GRAPH NEURAL NETWORKS WITH INTRA- AND INTER-SESSION INFORMATION FOR SESSION-BASED RECOMMENDATION

Hongying Liu, Tianwen Zhu, Fanhua Shang, Guoliang Ma, Muyao Liu, Yuanyuan Liu*

School of Artificial Intelligence, Xidian University

ABSTRACT

The recommendation with anonymous sessions is challenging due to lack of information and more uncertainty. Although promising results have been achieved, they are still insufficient since most methods utilize only inter-session information. In this paper, we propose a novel two-stage method, named Graph neural networks with Intra- and Inter-session information for Session-based Recommendation (GIISR). In the first stage, we present new graph convolutional networks and modified residual gate neural networks (MRGNNs) to learn intra-session information. In the second stage, we construct a new local sensitive inter-session graph using previous fused information to aggregate the relation between different sessions. Moreover, both a new soft-attention mechanism and hybrid embedding technique are employed to consider the global preference and current interest. Comprehensive experimental results on representative datasets demonstrate that GIISR greatly outperforms state-of-the-art session-based recommendations. The source code of our method can be available at https://github.com/ChenaniahMe/GIISR.

***Index Terms*—** Graph neural network, intra-session and inter-session information, residual gate neural network

1. INTRODUCTION

Most of recommendation methods utilize the user history information and past profiles in the process of recommendation. However, the user history information and profiles may not be available as they are not recorded in most real-world applications. An increasing number of methods including the traditional and the deep learning-based, have been proposed for these session-based recommendations. The traditional methods, e.g. [1, 2, 3], adopt Markov chains to predict user's next behavior in terms of the previous one. Besides, many deep neural networks have been introduced for recommendations. For example, the recurrent neural networks (RNNs) are verified effective, In [4, 5], RNN is used for session-based recommendation. This model is improved by [6] that considers data augmentation techniques and temporal shifts of user behaviors. In [7], RNNs are utilized to mix the sequential information and combine neighborhood-based method. NARM [8] employs the attention mechanism to obtain dominating purpose of users. STAMP [9] is proposed to capture both users' overall interests and interests of the last click. Recently, the graph neural network is also proposed for session-based recommendation (SR-GNN) [10]. It models the items in a session with graph structured data, and uses a graph neural network to extract the transitions of items from this data, which yields sound results.

All the methods mentioned above may attain sound results. Nevertheless, they only consider the sequential behavior of users in current session, and the inter-session information between different sessions is not utilized. As the users' history information and profiles are lacking, the clicks in a single session are not adequate for accurate recommendation. Especially, the STAN method [11] employs the relation between different sessions and shows significant improvement. However, the performance of STAN is inferior to SR-GNN. It is likely because intra-session information captured by graph structure in SR-GNN may play an important role for the performance.

To address the limitations mentioned above, we propose a novel two-stage method, named Graph neural networks with Intra- and Inter-session information for Session-based Recommendation (GIISR), as shown in Fig. 1. To the best of our knowledge, *our method is the first work that exploits both the intra- and inter-session information using graph neural networks for more accurate recommendations*. GIISR mainly includes two stages: In the first stage, we construct two Graph Convolutional Networks (GCNs) to learn the relation of the intra-session items.Furthermore, in the second stage, we construct an inter-session graph, which also contains the intra-session relation fused from the first stage to aggregate the information between different sessions. Moreover, we employ the soft-attention mechanism and hybrid embedding to consider global preference and current interest. Finally, the probability of the next click item can be predicted.

The main contributions of this work are summarized as follows. 1) We propose a novel two-stage method: GIISR. It exploits not only the intra-session information but also the inter-session relations for session-based recommendations. 2) In the first stage of GIISR, a new graph convolutional network and one new modified residual gate neural network are presented for modeling intra-session information.

Thanks to the National Natural Science Foundation of China (Nos. 61876220, 61876221, 61976164, 61836009) for funding.

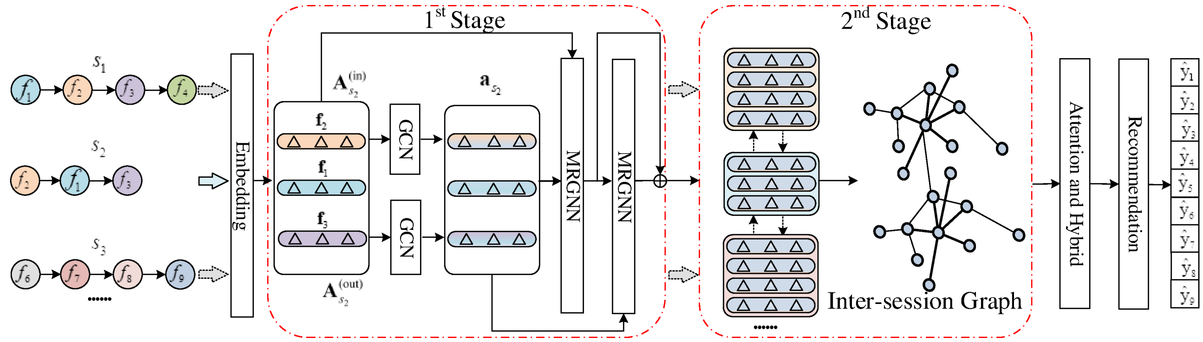


Fig. 1. The workflow of the proposed GIISR method. After embedding all the sessions, both new graph convolutional networks and Modified Residual Gated Neural Networks (MRGNNs) are constructed to learn intra-session information in the first stage. In the second stage, we aggregate the information between sessions with an inter-session graph. $\oplus$ denotes residual connection.

3) In the second stage, the inter-session relations are aggregated by our local sensitive inter-session graph. Moreover, a new soft-attention mechanism and hybrid embedding technique are employed to consider the global preference and current interest. 4) Experimental results on representative real-world datasets demonstrate that GIISR significantly improve the performance than existing state-the-of-art methods.

2. OUR INTRA- AND INTER-SESSION INFORMATION APPROACH

In this section, we depict our GIISR method in Fig. 1 in detail. GIISR mainly includes two stages. In the 1st stage, two single-layer graph convolution networks are proposed to learn intra-session information, and then modified residual gated networks (MRGNNs) are presented to further enhance this information. In the 2nd stage, we construct a new inter-session graph to aggregate information between different sessions.

Notations: Let $S = \{s_1, s_2, \cdots, s_n\}$ denote a set of sessions. A single session sequence s_i can be represented as $\{f_{i,1}, f_{i,2}, \cdots, f_{i,m}\}$, where $f_{i,m}$ is a clicked item ordered by timestamps and m is the number of items in this session. The purpose of our algorithm is to predict the next click, i.e., the item $f_{i,m+1}$. It outputs the probability values of all items, and thus the value of $\hat{y}$ is used to represent the prediction score of corresponding item. In $\hat{y}$, the top-M candidate items are viewed as recommendations.

2.1. Stage 1: Learning Intra-session Information

In current session, a click sequence can be viewed as a directed graph $\mathcal{G}_s = (\mathcal{F}_s, \mathcal{E}_s)$, where $\mathcal{F}_s$ and $\mathcal{E}_s$ are nodes and edges, respectively. An item can be represented by a node $f_{i,m} \in \mathcal{F}_s$. The relationship between items can be represented as $(f_{i,m-1}, f_{i,m}) \in \mathcal{E}_s$ in current session s_i. Given current session s_i, we construct an incoming graph $\mathbf{A}_{s_i}^{(\text{in})} \in \mathbb{R}^{m\times m}$ and an outgoing graph $\mathbf{A}_{s_i}^{(\text{out})} \in \mathbb{R}^{m\times m}$.

GCN. In the first stage of GIISR, we construct two graph convolutional networks (GCNs) for the intra-session embedding. The intra-session information for the incoming graph is calculated as follows:

$$\mathbf{a}_{s_i}^{(\text{in})} = \mathbf{A}_{s_i}^{(\text{in})} \left[\mathbf{f}_{i,1}, \ldots, \mathbf{f}_{i,m}\right]^\top \mathbf{H_1} + \mathbf{b_1}, \tag{1}$$

where $\mathbf{f}_{i,m} \in \mathbb{R}^d$ is the latent vector of node $f_{i,m}$, d is the dimension of the vector, $\mathbf{H_1} \in \mathbb{R}^{d\times d}$ is the weight, and $\mathbf{b_1}$ is the bias. Then the intra-session information for the outgoing graph is computed by

$$\mathbf{a}_{s_i}^{(\text{out})} = \mathbf{A}_{s_i}^{(\text{out})} \left[\mathbf{f}_{i,1}, \ldots, \mathbf{f}_{i,m}\right]^\top \mathbf{H_2} + \mathbf{b_2}, \tag{2}$$

where $\mathbf{H_2} \in \mathbb{R}^{d\times d}$ is the weight, $\mathbf{b_2}$ is the corresponding bias. Both Eqs. (1) and (2) are utilized to embed intra-session information between different items about incoming edges and outgoing edges, respectively. And we perform a concatenation of $\mathbf{a}_{s_i}^{(\text{in})}$ and $\mathbf{a}_{s_i}^{(\text{out})}$ as follows:

$$\mathbf{a}_{s_i} = [\mathbf{a}_{s_i}^{(\text{in})}; \mathbf{a}_{s_i}^{(\text{out})}]. \tag{3}$$

MRGNN. In order to strengthen the relationship between items, we propose a modified residual gated neural network (MRGNN) to further learn intra-session information. A M-RGNN with update gate $\mathbf{z}_{s_i}$ is represented as:

$$\mathbf{z}_{s_i} = \text{PReLU}\left(\mathbf{a}_{s_i}\mathbf{W}_z + \mathbf{f}_{i,:}^\top \mathbf{U}_z\right), \tag{4}$$

where $\mathbf{f}_{i,:}$ is the latent matrix of all the nodes in current session, $\text{PReLU}(\cdot)$ is an activation function [12], and $\mathbf{W}_z \in \mathbb{R}^{2d\times d}$ and $\mathbf{U}_z \in \mathbb{R}^{d\times d}$ are corresponding weights. Its reset gate $\mathbf{r}_{s_i}$ is:

$$\mathbf{r}_{s_i} = \text{PReLU}\left(\mathbf{a}_{s_i}\mathbf{W}_r + \mathbf{f}_{i,:}^\top \mathbf{U}_r\right), \tag{5}$$

where $\mathbf{W}_r \in \mathbb{R}^{2d\times d}$ and $\mathbf{U}_r \in \mathbb{R}^{d\times d}$ are corresponding weights. By using Eqs. (4) and (5), the features or information in session s_i have been mapped to a compact space. Note that we use PReLU function here, since the output of PReLU follows a zero-mean distribution, it can speed up the training process and alleviate the vanishing gradient problem compared with a conventional Sigmoid. Then in order to preserve the original information of items and utilize the learned information, the node $f_{i,:}$ is updated as

$$\mathbf{f}_{i,:}^{MG1} = \mathbf{z}_{s_i} \odot \mathbf{f}_{i,:}^{\top} + \mathbf{z}_{s_i} \odot \mathbf{r}_{s_i}, \tag{6}$$

where $\odot$ denotes element-wise multiplication.

Additionally, to further exploit the item relations in a single session, a residual connection is added at the end of the second MRGNN, as shown in Fig. 1.

$$\mathbf{f}_{i,:}^{MG2} = \mathbf{f}_{i,:}^{MG1} + MRGNN(\mathbf{f}_{i,:}^{MG1}). \tag{7}$$

2.2. Stage 2: Aggregating Inter-session Information

Most session-based recommendation methods only consider intra-session information in current session. A better recommendation method based on session should further consider inter-session information between different sessions. By learning intra-session item embedding, we obtain the vector representations of all the sessions, which contain intra-session information. Now we construct a new local sensitive inter-session graph, whose nodes are the session vectors: $s_1, s_2, ..., s_n$, and $\mathbf{s}_i = \left[\mathbf{f}_{i,1}^{MG2}, \dots, \mathbf{f}_{i,m}^{MG2}\right]$. The similarity $\mathbf{B}_{ij}$ between two session vectors $\mathbf{s}_i$ and $\mathbf{s}_j$ is defined as

$$\mathbf{B}_{ij} = \exp\left(\frac{-D^2\left(\mathbf{s}_i, \mathbf{s}_j\right)}{\theta_i \theta_j}\right), \tag{8}$$

where $D(\cdot)$ denotes the Euclidean distance, $\theta_i = D(\mathbf{s}_i, \mathbf{s}_k)$, and $\mathbf{s}_k$ is the k-th nearest neighboring session vector of $\mathbf{s}_i$. Note that θ_i and θ_j are the local scaling factors [13]. Then the new session vector $\mathbf{s}_i$ is represented as

$$\mathbf{g}_i = \mathbf{B}_{i:}\left[\mathbf{s}_1, \dots, \mathbf{s}_n\right]^{\top}, \tag{9}$$

where n is the number of sessions in each mini-batch.

2.3. Generating Embedding and Recommendation

In our GIISR method, we consider the global preference by the soft-attention mechanism after using intra-session and inter-session information. Through the two-stage information learning, the updated vector $\mathbf{g}_i$ is represented as: $[\mathbf{f}'_{i,1}, \mathbf{f}'_{i,2}, ..., \mathbf{f}'_{i,m}]$. Unlike [10], we utilize the two-stage information before using the soft-attention mechanism, and the soft-attention can be formulated as follows:

$$\mathbf{f}'_{i,g} = \sum_{j=1}^{m} \mathbf{q}^{\top} \sigma\left(\mathbf{W}_1 \mathbf{f}'_{i,m} + \mathbf{W}_2 \mathbf{f}'_{i,j}\right) \mathbf{f}'_{i,j}, \tag{10}$$

where $\mathbf{q} \in \mathbb{R}^d$, $\mathbf{W_1}, \mathbf{W_2} \in \mathbb{R}^{d\times d}$, and $\sigma(\cdot)$ is the sigmoid activation function. $\mathbf{s}_{i,m}$ represents the last-clicked item to consider current interest in a session. The global preference and current interests can be considered at the same time as hybrid embedding,

Table 1. Performance of GIISR and other baseline methods.

Method	Diginetica		Yoochoose 1/64		Yoochoose 1/4	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
POP	0.89	0.20	6.71	1.65	1.33	0.30
S-POP	21.06	13.68	30.44	18.35	27.08	17.75
Item-KNN [14]	35.75	11.57	51.60	21.81	52.31	21.70
BPR-MF [15]	5.24	1.98	31.31	12.08	3.40	1.57
FPMC [2]	26.53	6.95	45.62	15.01	-	-
GRU4REC [4]	29.45	8.33	60.64	22.89	59.53	22.60
NARM [8]	49.70	16.17	68.32	28.63	69.73	29.23
STAMP [16]	45.64	14.32	68.74	29.67	70.44	30.00
SR-GNN [10]	50.73	17.59	70.57	30.94	71.36	31.89
STAN [11]	-	-	69.45	28.74	70.07	28.89
GIISR(ours)	**52.48**	**18.55**	**70.92**	**32.06**	**71.95**	**32.57**

$$\hat{\mathbf{s}}_{i,j} := {\mathbf{f}'_{i,j}}^{\top} \mathbf{W}_3(\mathbf{f}'_{i,m} + \mathbf{f}'_{i,g}), \tag{11}$$

where $\mathbf{W_3} \in \mathbb{R}^{d\times d}$. $\hat{\mathbf{s}} \in \mathbb{R}^{|\mathcal{F}_s|}$ is the scores of all recommendation items. Then we utilize the Softmax function to obtain the output for our model: $\hat{\mathbf{y}} = \mathrm{softmax}(\hat{\mathbf{s}})$. Finally, the probability of the item $\hat{\mathbf{y}}_\mathbf{i} \in \hat{\mathbf{y}}$ in next click can be calculated by the following cross-entropy loss function:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{|\mathcal{F}_s|} \mathbf{y}_i \log\left(\hat{\mathbf{y}}_i\right) + \left(1 - \mathbf{y}_i\right) \log\left(1 - \hat{\mathbf{y}}_i\right), \tag{12}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{F}_s|}$ is the output of the model of our model.

3. EXPERIMENTS AND ANALYSIS

3.1. Datasets and Data Preparation

We evaluate our GIISR method on two representative real-world datasets, i.e., Diginetica[1] and Yoochoose[2]. For fair comparison, following [8, 10], we initialize all parameters by a Gaussian distribution, whose mean is 0 and standard deviation is 0.01. Besides, all parameters are optimized by the mini-batch Adam optimizer with a learning rate of 0.001 and decay by 0.1 after each 3 epochs. The number of epochs is 30. We randomly select 10% of the training set as the validation set to tune other hyper-parameters. The L2 penalty and the dimensionality of latent vectors are set to 10^{-5} and 100, respectively. We conduct all the experiments on a server with 64 GB memory, and one GeForce GTX 1080 Ti.

3.2. Comparison with Baseline Algorithms

In order to demonstrate the performance of our GIISR method, we compare it with several state-of-the-art session-based recommendation methods in terms of P@20 and MRR@20. The experimental results are shown in Table 1, where

[1]http://cikm2016.cs.iupui.edu/cikm-cup
[2]http://2015.recsyschallenge.com/challege.html

the best performance is highlighted in boldface. Note that the results of FPMC on Yoochoose 1/4 and STAN on Diginetica are not reported as in their work.

It is clear that the proposed GIISR method achieves the best performance on all these datasets. For example, it attains 32.06 of MRR@20 on the Yoochoose 1/64 dataset, while the second best one is SR-GNN, which is only 30.94. That is, our GIISR outperforms SR-GNN as much as 3.62%. The reason is probably that our GIISR exploits both the intra-session and inter-session information for recommendation, while SR-GNN only employs the intra-session information.

The results show that the performance of both POP and S-POP are relatively lower. Nevertheless, S-POP is superior to the methods like POP, BPR-MF and FPMC, because it takes into consideration of session contextual information. Although Item-KNN only considers the similarity of items in current session, obtaining better results than FPMC. STAN utilizes the item relevance and session-level similarity, and outperforms general deep-learning methods, demonstrating the significance of both the intra-session information in current session and inter-session similarity in all the sessions. The performance of NARM and GRU4REC are better than traditional methods, which indicates the superiority of deep learning-based methods. They utilize RNNs to obtain user's general interests. STAMP further improves the performance by enhancing short-term memory, which uses the last click in current session. Compared with SR-GNN, our GIISR employs GCN and MRGNN, which renders it to learn intra-session information in current session. And GIISR adopts soft-attention mechanism and hybrid embedding to address global preference and current interests, which makes it automatically select the most important items.

Table 2. Performance comparison of SR-GNN and GIISR.

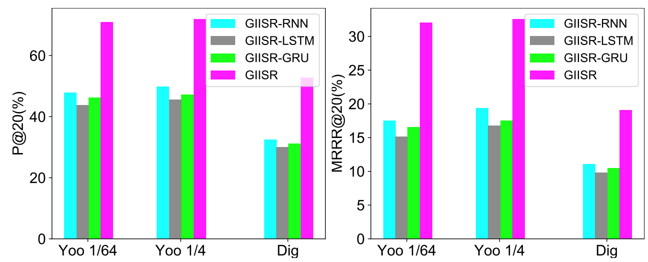
Method	Dataset	Time (seconds)	P@20	MRR@20
SR-GNN [10]	Diginetica	890	50.73	17.59
	Yoochoose 1/64	739	70.57	30.94
	Yoochoose 1/4	12021	71.36	31.89
GIISR (ours)	Diginetica	811	**51.75**	**17.96**
	Yoochoose 1/64	700	**70.76**	**31.29**
	Yoochoose 1/4	10980	**71.39**	**31.96**

3.3. Comparison with GNN-based Methods

We also compare the efficiency of our GIISR with SR-GNN, as shown in Table 2. Note that here we only show the results of our GIISR with the first stage technique for recommendation. As our GIISR uses graph convolutional networks and MRGNNs, and SR-GNN also adopt GNN, we compare their performance in each training epoch on the three datasets. Moreover, the results of both P@20 and MRR@20 are also reported. Note that the batch size of all the datasets is set to 100 for fair comparison. It can be seen that GIISR costs less running time for each training epoch than that of SR-GNN on all these datasets. It can be also observed that larger datasets need longer training time. In particular, GIISR gains superior results in terms of the metrics for all the datasets.

3.4. Analysis on the Intra-session Learning

In order to evaluate the effectiveness of our MRGNN in learning intra-session information in our GIISR method, we add different networks after our GCN in the architecture, and compare their performance in Fig. 2. Our GIISR with three different networks are: (1) GIISR-RNN; (2) GIISR-LSTM (Long Short-term Memory [17]); (3) GIISR-GRU (Gated Recurrent Unit [18]), respectively. In this experiment, we fix the batch size to 100, and $k = 10$ for different networks.



(a) P@20 (b) MRR@20

Fig. 2. Performance comparison of our GIISR with different learning networks.

The results show that our GIISR outperforms GIISR-RNN, GIISR-LSTM and GIISR-GRU. The results of P@20 and MRR@20 of GIISR on the three datasets are the highest. It is probably that the proposed method can obtain more accurate representation for intra-session information. For GIISR-RNN, GIISR-LSTM and GIISR-GRU, they adopt different gate mechanisms in the network. They can reinforce intra-session information by our sequential method. However, they are inferior to our GIISR with MRGNN. It is likely because that our MRGNNs can better map the intra-session information through residual connection.

4. CONCLUSION

Session-based recommendation is important to the user whose historical information does not exist. In this paper, we proposed a novel method named GIISR to consider both intra- and inter-session information. In its first stage, we first presented a new graph convolutional network, and then proposed a modified residual gated network, which together learn intra-session information between different items. In the second stage, we constructed an inter-session graph to aggregate the relation between different sessions. Extensive experiments were conducted to compare the performance of our method with those of the state-of-the-art ones. All the results show that GIISR outperforms the second best method as much as 3.62 % in terms of MRR @20 on the Yoochoose dataset. In the future, more strategies will be studied to further enhance the performance of our method.

5. REFERENCES

[1] Guy Shani, David Heckerman, and Ronen I Brafman, "An mdp-based recommender system," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1265–1295, 2005.

[2] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[3] Wanrong Gu, Shoubin Dong, and Zhizhao Zeng, "Increasing recommended effectiveness with markov chains and purchase intervals," *Neural Computing and Applications*, vol. 25, no. 5, pp. 1153–1162, 2014.

[4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and D Tikk, "Session-based recommendations with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[5] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 241–248.

[6] Yong Kiam Tan, Xinxing Xu, and Yong Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 17–22.

[7] Dietmar Jannach and Malte Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 306–310.

[8] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[9] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1831–1839.

[10] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 346–353.

[11] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff, "Sequence and time aware neighborhood for session-based recommendations: Stan," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1069–1072.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[13] Lihi Zelnik-Manor and Pietro Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2005, pp. 1601–1608.

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[15] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[16] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1831–1839.

[17] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

SUPPLEMENTARY MATERIALS FOR GRAPH NEURAL NETWORKS WITH INTRA- AND INTER-SESSION INFORMATION FOR SESSION-BASED RECOMMENDATION

Hongying Liu, Tianwen Zhu, Fanhua Shang, Guoliang Ma, Muyao Liu, Yuanyuan Liu*

School of Artificial Intelligence, Xidian University

1. RELATED WORK

In this section, we briefly review some representative related work on session-based recommendations including the conventional methods, and sequential methods. Moreover, some graph neural networks are also introduced.

1.1. General Recommendation

General recommendation is usually implemented on the base of item-to-item recommendation methods. Items that frequently occur together within sessions are considered to be similar. In this setting, a similarity matrix for the item-to-item relation is computed previously with the utilization of session data. For instance, the collaborative filtering method [1] is proposed to capture user's general interests. In addition, the matrix factorization methods [2, 3, 4] are utilized by factorizing the rating matrix of user-item into two low-rank matrices. Each of the low-rank matrices is used to represent general interests with latent vectors of users or items.

1.2. Sequential Recommendation

Sequential recommendation methods use the previous sequential connections between the user behaviors to predict users' next action based on Markov chains and deep learning models. Based on Markov chains, [5] employs Markov decision processes and uses the transition probabilities between items to compute the next recommendation. FPMC [6] can obtain a better prediction result through the factorization of users' personalized probability transition matrices, considering sequential behavior between each two adjacent clicks.

Deep learning has been applied successfully in various fields, such as natural language processing [7, 8], conversation machine [8]. Among numerous deep neural networks, it has been proven that the recurrent neural networks (RNNs) are effective in sequential data, such as sequential next prediction [9], location prediction [10], and next basket recommendation [11]. [12] is the first work to use RNNs for solving session-based recommendation problems. [13] uses parallel RNNs to solve this problem. Both [12] and [13] are fine for modeling sequential data because they can consider the historical behavior of users in predicting the next click. Subsequently, [14] is proposed to improve the performance of session-based recommendation with RNNs, which uses a data augmentation technique and takes into consideration temporal shifts of user behaviors. In [11], RNNs are utilized to obtain the dynamic representation from each basket based on dynamic recurrent model. [15] combines the neighborhood-based method with RNNs to utilize the sequential information. NARM [16] employs the attention mechanism to obtain main purpose of users from the hidden states based on RNNs. STAMP [17] is proposed to obtain users' current interests and general interests.

1.3. Graph Neural Networks

In recent years, graph neural networks (GNNs) have achieved significant development in various areas, e.g., social networks [18, 19], molecular structure [20], knowledge graphs [21]. Any data can be represented as nodes and be constructed edges. DeepWalk [22] is regarded as the first graph embedding method according to the idea of representation learning and word2vec [23], using skip-gram model on the random walks. Meanwhile, other methods such as node2vec LINE [24], TADW [25] and [26] are the most representative.

As for nodes embedding, convolutional methods on graph neural networks can be divided into two categories: spectral and non-spectral domain. Graph convolutional network [18] simplifies the approximation of graph Laplacian. Graph Neural Networks (GNNs) [27, 28] are proposed to generate graph representation. [29] employs gated recurrent units. Different from the tasks handled by Convolutional Neural Networks (CNNs), GNNs focus on graph representation [30], node classification [18] and link prediction [31]. So far, GNNs have been applied in many fields such as image segmentation [32], action recognition [33], and session-based recommendation system [34, 35].

2. OUR INTRA- AND INTER-SESSION INFORMATION APPROACH FOR SESSION-BASED RECOMMENDATION

In this section, we depict our GIISR method in detail.

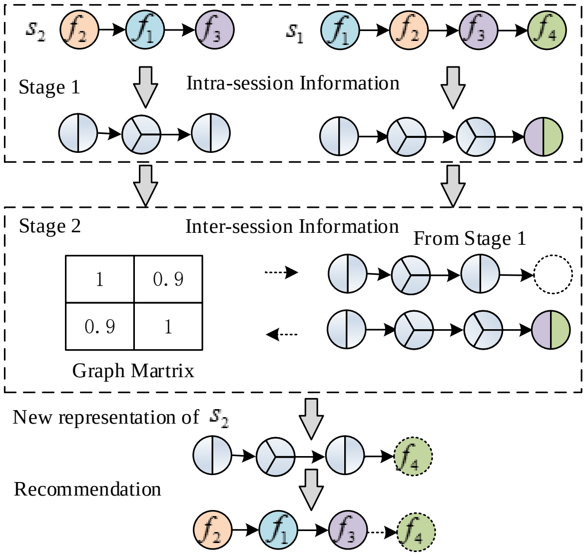


Fig. 1. An example of information propagation in our two stage method GIISR.

2.1. Two-stage Information Propagation

An example of the two-stage information propagation in our method is illustrated in Figure 1. In Stage 1, two sessions: $s_1 = \{f_1, f_2, f_3, f_4\}$ and $s_2 = \{f_2, f_1, f_3\}$ are processed by GCN and MRGNN. Then the intra-session information has been fused. For example, as f_1 follows f_2 in session s_2, it may contain information of f_2. And similarly, in session s_1, f_2 may fuse information from f_1 and f_3. In Stage 2, the session vectors have similarities as they include intra-session information from Stage 1. Therefore, we can calculate the similarity between sessions and construct an inter-session graph matrix. Then through soft-attention, the final recommended item is attained as f_4 for session s_2.

3. EXPERIMENTAL DETAILS

3.1. Implementation Details

The details of the experimental datasets are shown here. The Yoochoose dataset is released by RecSys Challenge 2015, which consists of a series of user-click events data from an e-commerce website in six months. The Diginetica dataset is obtained from CIKM Cup 2016, for which only the transactional data is used.

For the sake of fair comparison, following [16, 34], we use the items with length larger than 1 and appearing greater than 4 times in all datasets. So that the Yoochoose dataset consists of 7,981,580 sessions and 37,483 items, while the Diginetica dataset consists of 204,771 sessions and 43097 items. Following [14], we use all the session $S = \{s_1, s_2, \cdots, s_n\}$ to split the input sequences and corresponding labels, where n is the number of sessions. The single input sequence is represented as $s_i = \{f_{i,1}, f_{i,2}, \cdots, f_{i,m}\}$, where m is the length of the session s_i. And thus, we generate sequences and labels as $([f_{i,1}], f_{i,2}), ([f_{i,1}, f_{i,2}], f_{i,3}), \cdots, ([f_{i,1}, f_{i,2}, \cdots, f_{i,m-1}], f_{i,m})$, where $[f_{i,1}, f_{i,2}, \cdots, f_{i,m-1}]$ is the sequence and $f_{i,m}$ denotes the label. Following [17, 34], we adopt the common fractions 1/64 and 1/4 for Yoochoose dataset. The statistics of these datasets are shown in Table 1.

Table 1. Statistics of datasets used in this study.

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

3.2. Baseline Methods

POP and S-POP recommend the top-N popular items on the frequency of their occurrences in the training set. Even though they are simplicity, the performance on real-world datasets proved it a strong baseline in certain domains.

Item-KNN [36] recommends the candidate item that is similar to existing items based on the Cosine similarity between them in the sessions. Besides, in order to get rid of the rarely clicked items with coincidental high similarities, a regularization is also utilized in this process.

BPR-MF [37] utilizes stochastic gradient descent to optimize a pairwise ranking objective funciton. The mean of similarities between the candidate's and the existing items' latent factors are used to evaluate the score of recommendation.

FPMC [6] is a hybrid model, predicting sequentially based on Markov chain for the next-basket recommendation. Besides, the latent representations of users will not be considered when evaluating the scores of recommendation.

GRU4Rec [12] consists of many GRU units, which constitutes a deep RNN model to predict user sequences for session-based recommendation. A session-parallel mini-batch training process, as well as ranking-based loss functions, are employed to promote the training.

NARM [16] utilizes the attention mechanism to obtain main purpose from the hidden states of RNN. Then the recommendations are generated by the final representation, which combines with the sequential behavior.

STAMP [38] employs both of users' overall interests for the current session and their interests for the last click currently.

STAN [39] employs recent session and the position of an item in a current session. In inter-session, the recommendable position of the item is also used.

SR-GNN [34] utilizes a GNN and soft-attention mechanism to consider the complex transitions of items.

Table 2. Performance of GIISR with different number of inter-sessions.

Dataset	k=7		k=8		k=9		k=11		k=15		k=20	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
Yoochoose 1/64	70.32	31.07	70.81	31.69	**70.92**	**32.06**	70.67	31.42	70.41	31.26	70.00	30.93
Yoochoose 1/4	71.37	31.96	71.42	32.13	71.62	32.26	71.64	32.29	**71.95**	**32.57**	71.47	32.13
Diginetica	52.31	18.39	52.26	18.37	52.40	18.44	52.41	18.53	**52.48**	**18.55**	52.40	18.62

Both P@20 and MRR@20 are widely used as evaluation metrics. P@20 (Precision) represents the proportion of all test cases which has correctly recommended items amongst the top-20 items. MRR@20 (Mean Reciprocal Rank) is the average of reciprocal ranks of all desire items. We set the reciprocal rank to 0 if the rank exceeds 20. It is important in the field of recommendation systems because it takes into account the ranking order of recommendation list.

3.3. Analysis on the Similarity Metrics for Inter-session Graph

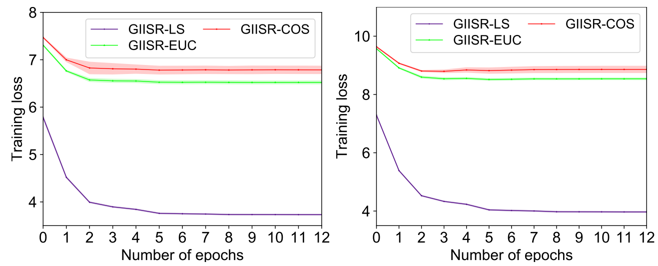
In this subsection, we analyse our GIISR with different similarity metrics for constructing the inter-session graph as shown in Table 3. The following three metrics are compared: (1) similarity with the cosine distance (GIISR-COS); (2) similarity with the Euclidean distance (GIISR-EUC); (3) Euclidean distance is calculated at first, and then utilize local-scaling method to compute the similarity (GIISR-LS).This is also our original GIISR. Note that we fix batch size to 100, and $k = 10$ for all these datasets.

It can be observed that the performance of GIISR-LS is superior to the other two methods. The values of P@20 and MRR@20 are higher than GIISR-COS and GIISR-EUC. This validates the effectiveness of our GIISR with the metric of local scaling. Furthermore, the table shows that the performance of GIISR-ECU is better than GIISR-COS, which indicates that the Euclidean distance is more suitable for these relative low-dimensional datasets.

In addition, the training loss of GIISR-COS, GIISR-EUC, and GIISR-LS are shown in Figure 2. It can be seen that the training loss of both GIISR-COS and GIISR-EUC is much higher than GIISR-LS, which demonstrates that the GIISR with local scaling metric obtains better convergence results. Moreover, the standard errors of each studied metric show that the GIISR-LS is more stable than other metrics. This also verifies the effectiveness of the local scaling metric for construction of the inter-session graph.

3.4. Analysis on the Number of Inter-sessions

In this subsection, we analyze the influence of the number of inter-sessions k (in Eq.(8) in the main paper) on performance of GIISR in Table 2. We also fix batch size to 100. It can be seen that the optimal number of inter-sessions varies with different datasets. For Yoochoose 1/64, when the value of k is around 9, the performance is highest. And when k is around 15, the performance of Yoochoose 1/4 and Diginetica is better than that of other settings. In summary, the value of k can be set around 10 for other datasets.



(a) Yoochoose 1/64 (b) Diginetica

Fig. 2. The training loss of GIISR with different similarity metrics. Note that the standard error is also plotted for each case.

Table 3. The performance of GIISR with different similarity metrics in constructing inter-session graph.

Method	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
GIISR-COS	25.70	6.47	24.58	6.48	5.35	1.18
GIISR-ECU	31.23	8.05	33.45	11.38	8.29	1.90
GIISR-LS	**70.86**	**31.61**	**71.67**	**32.42**	**52.47**	**18.87**

Table 4. The performance of GIISR with different structure in the inter-session stage.

Method	Diginetica		Yoochoose 1/64	
	P@20	MRR@20	P@20	MRR@20
GIISR-FCN	20.08	3.8	30.82	19.10
GIISR (Graph)	**52.48**	**18.55**	**70.92**	**32.06**

3.5. The Ablation of the Graph in Inter-session Stage

In the second stage of our GIISR, we utilize a local sensitive inter-session graph to aggregate the relation between different sessions. In order to show its effectiveness, we replace the graph with a three-layer fully connected neural network (FCN). Note that consider the size of the datasets, we do not use deeper networks for this experiment. The experimental results are provided on yoochoose 1/64 and yoochoose 1/4 datasets, listed in Table 4. Obviously, the performance of our GIISR with the graph is much higher than that with a three-layer neural network, since our local sensitive inter-session graph is devised for accurate computation for the similarity between sessions. This result confirmed the importance and the effectiveness of our local sensitive inter-session graph in the second stage.

4. REFERENCES

[1] Greg Linden, Brent Smith, and Jeremy York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[2] Andriy Mnih and Russ R Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[3] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[4] Yehuda Koren and Robert Bell, "Advances in collaborative filtering," in *Recommender systems handbook*, pp. 77–118. Springer, 2015.

[5] Guy Shani, David Heckerman, and Ronen I Brafman, "An mdp-based recommender system," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1265–1295, 2005.

[6] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[8] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[9] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu, "Sequential click prediction for sponsored search with recurrent neural networks," in *AAAI*, 2014.

[10] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[11] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 729–732.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and D Tikk, "Session-based recommendations with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[13] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 241–248.

[14] Yong Kiam Tan, Xinxing Xu, and Yong Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 17–22.

[15] Dietmar Jannach and Malte Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 306–310.

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[17] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1831–1839.

[18] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *In ICLR*, 2016.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[20] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur, "Protein interface prediction using graph convolutional networks," in *Advances in neural information processing systems*, 2017, pp. 6530–6539.

[21] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto, "Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach," *arXiv preprint arXiv:1706.05674*, 2017.

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[23] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, vol. 2013, 01 2013.

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[25] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang, "Network representation learning with rich text information.," in *IJCAI*, 2015, vol. 2015, pp. 2111–2117.

[26] Aditya Grover and Jure Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[27] Marco Gori, Gabriele Monfardini, and Franco Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* IEEE, 2005, vol. 2, pp. 729–734.

[28] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[29] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel, "Gated graph sequence neural networks," *Computer Ence*, 2015.

[30] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in neural information processing systems*, 2018, pp. 4800–4810.

[31] Muhan Zhang and Yixin Chen, "Link prediction based on graph neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 5165–5175.

[32] Pedro F Felzenszwalb and Daniel P Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921.

[34] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 346–353.

[35] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin, "Exploiting cross-session information for session-based recommendation with graph neural networks," 2020, vol. 38, pp. 1–23, ACM Transactions on Information Systems.

[36] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[37] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[38] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1831–1839.

[39] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff, "Sequence and time aware neighborhood for session-based recommendations: Stan," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1069–1072.