Dual-Channel Graph Contextual Self-Attention Network for Session-Based Recommendation

Teng Huang[1], Huiqun Yu[1,2(✉)], and Guisheng Fan[1(✉)]

[1] Department of Computer Science and Engineering,
East China University of Science and Technology, Shanghai, China
{yhq,gsfan}@ecust.edu.cn
[2] Shanghai Key Laboratory of Computer Software Evaluating and Testing,
Shanghai, China

Abstract. The session-based recommendation task is a key task in many online service websites (such as online music, e-commerce, etc.). Its goal is to predict the user's next possible interactive item based on an anonymous user behavior sequence. However, existing methods do not take into account the information implicit in similar neighbor sessions. This paper proposes a new dual-channel graph neural network combined with a self-attention network model. In this model, the graph neural network is used to model the session sequence, and the learning is different through two independent channels between and within the session. The model uses the attention network to learn the weights of different items for the recommendation results. Several experiments are done on two public e-commerce data sets and the results show that the performance of the model proposed in this paper is better than the general methods.

Keywords: Session-based recommendation · Graph neural network · Self-attention network

1 Introduction

With the development of information technology and Internet technology, people have gradually entered the era of information explosion, and the problem of information overload has become one of the urgent problems. In this era, personalized recommendation system came into being [13]. The traditional recommendation system relies on collecting the user's personal information or historical operation records to make recommendations. However, in many network service scenarios (such as online music, online movie websites, e-commerce, etc.), due to privacy protection and other considerations, users may not log in to the website when obtaining network services. To solve such problems, session-based recommendation system has been proposed to predict the next possible interaction (such as clicking, etc.) items [4, 19].

The definition of session in this field is similar to the general concept of web. In general, the session sequence in the session-based recommendation task refers to a sequence of items composed of all items of user interaction within a period of time, such as a few minutes or a few hours, arranged according to time. For example, the

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 244–258, 2021.
https://doi.org/10.1007/978-3-030-92635-9_15

products that a user browses and clicks on an e-commerce website within one hour constitute a session sequence.

Considering the extremely high value of session-based recommendations, various methods have been proposed. Traditionally, the method based on the Markov chain [9] will predict the user's next behavior based on the previous behavior. In recent years, researchers have introduced neural network models into session-based recommendation tasks and have achieved better results [17]. For instance, GRU4REC [4] uses GRU units to model session sequences. NARM [6] designed a global and local encoder to capture the sequential behavior of the session sequence and the main purpose of user. SR-GNN [15] model the session sequence into a session graph, and use the graph neural network method to model the complex conversions between separate items.

Although the above methods have achieved good recommendation results, these models also have some problems. For example, GRU4REC only considers the sequential behavior information of the session sequence, and does not consider the main purpose of the user. NARM can only model the sequential relationship between consecutive items, ignoring the complex conversion between different items, and these conversions reflect the user's behavior pattern, which is of great significance to the recommendation system. SR-GNN does not consider the similar item conversion implicit in the neighbor sessions, and the hidden layer representation vectors of the item node may not be accurate enough.

In order to further improve the accuracy of the recommendation system, this paper proposes a new model, a dual-channel graph contextual self-attention network for session-based recommendation, which is called DCGC-SAN for short. This model takes into account the differences in item conversion between the current session and neighbor sessions, and uses two channels, intra-session and inter-session channels, to respectively propagate the embedding vectors of the learning graph nodes, then aggregates the features of the two channels through a fusion function, and combines the self-attention network to learn contextual information as the main purpose of the user. Finally, the user's short-term interest and the main purpose are combined to calculate the session representation vector, which is finally used to generate the TOP-N recommendation list.

The main contributions of this paper are summarized as follows:

- To better learn the embedding vectors of graph nodes, a new DCGC-SAN is proposed, which divides the intra-session and inter-session channels learning of accurate graph node embedding vectors.
- The self-attention network is applied to capture the main purpose of the user.
- A large number of experiments and analyses conducted on two public e-commerce data sets show that DCGC-SAN is better than baseline methods in terms of recommendation accuracy.

The rest of this paper is structured as follows. We review related researches and works in Sect. 2. Section 3 presents the proposed method of our model. The experiments and analysis are presented in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Related Work

2.1 Neural Network Model

Hidasi et al. [4] first proposed that a recurrent neural network with GRU units should be used to extract the features of session sequences. This is the first time that someone introduced a recurrent neural network into session-based recommendation tasks. In order to improve the basic RNN model, Tan et al. [10] used sequence preprocessing, and data enhancement to reduce the occurrence of overfitting and improve the performance of the RNN model. After that, researchers hope to use the implicit collaboration information in similar sessions. For example, KNN-RNN proposed by Jannach et al. [5] uses a heuristic nearest neighbor (KNN) method to sample the nearest neighbors and introduces GRU4REC to capture the sequential relationship. The model based on the graph neural network considers the complex conversion between all items in the item space and changes the pattern dominated by the recurrent neural network models. The most important point of the method based on graph neural networks is the method of graph node embedding vector. Typical methods such as gated graph neural network (GGNN) [7] extend GNN to sequence output. SR-GNN proposed by Wu et al. [15] modeled the session sequence into a session graph, and used the GGNN method to learn intricate item conversions. Recently, inspired by the success of convolution neural networks in the image field, Bruna et al. [2] proposed Spectrogram Convolution Neural Networks (GCN). Wu et al. [14] found that there are many unnecessary calculations in GCN. GCN can be simplified by removing non-linearity and folding the weight matrix to obtain higher efficiency without compromising accuracy.

2.2 Attention Mechanism

The attention mechanism is commonly used in fields such as natural language processing and computer vision. In session-based recommendation tasks, standard soft attention mechanisms have been studied [6], such as NARM and STAMP. Vaswani et al. [11] proposed a self-attention-based model, Transformer, which is a network without any CNN or RNN units. It can model the dependencies between different words and achieve sota performance on machine translation tasks. Anh et al. [1] introduced the Transformer to the session-based recommendation task, which took less time on the premise of achieving better recommendation performance. Xu et al. [16] learned local context information through the attention network and also achieved impressive performance improvements.

3 Proposed Method

3.1 Problem Statement

The goal of the session-based recommendation system is to predict the user's next most likely interactive item based on the current user's interaction sequence. Let $V = \{v_1, v_2, v_3, ..., v_m\}$ denote the collection of items that have appeared in all session sequences. For each session sequence $S = \{v_{s,1}, v_{s,2}, v_{s,3}, ..., v_{s,n}\}$, $v_{s,i} \in V$ represents

the user's interactive items in the current session sequence. The goal of the recommendation model is to predict the next possible interactive item $v_{s,i+1}$ for current user. For the session sequence S, the model outputs the probability of all items $\hat{y}$ as the next user interacting item, and the TOP-N recommendation list generated for the user is the TOP-N with the largest score.

3.2 Model Overview

DCGC-SAN consists of three parts. The first part is the graph neural network. The main task of this part is to model the current session sequence and neighbor session sequences, divide different channels to learn accurate graph node embedding vectors. The second part is the self-attention network. The principal task of this part is to capture the main purpose of the user and incorporate rich contextual information into the model. The third part is the prediction layer, which combines the previously generated session sequence representation vector and the graph node embedding vector to calculate the possible interaction probability of all items in the item space. The architecture of DCGC-SAN is shown in Fig. 1.

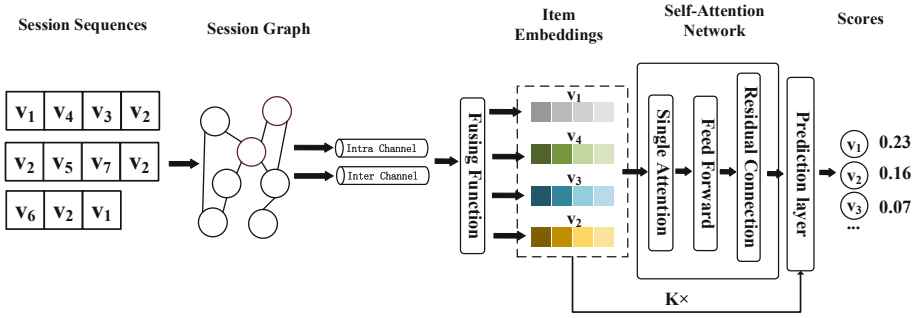

Fig. 1. DCGC-SAN architecture

3.3 Session Graph Construction

First, we need to construct a session graph. For the current session sequence S, extract the n neighbor sessions that are most similar to S. In order to reduce the complexity of calculating the similarity between session sequences, we use the number of repetitive items in different sessions as a measure of similarity. Sort all the session sequences in reverse order, and sample the n session sequences at the top to form a neighbor session set N_s. Model the current session and neighbor session sets as session graph ζ_s. Each node in the session graph represents an item v_i, and each edge (v_i, v_j) represents that the user interacts with the item v_j after interacting with the item v_i in the current session or neighbor sessions.

Figure 2 is the complete process of constructing a session graph, in which the red arrow represents the item conversions in the current session, and the black arrow represents the item conversions in the neighbor sessions.

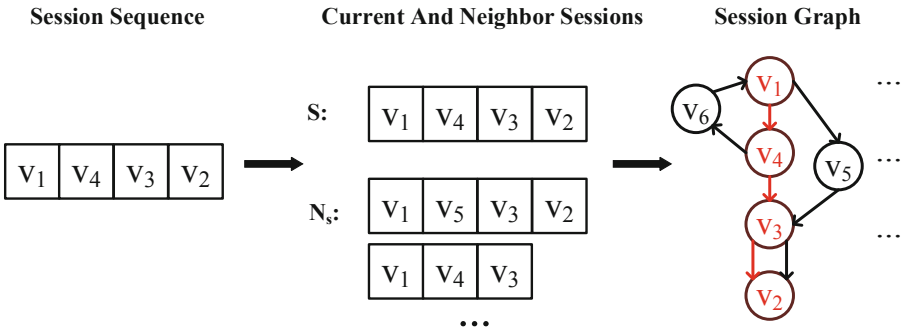

Fig. 2. The construction process of the session graph

3.4 Item Embedding Learning

After the construction of the session graph is completed, the follow-up task is to learn accurate item node embedding vectors from the session graph by graph neural network. The difference between the GGNN used in the DCGC-SAN and SR-GNN is that we use two channels: intra-session and inter-session channels to aggregate the item conversion in the current session and the neighbor sessions. We think different conversion has different meanings. The former reflects the current interests and preferences of users, and the latter reflects the conversions between global items. The embedding vectors of the learning graph nodes are aggregated within these two channels, and the computational complexity is simplified according to the simplified graph convolution calculation method in the SGCN [14].

$$\mathbf{v}_i^{(k)\prime} = \frac{1}{d_i + 1}\mathbf{v}_i^{(k-1)} + \sum_{v_j \in V_s} \frac{a_{ij}}{\sqrt{(d_i + 1)(d_j + 1)}}\mathbf{v}_j^{(k-1)} \tag{1}$$

$$\mathbf{v}_i^{(k)\prime\prime} = \frac{1}{d_i + 1}\mathbf{v}_i^{(k-1)} + \sum_{v_j \in V_{N_s}} \frac{a_{ij}}{\sqrt{(d_i + 1)(d_j + 1)}}\mathbf{v}_j^{(k-1)} \tag{2}$$

Formulas (1) and (2) are two channel learning and node update formulas, where d_i is the degree of the node v_i in the adjacency matrix. a_{ij} indicates whether there is an edge between v_i and v_j in the session graph, if there is an edge, $a_{ij} = 1$, if there is no edge, $a_{ij} = 0$.

Finally, the item graph node embedding vector set learned through the intra-session channel is embedded in the vector set $\mathbf{V}_s = \{\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3, ..., \mathbf{v}'_n\}$, where $v'_i = v_i^{(k)\prime}$; the item graph node embedding learned through the inter-session channel represents the vector set $\mathbf{V}_{N_s} = \{\mathbf{v}''_1, \mathbf{v}''_2, \mathbf{v}''_3, ..., \mathbf{v}''_n\}$, where $v''_i = v_i^{(k)\prime\prime}$.

$$\mathbf{v}_i = \text{item\_fusing}(\mathbf{v}'_i, \mathbf{v}''_i) \tag{3}$$

According to formula (3), we aggregate the information in the two channels through the item fusing function, extract effective features, and learn accurate graph node embedding representation vectors. The two vector aggregation methods are used in the following three methods in this article:

- Average pooling (avg): fusion of information from different channels by taking the average of two vectors in each dimension.
- Maximum pooling (max): fusion of information from different channels by taking the maximum value of two vectors in each dimension
- Direct connection (concat): connect two vectors and reduce the dimensionality through a single-layer linear neural network, where the parameter $\mathbf{W} \in \mathbb{R}^{2d \times d}$.

$$\mathbf{v}_i = \sigma(\mathbf{W}[\mathbf{v}_i', \mathbf{v}_i'']) \tag{4}$$

3.5 Self-attention Network

DCGC-SAN uses a self-attention network to model the main purpose of the current session sequence. The self-attention mechanism is a special attention mechanism. It captures the correspondence between the entire input and output by calculating the conversion weight between each item and all other items in the sequence, which can effectively solve the problem of long-distance items. It is easy to lose the problem of relying on information [11].

For the input session sequence $S = \{v_{s,1}, v_{s,2}, v_{s,3}, ..., v_{s,n}\}$, the d-dimensional embedding vector $\mathbf{S} = [\mathbf{v}_{s,1}, \mathbf{v}_{s,2}, \mathbf{v}_{s,3}, ..., \mathbf{v}_{s,n}]$ is obtained after the embedding layer, and input into the self-attention network. The self-attention network is composed of a single-layer attention layer, a feedforward neural network, and a residual connection layer.

The main function of the single-layer attention layer is to obtain the weight relationship between each item and other items in the session sequence, so that when modelling a single item, the information of other items in the same session sequence can be taken into account, which introduces rich context information to the entire model. The calculation formula of the single attention layer is shown in formula (5).

$$\mathbf{E} = \text{softmax}\left(\frac{(\mathbf{S}\mathbf{W}^Q)(\mathbf{S}\mathbf{W}^K)^T}{\sqrt{d}}\right)(\mathbf{S}\mathbf{W}^V) \tag{5}$$

The weight matrices $\mathbf{W}^Q$,$\mathbf{W}^K$ and $\mathbf{W}^V \in \mathbb{R}^{2d \times d}$, $\mathbf{E}$ represents the output of the single-layer attention layer.

The main function of the feedforward neural network is to introduce a nonlinear transformation to the attention network, so that the attention network can obtain information of different spatial dimensions. In order to solve the problems such as the disappearance of the gradient caused by the network level being too deep, a new residual network layer is added after the feedforward neural network [3]. The calculation formulas of the feedforward neural network layer and the residual layer are shown in formula (6).

$$\mathbf{O} = \text{Relu}(\mathbf{E}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 + \mathbf{E} \tag{6}$$

$\mathbf{W}_1$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the d-dimensional bias vectors, and the integration of the previous three layers is defined as:

$$\mathbf{O} = \text{F}(\mathbf{S}) \tag{7}$$

In order to study the most suitable attention network layers for the user's main purpose in modeling session sequences, we repeatedly apply attention layers in the model to form attention networks to capture different types of features. The first layer is defined as $\mathbf{O}^{(1)} = \text{F}(\mathbf{S})$, and the kth layer is defined as:

$$\mathbf{O}^{(k)} = \text{F}(\mathbf{O}^{(k-1)}) \tag{8}$$

Take the nth row vector of the matrix $\mathbf{O}^{(k)}$ as the output of the attention network and at the same time as the main purpose of the user in the session sequence.

$$\mathbf{C}_t^l = \mathbf{O}_{\text{n}}^{(k)} \tag{9}$$

3.6 Prediction Layer

In order to better predict the user's possible next interactive item, we use the embedding vector of the user's last interactive item in the current session sequence as the user's short-term interest $\mathbf{C}_t^g$:

$$\mathbf{C}_t^g = \mathbf{v}_{s,n} \tag{10}$$

Combining the user's main purpose $\mathbf{C}_t^l$ calculated by the attention network layer, fusion of the two information forms the final session representation vector $\mathbf{C}_t$.

$$\mathbf{C}_t = \sigma(\mathbf{W}_g\mathbf{C}_t^g + \mathbf{W}_l\mathbf{C}_t^g) \tag{11}$$

Then calculate the probability $\hat{y}_i$ of each candidate item $v_i \in V$ as the user's next interactive item. The calculation formula is as follows:

$$\hat{y}_i = \text{softmax}(\mathbf{C}_t^T\mathbf{v}_i) \tag{12}$$

$\mathbf{v}_i$ is the embedding layer representation vector of the item v_i. Finally, the model is trained by minimizing the cross-entropy loss function.

$$Loss(y_i, \hat{y}_i) = -\sum_{i=1}^{n} (y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \tag{13}$$

y_i represents the item that the user actually interacts with.

4 Experiments and Analyses

In this section, we first prepare the data needed for the experiments, conduct the experiments, and use the experimental results to answer the following two questions:

Question 1: Can DCGC-SAN proposed in this paper get better recommendation results than the baseline methods?
Question 2: What are the effects of the parameters of DCGC-SAN, such as the number of neighbor sessions sampled and the dimension of the embedding vector, on the recommendation performance?

4.1 Datasets

In order to test the performance, we conducted experiments on two public e-commerce data sets Yoochoose and Diginetica.

The Yoochoose dataset is a public dataset released by RecSys Challenge in 2015. It contains a collection of sessions from a retailer, where each session is encapsulating the click events that the user performed in the session. The Diginetica dataset comes from the competition CIKM 2016. In our experiments, only the data of the transaction part is used.

For fairly comparison, we follow the common data set processing methods in previous studies [4, 6, 8, 10]. All the session sequences are rearranged chronologically, and sessions with length one and items that appear less than five times are deleted. Furthermore, we used the last day of Yoochoose and the last seven days of Diginetica to generate test set. Since it is impossible for the model to generate items that have not appeared before, we delete items from test set that did not appear in the training set. And, moreover, due to the large amount of data in the Yoochoose, the experiments only use the latest 1/64 data, which is called the Yoochoose 1/64.

The statistics of the two data sets after the data preprocessing stage are shown in Table 1.

Table 1. Dataset statistics

Datasets	Yoochoose 1/64	Diginetica
All the items	557,248	982,961
Training sessions	369,859	719,470
Test sessions	55,898	60,858
Items	16,766	43,097
Average length	6.16	5.12

4.2 Evaluation Metric

We use two metrics, P@20 and MRR@20, to evaluate the performance.

P@20: The recommendation accuracy rate of the model, which represents the proportion of correctly recommended items appearing in the Top-20 recommendation list.

MRR@20: The reciprocal average of the correct item's ranking in the Top-20 recommendation list. MRR considers the ranking order of recommended items. The higher the MRR, the higher the ranking of the correctly recommended item in the recommendation list. If the correct item rank is greater than 20, then the value of MRR is 0.

4.3 Experiment Settings

The experiment uses a Gaussian distribution with a mean of 0 and a standard deviation of 0.1 to initialize all parameters in the DCGC-SAN. The optimizer uses the Adam [20] optimizer, initial learning rate is set to 1e-3, and the attenuation is set to 0.1 every 3 training rounds. The batch size is set to 128, and the L2 regularization parameter is set to 1e-5 to reduce overfitting.

4.4 Comparison with Baseline Methods

In order to answer question 1, we conducted experiments on two data sets on the DCGC-SAN and baseline methods. The experimental results are illustrated in Table 2, and the best results are marked in bold.

Table 2. Performance comparison between different methods

Methods	Yoochoose 1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
GRU4REC [4]	60.64	22.89	29.45	8.33
NARM [6]	68.32	28.63	49.70	16.17
STAMP [8]	68.74	29.67	45.64	14.32
KNN-RNN [5]	62.36	24.05	31.89	9.65
CSRM [12]	70.79	30.48	50.55	16.38
SR-GNN [15]	70.57	30.94	50.73	17.59
GC-SAN [16]	70.66	30.04	51.34	17.61
TAGNN [18]	71.02	31.12	51.31	**18.03**
DCGC-SAN	**71.37**	**31.73**	**51.60**	17.76

From the experimental results in the table above, we can conclude that:

In the RNN-based recommendation model, GRU4REC uses GRU to capture the sequence information of the session sequences, which is the basic model. Both NARM and STAMP apply the attention mechanism on the final interactive items, in order to capture sequential information. The performance of these two models is better than GRU4REC, which shows the effectiveness of the attention mechanism in session-based recommendation tasks. CSRM introduces the collaboration information in the field into the model. Compared with KNN-RNN, the main difference between the two models is that KNN-RNN uses fixed parameters to combine the two, while CSRM combines more complex nonlinear conversion, so the performance of CSRM is better. Moreover, the performance of KNN-RNN and CSRM is better than that of the basic RNN model, which shows that the collaborative information in the field can improve the performance.

The performance of SR-GNN is better than that based on the RNN model, which shows that the graph node embedding algorithm can effectively learn the representation vector of each graph node. Both GC-SAN and TAGNN are improvements to SR-GNN because of attention mechanism. They all improve the recommendation performance of the graph neural network to a certain extent, and TAGNN has achieved the best performance in the MRR@20 of Diginetica dataset.

DCGC-SAN divides two different channels, intra-session and inter-session channels, disseminates different information within the two channels, and learns more accurate graph node embedding vectors by fusing the information of the two channels. In terms of attention mechanism, our model abandons the simple soft attention mechanism and use a more complex attention network to assign the weight of different items in an adaptive weighting manner. It can be clearly seen from Table 2 that DCGC-SAN achieves the best performance on the P@20 and MRR@20 of the Yoochoose 1/64 dataset and P@20 of the Diginetica dataset.

In order to answer question 2, we modified the parameters of the model and made multiple sets of comparative ablation experiments.

4.5 The Influence of Model Parameters on Experimental Results

The Influence of the Number of Neighbors. DCGC-SAN samples and models similar neighbor session sequences, and the number of samples obviously affects the final recommendation performance. If only considering the item conversion of the current session, the model can dig deeper into the current user's preferences, but it will lose the important item conversions reflected in other similar sessions. If a large number of similar neighbor session sequences are introduced, it is equivalent to expanding the scope of obtaining item conversion, and excessive noise may be introduced. Therefore, on the Yoochoose 1/64 and Diginetica data sets, we increase the number of neighbor session from 0 to 160 to study the impact of different neighbor sample numbers on the recommendation performance. Due to limited space, only P@20 is considered here, and the following experiments only consider P@20 for the same reason.

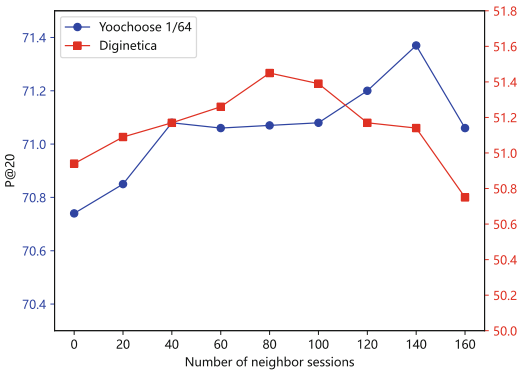

Fig. 3. The influence of the number of sampling neighbors

Figure 3 shows the influence of neighbors with different numbers of neighbors on the recommendation results. As the number increases, the general trend of model performance is to increase first and then decrease. The recommendation performance of DCGC-SAN on the Yoochoose 1/64 at n = 140 and Diginetica at n = 80 reaches its peak. The reason for the decline is that the number of neighbors sampled is too large, resulting in some neighbor sessions that are not very similar to the current session (the larger the number of neighbors, the worse the similarity of the session sequence), which introduces excess noise. Compared with n = 0, on the two datasets, the introduction of item conversion in neighbor sessions (choose the appropriate number of samples) significantly improves the performance of the model, which reflects the rationality of fusing dual-channel item conversion effectiveness.

The Influence of Channel Fusion Method. In order to obtain the most suitable fusion method of the two channel information, we analyzed how different fusion functions affect the experimental results. In the learning stage of the item node embedding vectors in the graph neural network, the following strategies are used. The experimental results are shown in Fig. 4.

- Only use the item node information of the current session (intra-only);
- Only use the item node information in the neighbor session (inter-only);
- The information of the two channels is merged. There are three ways of channel fusion, namely concat, max and avg.

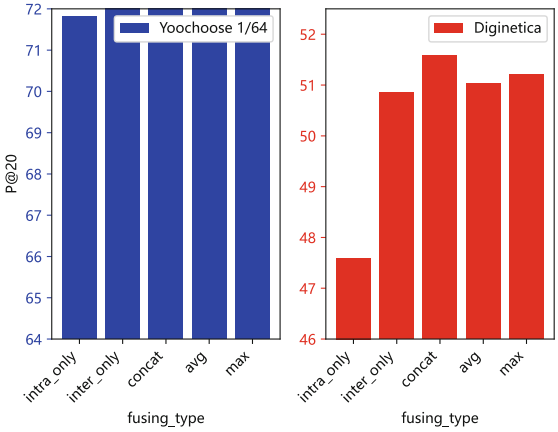

Fig. 4. The influence of channel fusion mode

It can be seen from Fig. 4 that the performance of the model using the current session only is the worst, and it is quite different from the other four methods, which shows that it is far from enough to consider only the item conversion in the current session. In comparison, the recommendation performance of the model that only uses the item conversion information of neighbor sessions is unexpectedly good. This is because although the item conversion in the current session sequence is discarded, the

appropriate item conversion can still be learned from a large number of neighbor sessions.

Among the three methods that integrate inter-session and intra-session channels, the concat fusion method works best because it is the most complicated of the three fusion methods and can better integrate the information of the two channels, while max and avg, comparatively speaking, it is too simple.

The Influence of Embedding Layer Dimensions. In Fig. 5, we study the influence of the size d of the embedding vector of the graph node on the experimental results. We increased d from 20 to 160 on the two data sets to explore the influence of the embedding layer dimension on the experimental results.

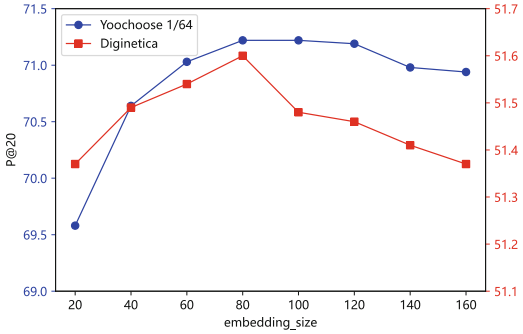

Fig. 5. The influence of embedding layer dimensions

It can be seen from Fig. 5 that ignorantly increasing the dimension of the graph embedding vector does not always improve the performance of the model. For the Yoochoose 1/64, the best embedding size is 80 or 100, and for the Diginetica, the best embedding size is 80. When the dimension of the embedding vector is less than the optimal dimension, the number of dimensions that can be expressed is limited, and the model performance is not too high. When the embedding dimension is greater than the optimal size, the model may be overfitting and the performance will decrease.

The Influence of the Number of Layers of the Attention Network. The attention network can capture the dependency between items that are far away. We first need to determine whether the attention network is effective. We apply the ordinary soft attention mechanism in SR-GNN to our model (DCG-SA), and compare it with the complete DCGC-SAN. Table 3 is the comparison result. From Table 3, it can be seen that the attention network can improve the performance of the recommendation system.

Table 3. Performance comparison of soft attention mechanism and attention network

Methods	Yoochoose 1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
DCG-SA	71.02	31.70	50.84	17.32
DCGC-SAN	**71.37**	**31.73**	**51.60**	**17.76**

In order to study the influence of the number of layers of the attention network on the recommendation results, we increased K from 1 to 5 on the two data sets to experiment with the model. The results of the experiment are shown in Fig. 6.

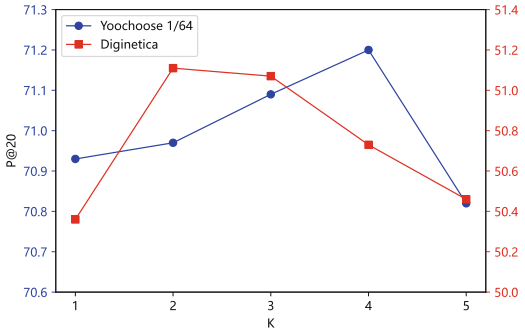

Fig. 6. The influence of K on experimental results

It can be seen from Fig. 6 that increasing K can improve the performance of the DCGC-SAN. After the best K value is exceeded, the larger the K, the worse the performance of the model. This is because too many network layers can obtain more abstract information, but will lose low-dimensional information, while network layers with too few numbers cannot obtain high-dimensional abstract information.

5 Conclusions

This paper proposes the DCGC-SAN model, which divides the intra-session and inter-session channels to learn the information of different item nodes in the learning stage of the graph node embedding vectors, and effectively integrates the two vectors through the fusion function, combined with the attention network capturing the dependency information between items that are far apart. Experimental results on two public e-commerce data sets show that our model is better than baseline methods in terms of recommendation accuracy.

References

1. Anh, P.H., Bach, N.X., Phuong, T.M.: Session-based recommendation with self-attention. In: Proceedings of the Tenth International Symposium on Information and Communication Technology, pp. 1–8 (2019)
2. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. ArXiv Prepr. ArXiv13126203 (2013)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
4. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based Recommendations with Recurrent Neural Networks. ArXiv151106939 Cs (2016)
5. Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 306–310 (2017)
6. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1419–1428. Association for Computing Machinery, New York, USA (2017). https://doi.org/10.1145/3132847.3132926
7. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. ArXiv Prepr. ArXiv151105493 (2015)
8. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: STAMP: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1831–1839 (2018)
9. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)
10. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 17–22 (2016)
11. Vaswani, A., et al.: Attention is all you need. ArXiv Prepr. ArXiv170603762 (2017)
12. Wang, M., Ren, P., Mei, L., Chen, Z., Ma, J., de Rijke, M.: A collaborative session-based recommendation approach with parallel memory modules. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 345–354 (2019)
13. Wang, S., Pasi, G., Hu, L., Cao, L.: The era of intelligent recommendation: editorial on intelligent recommendation with advanced AI and learning. IEEE Ann. Hist. Comput. **35**, 3–6 (2020)
14. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International Conference on Machine Learning, pp. 6861–6871. PMLR (2019)
15. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 346–353 (2019)
16. Xu, C., et al.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI, pp. 3940–3946 (2019)
17. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 729–732 (2016)

18. Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., Tan, T.: TAGNN: target attentive graph neural networks for session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1921–1924 (2020)
19. Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., Cai, D.: What to do next: modeling user behaviors by time-LSTM. In: IJCAI, pp. 3602–3608 (2017)