Dynamic Graph Attention-Aware Networks for Session-Based Recommendation

Ahed Abugabah
Zayed University
Abu Dhabi, UAE
ahed.abugabah@zu.ac.ae

Xiaochun Cheng
Middlesex University
London, UK
x.cheng@mdx.ac.uk

Jianfeng Wang
Sun Yat-sen University
Guangzhou, China
wjf739@gmail.com

***Abstract*—Graph convolutional neural networks have attracted increasing attention in recommendation system fields because of their ability to represent the interactive relations between users and items. At present, there are many session-based methods based on graph neural networks. For example, SR-GNN establishes a user's session graph based on the user's sequential behavior to predict the user's next click. Although these session-based recommendation methods modeling the user's interaction with items as a graph, these methods have achieved good performance in improving the accuracy of the recommendation. However, most existing models ignore the items' relationship among sessions. To efficiently learn the deep connections between graph-structured items, we devised a dynamic attention-aware network (DYAGNN) to model the user's potential behavior sequence for the recommendation. Extensive experiments have been conducted on two real-world datasets, the experimental results demonstrate that our method achieves good results in capturing user attention perception.**

***Index Terms*—Session-based recommendation, Graph neural networks, ranking**

I. Introduction

Recommendation systems aim to help users select interesting content and improve customers' online experiences in many application domains. The task of sequential recommendation is to predict the users' personal preferences from their past behavior. In the online environment, many platforms such as Reddit, Xing and Taobao can use the interaction between users and items to predict user preferences. Considering the high practical value of interaction, a lot of approaches have been devised to learn sequential features for session-based recommendation. Ying *et al.* [1] devised a highly-scalable graph convolutional neural network (GCN) framework to aggregate the embedding of nodes (i.e., items). Monti *et al.* [2] combined graph convolutional network and graph recurrent network to make full use of the local stationarity structures of user/item graphs. Although these methods have made great success in the task of modeling sequences, there are still great challenges in modeling complex relationships based on user behavior sequences.

Recommendation systems can be roughly categorized into feature-based recommendation [3], social recommendation [4] [5] and sequential recommendation [6]. By combining the information of users and the features of items, the feature-based method is used to model the interactive behaviors of users and items, and predict the probability that the user will select items [3]. Although it can effectively learn the embedded vector of items in the user-item interaction network, the computational costs are tremendous to capture user preference. Social recommendation [4] [7] based on social information [8] can alleviate data sparsity and cold-start problems, but the premise of these studies is that all users of social links have similar preferences. Since users' interests are dynamic [9] [10], it depends not only on the preferences of users, but also needs to understand the shift of user's interests.

Existing recommendation methods mainly depend on the Markov Chains (MCs) [11] [12] [13], and Recurrent Neural Networks [14], and Graph Neural Networks (GNNs) [15] [16]. He *et al.* [11] devised a personalized sequential recommendation model based on a Markov chain to deal with the anonymous session recommendation problem [17]. Wu *et al.* [16] used a graph neural network to solve the complex relationship between items and recommended items. However, these graphs are typically multi-relational and heterogeneous. Similar to SR-GNN [15], Song *et al.* [9] also modeled a dynamic user's behaviors with a recurrent neural network to infer the influencers based upon users' current interests.

Although these methods are state-of-the-art and achieve satisfactory results, there are some issues that can be researched in detail. Existing methods lack the capability of local dependencies so that they cannot perform representation for context sequences in the session. In addition, one important property of the model is that the relations between disordered structural items achieve high performance. However, the existing methods usually extract discrete features [18] manually, and they lack the ability to represent the embedded data.

To this end, we come up with a dynamic attention-aware model through a graph neural network, which maybe aware of the change in user attention in the session, capture the user's dynamic interest, and recommend items. Fig. 1 shows the sequence of interactions between users and items. Since the nodes between the graphs have a strong relationship, the user's session sequence is constructed to construct the users session graph. In the graph, nodes v_1, v_3, v_4, and v_5 form the core connection component, which is reflected by the edge relationship. Compared with the previous version [15], we further integrate attention perception and the LSTM mechanism into graph neural networks. Extensive experiments conducted on the two datasets show the effectiveness of this method in

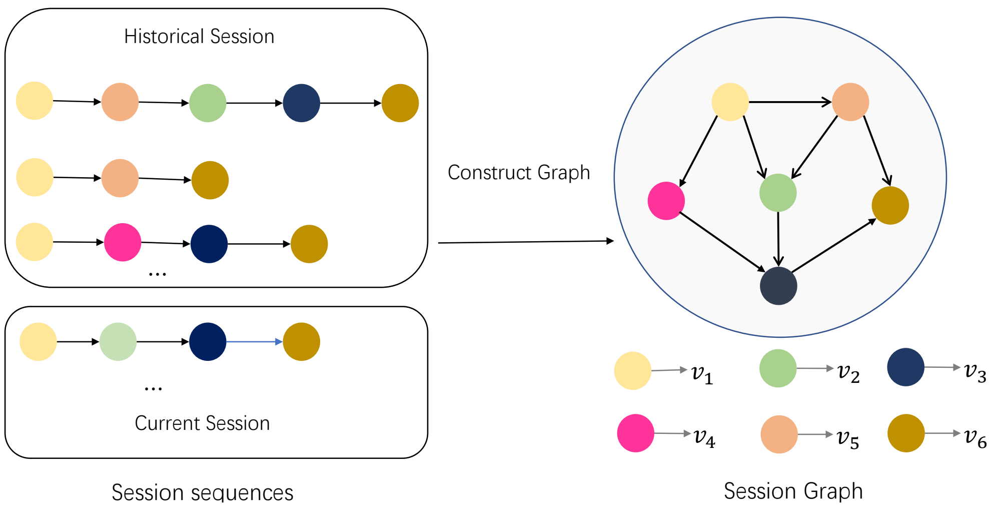

Fig. 1. User behavior session construction graph. In session sequences, a user's interaction sessions includes historical sessions and the current session. The different color circles in the graph represent different items which is v_1, v_2, v_3, v_4, v_5 and v_6.

solving session recommendations.

The main contributions of this work are as follows.

- We devised a new method for the session-based recommendation scenario, which captured the items inherent relationship in different session, which was key to the item level feature representation. The method can predict users' interests without requiring users to log in with detailed identity information.
- We used the dynamic attention mechanism to explicitly model the relation between the user's historical interests and current session, which more accurately captured the user's interest and made accurate predictions.
- Extensive experiments conducted on the two datasets have proved the validity of different components of our model, which can significantly improve the performance of the session recommendation.

II. Related work

Session-based recommendation is an application of recommender systems based upon implicit feedback, where users' identities are unknown, and have some positive signal (e.g., purchases or clicks) [11]. These positive observations are regarded as a form of sequential data obtained by tracking users' records in a sequence. Most existing sequential recommendation models adopt the self-attention mechanism to capture distant item-item transitions in a sequence and have achieved good performance. One of the advantages of attention mechanisms is that they allow to the variable-sized inputs, focusing on the most significant parts of the input to forward propagation. [19] proposed a model based on a convolution architecture to process graphs of arbitrary size and shape. Vaswani *et al.* [20] showed that self-attention can improve a method based on RNNs or convolutions. However, it is still challenging to establish complex contextual information between adjacent items.

Recently, many models have been proposed. The RNN model is the most common model, and it is used to model the sequence model. Hidasi *et al.* [21] modeled the user's behavior sequence based on the features of the clicked items and the user's click. Further, Li *et al.* [22] used an attention mechanism combined with RNN to capture the user's sequence behavior characteristics and main intentions. Jin *et al.* [23] combined RNN-based network with key-value to capture user preferences and sequence-level user preferences.

Graph network methods have been widely used to solve sequence-based recommendation tasks [24] [25]. Due to the unique structure of graph neural networks, many various networks based on the graph have been devised. Based on a mechanism similar to a messaging network, many GNN methods [26]–[28] are devised to calculate the information relationship between nodes. Wu *et al.* [15] put forward SR-GNN to aggregate item features into session features by encoding item features. Similarly, Wang *et al.* [25] proposed the non-local neural network (NLNN) by combining a variety of self-attention methods.

Nowadays, many neural network-based models are used to represent and understand graph structures. Cohn *et al.* [26] proposed GatedGNN based on gated recurrent units to calculate gradient. Graph Attention Network has been widely used to solve the sequence-based recommendation problems. Vardhan *et al.* [27] proposed graph attention network to learn different nodes and neighbor nodes weights based on attention mechanism. These models using the graph convolution network have great advantages in processing graph-structured data. Chan *et al.* [29] proposed a the CNN-based CTR prediction method based on CNN to model the sequence. Considering that user preferences are more focused or the ongoing session, we focus on the user's dynamic interest representations to enhance session representation of user.

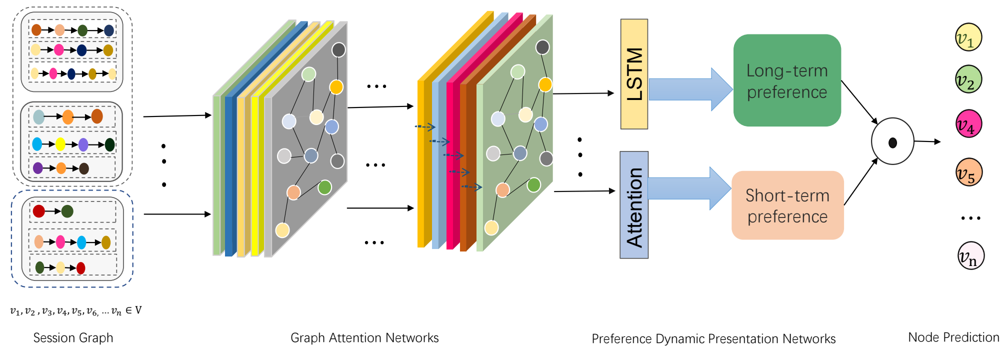

Fig. 2. An illustration of our model. In the model, each edge, denoted by a line, only connects two vertices. Each edge, denoted by a colored ellipse, connects more than two vertices. The model is divided into four parts, which are constructing conversation graph, graph attention network extracting node information, preference dynamic presentation network, node prediction network.

TABLE I
NOTATION

Symbol	Definition
E	The set of edges in a graph
V	The set of nodes in a graph
G	A graph
n	The number of the graph nodes
i	The i-th clicked item with a session
$\mathbf{x_i}$	The feature vector of the i-th node
$\hat{y}$	The probabilities of the next-clicked nodes
σ	The Sigmoid function
θ	The learnable parameter

III. Method

In this section, we will introduce a dynamic graph attention-aware network for session-based recommendations, as shown in Fig 2. Three modules are included in our method. First, we model all the session sequences as session graphs. Secondly, we design a dynamic attention-aware network based on graph attention network and update the features of graph nodes to select relatively important items from the session to capture the user's main intentions. Combined the long short-term memory and self-attention mechanism to model the user's long-term and short-term interests, and capture the user's dynamic interests. Finally, we establish a unified interest representation model based on the user's long- and short-term interests, and use the Softmax function to calculate the similarity between the prediction and the truth. Frequently used notation is summarized in Table I.

A. Construct session graph

Session-based recommendations are designed to predict the next item that a user will click on. It is influenced not only by the user's long-term interest, but also but also by the current interaction sequence. In the paper, we denote a graph with $G = (V, E)$, where V denotes the nodes set, and E denotes the set of edges that connects two related items. The user's session sequence can be represented as $s = [v_1, v_2, \dots, v_n]$, and the embedded items can be simply defined as v_{n-1} of the last-clicked item v_n.

In a session graph, each edge $(v_{i-1}, v_i) \in E$ means the connection relationship between item v_i and v_{i-1} in the session s. For every node v, the item $v \in V$ was mapped into an latent embedding space, and the item latent vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, $\mathbf{x}_i \in R^d$, d is the embedding dimensions of items. Our goal is to predict the top N candidates based on the user's final unified vector representation [30].

Compared with the sequence structure data, the graph structure data has rich structural information, and it is efficient to capture the transition relationship between each user's items according to the structure information of the graph. The input item is represented as a dense vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n]$ converted by the embedding layer and fed into the deep neural network. The output feature is $\mathbf{x}' = [\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_i, \dots, \mathbf{x}'_n]$. Given that GAT [27] can select relatively important nodes, this method is suitable for evaluating the importance of nodes [31]. For each node v in the graph, the propagation between the different nodes can be formalized as equation (1):

$$\mathbf{e}_{ij} = \boldsymbol{\alpha}\left(\mathbf{W}\mathbf{x}_i, \mathbf{W}\mathbf{x}_j\right), \tag{1}$$

where α is the attention coefficient, which represents the importance of node i to node j, and W is the weight parameter in the aggregation layer.

$$\boldsymbol{\alpha}_{ij} = \frac{\exp\left(\mathbf{e}_{ij}\right)}{\sum_{m \in N(i)} \exp\left(\mathbf{e}_{ik}\right)}, \tag{2}$$

where the attention weight α_{ij} is determined by an item i and j. $N(i)$ is all adjacent nodes of node i. Considering that information in the embedding may have different priority, the different weight is assigned to different nodes, and the attention-aware mechanism is adopted to represent the preference. The weights are as equation (3).

$$\mathbf{x}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}\mathbf{x}_j\right), \tag{3}$$

where $\mathbf{W}$ is the weight matrix of feature $\mathbf{x}_j$, and $W \in \mathbb{R}^{d\times d}$ control the weights of the different items. $\mathbf{x}_j$ represents the neighbor nodes of node i. $\sigma(\cdot)$ denotes the sigmoid function that was used to regularize all adjacent nodes of node i. The node i has a multi-end attention mechanism in the neighborhood, and the final representations of the nodes are obtained by means of average multi-head attention as equation (4).

$$\mathbf{x}_i' = \sigma\left(\frac{1}{k}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}\mathbf{W}\mathbf{x}_j\right), \tag{4}$$

where k is the number of the adjacent nodes of node i.

B. Generating user dynamic representation

We apply Long Short Term Memory (LSTM) network [32] as the recurrent cell. The LSTM can be described as equation (5).

$$\begin{aligned}
\mathbf{in}_t^u &= \sigma\left(\mathbf{W}_{in}^1\mathbf{x}_i' + \mathbf{W}_{in}^2\mathbf{h}_{t-1}^u + \mathbf{b}_{in}\right)\\
\mathbf{f}_t^u &= \sigma\left(\mathbf{W}_f^1\mathbf{x}_i' + \mathbf{W}_f^2\mathbf{h}_{t-1}^u + \mathbf{b}_f\right)\\
\mathbf{o}_t^u &= \sigma\left(\mathbf{W}_o^1\mathbf{x}_i' + \mathbf{W}_o^2\mathbf{h}_{t-1}^u + \mathbf{b}_o\right)\\
\mathbf{c}_t^u &= \mathbf{f}_t^u\mathbf{c}_{t-1}^u + \mathbf{in}_t^u\tanh\left(\mathbf{W}_c^1\mathbf{x}_i' + \mathbf{W}_c^2\mathbf{h}_{t-1}^u + \mathbf{b}_c\right)\\
\mathbf{h}_t^u &= \mathbf{o}_t^u\tanh\left(\mathbf{c}_t^u\right)
\end{aligned} \tag{5}$$

where $\mathbf{in}_t^u$, $\mathbf{f}_t^u$, $\mathbf{o}_t^u$ denotes the input, forget and output gates, respectively. The LSTM encodes the short-term interaction sequence of u into a hidden output vector $\mathbf{h}_t^u \in \mathbb{R}^{d\times 1}$ at time t. $\mathbf{c}_t^u$ represents the cell states of LSTM, and $\mathbf{b}$ represents the biases. In the first layer, 1 LSTM network can be connected, in which the cell status and hidden representation can be propagated from t-1 to t.

For the sake of calculating the importance of node in the current session, we consider the node's sequence behaviors represented by the vector that contain some useful information about the user's primary shopping intent. Therefore, we use the representation of the sequence in the session.

We resort to the self-attention mechanism, which focuses on solving the importance of different items, discovers the user's short-term interest, and seeks out the user 's attention shift to different items. So, we first aggregate the information of all nodes as a query feature:

$$\mathbf{q}_i = \mathrm{ReLU}\sum_{i=1}^{N}\mathbf{W}\mathbf{h}_t^u \tag{6}$$

where $\mathbf{W}$ is the learnable parameter matrix. Then the attention scores of all nodes can be calculated as equation (7).

$$\boldsymbol{\beta}_i = \mathrm{Sigmoid}\left(U_s\tanh\left(\mathbf{W}_h\mathbf{h}_t^u + W_q\mathbf{q}_i + \mathbf{b}_s\right) + \mathbf{b}_u\right) \tag{7}$$

where $\mathbf{U}_s, \mathbf{W}_h, \mathbf{W}_q$ are the learnable parameter matrixes. $\mathbf{b}_s$ and $\mathbf{b}_u$ are the bias. We use the non-linear function of Sigmoid due to the possibility of existing multiple items.

Similar to the model SR-GNN [15], we use the attention mechanism to encode the current embedding matrix to global representation and local representation, where global representation denotes the user's general interest and local representation denotes the user's new interest. The global representation is defined as equation (8).

$$\begin{aligned}
\mathbf{F}_i^g &= \sum_{i=1}^{N}\mathbf{h}_t^u\\
\mathbf{F}_i^l &= \sum_{i=1}^{N}\beta_i\cdot\mathbf{h}_t^u
\end{aligned} \tag{8}$$

Combining the long and short-term interests of the user to form a unified user representation, the uniformly represented vector is computed by the concatenation of the local and global vectors as equation (9).

$$\mathbf{C} = Concat(\mathbf{F}_i^g, \mathbf{F}_i^l)\cdot\mathbf{B} \tag{9}$$

where $\mathbf{C}$ represents the user's unified interest representation and $\mathbf{B}$ represents the matrix, and matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$ compresses two combined embedding vectors into the latent space $\mathbb{R}^d$.

C. Model Prediction

After passing the concatenated vectors to a feed forward neural net, generating the softmax output layer, and adopting cross entropy as a loss function for training, we get the score for each candidate item $v_i \in V$ by a softmax function as equation (10).

$$\hat{y}_i = softmax(\mathbf{C}\cdot\mathbf{x}_i) \tag{10}$$

where $\hat{y} \in R^m$ denotes the probabilities of nodes being the next click in session s. For each session graph, the loss function is defined as the cross-entropy of the prediction and the ground truth. Finally, we train our model by minimizing the following objective function as equation (11).

$$L(\hat{y}) = -\sum_{i=1}^{m} y_i\log\left(\hat{y}_i\right) + (1-y_i)\log\left(1-\hat{y}_i\right) + \lambda\|\theta\|^2, \tag{11}$$

where $\hat{y}$ denotes the one-hot encoding vector of the ground truth item, and θ is the set of all learnable parameters. λ is the parameter to reduce the loss errors and regularizations.

IV. Experiments and Analysis

In this section, we conduct experiments on different data sets to verify the effectiveness of our method. We first summarize the datasets and parameter settings of our experiment. Then we compare our method with other latest methods. Finally, we make a comprehensive assessment of our method.

A. Datasets

Our experimental data is mainly collected in two standard datasets, and the advantages and disadvantages of different recommendation algorithms are evaluated through a series of comparative experiments. For session-based recommendations, we adopt the last seven days of session data as the test dataset, and the rest as train dataset. Similar to [33], for a session sequence $S = \{s_1, s_2, \ldots, s_n\}$, we generate the input and corresponding labels $(\{s_1\}, s_2), (\{s_1, s_2\}, s_3), \ldots, (\{s_1, \ldots, s_{n-1}\}, s_n)$ for training and testing on the two datasets.

TABLE II
STATISTICS DETAILS OF DATASETS.

Dataset	train	test	items	avg.len
Xing	69,135	22,548	36,968	5.43
Reddit	901,161	234,327	16,766	6.16

- Xing [1]. The Xing Data is provided by a job search website that helps recruiters choose suitable candidates. Xing has fifteen million users and approximately one million job interaction records on the platform. User behavior includes click, browse, mark and delete actions. To protect the privacy of users, the dataset is anonymized after being extracted from real user profile data, so the data is incomplete and full of noise. We process the data, divide it into sessions at half-hour intervals, and remove repeated interactions of the same type within sessions in the session. We delete items with a length of less than twenty and users with a session length of less than two to model user interactions.
- Reddit [2]. The dataset contains the username, the username comment, and the timestamp of the comment. Each row denotes the user's comment and the meaning of Reddit data. We process each user's record and divide it into multiple sessions. In the process, we used the same method as for the Xing data set, but the processing time is half hour.

Before the experiment, we preprocessed the data sets. For each user, Eighty percent of user' sessions was divided into the training set. The remaining session was used as a test set. The descriptive statistics of two datasets after the preprocessing steps are in Table II.

B. Implementation details

In our model, the item dimension is set to 50, and the user's embedded dimension is set to 30 for all the experiments. The optimization function uses Adam, and the initial learning rate is set to 0.001. In addition, the batch size and the L2 regularization are set to 10 and 10^{-5}, respectively. For Xing and Reddit, the historical session length is set to 50 and 30, respectively. The Gaussian distribution initializes all parameters with a mean of 0 and a standard deviation of 0.1. The platform we use is the GeForce GTX Titan GPU to accelerate model training.

C. Baseline methods

In order to evaluate the performance of the model, we use five methods: Item-KNN, FPMC, GRU4Rec, H-RNN, and SR-GNN. Among them, the SR-GNN model is based on the graph model. A brief description of the five methods follows:

- FPMC [34] is a sequence modeling method based on Markov chains to predict the user's next action according to the previous action.

[1]https://2016.recsyschallenge.com/
[2]https://www.kaggle.com/colemaclean/subreddit-interactions

- GRU4Rec [35] is a sequence modeling method based on recurrent neural networks to model user sequences for the session-based recommendation.
- H-RNN [36] is a sequence modeling method extended to recurrent neural networks to predict the user's interest in the current session on the basis of recent sessions.
- Caser [37] is a sequence modeling method based on convolution kernels similar to image convolutions operation, and using convolution filters to learn the sequence pattern in order to learn user's preferences.
- SR-GNN [15] employed GNN with attention mechanisms to model the sequence as a session to capture the main purpose of the user.

D. Evaluation methods

Since the recommendation system recommends a limited items every time, these items that suit the user should appear in the limited number of items listed in the candidate products. We use the following two methods to evaluate the quality of the model.

- Recall: We adopt the Recall@k indicator for evaluation, which statistically tests the likelihood that the user's actual clicked item appears in the top 20 of the recommended candidates. Recall does not consider the position order relationship of the items that the user actually clicks on the candidates, and only needs the first N products to appear in the recommendation list.
- MRR (Mean Reciprocal Rank): Another evaluation metric is MRR@k, which is the reciprocal average of the number of items clicked in the list. If the item doesn't appear in the top 20 of the recommendation list, then the MRR should be set to zero. Considering the items of the evaluation recommendation list, MRR is a very important indicator in some sequentially sensitive tasks.

E. Comparisons of Performance

We compare our algorithm with other algorithms in detail. All algorithms are tested on two datasets. In addition, Table III illustrates in detail the results of our baseline approach.

TABLE III
THE PERFORMANCE OF OUR METHOD WITH OTHER BASELINE METHODS OVER TWO DATASETS.

Algorithm	Reddit		Xing	
	Recall@20	MRR@20	Recall@20	MRR@20
FPMC	0.3885	0.0454	0.5012	0.1469
GRU4Rec	0.4432	0.2600	0.5016	0.1723
H-RNN	0.5004	0.2744	0.5058	0.1725
Caser	0.5033	0.1349	0.5072	0.1737
SR-GNN	0.6180	0.2855	0.5107	0.1740
DYAGNNRec	0.6795	0.3388	0.5168	0.2052

From the results of the experiment, we can draw the following conclusions: (1) In Xing, when we replace the user's factor with the average value of the hidden factor of the hidden matter in the current meeting, the performance is not satisfactory. We take each conversation sequence in the algorithm as a new session sequence, which requires a lot of memory space. As a result, the experimental results on this dataset were not given. The existence of these problems is based on a matrix with a number of users and is not applicable to the recommended task group for the session. (2) All in all, the performance of the three models based on the graph network is superior to the traditional method, which shows that the graph network can make good use of the sequence information in the session. (3) In view of the current user's sequence behavior and main intentions in the model, we conclude that the performance exceeds all the methods on the Recall@20. Taking the Reddit dataset as an example, our method is compared with improved SR-GNN at $Recall@20$ and $Recall@20$ On the evaluation. The performance of our method was increased by 6.15% and 5.33%. Our approach is a significant improvement on the Xing both Recall@20 and MRR@20 compared to SR-GNN.

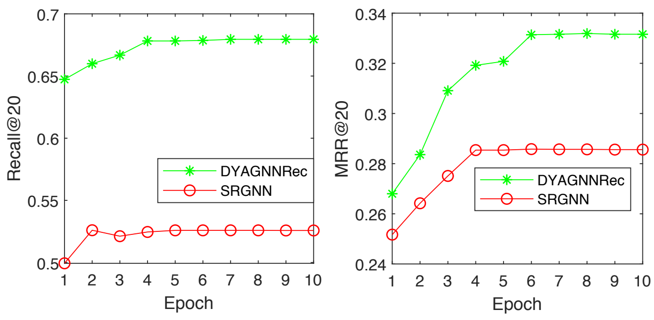

Fig. 3. The performance of the Reddit dataset on different epochs.

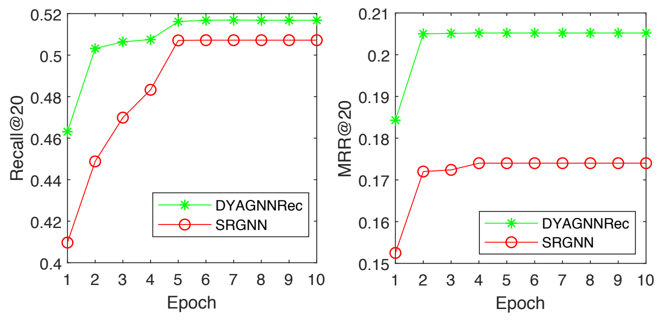

Fig. 4. The performance of the Xing dataset on different epochs.

V. Conclusions

In this paper, we have devised a novel dynamic attention-aware neural network method for session-based recommendation, which captures the important neighbor nodes of items with the graph attention network, models the long-term preferences by adopting the short-term and long-term memory unit, and integrates the short-term interests of the attention mechanism to form a unified interest expression. The model not only relies on the association relationship between nodes to select the most relevant neighbors for each node and update its feature but also uses attention awareness to explicitly model the impact of historical sessions on current sessions. Comprehensive experiments conducted on two public datasets show the effectiveness of different parts in our model and confirm that the devised model delivers a significant improvement on MRR@20 and Recall@20.

References

[1] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD*, 2018, pp. 974–983.
[2] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *NIPS*, 2017, pp. 3700–3710.
[3] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation," in *SIGKDD*, 2017, pp. 1245–1254.
[4] K. Shu, S. Wang, J. Tang, Y. Wang, and H. Liu, "Crossfire: Cross media joint friend and item recommendations," in *WSDM*, 2018, pp. 522–530.
[5] F. Corò, G. D'Angelo, and Y. Velaj, "Recommending links to maximize the influence in social networks," in *IJCAI*, 2019, pp. 2195–2201.
[6] Y. Kim, K. Kim, C. Park, and H. Yu, "Sequential and diverse recommendation with long tail," in *IJCAI*, 2019, pp. 2740–2746.
[7] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, "Deep adversarial social recommendation," in *IJCAI*, 2019, pp. 1351–1357.
[8] I. A. Ridhawi], S. Otoum, M. Aloqaily, Y. Jararweh, and T. Baker, "Providing secure and reliable communication for next generation networks in smart cities," *Sustainable Cities and Society*, vol. 56, p. 102080, 2020.
[9] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *WSDM*, 2019, pp. 555–563.
[10] W. Liu, Z. Wang, B. Yao, and J. Yin, "Geo-alm: POI recommendation by fusing geographical information and adversarial learning mechanism," in *IJCAI*, 2019, pp. 1807–1813.
[11] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *ICDM*, 2016, pp. 191–200.
[12] C. Cai, R. He, and J. McAuley, "SPMC: socially-aware personalized markov chains for sparse sequential recommendation," in *IJCAI*, 2017, pp. 1476–1482.
[13] W. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*, 2018, pp. 197–206.
[14] K. Song, M. Ji, S. Park, and I. Moon, "Hierarchical context enabled recurrent neural network for recommendation," in *AAAI*, 2019, pp. 4983–4991.
[15] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *AAAI*, 2019, pp. 346–353.
[16] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019, pp. 6861–6871.
[17] Q. Yang, Z. Zhou, Z. Gong, M. Zhang, and S. Huang, Eds., *PAKDD*, ser. Lecture Notes in Computer Science, vol. 11439, 2019.
[18] M. H. Bhatti, J. Khan, M. U. G. Khan, R. Iqbal, M. Aloqaily, Y. Jararweh, and B. Gupta, "Soft computing-based EEG classification by optimal feature selection and neural networks," *IEEE Trans. Industrial Informatics*, vol. 15, no. 10, pp. 5747–5754, 2019.
[19] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *NIPS*, 2015, pp. 2224–2232.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.
[21] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *RecSys*. ACM, 2016, pp. 241–248.
[22] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *CIKM*. ACM, 2017, pp. 1419–1428.

[23] H. Jin, W. X. Zhao, H. Dou, J. R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *SIGIR*. ACM, 2018, p. 505–514.
[24] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *WWW*, 2019, pp. 417–426.
[25] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.
[26] D. Beck, G. Haffari, and T. Cohn, "Graph-to-sequence learning using gated graph neural networks," in *ACL*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018, pp. 273–283.
[27] "Graph attention networks," *CoRR*, vol. abs/1710.10903, 2017.
[28] L. Tseng, Y. Wu, H. Pan, M. Aloqaily, and A. Boukerche, "Reliable broadcast in networks with trusted nodes," in *GLOBECOM*. IEEE, 2019, pp. 1–6.
[29] P. P. K. Chan, X. Hu, L. Zhao, D. S. Yeung, D. Liu, and L. Xiao, "Convolutional neural networks based click-through rate prediction with multiple feature sequences," in *IJCAI*, 2018, pp. 2007–2013.
[30] F. Lv, T. Jin, C. Yu, F. Sun, Q. Lin, K. Yang, and W. Ng, "SDM: sequential deep matching model for online large-scale recommender system," in *CIKM*, 2019, pp. 2635–2643.
[31] L. Zhang, X. Wang, H. Li, G. Zhu, P. Shen, P. Li, X. Lu, S. A. A. Shah, and M. Bennamoun, "Structure-feature based graph self-adaptive pooling," *CoRR*, vol. abs/2002.00848, 2020.
[32] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *CoRR*, vol. abs/1508.01991, 2015.
[33] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *IJCAI*, 2019, pp. 3940–3946.
[34] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.
[35] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.
[36] M. Ruocco, O. S. L. Skrede, and H. Langseth, "Inter-session modeling for session-based recommendation," in *DLRS@RecSys*, 2017, pp. 24–31.
[37] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *WSDM*, 2018, pp. 565–573.