Sequential Recommendation Based on Long-Term and Short-Term User Behavior with Self-attention

Xing Wei[1,2], Xianglin Zuo[1,2], and Bo Yang[1,2(✉)]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
247334537@qq.com , ybo@jlu.edu.cn

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering Attached to the Ministry of Education, Jilin University, Changchun 130012, China

Abstract. Product recommenders based on users' interests are becoming increasingly essential in e-commerce. With the continuous development of the recommendation system, the available information is further enriched. In the case, user's click or purchase behavior could be a visual representation of his or her interest. Due to the rapid update of products, users' interests are not static, but change over time. In order to cope with the users' interest changes, we propose a desirable work on the basis of representative recommendation algorithm. The sequence of user interaction behavior is thoroughly utilized, and the items that users interact at different times have different significance for the reflect of users' interests. By considering the user's sequential behaviors, this paper focuses on the recent ones to obtains the real interest of user. In this process, user behavior is divided into long-term and short-term, modeled by LSTM and Attention-based model respectively for user's next click recommendation. We refer this model as LANCR and analyze the model in experiment. The experiment demonstrates that the proposed model has superior improvement compared with standard approaches. We deploy our model on two real datasets to verify the superior performance made in predicting user preferences.

Keywords: Recommender system · Long short-term memory · Sequential recommendation

1 Introduction

Recommendation system plays an essential role in current shopping websites and e-commerce. Network platform hopes to recommend products that may be of interest to users. Nowadays, users have exploded numbers of social network, and the information generated by users has also proliferated. To effectively solve this problem, various personalized recommenders have been made to provide

C. Douligeris et al. (Eds.): KSEM 2019, LNAI 11775, pp. 72–83, 2019.
https://doi.org/10.1007/978-3-030-29551-6_7

users with high quality items. Among the traditional personalized recommendation algorithms, the collaborative filtering algorithm [5] is undoubtedly successful. The collaborative filtering recommendation algorithm is based on similarity preference between users, but it cannot effectively recommend when facing with user's dynamically changing interest characteristics. Traditional recommendation system can be mainly divided into two methods: analyzing user's social relationship or calculating the similarity of goods. There are also algorithms that take both into account. These classical algorithms do open up the scope of recommendation and relief the problem of cold start to a certain extent. However, with the increase of user amount and the renewal of commodities, both user's social relationship and his preference to goods will change with time, which should be taken into consideration.

Recommender algorithm based on user behavior is of great importance in personalized recommendation system. Before utilizing user behavior data to design the algorithm, researchers first need to analyze the user behavior data to understand the general rules contained in the data, so that they could have the design of recommender model. Recommendation system obtains the user's interest through their behavior. Network information categories and new topics increase day by day. How to grasp the change of user is the key to improve the accuracy of recommendation results.

In the personalized recommendation system, acquiring users' interests is fundamental. In large data-intensive networks, user's preferences can be described by his or her historical behavior [4]. Initial research often extracts features from user's actions manually and aggregates them to get his attributes. However, manual extraction may not fully represent the feature itself. After extracting features by algorithmic techniques, most of current technologies deal with each user's historical behavior separately, but they do not reflect the information that users share among different behaviors [6]. In real life, the historical behavior of users is often related with others. It is proposed that the customer's purchase behavior may change with time [1], and the user's click doings in the previous and current period can be extracted separately. In other words, the predicted results to users will be influenced by both long-term and short-term interests of users. In recent studies, when dealing with top-N recommendation problems [3], it is found that the user interaction order plays an imperative role in describing his interest. The concept of perceiving user sequence intention is also proposed, and is commendably applied in next item recommendation [9].

Some of classical algorithms have been applied when facing with the problem that users' interests change along with time. Markov chain method can predict the next action of user by describing the user's latest behaviors. With the development of deep learning, RNN and LSTM algorithms [13] have been used to receive sequential input [2]. In addition to NLP applications, they can also be used for sequence recommendation. The attention mechanism was proposed in the process of neuro-machine translation (NMT) while using the encoder-decoder structure [17]. At present, attention machine is common in deep learning, which is not limited to encoder-decoder hierarchy. The algorithm of learning natural language from images is introduced [16], and the experimental results are improved by adding attention mechanism. The most advanced sentence representation model has been significantly improved by applying the same mechanism to a single sentence [15]. Attention mechanism connects two different modules by learning the weight between data.

In this paper, a new temporal recommendation model is proposed for user's sequential click actions. Our main contributions are as follows:

– The user's interest is divided into long-term and short-term, which enhances the interpretability of products recommended to users.
– A sequence recommendation is proposed, which masters the potential links between projects, and graspes the overall preferences of users, so that it focuses on both the user's global interests, and his recent actions.
– We completed our experiments on CiaoDVD and Amazon-Books datasets. The experimental results demonstrate the performance of the model.

Section 2 introduces the related algorithms and work. Section 3 describes the details of our model and gives the loss function. In Sect. 4, we have a comparative experiment to analyze the experimental results on real data sets. Finally, Sect. 5 summarizes the full text.

2 Related Works

2.1 Next Item Recommendation

In traditional recommendation system, collaborative filtering based on matrix factorization outputs the similarity matrix of users or commodities by studying the relationship matrix obtained from the user's behavior log. The recommendation algorithm based on naive bayesian classification is relatively simple to implement and has high accuracy, but it is suitable for classification problems with small amount of data and fewer categories.

The goal of sequential recommendation system could be broadly expressed as a linear combination of the users' long-term and short-term preferences [7]. Compared with a large number of previous models, which focus on users' long-term interests, session-based recommendation system is combined with GNN [8], emphasizing the importance of modeling short-term user preferences. RNNs for session-based recommendations [21] evaluated several improvements on basic RNN, and multiaspect experiments demonstrates that accounting for user's temporal shift behavior is possible to improve the performance of session-based recommendation system. In mobile application recommendation, user's historical data can also provide conspicuous assistance [23]. Combined with user's sequence of applications operated recently, the historical application can provide high accuracy in prediction scheme. User-based RNN is a new framework extended on RNN, which models user behavior, captures the dependencies between user events to full advantage, and shows the development of the original architecture in the field of recommendation [24].

Sequential recommendation predicts consecutive items according to user's historical behavior and the key work is to capture interactions within users and commodities. Markov chain predicts the next-time action by observing users' recent behaviors. The probability distribution of the next state is only determined by the current state, but the actions before this in the time series is irrelevant. MC [20] algorithm belongs to the property of no memory. In the process of improving recommender algorithm. There was a classical algorithm, which combines MC with matrix factorization for next item recommendation. In sequential recommendation field, recurrent neural network and its two most widely used variants, LSTM and GRU, are more commonly applied at present.

In online stores, recommending proper product is the core issue of recommendation system, accurately recommending the next product to users means attracting more customers and promoting the sales of products efficiently. According the market requirement, [10] proposes a long-term demand perception model, which considers that repeat purchase action represents the long-term needs of users, while complementary purchase means short-term needs. Through the analysis of customers' behaviors, the existing methods mainly use short-term feedback, but do not fully consider users' long-term stable preferences [11], or simply analyze users' long-term habits, but do not pay attention to the changes of users' favorite things over time.

The key point of our work is dividing the user's behaviors into long-term and short-term when considering their temporality, and we take different deep learning methods to model items. After joint training, the system eventually recommends the next item.

2.2 Long-Term and Short-Term Interests

Long short term memory network (LSTM) is a particular type of recurrent neural network, which learns long-term dependence information for a specific work. When facing with sequential problems, LSTM has obtained correct success and has been proverbially spreaded. It avoids completely long-term dependency problems through deliberate design. In practice, keeping the long-term information in memory is the core behavior of LSTM, rather than consuming a large amount of manpower. In standard RNN, this replication module has only one simple structure, such as a *tanh* layer. LSTM has the same mode, but the repetitive modules have a different frame. They have four layers of nervous system, coordinating and assisting each others to determine the output results. The specific principles will be introduced later.

The analysis and calculation of users' short-term interests is pretty to be accomplished by attention mechanism. Attention-based model is actually a measure of similarity. The more similar to the target state, the greater the weight of the current input is, indicating that the output depends more on the current input. Attention model has been originally applied in the field of image [14]. Subsequently, it was introduced into the field of NLP and applied to machine translation, and its main function is to learn the relationship between words. On this basis, the concept of soft attention model was introduced, which means each item is required, then a probability distribution of attention is calculated and weighted. Attention mechanism can be used not only to process the encoder or the hidden layer in front of it, but also to obtain the distribution of other features without additional information.

Self-attention mechanism is a special case, which focuses on itself and extract relevant information from its own project. Self-attention is also often referred as intra attention and has been widely used in machine learning. It refines the representation by matching a single sequence with itself. Attention-based also takes an advantage in learning context embedding. According to the items observed in the current transaction, they are weighted by different correlations to output a suitable next item [26]. In the case of limited background knowledge, self-attention can preserve the context sequence information and capture the relationship between elements. It has been proved that the model combining self-attention with sequential series can be superior to the state-of-the-art technologies in both sparse and dense datasets [25].

In our work, LSTM aims at users' long-term interests, whereas self-attention mechanism corresponding the dependency relationship of the users' short-term sequential behavior.

3 The Proposed Model

We now propose a sequential recommender system that combines LSTM and self-attention for user next click recommendation, named LANCR. The structure of this section is as follows: first, code and classify each commodity, the method selected in which is one-hot code. Next, introduce the technology proposed in detail and give the model diagram. Finally, we present the recommender results and the metrics arised in the process of building the model.

3.1 Embedding Layer

For each project, its own attributes such as movie categories, directors, actors, as well as abstracts, authors, and fields of the paper, can be together invested in describing the product itself. In machine learning application tasks, non-contiguous data is often encoded by numbers. The above attributes do not have a mathematically continuous relationship between each other, but in another aspect, they are considered as a mathematical order relationship. One-hot coding, also known as one-bit effective coding, using an *N-bit* status register to encode N states, each state being independent of its register bits, and at any time, only one of them is valid. One-hot encodes a value corresponding to the discrete feature to a point in Euclidean space. For example, there is a discrete form representing a movie, and the input entry is [category = action, director = Joe Russell, actor = Robert Downey&Chris Evans], which will be mapped to European space and the result of coding is as follows:

$$[0, 0, 1, 0, ..., 0][0, 1, 0, 0, ..., 0][0, 1, 0, 1, ..., 0] \tag{1}$$

Such three binary parts correspond to the three features of the original data as a vector form, which can be linked to the neural network. Given a sequence of user's click behaviors $V = \{V_1, V_2, \ldots, V_n\}$, where V_i represents the i^{th} item that the user clicked and n is the total number of user click sequences. After one-hot coding mentioned above, the result of item embedding layer is as follows:

$$\mathbf{e} = [e_1, e_2, \ldots, e_n] \tag{2}$$

e_i is the embedding result of V_i, which is later incorporated into the model as input of new time.

3.2 Sequence Recommendation with Self-attention

The input of model is user's click behavior sequence, and after encoding, we get the sequential embedding $\mathbf{e}$. LSTM system receives the embedding of input ordered by time, and then calculates the output of hidden layer through forget gate, input gate and output gate. In this process, the cell state is updated and the calculation results are transmitted to the next cycle. The LSTM part of our proposed model is shown in Fig. 1, where shows a structure of one time step. The new cell state and output gate determine the output of the new hidden layer.

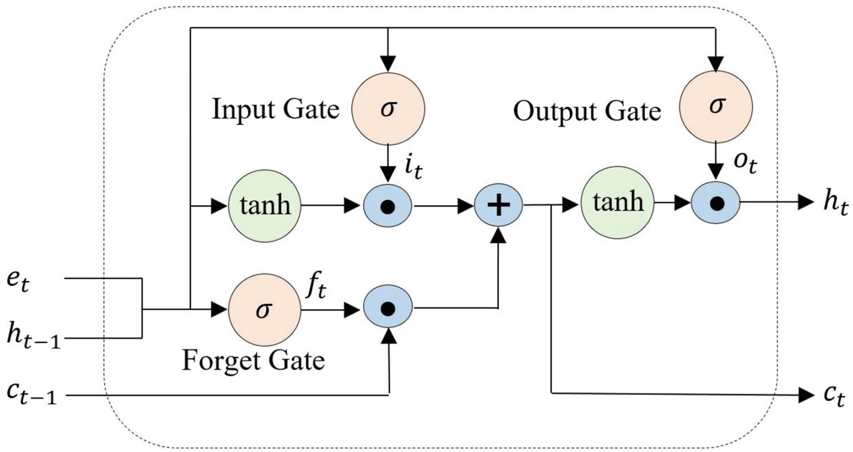

Fig. 1. The LSTM part of model [13].

The first step of LSTM is to select the information to be discarded in the sequence, which is determined by the forget gate. The gate receives the hidden layer content of the previous moment h_{t-1} and the new input vector e_t, getting a value between 0 and 1. If the value is 1, it means complete reservation while 0 means abandonment. After discarding elements, input gate determines which new information should be stored and updated. Then, a *tanh* layer creates a new candidate value vector, which is added to the state. The old cell state will be updated. The previous steps have decided what will be done, and now actually going to complete. The model then discards the information that needs to be abandoned. In the end, the LSTM system computes the output value. This result is based on cell state. The model is conveyed as follows [13]:

$$i_t = \sigma(W_{ei}e_t + W_{hi}h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{ef}e_t + W_{hf}h_{t-1} + b_f) \tag{4}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W_{ec}e_t + W_{hc}h_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma(W_{eo}e_o + W_{ho}h_{t-1} + b_o) \tag{6}$$

$$h_t = o_t \cdot tanh(c_t) \tag{7}$$

where σ is the sigmoid function, e_t is the input item at time t, f_t expresses the forgetting vector, i_t is the input gate vector, which means the retention of new information, c_t is the updated cell state, o_t represents the output gate vector, h_t is the hidden layer state at new time, and it is provided to next part.

Next, we will calculate the latest hidden layer outputs. Through the self-attention mechanism, the hidden output is obtained as $h = \{h_1, h_2, \ldots, h_n\}$. For several recent outputs of the LSTM, $\{h_n - step + 1, h_n - step + 2, \ldots, h_n\}$ is used as the memory and input to calculate the attention output r_i, where *step* is the step size we choose to model the user's short-term interest (e.g. 5, 10), n is the length of items, and the calculation formulas are as follows [14]:

$$s_j^i = V^{'} relu(W_j h_j + W_i h_i) \tag{8}$$

$$a_j^i = \frac{exp(s_j^i)}{\sum_{j=1}^{n} exp(s_j^i)} \tag{9}$$

$$r_i = \sum_{j=1}^{step} a_j^i h_j \tag{10}$$

The formulas could be thought as calculating links between several recent commodities, r_i is the output result these items, a_j^i can be seen as a probability, reflecting the importance of h_j to r_i, s_j^i is the similarity operation of items, and $V^{'}$ is the combination of embedding $\mathbf{e}$, W_i is the parameter to be learned. The work process of self-attention mechanism in the model is to receive the hidden layer set of the previous part, each row represents a vector representation of an item, and the similarity calculation is performed between the rows and rows to obtain the weights. The weights and corresponding items are summed to get the results after analyzing the attention. We recommend the next item according to the sequence clicked by users. The model diagram is as Fig. 2.

After learning the long-term interest of the user, the latter projects are separately captured, so that the recent projects include not only the order relationship of the time, but also the interaction between each others. This part calculates the next item affected by the short-term actions. Its prediction result becomes Eq. (11):

$$\hat{y_i} = \frac{r_i}{step} \tag{11}$$

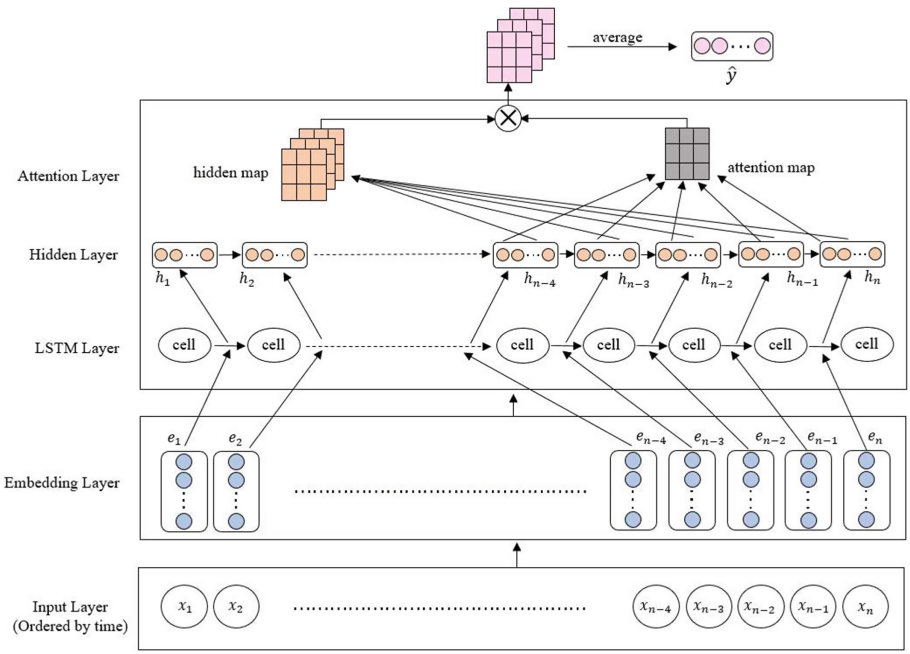

Fig. 2. The sequential recommender framework. This example takes the short-term step value of the user as 5.

$\hat{y}_i$ is the result predicted for user i. Let y_i be the next actual interactive item of the user i, and we predict the probability distribution of the next item by using the derived $\hat{y}_i$ as a softmax regression. It is the average value obtained by finding relations between recent projects. As shown in Fig. 2, when *step* is 5, all items embedding obtains the hidden layer through LSTM part, then calculates the weight of the $\{h_{n-4},\ h_{n-3},\ \ldots,\ h_n\}$, adds up the average, and takes the mean value as prediction result, instead of only removing the last output as the result. The definition of loss function is given as:

$$\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i^T log\sigma(w\hat{y}_i + b) \tag{12}$$

where y_i denotes the tag of the ith item, which is set to 1 if user clicks the result, while 0 means not, where w and b are the parameters of softmax regression and $\sigma = 1/(1 + exp(-x))$, to calculate the probability distribution. N is the total number of user's click items. We sum up losses to optimize our model. For each user, there is a loss function in the same format, setting the total number of users to M. The training process is to minimize the objective function as follow:

$$\mathscr{F} = \sum_{j=1}^{M} \mathscr{L}_j + \lambda \left\|\theta\right\| \tag{13}$$

where λ controls the regularization term and θ denotes the parameters in this model, including LSTM and self-attention mechanism.

4 Experiments

4.1 Experiment Setup

Datasets. In this section, we apply the proposed model to two real datasets. CiaoDVD is a DVD category dataset that was crawled from the website in December 2013. It includes the movie category, 72,655 users' clicks on the movie, and the click time. The cases for training and testing are movies clicked by users, and some attribute information is used in the experiment. We sorted the items in order and remove datas with too short sequence. Amazon-Books is a user purchase dataset with records and comments. Due to some loss of dataset, we randomly selected most of them for experiments, which contains 1,507,155 links between users and items. The screened test case is the goods purchased by the user within ten years. After processing the dataset, a sequence of items is obtained. The specific capacity of the datasets is shown in Table 1.

Table 1. Dataset capacity

Dataset	#Users	#Items	#User-Item actions
CiaoDVD	17,615	16,108	72,655
Amazon-Books	57,015	98,570	1,507,155

Evaluation Metrics. Two evaluation metrics are used in the experiments: Logloss (cross entropy) and AUC (Area Under ROC). A common application of AUC is an offline assessment of CTR. The evaluation of CTR plays an important role in the company's technical processes. The sigmoid function is used to relate the feature input to the probability. The probability of this output is the estimated value of the click rate.

Model Comparison. We compare this model with multiple recommendation algorithms. First, FM [18] is a classic algorithm for click rate prediction. And then BPR [19] is a personalized sorting algorithm for project prediction. At the same time, the matrix factorization recommendation algorithm based on Markov Chain [2] is compared. Finally, it is compared with the basic LSTM model. Besides paralleling with other models, we choose different *step* to find the best length of users' short-term action to the result.

4.2 Results and Analysis

The experimental results of LANCR and the comparative experiments are shown in Table 2, and we have selected several different *step* (the step size of user short-term length) and tested our models separately. The results of the test are in Table 3.

Accuracy of Proposed Model. Table 2 shows the comparison results of two baselines and the proposed method on two real datasets. From Table 2, it can be seen that LANCR is better than the basic regression model FM and bayesian probability model BPR. Among the next recommender problems, it also surpasses representative models. And the model with self-attention mechanism is more effective than the single LSTM.

Table 2. Comparative results

Dataset	Metric	FM	BPR	FPMC	LSTM	LANCR
CiaoDVD	AUC	0.7541	0.7634	0.7726	0.7943	**0.7951**
	LogLoss	0.3524	0.3351	0.327	0.3081	**0.2979**
Amazon-Books	AUC	0.8051	0.8262	0.8313	0.8526	**0.8623**
	LogLoss	0.298	0.2751	0.2538	0.2341	**0.2243**

Impact of Step Length. Table 3 shows the impact of the short-term *step*. We find that the appropriate step size can achieve higher accuracy, which means that the definition of user's short-term interest is not arbitrary. The experimental results show that the proper *step* is dependent on the datasets. For dataset Ciao, when the user's short-term step size is defined as 5, we get better results. In other words, focusing on the analysis of users' latest 5 clicks is more helpful for the next recommendation result. However, we can see that there are different results on Amazon-book dataset. When $step = 10$, the best result is achieved. Compared with Ciao, Amazon-Book dataset has a larger capacity and a relatively large number of clicked items. It cannot be deemed absolutely that the selection of *step* is affected by sample size, but we will then validate this conclusion with more datasets than before.

Table 3. Result of different steps

Dataset	Model	Metric	Step=3	Step=5	Step=8	Step=10
CiaoDVD	LANCR	AUC	0.7182	**0.7951**	0.7763	0.7523
Amazon-Books	LANCR	AUC	0.8367	0.8521	0.8514	**0.8623**

The proposed model integrates extracted features in the training process with large datasets in the experimental process. Facing a long item sequence, we choose the information that should be selected, and discard the information which has lost reference value for a long time, and deploy attention mechanism to capture the potential links between recent items, instead of compressing all of them into a hidden layer, and the model can obtain more abundant information. In terms of the association within our model, self-attention receives the global association of the items. Since the user's long-ago interest may change over time, short-term item selection is more representative of his real interest. Through the joint modeling of user's interaction, more accurate recommendation results will be provided.

5 Conclusion and Future Work

In this paper, we propose LANCR, a sequential recommendation algorithm based on long-term and short-term interest of users. The model applies LSTM to filter users' long-term interests, and uses the self-attention mechanism to calculate the user's short-term intentions and finally predicts his next click. By conducting experiments on real datasets and comparing experimental results, we believe that taking user's sequential click items into consideration has a great impact on the recommendation results. There may be some links within short-term items, not just the time order. In addition, as time progresses, user's recent click behavior is more valuable on expressing user's true intentions. So it is necessary to focus on the relationship between recent click items. In the future work, in order to observe experimental results more accurately, we will build a multi-dimensional item embedding, considering more attributes of the projects, and in the case of datasets, it's optative to add the user's own attributes to get more accurate prediction, as well as its interpretability.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China under grants 61876069 and 61572226, and Jilin Province Key Scientific and Technological Research and Development project under grants 20180201067GX and 20180201044GX.

References

1. Liu, D.R., Lai, C.H., Lee, W.J.: A hybrid of sequential rules and collaborative filtering for product recommendation. Inf. Sci. **179**(20), 3505–3519 (2009)
2. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820. ACM (2010)
3. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 565–573. ACM (2018)
4. Zhou, C., et al.: ATRank: an attention-based user behavior modeling framework for recommendation. In: Proceeding of the 32nd AAAI Conference on Artificial Intelligence (2018)
5. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
6. Ni, Y., et al.: Perceive your users in depth: learning universal user representations from multiple e-commerce tasks. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 596–605. ACM (2018)
7. Villatel, K., Smirnova, E., Mary, J., et al.: Recurrent neural networks for long and short-term sequential recommendation. arXiv preprint arXiv:1807.09142 (2018)
8. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. arXiv preprint arXiv:1811.00855 (2018)
9. Zhang, S., Tay, Y., Yao, L., Sun, A.: Next item recommendation with self-attention. arXiv preprint arXiv:1808.06414 (2018)

10. Bai, T., Du, P., Zhao, W.X., Wen, J.R., Nie, J.Y.: A long-short demands-aware model for next-item recommendation. arXiv preprint arXiv:1903.00066 (2019)
11. Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., Chen, E.: Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1734–1743. ACM (2018)
12. Chung, J., Ahn, S., Bengio, Y.: Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:1609.01704 (2016)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
15. Wang, S., Zhang, J., Zong, C.: Learning sentence representation with guidance of human attention. arXiv preprint arXiv:1609.09189 (2016)
16. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 21–29 (2016)
17. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
18. Rendle, S.: Factorization machines. In: 2010 IEEE International Conference on Data Mining, pp. 995–1000. IEEE (2010)
19. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
20. Gilks, W.R., Richardson, S., Spiegelhalter, D.: Markov Chain Monte Carlo in Practice. Chapman and Hall/CRC, Boca Raton (1995)
21. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 17–22. ACM (2016)
22. Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 843–852. ACM (2018)
23. Pu, C., Wu, Z., Chen, H., Xu, K., Cao, J.: A sequential recommendation for mobile apps: what will user click next app?. In: 2018 IEEE International Conference on Web Services (ICWS), pp. 243–248. IEEE (2018)
24. Donkers, T., Loepp, B., Ziegler, J.: Sequential user-based recurrent neural network recommendations. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 152–160. ACM (2017)
25. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE (2018)
26. Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., Liu, W.: Attention-based transactional context embedding for next-item recommendation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)