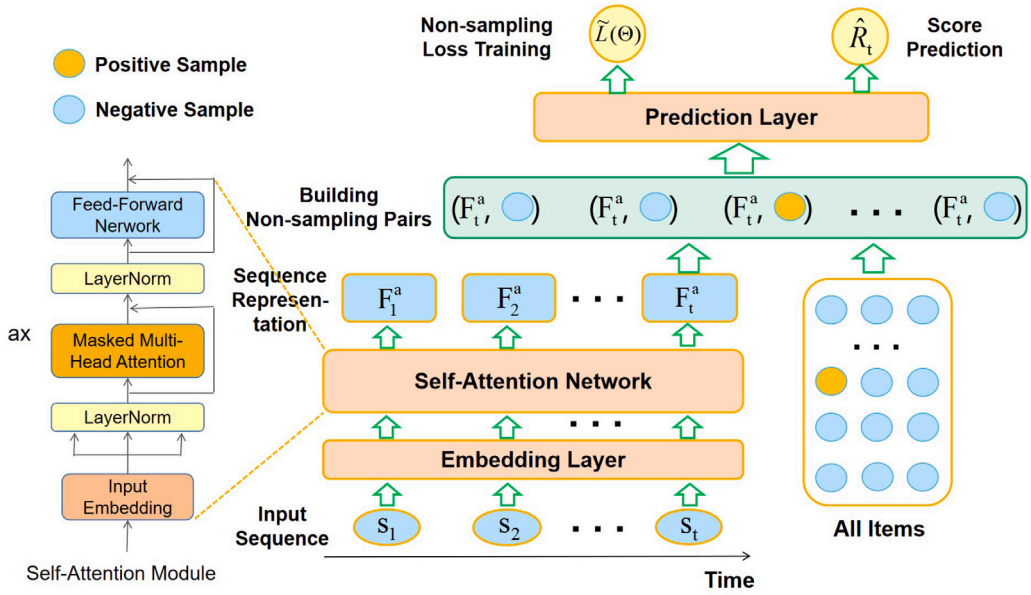


Fig. 1. The overview architecture of our proposed NSSR model.

3.2. Embedding layer

Given a user u's historical interaction sequence $(S_1^u, S_2^u, \ldots, S_{|S^u|-1}^u)$, we first convert it into a fixed-length sequence $s = (s_1, s_2, \ldots, s_L)$, where $L \in \mathbb{R}$ represents the maximum length that the model can handle. If the actual length of sequence is longer than L, then the nearest L items the user recently interacted with are considered. If the actual length of the sequence is shorter than L, we fill in the left side of the sequence until the length of the sequence is L, where $\mathbf{0}$ is used as the padding items. Suppose the item embedding matrix is $E \in \mathbb{R}^{|I| \times d}$, where $|I| \in \mathbb{R}$ is the number of all items, $d \in \mathbb{R}$ is the embedding dimension. Since the order of the items in the interaction sequence plays a very important role in the recommendation performance, and the self-attention mechanism does not inherently have the ability to model the position of item in the sequence. Therefore, we add the learnable positional embedding $P \in \mathbb{R}^{L \times d}$, so we have:

$$Y = \begin{bmatrix} E_1 + P_1 \\ E_2 + P_2 \\ \cdots \\ E_L + P_L \end{bmatrix} \tag{1}$$

where $Y \in \mathbb{R}^{L \times d}$ is the output of the embedding layer, $E_i \in \mathbb{R}^{1 \times d}$, $P_i \in \mathbb{R}^{1 \times d}$ are the item i's embedding representation and position representation in the sequence, respectively. In the experiment, we also tried to use the fixed position embedding method (Vaswani et al., 2017), but the final recommendation performance is not as good as the learnable position embedding.

3.3. Self-attention network

As shown in Fig. 1, the self-attention network is composed of ath self-attention modules. Each self-attention module consists of a masked multi-head attention layer and a forward–forward network. Below we will describe in detail the implementation and function of each part of the self-attention network.

3.3.1. Masked multi-head attention layer

The multi-head attention mechanism was first used in neural machine translation (Vaswani et al., 2017) to model the semantics of the entire sentence and achieved great success. The following formulas give the calculation function of masked multi-head attention mechanism:

$$\mathrm{MMH}(Q, K, V) = Concat(head_1, head_2, \ldots, head_{h_0}) \tag{2}$$

$$head_i = Masked\,Attention(QW_i^Q, KW_i^K, VW_i^V), i \in \{1, 2, \ldots, h_0\} \tag{3}$$

$$Masked\,Attention(\hat{Q}, \hat{K}, \hat{V}) = \left(\mathrm{softmax}\left(\frac{\hat{Q}\hat{K}^T}{\sqrt{d_0}}M\right)\right)\hat{V} \tag{4}$$

where $d_0 = d/h_0$, $d \in \mathbb{R}$ is the item embedding size, $h_0 \in \mathbb{R}$ is the number of heads. Q or $\hat{Q}$ represents the query vector, K or $\hat{K}$ represents the key vector, and V or $\hat{V}$ represents the value vector, M is the mask matrix to blind the future interaction information in the current interaction sequence. $W_i^Q \in \mathbb{R}^{d\times d_0}$, $W_i^K \in \mathbb{R}^{d\times d_0}$, and $W_i^V \in \mathbb{R}^{d\times d_0}$ represent three parameter matrices to be learned, which are used to realize linear mapping. The masked multi-head attention mechanism first split original embedding space, and then the attention mechanism is operated on each segmented embedding space, finally concatenate them. Among them, the attention weight calculation formula (4) calculates the correlation weight between the query vector $\hat{Q}$ and the key vector $\hat{K}$, and then multiplies it with the value $\hat{V}$ to get the final weight sum. That is, the original value $\hat{V}$ is given different weights in different dimensions of the feature space to obtain better feature representation. And $\sqrt{d_0}$ is used for scaling to prevent the inner product value from being too large.

In our NSSR model, we take the output $Y \in \mathbb{R}^{L\times d}$ of the embedding layer as the input of the masked multi-head attention layer, the specific calculation is as follows:

$$O = \mathrm{MMH}\left(LayerNorm(Y), Y, Y\right) + LayerNorm(Y) \tag{5}$$

$$\mathrm{LayerNorm}\ (x) = \alpha \odot \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{6}$$

where $O \in \mathbb{R}^{L\times d}$ is the output of masked multi-head attention layer. We only perform layer normalization operations on the query vector. And we also perform residual connection on the output of the masked multi-head attention layer, because residual connections can effectively make up for the serious problem of information loss caused by deep neural network, and at the same time can effectively use the features of different levels (He, Zhang, Ren and Sun, 2016).

The calculation formula of the layer normalization operation is formula (6), where $\odot$ represents the product between two matrix elements, μ, σ are the mean and standard deviation of the input x respectively, and α, β are the scaling factor and bias to be learned. It can aggregate too large or too small eigenvalues in the non-linear interval of the activation function to prevent gradient explosion or gradient disappearance (Ba, Kiros, & Hinton, 2016).

3.3.2. Feed-forward network

As shown in Fig. 1, the self-attention module mainly consists of two parts, one is the masked multi-head attention layer, and another is the feed-forward network.

The masked multi-head attention mechanism can take into account the features of the user's historical interactive items through weighting, but it is always a linear mapping. In order to give the model nonlinear modeling capabilities and fully consider the impact of interactions between features of different dimensions, we connect two fully-connected feed-forward layers:

$$F = FFN\,(O) = (Relu\left(LayerNorm(O)W_1 + b_1\right))W_2 + b_2 + LayerNorm(O) \tag{7}$$

where $O \in \mathbb{R}^{L\times d}$, $F \in \mathbb{R}^{L\times d}$ are the output of masked multi-head attention layer and the output of the feed-forward network, $W_1, W_2 \in \mathbb{R}^{d\times d}$, and $b_1, b_2 \in \mathbb{R}^{1\times d}$ are model parameters to be learned. Similarly, in order to prevent the phenomenon of gradient disappearance or gradient explosion in the process of model training, we perform the layer normalization operation on the input of the feed-forward network, and at the same time perform residual connection on its output.

3.3.3. Multiple self-attention modules

Considering the limited feature extraction ability of a single self-attention module, we stack the self-attention module, and then define the ath module as:

$$O^a = SAM\left(F^{a-1}\right) \tag{8}$$

$$F^a = FFN\left(O^a\right) \tag{9}$$

where $O^a \in \mathbb{R}^{L\times d}$ is the ath output of masked multi-head attention layer, $F^a \in \mathbb{R}^{L\times d}$ is the output of ath feed-forward network. And we use the $O \in \mathbb{R}^{L\times d}$ to initialize the input of the first masked multi-head attention layer.

However, when we stack more self-attention modules, the amount of parameters to be learned will increase, which may lead to overfitting during model training. Inspired by the paper (Vaswani et al., 2017), we adopted dropout operations in the self-attention modules to realize model regularization.

3.4. Prediction layer

Given the first t items $(s_1, s_2, \ldots, s_t)$, after a self-attention modules, we predict the next item based on the current sequence representation $F_t^a \in \mathbb{R}^{1\times d}$. Specifically, we use the matrix decomposition layer to predict the probability that the next item is the ith item:

$$\hat{R}_{t,i} = h(F_t^a \odot E_i)^T \tag{10}$$

where $\hat{R}_{t,i} \in \mathbb{R}$ represents prediction score of candidate item $i \in I$ at time $t \in \{1, 2, \ldots, L\}$, $E_i \in \mathbb{R}^{1\times d}$ is the item i's embedding representation, and $h \in \mathbb{R}^{1\times d}$ is a randomly initialized vector and need to be learned during model training. Therefore, we can generate the final recommendation list by sorting the scores $\hat{R} \in \mathbb{R}^{1\times|I|}$ of all candidate items. In our experiment, in order to prevent overfitting due to too many model parameters, we used the method of sharing the item embedding parameter.

3.5. Non-sampling mechanism training

Because most sequential recommendation models are based on negative sampling strategies (e.g. random uniform sampling), sequential recommendation methods have inherent shortcomings in sampling design. Therefore, we propose the non-sampling training mechanism for the sequential recommendation, which overcomes the shortcomings of inadequate training caused by the negative sampling mechanism, and redefine a new loss function to reduce the time complexity of the traditional non-sampling method. The following will detail the non-sampling training mechanism in our NSSR model:

In the user interaction implicit feedback data of sequential recommendation, the interaction data of user u and item i is defined as follows:

$$R_{u,i} = \begin{cases} 1, & \text{if } user\ u\ interacts\ with\ item\ i \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Generally, traditional non-sampling learning uses the loss function of weighted regression:

$$L(\Theta) = \sum_{u\in U}\sum_{i\in I} c_{ui}\left(R_{u,i} - \hat{R}_{u,i}\right)^2 \tag{12}$$

$$\hat{R}_{\mathrm{u,i}} = p_u q_i^T \tag{13}$$

where $U \in \mathbb{R}$ is the set of all users, $I \in \mathbb{R}$ is the set of all items, $c_{ui} \in \mathbb{R}$ is the weight of a sample, and $p_u \in \mathbb{R}^{1\times d}$ is the user u's representation, $q_i \in \mathbb{R}^{1\times d}$ the item i's representation. So we can get the time complexity of the loss function (12) as $O(|U||I|d)$, d is the item embedding size, which is too high to train the neural network-based model. Inspired by the literature (Chen, Zhang, Zhang, Liu, & Ma, 2020), we have the following theorem:

Theorem 1. *For a prediction function is the following generalized matrix factorization model,*

$$\hat{R}_{\mathrm{u,i}} = h^T\left(p_u \otimes q_i\right) \tag{14}$$

where $p_u \in \mathbb{R}^{d\times 1}$, $q_i \in \mathbb{R}^{d\times 1}$ is user representation and item representation, $h \in \mathbb{R}^{d\times 1}$ is the parameter of prediction layer, and $\otimes$ refers to vector dot product operations. The gradient of the non-sampling loss function (12) is equivalent to the following formula:

$$\widetilde{L}(\Theta) = const + \sum_{u\in U_B}\sum_{i\in I} c_{ui}^{I-}\hat{R}_{u,i}^2 + \sum_{u\in U_B}\sum_{i\in I^+}\left(\left(c_{ui}^{I+} - c_{ui}^{I-}\right)\hat{R}_{u,i}^2 - 2c_{ui}^{I+}\hat{R}_{u,i}\right) \tag{15}$$

where $const \in \mathbb{R}$ is a constant symbol, $U_B \in \mathbb{R}$ is a batch of users, I^+ is set of positive samples, $c_{ui}^{I-} \in \mathbb{R}$ is the weight of negative samples, c_{ui}^{I+} is the weight of positive samples.

Since the prediction layer of our NSSR model is also a matrix factorization model, and we take the sequence feature F_t^a obtained by the user interaction sequence through the self-attention network as the current user representation. So we have:

$$p_u = (F_t^a)^T \tag{16}$$

$$q_i = (E_i)^T \tag{17}$$

Finally, combined with formulas (10) (15) (16) and (17), the new non-sampling training loss function of our NSSR model is equivalent to:

$$\widetilde{L}(\Theta) = \sum_{k=1}^{d}\sum_{l=1}^{d}\left(\left(\sum_{t\in S_B} F_{t,k}^a F_{t,l}^a\right)\left(\sum_{i\in I} c^{I-} E_{i,k} E_{i,l}\right)(h_k h_l)\right) + \sum_{t\in S_B}\sum_{i\in I^+}\left(\left(1-c^{I-}\right)\hat{R}_{t,i}^2 - 2\hat{R}_{t,i}\right) + \lambda\|\Theta\|^2 \tag{18}$$

where $S_B \in \mathbb{R}$ is a batch of interaction sequences, $c^{I-} \in \mathbb{R}$ is the average weight of negative samples, $F_t^a \in \mathbb{R}^{1\times d}$ is sequence representation at current t, $E_i \in \mathbb{R}^{1\times d}$ is the embedding representation of the item i, $h \in \mathbb{R}^{1\times d}$ are parameters to be learned in the prediction layer, k and l represent the index of the embedding dimension, λ is the regularization coefficient, Θ is all parameters to be learned in our NSSR model. Therefore, the non-sampling loss (18) updates Θ through back propagation until convergence. In this way, the theoretical time complexity of our non-sampling mechanism is reduced to $O\left((|B|+|I|)\mathrm{d}^2 + \left|I^+\right| d\right)$, $|I^+|$ is the number of positive feedback samples. Because the number of positive feedback samples is generally much smaller than $|S_B||I|$, the time complexity of our NSSR model is acceptable in real recommendation scenarios.

4. Experimental results and discussion

4.1. Datasets

We experienced our NSSR model on three public datasets. The three datasets are:

(1) **Movielens-1M**: MovieLens dataset was originally an official dataset for movie classification, and now is widely used in the research of sequential recommendation methods. In our experiment, the version we choose is MovieLens-1M, a movie dataset containing one million ratings.

(2) **Amazon Beauty**: The Amazon product dataset was first published by Professor Julian McAuley in Ref. He, and McAuley (2016a). This dataset covers the behavior records of users purchasing goods from May 1996 to July 2014, including various information such as user reviews and purchase timestamps for products. In our experiment, we select the "Beauty" category dataset, because the Amazon Beauty dataset has high sparsity and diversity.

(3) **Foursquare_TKY**: The foursquare dataset (Zhao et al., 2020) contains the check-in data in New York and Tokyo from April 2012 to February 2013. Among them, New York has 227,428 check-in records, and Tokyo has 573,703 check-in records. Each check-in record includes user ID, point of interest ID, timestamp, GPS location and other semantic information. This data set is often used to study LBSN recommendation based on time and location. In our experiment, we choose the "Yokyo" dataset for sequential recommendation and named "Foursquare_TKY".

4.2. Compared methods

In the experiment, we used the following six mainstream sequential recommendation compared methods:

(1) **Pop**: This is a non-personalized recommendation model based on item popularity, and it always recommends the most popular item to the user every time;

(2) **BPR-MF** (Rendle et al., 2012): This method is a classic matrix factorization recommendation model. But in the sequential recommendation, the difference from the traditional matrix factorization model is that it regards the sequence of user interaction items as the current user representation, and uses the Bayesian ranking objective function to optimize;

(3) **FPMC** (Rendle et al., 2010): This method combines matrix factorization and first-order Markov chain to perform sequential recommendation. The advantage is that it can capture the user's short-term preferences and model the transition relationship from item to item;

(4) **GRU4Rec** (Hidasi et al., 2015): This method uses GRU units to build a recurrent neural network, which was originally used for session-based user click sequence modeling. In this experiment, we use embedding vectors instead of one-hot vectors to represent items;

(5) **Caser** (Tang & Wang, 2018): This method is based on convolutional neural network, which is recommended by applying horizontal and vertical convolution operations, so as to capture the transfer relationship of high-order Markov chains in the sequence;

(6) **SASRec** (Kang & McAuley, 2018): This method is based on self-attention, which only uses a multi-head attention mechanism to model the transfer relationship between items in the sequence, thereby implements the function of recommending the next item. It performances well for sequence modeling currently.

4.3. Experimental details

BPR-MF, FPMC, GRU4Rec, Caser, and SASRec are all implemented based on the source code provided by the corresponding authors. The input items embedding dimension of all models is fixed at 50. For Movielens-1M, the maximum length is 200, and dropout ratio is 0.2; For Amazon Beauty, the maximum length is 50, and dropout ratio is 0.5; For Foursquare_TKY, the maximum length is 100, and dropout ratio is 0.2. The settings of other hyper-parameters of the compared methods are consistent with those suggested by the author of the original paper. During the preprocessing of three datasets, we excluded user interaction sequences whose sequence length is less than 5, and also removed items that appear less than 5 times. The training dataset construction method is to eliminate the last two items of each original sequence in the data, and then for the current moment, the next item in the sequence is used as a positive sample, and the other items in the training dataset are used as a negative sample. For the division of the verification dataset and the test dataset, the idea of leaving one is adopted (Zhao, Mu et al., 2020). In NSSR model, for the above three datasets, we set the average weight c_{ui}^{I-} of negative samples to 0.001.

4.4. Evaluation metrics

In our experiment, we use hit rate (Hit) and normalized distributed cumulative gain (NDCG) to evaluate the performance of sequential recommendation. Hit is a recall-based index used to measure whether the target item is in the top-k position of the recommended list; NDCG is sensitive to the position of the target item, the top-ranked item will get higher score. Therefore, in sequential recommendation, we need to rank the predicted scores of all candidate items for each sequence z. It is defined as $R_z = r_1, r_2, \ldots, r_{|I|}$, where $|I|$ is the total number of candidate items, and r_i is the model predicted score ranking of the candidate item i in the sequence z. Suppose that all users' interaction sequence set is Z, the ground truth of the current sequence z's next interaction is the item $t \in [1, |I|]$. Then the calculation formulas of these two indicators are as follows:

$$Hit@K = \frac{1}{|Z|}\sum_{z}^{|Z|} f\left(\mathrm{r_t} < K\right) \tag{19}$$

$$NDCG@\ \mathrm{K} = \frac{1}{|Z|}\sum_{z}^{|Z|} \frac{2^{f(\mathrm{r_t}<K)} - 1}{\log_2\left(\mathrm{r_t} + 2\right)} \tag{20}$$

where $f(x)$ is an indicator function, when x is true, the function value is 1, otherwise it is 0. r_t is the predicted score ranking of the item t.

4.5. Recommendation performance

As shown in Tables 3–5, the best performing result in each column is in bold, and the second-best performing result in each column is underlined. We can easily see that our NSSR model achieves the best recommendation performance in the three public datasets. And the improvement of our NSSR model relative to the best baseline is shown in the last row. Moreover, we can get the following conclusions:

(1) The non-sampling training mechanism of our NSSR model allows the embedding representation of all items to be fully trained, so it obtains a higher recommendation performance, and compared with the baseline SASRec model, the recommended indicators are improved by about 29.3%, 25.7% and 42.1% on the Movielens-1M, Amazon Beauty and Foursquare_TKY dataset;

(2) We can also find that the traditional FPMC model recommendation performance is better than some models based on neural network, especially in the sparse dataset Amazon Beauty and Foursquare_TKY. We think that because the sparser the dataset is, the more important the items recently interacted with by the user are, so the FPMC model based on a simple first-order Markov chain shows greater advantages.

(3) At the same time, we discuss the relationship between the sparsity of the dataset and the recommendation performance for our NSSR model, as shown in Table 6. We can see that for the NSSR model, the longer the average sequence length of the training samples of the dataset, the higher the density of the dataset, and the higher Hit@100 and NDCG@100 of model recommendation performance. Other baseline models have similar rules. The reason is that the higher the density of the dataset, the longer the interaction sequence, the better the user's true preferences can be portrayed, and the recommendation performance will be more accurate.

4.6. Influence of non-sampling mechanism

As shown in Fig. 2, in order to illustrate the superiority training speed and effectiveness of our NSSR model's non-sampling mechanism, we compared three deep learning models with different network architectures. We can see that under the premise of the same model training time, the recommendation performance index Hit@100 of our proposed NSSR model significantly better than the other three baseline models, which further illustrates the non-sampling training mechanism is helpful to fully train model, so as to improve the recommendation performance of our NSSR model.

Moreover, in terms of model convergence speed, the convergence speed of the NSSR model is 2.2 s/epoch, which is significantly better than that of 30.4 s/epoch of GRU4Rec model based on the recurrent neural network and 32.3 s/epoch of Caser model based on the convolutional neural network. However, the convergence speed of our NSSR model is a little slower than that of 1.7 s/epoch of

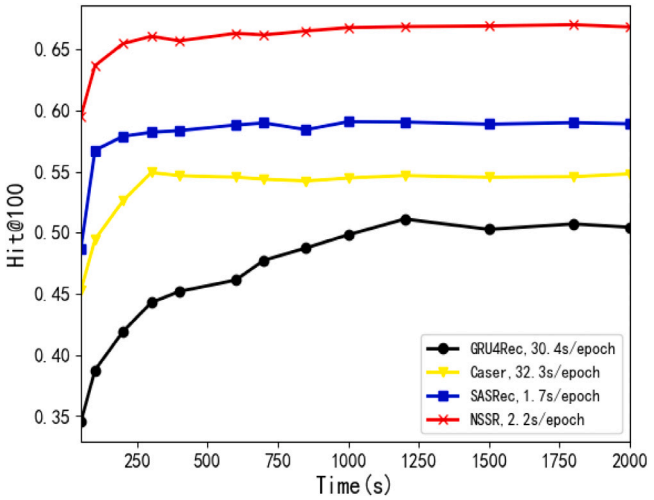
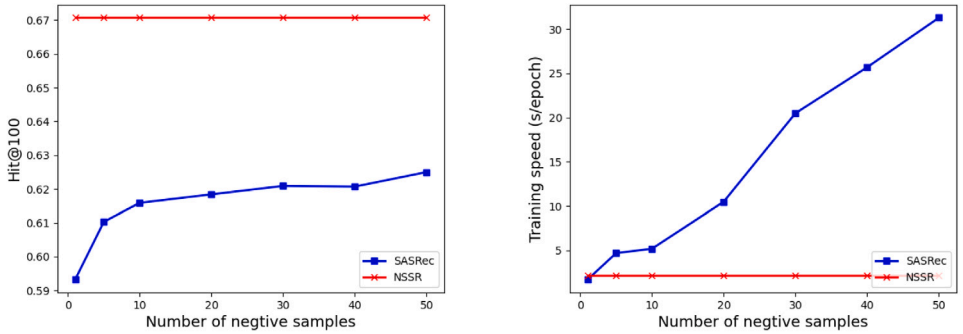


Fig. 2. Line graph of Hit@100 change with model training time (Movielens-1M).



(a) The relationship between Hit@100 indicator and number of negative samples.

(b) The relationship between model training speed and number of negative samples.

Fig. 3. The impact of the number of negative samples on model recommendation performance and training speed.

SASRec model, but it is worthwhile to spend a little more time training in exchange for a significant improvement in recommendation performance.

In order to further explore the influence of the number of negative samples on the sequential recommendation methods, we conduct a comparison experiment of recommendation index Hit@100 and the training speed with the number of negative samples. From Fig. 3, we can see that as the number of negative samples increases, the recommended index Hit@100 of the baseline model SASRec will increase, but the training speed will also decrease sharply. This is because the number of negative samples increases, the more fully the model is trained, but at the same time the calculation amount of model prediction also increases, resulting in a decrease in the speed of model training. For our NSSR model, the advantages of non-sampling mechanism are fully utilized and better recommendation performance is achieved. At the same time, we redefine the loss function so that our NSSR model will not significantly increase the time cost of non-sampling training.

4.7. Influence of different components in self-attention networks

In order to fully understand our NSSR model, we conducted ablation experiments on different components of the self-attention network, as shown in Table 7:

Through the above experimental results table, we can see that the position embedding, residual linking, layer normalization and the number of layers in the self-attention network will have a greater impact on the final recommendation performance, among which the residual connection module is most influential, this also shows that the residual connection is an important component of feature enhancement. We have obtained through experiments that it is more appropriate to set the number of self-attention layers to 2. If the number of layers is too few, the capability of feature extraction is insufficient. If there are too many layers, too many parameters to be learned will easily lead to over-fitting.

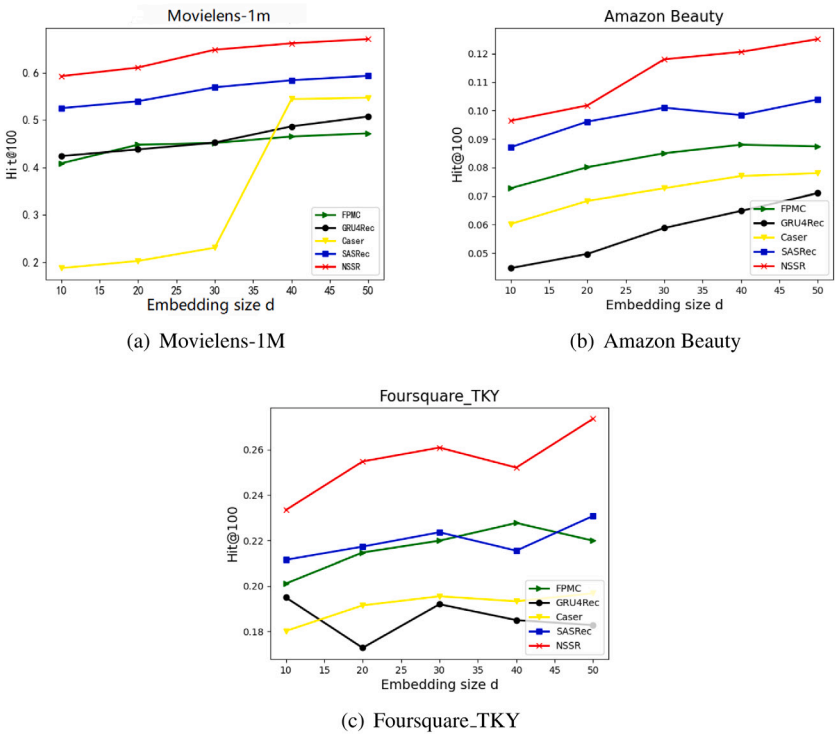


Fig. 4. Line graph of Hit@100 change with items embedding dimension d. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.8. Influence of hyper-parameters

4.8.1. Impact of the negative sample average weight c^{I-}

As can be seen from Table 8, when the average negative sample weight c^{I-} in the loss function (18) is too high or too low, the model recommendation performance is not optimal. When the value of the average negative sample weight c^{I-} is less than 0.001, the recommendation performance of our NSSR model in three datasets is better than the baseline model SASRec. Overall, when the average negative sample weight is small, the model recommendation performance is better. This also shows that the impact of a single negative sample training is relatively small in the sequential recommendation, so all sampling mechanism is good. In our experiment, the value of hyperparameter c^{I-} is unified to 0.001.

4.8.2. Impact of items embedding dimension d.

As shown in Fig. 4, we explore the relationship between the item embedding dimension and evaluation metric Hit@100 for our NSSR model. It can be seen from the figure: (1) As the item embedding dimension increases, the Hit@100 value is also improved, and shows a certain positive correlation. Especially for the Caser model, the item embedding dimension has a great influence on the final recommendation performance in the Movielens-1M. We think that because the Caser model itself is modeled on the embedded representation of items in the interactive sequence as "images", the reduction of item embedding dimensions will directly reduce the interaction of item features in the same sequence, so the performance will decrease accordingly.

(2) The Hit@100 indicator size of our NSSR model is negatively correlated with the sparsity of the dataset. We think this is because our NSSR model predicts probabilistically for all items, and the number of items in sparse datasets tends to be larger, so that the noise predicted by the model increases, resulting in a decrease in recommended performance indicators. And in sparse datasets, such as Amazon Beauty and Foursquare_TKY, the FPMC method based on Markov chain performs better than the deep learning methods GRU4Rec and Caser. This also shows that the Markov chain-based model is suitable for sequence modeling of sparse datasets.

(3) We also see that our NSSR model is significantly higher than the best baseline SASRec model in all item embedding dimensions, and further reflects the effectiveness of non-sampling mechanism. As the embedding dimension increases, the recommendation performance of the NSSR model steadily improves, which reflects the robustness of our NSSR model.

4.8.3. Impact of maximum sequence length L

It can be seen from Table 9 that the setting of the model input interaction sequence's maximum length L has a direct effect on the recommendation performance of our NSSR model. It can be seen from Table 2 that the average sequence length of the Movielens-1M

Table 2
Datasets statistics(after preprocessing).

Dataset	Movielens-1M	Amazon Beauty	Foursquare_TKY
Number of users	6040	52 024	2292
Number of items	3416	57 289	7057
Average length of sequences	163.5	5.63	54.09
Density	4.840%	0.013%	0.795%

Table 3
Recommendation performance on Movielens-1M.

Movielens-1M	Hit@50	Hit@100	NDCG@50	NDCG@100
Pop	0.1021	0.1747	0.0255	0.0372
BPR-MF	0.2262	0.3314	0.0595	0.0765
FPMC	0.3430	0.4719	0.1012	0.1221
GRU4Rec	0.3639	0.5073	0.1081	0.1313
Caser	0.4065	0.5455	0.1217	0.1442
SASRec	0.4455	0.5932	0.1275	0.1515
NSSR	**0.5474**	**0.6707**	**0.1858**	**0.2059**
Improvement.	22.87%	13.06%	45.70%	35.91%

Table 4
Recommendation performance on Amazon Beauty.

Amazon Beauty	Hit@50	Hit@100	NDCG@50	NDCG@100
Pop	0.0295	0.0483	0.0080	0.0110
BPR-MF	0.0487	0.0672	0.0127	0.0157
FPMC	0.0657	0.0874	0.0210	0.0245
GRU4Rec	0.0492	0.0710	0.0135	0.0182
Caser	0.0508	0.0780	0.0147	0.0191
SASRec	0.0699	0.1039	0.0202	0.0257
NSSR	**0.0870**	**0.1251**	**0.0270**	**0.0332**
Improvement.	24.46%	20.40%	28.57%	29.18%

Table 5
Recommendation performance on Foursquare_TKY.

Foursquare_TKY	Hit@50	Hit@100	NDCG@50	NDCG@100
Pop	0.1188	0.1844	0.0323	0.0429
BPR-MF	0.0968	0.1429	0.0258	0.0333
FPMC	0.1545	0.2199	0.0426	0.0532
GRU4Rec	0.1134	0.1828	0.0297	0.0409
Caser	0.1291	0.1968	0.0408	0.0517
SASRec	0.1571	0.2308	0.0434	0.0553
NSSR	**0.2007**	**0.2736**	**0.0731**	**0.0849**
Improvement.	27.75%	18.54%	68.43%	53.53%

Table 6
The impact of the sparsity and average length of the dataset on the recommendation performance of our NSSR model.

Dataset	Movielens-1M	Amazon Beauty	Foursquare_TKY
Average length	163.50	54.09	5.63
Density	4.840%	0.7950%	0.0013%
Hit@100 of NSSR	0.6707	0.2736	0.1251
NDCG@100 of NSSR	0.2059	0.0849	0.0332

Table 7
Influence of different components in self-attention networks.

Hit@100	Movielens-1M	Amazon Beauty	Foursquare_TKY
Original	0.1021	0.1747	0.0255
No position embedding	0.2262	0.3314	0.0595
No residual connection	0.3430	0.4719	0.1012
No layer normalization	0.3639	0.5073	0.1081
Number of self-attention layer = 1(default 2)	0.4065	0.5455	0.1217

Table 8
The effect of setting the negative sample weight average c^{I-} on the recommendation indicator Hit@100 of NSSR model.

c^{I-}	Movielens-1M	Amazon Beauty	Foursquare_TKY
0.1	0.5808	0.0853	0.1684
0.01	0.6353	0.1066	0.1842
0.001	**0.6707**	**0.1251**	0.2491
0.0005	0.6651	0.1172	**0.2526**
0.0001	0.6263	0.1013	0.2367

Table 9
The effect of setting the maximum sequence length L on the recommendation effect of NSSR model(Movielens-1M).

n	Hit@50	Hit@100	NDCG@50	NDCG@100
10	0.4419	0.5604	0.1518	0.1710
50	0.5334	0.6546	0.1805	0.2002
100	0.5422	0.6656	0.1853	0.2053
150	0.5442	0.6671	0.1827	0.2027
200	**0.5474**	**0.6707**	**0.1858**	**0.2059**

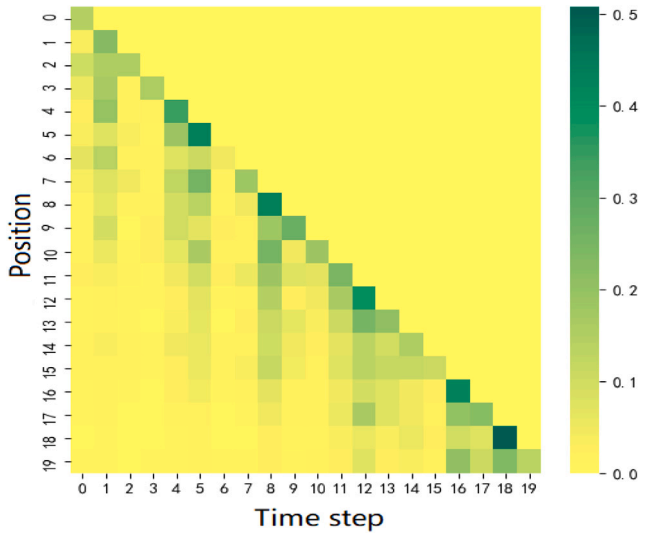


Fig. 5. Visualization diagram of item position weight in our NSSR model (Movielens-1M). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

dataset is 163.5, but when L is 100, the model recommendation performance has achieved good results, indicating that the user's recent interactive items have played a more important role in the sequential recommendation. At the same time, as the sequence length increases, the model recommendation performance continues to improve, reflecting the user's early historical action is also useful for sequence modeling. In the same way, for the other two datasets, after comparing the results of ablation experiments, we selected the appropriate maximum sequence lengths respectively. The maximum sequence length of the Amzon Beauty dataset is 50, and the maximum sequence length of the Foursquare_TKY dataset is 100.

4.9. Visualization experiment

In order to further understand the reason why our NSSR model can perform better in sequential recommendation, we explore the influence of the user's previous interactive item's position on the next item recommendation, we conducted a sequence item position weight visualization experiment on Movielens-1M dataset. We draw a heat map Fig. 5 by saving the weights of the 20 item positions that the user has recently interacted with. The vertical axis represents the position of the user's interactive item at the current moment, and the horizontal axis represents the position of the user's previous interaction item. The color represents the weight size between the two. We can see from the picture:

(1) The upper right corner is all yellow, indicating that the NSSR model does not use the user's future interactive item information, which corresponds to the mask mechanism of the masked multi-head attention layer in our NSSR model and is in line with the basic requirements of the sequential recommendation task;

(2) The green parts with large weights in Fig. 5 are concentrated near the diagonal, and most of the bottom left corner is yellow, which shows that the items recently interacted by the user play a more important role in our NSSR model;

(3) In addition to the diagonal lines, there are more green parts in other parts of Fig. 5, which shows that our NSSR model has the ability to capture long-term preferences of users, that is, the ability to model long sequence predictions.

5. Conclusion and future work

In this paper we propose a sequential recommendation method NSSR, which combines the advantages of both non-sampling training and self-attention mechanism for the first time. On the basis of self-attention mechanism modeling items and item transition relationship, we take the interaction item sequence as the current user representation, and we propose a new loss function to realize the non-sampling mechanism to ensure efficient training of our NSSR model. Finally our NSSR model obtains the current best recommended performance on three public datasets. In addition, we put forward two suggestions about sequential recommendation:

(1) In view of the limitations of the negative sampling mechanism in sequential recommendation, we proposed an improved method for non-sampling training for the first time, and we will do further exploration in difficult samples mining (Ding et al., 2020; Wang, Xu, He, Cao, Wang et al., 2020);

(2) For sequential recommendation, the user's dwell time on the item is very important. Although the researchers have recently proposed some improved methods of using time information (Li, Wang et al., 2020), there is few work that considers fine-grained time information, such as seasons and holidays.

CRediT authorship contribution statement

Guangjin Chen: Methodology, Software, Data curation, Writing – original draft, Formal analysis. **Guoshuai Zhao:** Conceptualization, Methodology, Formal analysis, Supervision. **Li Zhu:** Writing – review & editing. **Zhimin Zhuo:** Writing – review & editing. **Xueming Qian:** Resources, Supervision.

Acknowledgments

This work was supported in part by the NSFC, China under Grants 61902309, 61701391, and 61772407; in part by ShaanXi Provincial Natural Science Foundation under Grant 2018JM6092; in part by the Fundamental Research Funds for the Central Universities, China (xxj022019003); in part by China Postdoctoral Science Foundation under Grant 2020M683496; in part by the National Postdoctoral Innovative Talents Support Program, China (BX20190273); and in part by Humanities and Social Sciences Foundation of Ministry of Education, China (16XJAZH003).

References

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Bao, J., Zheng, Y., Wilkie, D., & Mokbel, M. (2015). Recommendations in location-based social networks: a survey. *GeoInformatica*, *19*(3), 525–565.

Cao, D., Nie, L., He, X., Wei, X., Shen, J., Wu, S., & Chua, T. S. (2017). Version-sensitive mobile app recommendation. *Information Sciences*, *381*, 161–175.

Chen, Yen-Liang, Yeh, Yi-Hsin, & Ma, Man-Rong (2021). A movie recommendation method based on users' positive and negative profiles. *Information Processing & Management*, *58*(3), Article 102531.

Chen, C., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2020). Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)*, *38*(2), 1–28.

Cheng, Z., & Shen, J. (2014). Just-for-me: an adaptive personalization system for location-aware social music recommendation. In *Proceedings of international conference on multimedia retrieval* (pp. 185–192).

Cheng, Z., & Shen, J. (2016). On effective location-aware music recommendation. *ACM Transactions on Information Systems (TOIS)*, *34*(2), 1–32.

Cheng, Z., Shen, J., Zhu, L., Kankanhalli, M. S., & Nie, L. (2017). Exploiting music play sequence for music recommendation. In *IJCAI, Vol. 17* (pp. 3654–3660).

Cheng, C., Yang, H., Lyu, M. R., & King, I. (2013). Where you like to go next: Successive point-of-interest recommendation. In *Twenty-third international joint conference on artificial intelligence*.

Cui, C., Shen, J., Nie, L., Hong, R., & Ma, J. (2017). Augmented collaborative filtering for sparseness reduction in personalized POI recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(5), 1–23.

Ding, J., Quan, Y., Yao, Q., Li, Y., & Jin, D. (2020). Simplify and robustify negative sampling for implicit collaborative filtering. arXiv preprint arXiv:2009.03376.

Fang, H., Zhang, D., Shu, Y., & Guo, G. (2020). Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, *39*(1), 1–42.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference* (pp. 1459–1468).

Gao, H., Tang, J., Hu, X., & Liu, H. (2015). Content-aware point of interest recommendation on location-based social networks. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 29*, No. 1.

Hao, J., Dun, Y., Zhao, G., Wu, Y., & Qian, X. (2021). Annular-graph attention model for personalized sequential recommendation. *IEEE Transactions on Multimedia*.

He, R., Kang, W. C., & McAuley, J. (2017). Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 161–169).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

He, R., & McAuley, J. (2016a). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international conference on world wide web* (pp. 507–517).

He, R., & McAuley, J. (2016b). Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)* (pp. 191–200). IEEE.

He, X., Zhang, H., Kan, M. Y., & Chua, T. S. (2016). Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 549–558).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 eighth IEEE international conference on data mining* (pp. 263–272). Ieee.

Jiang, S., Qian, X., Mei, T., & Fu, Y. (2016). Personalized travel sequence recommendation on multi-source big social media. *IEEE Transactions on Big Data*, *2*(1), 43–56.

Jiang, S., Qian, X., Shen, J., Fu, Y., & Mei, T. (2015). Author topic model-based collaborative filtering for personalized POI recommendations. *IEEE Transactions on Multimedia*, *17*(6), 907–918.

Kang, W. C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)* (pp. 197–206). IEEE.

Li, H., Lu, H., Cheema, M. A., Shou, L., & Chen, G. (2020). Indoor mobility semantics annotation using coupled conditional Markov networks. In *2020 IEEE 36th international conference on data engineering (ICDE)* (pp. 1441–1452). IEEE.

Li, R., Shen, Y., & Zhu, Y. (2018). Next point-of-interest recommendation with temporal and multi-level context attention. In *2018 IEEE international conference on data mining (ICDM)* (pp. 1110–1115). IEEE.

Li, J., Wang, Y., & McAuley, J. (2020). Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining* (pp. 322–330).

Li, S., Xie, G., Ren, J., Guo, L., Yang, Y., & Xu, X. (2020). Urban PM2. 5 concentration prediction via attention-based CNN–LSTM. *Applied Sciences*, *10*(6), 1953.

Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., & Rui, Y. (2014). GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 831–840).

Liang, D., Charlin, L., McInerney, J., & Blei, D. M. (2016). Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web* (pp. 951–961).

Liu, Y., Pham, T. A. N., Cong, G., & Yuan, Q. (2017). An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment*, *10*(10), 1010–1021.

Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 30*, No. 1.

Liu, Peng, Zhang, Lemei, & Gulla, Jon Atle (2020). Dynamic attention-based explainable recommendation with textual and visual fusion. *Information Processing & Management*, *57*(6), Article 102099.

Ma, C., Ma, L., Zhang, Y., Sun, J., Liu, X., & Coates, M. (2020). Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*, No. 04 (pp. 5045–5052).

Ma, Yaxue (2020). Location recommendation by combining geographical, categorical, and social preferences with location popularity. *Information Processing & Management*, *57*(4), Article 102251.

Manotumruksa, J., Macdonald, C., & Ounis, I. (2017). A deep recurrent collaborative filtering framework for venue recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 1429–1438).

Manotumruksa, J., Macdonald, C., & Ounis, I. (2018). A contextual attention recurrent architecture for context-aware venue recommendation. In *The 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 555–564).

Qian, X., Feng, H., Zhao, G., & Mei, T. (2013). Personalized recommendation combining user interest and social circle. *IEEE Transactions on Knowledge and Data Engineering*, *26*(7), 1763–1777.

Ren, R., Liu, Z., Li, Y., Zhao, W. X., Wang, H., Ding, B., & Wen, J. R. (2020). Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 89–98).

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web* (pp. 811–820).

Shen, J., Wang, M., Yan, S., & Cui, P. (2013). Multimedia recommendation: technology and techniques. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (pp. 1131–1131).

Sun, Y., Zhao, P., & Zhang, H. (2018). Ta4rec: Recurrent neural networks with time attention factors for session-based recommendations. In *2018 international joint conference on neural networks (IJCNN)* (pp. 1–7). IEEE.

Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 565–573).

Tang, H., Zhao, G., Bu, X., & Qian, X. (2021). Dynamic evolution of multi-graph based collaborative filtering for recommendation systems. *Knowledge-Based Systems*, *228*, Article 107251.

Tang, H., Zhao, G., Wu, Y., & Qian, X. (2021). Multisample based contrastive loss for top-k recommendation. *IEEE Transactions on Multimedia*, http://dx.doi.org/10.1109/TMM.2021.3126146.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez …, A. N., & Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

Wang, Yang, & Han, Lixin (2021). Adaptive time series prediction and recommendation. *Information Processing & Management*, *58*(3), Article 102494.

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. (2019). Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830.

Wang, X., Xu, Y., He, X., Cao, Y., Wang, M., & Chua, T. S. (2020). Reinforced negative sampling over knowledge graph for recommendation. In *Proceedings of the web conference 2020* (pp. 99–109).

Wu, Y., Li, K., Zhao, G., & Xueming, Q. I. A. N. (2020). Personalized long-and short-term preference learning for next POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33*, No. 01 (pp. 346–353).

Xu, C. (2018). A novel recommendation method based on social network using matrix factorization technique. *Information Processing & Management*, *54*(3), 463–474.

Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang ., F., & Zhou, X. (2019). Graph contextualized self-attention network for session-based recommendation. In *IJCAI, Vol. 19* (pp. 3940–3946).

Yang, C., Bai, L., Zhang, C., Yuan, Q., & Han, J. (2017). Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1245–1254).

Yang, C. L., Hsu, S. C., Hua, K. L., & Cheng, W. H. (2019). Fuzzy personalized scoring model for recommendation system. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 1577–1581). IEEE.

Yao, Z., Fu, Y., Liu, B., Liu, Y., & Xiong, H. (2016). POI recommendation: A temporal matching between POI popularity and user regularity. In *2016 IEEE 16th international conference on data mining (ICDM)* (pp. 549–558). IEEE.

Yin, H., Wang, W., Wang, H., Chen, L., & Zhou, X. (2017). Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *29*(11), 2537–2551.

Zhao, G., Liu, Z., Chao, Y., & Qian, X. (2021). CAPER: Context-aware personalized emoji recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *33*(9), 3160–3172.

Zhao, G., Lou, P., Qian, X., & Hou, X. (2020). Personalized location recommendation by fusing sentimental and spatial context. *Knowledge-Based Systems*, *196*, Article 105849.

Zhao, P., Luo, A., Liu, Y., Zhuang, F., Xu, J., Li ., Z., & Zhou, X. (2020). Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Zhao, W. X., Mu, S., Hou, Y., Lin, Z., Li, K., Chen ., Y., & Wen, J. R. (2020). RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. arXiv preprint arXiv:2011.01731.

Zhao, Z., Song, R., Xie, X., He, X., & Zhuang, Y. (2015). Mobile query recommendation via tensor function learning. In *Twenty-fourth international joint conference on artificial intelligence* (pp. 4084–4090).

Zhao, K., Zhang, Y., Yin, H., Wang, J., Zheng, K., Zhou, X., & Xing, C. (2020). Discovering subsequence patterns for next POI recommendation. In *Proceedings of the Twenty-Ninth international joint conference on artificial intelligence* (pp. 3216–3222).