ITC 1/52

Information Technology and Control
Vol. 52 / No. 1 / 2023
pp. 111-127
DOI 10.5755/j01.itc.52.1.32138

Adaptive Context-Embedded Hypergraph Convolutional Network for Session-based Recommendation

Received 2022/08/22 | Accepted after revision 2022/11/07

https://doi.org/10.5755/j01.itc.52.1.32138

HOW TO CITE: Zhao, C., Cao, H., Lv, P., Chu, Y., Wang, F., Liao, T. (2023). Adaptive Context-Embedded Hypergraph Convolutional Network for Session-based Recommendation. *Information Technology and Control*, 52(1), 111-127. https://doi.org/10.5755/j01.itc.52.1.32138

Adaptive Context-Embedded Hypergraph Convolutional Network for Session-based Recommendation

Chenyang Zhao

College of Information Science and Engineering, Henan University of Technology, Zhengzhou, China; e-mail: zhaochy2005@163.com

Key Laboratory of Grain Information Processing and Control (Henan University of Technology), Zhengzhou, China; e-mail: zhaochy2005@163.com

Heling Cao

College of Information Science and Engineering, Henan University of Technology, Zhengzhou, China; e-mail: caohl@haut.edu.cn

Pengtao Lv, Yonghe Chu, Feng Wang, Tianli Liao

College of Information Science and Engineering, Henan University of Technology, Zhengzhou, China; e-mails: pengtaolv@163.com, chuyonghe@haut.edu.cn, wangfeng_haut@haut.edu.cn, tianli.liao@haut.edu.cn

Corresponding author: Chenyang Zhao(zhaochy2005@163.com)

Graph neural network (GNN) based approaches have been successfully applied to session-based recommendation. However, most of the existing methods do not fully take advantage of the context information in the session when capturing user's interest, and there are few studies on context adaptation. Moreover, hypergraph has potential to express complex relations among items, but it has remained unexplored. Therefore, this paper proposes an Adaptive Context-Embedded Hypergraph Convolutional Network (AC-HCN) for session-based recommendation. At first, all sessions are constructed as a session hypergraph. Then, the representation of each item in the session hypergraph is learned using an adaptive context-embedded hypergraph convolution.

In the convolution, different types of context information, both from the current item and its neighborhoods, are adaptively integrated into the representation learning of the current item. Meanwhile, an adaptive transformation function is employed to effectively eliminate the effects of irrelevant items in the learning. Next, the learned item representations are combined with time interval embeddings and reversed position embeddings to generate enhanced item representations, which fully reflect time interval information and sequential information between items in the session. Finally, based on the enhanced item representations in the session, we use a soft attention mechanism to obtain user's interest, and then give a recommendation list for user. Extensive experiments on the real-world datasets show that the proposed model is superior to the state-of-art methods.

KEYWORDS: Session-based recommendation, Hypergraph convolution; Context information; Context adaptation; Time interval.

1. Introduction

Nowadays, due to the information overload on the internet, users are facing the embarrassment of not being able to find the information they are interested in [36]. Recommendation system has become an essential tool to ease the problem. Conventional recommendation methods (e.g., collaborative filtering [21]) usually depend on complete user profiles and sufficient historical data, but in many real-world scenarios, such information is unavailable or limited available [7]. Thus, more and more researches are devoted to session-based recommendation (SBR). Generally, a session is denoted as a sequence of interactions in which a user purchases multiple items within a given time period, and SBR focuses on next-item prediction based on the current session.

The even-enriching context information, such as time, holiday, week and location, has been collected by recommendation systems through explicit or implicit user feedback [29]. Context information has been proved to play a pivotal role in improving recommendation accuracy [1]. For example, a user usually tends to relax at home on weekend, books on entertainment might be more attractive than books on professional topics. Therefore, how to fully take advantage of rich context information in SBR to improve the modeling of user's behaviors in a more appropriate way is a challenging and critical issue.

In recent years, a lot of research achievements using deep learning have been made in SBR. Among them, approaches based on Recurrent Neural Networks (RNNs) [10, 27, 39] and approaches based on Graph Neural Networks (GNNs) [8, 28, 30] have shown great performance.

In RNNs-based approaches, items in a session are time-dependent and ordered, so the session is typically modeled as unidirectional sequence. However, this assumption may trap these models, because there may be no such strict chronological order like linguistic sequences. In real scenarios, the relative order of items might not matter so much. For example, there might be no difference between a user playing a music album and playing the music from the album in order. Hence, the user's interest learned by these sequence models might be prone to inaccurate when the temporal order between items is considered too much [33].

Recently, GNNs-based approaches have become popular in SBR. Unlike the RNNs-based approaches, GNNs-based approaches model session data as directed graphs and regard item transitions as pairwise relations, which slightly relaxes temporal dependence between consecutive items. However, although these approaches focus on pairwise relations, they would probably ignore more complex item correlations. In reality, an item transition is often influenced by the joint effect of previous items and complex relations among items. Since hypergraph can express complex many-to-many and high-order relations among items, Hypergraph Neural Networks (HGNNs) [6, 12] have entered the field of vision of researchers. Later, Hypergraph Convolutional Network [2] is devised. After that, some researchers [26, 33] begin to apply hypergraph learning to SBR and achieve greater improvements compared with GNNs-based models.

Since context information plays a crucial role in modeling user's behaviors, researchers have been investigating how context information can be exploited in recommendation models when trying to implement SBR with various deep neural networks. At present, researches based on context awareness and deep neural networks fall into two categories: context-aware RNNs-based methods [17, 24] and context-aware GNNs-based methods [5, 14]. These studies further improve the recommendation accuracy.

Despite promising results of existing approaches, they still suffer from the following limitations. First, given a session, the user's general interest representation is usually obtained by aggregating item representations in the current session. However, existing researches neglect or fail to fully reflect the influences of context information on the item representation learning, resulting in the lack of context adaptation in the user interest representation. In other words, in addition to reflecting the complex high-order relations among items, the user interest representation should also embody the interaction scenarios in which user purchases items. Second, we also observe that different types of context information may have different degrees of impact on item representation learning. While existing studies neglect this aspect. Third, some HGNNs-based approaches, such as literature [33], relax the temporally dependence between items and extract complex high-order relations among items by connecting items with each other in a session. It is proven to be effective, but may lead to item representation learning being easily disturbed by irrelevant items. In final, many existing GNNs-based studies, such as literatures [28, 33], only focus on the effect of item order in session by using the reversed position embeddings, but ignore the time intervals between adjacent items, which represent the durations that user browses each item in session. These time intervals imply user's preference to some extent. In general, the more time user spends on an item, the more likely like it.

To address the above issues, we propose a novel Adaptive Context-Embedded Hypergraph Convolutional Network (AC-HCN) to model the high-order relations among items and context information by a more suitable manner. In AC-HCN, we propose to use hypergraph convolution to learn item representations from session hypergraph and context information. In addition, we also apply $\alpha - \mathrm{entmax}$ function [37] and time interval embeddings to further raise the recommendation accuracy.

The main contributions of this work are summarized as follows:

1. We propose a novel adaptive context-embedded hypergraph convolution for item representation learning, which incorporates context information into hypergraph convolution. Meanwhile, an adaptive $\alpha - \mathrm{entmax}$ function is applied to eliminate irrelevant items in the learning.
2. We employ attention mechanism to distinguish the influences of different types of context information on item representation learning.
3. In addition to the reversed position embeddings, we put forward the time interval embeddings to further improve the recommendation efficiency.
4. Extensive experiments conducted on the real-world datasets show that AC-HCN evidently outperforms the state-of-art baselines.

2. Related Work

In this section we first review some related work on SBR. Next, we introduce the context-aware recommendation methods using deep neural networks.

2.1. Session-based Recommendations

Much progress has been made in SBR, mainly including methods based on Markov chains, methods based on collaborative filtering and methods based on deep neural networks.

The initial SBR mainly focuses on Markov chains based methods, such as literatures [3, 32]. The item transitions in a session are modeled as Markov chain, and then the prediction probability is calculated to predict next item that is likely to be clicked. But the prediction only focuses on the local information of the session and lacks the consideration of the global information.

Methods based on collaborative filtering depend on the similarity between items according to the co-occurrence of items in the sessions. Item-KNN [21] proposed by Sarwar et al. is the representative work. However, the methods of capturing item relations in similarity way only obtain the co-occurrence of items, and fail to capture more accurate global information of the session. As another representation of collaborative filtering, matrix factorization methods, such as literatures [9, 13], decompose the user-item rating matrix to obtain user and item latent vectors. However, these methods are not suitable for new sessions that do not include any items (the cold start problem). Later, some researchers attempt to combine Markov chains and matrix factorization to model item relations for further improving the model accuracy. FPMC [20] presented by Rendle et al. is the representative work. However, since the natural shortcomings of these models, both methods based on Markov chain and methods based on collaborative filtering cannot be used to capture high-order relations among items.

With the boom of deep learning, RNNs have been attracted much attention because of their ability of exploiting sequential data. Many researchers have applied RNNs to SBR. Hidasi et al. [10] designed the GRU4REC method based on an improved Gated Recurrent Unit (GRU, variation of RNN) in SBR for the first time and achieved great success. Following the work, Li et al. [15] proposed the NARM method and Liu et al [16] put forward the STAMP method. Compared with GRU4REC, both methods utilized attention mechanism in RNNs. Later researchers explored further. Based on the assumption that user's behaviors are scattered across domains, Wang et al. [27] devised the cross-domain and user-level RNNs-based method to capture user's global interest from cross-domain sessions. Sheng et al. [22] fused a time-based directional attention mechanism with RNN to capture the sequential patterns in the session, which improves the accuracy of modeling user preference. Zhang et al. [39] applied two GRUs to explore the global and local preferences of user respectively, and then introduced a parallel co-attention mechanism to capture the interaction of both preferences. Although these RNNs-based methods can handle the dependence between items in the session, they overemphasize temporal order between items, which would probably make them prone to over-fitting.

To alleviate the above problem, GNNs-based methods recently have become the hotspot in SBR. Wu et al. [30] proposed the SR-GNN method which used gated graph neural network to capture user's local and global preferences, and utilized the soft attention to integrate the two kinds of preferences to achieve session representation. Following the success of SR-GNN, Yu et al. [35] presented a target-aware attention and employed it in gated graph neural network, which aimed to obtain dynamic user interest representation. Wang et al. [28] devised a session graph and a global graph from all sessions, and proposed the GCE-GNN method to learn item transitions over the two graphs respectively, which considered item relations via all sessions. Huang et al. [11] developed a position-aware attention to learn item transition patterns in individual session, and employed a graph hierarchical relation encoder to capture the cross-session item transitions. Pan et al. [18] designed the DGNN method in which the graph structure and the temporal dynamics were considered for learning the dynamic item embeddings. To learn both sequential and non-sequential item transitions, Gwadabe et al. [8] proposed the IC-GAR method to capture the complex transitions between items in the session. Wang et al. [25] presented the SGNN method which can model user's behaviors from spatial and temporal perspectives. However, although these methods focus on pairwise relations between items instead of strictly temporal order in RNNs-based methods, they also ignore complex many-to-many item correlations in the session. In real-world scenarios, an item transition is often influenced by the joint effect of previous items and complex relations among items.

Since hypergraph has inherent way to express complex high-order relations among items, HGNNs-based methods [6, 12] have been favored by many researchers. At present, studies on the topic are just in infancy, and there are a few relevant researches. Bandyopadhyay et al [2] applied graph convolution to hypergraph and devised a line hypergraph convolutional network. Wang et al. [26] adopted hypergraph to represent item correlations and developed a next-item framework based on hypergraph convolutional network. However, these methods are not designed for SBR. After that, Xia et al. [33] constructed the DHCN model for SBR which bridged hypergraph neural network and SBR. The method firstly devised a hypergraph and a line graph to model all sessions by two channels, and then designed a hypergraph convolutional network to capture the complex high-order correlations among items. However, since the method relaxed the strictly temporal dependence between items by connecting items with each other in the session, it might be easily disturbed by irrelevant items in the item representation learning. By contrast, we try to adopt an adaptive transformation function to overcome the problem. Meanwhile, we design a novel context-embedded hypergraph convolution to aggregate context information, which can achieve more informative and adaptive item representations.

2.2. Context-aware Recommendations

While researchers are constantly trying to apply various deep neural networks to SBR, they are also studying how to apply context information to SBR. Since AC-HCN is based on deep neural network, we only introduce the context-aware recommendations using deep neural networks. Current researchers mainly focus on the context-aware RNNs-based methods and GNNs-based methods, while few researchers have studied the context-aware HGNNs-based methods.

In the context-aware RNNs-based methods, Manotumruksa et al. [17] proposed a context-aware GRU recommendation framework to obtain dynamic user preference by adding context attention gate, time gate, and location gate in the GRU unit. Yuan et al. [38] inputted four kinds of context information, including input context, correlation context, static interest context and transition context, into the GRU framework through redefined update gates and reset gates. Wang et al. [24] proposed a recommendation model based on Long Short-Term Memory (LSTM, variation of RNN), which employed time interval and duration information to obtain user's interest. Wu et al. [31] projected context information into a uniform latent vector space and then fused them into the RNN model by means of three combinations including add, stack, and multilayer perception.

In the context-aware GNNs-based methods, Li et al. [14] presented the CA-GGNN method based on context-aware and gated graph neural network. In the method, the session data was represented by the graph structure, and then various context information and session data were incorporated into gated graph neural network. Tang et al. [23] built a time enhanced session graph and then captured user interest shift in the session by the time-based GNN. Feng et al. [5] proposed a context-aware item embedding method to aggregate the auxiliary information from item themselves and their neighborhoods.

The context-aware HGNNs-based methods are in infancy. Peng et al. [19] put forward the GC-HGNN model, which utilized the global and local context information of items to learn user's preference. But the method did not use interaction context information.

In summary, these methods are sufficient to demonstrate that context information can achieve promising performance. However, they fail to fully demonstrate context adaptation. For example, they cannot distinguish the influences of different types of context information on the item representation learning. By contrast, we not only employ the attention mechanism to distinguish the different influences, but also integrate the reversed position embeddings and the time interval embeddings with item representations.

3. The Proposed Model

In this section, the proposed AC-HCN model is presented in detail. In section 3.1, we give the problem formulation. In section 3.2, we overview the framework of AC-HCN. In section 3.3, we introduce session hypergraph. In section 3.4, we describe the context representation based on attention. In section 3.5, we give the details of context-embedded hypergraph convolution. In section 3.6, we depict the joint integration of the position information and the time interval information between items. In section 3.7, we present the representation generation of user interest. In final, the model training and recommendation are introduced in section 3.8.

3.1. Problem Formulation

Let $V = \{v_1, v_2, \cdots, v_N\}$ denote the set of items, where N is the number of items. Let $S = \{s_1, s_2, \cdots, s_M\}$ represent the entire session set, where M is the number of sessions. A session is denoted as $s = [v_1, v_2, \cdots, v_n]$ including interacted items with a chronological order, n is the length of the session. In addition, $T = [T_1, T_2, \cdots, T_K]$ indicates the set of interaction context types, K is the number of types. Here, interaction contexts refer to the interaction scenario information (e.g., time, holiday, week), which are the important parts of context information. Each interaction context type T_k $(1 \le k \le K)$ has a set of context values. Given a session s, there is a corresponding interaction context sequence $C = [C_1, C_2, \cdots, C_n]$, where C_i $(1 \le i \le n)$ is also a sequence including interaction contexts in which user purchases item v_i. Concretely, C_i is defined as $[c_i^1, c_i^2, \cdots, c_i^K]$, where c_i^k $(1 \le k \le K)$

is the context value according to interaction context type T_k. So C_i contains a total of K context values corresponding to K interaction context types. In addition, we also set e_i^v to be a learnable context representation of item v_i in the session s, which is also the part of context information and represents entire properties of the item (e.g., category, price). Since session-based recommendations are designed for anonymous users, the context representation of user is not considered in our model. Therefore, given a session and its relevant context information, the task of our work is to obtain the top-N items that the user is most likely to visit next.

3.2. The Framework of AC-HCN

We illustrate the framework of AC-HCN in Figure 1. At first, all sessions are modeled as a session hypergraph via shared items, and each session is treated as a hyperedge in which all items are connected with each other. Then, the adaptive context-embedded representations of all items in the session hypergraph are learned through the context-embedded hypergraph convolutional layer. Instead of softmax function, we use an adaptive $\alpha-\text{entmax}$ function in the learning to eliminate irrelevant items. Meanwhile, we also distinguish influences of different type of context information in the learning. After that, we integrate the reversed position embeddings and time interval embeddings with item representations by the learnable position matrix and time interval matrix in the position and interval hybrid layer. Next, in the softmax attention layer, for each session, the representation of user interest is obtained by aggregating item representations with different weights in the session. Finally, for each session, we predict the probability of each candidate item being the next click in the prediction layer.

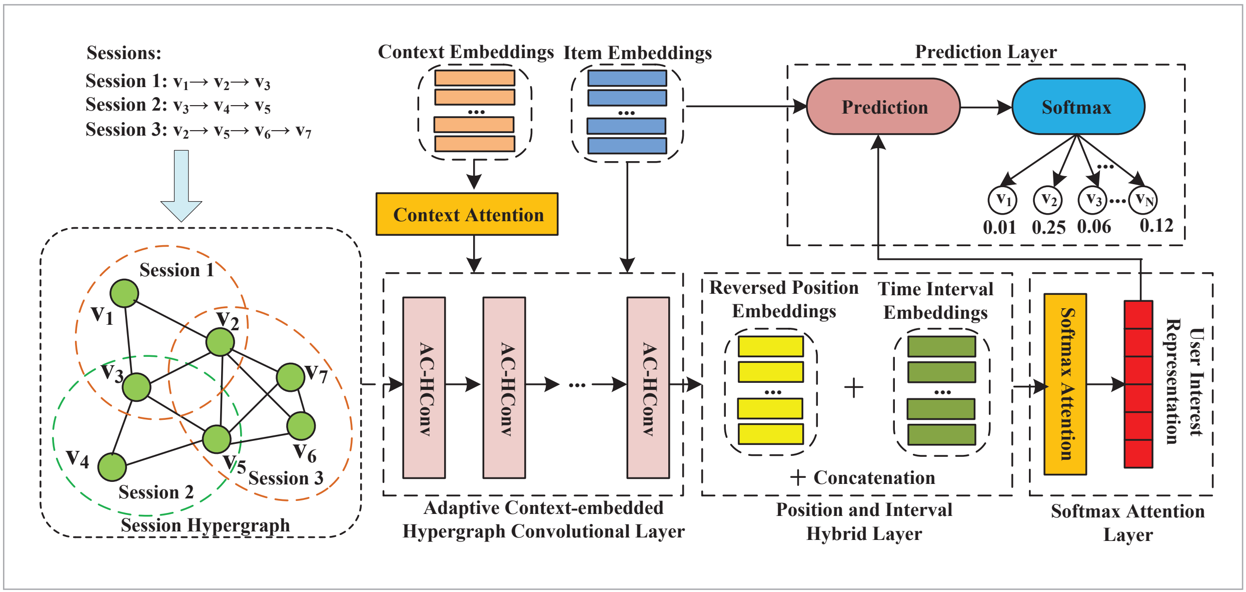
3.3. Session Hypergraph

To capture the complex high-order translations in the session, we refer to literature [33] and adopt the hypergraph to model sessions. Formally, a hypergraph is denoted as $HG=(V,E)$, where V is the set of N unique vertices and E is the set of M hyperedges. Each hyperedge ε E contains two or more vertices. Based on the hypergraph conception, we construct the session hypergraph.

In the session hypergraph, items in all sessions consist of the set of vertices and all sessions comprise the set of hyperedges. Each item in the session is viewed as a vertex in the hyperedge. Instead of sequential dependence, items in each hyperedge are connected with each other in order to better capture the complex translations among items. Different sessions are connected via shard items. The left of Figure 1 shows the

Figure 1

The overview of the proposed AC-HCN model



construction of the session hypergraph. Furthermore, each hyperedge in the session hypergraph is assigned a positive weight and all values construct a diagonal matrix $W \in R^{M \times M}$. An incidence matrix $H \in R^{N \times M}$ is defined to describe the relations between vertices and hyperedges, where, $H_{i\varepsilon} = 1$ if the hyperedge $\varepsilon \in E$ contains vertex $v_i \in V$, otherwise 0.

3.4. Attention-based Context Representation

For each item in the session hypergraph, we adopt the attention mechanism to generate a weighted interaction context representation, which implicitly contains influences of different types of interaction contexts.

Specifically, given the interaction contexts $C_i = \left[c_i^1, c_i^2, \cdots, c_i^K\right]$, corresponding to item v_i in a session. Each context value c_i^k in C_i is converted into a context embedding $e_i^{c,k} \in R^d$ by looking up a learnable parameter matrix corresponding to context type T_k, d is the dimension of embedding. Thus, context embeddings from all values in C_i constitute a context embedding matrix $E_i^c \in R^{d \times K}$. Next, the attention mechanism, defined as Equation (1), is used to generate a weighted vector $\omega_i \in R^K$, in which each element ω_i^k $(1 \le k \le K)$ implies the influence weight of the context types T_k when user interacts item v_i in the session. Based on the weighted vector, a weighted interaction context representation $e_i^c \in R^d$ related with interaction contexts C_i is obtained according to Equation (2).

$$\omega_i = \text{softmax}(q_c^T \tanh(W_c E_i^c)) \tag{1}$$

$$e_i^c = \sum_{k=1}^{K} \omega_i^k e_i^{c,k}, \tag{2}$$

where $q_c \in R^d$ and $W_c \in R^{d \times d}$ are the learnable parameter vector and parameter matrix respectively.

At last, for a given item v_i in the session, the weighted interaction context representation e_i^c and the context representation e_i^v of item v_i are integrated into an overall context representation $e_i^{c,*} \in R^d$ using Equation (3).

$$e_i^{c,*} = \tanh(W_o[e_i^c \,||\, e_i^v] + b_o), \tag{3}$$

where || is the concatenation. $W_o \in R^{d \times 2d}$ and $b_o \in R^d$ are the learnable parameter matrix and parameter vector respectively.

3.5. Context-embedded Hypergraph Convolution

Referring to the hypergraph convolution in literature [33], the basic concept of the hypergraph convolution is defined as Equation (4).

$$x_i^{(l+1)} = \sigma(\sum_{j=1}^{N} \sum_{\varepsilon=1}^{M} H_{i\varepsilon} H_{j\varepsilon} W_{\varepsilon\varepsilon} x_j^{(l)}), \tag{4}$$

where $x_i^{(l)} \in R^{d^{(l)}}$ denotes the representation of item v_i of dimension $d^{(l)}$ in the l-th layer of the hypergraph convolutional network.

As shown in Equation (4), the representation update of each item in the session hypergraph depends on the message aggregation from the current item and its neighborhoods which are connected with the current item. However, context information do not been taken into account in the hypergraph convolution. Based on the thought, we propose a novel adaptive context-embedded hypergraph convolution.

Supposing that the current item v_i in a session is connected with p items, the contributions of its connected items are obtained by Equations (5), (6) and (7).

$$z_{i,j} = q_s^T \sigma(W_s^1[e_i^{c,*} \,||\, x_i] + W_s^2[e_j^{c,*} \,||\, x_j] + b_s^1) \tag{5}$$

$$\beta_i = \alpha - \text{entmax}([z_{i,1}, z_{i,2}, \cdots, z_{i,p}]) \tag{6}$$

$$\alpha = \sigma(W_s^3 e_i^{c,*} + b_s^2) + 1, \tag{7}$$

where || is the concatenation. $z_{i,j}$ is correlation value of the current item v_i and its connected item v_j $(1 \le j \le p)$. $e_i^{c,*}$ and $e_j^{c,*}$ are the overall context representations of item v_i and item v_j. $x_i \in R^d$ and $x_j \in R^d$ are the representations of item v_i and item v_j. $q_s \in R^d$ is the learnable parameter vector, $W_s^1 \in R^{d \times 2d}$, $W_s^2 \in R^{d \times 2d}$ and $W_s^3 \in R^{d \times d}$ are the learnable parameter matrices, $b_s^1 \in R^d$ and $b_s^2 \quad R^d$ are the two biases, $\sigma(\cdot)$ denotes the activation function ReLU.

Note that, we apply $\alpha - \text{entmax}$ function [37] in Equation (6) to replace traditional softmax function. The following reasons are considered: First, in order to obtain the complex relations among items, all items in the session are connected with each other. This assumption is proven to be valid in literature [33], but it may bring some noises. Although softmax function returns the small positive weights for the useless items, these nonzero values may hinder finding the relevant items [37]. By contrast, $\alpha-\text{entmax}$ function tends to yield zero for the unrelated items, thus it has the ability to eliminate noises. Second, since context information involved in the sessions is different, sessions have their own parameter selection modes. The $\alpha-\text{entmax}$ function fittingly provides a way to adaptively learn α, allowing each session to choose reasonable parameter mode based on its context information.

Therefore, the adaptive context-embedded hypergraph convolution in our model is defined as Equation (8).

$$x_i^{(l+1)}=\sigma(e_i^{c,*}\otimes x_i^{(l)}+\sum_{j=1}^{N}\sum_{\varepsilon=1}^{M}H_{i\varepsilon}H_{j\varepsilon}W_{\varepsilon\varepsilon}\beta_i^j\,(e_j^{c,*}\otimes x_j^{(l)})), \quad (8)$$

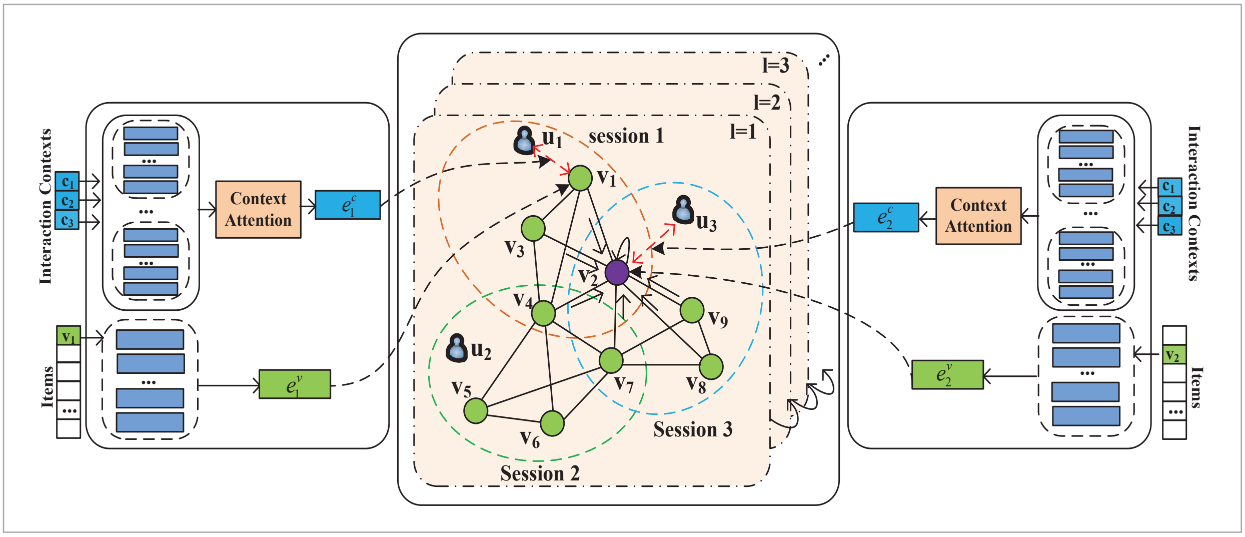
where $x_i^{(l)}$ and $x_j^{(l)}$ are the representations of item v_i and item v_j in the l-th layer of hypergraph convolution. $e_i^{c,*}$ and $e_j^{c,}$ are the overall context representation of item v_i and item v_j. β_i^j is the weight representing the contribution of item v_j to item v_i, which is computed according to Equation (6). If item v_j is not connected with item v_i, the value of β_i^j is 0. $\otimes$ denotes the element-wise product.

Figure 2 shows the demonstration of the adaptive context-embedded hypergraph convolution. Assuming that there are three context types of interaction contexts: hour c_1, week c_2 and holiday c_3, and three sessions: session 1, session 2 and session 3. These three sessions, regarded as three hyperedges, constitute the session hypergraph, and all items in each session are connected with each other. There is overlap between sessions due to the presence of shared items. For example, item v_2 is shared with session 1 and session 3. Without loss of generality, we only pay attention to the representation update of item v_2 in the demonstration. As shown in Figure 2, item v_2 is connected with items v_1, v_3, v_4, v_7, v_8 and v_9, so the representation update of item v_2 is derived from the message aggregation of itself and its connected items, which are shown by seven branches. We annotate these branches with seven arrows in Figure 2.

We only focus on the message aggregation of the branch v_1 to v_2. Assuming that the values of interaction contexts in which user u_3 interacts with current item v_2 in session 3 are [14 clock, Monday, True], and the values of interaction contexts in which user u_1 interacts with item v_1 are [8 clock, Monday, False]. Next, the values [14 clock, Monday, True] are firstly converted into context embeddings by three learnable parameter matrices corresponding to three interaction context type. Then, these context embeddings are translated into a weighted interaction context representation e_2^c using Equations (1) and (2). Considering the context e_2^v of item v_2, the overall context representation $e_2^{c,*}$ is obtained using Equation (3). Similarly, The values [8 clock, Monday, False] are converted and $e_1^{c,*}$ of item v_1 is obtained. After that, the contribu-

Figure 2

The demonstration of adaptive context-embedded hypergraph convolution



tion of v_1 to v_2 in the representation update of item v_2 is computed according to Equations (5), (6) and (7). In the same way, the contributions of all connected items of item v_2 are obtained. Finally, based on these contributions, the representation of item v_2 is updated from seven branches using Equation (8).

3.6. Integration of Positions and Time Intervals

Following literatures [28, 33], the temporal information in the session is also considered in AC-HCN to improve the recommendation performance. We integrate the reversed position embeddings with the item representations by a learnable position matrix $P_r = \left[p_1, p_2, \cdots, p_n\right]$, n is the length of the current session $\left[v_1, v_2, \cdots, v_n\right]$. Unlike the previous studies, we make use of time interval information between adjacent items in the session. Time interval between two adjacent items represents the time duration of user browsing an item. For instance, assuming that item v_1 and item v_2 are adjacent in the session, the time interval between them is the timestamp difference of them, which represents the time duration of user browsing item v_1. To some extent, the more time user spends on an item, the more likely like it.

For computational convenience, we replace the time interval information between two items with the percentage of each item's browsing time over the entire time spent by the user completing the session. Then, we divide the percentage values into 10 uncrossed segments with equal step 10%, denoted as (10*i*%,10(*i*+1)%] $(0 \le i \le 9)$. Each segment represents a type of time interval information, thus a total of 10 types are used to represent the time interval information. For example, user browses an item in a session for 75s, and the entire time of the session is 300s, the percentage of browsing time for the item is 25%, which belongs to the segment (21%,30%], so the browsing time of the item is recorded as the third type of time interval information.

Therefore, we integrate the time interval embeddings with the item representations by a learnable time interval matrix $W_t \in R^{10\times d}$ in which each row corresponds to a type of time interval information. The representation of the i-th item v_i in a session is defined as Equation (9).

$$x_i^* = \tanh(W_f[x_i \,||\, p_{n-i+1} \,||\, g_i] + b_f), \tag{9}$$

where || is the concatenation. $g_i \in R^d$ is the time interval embedding obtained by looking up the time interval matrix W_t. x_i is the learned representation of item v_i using the hypergraph convolutional layer. $W_f \in R^{d\times 3d}$ is the learnable parameter matrix and $b_f \in R^d$ is the bias.

3.7. User Interest Representation

Given a session, following literature [30], we obtain general user interest representation $\varphi_s \in R^d$ using Equations (10)-(11).

$$\lambda_i = q_g^T \cdot \sigma(W_g^1 x_s^* + W_g^2 x_i^* + b_g) \tag{10}$$

$$\varphi_s = \sum_{i=1}^{n} \lambda_i x_i^*, \tag{11}$$

where $x_s^* = \frac{1}{n}\sum_{i=1}^{n} x_i^*$ is session representation computed by averaging the item representations in the session. λ_i is the contribution of item v_i. $\sigma(\cdot)$ is the sigmod function. $W_g^1 \in R^{d\times d}, W_g^2 \quad R^{d\ \ d}$ and $b_g \in R^d$ are the learnable parameters.

3.8. Model Training and Recommendation

Given a session, by computing inner product between the user interest representation φ_s and each item representation x_i^*, we obtain the score vector $\hat{z}$, denoted as Equation (12).

$$\hat{z}_i = \varphi_s^T x_i^*, \tag{12}$$

where $\hat{z}_i$ is the predicted score of the candidate item v_i.

Next, a softmax function is employed to calculate the probabilities $\hat{y}$ of each candidate item being the next one in current session, shown as Equation (13).

$$\hat{y} = \mathrm{softmax}(\hat{z}). \tag{13}$$

In the model training, the cross-entropy loss function is used to optimize objective, shown as Equation (14).

$$\xi(\hat{y}) = -\sum_{i=1}^{m} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i), \tag{14}$$

where y is the one-hot ground truth. The model is trained by minimizing ξ with Adam to get high-quality session-based recommendation.

4. Experiments

We conduct extensive experiments on the real-world datasets to evaluate the performance of our AC-HCN model by answering the following research questions:

- RQ1: Does AC-HCN outperform state-of-the-art baselines in the real-world datasets?
- RQ2: Does the context information affect the performance of AC-HCN?
- RQ3: Are the time interval embeddings useful to AC-HCN?
- RQ4: Dose the $\alpha-\text{entmax}$ function with adaptive α perform better than other transformation functions with fixed α ?
- RQ5: Does different generation patterns of session representation affect the performance of AC-HCN?
- RQ6: How well does AC-HCN perform with different depths of the hypergraph convolution?

4.1. Datasets and Preprocessing

We evaluate the AC-HCN on the real-world datasets with rich context information, i.e. Yoochoose [34] and Diginetica [4]. The former is a public dataset from RecSys Challenge 2015, and the latter is from CIKM Cup 2016. For fair comparison, following literatures [14, 16, 30, 33], we filter out all sessions of length 1 and items appearing less than 5 times in both datasets. For the Yoochoose dataset, we set the sessions of the last day as the test set and the other data as training set. For the Diginetica dataset, the sessions of the last seven days as the test set and the rest as the training set. In addition, following literatures [14, 30, 33, 35], we also generate a lot of new sessions and corresponding labels on both datasets to extend training set and test set by splitting the raw input data. As the Yoochoose dataset is too large to be trained and tested by model, following literatures [14, 30, 35], we employ the most recent fractions 1/64 and 1/4 of the dataset. The statistics of the three datasets are summarized in Table 1.

Table 1
Dataset Statistics

Statistics	Yoochoose 1/64	Yoochoose 1/4	Diginetica
# clicks	557,248	8,326,407	982,961
# training	369,859	5,917,745	719,470
# test	55,898	55,898	60,858
# items	16,766	29,618	43,097
avg. len.	6.16	5.71	5.12

We extract context information from the three datasets and apply them to the AC-HCN model presented in the paper. Concretely, different types of context information can be obtained by using timestamp and other fields on three datasets. For the Yoochoose 1/64 dataset and Yoochoose 1/4 dataset, we collect four types of context information including seven days a week, six time periods in a day, working day or not, and six category types, a total of 504 context values. For the Diginetica dataset, three types of context information are extracted, including seven days a week, six periods in a month, and working day or not. Therefore, there are 84 context values. For three datasets, the time interval between each adjacent items in the session can be calculated by the timestamp difference between the two items, and then the time interval is converted to a type of time interval information, described in the section 3.6. Thus, 10 type values of the time interval information are used in our experiments.

4.2. Baseline Methods

We compare the proposed AC-HCN model with the following representative methods:

- POP and S-POP: they recommend the most frequent items in the training set and the current session respectively.
- GRU4REC [10]: It applies RNN to model user's click sequence in the first time. The samples and ranking-based loss are adopted.
- NARM [15]: It employs RNN with attention mechanism to model user's sequential behaviors.
- STAMP [16]: It utilizes the memory and attention mechanism to capture the user's long-term and short-term interests.
- SR-GNN [30]: It applies the gated graph neural network to capture the user's local and global preferences.
- CA-GGNN[14]: It is a context-aware method which integrates context information into the gated graph neural network.
- TAGNN [35]: It harnesses the power of graph neural network to model item transitions, in which the target-aware attention is used.

Table 2

The performance of AC-HCN compared with other baseline methods on three datasets

Method	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
POP	6.71	1.65	1.33	0.30	0.89	0.20
S-POP	30.44	18.35	27.08	17.75	21.06	13.68
GRU4REC	60.64	22.89	59.53	22.60	29.45	8.33
NARM	68.32	28.63	69.73	29.23	49.70	16.17
STAMP	68.74	29.67	70.44	30.00	45.64	14.32
SR-GNN	70.57	30.94	71.36	31.89	50.73	17.59
TAGNN	71.02	31.12	-	-	51.31	18.03
CA-GGNN	70.84	31.83	72.93	32.91	51.12	18.48
AC-HCN	**71.24**	**32.07**	**73.86**	**33.35**	**51.73**	**18.71**
Improve(%)	0.31	0.75	1.28	1.34	0.82	1.24

4.3. Evaluation Metrics

Following literatures [16, 30], we use P@N (Precision) and MRR@N (Mean Reciprocal Rank) to evaluate the experimental results. P@N evaluates the proportion of correctly recommended items of the top-N items. MRR@N is the average reciprocal ranks, which further considers the position of correctly recommended items in the ranked recommendation list.

4.4. Hyperparameter Setup

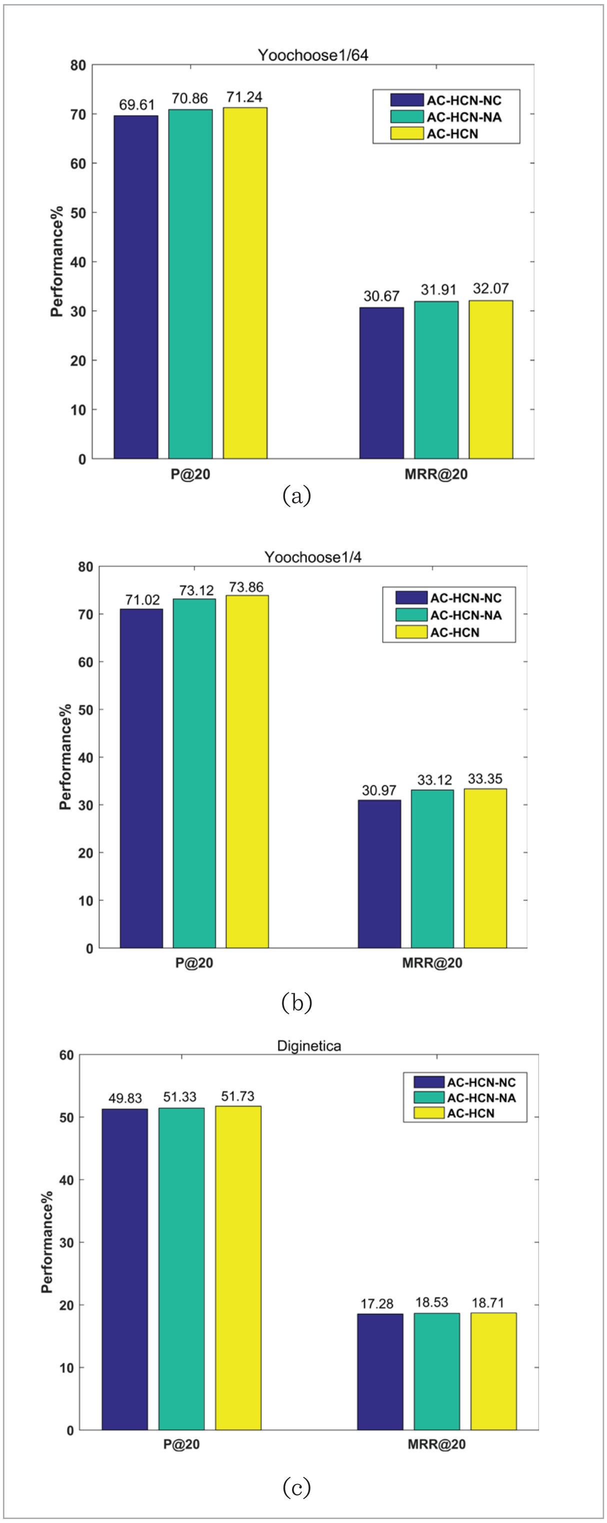
Following literatures [16, 30], for the general setting, the embedding size is 100, the batch size for mini-batch is 100, the L2 penalty is set to 10^{-5} and the iteration number is 30. All parameter vectors and matrices are initialized using the Gaussian distribution with the mean of 0 and the standard deviation of 0.1. The Adam optimizer is used to training parameters, where the learning rate is 0.001, and learning rate decay rate is 0.1. For these baseline methods, we directly use their results reported in the original papers if available, since the same datasets and evaluation settings are used.

4.5. Comparison with Baseline Methods (RQ1)

The experimental results of the performance comparison are reported in Table 2. It can be seen that GRU4REC, NARM, STAMP, SR-GNN, TAGNN, CA-GGNN and AC-HCN, which employ deep learning technique, yield encouraging results. This reflects that the superiority of deep learning technique in SBR. For the RNNs-based methods, NARM and STAMP perform better than GRU4REC. This is because that GRU4REC only use RNN to model user's sequential behaviors without considering the different importance of items in the session, and does not deal with user's interest shift. By contrast, NARM and STAMP not only employ RNN, but also utilize attention mechanism to assign different important values on items in the session. It is proved that the attention mechanism does play a crucial role on capturing user's interest. In addition, benefiting from the integration of user's short-term and long-term interests, STAM has some improvement over NARM. For the GNNs-based methods, SR-GNN, TAGNN and CA-GGNN outperform the RNNs-based methods, which is owed to the great modeling capacity of graph neural networks. Compared to SR-GNN, TAGNN and CA-GGNN further improve recommendation accuracy. TAGNN outperforms SR-GNN by adding the target-ware attention to gated graph neural network, and CA-GGNN performs better than SR-GNN by virtue of context information.

Figure 3

Influence of context information on recommendation



(a)

(b)

(c)

Compared with the best results of baseline methods, our model improves P@20 by about 0.31%, 1.28%, and 0.82%, and MRR@20 by about 0.75%, 1.34%, and 1.24% on three datasets, which benefits from the following three advantages: (1) SR-GNN, TAGNN and CA-GGNN can only capture the pairwise relations between items, while AC-HCN can extract complex many-to-many relations. This is because that AC-HCN models sessions as hypergraph, while comparison methods model sessions as directed graphs. Therefore, AC-HCN is more effective for capturing the complex relations among items. (2) Different from CA-GGNN, which only incorporate context information into gated graph neural network, AC-HCN not only embeds context information into hypergraph convolutional network, but also considers the influences of different types of context information on the item representation learning, which makes item representations have context adaptation. (3) Compared with existing GNNs-based methods, AC-HCN additionally utilizes time interval information in the temporal item sequence. Therefore, our proposed AC-HCN shows overwhelming superiority over all the baseline methods on three datasets.

4.6. Influence of Context Information (RQ2)

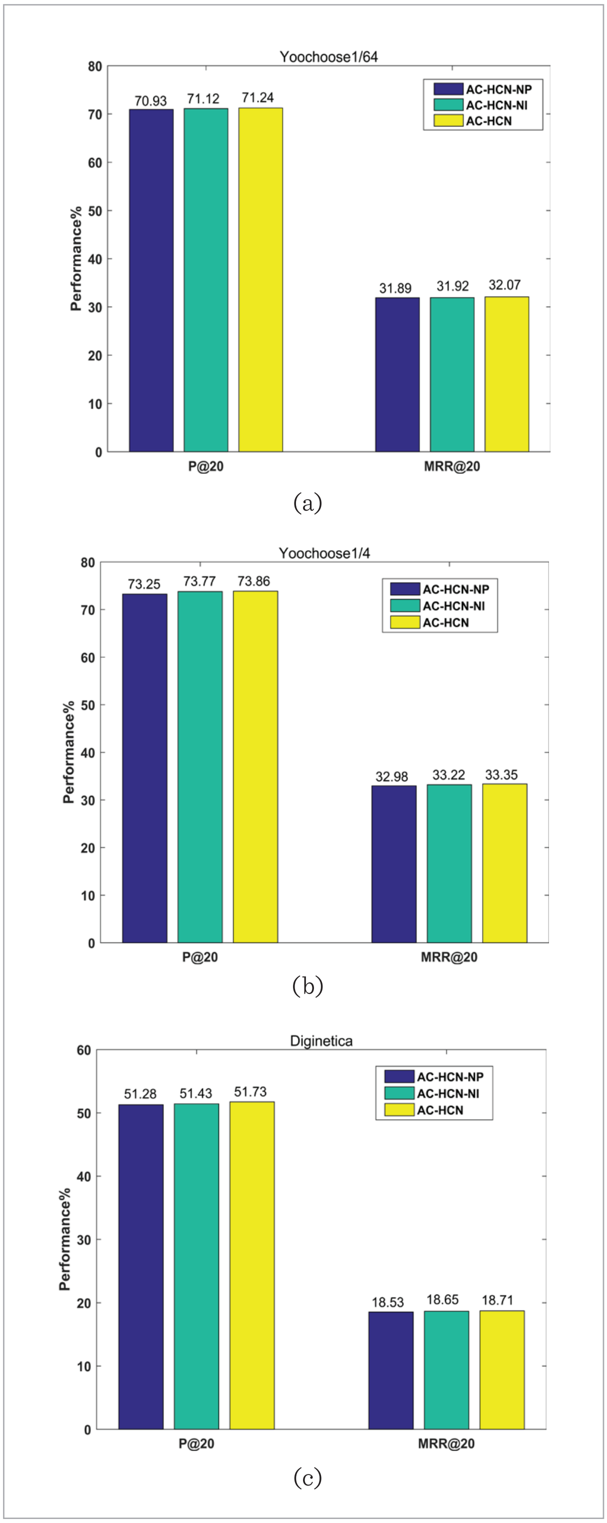
To investigate the contributions of context information and attention-based mechanism, we develop two variants of AC-HCN: AC-HCN-NC and AC-HCN-NA. AC-HCN-NC represents the version without context information, and AC-HCN-NA denotes the version with only context information but not attention-based mechanism. Figure 3 shows the comparison between the two variants with AC-HCN. The results in Figure 3 show that AC-HCN achieves the best performance among the three models. Specifically, on the one hand, compared with AC-HCN-NC, AC-HCN improves P@20 by about 2.34%, 3.99%, and 3.81%, and MRR@20 by about 4.56%, 7.68%, and 8.28% on three datasets. These improvements indicate that context information do have play an crucial role in the session-based recommendations. This is because that context information contains rich semantic information, which can help to model user behaviors. On the other hand, compared with AC-HCN-NA, AC-HCN improves P@20 by about 0.54%, 1.01%, and 0.78%, and MRR@20 by about 0.50%, 0.69%, and 0.97% on three datasets. These improvements illustrate that different types of context information do have distinguishing influences on the learning of item representations.

4.7. Influence of Time Interval Information(RQ3)

In order to demonstrate the contribution of time interval information in AC-HCN, we develop two variants of AC-HCN: AC-HCN-NP and AC-HCN-NI. AC-HCN-NP represents the version with only the time interval information but not the reversed position information, and AC-HCN-NI represents the version with only the reversed position information but not the time interval information. We compare the two variants with the full AC-HCN on three datasets. As can be observed in Figure 4, the contributions of both components are obvious. On the one hand, compared with AC-HCN-NP, AC-HCN improves P@20 by about 0.44%, 0.83%, and 0.88%, and MRR@20 by about 0.56%, 1.12%, and 0.97% on three datasets. The results demonstrate that the position factors in the session play a crucial role in improving recommendation accuracy. Although the GNNs-based methods relax the temporal relations between items, appropriate consideration of the temporal relations between items can help to accurately model user behaviors. This is why many previous researches adopt this strategy. On the other hand, compared with AC-HCN-NI, AC-HCN improves P@20 by about 0.17%, 0.12%, and 0.58%, and MRR@20 by about 0.47%, 0.39%, and 0.32% on three datasets. These findings validate that the time interval information between items in the session implies user's interest, and can be used to improve recommendation performance to some extent.

Figure 4

Influence of time interval on recommendation



(a)

(b)

(c)

4.8. Influence of Adaptive Transformation Function(RQ4)

In order to demonstrate the utility of $\alpha-\text{entmax}$ function using adaptive α, we carry out the comparison experiments using adaptive α and fixed α on three datasets. We range the values of α to 1, 1.2, 1.4, 1.6, and 1.8, where $\alpha=1$, the function is softmax. Table 3 illustrates the performance of AC-HCN with varied α on three datasets. As the observation from Table 3, the performances of AC-HCN with different fixed α are similar, and the performance of AC-HCN with adaptive α is better than all AC-HCN with fixed α. It is proved that each session has its own best α,

Table 3

The performance of AC-HCN with different transformation functions on three datasets

Function	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
softmax (α=1)	71.13	31.84	73.75	33.22	51.68	18.69
α=1.2	71.11	32.01	73.71	33.29	51.61	18.64
α=1.4	71.09	32.03	73.67	33.12	51.59	18.61
α=1.6	71.12	31.88	73.77	33.30	51.67	18.65
α=1.8	71.14	31.98	73.72	33.28	51.64	18.66
α-entmax	**71.24**	**32.07**	**73.86**	**33.35**	**51.73**	**18.71**

depending on the context information in the session. Meanwhile, when an item is unrelated with the current item, $\alpha-\text{entmax}$ function tends to yield zero for the contribution of the item. Therefore, AC-HCN with adaptive α performs better.

4.9. Influence of Aggregation Patterns (RQ5)

Since user interest representation is closely related to session representation, it is meaningful to compare AC-HCN performance with different generation patterns of session representation. We consider three strategies, i.e., mean pooling, max pooling and concatenation mechanism.

For mean pooling, the session representation is achieved by averaging every dimension value of each item representation in the session, and the k-th dimension of the session representation $x^*_{s,k}$ is computed by Equation (15).

$$x^*_{s,k} = \text{mean}(x^*_{i,k})\ . \tag{15}$$

For max pooling, the maximum value of every dimension for each item representation in the session is taken, denoted as Equation (16).

$$x^*_{s,k} = \max(x^*_{i,k})\ . \tag{16}$$

For concatenation, the session representation is the joint of each item representation in the session, denoted as Equation (17), where $W_h \in R^{d\times nd}$ is the transform parameter matrix.

$$x^*_s = W_h([x^*_1 \,\|\, x^*_2 \,\|\cdots\|\, x^*_n])\ . \tag{16}$$

Table 4 shows the performance of AC-HCN with different generation patterns of session representation on three datasets. It can be observed that AC-HCN with mean pooling outperforms AC-HCN with other generation patterns on three datasets in terms of P@20 and MRR@20. The performance of AC-HCN with max pooling is the worst on three datasets. It may be because that session representation is the

Table 4

The performance of AC-HCN with different generation patterns of session representation on three datasets

Measures	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
Max Pooling	70.89	31.77	73.13	31.97	50.55	17.85
Concatenation	71.21	32.01	73.82	33.25	51.68	18.69
Mean Pooling	**71.24**	**32.07**	**73.86**	**33.35**	**51.73**	**18.71**

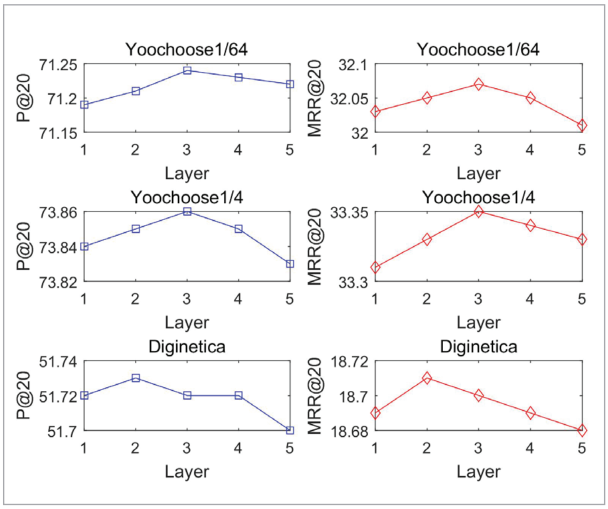
general characteristic of the session, and max pooling is not unsuitable for representing the session. Despite of using additional parameters, the performance of AC-HCN with concatenation is also worse than AC-HCN with mean pooling, possibly because too many parameters may lead to over-fitting. Therefore, mean pooling is used in our model.

4.10. Influence of Model Depth (RQ6)

To study the influence of the depth of hypergraph convolutional layer, we set the numbers of the layer within 1 to 5. Figure 5 shows the influence on performance when the layer takes different depths on three databases. The x-axis represents the different depths of the layer, and the y-axis represents the P@20 or MRR@20. From the experimental results, when layer depth is 3, the model achieves the best performance on Yoochoose dateset. When layer depth is 2, the model reaches the optimum state on Diginetica dataset. In addition, with the depth of layer increases, the performance of AC-HCN drops. The possible cause could be the over-fitting of the model.

Figure 5

Influence of model depth of hypergraph convolution



5. Conclusion

Existing GNNs-based methods for SBR only focus on the pairwise relations without considering the influence of context information, which leads to the failure to obtain the high-order relations among items and context adaptation. In the paper, we propose the AC-HCN model for SBR to address the problems. The model uses a context-embedded hypergraph convolutional network to learn the item representations. In learning of item representations, the model fully considers the complex relations among items using the session hypergraph, and applies $\alpha-\text{entmax}$ function to eliminate irrelevant items. Meanwhile, various kinds of context information with different contributions are considered. Moreover, to further enhance the performance of our model, we also use time interval information between items in the model training. Extensive experimental results on three real-world datasets demonstrate our AC-HCN model is superior to other advanced methods.

Acknowledgement

This research is supported by the Science & Technology Research Project of Henan Province (No.222102210140).

References

1. Adomavicius, G., Tuzhilin, A. Context-aware Recommender Systems. Proceedings of the 2008 ACM Conference on Recommender Systems, (RecSys 2008), Lausanne, Switzerland, October 23-25, 2008, 335-336. https://doi.org/10.1145/1454008.1454068
2. Bandyopadhyay, S., Das, K., Murty, M. N. Line Hypergraph Convolution Network: Applying Graph Convolution for Hypergraphs. arXiv, 2020. https://doi.org/10.48550/arXiv.2002.03392
3. Bonchi, F., Perego, R., Silvestri, F., Vahabi, H., Venturini, R. Recommendations for the Long Tail by Term-query Graph. Proceedings of the 20th International Conference Companion on World Wide Web, (WWW 2011), Hyderabad, India, March 28-April 1, 2011, 15-16. https://doi.org/10.1145/1963192.1963201
4. Diginetica Dataset. https://competitions.codalab.org/competitions/11161

5. Feng, C., Shi, C., Liu, C., Zhang, Q., Hao, S., Jiang, X. Context-aware Item Attraction Model for Session-based Recommendation. Expert Systems with Applications, 2021, 176, 114834. https://doi.org/10.1016/j.eswa.2021.114834

6. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y. Hypergraph Neural Networks. Proceedings of the 33rd AAAI Conference on Artificial Intelligence, (AAAI 2019), Honolulu, HI, USA, January 27-February 1, 2019, 33(1), 3558-3565. https://doi.org/10.1609/aaai.v33i01.33013558

7. Guo, L., Yin, H., Wang, Q., Chen, T., Zhou, A., Hung, N. Q. V. Streaming Session-based Recommendation. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD 2019), Anchorage AK, USA, August 4-8, 2019, 1569-1577. https://doi.org/10.1145/3292500.3330839

8. Gwadabe, T. R., Liu, Y. IC-GAR: Item Co-occurrence graph Augmented Session-based Recommendation. Neural Computing and Applications, 2022, 34, 7581-7596. https://doi.org/10.1007/s00521-021-06859-x

9. He, X., Zhang, H., Kan, M. Y., Chua, T. S. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2016), Pisa, Italy, July 17-21, 2016, 549-558. https://doi.org/10.1145/2911451.2911489

10. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D. Session-based Recommendations with Recurrent Neural Networks. arXiv, 2016. https://doi.org/10.48550/arXiv.1511.06939

11. Huang, C., Chen, J., Xia, L., Xu, Y., Dai, P., Chen, Y., Bo, L., Zhao, J., Huang, J. X. Graph-Enhanced Multi-Task Learning of Multi-Level Transition Dynamics for Session-based Recommendation. Proceedings of the 35th AAAI Conference on Artificial Intelligence, (AAAI 2021), Online, February 2-9, 2021, 35(5), 4123-4130. https://doi.org/10.1609/aaai.v35i5.16534

12. Jiang, J., Wei, Y., Feng, Y., Cao, J. Gao, Y. Dynamic Hypergraph Neural Networks. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China, August 10-16, 2019, 2635-2641.https://doi.org/10.24963/ijcai.2019/366

13. Koren, Y., Bell, R., Volinsky, C. Matrix Factorization Techniques for Recommender Systems. Computer, 2009, 42(8), 30-37. https://doi.org/10.1109/MC.2009.263

14. Li, D., Gao, Q. Session Recommendation Model Based on Context-Aware and Gated Graph Neural Networks. Computational Intelligence and Neuroscience, 2021, 7266960. https://doi.org/10.1155/2021/7266960

15. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J. Neural Attentive Session-based Recommendation. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, (CIKM 2017), Singapore, Singapore, November 6-10, 2017, 1419-1428. https://doi.org/10.1145/3132847.3132926

16. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD 2018), London, UK, August 19-23, 2018, 1831-1839. https://doi.org/10.1145/3219819.3219950

17. Manotumruksa, J., Macdonald, C., Ounis, I. A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation. Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2018), Ann Arbor, MI, USA, July8-12, 2018, 555-564. https://doi.org/10.1145/3209978.3210042

18. Pan, Z., Chen, W., Chen, H. Dynamic Graph Learning for Session-Based Recommendation. Mathematics, 2021, 9(12), 1420. https://doi.org/10.3390/math9121420

19. Peng, D., Zhang, S. GC-HGNN: A Global-context Supported Hypergraph Neural Network for Enhancing Session-based Recommendation. Electronic Commerce Research and Applications, 2022, 52, 101129. https://doi.org/10.1016/j.elerap.2022.101129

20. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. Factorizing Personalized Markov Chains for Next-basket Recommendation. Proceedings of the 19th International Conference on World Wide Web, (WWW 2010), Raleigh, North Carolina, USA, April 26-30, 2010, 811-820. https://doi.org/10.1145/1772690.1772773

21. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web, (WWW 2001), Hong Kong, China, May 1-5, 2001, 285-295. https://doi.org/10.1145/371920.372071

22. Sheng, Z., Zhang, T., Zhang, Y. HTDA: Hierarchical Time-based Directional Attention Network for Sequential User Behavior Modeling. Neurocomputing, 2021, 441, 323-334. https://doi.org/10.1016/j.neucom.2021.02.006

23. Tang, G., Zhu, X., Guo, J., Dietze S. Time Enhanced Graph Neural Networks for Session-based Recommendation. Knowledge-Based Systems. 2022, 251, 109204. https://doi.org/10.1016/j.knosys.2022.109204

24. Wang, D., Xu, D., Yu, D., Xu, G. Time-aware Sequence Model for Next-item Recommendation. Applied Intelligence, 2021, 51, 906-920. https://doi.org/10.1007/s10489-020-01820-2

25. Wang, H., Zeng, Y., Chen, J., Zhao, Z., Chen, H. A Spatiotemporal Graph Neural Network for Session-based Recommendation. Expert Systems with Applications. 2022, 202, 117114. https://doi.org/10.1016/j.eswa.2022.117114

26. Wang, J., Ding, K., Hong, L., Liu, H., Caverlee, J. Next-item Recommendation with Sequential Hypergraphs. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2020), Virtual Event, China, July 25-30, 2020, 1101-1110. https://doi.org/10.1145/3397271.3401133

27. Wang, Y., Guo, C., Chu, Y., Hwang, J., Feng, C. A Cross-domain Hierarchical Recurrent Model for Personalized Session-based Recommendations. Neurocomputing, 2020, 380, 271-284. https://doi.org/10.1016/j.neucom.2019.11.013

28. Wang, Z., Wei, W., Cong, G., Li, X., Mao, X., Qiu, M. Global Context Enhanced Graph Neural Networks for Session-based Recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2020), Virtual Event, China, July 25-30, 2020, 169-178. https://doi.org/10.1145/3397271.3401142

29. Wu, S., Liu, Q., Wang, L., Tan, T. Contextual Operation for Recommender Systems. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(8), 2000-2012. https://doi.org/10.1109/TKDE.2016.2562621

30. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T. Session-based Recommendation with Graph Neural Networks. Proceedings of the 33rd AAAI Conference on Artificial Intelligence, (AAAI 2019), Honolulu, HI, USA, January 27-February 1, 2019, 33(1), 346-353. https://doi.org/10.1609/aaai.v33i01.3301346

31. Wu, T., Sun F., Dong, J., Wang, Z., Li, Y. Context-aware Session Recommendation Based on Recurrent Neural Networks. Computers and Electrical Engineering, 2022, 100, 107916. https://doi.org/10.1016/j.compeleceng.2022.107916

32. Wu, X., Liu, Q., Chen, E., He, L., Lv, J., Cao, C., Hu, G. Personalized Next-song Recommendation in Online Karaokes. Proceedings of the 7th ACM Conference on Recommender Systems, (RecSys 2013), Hong Kong, China, October 12-16, 2013, 137-140. https://doi.org/10.1145/2507157.2507215

33. Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., Zhang, X. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. Proceedings of the 35th AAAI Conference on Artificial Intelligence, (AAAI 2021), Online, February 2-9, 2021, 35(5), 4503-4511. https://doi.org/10.1609/aaai.v35i5.16578

34. Yoochoose Dataset. http://2015.recsyschallenge.com/challege.html

35. Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., Tan, T. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2020), Virtual Event, China, July 25-30, 2020, 1921-1924. https://doi.org/10.1145/3397271.3401319

36. Yu, S., Yang, M., Qu, Q., Shen, Y. Contextual-boosted Deep Neural Collaborative Filtering Model for Interpretable Recommendation. Expert Systems with Applications, 2019, 136, 365-375. https://doi.org/10.1016/j.eswa.2019.06.051

37. Yuan, J., Song, Z., Sun, M., Wang, X., Zhao, W. X. Dual Sparse Attention Network for Session-based Recommendation. Proceedings of the 35th AAAI Conference on Artificial Intelligence, (AAAI 2021), Online, February 2-9, 2021, 35(5), 4635-4643. https://doi.org/10.1609/aaai.v35i5.16593

38. Yuan, W., Wang, H., Yu, X., Liu, N., Li, Z. Attention-based Context-aware Sequential Recommendation Model. Information Sciences. 2020, 510, 122-134. https://doi.org/10.1016/j.ins.2019.09.007

39. Zhang, J., Ma, C., Zhong, C., Mu, X., Wang, L. MBPI: Mixed Behaviors and Preference Interaction for Session-based Recommendation. Applied Intelligence, 2021, 51, 7440-7452. https://doi.org/10.1007/s10489-021-02284-8

This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).