Research Article

Recommendation Based on Users' Long-Term and Short-Term Interests with Attention

Qiaoqiao Tan and Fang'ai Liu

School of Information Science and Engineering, Shandong Normal University, Jinan 250300, China

Correspondence should be addressed to Fang'ai Liu; lfa@sdnu.edu.cn

Received 5 June 2019; Accepted 5 September 2019; Published 17 October 2019

Academic Editor: Bogdan Smolka

Copyright © 2019 Qiaoqiao Tan and Fang'ai Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recommendations based on user behavior sequences are becoming more and more common. Some studies consider user behavior sequences as interests directly, ignoring the mining and representation of implicit features. However, user behaviors contain a lot of information, such as consumption habits and dynamic preferences. In order to better locate user interests, this paper proposes a Bi-GRU neural network with attention to model user's long-term historical preferences and short-term consumption motivations. First, a Bi-GRU network is established to solve the long-term dependence problem in sequences, and attention mechanism is introduced to capture user interest changes related to the target item. Then, user's short-term interaction trajectory based on self-attention is modeled to distinguish the importance of each potential feature. Finally, combined with long-term and short-term interests, the next behavior is predicted. We conducted extensive experiments on Amazon and MovieLens datasets. The experimental results demonstrate that the proposed model outperforms current state-of-the-art models in Recall and NDCG indicators. Especially in MovieLens dataset, compared with other RNN-based models, our proposed model improved at least 2.32% at Recall@20, which verifies the effectiveness of modeling long-term and short-term interest of users, respectively.

1. Introduction

Using historical data to predict future behaviors is the cornerstone of many recommendation systems. Past behavior sequences usually reflect user's next behavior, which is both effective and intuitive [1]. Massive user behavior (such as browsing and clicking) records in network reflect user's rich consumption patterns, researchers have done a lot of work on this, and much work has already made clear decisions using this rich sequence patterns.

In fact, user's interaction behaviors form a sequence naturally over time, dynamically revealing user's long-term historical preferences and short-term consumer motivations as shown in Figure 1, which is a typical online shopping scene for Amazon. This user may be a fan of Marvel, and the user has browsed or purchased many cartoon mobile phone cases. If only current user behavior sequences are considered, similar phone cases will be recommended, as shown in green chart of Figure 1. Other items related to Marvel will be recommended according to general user interests, as shown

in blue chart in Figure 1. Actually, more attention should be focused on phone case by considering user's long-term preferences and current consumption preferences, and the phone case with "Marvel" will be more suitable for user's taste. Therefore, the personalized recommendation should not only consider user's long-term interest but also consider current short-term interest and give recommendation result by distinguishing user behaviors. Existing works mainly include two categories: the first category is general recommendation, such as collaborative filtering [2], matrix decomposition [3], BPRMF [4], and so on. Most methods focus on extracting feature similarity of users or items, ignoring preference dynamics hidden in behavior sequences. The other category is the next item recommendation based on behavior sequences [4–6], which is recorded by associated timestamps, and accumulated data can model current consumption preferences [7]. But, the preference dynamics and how to perfectly combine historical stability preferences with current consumption preferences remain largely unexplored. In addition, user's interests in real life changes over

FIGURE 1: Recommendations that combine long-term preferences with short-term consumer motivation.

time [8]. In the long run, topics of user's interest are stable or fluctuating, while current content that users interested in is often affected by user concerns. For example, a user may focus on the topic of "sports". He may be more concerned about "English Premier League" in near future, while next month may turn to "NBA." Therefore, effective information filtering and personalized recommendations are particularly important. Most existing models directly adopt user behaviors as interest, ignoring the mining and representation of implicit features, but user interests are hidden in behavior sequences, which ignores dependencies between behaviors [9]. In addition, the only thing that affects final behavior is interests associated with the target item.

Therefore, this paper proposes a behavior-aware neural network with attention (Att-BNN), which combines GRU with self-attention to model long-term and short-term preferences for users. Firstly, the embedding vector of items is obtained by unified representation space, and then user's behavior sequences is explored with self-attention network and GRU network, respectively. Experiments show that the proposed model is greatly improved compared with other sequence recommendation models, which proves that both local dependence and global validity should be considered. In summary, the main contributions of this article are as follows:

- (1) This paper constructs a new behavior-aware neural network with attention model (Att-BNN) for the next item recommendation, including item embedding with user interaction sequences and two neural networks—self-attention network and Bi-GRU network—to learn user's historical preferences and current consumption preferences, and then integrate them closely to better locate user interests.

- (2) In long-term preferences learning module, we introduce attention to Bi-GRU for modeling user interest change, which can identify items that are more relevant to user's interest.
- (3) The experimental results demonstrate that proposed model Att-BNN achieves significant and consistent improvements over state-of-the-art models. In addition, the impact of key parameters and model structure has been extensively studied to verify the effectiveness of the proposed method.

2. Related Work

There are two lines of research related to our work. The first involves common recommendation methods based on user interests, and the second is associated with attention mechanism employed by recommendation. We present brief reviews of these two research as follows.

2.1. User Interest-based Recommendation. Recommendation systems play an important role in many Internet-based services, such as e-commerce. Related works can be summarized into two categories: general recommendations and sequence recommendations.

Most recommendation algorithms focus on mining the correlations between users or items. The most common collaborative filtering is based on neighborhoods [2] and factorization [10–12]. For example, Bell and Koren proposed a new method of forming user preference ratings by simultaneously obtaining interpolation weights of all nearest neighbors, which greatly improved the prediction accuracy [2]. Liu et al. [11] proposed the IMF method to explore

hesitant behavior of users in selecting similar products. In recent years, deep learning has also been successfully applied to classic user-item matrix reconstruction problems in collaborative filtering, for example, combined with deep learning to extract and fuse auxiliary features, such as image features [13] and text information [14]. Most of these are based on mining static correlations between users or items. The dynamic changes of user preferences and current consumption preferences have not received much attention. On the other hand, sequence-based recommendations have received increasing attention in recent years. Gradually, accumulated interaction data can be modeled dynamically to obtain user preferences [15, 16]. For example, Xiong et al. [17] established a Bayesian probabilistic tensor factorization method to model temporal changes. He and McAuley [16] proposed a sequence recommendation method that combines similarity-based methods with Markov chains. In recent years, the combination of different neural networks has been widely used in representation learning and sequence modeling, especially in sequence recommendation [18]. For example, Hidasi et al. [6] first applied RNN (recurrent neural network) to session-based recommendations to model the entire session. Quadrana et al. [19] focused on session-based recommendation and proposed a hierarchical recurrent neural network for cross-session information transfer. Recently, Tang and Wang [20] proposed a convolutional sequence embedding model that uses horizontal and vertical convolutional filters to learn user's instantaneous trajectory, which has better performance [21]. Although these methods have taken into account user's sequence information, there has not been much development in behavior consistency and preference dynamics.

2.2. Attention-Based Deep Recommendation. Neural network techniques have been widely used in recommendation systems. But most RNN-based models directly take hidden state in sequences as user interests and treat the dependencies between all adjacent behaviors as equivalent. However, not all user behaviors depend on each adjacent behavior. For target items, these models can only get a fixed track of interests and are disturbed by interest changes. Inspired by human visual attention, in [22], an attention mechanism for natural language processing is proposed. The attention mechanism simulates characteristics of human brain attention. The core idea is to allocate more attention to important content, less attention to other parts, and focus on most important part of target. It has been widely used in many fields such as natural language processing and computer vision. Attention mechanisms can be combined with CNN and RNN to overcome shortcomings. For example, Bahdanau et al. [23] proposed the concept of neural attention, emphasizing weighting each word by learning attention vector to highlight important parts. Chen et al. [24] introduced attention mechanisms into collaborative filtering in multimedia recommendations to model item-level and component-level implicit feedback. Wang et al. [25] proposed the DAMD model for article recommendation and adaptively combined multiple prediction models using the attention model. In [26], a method to mine the hidden features in

advertising data for click-through rate prediction is proposed. In [27], a hierarchical attention model for rating prediction by leveraging user and product reviews is proposed. In order to capture the process of interest changes related to target item, in question and answer system (QA), Xiong et al. [28] used attention-based GRU model to enhance the sensitivity of position and input, inspired by AGRU in QA system, Zhou Et al. [29] proposed GRU with attentional gate (AUGRU) to activate relevant interest in the process of interest change.

In this paper, we present a behavior-aware neural network with attention [22], suggesting that attention mechanism can learn the importance of past behavior. This paper designs a model that combines self-attention and Bi-GRU to learn user's short-term preferences and long-term stable preferences, both of which come from user's rich interaction behavior. Especially, in long-term preferences learning process, attention mechanism is used to capture user's interest changes so as to locate user's interest more accurately.

3. Methods

We will introduce the proposed model in this section. First, we give a formal definition of next item recommendation task and introduce the proposed Att-BNN model, which is composed of the following major components: items embedding, short-term preferences learning with self-attention, and long-term preferences learning based on Bi-GRU.

3.1. Preliminaries. The personalized next item recommendation can be defined as follows: since user interaction naturally forms a sequence over time, the log history H in information system is a set of sequence interactions, $S = \{H_1, H_2, \dots, H_n\}$, where n indicates the number of users. Given target user u and its interaction sequence $H_u = \{(x_1, b_1), (x_2, b_2), \dots, (x_T, b_T)\}$ $H_u \in S$, x_j represents the j th item that user u interact and b_j indicates the type of behavior (for example, click, collect, and purchase). The personalized recommendation task is to predict the item x_{T+1} that user u is most likely to access next time.

In this paper, we propose a novel behavior-aware neural network with self-attention (Att-BNN) to solve the next item recommendation task. As shown in Figure 2, Att-BNN consists of three main components: item embedding and short-term and long-term preference learning. Specifically, the unified item representation space is obtained through neural embedding model to learn the hidden vectors of item, and sequence similarity between items is captured. Then, two interactive behavior learning networks are designed, which are named as short-term interest learning with self-attention and long-term preference learning based on Bi-GRU to explore the current consumption interest and historical stability preference of users over time. Finally, combine the two interactions in potential space to derive representation vector of target items and recommend possible top- k items to user.

3.2. Item Embedding. In the first stage of the Att-BNN model, each item is embedded in a unified representation space by learning the similarities between items. The

FIGURE 2: Behavior-aware neural network with attention.

previous methods typically represent item with one-hot coding or added additional embedding layers in deep learning framework. Actually, one-hot encoding network may take a lot of time, and it cannot be optimized well due to high sparsity [15]; on the other hand, adding additional embedding layers may make performance degradation to some extent [30]. Moreover, neither of these methods can reflect item similarity implied in user interaction. Therefore, it is necessary to find an effective method to learn high-quality item vectors directly from user behavior sequences, so that similar items are closer to each other. In recent years, neural embedding technology has made great progress in many fields, such as NLP [31, 32] and social networks [33]. In these efforts, item2vec [34] is one extension of skip-gram with negative sampling model [5], which produces item embedding for item-based collaborative filtering recommendations.

In order to further capture characteristics of behavior sequence, this paper uses improved item embedding method, called w-item2vec, which regards the frequency of item as weighting factor, and directly generates item representation. Inspired by item2vec [34], w-item2vec also uses skip-gram with negative sampling method [5]. Specifically, given the sequence $H'_u = \{x_1, x_2, \dots, x_N\}$ of user u from interaction sequence H_u , the skip-gram model of w-item2vec aims to maximize the following objective function:

$$\arg \max \text{opt} = \frac{1}{K} \sum_{i=1}^K \sum_{j \neq i}^K \log p(x_j | x_i), \quad (1)$$

$$p(x_j | x_i) = \frac{\exp(w_i^T * v_j)}{\sum_k (\exp(w_i^T * v_k))}, \quad (2)$$

where K is the length of the sequence S_u , $p(x_j | x_i)$ is the softmax function, and w_i and v_i are the target item x_i and its context hidden vector. To alleviate the complexity of calculating gradient $\nabla(x_j | x_i)$, equation (2) is usually replaced by negative sampling:

$$p(x_j | x_i) = \sigma(w_i^T * v_j) \prod_{k=1}^Q \sigma(-w_i^T * v_k), \quad (3)$$

where $\sigma(x) = 1/(1 + \exp((-x)))$ and Q is number of negative samples corresponding to each positive sample. The frequency of item that occurs in interaction sequence is taken as the weight in negative sampling process, and following improvement is made to formula (3):

$$p(x_j | x_i) = \sigma^{\Lambda(x_i)}(w_i^T * v_j) \prod_{k=1}^Q \sigma(-w_i^T * v_k)^{\Lambda(x_i)}, \quad (4)$$

where $\Lambda(x_i)$ is weight of item x_i in sequence. And, objective function in equation (1) can be defined as follows:

$$\begin{aligned}
\arg \max opt &= \frac{1}{K} \sum_{i=1}^K \sum_{j \neq i}^K \log p(x_j | x_i) \\
&= \frac{1}{K} \sum_{i=1}^K \sum_{j \neq i}^K \Lambda(x_i) (\Lambda(x_i) * \log \sigma(w_i^T * v_j)) \\
&\quad + \sum_{k=1}^Q \log \sigma(-w_i^T * v_k).
\end{aligned} \tag{5}$$

Finally, w-item2vec is trained by gradient descent to obtain a high-quality distribution vector representation of all items. Based on w-item2vec, user's interaction sequence can be used to capture items similarity and form a unified item representation space. For each user u , an embedding sequence $P_u = \{v_1, v_2, \dots, v_T\}$ can be formed with the set of embedding items, where v_j represents d -dimensional embedding vector of item x_j .

3.3. Short-Term Preference Learning with Self-Attention. Recommendation can be regarded as sequence problem, that is, predicting next possible behavior through items sequence in the interaction history. The common solutions to sequence problems are RNN and CNN. RNN can capture relationship between adjacent items and long-term relationship that is preserved in hidden layer state; CNN can capture relationship between items through sliding window of convolution kernel. However, neither of these methods explicitly models the relationship between items, and it is necessary to model this possible relationship. The short-term interaction behavior reflects user's recent interest, so modeling short-term interaction behavior helps to better understand current motivation of users. Therefore, this paper draws on attention mechanism proposed in neural network, applies it to recommendation to capture sequence pattern, models user's recent interaction trajectory, and improves the interpretability and accuracy of recommendation. The specific structure is shown in Figure 3. This is explained in detail in Section 3.3.2.

Self-attention is a kind of attention mechanism, which is formed by self-matching of a single sequence and successfully applied in various fields. Different from using the limited knowledge in overall context to learn representation based on standard attention mechanism, the self-attention can maintain the context sequence information and capture the relationship between elements in sequence without considering the distance, which focus on the common learning and self-matching of two sequences, in which the attention weight of one sequence is conditioned on another sequence. It has just received attention because of the recent successful application to machine translation. It can replace RNN and CNN in sequence learning to achieve higher accuracy with lower computational complexity. In this paper, we use the self-attention mechanism to model the dependence and importance of user's short-term behavior pattern and learn user's recent interest from the recent interaction record.

The input of self-attention module is divided into three parts: *Query*, *Key*, and *Value*. The output is the attention

value obtained by weighting *Value* according to weight coefficient, and the weight matrix is calculated by similarity between *Query* and *Key*. The values of *Query*, *Key*, and *Value* are derived from user's recent interaction history. The basis of self-attention mechanism is a scaled dot-product, except $\sqrt{d}$ which is used to adjust the inner product so that it is not too large or too small. It is assumed that user's short-term interest preference can be obtained from the L (e.g., 5 and 10) items that user has recently interacted with. Each item can be represented as a d -dimension embedding vector. The short-term embedding sequences $X_t^u = \{v_{t-L}, \dots, v_{t-2}, v_{t-1}\}$ is derived from P_u . The recently interacted items make up the matrix X_t^u :

$$X_t^u = \begin{bmatrix} v_{(t-L)1} & v_{(t-L)2} & \dots & v_{(t-L)d} \\ \vdots & \vdots & \vdots & \vdots \\ v_{(t-2)1} & v_{(t-2)2} & \dots & v_{(t-2)d} \\ v_{(t-1)1} & v_{(t-1)2} & \dots & v_{(t-1)d} \end{bmatrix}. \tag{6}$$

In self-attention mechanism, the values of *Query*, *Key*, and *Value* of user u are equal to X_t^u at time step t . *Query* and *Key* obtain corresponding Q' and K' by nonlinear transformation, where W_Q and W_K are weight matrices of *Query* and *Key*, and ReLU is used as activation function for nonlinear transformation. The final weight matrix can be calculated as follows:

$$\begin{aligned}
Q' &= \text{ReLU}(X_t^u W_Q), \\
K' &= \text{ReLU}(X_t^u W_K), \\
s_t^u &= \text{softmax}\left(\frac{Q' K'^T}{\sqrt{d}}\right).
\end{aligned} \tag{7}$$

The output s_t^u is an $L \times L$ weight matrix, which represents the similarity of L items. Finally, *Value* is weighted and summed to obtain the output of self-attention module. $\sqrt{d}$ is used to scale attention of dot product, and d is usually set to a larger value (such as 100), so the scaling factor can slow down the gradient effect. Using mask operation (the diagonal of mask factor matrix) before softmax avoids high matching scores between same vector of *Query* and *Key*. Finally, the coefficient matrix is multiplied by *Value* to form final weighted output of self-attention module:

$$a_t^u = s_t^u X_t^u, \tag{8}$$

where $a_t^u \in R^{L \times d}$ is user's L short interest representations. In order to learn one attention finally, L self-attention representations are averaged as user's final short-term interest preference ψ_s :

$$\psi_s = \frac{1}{L} \sum_{l=1}^L a_{il}^u. \tag{9}$$

3.4. Long-Term Preference Learning Based on Bi-GRU. A good recommendation algorithm should consider not only user's current consumption motivation but also long-term

FIGURE 3: Self-attention module.

stability preferences. Therefore, in addition to using self-attention to model user's short-term interests, this paper uses Bi-GRU based on attention to learn user's long-term preference. The module mainly consists of two parts, the first part is Bi-GRU and the second part is an attention mechanism, which provides a weight vector for each hidden layer state of the Bi-GRU layer. The series weight vectors are obtained by dot product of hidden state in Bi-GRU, and final weighted GRU hidden state is learned to obtain item embedding representation.

In user's long-term behavior, only part of behavior can reflect user's preference, so first define a function to determine which interactions $(v_l, b_l) \in H_u$ can be an element of long-term preference behavior. The function is defined as follows:

$$D_{PBL}(v_l, b_l) = \emptyset (b_l \in B). \quad (10)$$

Among them, B is a set of preference behaviors, which contain types of preference behaviors, such as collections and purchases. So, the user's long-term embedding sequences $D_{PBL}(v_l, b_l) = \{(v_{l1}, b_{l1}), (v_{l2}, b_{l2}), \dots, (v_{lT}, b_{lT})\}$, $v_l \in P_u, b_l \in B$.

Unlike short-term preference learning, long-term preference learning is a global representation of a user's historical preferences. User's historical behavior data are very large even for a short period of time (such as two weeks). In order to balance efficiency and performance, this paper uses LSTM variant GRU to model dependence between user behaviors, and its input is user's historical interaction behavior. Hidasi et al. [6] have demonstrated that GRU performs better than basic RNN units and long-short time memory units (LSTM) [35] in session-based recommendation tasks, so we use GRU instead of the most basic RNN network. GRU is better to handle gradient disappearance problems and has lower complexity than LSTM.

Inspired by [36] and bidirectional RNN [37], in order to obtain the dependencies between items in sequence, this paper proposes a contextual Bi-GRU to learn user's current consumption motivation, called Bi-CGRU. After initialization, the hidden state of each interaction at step t is obtained by hidden state of previous state, current item embedding, and behavior vector. Among them, the Bi-CGRU cell status is as follows:

$$\begin{aligned} u_t &= \sigma(W^u v_t + U^u h_{t-1} + B^u b_t + b^u), \\ r_t &= \sigma(W^r v_t + U^r h_{t-1} + B^r b_t + b^r), \\ \tilde{h}_t &= \tan h(W^h v_t + r_t \circ U^h h_{t-1} + B^h b_t + b^h), \\ h_t &= (1 - u_t) \circ h_{t-1} + u_t \circ \tilde{h}_t. \end{aligned} \quad (11)$$

where σ is sigmoid activation function, $\circ$ is element product, u_t and r_t represent the update gate and reset gate of step t , v_t is item embedding vector, b_t is behavior vector, b^u, b^r , and b^h are bias terms, h_t is output of step t , $W^u, W^r, W^h \in R^{n_H \times n_I}$, $U^u, U^r, U^h \in R^{n_H \times n_I}$, n_H is the hidden unit size, and n_I is the input size.

At each time step t , the forward hidden state $\vec{h}_t^p$ is calculated from the previous hidden state $\vec{h}_{t-1}^p$ and current item-behavior pair. The backward GRU uses future state $\overleftarrow{h}_{t+1}^p$ and current item-behavior to update $\overleftarrow{h}_t^p$. Therefore, in long-term preference learning, each state can be connected by the forward and backward hidden states, as follows:

$$h_t = \left[\vec{h}_t^p \parallel \overleftarrow{h}_t^p \right]. \quad (12)$$

3.5. Interest Change Learning. Affected by external environment and internal cognitive changes, user interests tend to change over time. Therefore, in order to accurately recommend next items that meet user interest, it is necessary to grasp process of user's interest change while digging user's interest. By analyzing the characteristics of user changes, local activation ability of attention mechanism and sequence learning ability in GRU are combined to model user interest changes. Each step of local activation in GRU enhances the impact of related interests and diminishes interest drifting, which helps to better obtain user interest changes.

The hidden state of GRU in interest change module is represented by $i_t' h_t'$, the input is the output state $i_t' = h_t$ in the Bi-GRU layer, and the final interest is represented by the last hidden state h_T' . The attention function in interest change module is defined as follows:

$$\alpha_t = \frac{\exp(h_t W e_a)}{\sum_{j=1}^T \exp(h_j W e_a)}, \quad (13)$$

where e_a is embedding vector of target item, $W \in n_H \times n_A$, n_H is dimension of hidden state, and n_A is embedding vector dimension of target item. The attention score reflects relationship between target item e_a and input h_t . The closer the relationship, the greater the attention score.

The following describes how to combine attention mechanisms and GRU to model user interest changes. This article combines GRU and attention update gates to learn user's interest changes, namely, AUGRU. Some researchers originally proposed GRU with attentional input (AIGRU), which directly affects the input of GRU with α_t , but this method is not very effective, because even if 0 is entered, it will change the hidden state of GRU, so the less relevant interest will also be learnt that affects the interest change. Later, some researchers made improvements and proposed attention-based GRU (AGRU), which uses the attention score to replace the update gate in GRU and directly control the update of the hidden state. However, in the original GRU, it is a vector to control the hidden state update. To solve this problem, this paper uses α_t to affect the GRU update gate u_t . As shown in Figure 4, it indirectly affects the hidden state update and preserves the impact of each dimension:

$$\begin{aligned} \tilde{u}'_t &= \alpha_t * u'_t, \\ h'_t &= (1 - \tilde{u}'_t) \circ h'_{t-1} + \tilde{u}'_t \tilde{h}_t. \end{aligned} \quad (14)$$

where u'_t is original update status of AUGRU, $\tilde{u}'_t$ indicates the attention update gate in AUGRU, and h_t , h'_{t-1} , and $\tilde{h}_t$ indicate the hidden states of AUGRU. In AUGRU, all dimensions of gate are updated by the attention score, so that the irrelevant interest has less influence on update gate, and the interference in interest change process is effectively avoided.

We use the last hidden state h'_T of AUGRU to represent user's preference behavior, so as to get user's long-term interest preference $\psi_l = h'_T$. The embedding vector of user interaction sequence is used as the input of the above two networks. The long-term preference learning can express the interest of each user, which helps to better understand the historical stability preference of user. In the process of user preference modeling, ψ_s and ψ_l are obtained by modeling user behavior from short-term consumption motivation and long-term preference, respectively. Then, through a fully connected layer, we can get d -dimensional representation of the next item.

3.6. Model Learning. Given the short-term attention preference and long-term preference at time step $t-1$, the task of this paper is to predict the items (indicated by S'_t) that user interacts at time step t . By using embedding vector of sequence item as the input of the network, the behavior learning module can learn user's current consumption motivation and historical preference by controlling the state updates of the two network structures. After behavior learning, the next item that can be predicted is represented by v_t , which is a d -dimensional vector. Two behaviors are

FIGURE 4: The structure of AUGRU.

learned from the entire interaction sequence H using cross entropy loss function, defined as follows:

$$C = \frac{1}{|S|} \sum_{H_u \in S} \frac{1}{(|H_u| - L - 1)} \sum_{t=L+1}^{|H_u|} \xi(\hat{v}_t, v_t), \quad (15)$$

where ξ is the CEL function, v_t is the item representation of target user u next access, L is the control signal, $|H_u|$ indicates the length of the interaction sequence H_u , and $|S|$ is the number of users; more details are introduced in experimental section.

After training the model, in test stage, given a user's interaction history $H_u = \{(x_1, b_1), (x_2, b_2), \dots, (x_{t-1}, b_{t-1})\}$, the next possible interaction of user u can be predicted as follows: (1) using model to train user's historical interaction sequence H_u to obtain user's short-term interest preference and long-term interest preference at $t-1$; (2) forming next item embedding vector v_t and calculating similarity with all candidate items in the hidden space; (3) recommending top k possible items to the user in the unified representation space.

4. Experiment

This paper conducts extensive experiments on the proposed model based on realistic datasets. This section first describes the experimental setup and then demonstrates the validity of the proposed model from several aspects.

4.1. Datasets. We experimented with the following data sets.

MovieLens: a popular benchmark dataset for evaluating the performance of recommended models. Two versions were used in the experiment: MovieLens 100K and MovieLens 1M. The type of behavior includes click and collect.

Amazon: a user-purchase and rate dataset, a well-known e-commerce platform. Due to limited space, there are three subcategories in this work: Android Apps, digital music, and instant video.

LastFM: a dataset with user tags collected from the last.fm online music system.

JD: provided by Jingdong, a Chinese e-commerce company, one of the two largest B2C online retailers in China by volume and revenue. The type of behavior includes click, collect, and purchase.

For the reliability of experimental results, we did necessary preprocessing. First filter out those users whose interaction behavior less than 10 and those with less than 5 occurrences. The dataset is divided into training set and test set, respectively, where 90% of interaction behavior is selected as training set and rest for testing. Considering that the collaborative filtering method cannot recommend items that did not appear before, we filter out item interactions from test set that are not present in training set. Similarly, we will also remove user information that does not appear in the training set from the test set. The dataset after data preprocessing is shown in Table 1.

4.2. Baseline Method. In order to verify the validity of Att-BNN model, we compare it with four traditional methods and two advanced RNN-based models on different evaluation indicators. The following is a brief introduction to these several baseline models.

S-POP: a method of sorting items based on popularity and recommending items to user with the highest frequency. The list of recommendations changes as the sequence grows.

BPRMF [38]: it uses paired Bayesian personalized sorting optimization to maximize the gap between positive and negative items, which does not model sequence information.

FPMC [39]: a method of combining matrix factorization and Markov chain for the next item recommendation, which can capture user preferences and sequence behavior. Since there is no specific user profile in sequence-based recommendation, in order to make FPMC applicable to sequence-based recommendation, the potential representation of user is not considered in calculating candidate item recommendation score.

Item-KNN [21]: selects items similar to the previous items that are accessed by the user, in which the similarity between items is defined as the number of co-occurrences in the sequence.

GRURec: a basic GRU based on top1 loss function of the parallel small batch, proposed in [6]. This model uses parallelism and minibatch training to learn model parameters.

GRU4Rec Con: similar to GRURec, except that this method does not use session partitioning, and user's interaction sequence is sent to GRURec independently as a whole [40].

HRNN Init [19]: a hierarchical RNN for personalized cross recommendation based on GRU4Rec and adding an additional GRU layer to model the information in user session to track user interest changes. It is a more advanced method in the next item recommendation.

Among all these methods, POP, BPRMF, FPMC, and Item-KNN are traditional recommendation methods, and GRURec and HRNN are neural network-based methods.

4.3. Evaluation Indicators. Since the recommendation system can recommend several items at a time and related items should be placed in front of the recommendation list, in order to evaluate the performance of the next item

recommendation model proposed in this paper, two evaluation indicators Recall and normalized discounted cumulative gain (NDCG) are used in this paper.

Recall: we use Recall@20, which measures the probability that the items actually clicked by the user in the test set will appear in top 20 of the recommended list. Recall @N does not consider the order in which the user actually clicks on the item in the recommendation list, as long as it appears in the top N position of the recommendation list. In addition, this indicator is often used with indicators such as click-through rate (CTR).

NDCG: evaluates the quality of recommended list based on the sorted position of the correct item. Evaluating a recommendation system makes it impossible to evaluate not only one user's recommendation list and corresponding results but also to evaluate users in the entire test set and their recommended list results. The evaluation scores for different user recommendation lists need to be normalized and defined as follows:

$$\text{NDCG@N} = \frac{1}{Z} \sum_{i=1}^N \frac{2^{r_i} - 1}{\log_2(i + 1)}, \quad (16)$$

where r_i is the hierarchical correlation of the items at location i . In this paper, a simple binary correlation is used: if the item is in the test set, $r_i = 1$, otherwise 0. Z is the normalized constant, which is the maximum possible value of DCNG@N.

In short, the higher the two evaluation indicators, the better the results.

4.4. Implementation Details. This paper implements the proposed model and several previous baseline models on Tensorflow. For fair comparison, all hidden units in the RNN model are set to 200. First, the embedded matrix and GRU weights are initialized. Each dataset is trained to be a 200-dimensional word embedding vector in advance with skip-gram, and then 100 hidden units are created in the GRU model, such that the combination of forward and backward GRUs can form a 200-dimensional item representation. Therefore, the embedding dimension of item is set to 200 and is randomly initialized based on a uniform distribution $(-0.01, 0.01)$. In this paper, an adaptive gradient optimizer with a batch size of 64 is used, and the initial learning rate of all datasets is set to 0.01 for optimization. The weight matrix of nonlinear layer in self-attention module is regularized by 12. The regularization rate λ is adjusted between $\{0.1, 0.01, 0.001, 0.0001\}$, and the packet loss rate is adjusted between $\{0, 0.3, 0.5, 0.7\}$. In MovieLens, the sequence length L is set to 5 and all other datasets are set to 3. A packet loss method with a packet loss rate of 0.5 is used at the fully connected layer to reduce overfitting.

4.5. Performance Comparison. To demonstrate the importance of the proposed model in this paper, we compared the other methods with Att-BNN in the next item recommendation task. Table 2 gives the experimental results of the seven basic methods and the model proposed in this paper

TABLE 1: Experimental dataset statistics.

Dataset	No. of users	No. of items	No. of interactions	Avg. no. actions per user
ML-100k	932	1,670	99,684	106.9
ML-1M	6,000	3,692	99,8959	166.49
Android app	21,256	19,183	355,499	16.72
Digital music	2,763	13,095	64,277	23.26
Instant video	1,000	3,267	15,796	15.796
LastFM	938	12,073	83,329	88.84
JD	102,703	24,677	37,059,872	360.84

TABLE 2: Performance comparison in terms of Recall@20 and NDCG on all datasets.

Dataset	Metric	S-POP	BPRMF	FPMC	Item-KNN	GRURec	GRU4RecCon	HRNN Init	Att-BNN
ML-100K	Recall@20	0.3044	0.3131	0.4562	0.5160	0.6064	0.6784	0.6935	0.7094
	NDCG	0.1835	0.1208	0.1501	0.2108	0.2289	0.2900	0.3087	0.4108
ML-1M	Recall@20	0.2708	0.3400	0.4125	0.5231	0.5953	0.6911	0.7031	0.7344
	NDCG	0.1775	0.0157	0.1619	0.2170	0.2260	0.2922	0.3047	0.4258
Android app	Recall@20	0.2107	0.1519	0.3155	0.2835	0.4382	0.5795	0.5821	0.6391
	NDCG	0.1469	0.0863	0.0892	0.0945	0.1546	0.2493	0.3047	0.5022
Digital music	Recall@20	0.1206	0.1321	0.2107	0.1680	0.2896	0.4872	0.4966	0.6231
	NDCG	0.0773	0.0731	0.0820	0.0822	0.1883	0.2043	0.3152	0.4671
Instant video	Recall@20	0.2315	0.1724	0.3428	0.3147	0.4526	0.6031	0.6119	0.6359
	NDCG	0.1257	0.0913	0.0936	0.0917	0.1629	0.2503	0.2634	0.4062
Last FM	Recall@20	0.3162	0.2752	0.3769	0.4265	0.5371	0.7254	0.8105	0.8735
	NDCG	0.2355	0.0861	0.1048	0.1327	0.1852	0.3372	0.4759	0.5582
JD	Recall@20	0.5854	0.1873	0.2433	0.1246	0.7034	0.7934	0.8073	0.8593
	NDCG	0.2176	0.0664	0.0496	0.0361	0.4198	0.5932	0.6098	0.6742

on several datasets. It can be observed that Att-BNN shows the best performance on all datasets. This illustrates the effectiveness of the proposed method. It is worth noting that the method in this paper is superior to several baseline models in prediction accuracy and ranking quality. This suggests that Att-BNN good at solving personalized sequence information based on user interaction, not only on dense data sets like MovieLens but also on sparse data sets such as Amazon.

In addition, we analyze several comparative baseline methods. FPMC based on Markov chain can achieve consistent performance on dense and sparse datasets. When the user factor is replaced by the average value of hidden vector of item that has already appeared in current sequence, BPRMF does not perform well on some datasets, and the model is based on the assumption that the next item of user is only affected by the most recent action. This assumption may apply to sparse data because interactions are very discrete in time. However, it may not be suitable for frequent interaction between users and system, which also reflects that some matrix factorization methods based on users are not suitable for sequence-based recommended tasks. On the whole, all RNN-based models (GRURec, HRNN, and Att-BNN) are obviously superior to the traditional methods, which show that the RNN-based models that model interaction sequence for the user have better ability to recommend the next item than traditional methods. The comparison between RNN-based models highlights the effectiveness of considering users' long-term preferences in the next item recommendation, so the model considering personalized information (such as Att-

BNN and HRNN Init) is superior to those that do not consider (GRURec and GRURec Con). In addition, we note that the RNN-based model performs better on JD datasets than on other datasets. One possible explanation is that JD has more users and interactions, while the number of goods is relatively small. Moreover, the number of users in JD dataset is much larger than that of items, which makes the prediction candidate set smaller and the results more accurate. In fact, this scenario is more common in real world, such as online retailers and B2C platforms.

In short, Att-BNN did its best on all datasets, followed by HRNN Init; both methods take user representation into account. This clearly indicates that user interaction behavior may follow the general short-term interest preference and reflect the stable long-term preference. The Att-BNN model proposed in this paper distinguishes the model user historical preference and current consumption motivation and considers user interest change in the process of modeling, so as to get better personalized recommendation results. In order to further understand the model proposed in this paper, the influence of some key parameters is explored, such as aggregation method and mining sequence length of short-term preferences.

4.6. Self-Attention Effects. Although the effectiveness of self-attention can be inferred from Table 2, in order to verify the efficiency of self-attention, the self-attention module is removed from Att-BNN and replaced by the average of X_t^u instead of m_t^u :

$$m_t^u = \frac{1}{L} \sum_{l=1}^L X_{tl}^u. \quad (17)$$

Table 3 gives a comparison with and without self-attention, and it can be observed that self-attention does improve the performance of model. It can also be seen from Tables 2 and 3 that even without self-attention, the model of this paper can still outperform all baseline models in all data sets. In addition, two observations were drawn from the results. First, the self-attention matrix is column distinguishable, and each column represents the importance and weight of the corresponding action. Intuitively, the self-attention matrix allows to check how much each action contributes to the final short-term preference representation. Second, Att-BNN does not simply emphasize recent interactions but automatically learns and assigns different weights to previous behaviors. For example, user “326” assigns a higher weight to the second and third items, while user “386” assigns a higher weight to the most recently interacted items. Obviously, attention weights can bring the benefits of interpretability.

4.7. Impact of Aggregation Method. As mentioned earlier, different aggregation strategies can be used to obtain a short-term preference representation of user. This article explores the applicability of four types strategies to verify their applicability. Table 4 shows the results of using different polymerization methods, and it can be observed that “average” has considerable performance on sparse and dense data sets. The other three aggregation methods seem to perform poorly in sparse data sets. This is also reasonable because m_t^u affects the embedding of the next item X_{t+1}^u and the average aggregation can retain more information.

4.8. Effect of Sequence Length L . Figure 5 shows the effect of sequence length L in short-term preference learning, and it can be observed that the value of L is highly dependent on the density of the data set. The average number of behaviors per user in MovieLens dataset is greater than one hundred, so setting L to a larger value helps improve performance. However, L should be set to a smaller value on sparse data set, as increasing L will result in a reduction in training samples. Self-attention can capture the dependencies between distant locations, theoretically allowing learning of lengthy sequences. This also justifies our approach to modeling preferences.

4.9. Effect of Potential Dimension d . Figure 6 shows Recall@20 for each dimension d while keeping other parameters constant. It can be observed that the Att-BNN model is always superior to other baseline models in all dimensions. Second, too large potential dimension does not necessarily make the model perform best, possibly because of overfitting. Third, some models are not stable, which may limit their usefulness.

TABLE 3: Recall@20 of Att-BNN with and without self-attention.

Dataset	ML-100K	ML-1M	Digital music	Instant video
With self-Att	0.7094	0.7344	0.6231	0.6359
Without self-Att	0.6952	0.7096	0.6109	0.6136

TABLE 4: Recall@20 of Att-BNN with different aggregation methods.

Dataset	ML-100K	ML-1M	Digital music	Instant video
Mean	0.7094	0.7344	0.6231	0.6359
Sum	0.6883	0.7120	0.6074	0.6246
Max.	0.7054	0.7147	0.5932	0.6166
Min.	0.7043	0.7159	0.5725	0.5829

FIGURE 5: Effects of the sequence length L on datasets.

4.10. User History Length Analysis. This paper further analyzes and evaluates the proposed Att-BNN model and other RNN-based models under different user history length information and verifies user’s ability to express long-term preferences. We believe that user history length has an impact on the performance of the recommendation system, so the evaluation is grouped based on the length of user interaction. Specifically, we use two datasets for analysis and divide users into three groups: historical interactions are less than 300, between 300 and 500, and over 500. The performance of these three user groups was recorded separately to measure the complex effects of long-term preference dynamics in the Att-BNN model and other RNN-based models.

As shown in Figure 7, first focus is on the improvement of Amazon dataset with the increase in user behavior length. It can be observed that the Recall@20 of Att-BNN model proposed in this paper has a significant increase with the increase of user history length. In MovieLens dataset, Att-BNN improved by at least 2.32% compared to other RNN-based models due to the large number of user interactions. Interestingly, the Att-BINN and HRNN models have improved with historical lengths, but GRURec and GRURec Con have not continued to

FIGURE 6: Effects of the number of latent dimensions on (a) MovieLens 100K and (b) Amazon Digital Music.

FIGURE 7: Effects of different history lengths. (a) Recall@20 and (b) NDCG on Amazon Digital Music. (c) Recall@20 and (d) NDCG on MovieLens.

improve over the 300–500 and above. In summary, the length of user interaction does have an impact on the performance of the recommendation system. These results clearly demonstrate the effectiveness of using personalization strategies (such as user historical stability preferences) to improve recommendation performance.

5. Conclusion

In order to solve the problem of personalized next item recommendation, this paper proposes a new behavior-aware neural network model with attention mechanism. By incorporating attention into the recurrent neural network, our model can both model the behavior characteristics of users in current sequence and capture user's short-term consumption intentions. Since the user behavior naturally forms a sequence of interactions with time, this paper fuses user's long-term historical preference and short-term consumption motivation to predict its next behavior. First, use w-item2vec to generate a unified item representation space, get embedding representation of interaction sequence, and then learn user behavior distinctively and propose two user behavior learning methods. In short-term preference learning module, the self-attention mechanism is used to learn user recent behavior. On the other hand, we use the improved Bi-GRU model to learn user's long-term historical preferences, use the attention mechanism to model user interest changes, and better capture the interest associated with target item. Finally, extensive experiments were conducted in the dataset, and our model was superior to existing sequence-based recommendation algorithms in different evaluation indicators. The experimental results clearly show that the Att-BNN model can accurately capture the importance of the user's recent behavior and has good validity in the personalized next item recommendation.

In future work, we believe that the model can be improved in many aspects. Due to the limitation of dataset, we only use the ID information of commodity in the model, the information obtained is very limited, and there are certain restrictions on the model. In dataset, in addition to the item ID information, other information such as price and category of the item can be obtained. In future work, the dataset with such information can be focused on, and other auxiliary information can be integrated into the model to improve the recommendation accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the following grants: the National Natural Science Foundation of China 61772321, the

Natural Science Foundation of Shandong Province ZR2016FP07, the Innovation Foundation of Science and Technology Development Center of Ministry of Education, and New H3C Group 2017A15047 and CERNET Innovation Project NGII20170508.

References

- [1] S. Zhang, L. Yao, and A. Sun, "Next item recommendation with self-attention," *Computer Science*, 2018, <https://arxiv.org/abs/1808.06414>.
- [2] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 43–52, Omaha, NE, USA, October 2007.
- [3] X. He, H. Zhang, M.-Y. Kan, and C. Tat-Seng, "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*, ACM, pp. 549–558, Pisa, Italy, July 2016.
- [4] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the UAI*, Montreal, Canada, June 2009.
- [5] L. Jing, P. Ren, Z. Chen, Z. Ren, T. Lian, and M. Jun, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*, pp. 1419–1428, Singapore, November 2017.
- [6] B. Hidasi, A. Karatzoglou, L. Baltrunas et al., "Session-based recommendations with recurrent neural networks," *Computer Science*, 2015, <https://arxiv.org/abs/1511.06939>.
- [7] B. Wu, T. Mei, W.-H. Cheng, and Z. Yongdong, "Unfolding temporal dynamics: predicting social media popularity using multi-scale temporal decomposition," in *Proceedings of the AAAI*, pp. 272–278, Phoenix, AZ, USA, February 2016.
- [8] G. Zhou, X. Zhu, C. Song et al., "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp. 1059–1068, London, UK, August 2017.
- [9] J. Chung, C. Gulcehre, K. Cho et al., "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, <https://arxiv.org/abs/1412.3555>.
- [10] L. Qi, Y. Ge, Z. Li, C. Enhong, and H. Xiong, "Personalized travel package recommendation," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pp. 407–416, Vancouver, BC, Canada, December 2011.
- [11] L. Qi, X. Zeng, H. Zhu et al., "Mining indecisiveness in customer behaviors," in *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, pp. 281–290, IEEE, Atlantic City, NJ, USA, November 2015.
- [12] H. Zhao, L. Qi, Y. Ge, K. Ruoyan, and C. Enhong, "Group preference aggregation: a nash equilibrium approach," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 679–688, IEEE, Barcelona, Spain, December 2016.
- [13] R. He and J. McAuley, "VBPR: visual bayesian personalized ranking from implicit feedback," in *Proceedings of the AAAI*, pp. 144–150, Phoenix, AZ, USA, February 2016.
- [14] X. He, L. Liao, H. Zhang, N. Liqiang, H. Xia, and C. Tat-Seng, "Neural collaborative filtering," in *Proceedings of the 26th*

- International Conference on World Wide Web (WWW'17)*, pp. 173–182, Perth, Australia, April 2017.
- [15] X. Chen, H. Xu, Y. Zhang et al., “Sequential recommendation with user memory networks,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining—WSDM'18*. ACM, pp. 108–116, Marina Del Rey, CA, USA, February 2018.
 - [16] R. He and J. McAuley, “Fusing similarity models with Markov chains for sparse sequential recommendation,” in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 191–200, IEEE, Barcelona, Spain, December 2016.
 - [17] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and G. C. Jaime, “Temporal collaborative filtering with bayesian probabilistic tensor factorization,” in *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 211–222, Columbus, OH, USA, April 2010.
 - [18] B. Wu, W.-H. Cheng, Y. Zhang, H. Qiushi, L. Jintao, and M. Tao, “Sequential prediction of social media popularity with deep temporal context networks,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3062–3068, Melbourne, Australia, August 2017.
 - [19] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, “Personalizing session-based recommendations with hierarchical recurrent neural networks,” in *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys'17)*, pp. 130–137, Como, Italy, August 2017.
 - [20] J. Tang and K. Wang, “Personalized top-N sequential recommendation via convolutional sequence embedding,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, pp. 565–573, Marina Del Rey, CA, USA, February 2018.
 - [21] F. Yu, Q. Liu, S. Wu, W. Liang, and T. Tieniu, “A dynamic recurrent model for next basket recommendation,” in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 729–732, Pisa, Italy, July 2016.
 - [22] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.
 - [23] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *Computer Science*, 2014, <https://arxiv.org/abs/1409.0473>.
 - [24] J. Y. Chen, H. W. Zhang, and X. N. He, “Attentive collaborative filtering: multimedia recommendation with item-and component-level attention,” in *Proceedings of the 40th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–344, Tokyo, Japan, August 2017.
 - [25] X. Wang, L. Yu, K. Ren et al., “Dynamic attention deep model for article recommendation by learning human editors demonstration,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD '17*, pp. 2051–2059, Halifax, NS, Canada, August 2017.
 - [26] Q. Wang, F. Liu, S. Xing, X. Shuning, and Z. Xiaohui, “A new approach for advertising CTR prediction based on deep neural network via attention mechanism,” *Computational and Mathematical Methods in Medicine*, vol. 2018, Article ID 8056541, 11 pages, 2018.
 - [27] S. Xing, F. Liu, Q. Wang, Z. Xiaohui, and L. Tianlai, “A hierarchical attention model for rating prediction by leveraging user and product reviews,” *Neurocomputing*, vol. 332, pp. 417–427, 2018.
 - [28] C. Xiong, S. Merity, and R. Socher, “Dynamic memory networks for visual and textual question answering,” in *Proceedings of the 33rd International Conference on Artificial Intelligence Conference on Machine Learning*, pp. 2397–2406, New York, NY, USA, June 2016.
 - [29] G. Zhou, N. Mou, and Y. Fan, “Deep interest evolution network for click-through rate prediction,” in *Proceedings of the KDD*, London, UK, August 2018.
 - [30] X. He, X. Du, X. Wang, T. Feng, T. Jinhui, and C. Tat-Seng, “Outer product-based neural collaborative filtering,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 13–19, Stockholm, Sweden, July 2018.
 - [31] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
 - [32] T. Mikolov, I. Sutskever, K. Chen, S. Corrado, and D. Jeff, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3111–3119, Lake Tahoe, NV, USA, December 2013.
 - [33] A. Grover and J. Leskovec, “node2vec: scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 855–864, San Francisco, CA, USA, May 2016.
 - [34] O. Barkan and N. Koenigstein, “Item2vec: neural item embedding for collaborative filtering,” in *Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, Vietri sul Mare, Italy, September 2016.
 - [35] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
 - [36] S. Ghosh, V. Oriol, S. Brian et al., “Contextual LSTM (CLSTM) models for large scale NLP tasks,” 2016, <https://arxiv.org/abs/1602.06291>.
 - [37] M. Schuster and K. P. Kuldip, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
 - [38] H. Guo, T. Ruiming, Y. Ye, L. Zhenguo, and H. Xiuqiang, “DeepFM: a factorization-machine based neural network for CTR prediction,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1725–1731, Melbourne, Australia, August 2017.
 - [39] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized Markov chains for next-basket recommendation,” in *Proceedings of the 19th International Conference on World Wide Web—WWW'10*, Raleigh, NC, USA, April 2010.
 - [40] Y. K. Tan, X. Xu, and Y. Liu, “Improved recurrent neural networks for session-based recommendations,” in *Proceedings of the 1st. Workshop on Deep Learning for Recommender Systems*, pp. 17–22, Boston, MA, USA, September 2016.

Submit your manuscripts at
www.hindawi.com